Multimodal, Multilingual and Adaptive Dialogue System for Ubiquitous Interaction in an Educational Space

Ramón López-Cózar[1], *Zoraida Callejas*[1], *Miguel Gea*[1], *Germán Montoro*[2]

[1] Dept. Languages and Computer Systems, Granada University, Spain
[2] Dept. Computer Science and Communications, University Autónoma of Madrid, Spain
rlopezc@ugr.es, zoraida@correo.ugr.es, mgea@ugr.es, German.Montoro@uam.es

Abstract

This paper presents our current work in the UCAT project (Ubiquitous Collaborative Adaptive Training) concerned with setting up a multimodal, multilingual and adaptive dialogue system. The system goal is to assist teachers and students in some of their usual activities within an educational space (e.g. a University Faculty). The user-system interaction is carried out by means of speech, text, graphics and direct manipulation, either in English or Spanish. The system is adaptive as the interaction is carried out considering the user preferences previously stored in a user-profile database. The interaction is ubiquitous as the system takes into account the environment in which the user is interacting. Also, we plan the system can operate automatically some environment devices (e.g. professor offices' lights). The paper describes the architecture, current setting up and usage of the system, and sets out possibilities for future work.

1. Introduction

Ubiquitous computing, also known as pervasive computing or ambient intelligence, is an emerging technology that offers new opportunities and challenges as requires to employ new user interfaces to interact with the environment [1, 2, 3]. Among other application fields, this technology has been applied to educational environments, allowing the personalisation and adaptation of educational tools to the needs and/or preferences of teachers and students. For example, in the Classroom 2000 project [4] a system was developed to store the teacher activity at the blackboard, making it accessible to students via web pages. The teacher used an electronic blackboard for his explanations, including slides and additional materials. This blackboard allowed the teacher to make hand-made annotations on the slides, as he would do on a classical blackboard. All the teacher's activity was stored by a ubiquitous system together with timestamps, and then was made available for students, who thus could concentrate better on the teaching activity itself as they did not need to take class notes.

2. DS-UCAT system

The work we present in this paper is related to the Classroom 2000 project, as the system we are developing, termed DS-UCAT (Dialogue System for Ubiquitous Collaborative Adaptive Training) aims at providing assistance to teachers and students in some of their usual activities within different environments of an educational space, concretely library, professor offices and classrooms of a University's Faculty. A system goal is that teachers can make available class materials on a web server, which can be accessed by students in their own languages, if available. The system is multimodal since the user-system interaction is carried out via sound, speech, graphics and text [5]. The multimodal input allows to combine several modalities in one interaction. For example, a user can ask for information about available books on a particular subject either by speaking the subject, selecting it on the computer screen, using the mouse, or writing the subject in a form field. In response, a spoken message indicates the requested information is available on the screen, where it appears as a list of books in text format. We plan the system can be proactive, so that it can provide users with messages generated from the environment. For example, when a student passes by near the library, the system can remind him/her of borrowed books that must be returned soon to the library.

Fig. 1 shows the architecture of the system, which is comprised of an X+V document server connected with the users' mobile devices (Tablet PCs, laptop computers and PDAs) through wireless connections. In the current implementation we are only using laptop computers, which connect to the server through the wireless network of our lab.

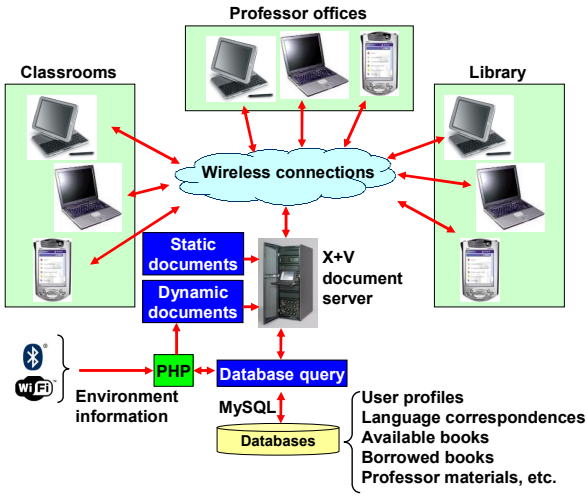

Figure 1: Architecture of the DS-UCAT system.

