This document is published in:

Corchado, E. et al. (eds.), 2012. *Hybrid Artificial Intelligent Systems*: 7th International Conference, HAIS 2012, Salamanca, Spain, March 28-30, 2012. Proceedings, Part I (Lecture Notes in Computer Science, 7208), Springer, pp. 49-60.

DOI: 10.1007/978-3-642-28942-2_5

Modeling Internet as a User-Adapted Speech Service

David Griol, Javier Carbó, and José Manuel Molina

Applied Artificial Intelligence Group
Computer Science Department
Universidad Carlos III de Madrid
28911 - Leganés, Spain
{david.griol,javier.carbo,josemanuel.molina}@uc3m.es

Abstract. The web has become the largest repository of multimedia information and its convergence with telecommunications is now bringing the benefits of web technology and hybrid artificial intelligence systems to hand-held devices. However, maximizing accessibility is not always the main objective in the design of web applications, specially if it is concerned with facilitating access for disabled people. This way, natural spoken conversation and multimodal conversational agents have been proposed as a solution to facilitate a more natural interaction with these kind of devices. In this paper, we describe a proposal to provide spoken access to Internet information that is valid not only to generate basic applications (e.g., web search engines), but also to develop dialog-based speech interfaces that facilitate a user-adapted access that enhances web services. We describe our proposal and detail several applications developed to provide evidences about the benefits of introducing speech to make the enormous web content accessible to all mobile phone users.

Keywords: Conversational Agents, Multimodality, Internet Modeling, VoiceXML, XHTML+Voice, Speech Interaction, Neural Networks.

1 Introduction

Continuous advances in the development of information technologies and the miniaturization of devices have made it possible to access information, web services, and artificial intelligence systems from anywhere, at anytime and almost instantaneously through wireless connections [2]. Although devices such as PDAs and smartphones are widely used today to access the web, contents are usually accessible only through web browsers, which are operated by means of traditional graphical user interfaces (GUIs). The reduced size of the screen and keyboards makes the use of these devices very difficult and also avoids the use of these applications by motor-handicapped and visually impaired users. The major drawback of the existing web infrastructure is that, the present web content was originally designed for traditional desktop browsers. This way, although mobile phones are designed to provide anytime and anywhere access to users, the challenge that is

presented to the present Internet world is to make the enormous web content accessible to all mobile phone users and by means of a more natural communication with the user.

Multimodal interfaces go a step beyond GUIs by adding the possibility to communicate with these devices through other interaction modes such as speech. Multimodal conversational agents [8,7] can be defined as computer programs designed to emulate communication capabilities of a human being including several communication modalities. The use of these agents provides three main benefits. Firstly, they facilitate a more natural human-computer interaction, as it is carried out by means of a conversation in natural language. Secondly, multimodal interfaces make possible the use of these applications in environments in which the use of GUI interfaces is not effective, for example, in-car environments. Finally, these systems provides the objective of facilitating the access to the web for people with visual or motor disabilities, allowing their integration and the elimination of barriers to Internet access [1].

Most of multimodal conversational agents to access web contents and services applications are currently developed using the VoiceXML language[1], given that it has been defined as the World Wide Web Consortium (W3C) standard to access Internet contents by means of speech. VoiceXML audio dialogs feature synthesized speech, digitized audio, recognition of spoken and DTMF key input (Dual-tone multi-frequency signaling), recording of spoken input, telephony, and mixed initiative conversations. The standard also enables the integration of voice services with data services using the client-server paradigm. Therefore, the VoiceXML standard facilitates the access to the net in new devices and environments by providing all these functionalities in combination with well-defined semantics (thus making XML documents universally accessible).

However, this programming language only allows the definition of a dialog strategy based on scripted Finite State Machines. This way, VoiceXML systems usually emphasize on the search of web documents in specific tasks and not on the interaction with the user. With the aim of creating dynamic and adapted dialogs, as an alternative of the previously described rule-based approaches, the application of soft computing models and statistical approaches to dialog management makes it possible to consider a wider space of dialog strategies [11,5]. The main reason is that these models can be trained from real dialogs, modeling the variability in user behaviors.

