



Universidad  
Carlos III de Madrid



This is a postprint version of the following published document:

Griol, David; Sanchís de Miguel Ángel, Araceli; Molina, José Manuel. *Giving Voice to the Internet by Means of Conversational Agents*. In: Corchado E., Lozano J.A., Quintián H., Yin H. (eds.) *Intelligent Data Engineering and Automated Learning – IDEAL 2014*. IDEAL 2014. *Lecture Notes in Computer Science*, vol 8669, pp. 441-448. Springer. [https://doi.org/10.1007/978-3-319-10840-7\\_53](https://doi.org/10.1007/978-3-319-10840-7_53)

© Springer International Publishing Switzerland 2014

# Giving Voice to the Internet by Means of Conversational Agents<sup>\*</sup>

David Griol, Araceli Sanchis de Miguel, and José Manuel Molina

**Abstract.** In this paper we present a proposal to develop conversational agents that avoids the effort of manually defining the dialog strategy for the agent and also takes into account the benefits of using current standards. In our proposal the dialog manager is trained by means of a POMDP-based methodology using a labeled dialog corpus automatically acquired using a user modeling technique. The statistical dialog model automatically selects the next system response. Thus, system developers only need to define a set of files, each including a system prompt and the associated grammar to recognize user responses. We have applied this technique to develop a conversational agent in VoiceXML that provides information for planning a trip.

**Keywords:** Conversational Agents, Spoken Interaction, POMDPs, Machine Learning, User Modeling, Neural Networks.

## 1 Introduction

A conversational agent can be defined as a software that accepts natural language as input and generates natural language as output, engaging in a conversation with the user [4]. Thus, these interfaces make technologies more usable, as they ease interaction, allow integration in different environments, and make technologies more accessible, especially for disabled people. Usually, these agents carry out five main tasks: Automatic Speech Recognition (ASR), Spoken Language Understanding (SLU), Dialog Management (DM), Natural Language Generation (NLG), and Text-To-Speech Synthesis (TTS). These tasks are typically implemented in different modules of the system's architecture.

When designing this kind of agents, developers need to specify the system actions in response to user utterances and environmental states that, for example, can be based on observed or inferred events or beliefs. This is the fundamental task of dialog management [4], as the performance of the system is highly

---

<sup>\*</sup> This work has been supported in part by the Spanish Government under i-Support (Intelligent Agent Based Driver Decision Support) Project (TRA2011-29454-C03-03), and Projects MINECO TEC2012-37832-C02-01, CICYT TEC2011-28626-C02-02, and CAM CONTEXTS (S2009/TIC-1485).

dependent on the quality of this strategy. Thus, a great effort is employed to empirically design dialog strategies for commercial systems. In fact, the design of a good strategy is far from being a trivial task since there is no clear definition of what constitutes a good dialog strategy [7].

Once the dialog strategy has been designed, the implementation of the system is leveraged by programming languages such as the standard VoiceXML [6], for which different programming environments and tools have been created to help developers. These programming standards allow the definition of a dialog strategy based on scripted Finite State Machines. With the aim of creating dynamic and adapted dialogs, the application of statistical approaches to dialog management makes it possible to consider a wider space of dialog strategies [7].

The most extended methodology for machine-learning of dialog strategies consists of modeling human-computer interaction as an optimization problem using Markov Decision Process (MDP) and reinforcement methods [2]. The main drawback of this approach is the large state space, whose representation is intractable if represented directly [9]. Partially Observable MDPs (POMDPs) outperform MDP-based dialog strategies since they provide an explicit representation of uncertainty [7]. Other interesting approaches for statistical dialog management are based on modeling the system by means of Hidden Markov Models, stochastic Finite-State Transducers, or using Bayesian Networks.

Additionally, the design of speech recognition grammars for the ASR and SLU tasks have been usually built on the basis of handcrafted rules that are tested recursively, which in complex applications is very costly [3]. However, as stated by [4], many sophisticated commercial systems already available receive a large volume of interactions. Therefore, industry is becoming more interested in substituting rule based grammars with other statistical techniques based on the large amounts of data available.

