Generic and Subjective Assisting Conversational Agents

François Bouchet
(bouchet@limsi.fr)

Université Paris-Sud XI – LIMSI-CNRS

Thesis supervised by
Jean-Paul Sansonnet (jps@limsi.fr)

UFRGS seminar
21/11/2008

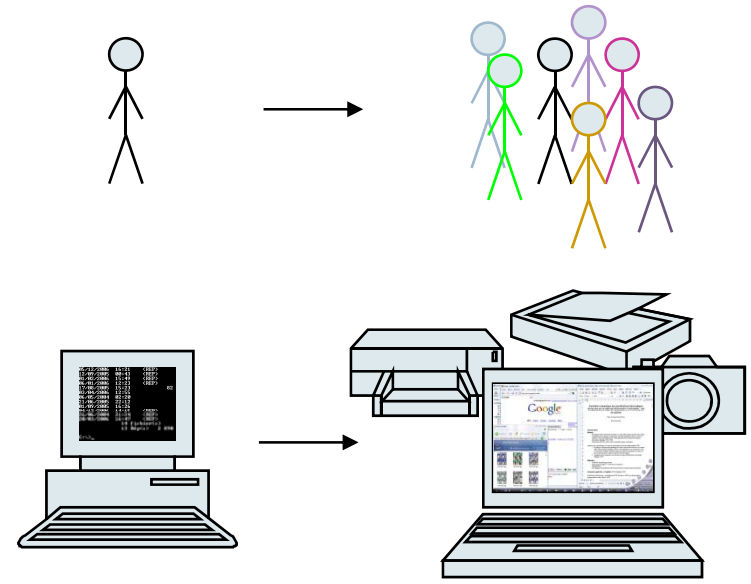
Outline

- The problematics of assistance
  - The increasing need for assistance
  - Help systems evolution
  - Assisting Conversational Agent definition
  - ACAs as the ideal assistance system

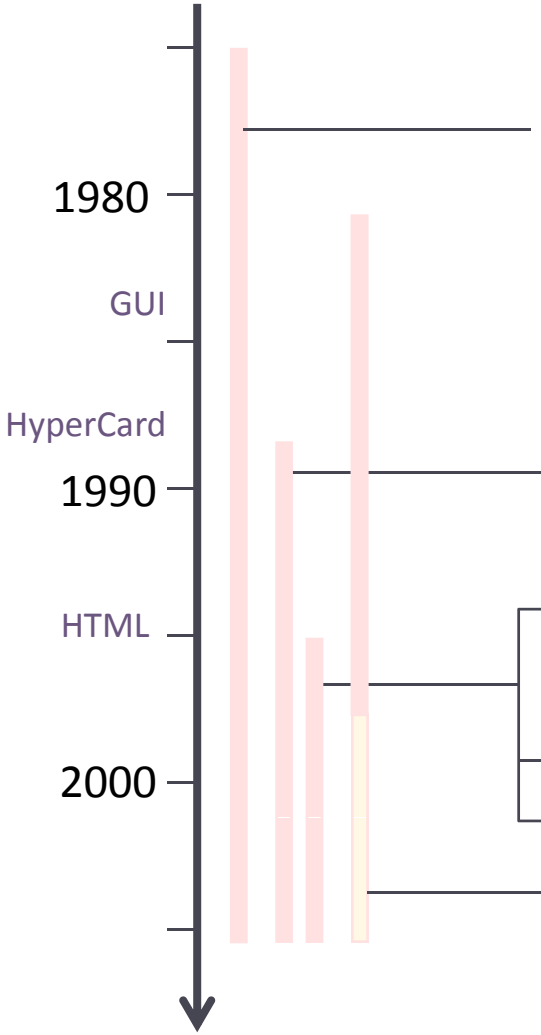
- Developing an ACA: the DAFT project
  - Genericity
    - Why? Very high development cost
    - How? Corpus of requests (grounding in reality)
    - How? Requests classes from a formal request representation
  - Rationality
    - Heuristic example
    - Beyond rationality: the need for personality

The increasing need for assistance

- Users evolution:
  - In number:
    600 millions (2002),
    1 billion (2008),
    2 billions (2015 – projection)
  - In variety:
    from computer scientists to everyone
- Hardware evolution (Moore's law):
  - Application fields: word processor, multimedia, video games, news…
  - Interaction fields: printer, scanner, PDA, camera, ambient electronic devices…
- Software evolution:
  - More numerous: dozens of word processors or image editors.
  - More complex: in 6 classical public applications
    150 « basic » actions (in menus);
    60 dialogue boxes ;
    80 tools (through icons).
    (Beaudoin-Lafon, 97)



Help systems evolution



External Help

Didactic help: tutorials more related to the domain than to the software itself

User manuals: instructions for use to learn how to handle the software

Static online help

- Help files
  - Didactic help: tutorials
  - Electronical user manuals: instructions for use
- Databases: keyword-based associated research
- Hypertext documentation: non-linear document exploration (browsing)

Dynamic online help

- Contextual Help Systems: model of the user + model of his current task(s)
- Multimodality in output: deictic + text/speech
- Natural language requests (text or speech)
- Task oriented dialogue: rational conversational agents

Assisting Conversational Agent: Definition

- **Assisting**: *« An Assisting Agent is a software tool with the capacity to resolve help requests, issuing from novice users, about the static structure and the dynamic functioning of software components or services »* (Maes, 94)

- **Conversational**: interaction in unconstrained natural language (NL)
  Why?
  Frustrated (novice) users spontaneously express use NL
  (⇔« thinking aloud effect » (Ummelen & Neutelings, 00))

- **Embodied**: given a graphical more or less realistic appearance
  Why?
  Increased agreeability and believability – « Persona Effect » (Lester, 97)

Help systems comparison

Help system	Reactivity	Vocabulary	Task-oriented	Dynamic	Personalized	Proactive
Paper documentation	-	-	-	-	-	-
Electronic documentation	+	-	-	-	-	-
FAQ, How-to, Tutorial	+	=	+	-	-	-
Contextual Help Systems	+	=	=	+	-	-
Assisting Conversational Agent	+	+	+	+	+	=

Reactivity: how fast is it for the user to open the help system when it needs it?

