VOICEXML-BASED DYNAMIC PLUG AND PLAY DIALOGUE MANAGEMENT FOR MOBILE ENVIRONMENTS

Botond Pakucs
Centre for Speech Technology (CTT)
KTH, Stockholm, Sweden
botte@speech.kth.se

Abstract In this paper it is argued for the necessity of a plug and play functionality in speech interfaces in mobile environments. Further, a VoiceXML based plug and play dialogue management solution is introduced. The paper focuses in particular on the plug and play functionality and the dynamic handling of the plug and playable dialogue management capabilities. The plug and play solution is applied in the SesaME architecture and employed within the framework of the PER demonstrator. Finally, first experiences related to the plug and play functionality of the dialogue management are discussed.

Keywords: Dialogue managment, speech interfaces, mobile environments, plug and play, VoiceXML, Java-to-XML mapping.

1. Introduction

Mobile and ubiquitous services will be of little use if the users have to find, identify and configure them. In mobile environments services and devices should be accessible directly, without the need of manually setting up parameters and configuring the functionality. For providing seamless access to locally available services and devices a plug and play functionality seems to be appropriate. Plug and play solutions offer fast access to services and devices without the need of manually locating and configuring them. Plug and play functionality supports also sharing of devices and services among several users. Therefore plug and play functionality is becoming more and more a desired feature of mobile services and devices.

For supporting a plug and play functionality in mobile environments a networked infrastructure is required. This networked infrastructure should support at least some standardised wireless communication capability, service look up, service discovery and some means for service coordination. Several different platforms, such as Jini (SunMicrosystems, 2002b) or Universal Plug and Play (UPnP) (Microsoft, 2000) provide this kind of functionality. Moreover, there exist already several experimental mobile service environments providing elaborated support for mobile users and mobile services such as the sView platform (Bylund, 2001) or the I-centric communication system (Arbanowski et al., 2001).

Speech based natural language interfaces appear to be desirable and advantageous in mobile situations and/or when hands and eyes are busy. The services the users may want to access by means of the voice can be classified in three major categories. The users may want to:

- *control appliances and environments*, such as, opening doors, calling elevators, controlling televisions, car stereos or the cooling in the office;

- *accessing public services and information* , such as financial information, weather services, ticket reservations, and various information retrieval tasks;

- *accessing personal information*, such as calendar data, address and phone lists, personal notes etc.

In mobile and ubiquitous computing environments speech interfaces should also support plug and play functionality. In mobile environments it is even more important to fine tune the speech based interactions to the current user and to the user's actual situation. For this purpose, a plug and play functionality appears to be the appropriate solution. Support for plug and play functionality in speech interfaces and dialogue management could be useful for reconfiguring system capabilities with new features and for extending the dialogue system capabilities with new tasks. A plug and play based solution could also enhance the portability of the dialogue system to new domains and languages.

In this paper a motivation will be given for the necessity of a dynamic plug and play functionality in mobile environments. Further, a VoiceXML based plug and play dialogue management solution will be described with focus on the dynamic handling of plug and playable dialogue management capabilities. The plug and play solution is employed in the framework of the PER application. Finally, some first experiences and further work will be presented and discussed.

2. Related work and background

Previously, in different spoken dialogue system projects and dialogue management toolkits, the plug and play concept was used for describing the ability to plug in and out different system components such as speech recognition, language processing (Rayner et al., 2001). These solutions were used for allowing researchers to experiment with different variants of the same system. Runtime reconfiguration of dialogue system components is not supported by these solutions.

Automated plug and play based reconfiguration of dialogue system components was developed in the framework of the DHomme project (Quesada et al., 2001; Rayner et al., 2001). The main goal was the enabling of the plug and play capabilities for recognition, parsing and semantic interpretation. The focus was on the management of the distributed linguistic information, the handling of distributed grammars. The plug and play functionality was applied in a single domain, the domain of networked home devices. Applying the plug and play functionality only for one single domain does not fit the requirements in the mobile environments.

In dynamically changing mobile environments the user's intentions and needs can change rapidly. The user should be able to initiate a new task while waiting for some other specific service to be completed and should have the possibility to easily cancel a previously issued command or change the parameters of some previously initiated service. Furthermore, the system itself should be able to interrupt an ongoing dialogue and direct the user's attention to higher priority events taking place in the user's immediate environment. Therefore, it is desired to support a wide range of domains within one and the same dialogue. For this reason, in mobile environments a *multi-domain approach* (Chung et al., 1999) is necessary to allow the user to transparently and seamlessly switch between several topic domains and services

2.1 User interface issues

Speech based interaction with ubiquitous services in mobile environments differs from accessing speech services through telephones or interacting with desktop based computers. Being on the run, and with hands, eyes and sometimes even with the mind busy, the user's requirements on the speech interfaces can be expected to increase. The user will not always have the time and patience to correct misrecognitions and misunderstandings.