As an attempt to improve the current technology, we propose to combine the flexibility of statistical dialog management with the facilities that VoiceXML offers, which would help to introduce statistical methodologies for the development of commercial (and not strictly academic) dialog systems. To this end, our technique employs a POMDP-based dialog manager. Expert knowledge about deployment of VoiceXML applications, development environments and tools can still be exploited using our technique. The only change is that transitions between dialog states is carried out on a data-driven basis (i.e., it is not a deterministic process). In addition, the system prompts and the grammars for ASR are implemented in VoiceXML-compliant formats (e.g., JSGF or SRGS).

Pietquin and Dutoit [5] described a similar proposal based on a graphical interface dedicated to ease the development of VoiceXML-based dialog systems. The main aim is focused on enabling non-specialist designers to semi-automatically create their own systems. In this case, the results of a MDP-based strategy learning method are provided in order to facilitate the design of the dialog strategy for the VoiceXML system. Speech grammars are not automatized by the proposal. Our goal is to make developers' work even easier with a very simple design of each VoiceXML file (they are only reduced to a system prompt and an automatic

generated speech grammar) and the complete automation of the dialog strategy (the next system prompt, i.e. the next VoiceXML file, is automatically selected by the statistical dialog model).

The remainder of the paper is as follows. Section 2 describes our proposal to integrate spoken interaction to web-based systems by means of the combination of a statistical dialog manager and a Voice-XML complaint platform. Section 3 presents the application of our proposal to develop a commercial system for planning a trip. This section also presents the results of its preliminary evaluation. Finally, Section 4 presents some conclusions and future research lines.

## 2 Our Proposal to Provide a Spoken Access to the Web

The application of POMDPs to model a spoken conversational agent is based on the classical architecture of these systems shown in Figure 1 [7]. As this figure shows, the user has an internal state  $S_u$  corresponding to their goal and the dialog state  $S_d$  represents the previous history of the dialog. Based on the user’s goal prior to each turn, the user decides some communicative action (also called intention)  $A_u$ , expressed in terms of dialog acts and corresponding to an audio signal  $Y_u$ . Then, the speech recognition and language understanding modules take the audio signal  $Y_u$  and generate the pair  $(\tilde{A}_u, C)$ .

This pair consists of an estimate of the user’s action  $A_u$  and a confidence score that provides an indication of the reliability of the recognition and semantic interpretation results. This pair is then passed to the dialog model, which is in an internal state  $S_m$  and decides what action  $A_m$  the conversational agent should take. This action is also passed back to the dialog manager so that  $S_m$  may track both user and machine actions. The language generator and the text-to-speech synthesizer take  $A_m$  and generate an audio response  $Y_m$ . The user listens to  $Y_m$  and attempts to recover  $A_m$ . As a result of this process, users update their goal state  $S_u$  and their interpretation of the dialog history  $S_d$ .

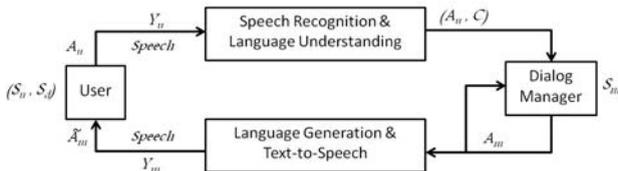


Fig. 1. Classical architecture of a conversational agent

One of the main reasons to explain the challenge of building conversational agents is that  $\tilde{A}_u$  usually contains recognition errors (i.e.,  $\tilde{A}_u \neq A_u$ ). As a result, the user’s action  $A_u$ , the user’s state  $S_u$ , and the dialog history  $S_d$  are not directly observable and can never be known to the system with certainty. However,  $\tilde{A}_u$

and  $C$  provide evidence from which  $A_u$ ,  $S_u$ , and  $S_d$  can be inferred. Therefore, when using POMDPs to model a conversational agent, the POMDP state  $S_m$  expresses the unobserved state of the world and can naturally be factored into three distinct components: the user’s goal  $S_u$ , the user’s action  $A_u$ , and the dialog history  $S_d$ . Hence, the factored POMDP state  $S$  is defined as:

$$s_m = (s_u, a_u, s_d) \quad (1)$$

The belief state  $b$  is then a distribution over these three components:

$$s_m = b_s = b(s_u, a_u, s_d) \quad (2)$$

The observation  $o$  is the estimate of the user dialog act  $\tilde{A}_u$ . In the general case this will be a set of  $N$ -best hypothesized user acts, each with an associated probability

$$o = [(\tilde{a}_u^1, p_1), (\tilde{a}_u^2, p_2), \dots, (\tilde{a}_u^N, p_N)] \quad (3)$$

where  $p_n = P(\tilde{a}_u^N | o)$  for  $n = 1 \dots N$ .