Vocabulary: are there strong constraints or limitations on the words the user has to know to be able to efficiently use the help system? (ex: specific keywords or grammar constructions for NL)

Task-oriented: does the help system explain procedures and not only define concepts?

Dynamic: does the help system change according to the application state?

Personalized: does the help system change according to the user?

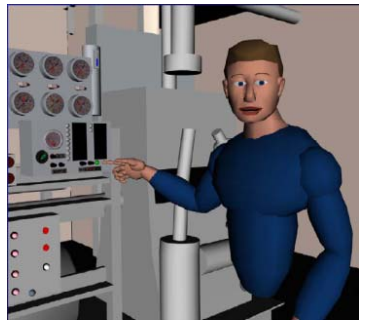
Proactive: does the help system appear only when asked for or can it anticipate the user needs (without being intrusive)?

Conclusion**:** Assisting conversational agents *potentially* seem to be the most efficient way to help novice users.

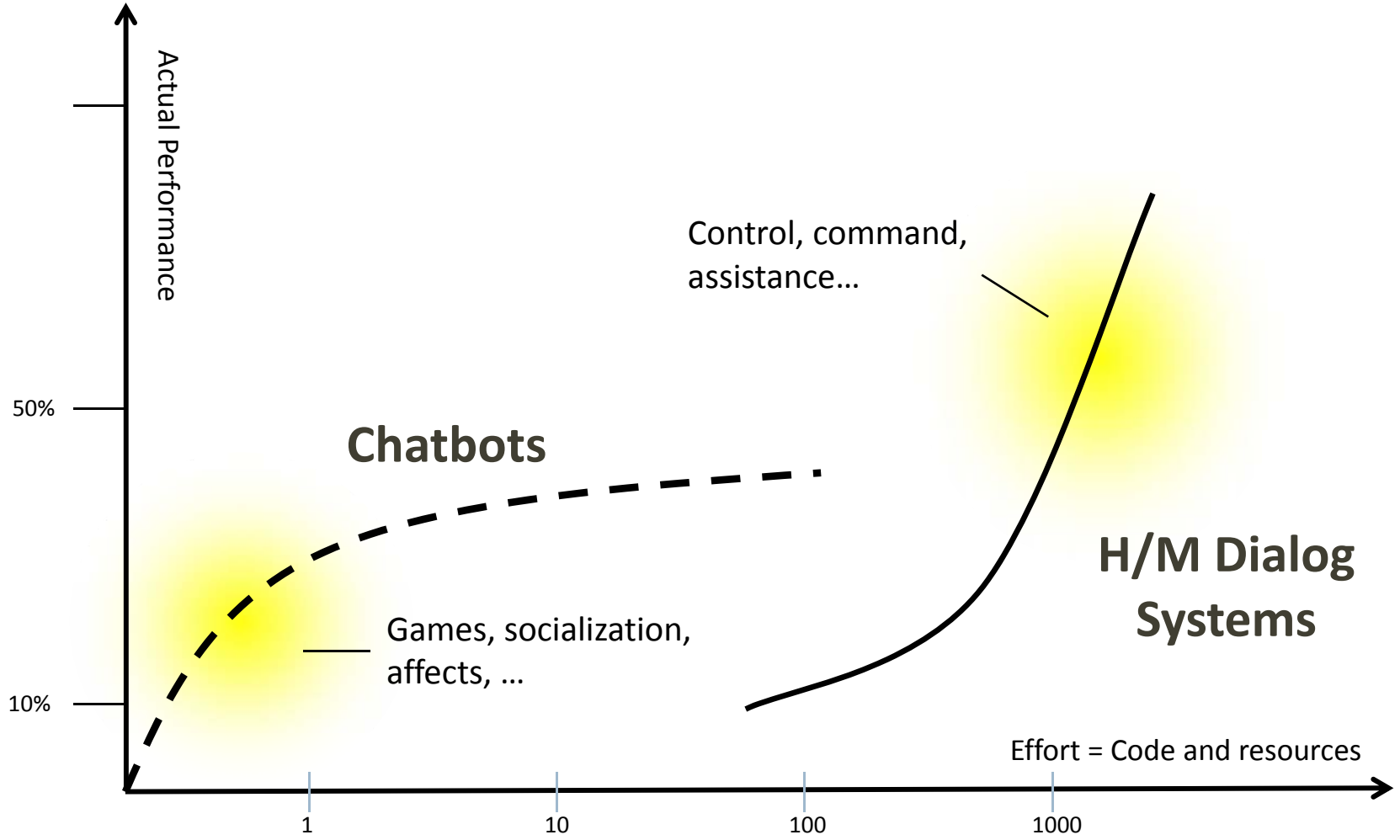
ECA's efficiency is crucial

- The « Clippie effect » (Randall & Pedersen, 98)
  - Implicit higher level of expectation
  - Failures feel more jarring
  - Perception shift is fast:
    friendly companion → useless gadget

- Some early successes though:
  - (Rickel & Johnson, 99)
    STEVE for procedural training
  - (Cassell et al., 99)
    REA a real estate simulation agent



ACA's development cost is high



In some cases: application development cost < ACA development cost !

