
Conception d'une Chaîne de Traitement de la Langue Naturelle pour un Agent Conversationnel Assistant

François Bouchet

Soutenance de thèse
29 juin 2010

Plan

- **Partie I : État de l'art**
 - Novices et assistance
 - Systèmes de dialogue
 - Agents conversationnels animés pour l'assistance

- **Partie II : Chaîne de traitement**
 - Principes, version antérieure et besoins
 - Constitution et étude du corpus Daft
 - L'analyseur syntaxico-sémantique : GRASP
 - Le langage formel de requêtes : DAFT
 - Le générateur automatique de requêtes formelles : DIG

- **Partie III : Validations et applications**
 - Identification automatique d'activités conversationnelles
 - Agents Web DIVA
 - Vers un agent rationnel et comportemental

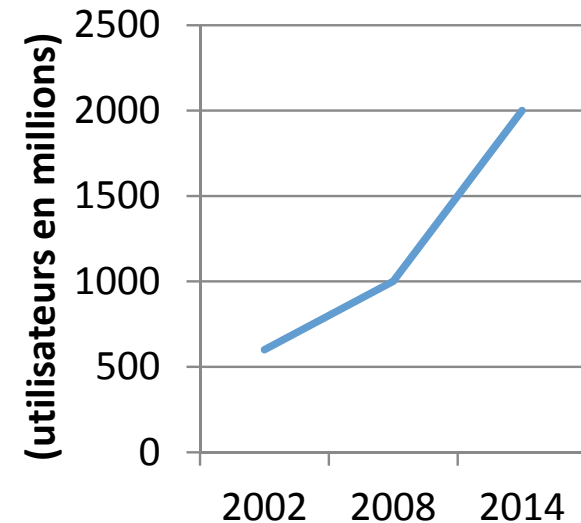
Plan

- **Partie I : État de l'art**
 - Novices et assistance
 - Systèmes de dialogue
 - Agents conversationnels animés pour l'assistance

- **Partie II : Chaîne de traitement**
- **Partie III : Validations et applications**

Problématique

- De plus en plus...
 - d'utilisateurs d'applications et services informatiques
 - de périphériques :
PC, consoles, téléphones, GPS, tablettes...
 - d'applications à disposition des novices :
200 000 sur l'AppStore, 50 000 sur l'Android Market...
 - De moins en moins... de temps pour apprendre
- ➔ L'assistance aux usagers novices est donc plus que jamais un sujet d'actualité
- Comment l'améliorer ?
 - Utilisation de la **langue naturelle** : moyen d'expression privilégié, en particulier des novices, en cas d'échec [Capobianco & Carbonell, 2001]
 - Utilisation d'un **agent conversationnel animé** : pour augmenter les chances d'utilisation de l'assistance et lutter contre le manque de motivation



Qui sont les utilisateurs novices ?

- **3 classes d'utilisateurs** d'un logiciel [Shneiderman, 1987] :
 - **Experts** : compétences informatiques et utilisation fréquente
 - **Occasionnels** : compétences informatiques et utilisation rare
 - **Novices** : peu de compétences informatiques
- **Qu'est-ce qu'un « novice » ?**
 - Peu ou pas de compétences en informatique [Shneiderman, 1987] [Goodwin, 1987]
 - < 10 heures d'expérience avec un système [Cohill & Williges, 1985]
 - Sans connaissances préalables du système [Fisher, 1991]
- **« Métaphore du bureau » non applicable :**
 - Non familier [Carroll & Rosson, 1987]
 - Génère des attentes non pertinentes [Streitz, 1988]

Nous considérerons « novice » au sens le plus large possible

Aide en ligne statique

- **Apports :**

- + Toujours accessible facilement
- + Recherche par mots-clés
- + Liens hypertextes

- **Limites :**

- « **Paradoxe de la motivation** » [Carroll & Rosson, 1987] :
volonté d'une aide immédiate et non optimale [Maïs, 1989]
- **Intrusivité** et confusion cognitive [Sebillotte & Scapin, 1994]
- Manque de proactivité : besoin d'un guide [Cohill & Williges, 1985]
- **Fossé cognitif** : modèle réel \neq modèle mental du novice
[Mack et al., 1983]

Systemes d'aide contextuelle

- **Adaptation à l'utilisateur :**
 - + Stéréotypes (classes)
 - + Modélisation des connaissances et de leur évolution [Paiva & Self, 1994]
 - Détermination des buts difficile (traces d'interaction, MOZ)
- **Adaptation à la tâche :**
 - + Automate à états finis → proactivité [Dzida et al., 1987]
 - + Plans sous-optimaux [Maïs, 1989]
- **Adaptation à l'application :**
 - + Détermination du modèle cognitif de l'utilisateur [Wærn, 1990]
 - + Evaluation du fossé cognitif [Leray & Sansonnet, 2007]
 - **Mais dans tous les cas, pas d'accroissement de l'utilisation de l'aide**

Aides multimodales et dialogiques

En entrée :

- + Modalité différente renforce la distinction « tâche à accomplir » vs « recherche d'assistance »
[Capobianco & Carbonell, 2006]
- **Langue naturelle écrite :**
 - + Intrusivité faible (champ texte)
 - + Pas d'apprentissage préalable de mots-clés
- **Langue naturelle parlée :**
 - + Impact faible sur réalisation de la tâche
 - + Intuitif pour utilisateurs
 - Limites de reconnaissance vocale non contrainte
- **Déictique :**
 - + Aide à la désambiguïsation des références langagières et anaphores

En sortie :

- **Synthèse vocale :**
 - + si en complément du texte
[Harrison, 1995]
 - pour instructions longues
[Carbonell & Kieffer, 2005]
- **Illustrations :**
 - + icônes en complément
[Morrell & Park, 1993]
 - + état du système [Harrison, 1995]
 - empiètement sur espace de travail
- **Animations :**
 - + augmente la mémorisation à long terme
[Palmiter & Elkerton, 1991]
 - + mise en évidence d'une zone
[Bederson et al., 1996]
 - inutile pour mouvements du curseur
 - conduit à une simple imitation

La langue naturelle une modalité prometteuse pour l'amélioration de l'assistance

Systemes de Dialogue Homme-Machine : quelques exemples

- **Tâches fortement contraintes** : gestionnaire de dialogue de type « machine à états finis »
 - Consultation d'horaires de cinéma : **MIMIC** [Chu-Carroll, 2000] ou de voyages : **GALAXY** [Goddeau et al., 1994] et **PHILIPS** [Aust et al., 1995]
 - Réservation de billets : **Colorado** [Pellom et al., 2001], **MERCURY** [Seneff & Polifroni, 2000]
- **Applications complexes** : **besoin d'un agent rationnel**, utilisant ses connaissances pour choisir ses actions afin d'atteindre ses buts
 - **SPA** [Wobcke et al., 2006] : agent de type BDI [Rao & Georgeff, 1991]
 - **TRAINS** [Allen, 1995] et **TRIPS** [Ferguson & Allen, 1998] : agents multiples communiquant entre eux en KQML [Finin et al., 1993] et avec l'utilisateur en langue naturelle pour résoudre ensemble un problème
 - **TRINDI** [Larsson & Traum, 2001] : moteur de dialogue basé sur le concept d'état informationnel, réutilisé dans des humains virtuels [Traum et al., 2002]
 - **COLLAGEN** [Rich & Sidner, 1997] : système de planification mais pas de langue naturelle
- **Multi-applications** : vers une certaine **généricité**
 - **OZONE** [Gaiffe et al., 2004] : pas d'agent, utilisation d'une version allégée de la SDRT
 - **ICAI** [Paraiso et Barthès, 2004] : approche dérivée d'agents BDI

Agents conversationnels animés pour l'assistance

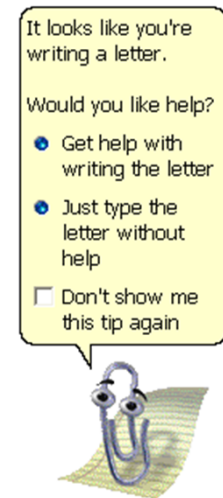
■ Avantages :

