From HTML to VoiceXML: A first approach.

César González Ferreras, David Escudero Mancebo, Valentín Cardeñoso Payo.
Departamento de Informática. Universidad de Valladolid. 47011 Valladolid, Spain.
{cesargf,descuder,valen}@infor.uva.es

Abstract

In this work, we discuss the construction process of the voice portal counterpart of a departamental web site. VoiceXML has been used as the dialog modelling language. A prototypical system has been built using our own VoiceXML interpreter, which easily integrates different implementation platforms. A general discussion of VoiceXML advantages and disadvantages is also reported and a simple startup procedure is proposed as a means to build voice portals starting from legacy web sites.

1 Introduction

Internet is our biggest information repository nowadays and web browsing the natural access mechanism to it. Pervasive as it is, however, network browsing is not the most used way to access information in everyday life. Mobile devices cope the market and, although new ways of interaction on 3G wireless telephony are under way, voice is the prominent communication channel in this environment.

Starting from a historically rich background, spoken language technology is becoming a central issue for many companies and the number of scenarios where vocal interaction is a must is steadily increasing. This is specially true when turning into multimodal interaction, where voice input could be the perfect complement to classical navigational paradigms based on graphical user interfaces.

Nevertheless, not enough attention has been paid yet to methodological approaches to spoken dialog systems design and scarcely any reference can be found to ways of exploiting all the legacy information systems that companies are developing and maintaining using generic web technology.

Furthermore, when such methodologies exist, they usually focus on multidevice deploying technologies where WAP and HTTP integration is the typical scenario.

Since spoken dialog based interaction is essentially affected by sequential mode of operation, complemented in best cases with some form of asynchronous reaction from the user side, the careful design of spoken dialog systems starting from their counterparts in web environments is still an open question.

In this work, we present our first efforts towards the definition of a integration methodology of web contents navigation and spoken driven information accessing. To this end, we present a first case of study in which a simple web site has been analyzed and rebuilt as a voice portal, using a VoiceXML platform developed at our site, which lays down an experimental environment where we can test future proposals.

2 Description of the Problem

The goal of our application is to access information through the telephone line. It is the voice portal version of the Computer Science Department web site at *Universidad de Valladolid*.

The system started from HTML material from the web pages of the Department, where you can find general information, detailed information of the teaching activities, a description of the research teams and contact information of each member.

All this information has to be provided through a voice interface, so we have to be aware of the characteristics of the voice compared to GUI: voice interaction must be sequential, while visual interaction can show all the information at once [4]. This implies that an explicit navigation model is needed, and a technique or language for modelling applications.

3 Solution

Finite State Diagrams (FSD) have been used to model the dialog. Using this formalism, we can specify dialog states and transitions between them [5]. In each state there is a system message, which will be send to the user, and a language model (LM), to describe the expected user input. Each transition has a condition which is used to select the next state.

This is the easiest approach, where the system has the initiative and guides the user through the contents of the portal. There are some limitations of flexibility, although this model is really suitable for navigation, because it resembles the web one.

To map web contents to a FSD, representing the way the system makes the navigation, we have proposed a set of rules to automatize that transformation:

1. The structure of the web site must be captured using a graph. This graph can be used as our first FSD, which has to be improved to achieve an user-friendly interaction.
2. Each state in the FSD with large amount of information to be send to the user has to be changed: navigation through its contents must be allowed. There are two ways of doing this:
   (a) Split that state into several ones, adding the proper transitions between them.
   (b) Enable a search mechanism through the contents of that state.

Eventually, LM must be associated to each state:

1. A grammar made up of all literal strings found in all associated conditions that go out from that state.
2. In search states, we have to build a grammar that can cope with all possible queries we can expect from the user.

Computer: Welcome to the Computer Science Department at Universidad de Valladolid ... Choose an option: information, teaching activities, research, directory.
Human: Directory
C: Directory service of the Computer Science Department. Tell me the name or surname of the person you are looking for:
H: Pérez.
C: Juan Pérez Crespo, phone number: 3456. Luisa Martín Pérez, phone number: 3452 ... Any other query?
H: No.
C: Do you want to exit?
H: Yes.

