A corpus-based NLP-chain for a web-based Assisting Conversational Agent

Mao Xuetao
LIMSI-CNRS
BP 133 F-91403 ORSAY Cedex
France
xtmao@limsi.fr

Jean-Paul Sansonnet
LIMSI-CNRS
BP 133 F-91403 ORSAY Cedex
France
jps@limsi.fr

François Bouchet
LIMSI-CNRS
BP 133 F-91403 ORSAY Cedex
France
bouchet@limsi.fr

ABSTRACT
Assisting Conversational Agents are Embodied Conversational Agents dedicated to the Function of Assistance for applications and services to the general public. Assisting agents for the general public are more and more required on the Internet-based new rich-client applications. We have developed a web-based framework to experiment with assisting agents, called the DIVA toolkit, where the Function of Assistance is a key issue, and the Natural Language modality a primary concern. This is why the DIVA toolkit is based on a **N**atural **L**anguage **P**rocessing chain (**NLP**-chain) handling the users' textual questions about the structure and the functioning of the DIVA web pages. This paper describes the architecture of the NLP-chain and focuses on the corpus-based approach developed so as to provide an actual grounding for the intermediate **F**ormal **R**equest **F**orm (FRF), at the heart of the NLP-chain.

Keywords
Web-based Agents, Corpus of assisting requests, Natural language request handling.

1. INTRODUCTION

1.1 Assisting Conversational Agents

1.1.1 Contextual help Systems
The problem of assistance in web-based applications has long been addressed in various ways as the number of ordinary users was increasing. A mere transposition of a paper-based documentation into an electronic version has been shown to be actually less efficient [16], which has led to focus on the notion of *adaptation*, to bridge the semantic gap between novice users and expert developers. The need for an adaptation of help systems according to the user's current task has progressively led to current research on Contextual Help Systems (CHS), which have been proven to be more efficient than non contextual ones in many cases [16]. In the same way, help systems can be adaptive depending on the profiles of the users: statically through the use of classes of users [35] or dynamically with regular updates of a model of the current user [27].

1.1.2 Conversational Agents for assistance
From a computational point of view, the main qualities of a help system are the *precision* and the *completeness* of its informational content. But when general public users are concerned, it has been shown that *ease of use* is the primary factor, otherwise the help system is merely left aside, users preferring to ask from friends "behind their shoulder" [17]. In relation to this statement, recent studies have shown the positive impact of multimodality for help systems, and particularly the linguistic modality [19] (i.e. natural language interaction), since it allows a clear separation between the user's task in the application and its interaction with the help system. One of the consequences is the development of **E**mbodied **C**onversational **A**gents (ECA) able to interact with users through multimodal dialogues [21].
ECAs have shown that a human-like presence can have a positive impact on the global acceptance of the system, thanks to increased agreeability and believability (this is the 'Persona Effect', as described in [29]). However, despite all the potential benefits, the use of ECAs as a support for the Function of Assistance raises two main issues:
- The first issue is that the *efficiency* of an ECA is more critical than with traditional help systems since, as explained in [33] with the example of the 'Clippie Effect' of Microsoft Office assistants. On the opposite, ECAs have been successfully used for training users to various tasks [34][20], but a lot of work remains to be done to attain a good efficiency and we think that it entails a thorough analysis of the Function of Assistance through a study of novice users' requests.

- The second issue is related to the choice to handle the *natural language modality*. One could argue that skilled users prefer interacting directly with Graphical User Interface (GUI) elements rather than using natural language for the control of software applications. But on the other side, when assistance is required, novice users (and sometimes expert ones too) have been shown to spontaneously express their frustration in natural language in front of the machine (not unlike the 'thinking aloud effect' [36]). Indeed, it seems that the natural language modality surges when things tend to go wrong. This is the reason why we give it a first class citizenship in the analysis of the Function of Assistance.

1.2 Web-based assisting agents

1.2.1 Web-based virtual agents
When we consider the issue of helping general public novice users, the web provides a large domain of applications and services that could be improved by hosting assisting agents. This is the reason why we are currently developing our research on the Function of Assistance in the context of web-based applications. To this purpose, we developed the DIVA toolkit which is dedicated to the support of assisting virtual agents on the Internet.

Nowadays, many web pages display virtual agents. They serve two main purposes:

1) *Informational* agents give predefined static information about a site or about a specific product within a site. Users cannot interact with the agent which is just a 'speaker'. Oddcast corp. is a major

provider of this technology on the web for several general public big corporations [1].

2) *Chatbot* agents are long time successors of ELIZA [38] which was originally designed as textual input/output interaction with a fake Rogerian therapist. Today outstanding web-based chatbots are [2][3][4][5]. They tend to incorporate speaking characters in order to personify the chatbot. This kind of technology has been used in many corporate web sites for the welcoming and the guidance of general public.

There are three main drawbacks to these frameworks:

1) The virtual characters are physically encased into a 'box' (<iframe>, Flash™ objects…), for instance on the top left-hand corner of the web page, and their interaction with the rest of the page is minimalistic: no drag-and-drop over the page, no deictic gesture, etc.

2) The agents have little or no access to the DOM-structure and to the informational content of the web page. This makes it very difficult for them to interact a) with the page content (no browsing and action on DOM objects of the page…) and b) with the users, e.g. for assisting purposes.

3) The Natural Language Processing (NLP) tools used in the chatbots are often trivial, mainly based on keyword matching, though Wallace's Alicebot [4] offers a more sophisticated approach using rules and meta rules described in XML (AIML — Artificial Intelligence Markup Language). Hence, if they truly succeed in the chatting context, the semantic analysis of the utterances is insufficient to support questions about the structure and the functioning of an application.