- « Persona Effect » [Lester et al., 1997] : **augmentation de la crédibilité** du système
- **Amélioration de la motivation** et de la mémorisation [Moreno & Mayer, 2000], même en l'absence de vrai dialogue humain-agent [Atkinson, 2000]
- Peut-être surtout un effet de la multimodalité, puisque l'ajout d'une voix de synthèse suffit parfois [Krämer et al., 2003]

■ Un risque :

- le « Clippy Effect » : génération d'attentes supérieures, et donc **sentiment de frustration renforcé** si les résultats ne sont pas à la hauteur [Xiao et al., 2004]

Bien qu'à double tranchant, un agent conversationnel animé semble être un moyen d'accroître l'utilisation de l'assistance

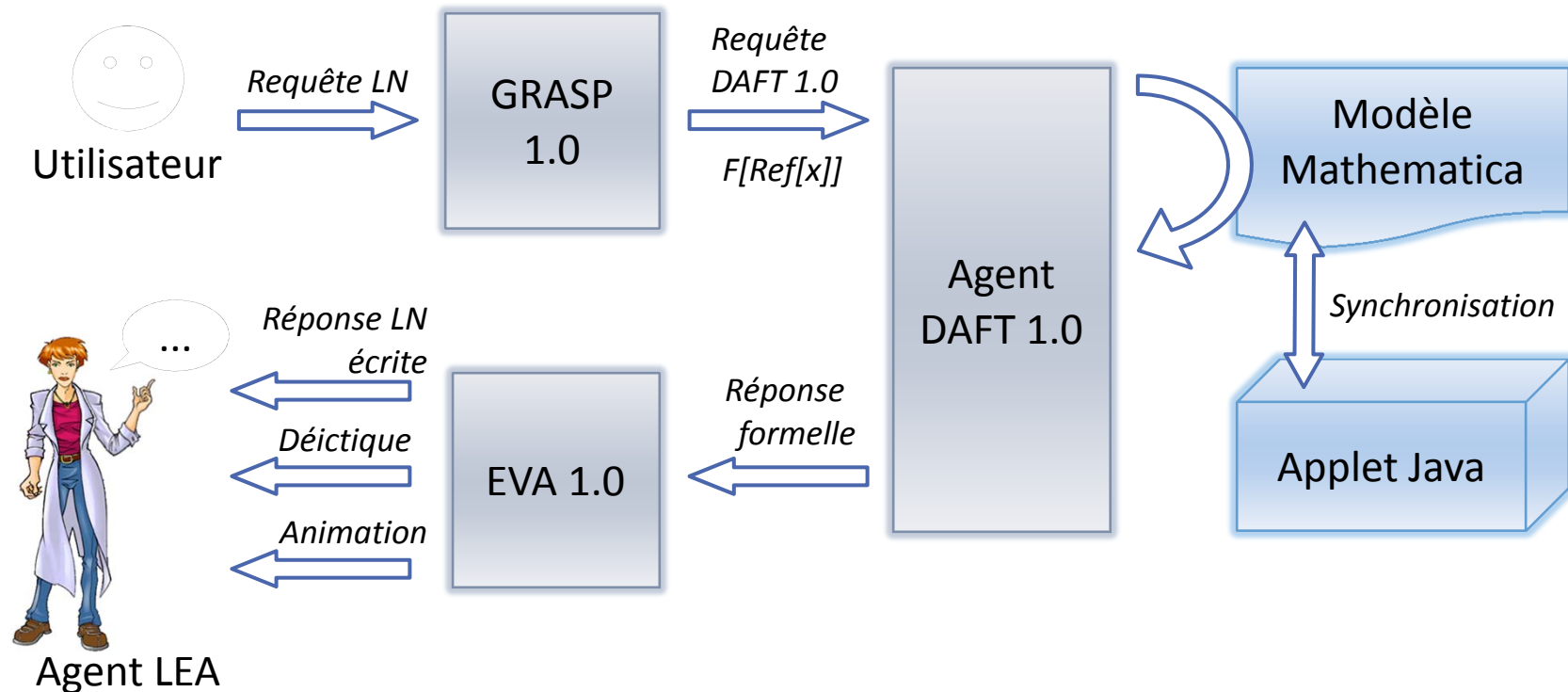


Plan

- **Partie I : État de l'art**
- **Partie II : Chaîne de traitement**
 - **Principes, version antérieure et besoins**
 - Constitution et étude du corpus Daft
 - L'analyseur syntaxico-sémantique : GRASP
 - Le langage formel de requêtes : DAFT
 - Le générateur automatique de requêtes formelles : DIG
- **Partie III : Validations et applications**

DAFT 1.0

- **Objectif du projet DAFT** : développement d'agents conversationnels assistants pour les utilisateurs novices, à la suite de [Sabouret, 2002] [Gérard, 2003] [Pitel, 2004]
- **Architecture du système en 2006** :



De DAFT 1.0 vers DAFT 2.0

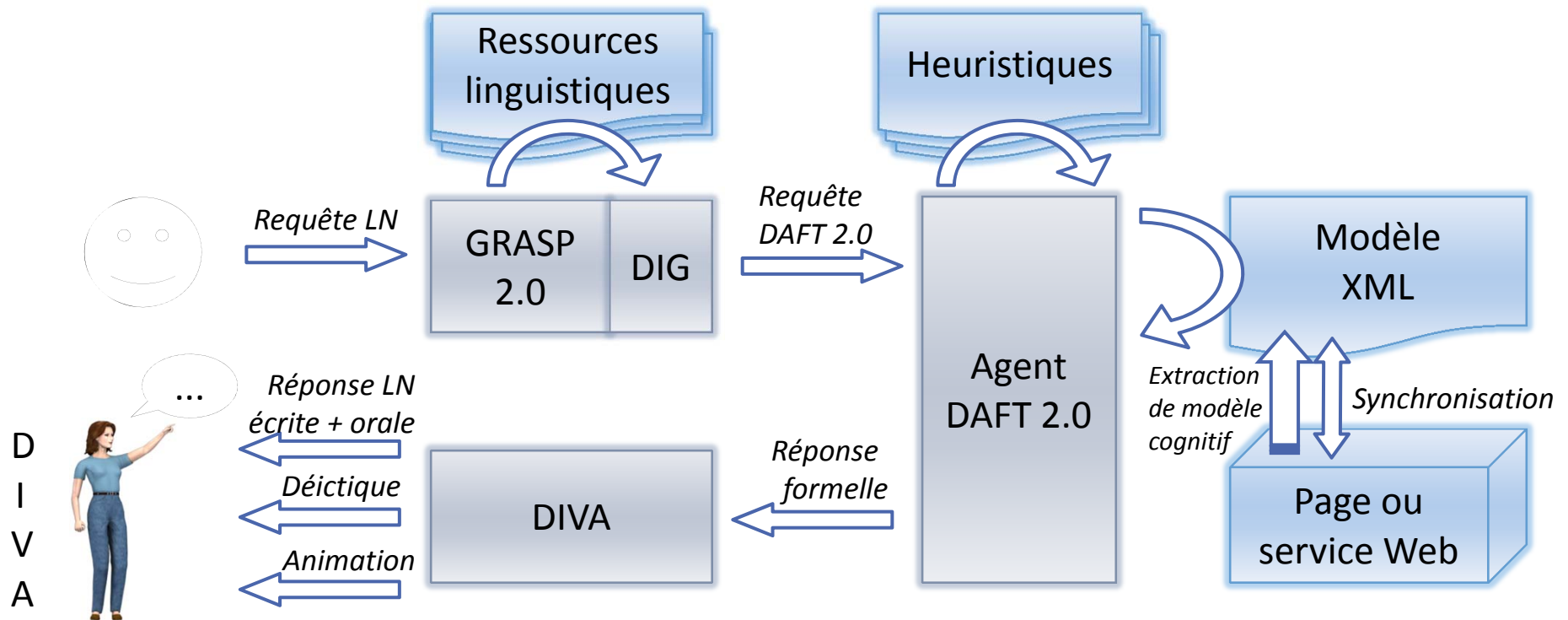
Limites de DAFT 1.0 :