Table 1: Sample Human-Computer interaction.

The finite state diagram associated to our application is shown in figure 2, which evolved from the web structure in figure 1.

A typical interaction scenario with the user can be seen in table 1.

4 Underlying Platform

Once the application is modelled, the next step is to design the underlying platform to implement the whole system.

Voice eXtensible Markup Language (VoiceXML) was chosen to modelling dialogs. It is one of the emerging voice/telephone technology, and is a standard that allows layered independence between the application and the technological platform at hand. As a result, developed applications can be executed in different platforms.

The **Document Server** is a common web server, which delivers VoiceXML pages for their interpretation. This model allows dynamically generated pages, making the implementation of voice applica-

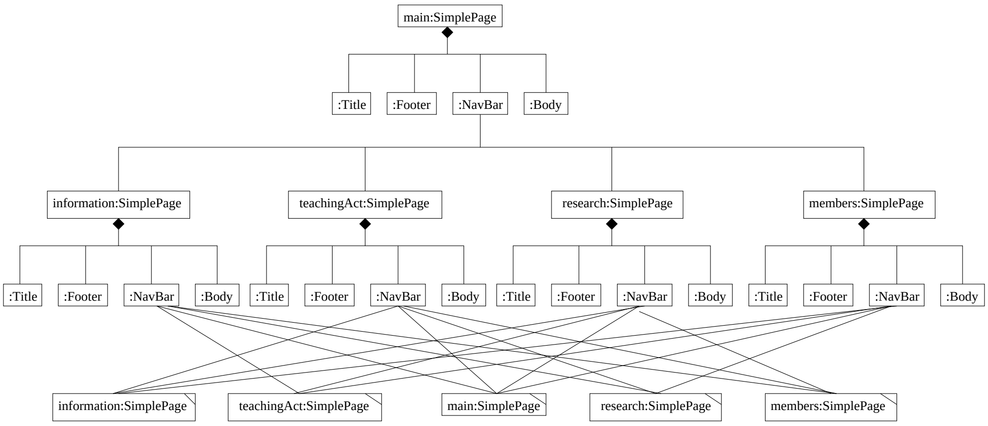

Figure 1: Department Web Site Structure.

