This document is published in:

Demazeau, Y. et al. (eds.) (2010) *Advances in Practical Applications of Agents and Multiagent Systems: 8th International Conference on Practical Applications of Agents and Multiagent Systems (PAAMS 2010)*, (Advances in Intelligent and Soft Computing, 70), Springer, 41-46.
DOI: http://dx.doi.org/10.1007/978-3-642-12384-9_6

An Architecture for the Design of Context-Aware Conversational Agents

David Griol, Nayat Sánchez-Pi, Javier Carbó, and José M. Molina

Group of Applied Artificial Intelligence (GIAA), Computer Science Department, Carlos III University of Madrid
e-mail:
{david.griol,nayat.sanchez,javier.carbo,josemanuel.molina}@uc3m.es

Abstract. In this paper, we present a architecture for the development of conversational agents that provide a personalized service to the user. The different agents included in our architecture facilitate an adapted service by taking into account context information and users specific requirements and preferences. This functionality is achieved by means of the introduction of a context manager and the definition of user profiles. We describe the main characteristics of our architecture and its application to develop and evaluate an information system for an academic domain.

1 Introduction

Ambient Intelligence systems usually consist of a set of interconnected computing and sensing devices which surround the user pervasively in his environment and are invisible to him, providing a service that is dynamically adapted to the interaction context, so that users can naturally interact with the system and thus perceive it as intelligent. To ensure such a natural and intelligent interaction, it is necessary to provide an effective, easy, save and transparent interaction between the user and the system. With this objective, as an attempt to enhance and ease human-to-computer interaction, in the last years there has been an increasing interest in simulating human-to-human communication, employing the so-called conversational agents [7]. Conversational agents have became a strong alternative to enhance computers with intelligent communicative capabilities with regard to the use of traditional interfaces, as speech is the most natural and flexible mean of communication among humans.

Funded by projects

CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02- 02/TEC, SINPROB, CAM MADRINET S-0505/TIC/0255 and DPS2008-07029-C02-02.

Adaptivity refers to several aspects in speech applications. In speech-based human-computer interaction users have diverse ways of communication. Novice users and experienced users may want the interface to behave completely differently, for example to have system-initiative instead of mixed-initiative. An example of the benefits of adaptivity in the interaction level can be found in [6]. The processing of context is essential in conversational agents to achieve this adapted behaviour and also cope with the ambiguities derived from the use of natural language [8].

In this paper we present an architecture for the design of conversational agents that provide personalized services by taking into account context information. In our architecture, agents handling different information sources and models interact to respond to the user's requests. Context information is used to provide a service that is then adapted to his location, geographical context, communication context, preferences, and needs. We provide a preliminary evaluation for the provisioning of personalized services to users in an academic information domain.

2 Our Architecture to Design Context-Awareness Conversational Agents

To successfully manage the interaction with the users, conversational agents usually carry out five main tasks: automatic speech recognition (ASR), natural language understanding (NLU), dialog management (DM), natural language generation (NLG) and text-to-speech synthesis (TTS). These tasks are usually implemented in different agents. Figure 1 shows a typical modular architecture of a conversational agent. Speech recognition is the process of obtaining the text string corresponding to an acoustic input. It is a very complex task as there is much variability in the input characteristics. Once the conversational agent has recognized what the user uttered, it is necessary to understand what he said. Natural language processing is the process of obtaining the semantic of a text string. It generally involves morphological, lexical, syntactical, semantic, discourse and pragmatical knowledge. The dialog management module updates the dialog context, provides a context for interpretations, coordinates the other modules and decides the information to convey and when to do it. Natural language generation is the process of obtaining texts in natural language from a non-linguistic representation. Finally, text-to-speech synthesizers transform the text into an acoustic signal.

As described in the introduction, context information is very valuable in order to enhance oral communication. For this reason, we have incorporated a context manager in the architecture of the designed conversational agents, as shown in Figure 1. This module deals with context information provided by the user and external positioning agents, and communicates this information to the different modules at the beginning of the interaction.

Kang et al [5] differentiate two types of context: internal and external. The former describes the user state (e.g. communication context and emotional state), whereas the latter refers to the environment state (e.g. location and temporal context). Most studies in the literature focus on the external context. In our case, context information that is managed by the context manager includes both internal (user's information and preferences) and external information (user's identification and location into the environment provided by a positioning agent).