- Phrases d'exemples arbitraires
→ manque de robustesse
- Structure monolithique des modules GRASP et DAFT
→ évolutivité difficile
- Structure basique du langage DAFT 1.0 et 23 classes d'actes de langage
→ incapacité à traiter toutes les requêtes d'assistance
- Modèle fermé (Mathematica) pour applets (Java) en perte de vitesse
- Agent style 2D cartoon (WebLea) issu de NICE [Buisine & Martin, 2005]
→ impact négatif sur apprentissage [Baylor & Kim, 2004]

Objectifs pour DAFT 2.0 :

- + Constitution préalable d'un **corpus de requêtes** d'assistance
- + **Externalisation des ressources** linguistiques et des heuristiques de réaction
- + Proposition d'un langage plus riche (DAFT 2.0) et développement d'un **module de génération de requêtes** en ce langage (DIG)
- + **Format ouvert** du modèle (XML) et passage au couple HTML/JavaScript
- + Passage à un agent DIVA type **3D réaliste**

DAFT 2.0



On s'intéressera essentiellement à l'analyse syntaxico-sémantique

Plan

- **Partie I : État de l'art**
- **Partie II : Chaîne de traitement**
 - Principes, version antérieure et besoins
 - **Constitution et étude du corpus Daft**
 - L'analyseur syntaxico-sémantique : GRASP
 - Le langage formel de requêtes : DAFT
 - Le générateur automatique de requêtes formelles : DIG
- **Partie III : Validations et applications**

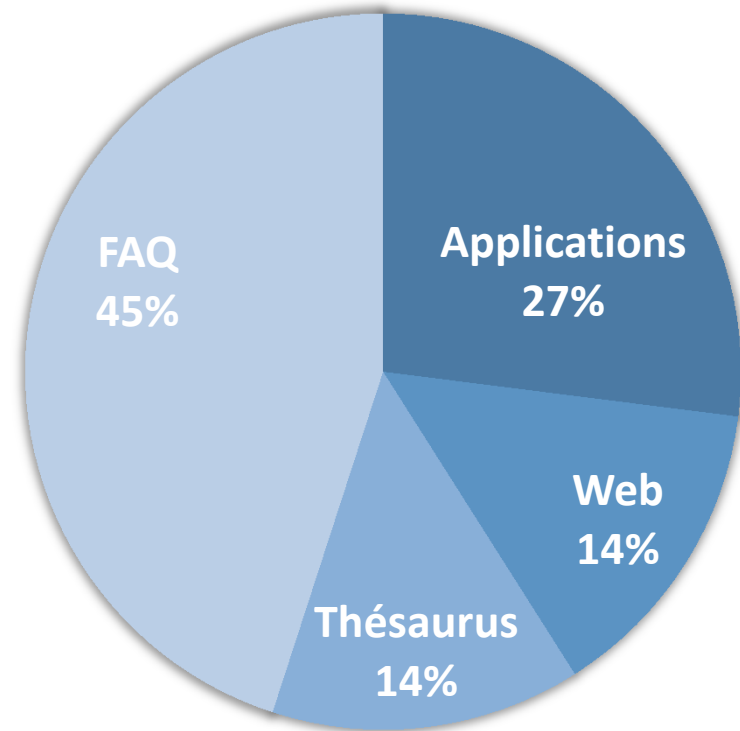
Corpus Daft : constitution (1)

- **Opération coûteuse :**
 - Techniquement : version préliminaire ou MOZ
 - Humainement : trouver les sujets novices
- **Contraintes :**
 - **Modalité écrite :**
 - Modalité orale peu répandue sur PC
 - Oral potentiellement déstabilisant [Ibister et al., 2000]
 - **Langue française :** peu de corpus disponibles
 - Nécessaire pour l'analyse grammaticale
 - Facilite la recherche d'utilisateurs novices
 - **Interaction Homme-Machine :** différente de homme-homme
 - Différences de comportement [Morel 1989 ; Luzzati, 1995]
 - Différences d'attentes [Amalberti et al., 1993]
 - Transposabilité des requêtes ?
 - **Requêtes isolées :** interface en langue naturelle et non dialogue
 - Abandon rapide (3 reformulations) en l'absence de réponse [Hasson, 2007]
 - Interaction limitée jugée satisfaisante par les utilisateurs [Falzon et al., 1986]

Corpus Daft : constitution (2)

2 méthodes employées :

- **Recueil** : interaction avec le système DAFT 1.0
 - En environnement contrôlé :
 - Compteur
 - Jeu de tours de Hanoï
 - Site web du groupe AMI
 - En environnement semi-ouvert :
 - Site web du GT ACA
- **Enrichissement** :
 - 2 thésaurus :
 - [Molinsky & Bliss, 1994]
 - [Atkins & Lewis, 1996]
 - FAQ applications complexes (Word et LaTeX)



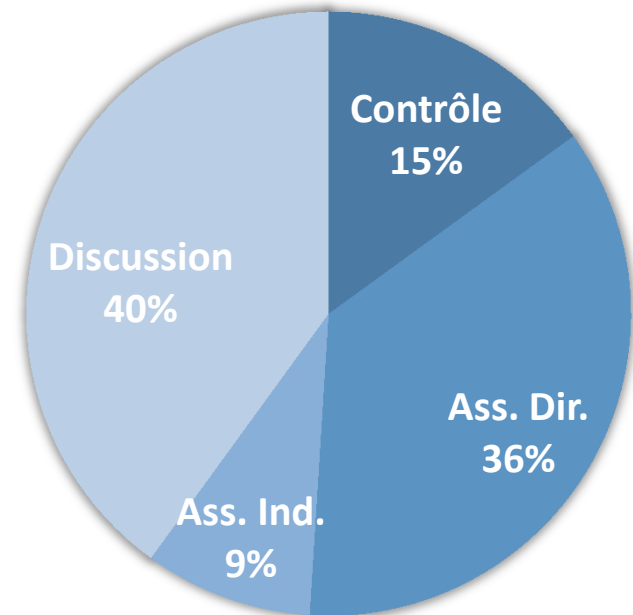
Constitution d'un corpus de 11626 requêtes

Activités conversationnelles du corpus Daft (1)

- **Contexte** : les utilisateurs devaient communiquer avec l'agent en cas de difficulté pour accomplir une tâche
- **Problème** : dans le corpus recueilli, toutes les requêtes ne semblent pas relever de l'assistance
- **Activité conversationnelle** = 1^{er} degré de l'intentionnalité communicative de l'utilisateur
- **Méthodologie** : annotation manuelle du corpus en deux temps :
 1. Sous-ensemble d'un dixième du corpus recueilli :
 - identification empirique des activités
 - définition de questions de manière systématique
 2. 2^e sous-ensemble d'un dixième du corpus recueilli :
 - application du protocole défini en 1.
 - estimation des proportions de chaque activité conversationnelle

Activités conversationnelles du corpus Daft (2)

- Orientées vers l'accomplissement de la tâche :
 - **Contrôle** : agent médiateur pour l'accomplissement d'actions
 - **Assistance** :
 - **Directe** : question explicite
 - **Indirecte** : expression implicite d'un besoin
- Orientées vers la discussion avec l'agent assistant :
 - **Réaction à une réponse** : accord/désaccord, incrédulité, incompréhension, insistance
 - **Fonctions communicatives** : début/fin interaction, phatiques
 - **Dialogue avec l'agent** : ordre à l'agent, question sur l'agent, avis sur l'agent, menace...
 - **Commentaires sur l'application** : sans valeur d'assistance
 - **Autres**



