

# An ACA-based Semantic Space for Processing Domain Knowledge in the Assistance Context

**Mao Xuetao**

*Wuhan university of  
Technology, Hubei, China,  
xtmao@msn.com*

**François Bouchet**

*LIMSI-CNRS, Orsay  
Cedex, France,  
bouchet@limsi.fr*

**Jean-Paul Sansonnet**

*LIMSI-CNRS, Orsay  
Cedex, France,  
jps@limsi.fr*

## ABSTRACT

*For decades, people have been willing to interact with embodied conversational agents. This has driven researchers to consider more about social engineering than pure technical programming in building agent intelligence. As a typical application, assisting conversational agents aimed at helping people with attempting web-based applications have seen efficiency in many areas from e-learning to e-business and on-line games. The most outstanding feature of this kind of applications is that the assisting agent plays a role as a mediator conversing with users regarding the profile or topic of the to-be-assisted application. In this paper, we present a semantic space to model the knowledge in the assistance context considering both static and dynamic properties of the helping systems.*

Keywords: agent, ACA, ECA, CHS, semantic space

## 1 Introduction

### 1.1 Assisting conversational agent

The increasing amount of web-based applications emerging on the Internet today brings user convenience as well as challenges. A help system therefore is indispensable in most occasions. However, the traditional Contextual Help System (CHS) [1-2] have met difficulties for the growing large size and hard to use for ordinary users along with the increasing complexity of the applications. Because of the "persona effect"[3], Embodied Conversational Agent (ECA) such as Microsoft Agents™ in Microsoft Office Suite have been introduced in help systems. These agents are statically coded by software developers with 2D or 3D cartoon characters that have the following limitations:

- Inputs limited to clicks and menus
- No proper dialogue model
- No application model
- No synchronization of speech or movements

- Emotions limited to cartoon "emotes"
- No reusability for other applications.

These systems have been developed with no concern of the issues a) in Natural Language Understanding (NLU) or b) in symbolic reasoning over the current state of the application. The consequence is that the believability of these agents is very weak, especially for expert users: this phenomenon is known as the "Clippie effect"[4]. As a matter of fact, when users feel frustrated navigating in the CHS to find the effective links of help information, they may naturally think about posting a request i.e. help demand in natural language (speech/text) to the software components they feel confused about into the computer application, web service or other ambient appliances.

As a combination of the traditional CHS and ECA system, we have defined the concept of Assisting Conversational Agent (ACA) which combines advantages to support complex web applications. It can give answers to the requests about the components through a conversation with the user. In addition, ACA can help people in learning the domain knowledge contained in the applications through a kind of natural and easy way.

### 1.2 Assistance context

The goal of Semantic Web research is to transform the Web from a linked document repository into a distributed knowledge base and application platform[5]. However, the Web contains not only static documents but also dynamic activities. From a linguistic perspective, there are explicit and implicit natural language conversations happening between users and Web applications. Among others, the interactive assistance is one of the most happened matters on the Web. Because of the strictness of the particular sub-domain, the assistance context of web-based applications, we can manage to model the knowledge within the sub-domain formally in a small world called a *semantic space*.

In the assistance context, a typical assisted

application is composed of two main software parts:

- Application Model: this is the *domain-specific* part that contains 1) the actual application code (mainly JavaScript) and 2) the modeling files containing the description and help information about the application (mainly XML files).

- Assistant Agent Model: this is the *generic* part that contains the domain independent tools. 1) Natural language processing tools translating the textual requests and the Graphical User Interface (GUI) events into the Formal Request Language (FRL). 2) Rule-based symbolic processing tools providing a library of standard reactions to FRL requests while browsing the application model. Figure 1 shows the typical path of a user request:

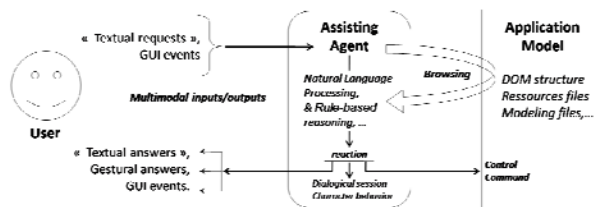


Figure 1: Structure of an Assisting Agent

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

The first step for the assistant dialogue system is to model the user's natural language requests and second the possible answers in the assistance context. Since the final purpose of a help system is to inform users with the knowledge of the application, the answers provide to the users must connect the certain

domain knowledge. In the following sections, we first present how the semantic spaces are used for processing natural language requests emerging in the assistance context, and in second time, we analyze the answers come from the knowledge of a special domain. We can prove that by this approach, domain knowledge can be delivered to the users interactively and satisfy their basic requirements.