In this paper we describe a proposal to model the web as an speech-based service by means of the combination of the VoiceXML standard and statistical methodologies for dialog management. This makes possible to generate not only general-purpose applications (e.g., speech-based access to web search engines or extended use applications like the Wikipedia), but also to develop enhanced speech-based interfaces that provide personalized access to web services in which dialog is required to iteratively exchange information and achieve the objectives (e.g., ask the user about different information items in order to provide them specific information related to travel planning, hotel booking, etc).

[1] http://www.w3.org/TR/voicexml20/

For the former applications we propose the use of the XHTML+Voice language[2] in combination with a specific strategy to dynamically create the different grammars in the application. This language combines the visual modality offered by the XHTML language and the functionalities offered by the VoiceXML language for the interaction by means of speech. For the latter applications we propose the use of a statistical dialog management methodology in combination with a user simulation technique. This combination makes possible to automatically learn a statistical dialog model to select systems responses adapted to each user and also include grammars which facilitate the interaction in natural language.

We have applied our proposal to develop several conversational agents which interact with different web-based applications, providing a sophisticated interface which merges voice with traditional GUIs. All these applications are easily interoperable so that they are very useful to evaluate the potential of voice interaction in general and specific domains, through a variety of resources and with different users. This way, one of the main objectives of our work is to adequately convey to users the logical structure and semantics of content in web documents.

The remainder of the paper is as follows. Section 2 describes our proposal to include speech to facilitate the information and services on the Internet. Section 3 describes the application of our proposal to develop several multimodal conversational agents. This section also summarizes the results of a preliminary evaluation of these agents. Finally, Section 4 provides some conclusions and future research work.

2 Our Proposal to Provide a Speech-Based Access to the Web

HTML is the language popularly used for marking up information on the web so that it can be displayed using commercially available browsers. However, the eXtensible Markup Language (XML) was developed as a solution to correct the limitation of the use of information that is available on the web by marking-up information in the web pages and also allowing developers to define their own tags with well-defined meanings that others can understand. The use of an XML-based language significantly improves the portability and interoperability of the programmable parts (including data and programs) of the service. One of the main objectives of XML and ontology-based languages is to adequately convey to users the logical structure and semantics of content in web documents.

Several proposals have been developed to translate HTML pages to speech [3]. These systems captured the text present in the web page and employed speech synthesis technologies to speak this information to the user, also introducing different sounds and tones to visually impaired can make-out the structure of the document. However, these systems capture only specific parts of the HTML code and fails to address the interactive features provided by the

[2] http://www.w3.org/TR/xhtml+voice/

HTML pages. Although additional proposals have been developed to translate from HTML or XML web pages to VoiceXML [10,4], they require a previous preprocessing of the code by means of the user or they only handle a subset of HTML tags. In addition, the XML language by definition and the extended use of Cascading Style Sheets (CSS) to describe the presentation semantics make a general definition of these translators almost impossible. For these reasons, we propose a specific translation from HTML pages to XHTML+Voice only to develop general-purpose multimodal applications (Section 2.1) and the use of a user-adapted statistical dialog management methodology to develop dialog-based applications which are focused on interactive dialog with users (Section 2.2). A number of currently extended speech-based applications with a general purpose (like the API proposal from google) are based on additional languages and technologies mainly focused on the development of general-purpose systems.

2.1 Developing General-Purpose Web Applications

As stated in the introduction, our proposal to generate speech-based interfaces for general-purpose web applications is based on the use of the XHTML+Voice language, thus combining speech access with visual interaction. Figure 1 shows the translation between a HTML document and its equivalent XHTML+Voice file. As it can be observed, the development of oral interfaces implemented by means of XHTML+Voice implies the definition of grammars, which delimit the speech communication with the system. The <*grammar*> element is used to provide a speech or DTMF grammar that specifies a set of utterances that a user may speak to perform an action or supply information, and for a matching utterance, returns a corresponding semantic interpretation.

