A MULTILINGUAL DIALOGUE SYSTEM FOR ACCESSING THE WEB

Marta Gatius, Meritxell González and Elisabet Comelles
Technical University of Catalonia, Software Department, Campus Nord UPC, Jordi Girona, 1-3 08034 Barcelona, Spain
{gatius,mgonzalez, comelles}@lsi.upc.edu

Keywords: Web dialogue systems, portability, dialogue management, multilinguality.

Abstract: In this paper we propose the use of multilingual multichannel dialogue systems to improve the usability of web contents. In order to improve both the communication and the portability of those dialogue systems we propose the separation of the general components from the application-specific, language-specific and channel-specific aspects. This paper describes the multilingual dialogue system for accessing web contents we develop following this proposal. It is particularly focused two main components of the system: the dialogue manager and the natural language generator.

1 INTRODUCTION

The large amount of services and documents in different languages available in Internet has increased the need for multilingual and multichannel systems guiding the user when accessing the web contents. Web interfaces need to be more interactive and adaptable to different types of users, applications and channels.

The main advantage of language modes of interaction (text and speech) is that they are friendly and easy to use. Furthermore, spoken interfaces are necessary in applications where no other mode of communication is possible, such as applications for telephones and vehicles. Additionally, the voice mode improves web usability, especially for handicapped people, such as elderly adults, who face specific challenges when accessing the web.

The language modes become more useful as new uses of the web are invented and more new channels are available. For example, the spreading of personal digital assistants and other mobile communication devices results in an increasing prevalence of voice interfaces.

This article describes the multilingual dialogue system (henceforth, DS) we developed to enhance the usability and accessibility of online public contents in the context of the European project HOPS (http://www.hops-fp6.org). Section 2 gives an overview of the existing DSs accessing the web. Section 3 presents the multilingual web DS we have developed. Section 4, describes the dialogue manager (henceforth, DM) component of the system. Section 5 explains how the system messages needed for each new service are generated for each language. And finally, the last section draws some conclusions.

2 WEB DIALOGUE SYSTEMS

Even though the first language systems were developed during the seventies, its importance has increased throughout the last two decades. The main cause of this increasing importance was the technical improvements in speech recognition technologies.

2.1 The Evolution

DSs have evolved towards improvements both in the functionality and in the engineering features of the development process. Several lines have been followed in this evolution. One of the most significant of these consists in separating the application and the presentation components. Important improvements in the friendliness of the interaction have been achieved by expanding linguistic and conceptual coverage as well as by integrating different modes and languages. Examples of systems supporting multimodality and multilinguality are described in (Herzog et al, 2004) and (Gatius, 2001).

Speech recognition in open domain has still performance problems. For this reason, the speech mode is basically used in restricted domain systems. However, application-restricted DSs are expensive to develop and difficult to reuse. For this reason, an important aspect of the DS development is portability, the ability to facilitate the adaptation of the DSs to different applications, languages and channels. The most relevant of those works propose a separated representation of different types of knowledge involved in the dialogue: task knowledge, dialogue knowledge and modality and language-specific knowledge. The next subsections give several examples of these works.

2.2 Languages Based on XML

The definition of standard languages based on XML has favoured the development of DSs for accessing web content. Several of those languages are explicitly aimed to the development of internet-powered telephone applications. The use of standard XML-based mark-up language reduces the cost of training, facilitates technology integration and improves flexibility.

The most well-known of these languages is VoiceXML. One of the main advantages of VoiceXML is that it separates the logic of dialogues from the low level details of the voice components. Two complementary standard languages are used in VoiceXML systems: the Speech Recognition Grammar Specification (SRGS), and the Speech Synthesis Mark-up Language (SSML). The SRGS defines the standard formalism for the words and sentences which can be recognized by a VoiceXML application. The SSML is the new standard way of producing content to be spoken by a speech synthesis system.

The dialogue management model in VoiceXML presents several advantages to the finite state model used in previous commercial systems. It facilitates the description of the slots representing the various kinds of information the user would be asked to fill. However, VoiceXML presents also limitations to support complex telephonic calls and other modes than voice (and touch-tone DTMF). In order to solve those limitations, other languages were developed.

The CCXML and the CallXML languages were developed to deal with control management in complex telephonic calls not supported by VoiceXML systems (calls including multiconference, transfers, etc.).