## 2 Semantic Space

### 2.1 Assistance NLP-chain

As the formal expression of the knowledge in the assistance context, the semantic space models the meta-knowledge with three axes: the F (Formalization rule) axis, the I (Interpretation rule) axis and the T (TOPIC) axis (figure 2). So the meta-knowledge can be defined as triplet  $k(f,i,t)$ .

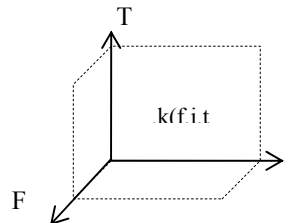


Figure 2: Semantic space

For example, the user asks "how old are you?" and the agent answers "25 years old I am." can be expressed as  $k(\text{ASKAGE}, \text{TELLAGE}, \text{AGEIS25})$ . This models an informative interaction between the user and the agent.

In the assistance context, each human-agent interaction process follows the NLP-chain (figure 3):

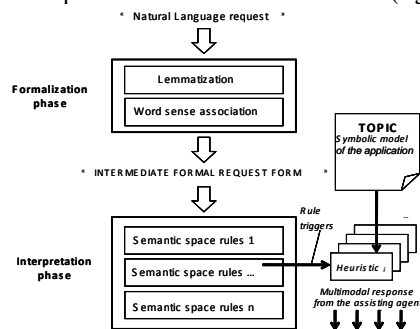


Figure 3: Structure of the NLP chain

1. Formalization phase will translate natural language queries within the assistance context into a formal request form (FRF). We also call this part the *semantic extractor*. It mainly includes defined formalization rules.

2. Interpretation phase will implement the reaction of the ACA and express the answer to the user according to some interpretation rules.

3. TOPIC models the *domain knowledge* to associate with the natural language generation rules in the interpretation phase.

## 2.2 Main entities

We have defined three main-level entities in an assistance context:

-The USER is an ordinary human in front of some computer displayed entities.

-The TOPIC is a symbolic expression of the domain knowledge about the web-based application.

-The ACA is a computer program designed to provide the answer, when the USER need help.

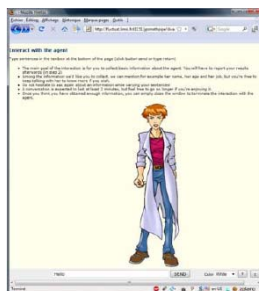
A conversational assistance always occurs around these three entities. Considering the personification of the ACA, we can define two main display cases regarding the way an ACA is associated with a TOPIC (Figure 3):

-Mediator agent: the ACA is displayed on top of the TOPIC it is assisting.

-Personified agent: only the TOPIC is displayed on screen, and the USER considers that the TOPIC and the ACA are the same entity.



(a) Mediator agent



(b) Personified agent

**Figure 3: The two main display cases**

In the first case displayed on figure 3, the domain knowledge is a famous game named Hanoi's towers. When the USER asks a question about the game, the

ACA displayed as a cartoon ghost will answer the USER with some information about the game. In the second case, the ACA is displayed as a picture of a woman who is also the domain knowledge. To know something about the woman, the USER directly asks her some questions. In both cases, a conversation always starts by the USER typing something in the inputbox.

## 2.3 Corpus collection

We inspect the real user's inputs in this context in past years. We had created experiments to investigate what really happens when a novice user is in front of the two kinds of display. As a result, a corpus of ~11000 requests gathered from three different sources (hundred of human subjects, two thesauri [Molinsky 1994; Atkins 1996] and some FAQ extracted from integrated help systems and websites concerning LaTeX and Microsoft Word) [7].

The use of those three complementary methods to build the Daft corpus allows us to have a rather representative corpus of assistance requests. Table 1 shows the translation (the original corpus is in french) of selected excerpts from the collected part of the Daft corpus, which reveals some of its characteristics:

**Table 1: Excerpts from the Daft corpus**

N°	Translation in English (including mistakes)
1	<b>clicks</b> the quit button
2	<b>clickon</b> the back button
3	ok, come back to <b>th ehomepage</b>
4	give me a map of the website
5	what is this window for,
6	<b>WDYM</b> by GT ACA
7	do the "close" button and the "quit" button work exactly the same way?
8	I have a question to ask to one of the members, how can I contact him?
9	when is the next meeting?
10	where <b>can-we</b> find the conference schedule?