We have defined a specific strategy to cover the widest range of search criteria by means of the definition of speech recognition grammars in the different applications. This strategy is based on different aspects such as the dynamic generation of the grammars built from the results generated by the interaction with a specific application (e.g., to include the results of the search of a topic using a speech-based search engine), the definition of grammars that includes complete sentences to support the naturalness of the interaction with the system (e.g., to facilitate a more natural communication and cover more functionalities), and the use of the ICAO phonetic alphabet[3] in the cases in which spelling of the words is required in order not to restrict the contents of the search or in situations in which repetitive recognition errors are detected (e.g., in order not to delimit the topics to search using a search engine).

2.2 From General-Purpose to More Natural Mixed-Initiative Dialogs

When designing dialog-based conversational agents, developers need to specify the actions a system should take in response to user speech input and the state

[3] International Civil Aviation Organization (ICAO) phonetic alphabet: http://www.icao.int/icao/en/trivia/alphabet.htm

```
% HTML document
<html>
<head>
<title>VoiceApp-Voice Browser</title>
</head>
<body>

<li>LINK 1:
<a href="http://www.beatles.com/">
<b>The Beatles</b> Find out all about
The Beatles...</li>

...

<li> LINK 10:
<a href="http://www.rarebeatles.com/">
<b>Songs, Pictures, and Stories of
The Beatles</b>
Beatles website for collectors
and fans ...</li>

</body>
</html>
```

```
% XHTML+Voice file
<?xml version="1.0" encoding="ISO-8859-1"?>
<html xmlns="http://www.w3.org/1999/xhtml"
 xmlns:vxml="http://www.w3.org/2001/vxml"
 xmlns:ev="http://www.w3.org/2001/xml-events"
 xmlns:xv="http://www.voicexml.org/2002/xhtml+voice">

<head>
<title>VoiceApp - Voice Browser</title>
<vxml:form id="nav">
    <vxml:block>
    To visit the links, you have to say
    "LINK" and thecorresponding number.
    </vxml:block>
    <vxml:field xv:id="app" name="app">
      <vxml:grammar src="inig.jsgf"/>
      <vxml:nomatch>
        <vxml:prompt>
        Please repeat again, I can not understand you.
        </vxml:prompt>
      </vxml:nomatch>
    </vxml:field>
    <vxml:filled mode="all">
      <vxml:prompt> Ok got them. </vxml:prompt>
      <vxml:elseif cond="app == 'home'"/>
        <assign name="window.location" expr="index"/>
      <vxml:elseif cond="app == 'link 1'"/>
        <assign name="window.location" expr="x1x"/>
     ...
      <vxml:elseif cond="app == 'link 10'"/>
        <assign name="window.location" expr="x10x"/>
      </vxml:if>
    </vxml:filled>
  </vxml:form>
<script src="java.js" type="text/javascript"></script>
</head>

<body id="docBody" ev:event="load" ev:handler="#nav">
<div id="cont" ev:event="click" ev:handler="#nav">
<h1>Results for: The Beatles</h1>

<li>LINK 1:<a href="http://www.beatles.com/">
<b>The Beatles</b> Find out all about
The Beatles...</li>

...

<li> LINK 10: <a href="http://www.rarebeatles.com/">
<b>Songs, Pictures, and Stories of The Beatles</b>
Beatles website for collectors and fans ...</li>

</body></html>
```

Fig. 1. Translation of a HTML document into an equivalent XHTML+Voice file

of the environment based on observed or inferred events, states, and beliefs. In addition, the conversational agent requires a dialog strategy that defines the conversational behavior of the system. Thus, a great effort is employed to empirically design dialog strategies, as the design of a good strategy is far from trivial since there is no clear definition of what constitutes a good strategy [11]. Additionally, speech recognition grammars for conversational agents have been usually build on the basis of handcrafted rules which are tested recursively,

which in complex applications constitutes a costly process [8]. As an alternative of the previously described rule-based approaches, the application of statistical approaches to dialog management makes it possible to consider a wider space of dialog strategies. The main reason is that statistical models can be trained from real dialogs, modeling the variability in user behaviors [11,5].