On se concentrera uniquement sur les requêtes orientées tâches

Comparaison à un corpus généraliste

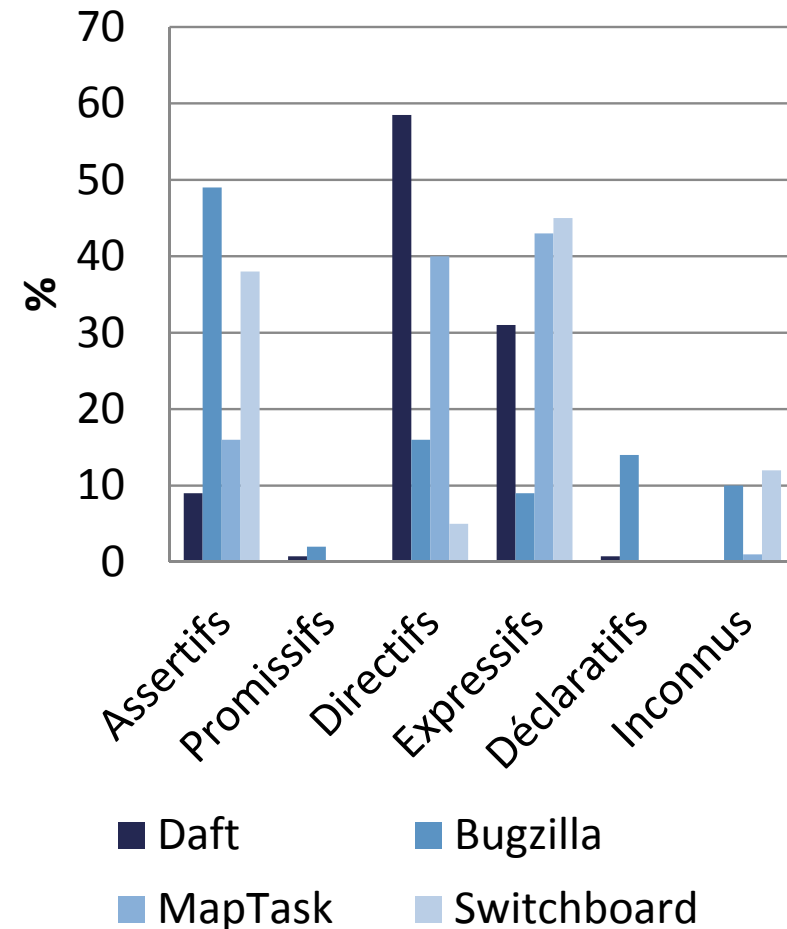
- Corpus généraliste MULTITAG [Paroubek, 2000] :
 - extraits de « Le Monde »
 - textes littéraires
- Méthodologie 1 : ramener MULTITAG à une taille comparable
 1. Sous-ensemble de MULTITAG ayant le même nombre de phrases
 2. Sous-ensemble de MULTITAG ayant le même nombre de flexions
- Résultats 1 : les requêtes de Daft ont :
 1. Des **phrases plus courtes** : 8,3 mots/phrases (vs 31 dans MULTITAG)
 2. Des **phrases moins riches** : 2 fois plus de lemmes dans MULTITAG, à nombre de flexions égales
- Méthodologie 2 : étudier le nombre de lemmes nouveaux lors d'une augmentation de la taille des deux sous-corpus considérés
- Résultats 2 : 3 fois plus de mots nouveaux pour MULTITAG
→ **le corpus Daft sature mieux son domaine de langue**

Comparaison à d'autres corpus orientés tâches (1)

- Corpus choisis :
 - **Switchboard** [Godfrey & Holliman, 1997] : 1155 dialogues téléphoniques (200 000 phrases)
 - **MapTask** [Anderson et al., 1991] : 128 dialogues
 - **Bugzilla** [Ripoche, 2006] : 1 200 000 commentaires issus de 128 000 rapports de bugs
- Méthodologie : comparaison de **profils interactionnels**
 - « Distribution des actes de dialogue dans une unité d'interaction donnée » [Ripoche, 2006]
 - Nécessité d'une **taxonomie d'actes de dialogue commune**
Taxonomie choisie : [Searle, 1969]
 - **Assertifs** : engagement du locuteur sur la vérité de ce qu'il énonce
 - **Promissifs** : engagement du locuteur à faire ce qu'il énonce
 - **Directifs** : demande d'engagement de l'allocutaire à faire ce qui est énoncé
 - **Expressifs** : expression de l'état d'esprit du locuteur par rapport à ce qu'il énonce
 - **Déclaratifs** : modification de l'état du monde par la simple énonciation

Comparaison à d'autres corpus orientés tâches (2)

- **Majorité de directifs (58%)** : mode d'expression plus direct qu'avec des humains :
 - l'agent est au service de l'utilisateur
 - doutes quant à ses capacités de compréhension
- **Expressifs (31%) préférés aux assertifs (9%)** : en détresse cognitive, **expression du ressenti** plutôt qu'un exposé objectif de la situation (comme dans Bugzilla)
- **Très peu de promissifs (< 1%)** : **agent perçu comme subordonné**, utilisateur non contraint à obéir
- **Quasi-absence de déclaratifs (< 1%)** : les actions sont effectuées directement, sans annonce particulière



Plan

- **Partie I : État de l'art**

- **Partie II : Chaîne de traitement**
 - Principes, version antérieure et besoins
 - Constitution et étude du corpus Daft
 - **L'analyseur syntaxico-sémantique : GRASP**
 - Le langage formel de requêtes : DAFT
 - Le générateur automatique de requêtes formelles : DIG

- **Partie III : Validations et applications**

GRASP : analyse syntaxico-sémantique

■ Motivation :

- Précision cruciale pour éviter le « Clippy Effect »
- Identification des verbes essentielle pour la détermination ultérieure des actes de langage
- L'assistance est assimilable à un sous-langage [Kittredge, 2003]
→ **possibilité d'un analyseur spécifique plus performant**

■ Méthodologie :

Requêtes bruitées → **approche bottom-up** :

- **Lemmatisation** : mots → lemmes/locutions
- **Étiquetage sémantique** : lemmes → lemmes + clés sémantiques
- **Analyse constituants** : association lemmes selon critères grammaticaux

GRASP : lemmatisation

- **Lexique** : 105 396 mots → 6325 flexions uniques → 3736 lemmes
+ lemmes et locutions proches = **4266 entrées**

```
{ LEM["initialiser"],  
  POS[VV],  
  FLEX[CONJ["initialiser", 7]],  
  KEYS[{"TORESET"}]}
```

```
{ LEM["inquiet"],  
  FLEX[{MS["inquiet"], MP["inquiets"],  
        FS["inquiète"], FP["inquiètes"]}],  
  POS[JJ],  
  KEYS[{"ISWORRIED"}]}
```

```
{ LEM[".a.cause.de"],  
  LOC[{"a", "cause", "de"}],  
  POS[GG],  
  FLEX["-"],  
  DICO[LOCUTION],  
  KEYS[{"BECAUSE"}]}
```

- **Lemmatiseur** : mots/locutions → nœuds
 - **Prétraitement** entités nommées et idiosyncrasies
 - **Correction orthographique** (1 mot ↔ 0 lemme – 8,4% des cas) :
 - Distance de Damerau-Levenshtein : 88,5% d'efficacité
 - **Étiquetage morphosyntaxique** (lemme → nature)
 - **Désambiguïsation** (1 mot ↔ + de 1 lemme – 7,4% des cas) :
 - Fréquence de N-grammes : 97,2% d'efficacité
 - STOPLIST pour cas restants

GRASP : étiquetage sémantique

- **1294 clés sémantiques** (\approx synsets WordNet [Fellbaum, 1998]) :
 - Annotation manuelle à partir du lexique, par regroupement
 - Organisées selon une ontologie à 2 niveaux : 17 classes, 128 sous-classes