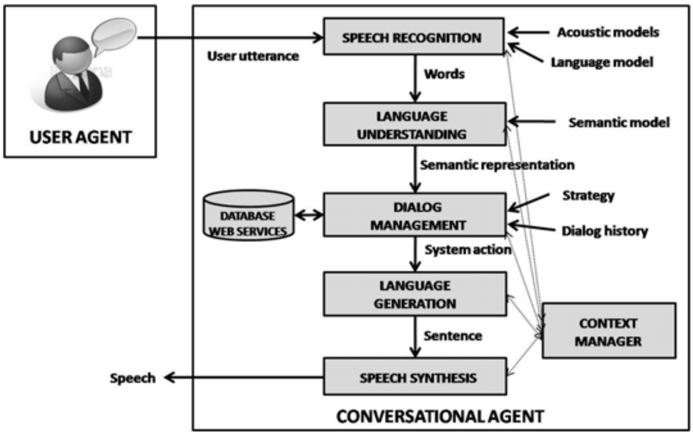

Fig. 1 Schema of the architecture of a conversational agent

Once the context information has been obtained, it must be internally represented within the conversational agent so that it can be handled in combination with the information about the user interaction and the environment. Different models have been proposed in the literature to define this representation [4, 1]. In our proposal, context information is represented by means of user profiles that are managed by the context manager. The information included in the user profiles can be classified into three different categories: general user information (user's name, gender, age, current language, skill level when interacting with dialog systems, possible pathologies or speech disorders, user's preferences detected during previous dialogs, etc.), general statistics (number of previous dialogs and dialog turns, their durations, the date of the last interaction with the system, etc.), and usage statistics and user privileges (counts of each action over the system that a user performs, and a mark of user clearance for each possible action).

3 Domain Application

We have applied our context aware architecture to design and evaluate an adaptive system that provides information in an academic domain [2]. This information can be classified in four main groups: subjects, professors, doctoral studies and registration. The system must have gathered some data by asking the user about the name of the subjects, degrees, groups names, professors, groups, semesters, name of a doctoral program, name of a course, or name of a registration deadline. The way in which the user is queried for this information follows in most cases a system-directed initiative. A set of four different scenarios has been used to evaluate our proposal for this task, taking into account the four different queries that a user can perform to the system. An example of a scenario to obtain the office location of a professor is as follows:

```
User name: Patricia López
Location: Campus University, Main Building,
Side A, First Floor
Date and Time: 2009-11-03, 9:00am
Device: PDAQ 00-1C-41-32-0A-59
Objective: To know the location of the office of the professor
David Smith.
```

As summarized in the description of our architecture, a positioning agent is used to determine user's position while he enters the Wifi network in the University campus. Then, the conversational agent that provides the academic information (and has been previously detected) asks the user agent about information for his identification to provide the personalized service. Once this identification is received by the context manager included in the conversational agent, it loads the specific context profile characteristics. This information is then consulted by the rest of the modules in the conversational agent to personalize the provided service. For this example of scenario, this profile contains the following information:

```
Name: Patricia López
-------------------
Gender: Female | Age: 21 | Language: Spanish | Skill level:
High | Pathologies: None...
-------------------
Preferences: Computer Science, Tutoring Information...
-------------------
Current_Location: Main Building, Side A, First Floor
```

To evaluate our architecture, we have acquired a set of dialogs for each one of the four scenarios defined for the task, whether considering the introduction of the context manager in our architecture or not. A dialog simulation technique has been used to acquire a total of 300 successful dialogs. In this technique we automatically acquire a dialog corpus by means of the introduction in our architecture of a dialog manager simulator and a user simulator [3]. Both modules use a random selection of one of the possible answers defined for the semantic of the task. A user request for closing the dialog is selected once the system has provided the information defined in the objective(s) of the dialog. The dialogs that fulfill this condition before a maximum number of turns (defined after a study of real human-human dialogs for each specific task) are considered successful. Instead of building the user profiles automatically during several dialog sessions, we build them manually in order to demonstrate the goodness of our context management approach.

We defined seven measures for the comparison of the dialogs acquired using or not context information: the percentage of successful dialogs, the average number of turns per dialog, the percentage of different dialogs, the number of repetitions of the most seen dialog, the number of turns of the most seen dialog, the number of turns of the shortest dialog, and the number of turns of the longest dialog. Using these measures, we tried to evaluate the success of our approach with regard to the incorporation of context information in the dialog manager as well as its efficiency with regard to the different objectives specified in the scenarios.

Figure 2 shows a dialog for this task acquired without incorporating the context manager, and a dialog for the same scenario acquired using our approach. It can be observed how the different system answers can be personalized by taking into account the information provided by the context manager. The results showed that the system shows a tendency of answering with higher agility and more natural than without including the context manager.