The transition function for an SDS-POMDP follows directly by substituting the factored state into the regular POMDP transition function and making independence assumptions:

$$\begin{aligned} P(s'_m | s_m, a_m) &= P(s'_u, a'_u, s'_d | s_u, a_u, s_d, a_m) = \\ &= P(s'_u | s_u, a_m) P(a'_u | s'_u, a_m) P(s'_d | s'_u, a'_u, s_d, a_m) \end{aligned} \quad (4)$$

This is the transition model. Making similar reasonable independence assumptions regarding the observation function gives,

$$P(o' | s'_m, a_m) = P(o' | s'_u, a'_u, s'_d, a_m) = P(o' | a'_u) \quad (5)$$

This is the observation model. The above factoring simplifies the belief update equation since substituting (8) and (9) into (1) gives

$$\begin{aligned} b'(s'_u, a'_u, s'_d) &= k \cdot \underbrace{P(o' | a'_u)}_{\text{Observation model}} \underbrace{P(a'_u | s'_u, a_m)}_{\text{User action model}} \\ &\sum_{s_u} \underbrace{P(s'_u | s_u, a_m)}_{\text{User goal model}} \cdot \sum_{s_d} \underbrace{P(s'_d | s'_u, a'_u, s_d, a_m)}_{\text{Dialog model}} b(s_u, s_d) \end{aligned} \quad (6)$$

As shown in Equation 6, the probability distribution for  $a'_u$  is called the user action model. It allows the observation probability to be scaled by the probability that the user would speak  $a'_u$  given the goal  $s'_u$  and the last system prompt  $a_m$ . The user goal model determines the probability of the user goal switching from  $s_u$  to  $s'_u$  following the system prompt  $a_m$ . Finally, the dialog model enables information relating to the dialog history to be maintained such as grounding and focus.

The optimization of the policy is usually carried out by using techniques like the Point-based Value Iteration or Q-learning, in combination with a user simulator. Q-learning is a technique for online learning where a sequence of sample dialogs are used to estimate the Q functions for each state and action. Given that a good estimate of the true Q-value is obtained if sufficient dialogs are done, user simulation has been introduced to reduce the too time-consuming and expensive task to obtain these dialogs with real users.

Simulation is usually done at a semantic dialog act level to avoid having to reproduce the variety of user utterances at the word or acoustic levels. At the semantic level, at any time  $t$ , the user is in a state  $s_u$ , takes action  $a_u$ , transitions into the intermediate state  $s'_u$ , receives machine action  $a_m$ , and transitions into the next state  $s''_u$ . To do this, we propose the use of a recently developed user simulation technique based on a classification process in which a neural network selects the next user response by considering the previous dialog history [1].

We also propose to merge statistical approaches with VoiceXML. To do this, a VoiceXML-compliant platform (such as Voxeo Evolution<sup>1</sup>) is used for the creation of Interactive Voice Response (IVR) applications and the provision of telephone access. Static VoiceXML files and grammars can be stored in the voice server. We propose to simplify these files by generating a VoiceXML file for each specific system prompt. Each file contains a reference to a grammar that defines the valid user's inputs for the corresponding system prompt.

The conversational agent selects the next system prompt (i.e. VoiceXML file) by consulting the probabilities assigned by the POMDP-based statistical dialog manager to each system prompt given the current state of the dialog. This module is stored in an external web server. The result generated by the statistical dialog manager informs the IVR platform about the most probable system prompt to be selected for the current dialog state. The platform just selects the corresponding VoiceXML file and reproduces it to the user.

### 3 Development of a Conversational Agent to Plan a Trip

We have applied our proposal to develop and evaluate the *Your Next Trip* system, which provides tourist information useful to plan a trip. The system was developed to provide telephonic access to the contents of a web portal that is updated dynamically from different web pages, databases, and the contribution of the users, who can add and edit the contents.