1.2.2 Objectives of the DIVA toolkit

DIVA stands for **D**OM **I**ntegrated **V**irtual **A**gents: emphasizing the *unique* feature of DIVA virtual agents that are completely integrated with the DOM (Document Object Model) tree structure of web pages. The DIVA Toolkit has been originally developed at LIMSI-CNRS for master-level teaching and research purposes on web-based Embodied Conversational Agents (ECA); it can be freely downloaded at the DIVA home page [6]. The main objectives of the DIVA toolkit are:

- A toolkit dedicated to Assisting Conversational Agents, where textual natural language interaction plays a primary role;
- An open programming framework making it easy and quick to develop and deploy new *experimental* ACAs in web-based applications;
- A completely DOM-integrated software architecture making it easy for the DIVA ACAs to access, both in read/modify modes, to the inner structure of the applications/services supported by the web pages.

1.3 Corpus-based Approaches

Corpus-based approaches are often used in linguistic researches to achieve a computational analysis in a specific domain of knowledge. [23] reports a new approach to automatic generation of back-of-book indexes for Chinese books using a corpus-based statistical algorithm. [13] automatically extends downwards an existing biomedical terminology by removing adjectival modifiers from terms extracted from a corpus of three million noun phrases extracted from MEDLINE and searching for demodified terms in the terminology. [32] argues that corpus based methods can be used in natural language generating (NLG) as it is used in natural language understanding (NLU) as well. Public large corpora are usually used to produce sub-corpus for a special purpose. Then operations such as extracting, filtering, sorting and statistical analysis are applied on the sub-corpus to produce a new corpus. The results of corpus manipulations are also often used to evaluate a given research theory or method.

In the linguistic perspective, DIVA ACA is also working in a NLU and NLG flow, which we call the NLP-chain that can employ corpus-based approaches too. Different corpora are used to do statistical analysis and comparison that will be described in section 3:

- Daft corpus is a French corpus created in DAFT project during 2004~2006 in LIMSI;
- Diva corpus is a French-English multilingual corpus created in DIVA project during 2007-2008 in LIMSI.

This paper is organized in three parts: the next section is dedicated to the description of the general architecture of the toolkit. In section 3 we present our corpus-based approach to the construction of the ontology of concepts used in the NLP chain. Finally, in section 4 we present the implementation of the complete NLP chain.

2. ARCHITECTURE OF DIVA

2.1 Web architecture

The web architecture of DIVA is displayed in figure 1. It is composed of two layers: a symbolic-server layer dedicated to data base resources management and symbolic computing and a rich-client layer supporting:

- The specific application/service web page;
- The animation of the graphic characters;
- The processing of the textual natural language interaction.

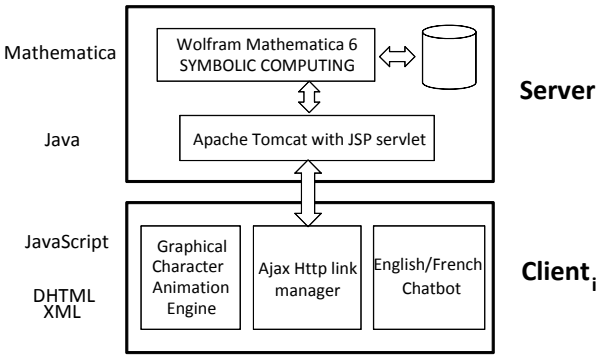

Figure 1: Web architecture of DIVA

The DIVA toolkit is a full Web 2.0 rich client technology based on JavaScript, JSP and Mathematica from Wolfram Research. It was intended to work with Trident (IE 6+), Gecko (Mozilla Firefox…) and WebKit (Safari…) layout engines. DIVA can be deployed in two main modes: 1) the rich-client mode enables web-apps to work locally, i.e. without server capabilities; 2) the symbolic-server mode offers more powerful web architecture but it requires an additional Mathematica engine.

2.2 NLP architecture

2.2.1 The assisting loop

The toolkit provides a Natural Language Processing (NLP) architecture which supports the assisting agent. Indeed, a typical assisted application is composed of two main software parts:

1) *Application Model*: this is the *domain-specific* part that contains a) the actual application code (mainly JavaScript) and b) the modeling files containing the description and help information about the application (mainly external XML files).

2) *Assisting Agent*: this is the *generic* part that contains the domain independent tools: a) the NLP tools translating the textual requests into the Formal Request Language (FRL); b) the rule-based symbolic processing tools providing a library of standard reactions to FRL requests while browsing the application model.

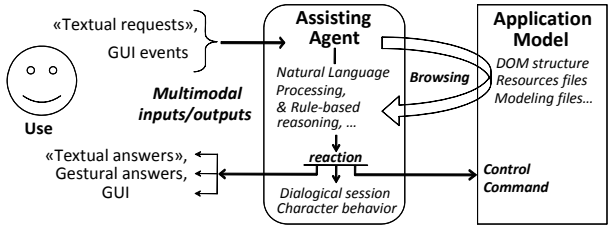

Figure 2: The typical path of a user's request

Erreur ! Source du renvoi introuvable. shows the typical path of a user's request:

1) The users can put textual utterances into the chatbox field. Alternatively, when the users trigger GUI events (mainly mouse & key events) the later are coerced into textual forms (e.g. "USER MOUSEDRAG") so as to unify the agent's multimodal inputs in a single formalism;

2) The textual input is transformed into a formal request by the NLP tools – this can require a customization phase;

3) A formal request is then 'resolved' by applying a list of so-called *semantic spaces*. A semantic space is a package of symbolic rules dedicated to a particular semantic domain. The rules browse the application model to retrieve the relevant information they need and build a formal reaction;

4) A formal reaction made of three main parts: the answer part of the reaction is sent to the user through multimodal devices; the control/command part of the reaction is applied to the runtime of the application; the dialogical part of the reaction updates the dialog session and the behavioral model of the character.

2.2.2 The NLP chain

Like most chatbots frameworks, the DIVA NLP-chain is based on pattern matching rules. Typical chatbot architectures are organized in two simple layers/phases:

1) The chunking phase: the textual utterance is trimmed from punctuation, accentuated characters, etc. and coerced into uppercase words which are finally stemmed (plural 's' or conjugation forms like 'ed' are systematically cut).