2.1. XHTML+Voice documents

The system is being set up using a toolkit [1] for building multimodal applications based on the W3C's XHTML+Voice language[2], also known as X+V. It is configured as a set of X+V documents, some of them are stored in the document server, while others are dynamically created using PHP

[1] http://www-306.ibm.com/software/pervasive/multimodal/
[2] http://www.w3.org/TR/xhtml+voice/

programs that take into account the user features and preferences (e.g. gender and preferred interaction language), as well as the data extracted from the databases. X+V documents are comprised of forms with fields that are filled in with the user input via speech, text or mouse clicks. To visualise these documents, users run in their communication devices the Opera browser[1], which supports multimodal interaction (voice, text and graphics). This browser is automatically installed and configured when the multimodal toolkit is installed. In addition to recognise spoken sentences concerned with our educational application domain, this browser allows the user to control the web navigation by means of spoken commands (e.g. "Opera reload", "Opera stop", "Opera close all", etc.).

X+V documents contain two parts. One is written in VoiceXML to set up the spoken interface, and is placed in the <head> … </head> section of the X+V documents. The other part is written in XHTML to create the visual interface, and is placed in the <body> … </body> section of the documents. In the remaining of this section we describe briefly the X+V documents used to implement the system.

2.1.1. Spoken interaction

In the system input, the speech recognition is carried out by the recogniser built-in in the Opera browser. In our setting the recognition is carried out in a tap-&-talk fashion, i.e. the user must click and hold a microphone button while speaking to the system (although it can be configured differently in the browser). In order to allow the spoken interaction, the user must enable [2] the voice feature of the browser, which automatically installs the necessary packages from the Internet. Speech recognition and understanding is carried out using JSGF grammars (Java Speech Grammar Format) that are used either at form or field level. Some of these grammars are static, while others are dynamically created by PHP programs which perform database queries and include the obtained data as grammar vocabulary (e.g. book titles). For example, using the grammar to recognise book queries, if a user utters the sentence *"I need books about Maths please"* the system fills in the form field "subject" with the word "Maths". The recognition grammars used to handle book queries must be updated as the library catalogue changes, so they are compiled dynamically from the contents of the *Available books* database. To update these grammars we have implemented a PHP program that carries out two tasks. Firstly, it accesses the database using MySQL functions and obtains the data from the available books, such as the book titles, authors and subjects. Secondly, it creates the grammars to recognise complete sentences as well as isolated data items (e.g. book title, authors and subjects) with the information gathered in the previous task.

In the system output, the speech synthesis is carried out by means of the sentences included into the <prompt> … </prompt> labels typically used in VoiceXML. These sentences are transformed into voice by a text-to-speech procedure using the Opera browser's built-in speech synthesiser. Some of these sentences are fixed, while others are created at run-time considering the user type (teacher or student), the user gender (which is necessary to create some Spanish sentences appropriately) as well as the data extracted from the databases.

2.1.2. Visual interaction

In the system input, the visual interaction is used to obtain data from the user via form fields and selection buttons typically used in XHTML (e.g. see Fig. 3). In the system output, the visual interaction is used to provide the data extracted from the database (e.g. list of available books) and information about the current user's name and type (e.g. see Fig. 2).

2.1.3. Connection of both parts

In XHTML+Voice, the connection between the spoken and the visual parts is carried out by events and event handlers, which are placed at the body section of the X+V documents. We have used some of the available X+V elements to handle these events. For example, when the document used to enter book queries is loaded into the browser, the event onload is thrown and, in response, a VoiceXML form called initial_vform is executed to handle this event. This procedure is carried out using the following X+V element:

```
<body ev:event="onload" ev:handler= #initial_vform">
```

XHTML+Voice also allows that a user utterance can fill in several form fields in one go (mixed initiative). To do so, we have used an <vxml:initial name="initial_vform"> …</initial> section, typically employed in VoiceXML, which allows recognising the user utterance using a form level grammar. Thus, in the case of the book query X+V document, the system generates the message *"Please enter a book query"* and the user can utter a variable number of data items (e.g. authors; authors and publication year; authors, publication year and subjects; etc.).

We also take into account the ev:event="onclick" event, which is thrown when the user clicks on a form field. The handler for this event is the VoiceXML code that obtains the value for that particular form field. For example, to obtain the book publisher when the user clicks on this field we use the following X+V element:

```
<input type="text" id="book_publisher" size="40"
ev:event="onclick" ev:handler="#voice_publisher"/>
```

2.2. Databases

In order to provide information to the users and interact with them properly, the system accesses several databases. One of them stores the user profiles, and contains personal data such as name, gender, address and telephone number. It also stores three personal preferences: i) use or not of oral interaction, ii) system voice type (male or female, if the oral interaction is selected), and iii) acceptation or not of incoming messages from the environment.

Another database stores language correspondences, i.e. expressions in several languages corresponding to particular sentence types which are used in the multilingual interaction, such as *"Bienvenido al sistema DS-UCAT"* for Spanish, and its correspondence in English *"Welcome to the DS-UCAT system"*.