We propose to merge statistical approaches with VoiceXML in order to benefit from the flexibility of statistical dialog management and the facilities that VoiceXML offers. Our technique employs a statistical dialog manager that takes into account the history of the dialog until the current moment in order to decide the next system prompt, whereas the system prompts and the grammars which indicate the possible user responses to them are implemented in VoiceXML [5]. This technique is based on the definition of a data structure that we call Dialog Register (DR), and contains the information provided by the user throughout the previous history of the dialog. For each time i, the selection of the next system prompt A_i is carried out by means of the following maximization:

$$\hat{A}_i = \underset{A_i \in \mathcal{A}}{\operatorname{argmax}} P(A_i | DR_{i-1}, S_{i-1})$$

where set $\mathcal{A}$ contains all the possible system answers and S_{i-1} is the state of the dialog sequence (*system-turn, user-turn*) at time i.

Each user turn supplies the system with information about the task; that is, he/she asks for a specific concept and/or provides specific values for certain attributes. However, a user turn could also provide other kinds of information, such as task-independent information. This is the case of turns corresponding to *Affirmation*, *Negation* and *Not-Understood* dialog acts. This kind of information implies some decisions which are different from simply updating the DR_{i-1}. For that reason, for the selection of the best system answer A_i, we take into account the DR that results from turn 1 to turn $i-1$, and we explicitly consider the last state S_{i-1}.

The selection of the system answer is then carried out through a classification process, for which a soft-computing methodology based on multilayer perceptrons (MLP) is proposed. The input layer receives the codification of the pair (DR_{i-1}, S_{i-1}). The output generated by the MLP can be seen as the probability of selecting each of the different system answers defined for a specific task.

A corpus of dialogs for the specific task is required to learn the dialog model. Our approach for automatically acquiring a dialog corpus is based on the interaction of a user agent simulator and a conversational agent simulator [6]. In our dialog simulation technique, both agents use a random selection of one of the possible responses defined for the semantics of the task (expressed in terms of user and system dialog acts). At the beginning of the simulation, the set of system responses is defined as equiprobable. When a successful dialog is simulated, the probabilities of the answers selected by the the conversational agent simulator during that dialog are incremented before beginning a new simulation.

Errors and confidence scores are simulated by a specific module in the simulator using a model for introducing errors based on the method presented in [9]. The generation of confidence scores is carried out separately from the model employed for error generation. This model is represented as a communication

channel by means of a generative probabilistic model $P(c, a_u|\tilde{a}_u)$, where a_u is the true incoming user dialog act $\tilde{a}_u$ is the recognized hypothesis, and c is the confidence score associated with this hypothesis. The probability $P(\tilde{a}_u|a_u)$ is obtained by Maximum-Likelihood using the initial labeled corpus acquired with real users and considers the recognized sequence of words w_u and the actual sequence uttered by the user $\tilde{w}_u$.

$$P(\tilde{a}_u|a_u) = \sum_{\tilde{w}_u} P(a_u|\tilde{w}_u) \sum_{w_u} P(\tilde{w}_u|w_u) P(w_u|a_u)$$

Confidence score generation is carried out by approximating $P(c|\tilde{a}_u, a_u)$ assuming that there are two distributions for c.

$$P(c|a_w, \tilde{a}_u) = \begin{cases} P_{corr}(c) & if \quad \tilde{a}_u = a_u \\ P_{incorr}(c) & if \quad \tilde{a}_u \neq a_u \end{cases}$$

3 Practical Applications

To test our proposal, we have developed several applications corresponding to both general-purpose speech interfaces and dialog-based interactive conversational agents. In order to provide web-content using speech, an interactive voice response (IVR) platform is required. We have selected the Prophecy IVR Platform[4]. This IVR can interpret the VoiceXML language and act like a client to the web-servers. This way, it can translate an incoming request to a URL, fetch the document, interpret it and return the output to a mobile client. The Prophecy Media Resource Control Protocol (MRCP) media server has been used for prompting, recording, automatic speech recognition, text-to-speech generation, and conferencing functionalities.

3.1 General-Purpose Applications

Regarding general-purpose applications we have developed *Voice Dictionary* and *Voice Browser*. Both applications consists of a set of X+V documents. Some of them are stored from the beginning in the server of the application, while others are dynamically generated using PHP and JavaScript. This dynamic generation takes into account the information extracted from different web servers and MySQL databases in the system, and a set of users preferences and characteristics (e.g., sex, preferred language for the interaction, number of previous interactions with the system, and preferred application).