- 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 [8], 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.

## 2.4 Semantic Extractor

In the formalization phase, we create a semantic extractor to extract the semantic from the raw inputs. The natural language requests (NLR) that frequently occurred in assistance context are translated into a formal request form (FRF). Several examples follow:

"Adopt a less provocative attitude, please."=>"TOTAKE QUANTONE LESSTHAN ISUNPLEASANT THEBELIEF TOSAYPLEASE" (6 KEYS)

"If I want to buy such a car, what can I do?"=>"QUEST IF THEUSER TOWANT TOOBTAIN such QUANTONE car WHAT TOCAN THEUSER TODO" (10 KEYS)

At first, all letters are translated into lowercase lemmas, some characters are replaced by white space and multiple white spaces are compressed. Then

through semantic extractor, most lemmas are translated into uppercase semantic keys except a few lemmas (e.g. 'such' and 'car' in the second example) corresponding to vocabulary specific to the assisted application and for which no semantic keys have been defined.

### 2.4.1 Semantic keys

The semantic keys or in short keys have been extracted from an analysis of the 11,000 corpus requests; they appear as uppercased English-like words. In the *semantic keys dictionary*, each key is associated with a short annotation that indicates its meaning. Table 2 shows some examples of semantic keys.

Table 2: examples of annotated semantic keys

keys	classes	Annotation
TOWORK	VERB	Denotes the general activity of achieving some work
TODERIVEFROM		Denotes the abstract action of inheriting/deriving its characteristics from something
ISHONEST	ADJECTIVE	Denotes the quality of somebody who is honest/sincere
ISFEMALE		Denotes the quality of a person with gender: female
THEAVATAR	NAME	Denotes the graphical/dialogical assisting character of the application
THEHELP		Denotes the service/help provided by somebody
WHAT	GRAMMATICAL	Denotes the grammatical WH-pronoun: what
WHY		Denotes the grammatical relation: why

For the sake of simplicity, in the first version of DIVA, a primary requirement was to restrict the number of semantic classes to less than 500 (e.g. EuroWordnet has more than 10,000 [10], but it covers the whole NL whereas it has been shown our assistance domain represents only 1% of it). By far, the total number of keys is 414, divided into five main classes: 128 NAMELIST, 23 CATEGORYLIST, 117 VERBLIST, 75 ADJECTIVELIST and 71 GRAMMATICAL & SPEECH ACT LIST.

### 2.4.2 F-rules

The extraction is performed by some F-rules (Formalization rules). An F-rule is mainly composed of an 'id', a 'pattern' and a 'filter part'. For example:

```
<rule id="463" pat="&lt;(.*) (how about)(?: \?)?
(.*)&gt;" go="NEXTRULE"><filter>[1,"QUEST
SOMETHING ABOUT",3]</filter></rule>
```

The 'filter part' contains the outputting FRF which is made up of a set of *semantic keys*. The extractor is making REGEXP patterns to match synsets and simple syntax of a natural language request and translate into semantic keys. The translation of NLR into FRF in the semantic space is a heuristic reasoning process. Usually a NLR will go through several F-rules in the semantic

extractor before being grounded on a FRF. For example:

```
NLR: How to move the red flag from the left to the right?
LEMMATIZATION < how to move the red flag from the left
to the right ? >
SPACE BEGIN -----
391 (<(.*) (come TODERIVEFROM|originate
TODERIVEFROM|TOBE TODERIVEFROM|hail
TODERIVEFROM|resided in|lived in|grew up in|derive
TODERIVEFROM|issue TODERIVEFROM|emanate
TODERIVEFROM|originate TODERIVEFROM|have roots
in|from) (.*)>) GO:NEXTRULE
FILTER < how to move the red flag TODERIVEFROM the left
to the right ? >
396 (<(.*)
(saunter|stroll|amble|march|stride|pace|hike|toddle|totter|stagger|
move|walk) (.*)>) GO:NEXTRULE
FILTER < how to AKOHUMANGESTURE the red flag
TODERIVEFROM the left to the right ? >
465 (<(.*) (how)(?: \?)? (.*)>) GO:NEXTRULE
FILTER < HOW to AKOHUMANGESTURE the red flag
TODERIVEFROM the left to the right ? >
695 (<(.*) ((?:the)?)ISPOSLEFT|left side|leftside|left) (.*)>)
GO:NEXTRULE
FILTER < QUEST HOW to AKOHUMANGESTURE the red
flag TODERIVEFROM the ISPOSLEFT to the right >
729 (<(.*)
(blanc|red|orange|yellow|green|blue|indigo|purple|violet|pink|bla
ck|white|gray|grey|gules|silver|golden|roseal|rosy|cyan|skyey)
(.*)>) GO:NEXTRULE
```