Conclusion: we should find a way to reduce the development cost → DAFT project

Outline reminder

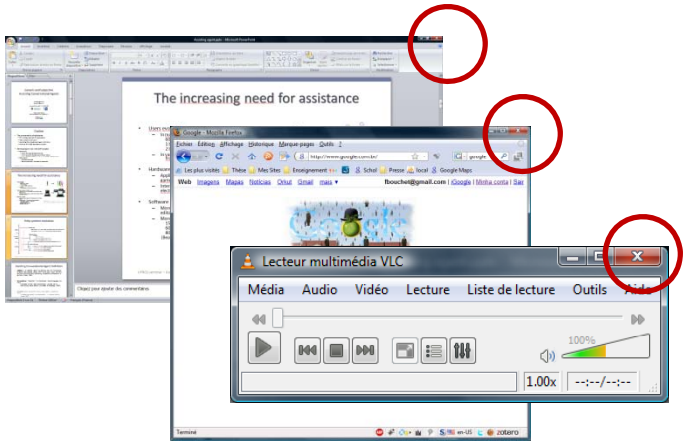
- The problematics of assistance
  - The increasing need for assistance
  - Help systems evolution
  - Assisting Conversational Agent definition
  - ACAs as the ideal assistance system

- Developing an ACA: the DAFT project
  - Genericity
    - Why? Very high development cost
    - How? Corpus of requests (grounding in reality)
    - How? Requests classes from a formal request representation
  - Rationality
    - Heuristic example
    - Beyond rationality: the need for personality

Genericity in applications and formulations

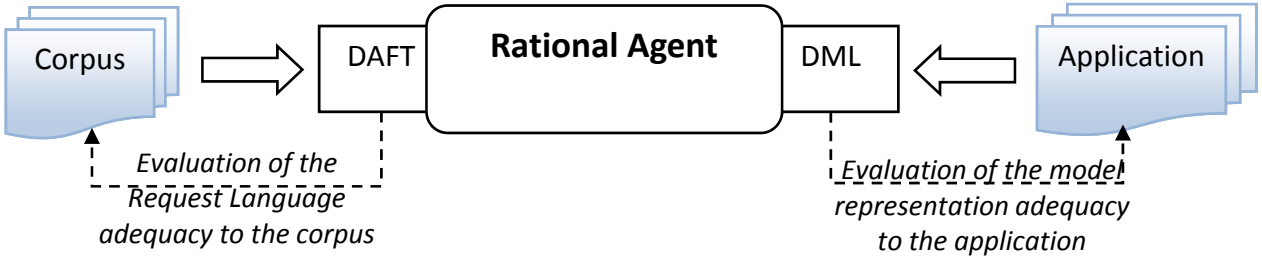
- Several applications, a similar action
  - « I want to quit ! »

  Need for a formal application model (Leray & Sansonnet, 06)



- Several formulations, a similar intention
  - What should I do to quit?
  - QUIT!!!
  - Would you be kind enough to tell me the way to quit this application?
  - …

  Need for a formal semantic-based request representation

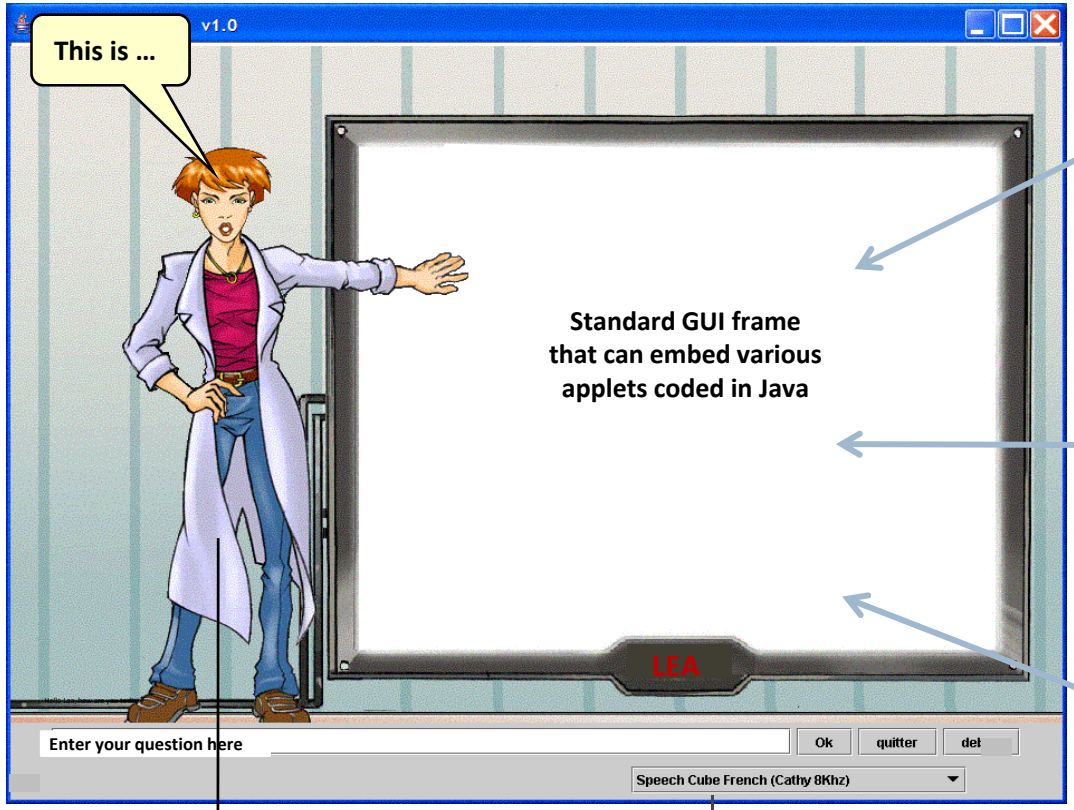


Conclusion: genericity can be reached only through an observation of real novice user's requests interacting with ACA embedded into different applications.