The *Voice Browser* application has been developed with the main objective of allowing speech access to both the search and presentation of the results in the interaction with the Google search engine. The application interface receives the contents provided by the user and displays the results both visually and using synthesized speech. This application also allows the multimodal selection of any of the links included in the result of the search by numbering them and allowing

[4] http://www.voxeo.com/products/voicexml-ivr-platform.jsp

using their titles as voice commands. Detailed instructions, help messages and menus have been also incorporated to facilitate the interaction.

The *Voice Dictionary* application offers a single environment where users can search contents in the Wikipedia encyclopedia with the main feature that the access to the application and the results provided by the search are entirely facilitated to the user either through visual modalities or by means of speech. Once the result of an initial search is displayed on the screen and communicated to the user by means of speech, they can easily access any of the links included in the result of the search or visit the rest of applications in the system with the possibility of interrupting the system's speech in any case. This functionality is achieved by means of the dynamic generation of the corresponding grammars, in which the different links that are present in the result of a specific search are included in the dynamic XHTML+Voice page automatically generated by means of a PHP script that captures the different information sources to inform the user about them (headings, text, contents, formulas, links, etc.).

A number of tests and verifications have been carried out to maximize the functionalities and accessibility of these two applications. These tests have been very important to detect and correct programming errors and accessibility problems. In addition, we have completed a preliminary assessment by means of a questionnaire to measure users subjective opinion about the system. The questionnaire contained five questions: i) Q1: *Did the system correctly understand you during the interaction?*; ii) Q2: *Did you understand correctly the messages of the system?*; iii) Q3: *Was it simple to obtain the requested information? / Was it simple to play the game?*; iv) Q4: *Do you think that the interaction rate was adequate?*, v) Q5: *Was it easy to correct mistakes made by the system?*; vi) Q6: *In general terms, are you satisfied with the performance of the system?* The possible answers to the complete set questions were the same: *Never*, *Rarely*, *Sometimes Usually* and *Always*. A numerical value between one and five was assigned for each answer (in the same order as they are shown in the questionnaire). Table 1 shows the average, maximum and minimum values obtained from the results provided by a total of 35 students and professors of our University using the different modules of the system without predefined scenarios.

The results of the preliminary evaluation of both applications show that the users who participated in the assessment positively evaluate the facility of obtaining the requested information by interacting with the system, the appropriate interaction rate during the dialog, and overall operation of the different applications in the system. The main problems mentioned by the users include the need

Table 1. Results of the preliminary evaluation of the *Voice Dictionary* and *Voice Search Engine* applications (1=minimal value, 5=maximum value)

	Q1	Q2	Q3	Q4	Q5	Q6
Average value	3.6	3.8	3.2	3.7	3.2	4.3
Maximum value	4	5	5	4	4	5
Minimal value	2	3	2	3	2	3

of improving the word error rate and achieve a better clarification of the action expected by the system at each moment of interaction. In addition, the 97% of the interactions finished achieving the objective(s) expected by the user, only the 4% of the systems turns corresponded to reprompts and the 12% to system confirmations. The error correction rate (computer as the average number of corrected errors per dialog divided by the number of corrected and uncorrected errors) was 91%.

3.2 Web Applications Based on Interactive Dialog

To test our proposal with a conversational agent focused on an interactive dialog with users, we have used the definitions taken to develop the EDECAN dialog system, which was developed in a previous study to provide information about train services, schedules and fares in Spanish [5]. The developed conversational agent generates a total of 51 different prompts.

A total of 100,000 dialogs was simulated using our user simulation technique and a set of scenarios covering the different queries for the system [6]. Then, the acquired dialogs were employed to automatically generate VoiceXML code for each system prompt and the grammar needed to correctly recognize the possible user responses. The 51 different system prompts have been automatically generated in VoiceXML using the proposed technique. For example, Figure 2 shows the VXML document to prompt the user for the origin city and the obtained grammar for ASR.