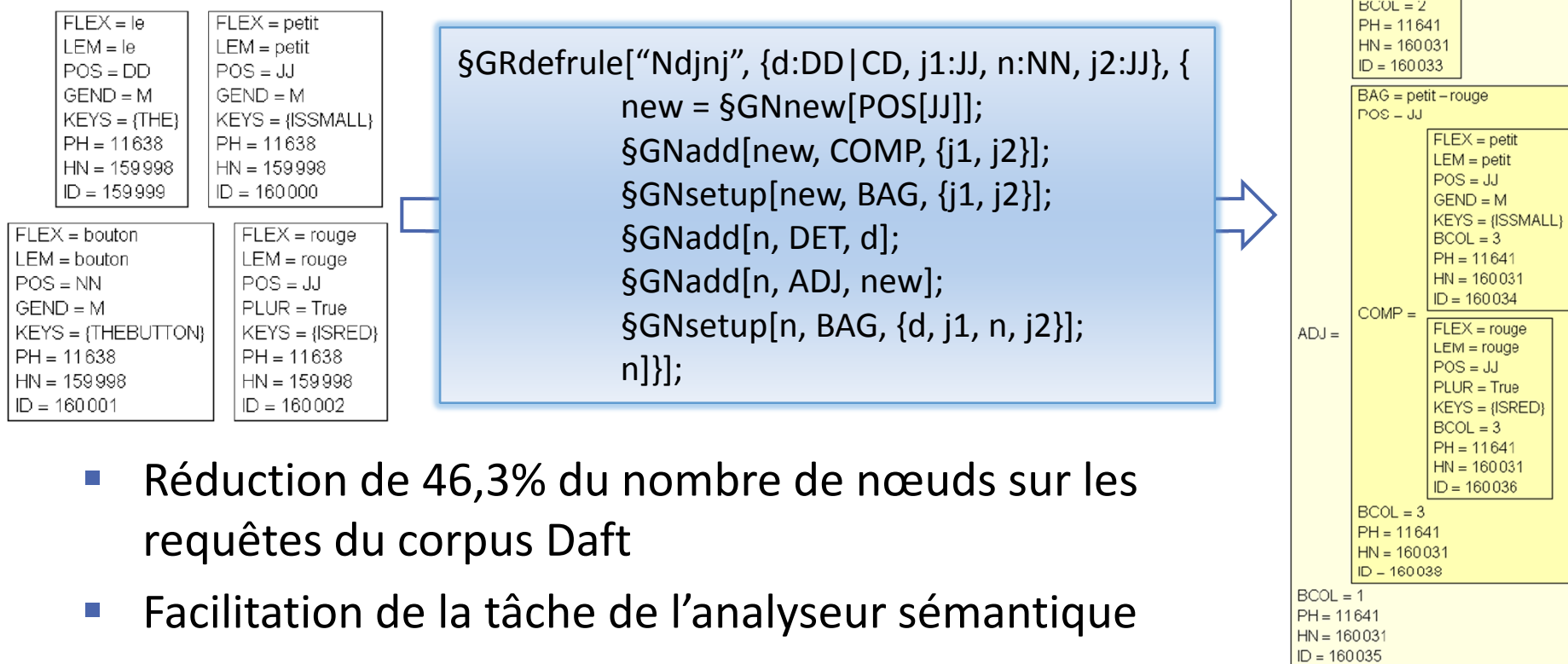
```
{ ID[TOCLICK],  
  ONTO[‘⌘INTERFACE’, ‘*BUTTON’],  
  TYPE[VERB],  
  COM[‘Denotes the action of clicking  
        with the mouse’]]  
  LEM[‘appuyer’, ‘clicker’, ‘enfonce’] }
```

```
{ ID[THEBUTTON],  
  ONTO[‘⌘INTERFACE’, ‘*BUTTON’],  
  TYPE[NAME],  
  COM[‘Denotes a button in the  
        application window’]]  
  LEM[‘bouton’] }
```

- **Désambiguïsation sémantique** : 1 lemme \leftrightarrow + de 1 clé (6,7% des cas)
 - **Approche statistique supervisée** : 97,5% d’efficacité
 - Annotation préalable des lemmes ambigus dans les requêtes du corpus
 - Utilisation de la fréquence des lemmes ou clés environnantes
 - Gestion des erreurs restantes par **règles ad hoc**

GRASP : analyse grammaticale

- **240 règles de réécriture :**
 - Observation de schémas dans le corpus
 - Définition des opérations de regroupement appropriées



- Réduction de 46,3% du nombre de nœuds sur les requêtes du corpus Daft
- Facilitation de la tâche de l'analyseur sémantique

Plan

- **Partie I : État de l'art**
- **Partie II : Chaîne de traitement**
 - Principes, version antérieure et besoins
 - Constitution et étude du corpus Daft
 - L'analyseur syntaxico-sémantique : GRASP
 - **Le langage formel de requêtes : DAFT**
 - Le générateur automatique de requêtes formelles : DIG
- **Partie III : Validations et applications**

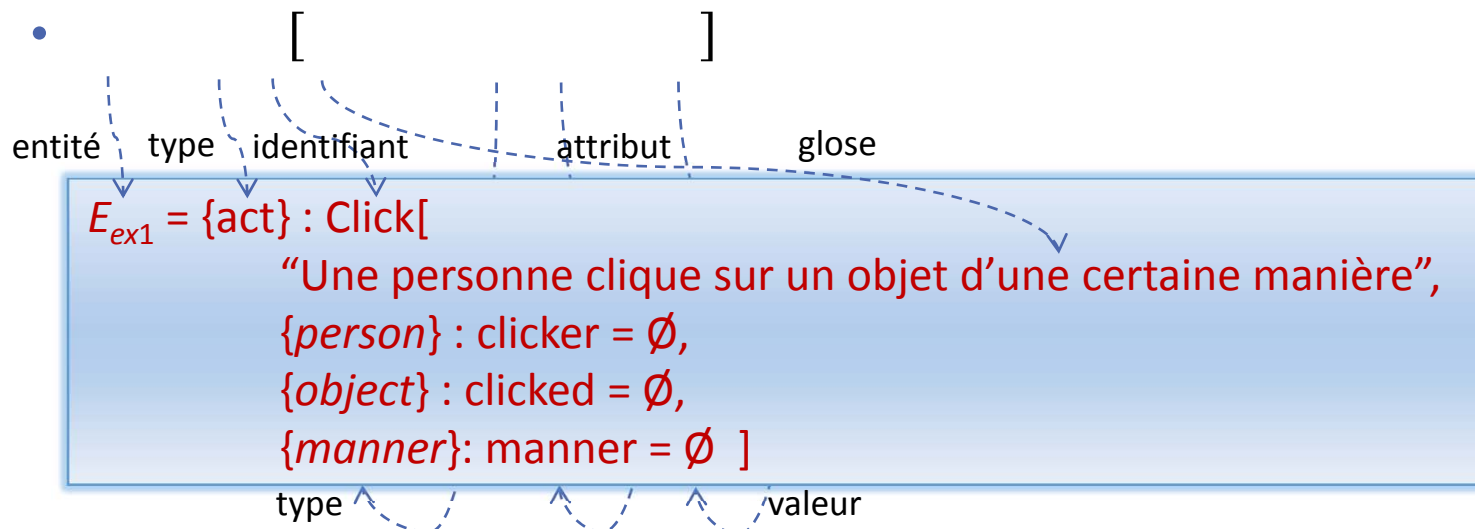
DAFT : langage formel de représentation des requêtes

■ Méthodologie de définition :

- Annotation du corpus
- Extraction des relations sémantiques, en particulier par rapport aux verbes

■ Structure des éléments du langage :

- Représentation de type « **frames** », héritée de DAFT 1.0, à la manière de TRAINS et TRIPS [Allen et al., 1995 ; 2000]