Figure 2 shows different snapshots of the portal, which contents include cities, places of interest, weather forecast, hotel booking, restaurants and bars, shopping, street guide, cultural activities (cinema, theater, music, exhibitions, literature and science), sport activities, festivities, and public transportation. In addition to provide specific information related to the previously described categories, the system also provides user-adapted recommendations based on the opinions and highest rated places in the application.

---

<sup>1</sup> <http://evolution.voxeo.com/>

Users can access these functionalities visually by means of the different web pages or orally by means of the application of our proposal with the combination of the POMDP-based dialog manager and the Voxeo Evolution Voice-XML complaint platform.

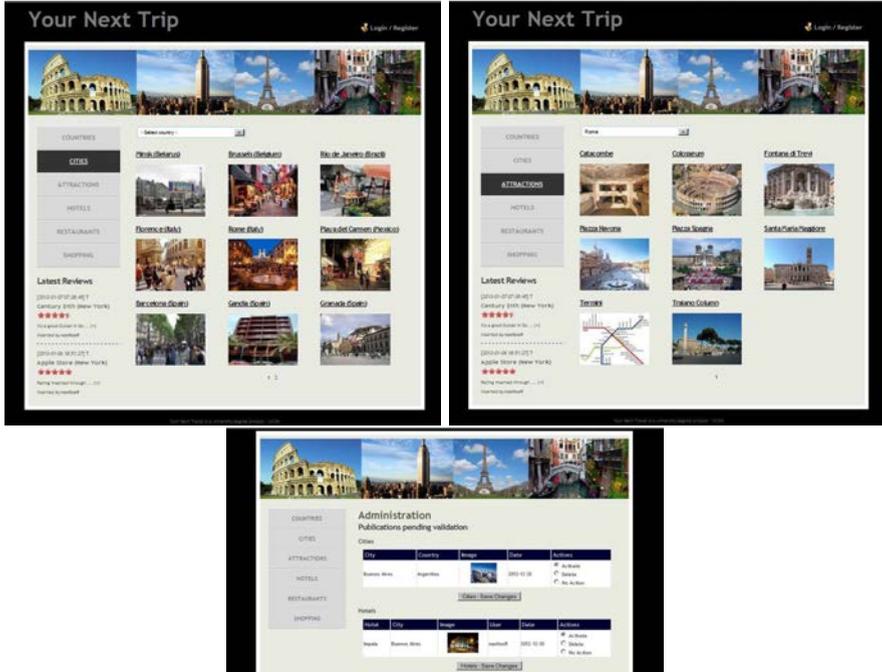


Fig. 2. Different snapshots of the *Your Next Trip* system

With regard to the POMDP-based dialog manager, rewards in the conversational agent were given based on the task completion rate and the number of turns in the dialog. The user modeling module described in [1] was initially trained using the 100 dialogs acquired with a Wizard of Oz experiment in which an expert simulated the system operation. The dialog manager was implemented and trained via interactions with the simulated user model to iteratively learn a dialog policy. A total of 150,000 dialogs was simulated. Using the definitions described in [8] for the summary Q-learning algorithm, the POMDP system was given 20 points for a successful dialog and 0 for an unsuccessful one. One point was subtracted for each dialog turn.

To assess the benefits of our proposal, we have already completed a preliminary evaluation of the developed system with recruited users and a set of scenarios covering the different functionalities of the system. A total of 150 dialogs for each agent was recorded from the interactions of 25 recruited users. These users

followed a set of scenarios that specify a set of objectives that must be fulfilled by the user at the end of the dialog and are designed to include the complete set of functionalities previously described for the system.

We asked the recruited users to complete a questionnaire to assess their opinion about the interaction. The questionnaire had seven questions: i) Q1: *How well did the system understand you?*; ii) Q2: *How well did you understand the system messages?*; iii) Q3: *Was it easy for you to get the requested information?*; iv) Q4: *Was the interaction with the system quick enough?*; v) Q5: *If there were system errors, was it easy for you to correct them?*; vi) Q6: *How did the system adapt to your preferences?*; vi) Q7: *In general, are you satisfied with the performance of the system?* The possible answers for each questions were the same: *Never/Not at all, Seldom/In some measure, Sometimes/Acceptably, Usually/Well, and Always/Very Well*. All the answers were assigned a numeric value between one and five (in the same order as they appear in the questionnaire).