Standard languages to support multichannel and multimodal communication have also been defined. Examples of these languages are the Speech Application Language Tags (SALT), the Extensible MultiModal Annotation (EMMA) language and the Synchronized Multimedia Integration Language (SMIL). Standard architectures facilitating the development of web DSs have also been designed, such as the MultiModal Architecture.

The standard languages mentioned favour the development and portability of DSs. However, the dialogues and language resources (grammars and system's messages) have to be defined for each new service. A step can be taken towards portability of DSs by isolating the application task knowledge and the dialogue strategies. In this line, there have been several proposals, such as the GEMINI platform (Hamerich et al, 2004) and the MIML language (Araki and Tachibana, 2006).

2.3 Web Interaction Management

Web interaction management is another related area of research focused on improving web usability and accessibility. There are different works on facilitating the web access through different modes, such as that for adapting web contents to different impairments (Richards and Hanson, 2004), the transformation rules for creating mixed-initiative dialogues (Narayan et al, 2004) and the framework for incorporating multimodal interfaces to already existing web applications (Ito, 2005).

The main difference between the works on DSs and those in the web interaction management area is that DSs are more concerned with achieving a friendly and robust communication in different languages, adapted to the user's needs, whereas the interaction management research is typically concerned with building simple dialogues automatically from web pages.

2.4 Large-Scale Dialogue Systems

There is also a lot of interest in the development of large-scale DSs supporting rich interactions to different applications in several languages and channels. Examples of those systems are TRIPS (Allen et al, 2001), STAPLE (Kumar, Cohen and Huber, 2002) and COLLLAGEN (Rich, Sidner and Lesh, 2001). Those systems are rather complex. They require a flexible and modular architecture model and appropriate software for the integration of modules.

As mentioned in (Herzog et al, 2004), for such large DSs a distributed organization constitutes the natural choice to develop a flexible and scalable software architecture, able to integrate heterogeneous modules. Different approaches have been followed when developing a distributed DS.

The Galaxy Communicator Software Infrastructure (Seneff, Lau and Polifroni, 1999) is a distributed, message-based architecture for developing spoken DSs. A central hub mediates the interaction among the dialog components (although peer-to-peer connections are also supported).

The Open Agent Architecture (Martin, Cheyer and Moran, 1999) is a framework supporting multimodal interaction. It integrates several heterogeneous agents controlled by a central unit which communicate between themselves via messages.

The MULTIPLATFORM (Herzog et al, 2004) is based on a distributed architecture which employs asynchronous message-passing to connect modules and does not rely on centralized control. This platform has been used to develop several DSs.

3 OVERVIEW OF THE SYSTEM

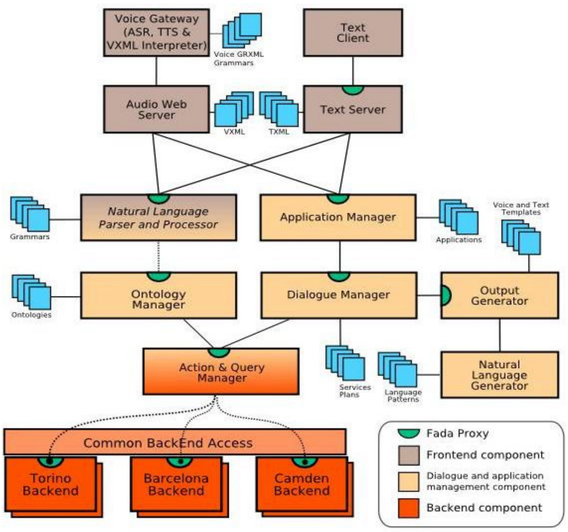

Figure 1: The architecture of the dialogue system.

The system we developed for guiding the user to access the web supports text and speech (through the telephone) in several languages: English, Spanish, Catalan and Italian. The current implementation of the DS has been adapted to two different types of services: a transactional service for large objects collection and an informational service giving information about the cultural events. The system has been deployed in the public administration of three different cities.

The architecture of the system is shown in Figure 1. One of the goals of the system design has been to facilitate the incorporation of new services, languages and channels. For this purpose, the channel-specific, language-specific and service-specific aspects of the system have been separated from the general components.

3.1 The Components of the System

As many existing web DSs our system is composed of three layers: Presentation (or front-end), Dialogue Management and Data. As can be seen in Figure 1 the components of the presentation layer are the following: The Voice Gateway and the Audio Web Server for speech mode, the Text Client and the Text Server, for text mode and the Natural Language Parser and Processor (NLPP) performing a deep syntactic and semantic analysis, used for both, voice and textual interaction.