Additionally, in the current configuration the system uses two other databases for experimental purposes, which in a real application of the system should be replaced by the real ones.

[1] http://www.opera.com

[2] This is only necessary if the browser is installed independently from the multimodal toolkit.

The *Available books* database stores information of books supposedly available in the Faculty's library, while the *Borrowed Books* database stores data about books borrowed by users of the educational space. Using these databases, the current system allows answering multimodal and multilingual queries, as the one shown in Fig. 3 for English. Using the form in this Figure, when the user clicks on a field, s/he receives a spoken message asking for the data to be entered (e.g. *"Book title?"*), that can be provided either orally or in text format. After the *Available books* database query is carried out, the system informs visually, using a table, about the available books, while at the same time generates the spoken message *"The following books are available"* (or *"No books were found"* if no records were retrieved from the database).

The *Professor materials* is a database to be created to store information about class materials made available by teachers. Our plan is that the X+V documents for the Classroom and Professor Office environments (also to be created) will show a view to query this database, similarly as the X+V document shown in Fig. 3 is used to query the *Available books* database.

2.3. Dialogue management

2.3.1. Interaction strategy

The system implements the two interaction strategies commonly used in spoken dialogue systems: mixed and system-directed. The use of one or another depends on the X+V document the user is interacting with. For example, when he interacts with the one to carry out book queries, the interaction is mixed in order to allow filling in several form fields in just one go (e.g. uttering the sentence *"I'd like to get information of books written by Allen concerned with Computer Science"*). If the sentence cannot be understood, the system prompts for the necessary data to query the database, field by field (system-directed strategy).

The system takes into account the typical events that can arise in a spoken interaction, i.e. the user asks for help, there is no input from the user, and the input cannot be recognised. These event types, usually called help, noinput and nomatch, respectively, are handled by the VoiceXML's catch element. For simplicity, the system generates the same response independently of the event type. As a sample, below is the X+V element used in the book query document to handle these events when entering the publication year:

```
<vxml:catch event="help nomatch noinput">
  Please say the publication year, for example 2005.
</vxml:catch>
```

2.3.2. Confirmation strategy

The confirmation strategy used in the current version of the system is explicit. For example, after the user utters the publication year for a book query, the system generates a confirmation prompt such as *"Did you say 2005?"*. The confirmation is handled in the VoiceXML part of the X+V documents, and is based on the confidence score provided by the VoiceXML interpreter. If the score is lower than a threshold, the data is considered unreliable and then the system generates the explicit confirmation.

2.4. System usage

To interact with the system, the user must firstly log in. From the login the system determines the user type (teacher or student) by querying the user-profile database. Finding out the user type allows adapting the interaction adequately in terms of interaction language (English or Spanish), use of oral interaction, and acceptation of incoming messages from the environment. As a sample, Fig. 2 shows the welcome X+V document shown on the browser for a user who selected English as the interaction language. As this user selected using oral interaction in his profile, the spoken message *"Hello Ramon, welcome to the DS-UCAT system"* is also generated when the document is shown.

As can be seen in Fig. 2, after the user has logged in he must choose an interaction environment (library, classroom or professor office). This initial selection allows him to interact in an environment which is not the one in which he is at the moment, thus enabling for instance to make book queries from a classroom.

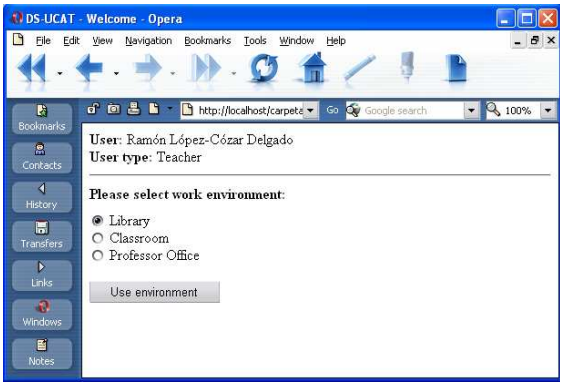

Figure 2: Welcome message (English interaction).

Fig. 3 shows the X+V document visualised in the browser if the user selects the Library environment. As this user selected the use of oral interaction in his profile, the message *"Please enter a book query"* is synthesised when the document is shown.

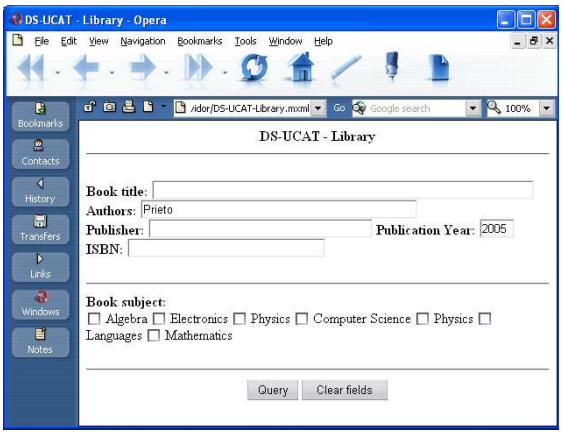

Figure 3: Book query (English interaction).