The DR defined for the system is a sequence of 15 fields, corresponding to the five possible queries that users can carry out to the system (*Hour, Price, Train-Type, Trip-Time, Services*) and the ten attributes that they can provide to complete these queries (*Origin, Destination, Departure-Date, Arrival-Date, Departure-Hour, Arrival-Hour, Class, Train-Type, Order-Number, Services*). This way, every dialog begins with a dialog register in which every value is equal to 0 and the greeting turn of the system, as it is showed following.

. . . . . .
S_1: *Welcome to the railway information system. How can I help you?*
A_1: (*Opening*:*Nil*:*Nil*)
DR_0: 00000-1000001000
. . . . . .

Each time the user provides information, this is used to update the previous DR and to obtain the new one. For instance, given a user turn providing the origin city, the destination city and the date, the new dialog register could be as follows.

. . . . . .
U_1: *I want to know timetables from Valencia to Madrid.*
Task Dependent Information: (*Hour*) [0.7] *Origin*:Valencia [0.2] *Destination*:Madrid [0.9]
Task Independent Information: None
DR_1: 10000-2100000000
. . . . . .

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml xmlns="http://www.w3.org/2001/vxml"
   xmlns:xsi="http://www.w3.org/2001/
   XMLSchema-instance"
   xsi:schemaLocation="www.w3.org/2001/vxml
   http://www.w3.org/TR/voicexml20/vxml.xsd"
   version="2.0" application="trains.vxml">
 <form id="origin_form">
  <field name="origin">
     <grammar type="application/srgs+xml"
           src="/grammars/origin.grxml"/>
     <prompt>Tell me the origin city.</prompt>
     <filled>
       <return namelist="origin"/>
     </filled>
  </field>
 </form>
</vxml>
```