The Daft corpus collection process

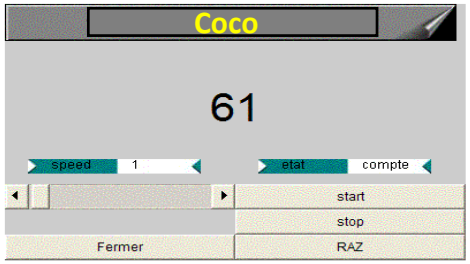
- Do we really need a new corpus? « yes » (hypothesis)

Interface of the Java Conversational assistant: DaftLea (LeGuern, 2004)

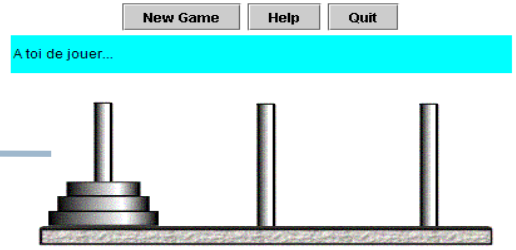


Embodied agent LEA, developed by J-C Martin et al. in the EU NICE project (2002-2005)

Text to speech synthesis (Acapela)



Component "Counter"



Component "Hanoi"

Component "AMI web site"

An excerpt from the Daft corpus

Orthographic noise
Idiosyncratic noise

a+
ah
à l'aide !
Allez à la page des projets
Allez ciao.
Alors ça vient?
alors là t'es completement paumé !
alors lâ t'es completement paumé !
a plus
appelle moi simplement … Sylvie
Appelles moi le manager du site
à quoi penses tu?
A quoi sers-tu?
A quoi sert-tu dis moi un peu?
As-tu des amis?
as tu des idées sur la manière de modifier cette pge ?
as-tu des informations sur les membres du GT ACA ?
as tu des informtion sur comment on peut s'abonner?
as tu entendu parler de Jean-Pierre Durand ?
auf viedersen
au revoir mon vieux
au sujet de cette page, que peux tu dire ?
avec ce corpus, tu sauras ce qu'est une anaphore ...
avec quoi je reviens?
bah!
Bah tu viens de dire que tu pouvais remonter le moral !
barre toi de là
ben alors reponds !!!!!!!!
be ouais tu comprends pas
Bizarre, si je clicke sur le lien du bas ça fait rien
bon
bon â rien !
bon, ça va comme ça !
Bon, dis-moi plutôt ce que tu sais faire plutôt
que de me montrer que tu ne comprends pas
ce que je dis
Bon je me casse. Bye.
bon j'en ai marre je me tire ...
bjr Marco
bonjour, Marco. Qu'est-ce qui te différencie
d'un robot anthropoide?
bonjourmon vieux
bon, reviens à la page d'accueil du site
bon week end
bon y a rien a tirer de toi !!
…

2 sentences

Daft corpus = 11 000 requests (2008)

Corpora comparison: is Daft different?

A generalist corpus:

- MULTITAG (Paroubek, 06): sentences from *Le Monde* newspaper and french novels.

  Methodology: statistical comparison

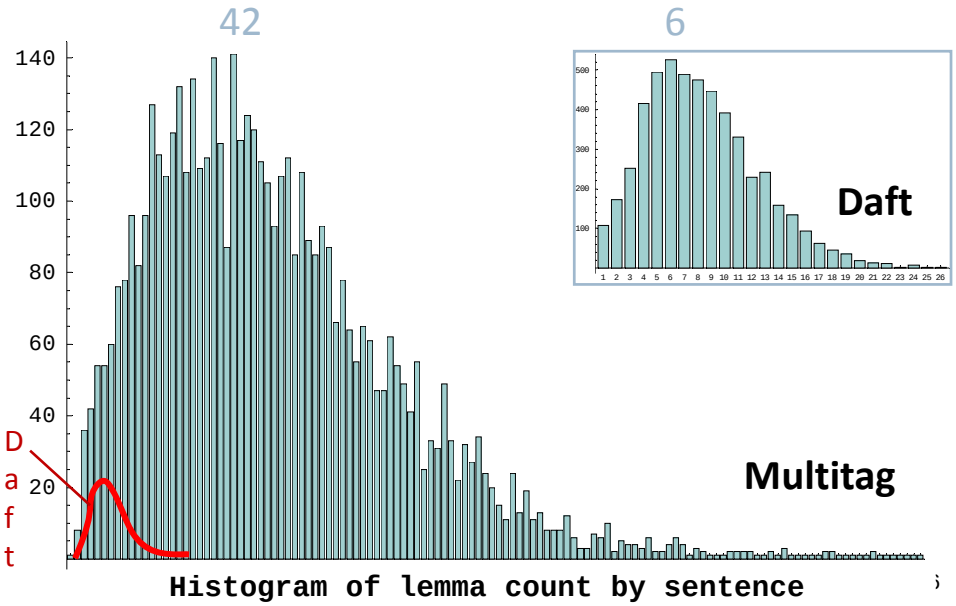
  Result: Variety of vocabulary and lexical diversity are quite restricted