2) The rule phase: the chunked sentence is then filtered by an ordered list of rules of the form (pattern → reaction); generally the first rule that applies ends the filtering.

The DIVA NLP-chain is also composed of two layers/phases but with a more sophisticated structure as shown in figure 2:

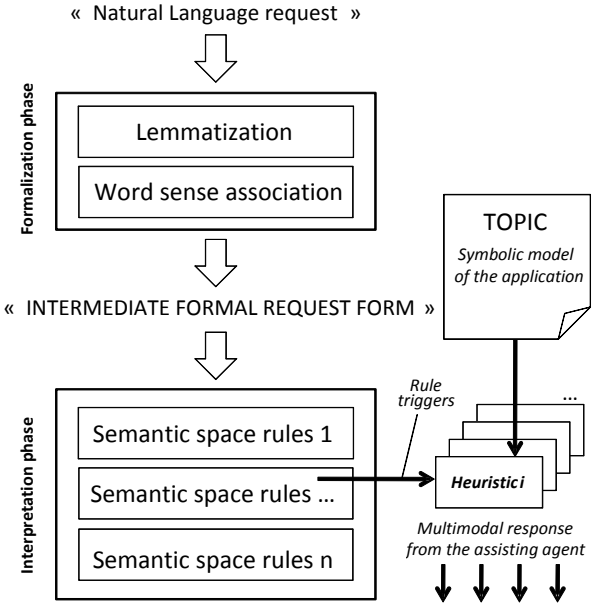

Figure 2: The DIVA NLP chain

1) The formalization phase: it is based on two sets of filtering rules applied in sequential order:
- Syntactical level: like in a typical chunking phase, the textual utterance is trimmed from punctuation, accentuated characters etc. coerced into lowercase words but no stemming is done so as to preserve the syntactical information. Word flexions are transformed into lemma (root words);
- Word-sense association level: lemmas are then transformed into semantic 'synsets' as in WordNet [24][1].

At the end of the formalization phase, the request is transformed into an intermediate formal form, called the Formal Request Form (FRF). In the FRF, a request is expressed by a sequence of abstract keywords, each of them being associated to a semantic concept defined by a gloss. We have defined our own simplified ontology of concepts, keywords that will be referred to in this paper not as 'synsets' (as in WordNet) but as 'semantic keywords' or in short 'keys'.

2) The interpretation phase: it is based on a set of rules of the form *pattern* → *reaction* where the *pattern* is expressed in terms of the FR of the request and the *reaction* is a procedural heuristic defining the behavior of the agent in response to the user's request. To build a reaction, the triggered heuristic can exploit two kinds of information: a) a representation of the current dialogical session and b) a symbolic model of the application, called the 'topic' that describes its specific features.

Actually, the set of interpreting rules is organized into several subsets, called 'semantic spaces' or in short 'spaces'. Each space is dedicated to a specific conversational activity or topic.

[1] Here, the issue of Word-Sense Disambiguation (WSD) is not much of a problem because we are often in a "One sense per discourse" situation, along with Z. Harris and more recently Yarowsky [25].

3. A CORPUS BASED APPROACH

3.1 The Daft corpus

3.1.1 Corpus objectives and collection

In parallel with the development of the DIVA toolkit we are currently studying the Function of Assistance in the Daft Project [28]. In this work, a primary requirement was to collect data in the context of ordinary people asking natural language general requests when they face an unfamiliar application. This led to the collection of a corpus of 11000 requests (referred to below, as the Daft corpus), gathered from three different sources, each of them providing roughly a third of the total corpus size:

1) During about two years (2004-2006), a hundred of human subjects not familiar with computer applications have been asked to use various applications of moderate complexity (three Java applets and two websites including an active one which could be dynamically edited by users), and to interact when needed with the embedded animated conversational agent connected to a first version of the Daft system (hence providing a basic feedback). This methodology allowed us to have a corpus *grounded into reality*.

2) Using two thesauri [30][12], we have manually constructed new requests based on the already collected ones, in order to provide a *wider linguistic coverage* of the assistance vocabulary.

3) In a last step, we have added to the corpus some FAQ extracted *from integrated help systems and websites concerning two widely* used document creation software (LaTeX and Microsoft Word), to have examples of requests uttered while using more complex applications than the ones previously tested.

The use of those three complementary methods to build the Daft corpus allows us to have a rather representative corpus of assistance requests, despite its rather small size coming from the difficulty to find a lot of available ordinary users.

The Table 1 shows selected excerpts from the collected part (part 1) of the Daft corpus, which reveals some of its characteristics (emphasized in bold):

- more than half of the user requests are not well-formed (expressions from the spoken language, spelling, syntactic or grammatical mistakes, acronyms from SMS and internet slang...) and some of those mistakes are not easy to detect and fix with classical natural language processing tools,

- requests are not stored as part of a dialogue, but as isolated sentences, since as mentioned by [18], in the domain of assistance, dialogical interactions are almost always limited to a single conversational turn and hence can, most of the time, be treated as isolated requests.

3.1.2 Analysis of the requests specificity

Once the corpus had been collected, one of our first objectives was to validate our hypothesis that natural language interaction in the context of assistance is clearly different from other linguistic contexts and hence required indeed the usage of a specific corpus. To exhibit the specific distributionality of the Daft corpus, a first comparison has been done with a generalist corpus made to represent the variety of natural language. We have chosen the Multitag corpus, which is made of sentences from the French newspaper *Le Monde* and novels [31]. We have kept only a subset of it in order to have the same number of different words as in the Daft corpus.

- The first observation is the variety of vocabulary of Multitag compared to the Daft corpus: 1460 sentences from Multitag have the same semantic variety as 5000 requests from the Daft corpus.