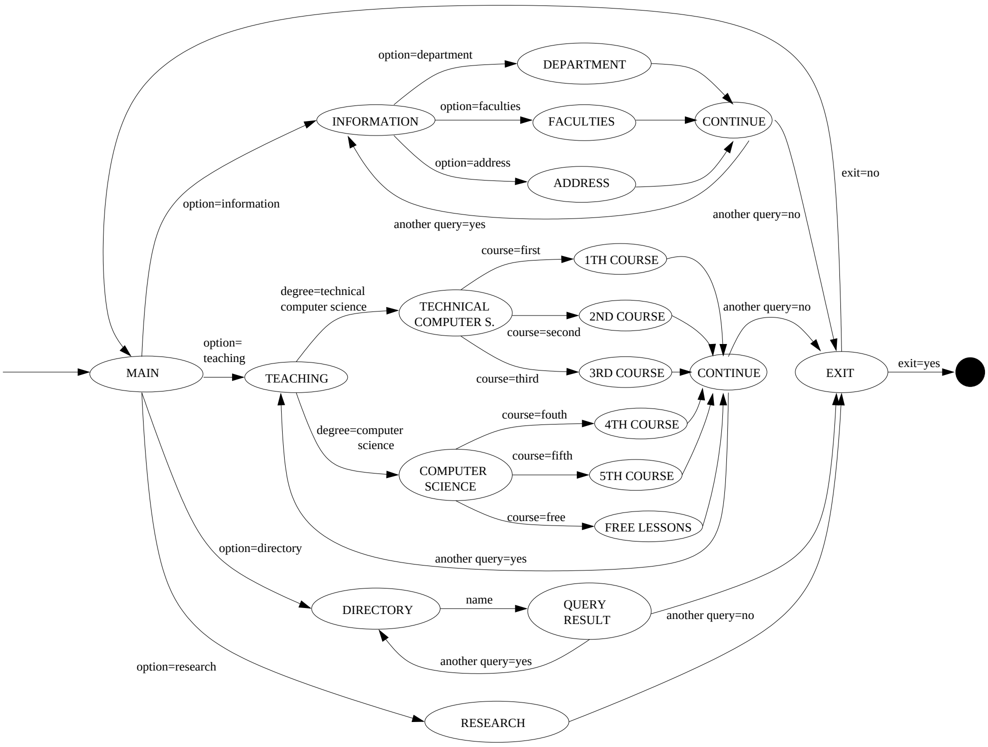

Figure 2: State Diagram of the application.

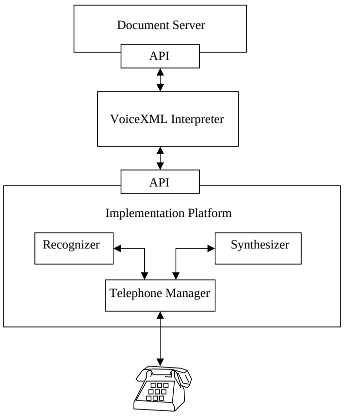

Figure 3: System Architecture.

tions more easy, and provides straightforward DB interoperability, an essential element of almost all dialog systems.

The **VoiceXML Interpreter** processes VoiceXML pages received from the Document Server and controls the Implementation Platform which copes with all low-level details of user interaction. The interpretation of the VoiceXML pages is done by the *Form Interpretation Algorithm* (*FIA*), as is described in the VoiceXML specifications [3].

The **Implementation Platform** hides the details of the hardware and software system responsible for the interaction with the user to the VoiceXML interpreter.

The implementation platform is made of three components, as can be seen in figure 3: synthesizer, recognizer and telephone manager. It was developed using both commercial and experimental ASR and TTS.

The system developed by our group implements the version 1.0 of the standard [3], but we are expecting to adapt to the newer version 2.0 [2], actually under development.

5 Conclusions and Future Work

Our main goal, the development of a Voice Application was achieved. We have also build a VoiceXML environment, which enables easy development of vocal applications.

VoiceXML is becoming a standard for browsing voice contents. It takes ideas from HTML, and borrows web information browsing model [4]. No doubt its key to success relies on application portability [1].

But that is not enough. An extension of the standard would be needed to overcome some of its limitations (e.g. [6]). There are two main points to be addressed: support of spontaneous user utterances and the ability to design more flexible dialogs which lets the user to take the initiative.

A major challenge would have to do with direct HTML contents reuse for spoken dialog environments. This is not an easy task, since web pages lack the structure needed to model voice navigation. A reorganization of contents is also needed to achieve a natural and user-friendly interaction with the user.

A set of rules were presented to map web contents into voice ones, although further work has to be done for accessing complex sites.

References

[1] Peter J. Danielsen. The promise of a voice-enabled web. *Computer*, 33(8), August 2000.

[2] W3C Working Draft. Voice eXtensible Markup Language (VoiceXML) Version 2.0, October 2001. <http://www.w3.org/TR/2001/WD-voicexml20-20011023/>.

[3] VoiceXML Forum. Voice eXtensible Markup Language (VoiceXML) Version 1.0, March 2000. <http://www.voicexml.org/specs/VoiceXML-100.pdf>.

[4] Bruce Lucas. VoiceXML for web-based distributed conversational applications. *Communications of the ACM*, 43(9), September 2000.

[5] David Sadek and Renato De Mori. *Spoken Dialogs with computers*. Academic Press, 1998.

[6] Augustine Tsai, Andrew N. Pargellis, Chin-Hui Lee, and Joseph P. Olive. Dialogue session management using VoiceXML. In *Eurospeech*, 2001.