To enter a spoken book query, the user must click and hold the tap-&-talk key, or click on the browser's microphone icon

while speaking to the system. Alternatively, he can enter the data required to query the database using the keyboard and the mouse.

2.5. Ubiquitous interaction

Our goal is that the user-system interaction can be carried out ubiquitously in the educational space, in the sense that the environment in which the user is interacting can be taken into account without the user being concerned. However, as the user localisation within the educational space is not yet implemented, at the moment we simulate this ubiquitous information by fixing manually a variable that indicates the current user position within the educational space (i.e. either library, classroom or professor office). This variable's value is taken into account when some X+V documents are dynamically created. For example, in Fig. 2 the pre-selected environment is "Library" because the user is supposed to be in this environment. To simulate this localisation, we simply set the variable's value to "Library" before the interaction takes place.

In order to set the value for this variable automatically, we plan to implement a procedure to detect the environment change as the user moves within the educational space. Our aim is that whenever the environment changes (e.g. a student leaves the classroom and enters into the library), the browser shows a small window to suggest the use of the working environment that best fit the new user localisation. For example, this window will suggest to use the "Library" environment (i.e. the book query X+V document) if the user enters into the library. In this situation, the user will be free to choose (or not) the new, proposed environment.

As mentioned above, we also plan that, by means of the ubiquitous interaction, the system can automatically operate some devices in the environment to change their status. For example, in a professor office the system could turn on/off lights or ambient music as the professor enters/leaves the office. To set up this feature, we plan to adapt a previously-created middleware that works as a layer between models of several environments, the physical world and the user interfaces, in such a way the changes in a model are immediately reflected into the real world, and vice versa.

Figure 4: Ubiquitous home environment developed in the *Odisea* project.

This middleware was developed within the *Odisea* project [6] in order to set up a dialogue interface for a home environment (Fig. 4).

3. Conclusions and future work

This paper has presented our current work in the UCAT project concerned with developing a multimodal, multilingual and adaptive dialogue system to assist teachers and students in some of their usual activities within an educational space. The paper has focused on the architecture and setting up of the system, discussing the X+V documents, databases and dialogue management. It has also addressed briefly the system usage, focusing on the selection of the working environment and the making of book queries (library environment). Finally, it has addressed the ubiquitous interaction, discussing the work done at the moment.

Future work includes creating the X+V documents and databases necessary to interact within the classroom and professor office environments. It also includes setting up a procedure to automatically localise the user within the educational space, since at the moment the system adaptation to this factor is based on a simulated procedure. To implement this system's feature, we plan to experiment with Bluetooth emitters (one per environment due to their little coverage) and WI-FI access points (at least three to obtain a complete coverage of all the educational space). Using RFID is another possibility.

Another line of future work is concerned with setting up the system's *proactivity* in order to provide the user with incoming messages generated from the environment. Also, we plan to implement the connection between the system and the intelligent environment developed in the *Odisea* project, which will enable e.g. switching on/off lights or ambient music as professors come in/out their offices.

4. Acknowledgements

The research presented in this paper has been funded by the Spanish Ministry of Science and Technology, under project TIN2004-03140 Ubiquitous Collaborative Adaptive Training.

5. References

[1] Weiser, M. The computer of the twenty-first century. Scient. American, pp. 94-107, 1991.

[2] Streitz, N. A. Smart artefacts and the disappearing computer. Proc. Smart Objects Conference, pp. 9-10, 2003.

[3] Satyanarayanan, M. Pervasive Computing: Vision and Challenges. IEEE Personal Communications, 8:4, pp 10-17, 2001.

[4] Abowd, G. D., Atkeson, C., Feinstein, A., Goolamabbas, Y., Hmelo, C., Register, S., Sawhney, N., Tani, M. Classroom 2000: Enhancing classroom interaction and review. Georgia Institute of Technology, Technical Report GIT-GVU-96-21, 1996.

[5] López-Cózar, R., Araki, M. Spoken, Multilingual and Multimodal Dialogue Systems: Development and Assessment. John Wiley & Sons Publishers, 2005.

[6] Montoro, G., Haya, P. A., Alamán, X. Context adaptive interaction with an automatically created spoken interface for intelligent environments. International Conference on Intelligence in Communication Systems, Bangkok, Thailand. November, 2004.