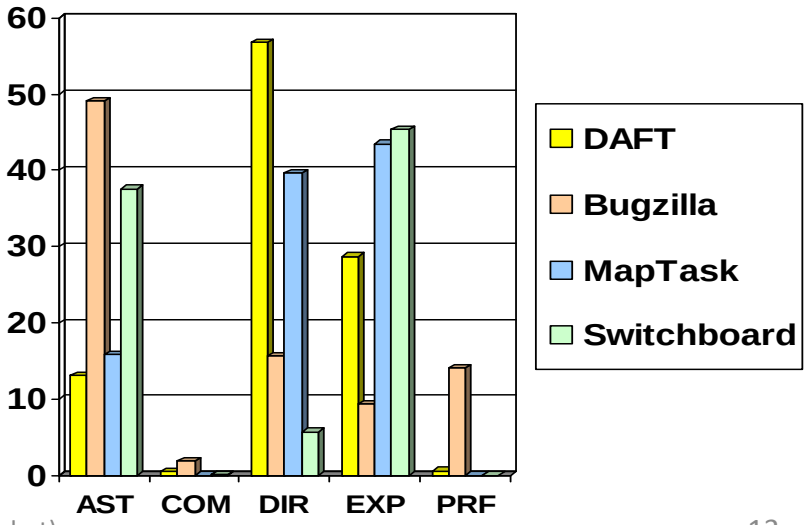
Task-oriented corpora:

- BUGZILLA: 6 000 000 comments about correcting Mozilla bugs
- MAPTASK (Jurafsky et al., 98): 128 dialogues about the building of a geographical map
- SWITCHBOARD (Carletta et al., 96): 200 000 utterances in telephonic conversations

  Methodology: speech act conversion into a similar ontology (Searle, 69) (Ripoche, 06)

  Result: There is a clear "not a human" effect:
  - More Directives (DIR)
  - More Performatives (PRF)
  - More Expressives (EXP)
  - Less Assertives (AST)
  - Lack of Commissives (COM)





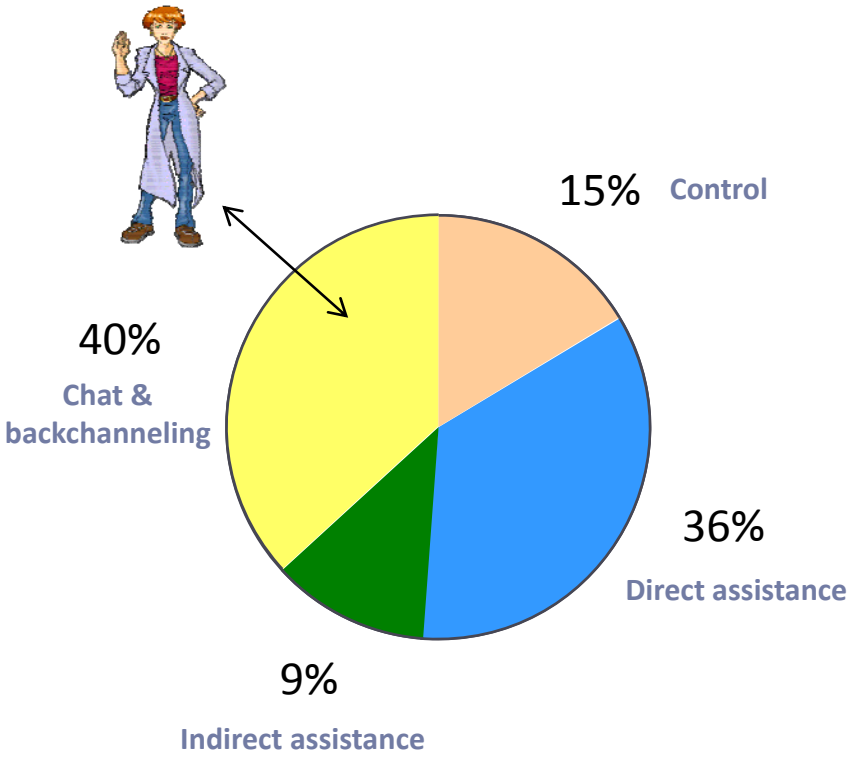
Conclusion: assistance is a linguistic genre and corpus was needed (Bouchet, 06)