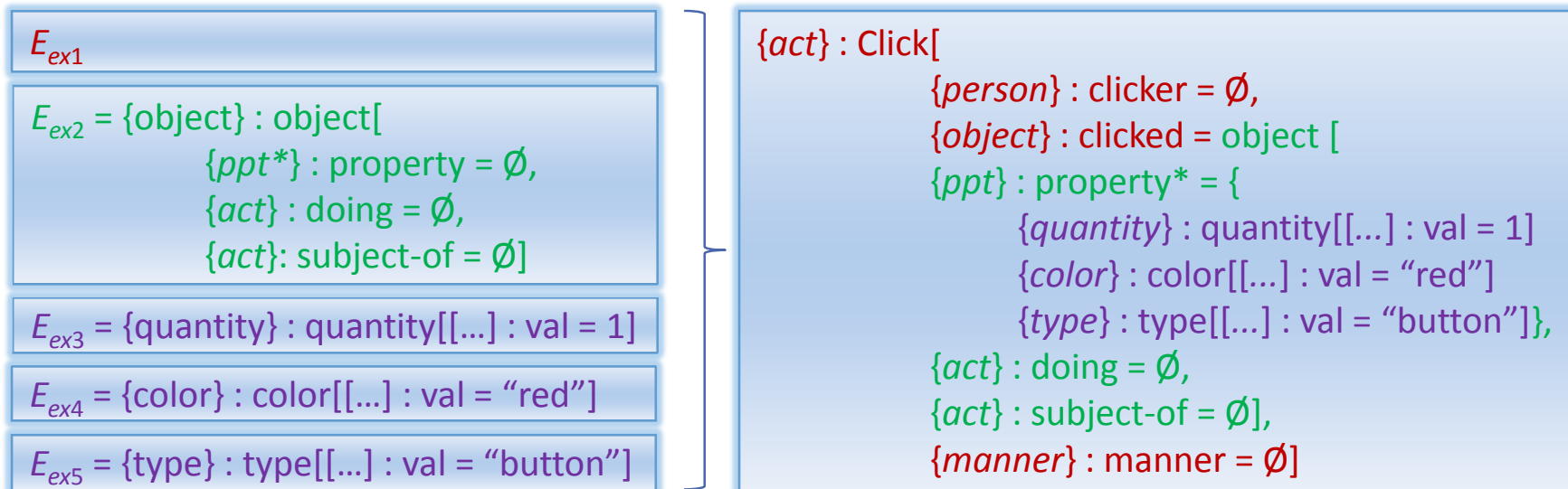


DAFT : entités du langage

- **237 entités** définies, selon 4 classes :
 - **44 Modalités** : actes de dialogue au sens large
POSSIBILITY, WAY, EXISTENCE, ASK, NEG, REGRET, REASON...
 - **66 Actions** : verbes d'opération et de prédication
Click, Modify, Show, Recommend, Start, Add, Move, Win...
 - **3 Références** : correspondant à une instance d'un élément concret ou abstrait du monde : *object*, *concept* et *person*
 - **124 Propriétés** : qualificatifs définissant les références
speed, *manner*, *time-duration*, *upset*, *text-bold*...

DAFT : format des requêtes

- **Requêtes formelles** = association d'entités
 - Restriction des combinaisons possibles grâce aux types des attributs et des entités
 - Ontologie de types
 - Structure globale généralement de la forme :
 $M_1[\dots M_i[A_1[R_1[P_1[\dots], \dots P_j[\dots]], R_k[\dots]], \dots A_l[\dots]], \dots]$
- Exemple : « Clique le bouton rouge »



DAFT : évaluation de la robustesse de la structure du langage

■ Méthodologie :

- Annotation manuelle d'un dixième des phrases recueillies :
 1. Définition des entités nécessaires au fur et à mesure
 2. Unification des entités utilisées
ex : `Hide[] = Modify[visibility[val=false]]`
 3. Unification des attributs des entités (noms, rôles, types associés)

■ Résultats :

- Plus de 90% des requêtes ont pu être correctement modélisées dans le formalisme proposé. Cas restants :
 - Absence de verbe : « carte du site », « membres »
→ besoin d'un verbe par défaut
 - Absence de référence : « go », « fais-le », « montre »
→ besoin d'une référence par défaut (dépendante du contexte)
 - Références complexes : « parle même si tu n'as rien à dire », « exécute la même chose qu'avant »

La structure du langage permet la représentation des requêtes recueillies

DAFT : évaluation de la couverture

■ **Méthodologie :**

- Annotation manuelle d'un second sous-ensemble d'un dixième des phrases recueillies :
 1. Sans autoriser la définition de nouvelles entités
 2. En définissant de nouvelles entités pour les phrases non annotées en 1.

■ **Résultats :**

- Ajout nécessaire de 4 actions et 2 modalités
- Après ajouts : couverture comparable à celle obtenue précédemment
- Un doublement de la taille du corpus n'entraîne qu'une augmentation de 5% du nombre d'entités nécessaires (conforme à la loi de Zipf)

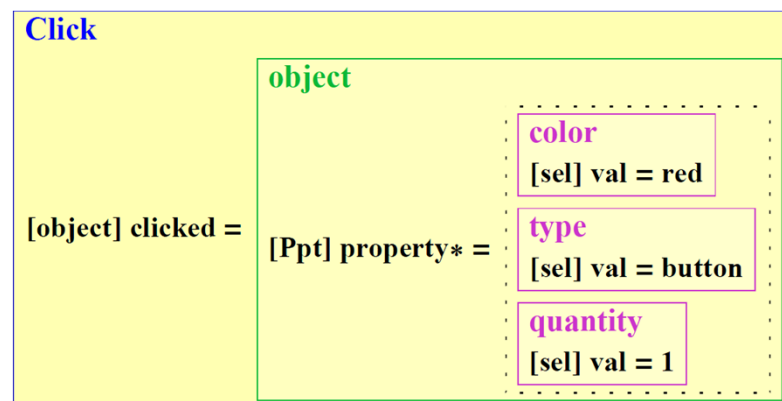
**Le corpus est de taille suffisante pour espérer couvrir
le domaine de l'assistance**

Plan

- **Partie I : État de l'art**
- **Partie II : Chaîne de traitement**
 - Principes, version antérieure et besoins
 - Constitution et étude du corpus Daft
 - L'analyseur syntaxico-sémantique : GRASP
 - Le langage formel de requêtes : DAFT
 - **Le générateur automatique de requêtes formelles : DIG**
- **Partie III : Validations et applications**

DIG : génération des requêtes formelles

- **Objectif : automatiser** de la production de requêtes en DAFT
- **Ressources en entrée :**
 - Requête structurée grammaticalement par GRASP
 - Ensemble de 237 entités différentes définies en DAFT
 - Ensemble de 1294 clés sémantiques
 - Ensemble de règles de combinaisons sémantiques
- **Sortie :** une requête structurée sémantiquement, représentée sous forme de boîtes imbriquées



DIG : lien clés sémantiques ↔ entités DAFT

- Ajout d'un champ DAFTINFO aux clés :

```
{ ID[TOCLICK],  
...  
DAFTINFO[{{_, Click[]}}]  
LEM['appuyer', 'clicker', 'enfonce'] }
```

1 clé = 1 entité

```
{ ID[THEBUTTON],  
...  
DAFTINFO[{{_, object[],  
          type[val="button"]}}]  
LEM['bouton'] }
```