S1: Welcome to the university information system. How can I help you?	*No prompts personalization.*
U1: I want to know information about the tutoring sessions of Professor David Smith.	
S2: Can you tell me the name of the subject?	*There is not information about the specific subject.*
U2: Language Processors.	
S3: The tutoring sessions of Professor Smith for Language Processors are Monday from 14:00 to 16:00 and Wednesday from 11:00 to 14:00. Anything else?	
U3: Can you tell me the office number?	
S4: The office number is 2.1.B.06 in the Main Building. Anything else?	*No information about the current position of the user.*
U4: No, thank you.	
S5: Thank you for using the university information system. Have a nice day!	

S1: Good evening Patricia, do you want to know tutoring information?	*Prompt personalization including the user's name. The system knows the preference of the user about consulting tutoring sessions.*
U1: Yes, I want to know the tutoring sessions of Professor Smith.	
S2: The tutoring sessions of Professor Smith for Language Processors are Monday from 14:00 to 16:00 and Wednesday from 11:00 to 14:00. Anything else?	*There is information about the specific subject.*
U2: Can you tell me the office number?	
S3: The office number is 2.1.B.06. You just have to walk along the Side A corridor and turn left. The office is the second door on the right. Anything else?	*The system uses current location of the user to personalize the answer.*
U3: No, thank you.	
S4: Thank you Patricia. Have a nice day!	*Prompt personalization including the user's name.*

Fig. 2 An example of a dialog for the academic domain without adding the context manager (above) and including this module in our architecture (below)

Table 1 shows the results of the comparison of the different measures for the academic information system. The first advantage of our approach is the number of dialogs that was necessary to simulate in order to obtain the total of 300 dialogs for the task. While, only a 3.7% of successful dialogs is obtained without using context information, this percentage increases to 17.5% when the context manager is introduced. The second improvement is the reduction in the number of turns. This reduction can also be observed in the number of turns of the longest, shortest and most seen dialogs. For this reason, the number of different dialogs is also lower using the context information.

Table 1 Results of the high-level dialog features defined for the comparison of the two kinds of dialogs for the academic information system

	Without Context Infor.	Using Context Infor.
Percentage of successful dialogs	3.7%	17.5%
Average number of user turns per dialog	4.99	3.75
Percentage of different dialogs	85.71%	67.52%
Number of repetitions of the most seen dialog	5	16
Number of turns of the most seen dialog	2	2
Number of turns of the shortest dialog	2	2
Number of turns of the longest dialog	14	12

4 Conclusions

In this paper, we have presented a multiagent architecture to develop context aware conversational agents. This allow us to deal with the increasing complexity that the design of this kind of systems requires, adapting the services that are provided by taking into account context information and user requirements and preferences by means of the introduction of a context manager. The results of the application of our architecture to evaluate an academic information system show how the main characteristics of the dialogs can be improved by taking into account context information. As a future work, we want to evaluate our system with real users and also carry out a study of the user rejections of system-hypothesized actions. Finally, we also want to apply our technique to carry out more complex tasks.

References

1. Doulkeridis, C., Vazirgiannis, M.: CASD: Management of a context-aware service directory. Pervasive and mobile computing 4(5), 737–754 (2008)
2. Griol, D., Callejas, Z., López-Cózar, R.: Acquiring and Evaluating a Dialog Corpus through a Dialog Simulation Technique. In: Proc. of the 9th SIGdial Workshop on Discourse and Dialogue, London, UK, pp. 326–332 (2009)
3. Griol, D., Hurtado, L., Sanchis, E., Segarra, E.: Acquiring and Evaluating a Dialog Corpus through a Dialog Simulation Technique. In: Proc. of the 8th SIGdial Workshop on Discourse and Dialogue, Antwerp, Belgium, pp. 39–42 (2007)
4. Henricksen, K., Indulska, J.: Developing context-aware pervasive computing applications: models and approach. Pervasive and mobile computing 2, 37–64 (2006)
5. Kang, H., Suh, E., Yoo, K.: Packet-based context aware system to determine information system user's context. Expert systems with applications 35, 286–300 (2008)
6. Litman, D.J., Pan, S.: Designing and evaluating an adaptive spoken dialogue system. User Modeling and User-Adapted Interaction 12, 111–137 (2002)
7. López-Cózar, R., Araki, M.: Spoken, Multilingual and Multimodal Dialogue Systems: Development and Assessment. John Wiley Sons, Chichester (2005)
8. Truong, H.L., Dustdar, S.: A survey on context-aware web service systems. International Journal of Web Information Systems 5(1), 5–31 (2009)