The two voice components had been adapted from those in the Loquendo platform (http://www.loquendocafe.com). VoiceXML is used to define the voice interaction with the user. The automatic speech recognizer uses grammars following the SRGS standard formalism to model the user's interventions. Once the sequence of words have been recognized they are passed to the NLPP. The speech synthesis system transforms the content of the system's answer into voice following the standard SSML.

The components of the dialogue management layer control the interaction with the user. These components are the Application Manager, responsible for the session management and the DM determining the dialogue flow. The DM includes two submodules: the Output Generator and the Natural Language Generator.

The data layer components are the Action & Query Manager, accessing the back-end, and the Ontology Manager using the ontologies which model the domain-specific knowledge. These ontologies are used by the NLPP to perform the semantic analysis. The ontologies are represented in the standard language OWL.

In our system the different components are integrated following a service-oriented architecture (SOA). Although asynchronous message-passing is more flexible than component-specific remote APIs (used in SOA) we found that synchronous communication between components was more appropriate for efficiency reasons. We use FADA technology (Federated Advanced Directory Architecture, http://sourceforge.net/projects/fada) to support the SOA principles. It is an implementation of the Jini Network Technology for working in WAN environments.

4 DIALOGUE MANAGEMENT

The dialogue management is concerned mainly with conversational control and guidance: who can speak at each state of the communication and what they can speak about. A proper design of the DM component reduces the cost of adapting the system to a new service. Dialogue management models have evolved from simple finite state automata, representing all possible interactions, to more complex models using plans to recognize the user's intention.

4.1 System-Driven Communication

The main goal in commercial DSs is robustness. For this reason, most of the DMs only support system-driven dialogues asking the user the information needed to accomplish the task. Following this approach, for the first prototype of the system we developed a DM supporting only system-driven dialogues. In the resulting system, once the user selects a specific service task, the system asks the information about the corresponding input parameters and finally, gives the results.

The DM we developed for the first prototype followed a structural dialogue state, as most of the DMs in commercial systems. For each service, a specific dialogue flow was defined in which there was an explicit description of the relation between states and actions.

The resulting application-driven dialogues have proved efficient for transactional web services but not for the informational web service. When seeking for specific information, the user usually can give different types of information, thus restricting the system search. Most existing DSs guide the user to introduce this information by asking very specific questions, that, in many cases, may look unnatural.

See for example, the web service on cultural events shown in Figure 2. When accessing the service the user can search an event giving different types of information: The particular event name, the event type, the location, the data or a combination of any of the above type of information.

In the first prototype, the DM tried to restrict the search on cultural events by asking a sentence such as: "*If you are looking for events, please say LIST OF EVENTS. If you know the event, and you are looking for information about it, please say SINGLE EVENT.*" In most cases, the users did not answer the question but instead they asked for specific information, such as "*I am looking for classical music concerts*".

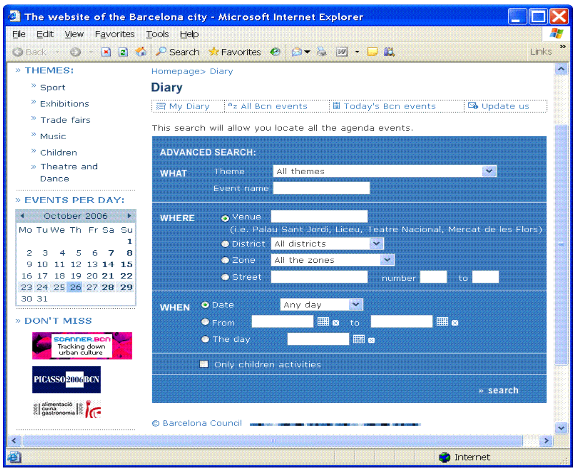

Figure 2: The web service on cultural events.

When seeking information, users may not have a well defined goal, and for this reason user's interventions can switch from one aspect of the topic to another. The specification of such types of interactions through a finite-state controller is difficult and can not be changed easily.

In order to improve both the communication and the engineering process of adapting the system to a new service, we decided to use an explicit dialogue model, defining general strategies to decide the next action. In particular, we followed the Information State (IS) approach, explained in the next section.