```
#JSGF V1.0;
grammar origin;
public <origin> = [<desire>]
[<travel> <city> {this.destination=$city}]
[<proceed> <city> {this.origin=$city}];
<desire> = I want [to know] | I would like
[to know] | I would like | I want | I need
| I have to;
<travel> = go to | travel to | to go to
| to travel to;
<city> = Murcia | Vigo | Sevilla | Huelva |
Cuenca | Lugo | Granada | Salamanca |
Valencia | Alicante | Albacete | Barcelona
| Madrid;
<proceed> = from | going from | go from;
```

Fig. 2. VoiceXML document to require the origin city (left) and grammar to capture the associated value (right)

In this case, the confidence score assigned to the attribute *Origin* (showed between brackets in the previous example) is very low. Then, a "2" value is added in the corresponding position of the DR_1. The concept (*Hour*) and the attribute *Destination* are recognized with a high confidence score, adding a "1" value in the corresponding positions of the DR_1. Then, the input of the MLP is generated using DR_1, the codification of the labeling of the last system turn (A_1), and the task-independent information provided in the last user turn (none in this case). The output selected for the MLP would consist in the case of requiring the departure date. This process is repeated to predict the next system response afterwards each user turn.

A total of 25 dialogs was recorded from interactions of six students and professors of our University employing the conversational agent developed for the task following our proposal. We considered the following measures for the evaluation: i) Dialog success rate (*%success*). This is the percentage of successfully completed tasks; ii) Average number of turns per dialog (nT); iii) Confirmation rate (*%confirm*). It was computed as the ratio between the number of explicit confirmations turns (nCT) and the number of turns in the dialog (nCT/nT); iv) Average number of corrected errors per dialog (nCE). This is the average of errors detected and corrected by the dialog manager; v) Average number of uncorrected errors per dialog ($nNCE$). This is the average of errors not corrected by the dialog manager; vi) Error correction rate ($\%ECR$). The percentage of corrected errors, computed as nCE/ (nCE + nNCE).

The results presented in Table 2 show that the developed conversational can interact correctly with the users in most cases, achieving a success rate of 94%. The dialog success depends on whether the system provides the correct data for every objective defined in the scenario. The analysis of the main problems

Table 2. Results of the evaluation of the railway information conversational agent

	%success	nT	%confirm	%ECR	nCE	nNCE
Conversational Agent	94%	10.6	37%	93%	0.89	0.06

detected in the acquired dialogs shows that, in some cases, the system did not detect that the user wanted to finish the dialog given that the the system was developed following a mixed dialog initiative, which allow users to control the dialog flow without requiring the use of submit commands. A second problem was related to the introduction of data in the DR with a high confidence value due to errors generated by the automatic speech recognizer that were not detected by the dialog manager. However, the evaluation confirms a good operation of the approach since the information is correctly given to the user in the majority of cases, as it is also shown in the value of the error correction rate.

4 Conclusions

In this paper, we have described a technique for providing speech access to Internet by means of conversational agents. Our proposal works on the benefits of statistical methods for dialog management and XHTML+Voice. The former provide an efficient means to exploring a wider range of dialog strategies, whereas the latter makes it possible to benefit from the advantages of using the different tools and platforms that are already available to simplify system development.

Two applications have been developed to study the XHTML+Voice to develop multimodal conversational agents that improve the accessibility to information on the Internet. These conversational agents respectively facilitate the multimodal access for the search of contents in the Wikipedia encyclopedia, and the complete implementation of a speech-based interface to an Internet search engine. We have also applied our technique to develop a conversational agent that provides railway information, which integrates our statistical dialog management technique for creating automatically VoiceXML documents to prompt the user for data, as well as the necessary grammars for ASR. This conversational agent has been enhanced by means of a user simulation technique in order to facilitate the automatic learning of the dialog model and the provision of system responses that are adapted to the specific evolution of the dialog.

The results of the subjective and objective evaluations of these agents show an appropriate interaction rate during the dialog and overall operation of the different applications in the system, thus providing a solution to both general-purpose speech-based interfaces to access web contents and user-adapted conversational agents oriented to slot-filling dialog tasks. Current research lines include the adaptation of the systems for its interaction using additional languages, more complex domains, and also considering information about users' preferences.

Acknowledgements. Research funded by projects CICYT TIN2011-28620-C02-01, CICYT TEC2011-28626-C02-02, CAM CONTEXTS (S2009/TIC-1485), and DPS2008-07029-C02-02.

References

1. Beskow, J., Edlund, J., Granström, B., Gustafson, J., Skantze, G., Tobiasson, H.: The MonAMI Reminder: a spoken dialogue system for face-to-face interaction. In: Proc. of Interspeech/ICSLP, pp. 296–299 (2009)
2. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
3. Danielsen, P.J.: The Promise of a Voice-Enabled Web. Computer 33(8), 104–106 (2000)
4. González Ferreras, C., Escudero Mancebo, D., Cardeñoso Payo, V.: From HTML to VoiceXML: A First Approach. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2002. LNCS (LNAI), vol. 2448, pp. 266–279. Springer, Heidelberg (2002)
5. Griol, D., Hurtado, L.F., Segarra, E., Sanchis, E.: A Statistical Approach to Spoken Dialog Systems Design and Evaluation. Speech Communication 50(8-9), 666–682 (2008)
6. Griol, D., Sánchez-Pi, N., Carbó, J., Molina, J.M.: An Agent-Based Dialog Simulation Technique to Develop and Evaluate Conversational Agents. In: Proc. of PAAMS 2011. AISC 2011, vol. 88, pp. 255–264 (2011)
7. López-Cózar, R., Araki, M.: Spoken, Multilingual and Multimodal Dialogue Systems. John Wiley & Sons Publishers (2005)
8. McTear, M.F.: Spoken Dialogue Technology: Towards the Conversational User Interface. Springer, Heidelberg (2004)
9. Schatzmann, J., Thomson, B., Weilhammer, K., Ye, H., Young, S.: Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System. In: Proc. of HLT/NAACL 2007, pp. 149–152 (2007)
10. Shao, Z., Capra, R.G., Pérez-Quiñones, M.A.: Transcoding HTML to VoiceXML Using Annotation. In: Proc. of ICTAI 2003, pp. 249–258 (2003)
11. Young, S.: The Statistical Approach to the Design of Spoken Dialogue Systems. Tech. rep., Cambridge University Engineering Department, UK (2002)