1 clé = N entités

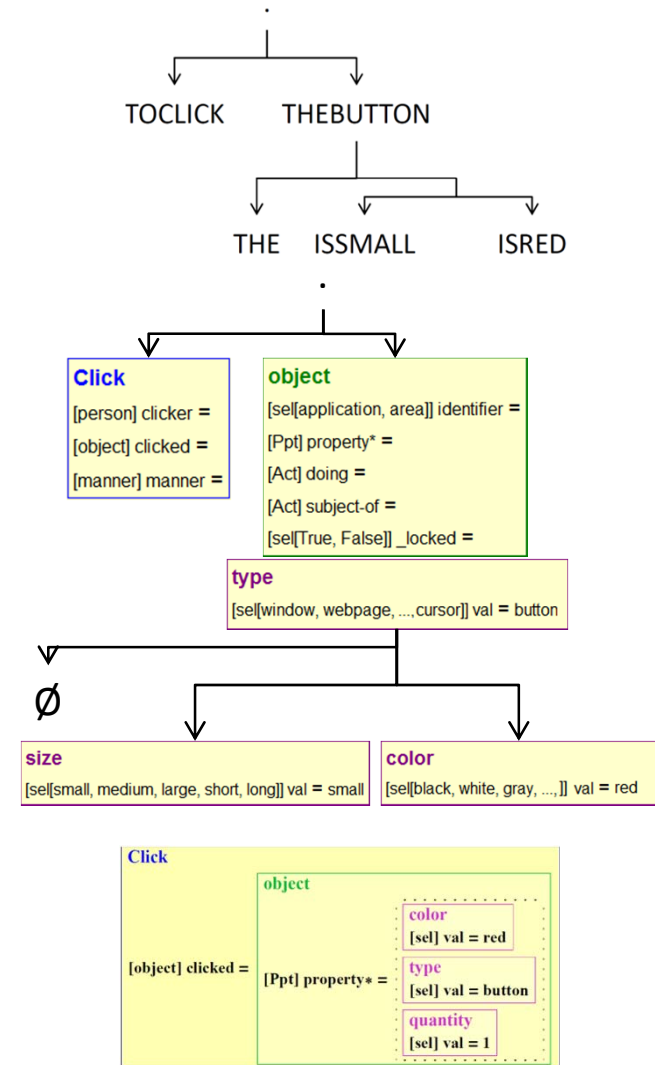
```
{ ID[ISRED],  
...  
DAFTINFO[  
          {NN, object[], color[val="red"]}  
          {JJ, color[val=red]}  
          }  
LEM['rouge'] }
```

1 clé = 1 entité sous condition

- 697 clés associées à 1 ou plusieurs entités DAFT, mais :
 - Des clés n'ont pas de valeur sémantique intrinsèque (ex : « que »)
 - 1/3 des requêtes ont toutes leurs clés ayant un champ DAFTINFO

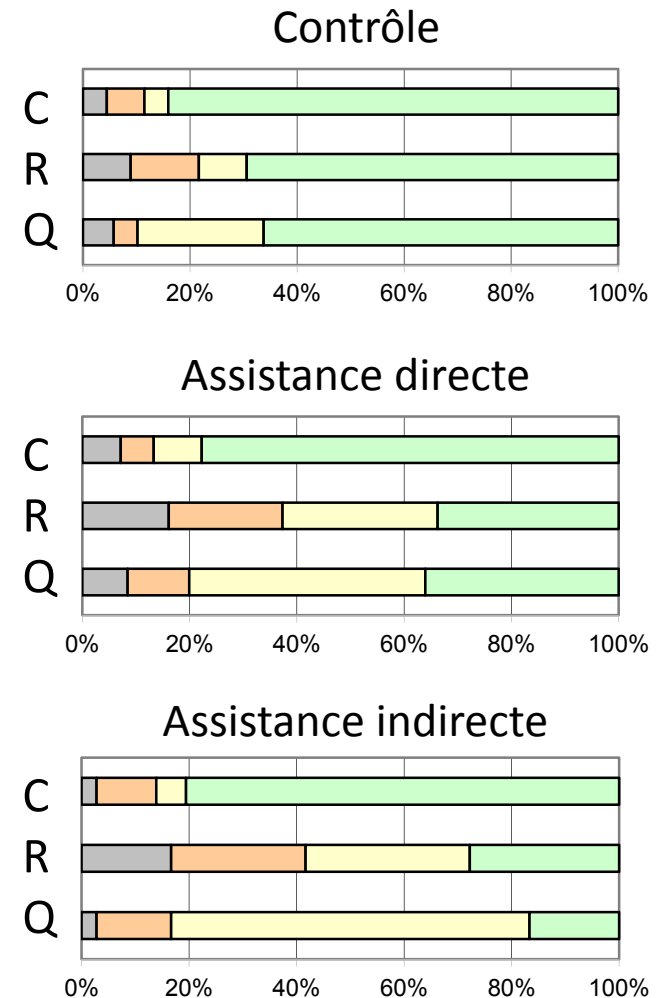
DIG : principe de fonctionnement

1. Création de l'**arbre des clés sémantiques** à partir de la requête fournie par GRASP
2. **Instanciation** d'entités : remplacement des clés sémantiques par les entités DAFT liées
3. **Combinaison** en partant des feuilles les plus profondes de l'arbre en appliquant :
 1. Les 24 règles de combinaisons spécifiques, si elles peuvent s'appliquer
 2. La règle de combinaison « par défaut » : « R_2 peut être fusionné dans R_1 si et seulement si il existe un attribut a de R_1 sans valeur associée et tel que R_2 a un type compatible avec celui de a . »



DIG : évaluation qualitative

- **Méthodologie** : 3 critères d'annotation, évalués sur une échelle de 0 à 3 :
 - **Qualité de la représentation** : la requête produite par DIG a-t-elle un sens ?
 - **Représentativité de la requête** : le sens est-il le même que la requête en langue naturelle ?
 - **Complétude de la requête** : retrouve-t-on toutes les informations présentes dans la requête en langue naturelle ?
- **Résultats** :
 - Complétude : 90% de requêtes parfaites, indépendamment de l'activité conversationnelle
 - Qualité : meilleure pour les requêtes de contrôle (90% de score ≥ 2) que pour l'assistance dans son ensemble (80%)
 - Représentativité : 80% pour le contrôle, 65% pour l'assistance directe, 60% pour l'assistance indirecte

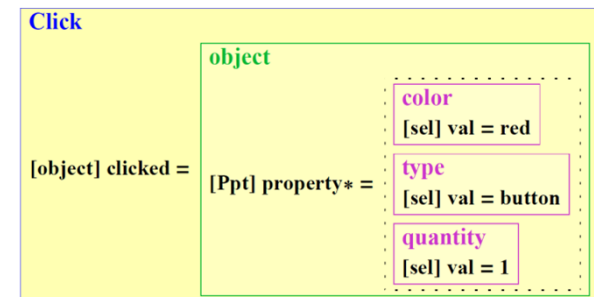


Plan

- **Partie I : État de l'art**
- **Partie II : Chaîne de traitement**
- **Partie III : Validations et applications**
 - **Identification automatique d'activités conversationnelles**
 - **Agents Web DIVA**
 - **Vers un agent rationnel et comportemental**

Classification automatique des requêtes selon l'activité conversationnelle (1)

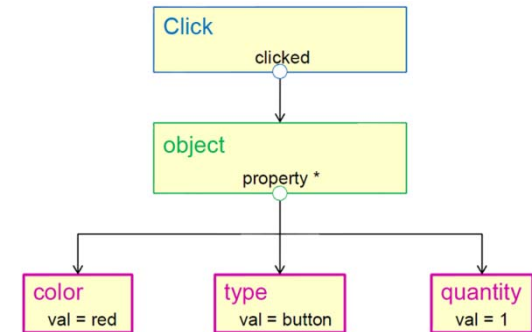
- **Motivation** : identification préalable de l'activité conversationnelle pour faciliter la tâche de l'analyste et/ou de l'agent rationnel
- **Méthode 1** : requête comme un tout
 1. **Longueur des requêtes en langue naturelle** :
Contrôle < Assistance
 2. **Longueur des requêtes formelles** :
requêtes d'assistance plus complexes que celles de contrôle, donc plus longues
 3. **Profondeur des requêtes formelles** :
Contrôle < Assistance, mais grandes variations
 4. **Nombre d'entités** :
fortement corrélé à longueur de la phrase en langue naturelle, donc Contrôle < Assistance
 5. **Combinaison de classifieurs 1—4** :
Meilleure performance avec KPPV : **51,9%**



Longueur : 1
Profondeur : 3
Nombre d'entités : 5

Classification automatique des requêtes selon l'activité conversationnelle (2)