4.2 The Information State Approach

The main reason for following the IS approach is that it covers dialogue phenomena useful in practical dialogues, such as feedback strategies related to grounding and addressing not raised issues. The IS theories support mixed and user-initiative dialogues, including confirmation and clarification dialogues, which can be initiated by the system or the user. Besides, those theories have been successfully applied in many projects and applications.

The main difference between the IS approach and the structural dialogue state approaches is that IS theories are based in a much richer representation of the dialogue context, including more mentalistic notions such as beliefs, intentions and plans. As defined in (Traum et al, 1999) an IS theory consists of the following components: The description of the information state (e.g., participants, beliefs, common ground, intentions, etc.), a set of dialogue moves, a set of rules updating the information state and a control strategy to select next rule.

We have implemented these components following the issue-based approach described in (Larsson, 2002). In this approach, system actions were defined in plans which contain information about the action the system should perform to achieve a specific goal.

4.3 Plans

In plan-based dialogue management, plans are usually generated dynamically during the communication using AI plan recognition techniques to recognize the user's intention and templates to performe actions. When knowing the goal, the system can optimize the plan to achieve it, considering the dialogue history. Although this technique is powerful, it is not the most efficient for all types of dialogues. For simple dialogues on web services, such as those supported by our system, plans can be defined and stored in a library when a new service is incorporated, they do not need to be generated dynamically. The DM we develop does not need general reasoning capabilities for planning and plan recognition because it is designed for guiding the user to access the web services, which usually are not very complex.

Each task a service can perform is considered a possible user goal. For this reason, for each task service we define a communication plan that has to be followed in order to perform the task. A task plan can be descomposed into actions and subtasks. Preconditions governing in which context a particular action must be done are also included. Three types of actions can appear in the plans used by our system: *Ask*, the system asks the user the information needed, *Answer,* the system gives information to the user and *Back-end access*.

Using these plans the system can address issues introduced by the user which had not been previously raised. For instance, if the user initially says *"I want information about classical music concerts"* the system can search for a plan in which *classical music concerts* could be the answer to an *Ask* action. Then, it would continue executing the other actions in the plan.

In order to facilitate the generation of plans for a new service, we have defined general templates for the two types of web services the system has to support: informational and transactional services. In transactional services the system has to ask the user all the information corresponding to the mandatory input parameters that have not been previously given. In informational services the system has to ask the user information to constrain the search.

5 THE SYSTEM MESSAGES

Generating the most appropriate system messages for each service is time consuming, especially in systems supporting several languages. This cost can be reduced when representing appropriately the relations between the different types of knowledge involved in communication: dialogue knowledge, conceptual knowledge and linguistic knowledge.

One of the most relevant works in this area is the Generalized Upper Model (Bateman, Magnini and Rinaldi, 1994), a general syntactico-semantic model used to generate text in several languages for different types of applications.

Our work differs from previous works in that area because it is focused in dialogues guiding the user to access transactional and informational web services. By representing appropriately the linguistic structures involved in the communication in these web services we can limit the language that has to be generated at run-time. Several of these linguistic structures can be reused across services. Others will have to be generated when incorporating a new service to the system.

Several system messages appear in all services. Most of them correspond to the dialogue acts common to all services, such as, formal opening, closing and thanking. There are some system messages which appear only in a specific type of service, either transactional or informational, but can also be reused through them. For example, in information services, several sentences presenting the results to the users can be needed. Some of them are independent of the particular service and can be applied to all informational services.

The system messages related to the parameters appearing in more than one service (such as name, address, telephone, date and price) are also reused. However, there are also many system messages that differ for each particular service. Basically, these messages are related to the dialogue acts the system can perform: Asking the user for specific data (the task to perform, the information to search and the value of the input task parameters), asking for confirmation and presenting the results.

We have defined general patterns to represent the several forms in which these dialogue acts can be expressed. These patterns are adapted to each specific service. In order to perform automatically this process we have adapted the syntactico-semantic taxonomy described in (Gatius, 2001). This taxonomy is reused in the four languages we are working with. For each language, each taxonomy class has been associated with the linguistic structures involved in the following dialogue acts: asking the parameter's value, giving its value, checking its value and confirming it.

When adapting the system to a new service, the task parameters have to be classified according to the syntactico-semantic taxonomy and linked to the corresponding lexical entries (in each language). Then, the system messages can be automatically generated in each language by adapting the general patterns associated with each taxonomy class to the particular lexical entries.