Corpus analysis

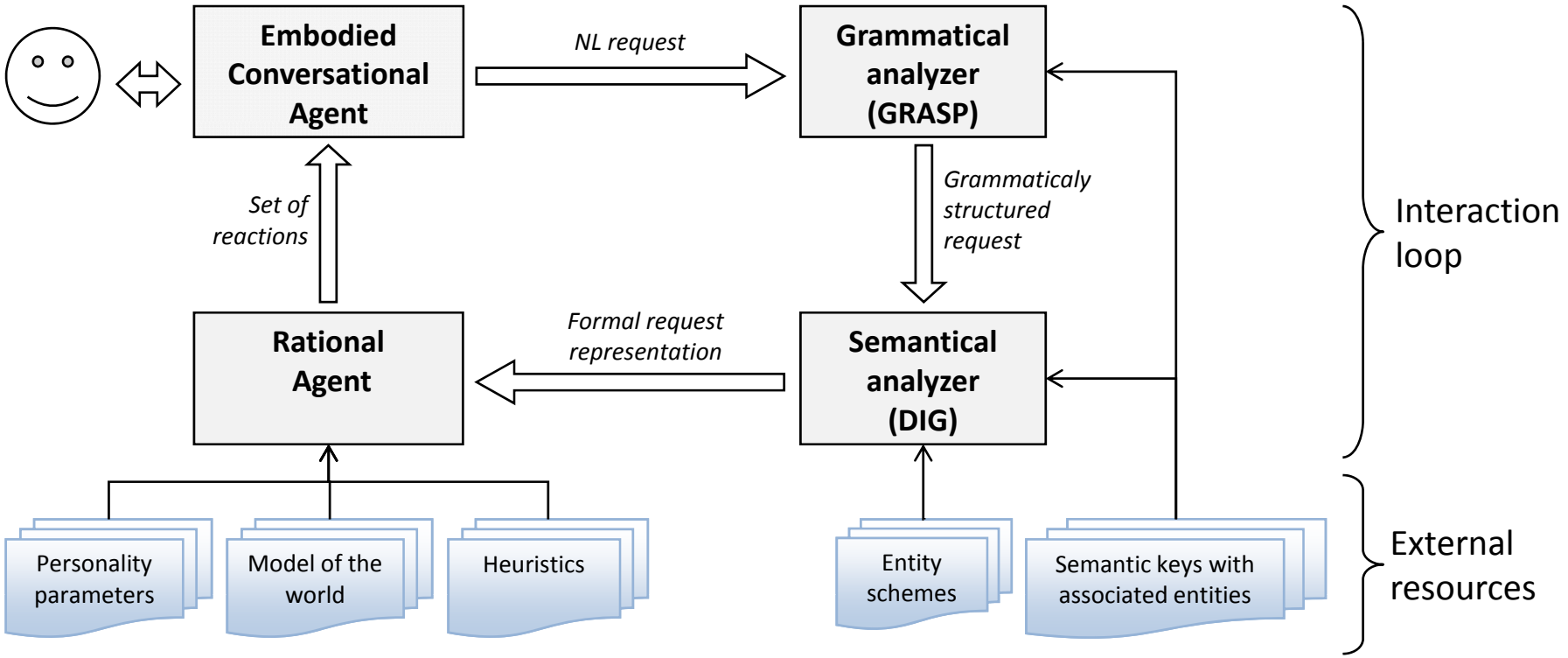
4 different conversational activities:

- **Control**: to make the agent interact himself directly with the application software in which it is embedded
  "Click the little red button"
- **Direct assistance**: explicit help request
  "Can I click the red button?"
- **Indirect assistance**: user's judgments implying a need for assistance (requires pragmatics consideration)
  "Pity the red button is not clickable"
- **Chat**: reactions to agent's answers, communicative functions and dialogue about the agent itself
  "Have you ever been to Brazil?" *(not in the corpus)*

Possibilities to automatically identify requests conversational activity, combining classifiers (KNN, Naive Bayes…) using a vector representation of the formalized request (Bouchet & Sansonnet, to be published)



DAFT system architecture



Formal Request Language (FRL)

- Methodology: corpus-based, for a tailored language (Bouchet & Sansonnet, 07)
- Frame-like approach to represent semantic: a kind of extended subset of FrameNet (Fillmore & Atkins, 98)

```
Modify[
      {object, concept} element
      {manner} manner
      {property} ppt
]
```

- Manual annotation allowed to define 4 classes of frames:
  - 44 Modalities: POSSIBILITY, KNOWLEDGE, LIKE, ASK, NEG…
  - 66 Actions: Click, Modify, Go…
  - 3 References: Person, Object, Concept
  - 124 Properties: color, editable, quantity, size…

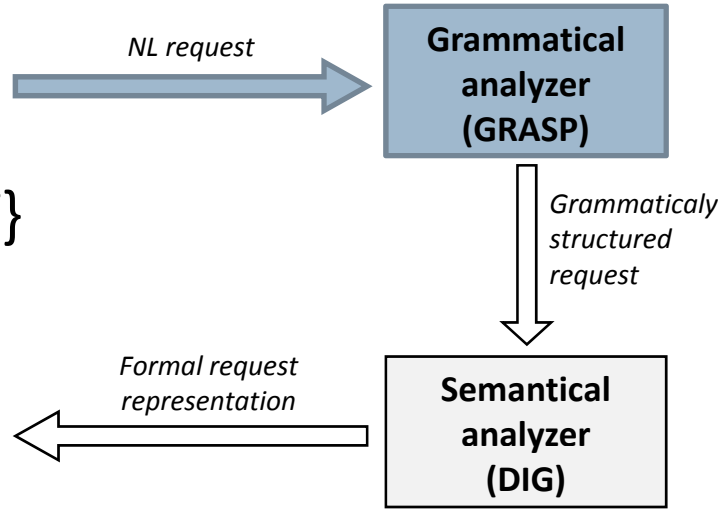
(Resource available online on www.limsi.fr/Individu/bouchet)