- The lexical diversity of Multitag is also greater: it contains over 3620 different lemmas, against only 1788 in the Daft corpus (with 1188 common lemmas between the two corpora).

- Finally, when doubling the size of the Daft corpus, whereas only 250 new lemmas are needed to cover them in the case of the corpus, Multitag requires introducing 1105 new lemmas.

This simple comparison is enough to show that the Daft corpus requests are clearly not comparable to any sentence of the natural language from a lexical point of view, as it uses a distinct and restricted vocabulary.

Table 1: Selected excerpts from the Daft corpus

N°	***Original collected request (in French)***	***Translation in English (including mistakes)***
1	**cliques** le bouton quitter	**clicks** the quit button
2	**cliquesur** le bouton retour	**clickon** the back button
3	ok, reviens à **l apage** d'accueil	ok, come back to **th ehomepage**
4	donne moi un plan du site	give me a map of the website
5	à quoi sert cette fenêtre**,**	what is this window for**,**
6	**c koi** le GT ACA	**WDYM** by GT ACA
7	est-ce que le bouton "fermer" et le bouton "quitter" fonctionnent exactement pareil ?	do the "close" button and the "quit" button work exactly the same way?
8	j'ai une question à poser à un des membres, comment je peux le joindre ?	I have a question to ask to one of the members, how can I contact him?
9	quand est la prochaine **reunion** ?	when is the next meeting?
10	où **peux t-on** trouver le programme de la conférence?	where **can-we** find the conference schedule?
11	existe t-il une version condensée de l'aide	is there a shorten version of the help
12	je ne **vosi** aucune page de **demso** !!	I **cna't** see any **demso** page!!
13	le lien me semble **cassée**	the link seems to me to be broken
14	j'ai été vraiment surpris de constater qu'il manque une fonction d'annulation globale	I was really surprised to see there's no global cancel function
15	ça serait quand même mieux si on pouvait aller directement au début	it'd be better to be able to go directly at the beginning
16	**auf viedersen**	**auf viedersen**
17	espèce de bon à rien !	you good for nothing!
18	Quel genre de musique tu aimes ?	What kind of music do you like?
19	tu t'habilles tous les jours de la **meme facon** ?	you're dressing the same way everyday?
20	ca marche **:-)**	works for me **:-)**

3.1.3 Characterization of the conversational activity

During the corpus collection phase, human subjects were requested to do some tasks for which they could ask help (if needed) from an artificial assistant agent embedded in the program to assist them. Subjects were completely free to act and particularly they could type what they wanted without any constraint. Consequently, various behaviors have been observed, with users sometimes completely abandoning their original task, and it eventually appeared that many of the collected sentences were not really linked to the assistance domain (cf. Table 1).

Hence we got interested in trying to identify and categorize those other conversational activities that were appearing in the corpus. For this purpose, we have randomly extracted from the actually collected part of the corpus (i.e. without any request from the manually built up parts mentioned earlier) two subsets of sentences, both having a size equal to the tenth of the total corpus size. In the first subset, we have manually gathered sentences by similar activities, which means depending on the user's intentions when he typed his request. This allowed us to distinguish four main activities with unequal repartitions (as shown on figure 3).

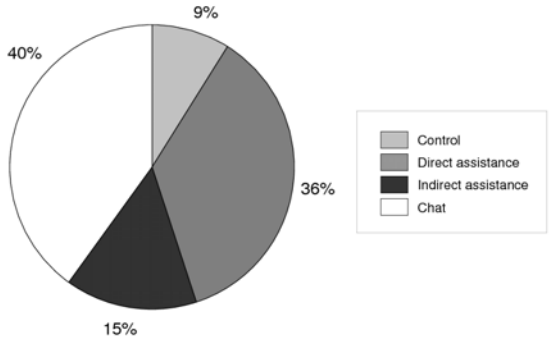

Figure 3: Distribution of conversational activities in the corpus

Knowing those categories, we have annotated the second subset to classify manually each sentence in one of those. The distribution determined this way was very close from the first subset, allowing us to generalize this result to the rest of the collected corpus. We can then consider our collected corpus can be divided into four 'subcorpora', each corresponding to a particular activity (the numbers given below as examples refer to sentences in Table 1):

1) **Control activity**: corpus made of direct *controls*, to make the agent itself interact directly with the application software in which it is embedded (sentences 1-4).

2) **Direct assistance activity**: corpus gathering *help requests* explicitly made by the user (sentences 5-11).

3) **Indirect assistance activity**: corpus made of user's judgments concerning the application that are actually implying the fact the user is in need of assistance; it certainly requires the system to use pragmatics to detect the implicit meaning (sentences 12-15).

4) **Chat activity:** corpus with all other activities which are not in direct relation with the application and often oriented towards the agent itself (sentences 16-20). The chat activity amounts to 40% of the total users' utterances. As we are mainly interested in control and assisting activities, one can view chat activity as 'noise' but we have to deal with it because control and assisting activities are in fact deeply embedded within chat activity. Indeed, according to the Hofstadter's 'Eliza effect' [22][26], the personification of the assisting agent by a virtual character prompts the users to enter into affective chat activity.

A more detailed analysis of the Daft corpus can be found in [15].

3.2 The DIVA corpus

3.2.1 A corpus dedicated to web-applications

In the context of natural language assisting requests we are interested in building a more specific corpus (registered in a situation of interaction with a DIVA agent on a web page) with two objectives: a) it is in English in order to provide at least two languages as the DIVA textual input modality b) it is should be small in order to exhibit a simplified set of semantic concepts to support the FRF, say less than 500 keys, as defined above. To this purpose, we constituted the so-called DIVA corpus that contains only 1612 utterances collected from various sources:

- 500 utterances are a random excerpt of the Daft corpus translated from French to English;
- 275 utterances where taken from the so-called "Marco corpus", registered with colleagues asking information about the website of the French Research group on Animated Conversational Agent [7]. It is also a translation from French to English;
- 875 questions were registered with ordinary people asking information about the DIVA web pages displaying various topics: 90 questions about a 'book ticket service', 282 questions about 'Mao Xuetao' as an ordinary person, 282 questions on 'Mao Zedong' as a famous person, 96 questions on a car company 'Renault', 125 questions about a product of this company, that is a 'Renault Scenic car'.