Also, from the interactions of the users with the system we completed an objective evaluation of the application considering the following interaction parameters: i) question success rate (*SR*), percentage of successfully completed questions: system asks - user answers - system provides appropriate feedback about the answer; ii) confirmation rate (*CR*), computed as the ratio between the number of explicit confirmations turns and the total of turns; iii) error correction rate (*ECR*), percentage of corrected errors.

Table 1 shows the average results of the subjective evaluation using the described questionnaire. It can be observed that the users perceived that the system understood them correctly. Moreover, they expressed a similar opinion regarding the easiness to understand the system responses. In addition, they assessed that it was easier to obtain the information specified for the different objectives, and that the interaction with the system was adequate and adapted to their preferences. An important point remarked by the users was that it was difficult to correct the errors and misunderstandings generated by the ASR and NLU processes in some scenarios. Finally, the satisfaction level also shows the correct operation of the system.

**Table 1.** Results of the preliminary evaluation with recruited users (For the mean value M: 1=worst, 5=best evaluation)

Q1	M = 4.45, SD = 0.49	
Q2	M = 4.37, SD = 0.47	
Q3	M = 4.05, SD = 0.55	
Q4	M = 3.66, SD = 0.53	
Q5	M = 3.19, SD = 0.61	
Q6	M = 3.89, SD = 0.46	
Q7	M = 4.21, SD = 0.32	
	SR	CR
	94.36%	19.00%
		ECR
		92.11%

The results of the objective evaluation for the described interactions show that the developed system could interact correctly with the users in most cases, achieving a success rate of 94.36%. The fact that the possible answers to the user's responses are restricted made it possible to have a very high success in speech recognition. Additionally, the approaches for error correction by means of confirming or re-asking for data were successful in 92.11% of the times when the speech recognizer did not provide the correct input.

## 4 Conclusions and Future Work

In this paper, we have described a proposal to provide spoken interaction to the web. Our proposal works on the benefits of the POMDP statistically method for dialog management and VoiceXML, respectively. The former provides an efficient means to explore a wider range of dialog strategies and also introduce user adaptation, 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.

We have applied our technique to develop a conversational agent that provides information to plan a trip, and have . The results of its evaluation show that the described technique can predict coherent system answers in most of the cases, also obtaining a high user's satisfaction level. As a future work, we plan to study ways for adapting the proposed statistical model to more complex domains.

## References

1. Griol, D., Carbo, J., Molina, J.: A statistical simulation technique to develop and evaluate conversational agents. *AI Communications Journal* 26(4), 355–371 (2013)
2. Levin, E., Pieraccini, R., Eckert, W.: A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on Speech and Audio Processing* 8(1), 11–23 (2000)
3. McTear, M.F.: *Spoken Dialogue Technology: Towards the Conversational User Interface*. Springer (2004)
4. Pieraccini, R.: *The Voice in the Machine: Building Computers That Understand Speech*. MIT Press (2012)
5. Pietquin, O., Dutoit, T.: Aided Design of Finite-State Dialogue Management Systems. In: *Proc. of ICME 2003*, vol. 3, pp. 545–548 (2003)
6. Rouillard, J.: Web services and speech-based applications around VoiceXML. *Journal of Networks* 2(1), 27–35 (2007)
7. Schatzmann, J., Weilhammer, K., Stuttle, M., Young, S.: A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies. *Knowledge Engineering Review* 21(2), 97–126 (2006)
8. Thomson, B., Schatzmann, J., Weilhammer, K., Ye, H., Young, S.: Training a real-world POMDP-based Dialog System. In: *Proc. of HLT 2007*, pp. 9–16 (2007)
9. Young, S., Schatzmann, J., Weilhammer, K., Ye, H.: The Hidden Information State Approach to Dialogue Management. In: *Proc. of ICASSP 2007*, vol. 4, pp. 149–152 (2007)