- **Principe** : mesurer la distance par rapport aux requêtes dont on connaît l'activité conversationnelles
- **Méthode 2** : requête comme un arbre
 - Utilisation d'une fonction noyau basée sur [Collins & Duffy, 2002], adaptée aux spécificités des requêtes DAFT
 - Meilleure performance avec PPV : **65,4%**
- **Méthode 3** : requête comme un vecteur
 - Utilisation de KPPV avec différentes distances (L^1 , L^2 , L^∞ , Bray-Curtis...) et différents classifieurs (issus de Weka)
 - Meilleure performance avec bayésien naïf : **74,1%**
- **Combinaison de méthodes 1, 2 et 3** :
 - Meilleure performance en combinant différents classifieurs de 3 avec table de décision : **76,1%**



ASK	0
NEG	0
...	...
Modify	0
Click	1
Show	0
...	...
object	1
concept	0
person	0
...	...
color	1
type	1
name	0
quantity	1

Évaluations d'usage : agents sur le Web

■ Besoin :

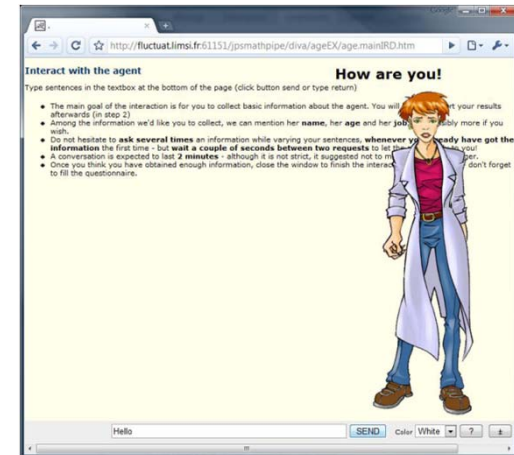
- Une plate-forme d'expériences sur le Web pour l'étude de phénomènes particuliers
- Un système léger facilement déployable

■ Solution : amélioration d'une architecture de type chatbot, par la réutilisation :

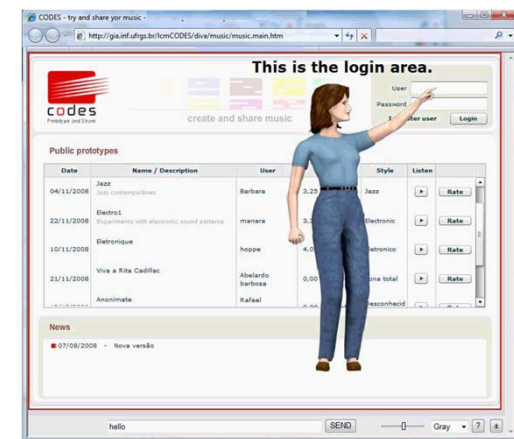
- de la **méthodologie** fondée sur une étude de corpus
- d'une partie des **ressources** de GRASP et de DIG

■ Validation :

- Réalisation d'une expérience basée sur des variations du comportement conversationnel d'un agent [Xuetao et al., 2009]
- Possibilité d'intégration au sein d'applications déjà existantes : exemple avec l'environnement Web collaboratif de création musicale CODES [Miletto et al., 2009]



[Xuetao et al., 2009]



[Miletto et al., 2009]

Vers des agents rationnels et comportementaux : besoins

- « Comment faire pour quitter ? »
 - Une réponse rationnelle et efficace n'est pas neutre
 - Tendance naturelle à l'anthropomorphisme des systèmes informatiques [Reeves & Nass, 1996], renforcée par la présence d'un agent animé
- **Nécessité d'un modèle psychologique :**
 - **Statique** : pour avoir une constance dans les réactions
 - **Dynamique** : pour s'adapter aux conséquences des interactions humain-agent
- L'ajout de règles pour gérer la psychologie peut poser problème :
 - Possibilité d'exhiber les règles appliquées (et donc de les ignorer)
 - Possibilité de primauté d'une règle rationnelle sur une émotion forte
 - Les règles pouvant aller à l'encontre de l'efficacité (pour améliorer le réalisme) risquent d'être éliminées par le self-monitoring [Sloman, 2000]
- **Nécessité de contraintes cognitives :**
 - **Non altérables** par des règles rationnelles
 - **Non détectables** par l'agent rationnel

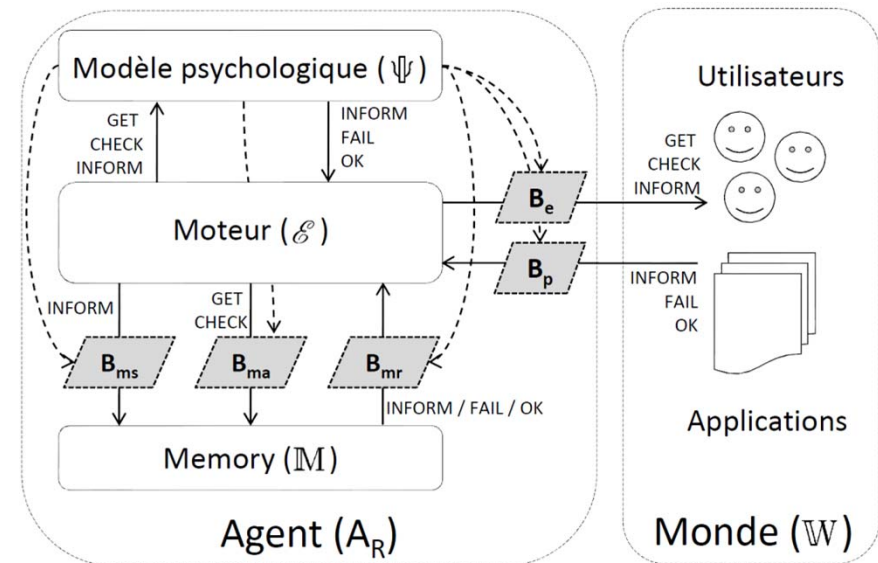
Vers des agents rationnels et comportementaux : architecture (1)

- Un agent $\langle \quad \rangle$ interagit avec le monde
- **Moteur** : en charge de l'application des heuristiques rationnelles de l'agent
- **Mémoire** : procédurale, sémantique, épistémique
- **Modèle psychologique** 4 types d'états mentaux :
 - **Statiques de l'agent** : modèle OCEAN [Goldberg, 1981]
 - **Dynamiques de l'agent** :
 - Physiologiques : énergie physique, bonheur physiologique
 - Epistémiques : confiance en soi, satisfaction intellectuelle
 - **Statiques du couple agent-utilisateur** : Autorité et Familiarité
 - **Dynamiques du couple agent-utilisateur** : Dominance, Affection, Confiance

Vers des agents rationnels et comportementaux : architecture (2)

- Les **biais cognitifs** sont des règles qui s'appliquent :
 - Hors du moteur rationnel
 - Sur les messages transitant entre les différents constituants internes de l'agent et vers le monde
 - Uniquement en fonction de l'état mental actuel

- On distingue :
 - Les **biais expressifs** B_e
ex : « Morosité » = ajout d'expressions de mauvaise humeur
 - Les **biais perceptifs** B_p
ex : « Victimisation » = perception de critiques non existantes
 - Les **biais de recherche en mémoire** B_{mr}
ex : « Manque de confiance en soi » = besoin de vérifier dans le monde en plus
 - Les **biais de lecture en mémoire** B_{ma}
ex : « Mauvaise foi » = simulation d'un « acte manqué »
 - Les **biais d'écriture en mémoire** B_{ms}
ex : « Amnésique » = ne stocke pas tout ce qui lui arrive en mémoire épisodique



Conclusion

- Il existe un besoin d'assistance des usagers novices, pouvant être étudié par l'analyse des requêtes en langue naturelle de ceux-ci
- Une chaîne de traitement de la langue naturelle a été réalisée pour permettre cette étude :
 - Fondée sur un **corpus**
 - **Couvre** le domaine de langue fermé de l'assistance
 - Permet l'extraction d'un **langage de représentation** des concepts-clés
- **Perspectives :**
 - Vers une prise en compte de la **pragmatique de l'assistance**
 - Utilisation de DAFT comme support de l'étude d'agents dotés d'une **architecture cognitive**