6 CONCLUSIONS

In this paper we propose the use of multilingual multichannel DSs to improve the usability of web contents. In order to improve the portability of those DSs we propose the separation of the general components from the application-specific, language-specific and channel-specific aspects.

Following this proposal we have developed a DS for accessing web contents in four languages: English, Spanish, Catalan and Italian. The core of this system is the DM component which controls the dialogue flow. The DM supports user-initiative dialogues, including confirmation and clarification dialogues. It uses plans describing the actions to follow in order to perform the specific service tasks. Informal tests have shown that this DM supports a friendlier communication than the one used in a previous prototype. Currently, a more formal evaluation of the whole system is being performed.

The process of incorporating a new service to the system implies generating the specific service plans as well as adapting the linguistic resources. In order to facilitate the task of generating plans for new services, we have defined general templates for two types of web services: informational and transactional. We have also defined general patterns to automatically generate the system messages in the four languages for new web services.

REFERENCES

Allen, J., Byron, D., Dzikovska, M., Ferguson, G., Galescu, L . and Stent, A. (2001) 'Toward Conversational Human-Computer Interaction', *AI Magazine,* vol. 22, no. 4, Winter, pp. 27-38.

Araki, M. and Tachibana, K. (2006) 'Multimodal Dialog Description Language for Rapid System Development', *Workshop Proceedings*, the 7th SIGdial Workshop on Discourse and Dialogue, Sidney, Australia, pp 109-116.

Bateman, J.A., Magnini, B. and Rinaldi, F. (1994) 'The Generalized {Italian, German, English} Upper Model', *Workshop Proceedings*, ECAI-94 Workshop on Implemented Ontologies, Amsterdam.

Gatius, M. (2001) 'Using an ontology for guiding natural language interaction with knowledge based systems'. Ph.D. thesis, Technical University of Catalonia. Available, http://www.lsi.upc.es/~gatius/tesis.html.

Hamerich, W., Schubert, V., Schless, V., Córdoba, R., Pardo, J., D'Haro, L., Kladis, B., Kocsis, O. and Igel, S. (2004) 'Semi-Automatic Generation of Dialogue Applications in the GEMINI Project', *Workshop Proceedings*, SIGdial Workshop on Discourse and Dialogue, Cambridge, USA, pp. 31-34.

Herzog, G., Ndiaye, A., Merten, S., Kirchmann, H., Becker, T., Poller, P. (2004) 'Large-scale software integration for spoken language and multimodal dialog systems', *Journal of Natural Language Engineering*, vol. 10, no. 3-4, September, pp. 283-305.

Ito, K. (2005) 'Introducing Multimodal Character Agents into Existing Web applications', *Conference Proceedings*, the World Wide Web Conference, Chiba, Japan, pp. 966-967.

Kumar, S., Cohen, P.R. and Huber, M.J. (2002) 'Direct Execution of Team Specification in STAPLE, *Conference Proceedings*, AAMAS Conference, Bologna, Italy, pp. 567- 568.

Larsson, S. (2002) 'Issue-based Dialogue Management'. PhD Thesis, Goteborg University.

Martin, D. L., Cheyer, A. J. and Moran, D. B. (1999) 'The Open Agent Architecture: A Framework for Building Distributed Software Systems', *Applied Artificial Intelligence*, vol. 13, pp. 91-128.

Narayan, M., Williams, C. Perugini, S. and Ramakrishnan, N. (2004) 'Staging Transformations for Multimodal Web interaction Management', *Conference Proceedings*, World Wide Web Conference, New York, USA, pp. 212-223.

Seneff, S., Lau, R. and Polifroni, J. (1999), 'Organization, communication and control in the Galaxy-II conversational system', *Conference Proceedings*, Eurospeech, Budapest, Hungary, pp. 1271-1274.

Rich, Ch., Sidner, C. and Lesh, N. (2001) 'COLLAGEN: Applying Collaborative Discourse Theory to Human-Computer Interaction', *AI Magazine*,vol. 22,n 4,15-25.

Richards, J. and Hanson, V. (2004), 'Web accessibility: A broader view' *Conference Proceedings*, World Wide Web Conference, New York, USA, pp. 72-79.

Traum, D., Bos, J., Cooper, R., Larson S., Lewin, I., Mathesson, C. and Poesio, M. (1999), 'A model of Dialogue Moves and Information State Revision'. Trindi Technical Report D2.1, [Online], Available, http://www.ling.gu.se/projekt/trindi//publications.htm