Example of request creation: lemmatization, semantic keys association

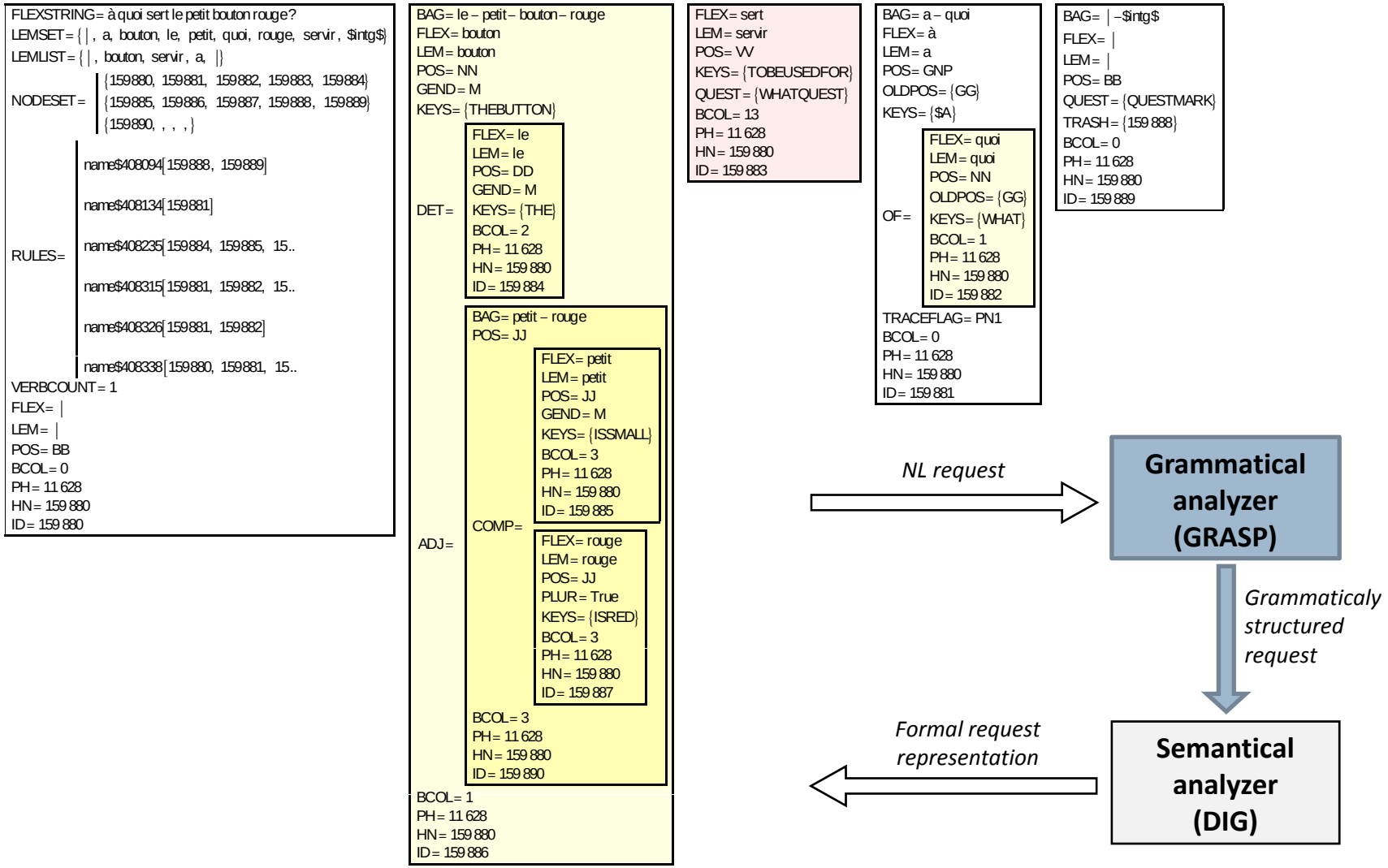
« A quoi sert le petit bouton rouge ? »
= « What is the little red button for? »

Lemmatization:
{|, a, bouton, le, petit, quoi, rouge, servir, $intg$}

Semantic keys association:
{$A, WHAT, TOBEUSEDFOR, THE, ISSMALL, THEBUTTON, ISRED, QUEST}