All the utterances add up to 1650. We dropped some repeated utterances and got the final DIVA corpus with 1612 items.

3.2.2 Extraction of the semantic keys

As shown in figure 2, we use an intermediate form between the syntactic layer and the interpretative layer. This form is the result of the translation of the word lemmas into their associated concepts, the semantic keys. We carried out a qualitative analysis upon this corpus in order to exhibit the occurring lemmas and to group them as synsets, i.e. semantic classes represented by a unique keyword (an UPPERCASE symbol).

The total number of keys is 436, divided into six main classes:

NAMES LIST	**132**
CATEGORIES LIST	**20**
VERBS LIST	**115**
ADJECTIVES LIST	**60**
LOCATIONS LIST	**23**
GRAMMATICALS & SPEECH ACTS LIST	**86**

Table 2 gives an excerpt of the DIVA keys together with their 'meaning', expressed as a shortened *gloss* sentence — à la WordNet. Keywords are conventionally built by choosing, as their root, the most salient lemma of its synset preceded by a tag indicating one of the classes above: THE- for nominal forms, TO- for verbal forms, IS- for adjectival forms, ISA- for categories etc. Other concepts, like grammatical ones, have no tags.

Table 2: Excerpts from the 436 semantic keys of DIVA

Keys	Gloss
TOWORK	Denotes the general activity of achieving some work
TODERIVEFROM	Denotes the abstract action of inheriting/deriving its characteristics from something
TOKNOW	Denotes the mental action of knowing something
TOHAVE	Denotes the grammatical auxiliary verb: to have
TOTAKE	Denotes the action of taking/receptionning something
TOMEET	Denotes the action of contacting/meeting somebody
TOEXIST	Denotes the abstract action of existing
TODO	Denotes the action of doing something
TOCAN	Denotes the abstract action of having the general capacity or right of doing something
TOSAYPLEASE	Denotes the expression of saying please to somebody
TOSPEAK	Denotes the action of speaking
TOLIKE	Denotes the mental action of liking/loving something/somebody
TOWANT	Denotes the mental action of desiring/wanting something or a state of affairs to happen
TOSAYHELLO	Denotes the expression of greeting somebody
TOSAYBYE	Denotes the expression of saying bye to somebody
TOPROVIDE	Denotes the general action of giving/sending something or some information to somebody/a system
TOOBTAIN	Denotes the general action of obtaining/acquiring something or some information
THEAVATAR	Denotes the graphical/dialogical assisting character of the application
THEHELP	Denotes the service/help provided by somebody
THEMAXIMUM	Denotes the maximum value that a variable can take
THEUSER	Denotes the user of the application at first person: I, me, myself
THETITLE	Denotes the title of a window or a frame in the window of the application
THEPICTURE	Denotes a picture in the window of the application
THECHOICE	Denotes the name of the mental action of choosing something
THEBELIEF	Denotes the name of the mental action of believing/supposing something
THENUMBER	Denotes the count of something/persons
ISHONEST	Denotes the quality of somebody who is honest/sincere
ISFEMALE	Denotes the quality of a person with gender: female
ISREAL	Denotes the quality of something that is real/physical
ISSAME	Denotes the quality of something that is equivalent/identical/similar to something
ISUNPLEASANT	Denotes the quality of something that is unpleasant
ISUNFRIENDLY	Denotes the quality of being unfriendly/impolite with somebody
ISPOSLEFT	Denotes the space position: left
ISPOSRIGHT	Denotes the space position: right
ISMANDATORY	Denotes the quality of something that is legally/physically mandatory/indispensable
WHAT	Denotes the grammatical WH-pronoun: what
WHY	Denotes the grammatical relation: why
WHERE	Denotes the WH-question: asking for the location of something
WHICH	Denotes the grammatical WH-pronoun: which =!= WHOSUB
OWNED	Denotes the quality of relation that the former possess the latter
QUANTSMALL	Denotes the quantity: a small quantity/a small amount
NEG	Denotes the grammatical relation: negation
IF	Denotes the grammatical relation: if
QUEST	Denotes the grammatical relation: question
UNDEFPRON	Denotes the grammatical pronoun: one
LESSTHAN	Denotes the quality of something that is less (according to some quality) than another thing =!= ISLOWERTHAN
IT	Denotes the grammatical pronoun: it
TOBE	Denotes the grammatical auxiliary verb: to be

Here are two examples of users' requests translated into FRF:

REQUEST1 = "*If I want to buy such a car, what can I do?"*

FRF1 = < QUEST IF THEUSER TOWANT TOOBTAIN such a car WHAT TOCAN THEUSER TODO >

The filtering process has extracted 9 keys from REQUEST1 that are put in FRF1 (see their gloss in table 2). Some lemmas are not translated because they are not in the ontology (e.g. *such*, *car*).

REQUEST2 = "*Adopt a less provocative attitude, please."*
FRF2 = < TOTAKE a LESSTHAN ISUNPLEASANT THEBELIEF TOSAYPLEASE > (gloss is in table 2)

For the sake of simplicity in the first version of the NLP-chain of version 1.5 of DIVA a primary requirement was to restrict the number of semantic classes drastically to less than 500. This number says a lot as compared with the synsets of EuroWordnet inter-lingual data base, as shown in Table 3, or the WordNet base which has the thinnest global ontology (>100 000).