FILTER < QUEST HOW to AKOHUMANGESTURE the AKOCOLOR flag TODERIVEFROM the ISPOSLEFT to the ISPOSRIGHT >

The NLR "How to move the red flag from the left to the right?" was filtered in turn by 11 F-rules (the rules' ids are listed) but only a subpart of them is shown here. Each rule is an independent generic semantic extractor which translates some synonyms into a semantic key. The final FRF "QUEST HOW TO AKOHUMANGESTURE the AKOCOLOR flag TODERIVEFROM the ISPOSLEFT TO the ISPOSRIGHT" represents a less precise semantic of the NLR.

## 2.5 I-rules

In the interpretation phase, the formal request form (FRF) as the input contains the semantics of USER's intention to retrieve some information about the topic of the application. So the primary task of the reaction is to give the related information. The behaviors decide the multimodal reactions of the ACA and what information should be retrieved according to the FRF.

In the conversational modal, the main reactive behaviors are 'say', 'do', 'saylater', 'reply', etc. Most behaviors are dedicated to a generic assisting conversational domain symbolized in FRF, making them easier to share and reuse from an experiment to another. Each behavior contains a set of I-rules (Interpretation rules) that defines a reaction of the agent to the USER input. For example, assume that the user asks the age to the agent:

"How old are you?" → <QUEST HOW ISOLD TOBE THEAVATAR>

Now we have the following I-rule:

```
<rule id="age" pat="QUEST THEAGE|HOW ISOLD">
<do>THETOPIC.age.asked++;
If (THETOPIC.age.asked >= 1)
TALK_prepend(['As I said', 'I've told you, ']);
If (THETOPIC.gender = 'female')
TALK.say('It's not polite to ask this.');
<say>
<p>I'm _THETOPIC.age_ years old</p>
<p>I'm _THETOPIC.age_ ...</p>
<p>My age is _THETOPIC.age_</p>
</say>
</rule>
```

The possibility to add several lines into the <say> tag introduces variability as one of the options shall be chosen randomly. It can use the meta-variable \_THETOPIC.age\_ thus producing for example: "I'm 25 years old". The <do> tag can contain some JavaScript and thus allows easy scripting. In this example, we take into account past interactions through the simple use of an additional property ('asked') associated to each fact 'I'. We also take into account a

static fact: the gender of the agent. In this concept, more facts can be added if needed.

## 2.6 TOPIC

We can see that to build a reaction the agent requires some kind of *knowledge base* registering the relevant assistance information about the application, but also about the agent's and user's profiles (e.g. to store the agent's age in the above example).

The TOPIC is the symbolic knowledge base prepared for the assistant conversation. For example, here is an extract of the topic file of an agent embodied by a literary person, Dorian Gray, comes from a fiction named "The portrait of Dorian Gray":

```
<gender gloss="sex = {male female};">male</gender>
<nationality gloss="the name of the country related to the entity">U.S.A</nationality>
<ethnicgroup gloss="the ethnic group of the person">robotic</ethnicgroup>
<age gloss="a number that human used to measure the existent years of a topic">my portrait will age in place of me</age>
<history gloss="the genesis or the history or the CV of a topic">...you'd better see the novel.</history>
<future gloss="describe the future of a topic">to be a hero</future>
<mood gloss="the general mood of a person = tender, rude">good</mood>
```

The name of the "attributes" (e.g. gender, nationality, ethnicgroup, ...) is selected from the *TOPIC ontology* displayed in figure 4.

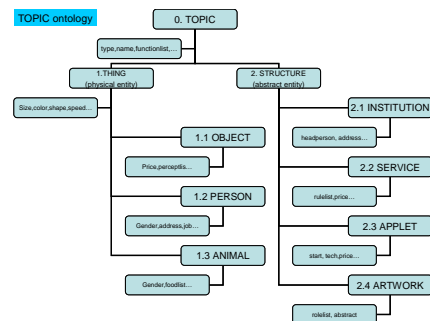


Figure 4: TOPIC ontology excerpt

Compare to the RDF (Resource Description Framework) <subject, predicate, object> triples, the TOPIC only records static subject and object because the predicate can be seen as always "assign". Unlike RDF, TOPIC is designed for human users to read and quickly create.