Example of request creation: grammatical analysis

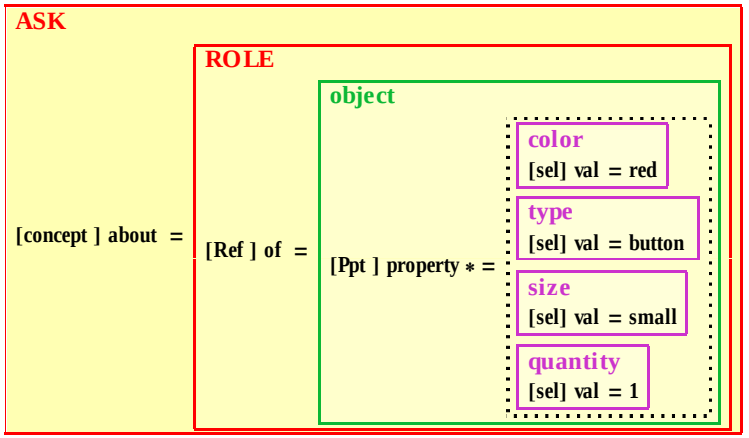


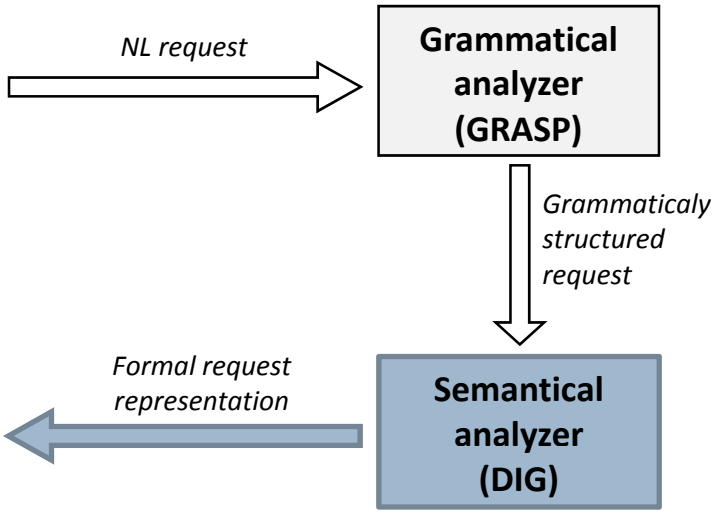
Example of request creation: semantical analysis

```
<?xml version="1.0" encoding="UTF-8" ?>
- <requests>
- <request name="ASK">
 <class>M</class>
 <type>concept</type>
 <node>159894</node>
- <fields>
- <field type="{concept, Act, object, person}" attribute="about">
- <request name="ROLE">
 <class>M</class>
 <type>concept</type>
 <node>159894</node>
- <fields>
- <field type="Ref" attribute="of">
- <request name="object">
 <class>R</class>
 <type>object</type>
 <node>159896</node>
- <fields>
 <field type="sel[application, area]" attribute="identifier" />
- <field type="Ppt" attribute="property*">
- <request name="size">
 <class>P</class>
 <type>size</type>
- <fields>
 <field type="sel[small, medium, large, short, long]" attribute="val">small</field>
 </fields>
 </request>
 </field>
 <field type="Act" attribute="doing" />
 <field type="Act" attribute="subject-of" />
 <field type="sel[True, False]" attribute="_locked" />
 </fields>
 </request>
 </field>
 <field type="object" attribute="regarding" />
 </fields>
 </request>
 </field>
 </fields>
 </request>
```

XML File

Graphical representation





(Interactive demo available online on www.limsi.fr/Individu/bouchet)

Back to Genericity: request classes

- Problematic: genericity-relevance trade-off
  (⇔ overfitting problem)
- Which level to consider?
  - We define both high and low level classes
  - They match concurrently the request, similarly to what exist in robotics with subsumption architecture (Brooks, 91)

- 32 high level request classes:

  ¬Auth[a], ?Way[a], ?Reason[s], ?Instant[a], ?Place[r], ?Effect[a/s], ?Can[a], ?Meaning[r/a], WantKnow[r/s], WantTodo[a]…

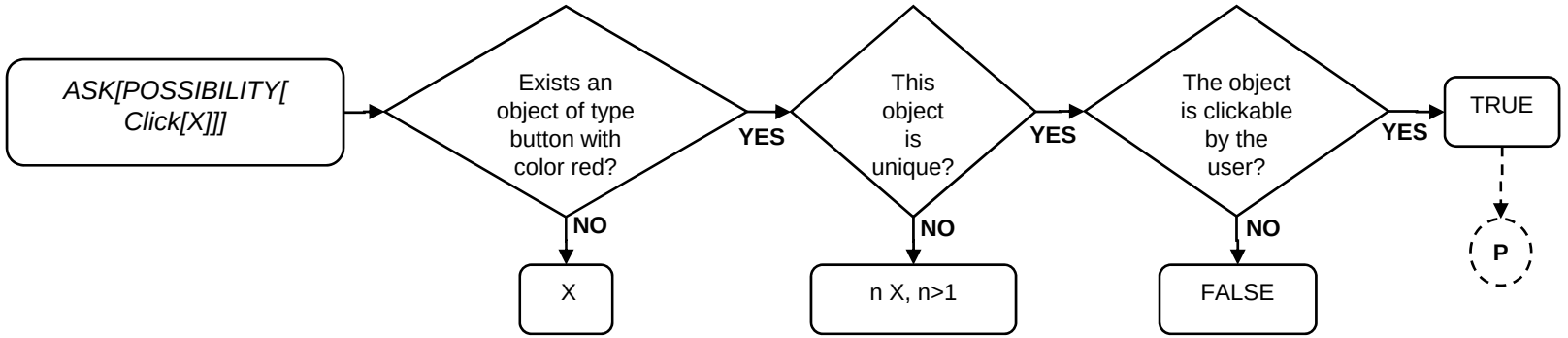
- Some lower request classes:

  RequestAction[a, DefiniteSingularReference]

  RequestAction[a, UndefiniteSingularReference]

Rationality: heuristics associated to request classes

« Can I click the red button ? »



Rationality: heuristics associated to request classes

« The button is not working »

Should we…

- explain why?
- fix it without explanation?
- fix it and explaining why it was not working?
- do the action that is executed by the button?

All those behaviors can be seen as rational/irrational depending on the context.

Rationality is not enough: need for personality

- An agent without personality would look incoherent:
  - Unable to choose the best behavior, it would have to take random decisions.
- An agent without personality is perceived to have one (Dryer, 1999)
- Rationality can't be achieved without pieces of subjectivity.

- Defining a personality = a way to keep agent's coherence, and hence its reliability from the user's point of view.

- How to define a believable personality?
  - Bringing back the novice into the loop, by making him become the assistant, to ground heuristics into reality.

Conclusion

- Assistance need is crucial, and ACAs are the best way to provide it (if they manage to be made efficient enough)

- Reducing the development cost of ACAs requires finding genericity in user's requests
- We need representative requests: use of a specific corpus
- Genericity is achieved through the definition of classes of requests. This requires to focus on the semantic of the request, which need to be transformed thanks to a formal request language based on the collected corpus.

- Multimodal reactions are associated to classes of requests
- Often, even on apparently rational question, rationality is not enough to choose => agent need a personality.