Table 3: Snapshot of the EuroWordnet inter lingual ontology, abridged from [37]

		Synsets	No. of senses	Entries	LIRels.	EQRels-ILI	Synsets without ILI
Dutch Wordnet	Nouns	34455	54428	45972	84869	26724	6070
	Verbs	9040	14151	8826	25973	26724	1133
	Other	520	1622	1485	797	n.a.	n.a.
Spanish Wordnet	Nouns	18577	41292	23216	40559	18634	0
	Verbs	2602	6795	2278	3749	2602	0
	Other	2191	2439	2439	10855	n.a.	n.a.
Italian Wordnet	Nouns	30169	34552	24903	83021	43848	98
	Verbs	8796	12473	6607	30757	27941	0
	Other	1463	1474	1468	3290	n.a.	n.a.
French Wordnet	Nouns	17826	24499	14879	39172	17815	16
	Verbs	4919	8310	3898	10322	4915	4
	Other	0	n.a.	n.a.	n.a.	n.a.	n.a.
••• •••							
WordNet1.5, in EuroWordNet format	Nouns	60521	107428	88175	159223	n.a.	n.a.
	Verbs	11363	25768	14734	24331	n.a.	n.a.
	Other	22631	54406	23708	27821	n.a.	n.a.

The small amount of semantic classes has two main drawbacks:
- *Silence*: In the situation of general public users, using a non restricted vocabulary, some lemmas will be absent. They are not translated into keys.
- *Noise*: When lemmas are translated into keys, the small amount of available target classes can lead to false-senses as one can see in the examples above with the associations: 'adopt' → TOTAKE, 'attitude' → THEBELIEF.

In order to estimate the impact of the silence issue, we have carried out an experiment with different web-applications, using the first version of the DIVA toolkit [14]: the number of specific lemma of a given application is quite small (less than 2%) allowing us to envision an *ad hoc* handling, i.e. a quick customization phase for each new application.
As for the noise issue, it can only be reduced by a more thin-grained ontology and more precise NLP tools, in future versions of the toolkit.

4. IMPLEMENTATION

4.1 The formalization phase

4.1.1 Preprocessing

When the user types a sentence in the agent chatbox, the input string is preprocessed according to the following conventions:
► the user string is transformed into lower cases
► in accentuated languages like French, the accents and cedilla are suppressed: "Déjà vu ça" => "deja vu ca".
► some characters are replaced by whitespaces: , . ; : # § % * + - _ ' ( ) { } [ ] < > @ & $. e.g. "(1+1)*2 = 5?" => " 1 1 2 =5?"
► some characters considered relevant are kept: ! ?
► multiple whitespaces are compressed into a single one
► finally the string is bracketed between "< " and " >"

Example:

"Hey!!! 2+2 gives me a ***BAD*** feeling of déjà-vu …"
⇒ < hey ! 2 2 gives me a bad feeling of deja vu >
Then a first set of filtering rules ℜ is applied to the preprocessed form according to the following algorithm:

```
let s be the string input form;
let a be the list of filtering rules ℜ={rᵢ}
foreach rᵢ in ℜ
  while s/:rᵢ ≠ s
    S := s/:rᵢ
   end
return s
```

Where /: is the rule-apply operator. The set of rules ℜ transforms the user sentence into a FRF request: words are replaced by concepts and some primary syntactico-semantic operations are performed, like detecting negation, interrogation, etc.

For example:
"What is the *WORKING* time of the service ????"
⇒ < what is the working time of the service ? >
⇒ < QUEST WHAT TOWORK THEDATE THEHELP >

4.1.2 ℜ-rules structure

ℜ-rules are defined in a symbolic form and implemented as an XML file. The general structure of a ℜ-rule is:

```
<rule  id  = "ruleid"
       pat = "RegularExpression"
       if  = "condition"
       go  = "continuation" >
   <filter>[w1,w2, ..wn]</filter>
</rule>
```

Where:

- `id` is an attribute containing a unique rule identifier.
- `pat` is an attribute containing a pattern-matching expression based on the well-known 'RegExpr' of Java.
- `if` is an optional attribute containing a boolean condition. The `pat` is tried only if the condition evaluates to `True`.
- `go` is an optional attribute indicating a specific sequencing mode after the application of the rule, making it possible to overrule the general rule-applying algorithm given above.
- the `<filter>` tag contains the rewritten output given as a sequence of items w_i that can be either a) DIVA keys or b) matched parts of the input form, referred to by their position: 0 (the whole), 1 (the first)… following a traditional RegExpr convention.

Example of a syntactical rule catching a negative phrase:

```
<rule id="neg1"
  pat="&lt;(.*)( am | are | is | were )not (.*)&gt;"
  go="NEXTRULE">
  <filter>["NEG","BE",1,3]</filter>
</rule>
```

So that < you are not a fool > ⇒ < NEG BE you a fool >

Where NEG and BE are DIVA grammatical keys.

Example of a semantic rule catching various flexions associated with the concept ISSIMPLE:

```
<rule id="lem332"
   pat="&lt;(.*)(easy|straightforward|
        uncomplicated|trouble (?: )?free|
        undemanding|effortless) (.*)&gt;"
  go="NEXTRULE">
  <filter>[1,"ISSIMPLE",3]</filter>
</rule>
```

4.2 The interpretation phase

As shown in figure 2, the interpretation phase consists in applying sequentially a set of semantic spaces, each one being a set ℑ of interpretative rules. ℑ-rules are dedicated to the support of the multimodal response of the agent to the user's request.

4.2.1 ℑ-rules structure

An interpretative rule, or ℑ-rule, has the same general structure as a filtering ℜ-rule. The main difference is that interpretative rules support several tags, dedicated to the multimodal response:

<do> executes an action on the DOM structure of the page;
<say> makes the agent display a textual answer in its balloon;
<saylater> idem to <say> but the answer is delayed;
<hint> displays a help message in the chatbox bar;

and many others.

Within the <do> tag it is possible to execute actions upon the DOM-structure of the application; especially one can read and write internal state variables and put them into the symbolic model of the application as attributes of a special global object called THETOPIC.