The ontology family tree is simple for its aim is to provide an easy guide to creating some TOPIC templates for ordinary users. Additionally, each of the attribute in the template is also attached a gloss for the users' convenience of designing a TOPIC instance. By this method, ordinary people can create TOPIC files

independently. All they need to do is to fill the value of the attributes according to the given template referencing the gloss. The ontology is not strict, thus the template is not fixed. Users can add, delete or edit the attributes list, but if so, it may cause mismatch by the NLP-chain. Therefore, relevant I-rules have to be added to the semantic space additionally. For example, suppose we add:

```
<glasses gloss="if the person wear glasses or not">yes, I wear glasses</glasses>
```

To a person's TOPIC template, we must also add an I-rule e.g.

```
<rule id="glasses" pat="do you wear glasses"><say><p>THETOPIC.glasses</p></say></rule>
```

to the semantic space so that the agent can generate a reply when asked "do you wear glasses". The gloss in the new added attribute is recommended because it is also helpful for the I-rule designer. This method brings convenience for functioning designer and the TOPIC designer to work collaboratively.

### 3 Conclusion

Novice users have faced difficulties using new developed Web-based applications as well as the traditional contextual help systems. We have utilized the assisting conversational agent (ACA) as a natural language interface to help novice users to retrieve useful information in a question-answer way with the agent. We argue that the knowledge in the assistance context can be described by a semantic space defined by axes of Formalization rules, Interpretation rules and TOPICs. The TOPIC models the static statements of the assistance context and the F-rules and I-rules models the dynamic request and answer behaviors. Different from the RDF defined in semantic web for machine accessing, the TOPIC ontology is for ordinary users to create static knowledge about the application. And both the F-rules and the I-rules are based on pattern matching to implement user-agent interaction. We have developed several experimental applications based on this approach. In future work, we will take into account a more generic symbolization of the knowledge in the assistance context, e.g. interpret the FRF into FRL.

### 4 References

- [1] Carenini G., Moore J. D., Generating explanations in context. Int. workshop on Intelligent User Interfaces, ACM Press pp 175-182, Orlando, 1993
- [2] Capobianco A., Questioning the effectiveness of

contextual online help: some alternative propositions HCI -INTERACT '03 M.Rauterberg et al. (Eds.), pp 65-72, IOS Press IFIP 2003

[3] James C. Lester, Sharolyn A. Converse, Susan E. Kahler, S. Todd Barlow, Brian A. Stone, Ravinder S. Bhogal, 1997, The persona effect: affective impact of animated pedagogical agents, Conference on Human Factors in Computing Systems archive, Proceedings of the SIGCHI conference on Human factors in computing systems table of contents, Atlanta, Georgia, United States, Pages: 359 – 366

[4] Ordinary User Oriented Model Construction for Assisting Conversational Agents, Leray, D.; Sansonnet, J.-P. Web Intelligence and Intelligent Agent Technology Workshops, 2006. WI-IAT 2006 Workshops. 2006 IEEE/WIC/ACM International Conference on Volume , Issue , 18-22 Dec. 2006 Page(s):355 – 358

[5] Ian Horrocks, 2007, Semantic web: the story so far, W4A '07: Proceedings of the 2007 international cross-disciplinary conference on Web accessibility, Banff, Canada, Pages: 120 – 125

[6] J-P. Sansonnet, D. Leray, KIWI: An environment for capturing the Perceptual Cues of an application for an Assisting Conversational Agent, AISB'07, Newcastle GB, 3-4 April 07

[7] M. Xuetao, J-P. Sansonnet, F. Bouchet, A corpus-based NLP-chain for a web-based Assisting Conversational Agent, Third Workshop on Animated Conversational Agents, WACA'2008, Paris, nov 2008

[8] A. Capobianco, 2003, "Questioning the effectiveness of contextual online help: some alternative propositions," Human-Computer Interaction INTERACT'03, pp. 65-72.

[9] F. Bouchet, J-P. Sansonnet, Caractérisation de Requêtes d'Assistance à partir de corpus, MFI'07, Paris, May 30th-June 1st 2007.

[10] P. Vossen, ed., EuroWordNet: a multilingual database with lexical semantic networks, Kluwer Academic Publishers, 1998.

[11] 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.