Within the content of the other tags it is possible to refer to an attribute a_i of the symbolic model of the application in the form THETOPIC.a_i while using the meta-form _THETOPIC.a_i_. For example in a string, the value of the attribute will be inserted dynamically.

Example: the user gives his/her name to the agent

"My name is Jane" ⇒ < USERNAME BE jane >

Now we have the ℑ-rule:

```
<rule id="name2"  pat="&lt; USERNAME BE (\w+) &gt;" >
 <do>
    THETOPIC.x = TALK_capitalizefirst(TALK_getmatch(1));
    If (THETOPIC.x == THEUSER.name)
       TALK_say(['I knew it already','You said it'],0,2);
    else THEUSER.name = THETOPIC.x;
  </do>
 <say>
   <p>From now I will call you _THETOPIC.name_.</p>
   <p>Ok you namle is _THETOPIC.name_ ...</p>
   <p>Ok you are _THETOPIC.name_"</p>
   <p>OK pour @</p>
 </say>
</rule>
```

Where the <do> tag contains a JavaScript code doing:

```
If TOPIC.name ≠ null
then reply("name already known"); exit;
else TOPIC.name ← "Jane";
     reply("I will call you _THETOPIC.name_");
```

The <say> tag can use the meta variable _THETOPIC.name_ thus producing for example "From now I will call you Jane.".

4.2.2 The reusability of space files

ℑ-rules that share the same semantic domain can be grouped into so-called Semantic Spaces. These semantic domains can be either:
- *Generic spaces*: they can be reused from applications to applications, or
- *Specific spaces:* they are dedicated to a particular application.

It is possible to attach several generic spaces to an agent and one or more specific spaces. This provides a simple but efficient approach to the reusability of the linguistic resources.

4.2.3 The topic files

Another attempt to improve the genericity of the NLP-chain consists in the possibility to define a symbolic model of the application based on an attribute-value representation. Again using XML notation, we describe the main attributes of an application in a so-called *topic* file that can be accessed in read/write mode using the THETOPIC.a_i notation in action code (e.g. in <do> tags) and the _THETOPIC.a_i_ notation in strings.

Again, topic attributes can be generic (i.e. shared through applications by generic spaces) or specific to a particular application. Typical generic attributes are: `name, nickname, type, subtype, gender, age, size, creator, manager, possessor, moodlevel, cooperationlevel` etc. They are optional but should be filled for each DIVA-compliant application in order to take advantage of the generic spaces. Some of them are considered as static during a session (`gender, age`) and others have a value that can evolve within a dialogical session (`moodlevel, cooperationlevel`) because they are updated by the agent, so as to maintain a model of the relationship between the user and the agent for example.

Web-bases applications can be dedicated to sub domains because the web page is about a *person*, an *institution*, a *product* etc. We can define sub generic topic attributes and related semantic spaces. For example, here is a topic file using sub generic attributes for a person (Mao Xuetao — form filled by himself):

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xml>
<topic id="TOPICMAO">
  <personClass>ordinary</personClass>
  <personBriefIntro>Mao Xuetao is an ordinary
   person who comes from China.</personBriefIntro>
  <personGender>male</personGender>
  <personAge encoding="JS">31</personAge>
  <personHeight encoding="JS">179</personHeight>
  <personWeight encoding="JS">85</personWeight>
  <personFigure>handsome</personFigure>
  <personFeatures encoding="JS">
        [["SkinColor","yellow"],
         ["EyesColor","black"],
         ["HairColor","black"],
         ["HairStyle","short"]]
  </personFeature>
  <personFitness>normal</personFitness>
  <personIntelligence>normal</personIntelligence>
  <personBloodType>unknown</personBloodType>
  <personName>Mao Xuetao</personName>
  <personFamilyName>Mao</personFamilyName>
  <personNickName>Mao</personNickName>
</topic>
</xml>
```

By the separation of modal rules and topics, we can see the obvious difference from AIML which implements the input pattern and the output utterance in a single file as shown in the following example taken from the [8]:

```
<category>
   <pattern>ARE YOU * BED</pattern>
   <template>I like sleeping in bed.</template>
</category>
<category>
   <pattern>ARE YOU * PYRAMID</pattern>
   <template>My pyramid logo was designed by Sage Greco
and Darren Langley.</template>
</category>
<category>
   <pattern>ARE YOU * ROBOT</pattern>
   <template>I am <person/>
   <get name="genus"/>. Do you like my kind?</template>
</category>
```

This makes it difficult to provide some kind of reusability. Moreover one can see that the XML resource above is mainly meant to provide *evasive* answers about a keyword or an association of keywords (e.g. ARE_YOU+BED matches either “Are you in bed” “I say: Are you a bed”, “Are you involved in some factory crafting beds?” etc. and returns the same reaction).

5. CONCLUSION

Concerning the issue of evaluation, the first claim of DIVA is to provide an open, web-based framework for experimentations on Assisting Conversational Agents: the DIVA toolkit is now operational in version 1.0. It is downloadable from the DIVA home page [6] and can be freely used for research and teaching purposes.

The second claim of DIVA is to be a first attempt towards genericity of the NLP-chain, which should make it easy and fast to deploy. Currently, DIVA is already used as part of three research projects: France-Brazil COFECUB Pedagogical Rational/Affective Intelligent Agents [9]; The Digiteo Labs IROOM ambient project [10]; and the Gestural Agents action at LIMSI-CNRS for deaf people [11]. In these three collaborative actions, the toolkit has been proved easy to integrate (e.g. on an ambient middleware layer for the IROOM project or with the Adobe FLEX technology in the CODES music learning and composing web site of the PRAIA project).

In the current version of the NLP-chain presented here, we introduce the notion of intermediate Formal Request Form (FRF) which improves on traditional chatbot technology. However, the FRF remains a ‘flat’ sequence of semantic items that will show its limits when we will develop larger web-applications, entailing a broader semantic domain (even several sub domains). Therefore, the next version of the NLP-chain will have to rely on a more structured formal request language, currently under development.

6. REFERENCES

1. Oddcast. http://www.oddcast.com.
2. Hal - Zabaware. http://zabaware.com/.
3. Jabberwacky. http://www.jabberwacky.com/.
4. Alicebot. http://alicebot.blogspot.com/.
5. Elbot by Artificial Solutions. http://www.elbot.com/.
6. DIVA - DOM Integrated Virtual Agent. http://www.limsi.fr/~jps/online/diva/divahome/index.html.
7. Groupe de Travail sur les Agents Conversationnels Animés. http://www.limsi.fr/aca/.
8. AIML example. http://www.alicebot.org/aiml/aaa/Bot.aiml.
9. PRAIA. http://gia.inf.ufrgs.br/praia/index.php.
10. IRoom. http://iroom.supelec.fr/wiki/Accueil.
11. Gestural Agents. http://www.limsi.fr/Individu/jps/online/diva/geste/geste.main.htm.
12. Atkins, B.T. and Lewis, H.M.A. Language in Use. In *The Collins-Robert French-English Dictionary*. Harper Collins Publishers, 1996.
13. Bodenreider, O., Rindflesch, T.C., and Burgun, A. Unsupervised, corpus-based method for extending a biomedical terminology. *Proceedings of the ACL-02 workshop on Natural language processing in the biomedical*

domain - Volume 3, Association for Computational Linguistics (2002), 53-60.

14. Bouchet, F. and Sansonnet, J. Etude d'un corpus de requêtes en langue naturelle pour des agents assistants. *WACA 2006 : Actes du Second Workshop Francophone sur les Agents Conversationnels Animés*, (2006).
15. Bouchet, F. and Sansonnet, J. Caractérisation de Requêtes d'Assistance à partir de corpus. *Actes de MFI'07*, (2007).
16. Capobianco, A. Questioning the effectiveness of contextual online help: some alternative propositions. *Human-Computer Interaction INTERACT'03*, IOS Press (2003), 65-72.
17. Capobianco, A. and Carbonell, N. Contextual online help: elicitation of human experts' strategies. *Proceedings of HCI'01*, (2001), 824-828.
18. Capobianco, A. and Carbonell, N. Contextual Online Help: a Contribution to the Implementation of Universal Access. In S. Keates, P.J. Clarkson and P. Robinson, eds., *Universal Access and Assistive Technology*. Springer, London, 2002, 131-140.
19. Carbonell, N. Towards the design of usable multimodal interaction languages. *Universal Access in the Information Society 2*, 2 (2003), 143-159.
20. Cassell, J., Bickmore, T., Billinghurst, M., et al. Embodiment in conversational interfaces: Rea. *CHI ?99: Proceedings of the SIGCHI conf. on Human factors in comp. syst.*, ACM Press (1999), 520-527.
21. Cassell, J., Sullivan, J., Prevost, S., and Churchill, E., eds. *Embodied Conversational Agents*. MIT Press, 2000.
22. Chalmers, D.J., French, R.M., and Hofstadter, D.R. High-level perception, representation, and analogy: A critique of artificial intelligence methodology. *Journal of Experimental & Theoretical Artificial Intelligence 4*, 3 (1992), 185-211.
23. Chang, J., Tseng, T., Cheng, Y., et al. A corpus-based statistical approach to automatic book indexing. *Proceedings of the third conference on Applied natural language processing*, Association for Computational Linguistics (1992), 147-151.
24. Fellbaum, C. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.
25. Gale, W.A., Church, K.W., and Yarowsky, D. One sense per discourse. *Proceedings of the workshop on Speech and Natural Language*, Association for Computational Linguistics (1992), 233-237.
26. Hughes, L. The Eliza Effect: Conversational Agents and Cognition. 2006.
27. Jameson, A. Adaptive interfaces and agents. In *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*. Lawrence Erlbaum Associates, Inc., 2003, 305-330.
28. Leray, D. and Sansonnet, J. Assisting Dialogical Agents Modeled from Novice User's Perceptions. *Proceedings of KES'07*, (2007), 1122-1129.
29. Lester, J.C., Converse, S.A., Kahler, S.E., Barlow, S.T., Stone, B.A., and Bhogal, R.S. The persona effect: affective impact of animated pedagogical agents. *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM (1997), 359-366.
30. Molinsky, S.J. and Bliss, B. Inventory of functions and conversation strategies: The Comprehensive course in functional English. In Prentice Hall, 1994, 177-187.
31. Paroubek, P. Language Resources as by-Product of Evaluation: The MULTITAG Example. *LREC'2000: Proceedings of 2nd International Conference on Language Resources & Evaluation*, (2000), 151-154.
32. Rambow, O. Corpus-based methods in natural language generation: friend or foe? *Proceedings of the 8th European workshop on Natural Language Generation - Volume 8*, Association for Computational Linguistics (2001), 1-2.
33. Randall, N. and Pedersen, I. Who exactly is trying to help us? The ethos of help systems in popular computer applications. *Proceedings of the 16th annual international conference on Computer documentation*, ACM (1998), 63-69.
34. Rickel, J. and Johnson, W.L. Animated agents for procedural training in virtual reality: perception, cognition and motor control. *Applied Artificial Intelligence 13*, (1999), 343-382.
35. Shneiderman, B. *Designing the user interface (2nd ed.): strategies for effective human-computer interaction*. Addison-Wesley Longman Publishing Co., Inc., 1992.
36. Ummelen, N. and Neutelings, R. Measuring reading behavior in policy documents: a comparison of two instruments. *IEEE Transactions on Professional Communication 43*, 3 (2000), 292-301.
37. Vossen, P. WordNet, EuroWordNet and Global WordNet. *Revue française de linguistique appliquée 7*, 1 (2002), 27-38.
38. Weizenbaum, J. ELIZA: a computer program for the study of natural language communication between man and machine. *Commun. ACM 9*, 1 (1966), 36-45.