Consequently, in mobile environments the user's interaction with the distributed services and appliances should be a not annoying interaction

also called *calm computing* (Weiser and Brown, 1996). The user should be able to concentrate upon the task to be performed and not be forced to cope with interface issues. Intuitiveness, ease of use, and seamless access are some major desired features. Providing such functionality and adapting and fine tuning the interactions to the specific user and the user's actual situation could be supported through some plug and play functionality.

One possible solution to provide such adaptations is to dynamically download user specific data and information to the service provider side, into the appliances. This solution raises however some serious integrity and security related issues. Can service providers be trusted, are services trustworthy? Which kind of user data should be downloaded to the service provider? What about integrity?

Another argument against this solution is the problems which may be caused by *multiple concurrent speech interfaces*. In the near future we may expect a multitude of speech interfaces embedded in the same environment. When several services and appliances are listening for user commands, or even taking initiative pro-actively, it is possible, due to errors or misrecognitions, that several speech based services may be triggered by a single user utterance. As far as we know, the effects of using several concurrent speech interfaces at the same time have never been studied. Means to coordinate and control the various speech interfaces are required to avoid the introduction of new usability related problems. The usability and user interface issues should be considered for whole environments rather than for isolated services and appliances.

2.2 A human-centered approach

An alternative solution for adapting and fine tuning the interaction to individual users and their actual situation is to download domain specific data to the user side into some personal appliance. Such a solution, a *human-centered approach*, was proposed for providing unobtrusive and user friendly speech interfaces and seamless access to services and appliances in mobile environments (Pakucs, 2002).

According to the human-centered approach each user is expected to use an individual and *highly personalised universal speech interface* to access a multitude of services and appliances. Application specific data, *service descriptions*, has to be stored locally at the service provider side. Accessing the services and appliances is done through the personalised speech interface integrated into some personal and wearable appliance such as a mobile phone or a PDA. The available service descriptions, including dialogue management capabilities, has to be dy-

namically plugged into the personalised speech interface whenever the user enters a new environment.

The human-centered approach provides extensive support for adapting and fine tuning the interaction to individual users. By employing a personalised speech interface the handling of the security and integrity issues is also facilitated. Moreover, due to the dynamic plug and play functionality all local services and appliances will be available for all potential users. Detailed description and discussion of the advantages and challenges of the human-centered approach is given in (Pakucs, 2002).

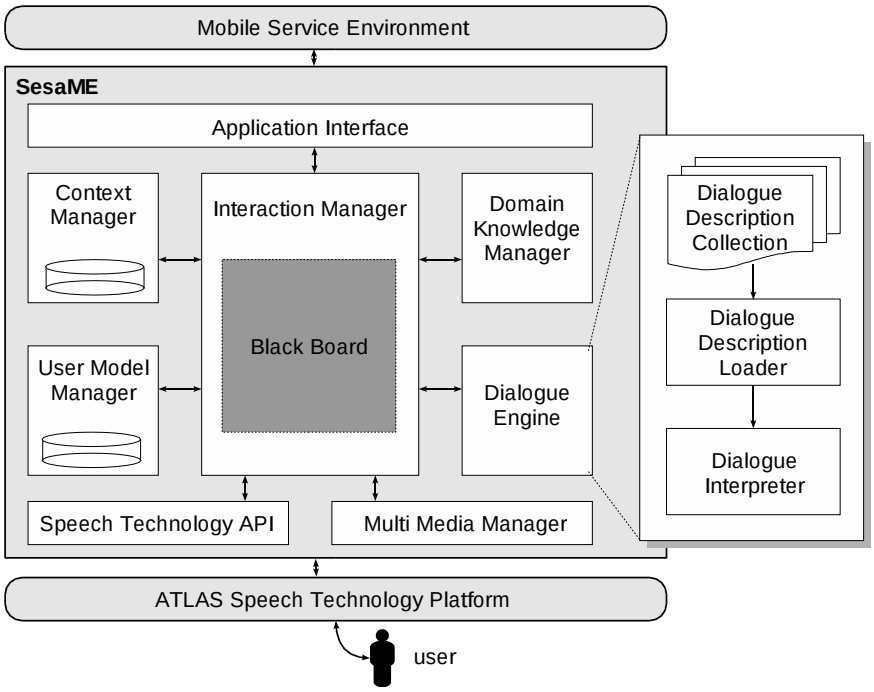

Figure 1. SesaME system architecture with the Dialogue Engine.

3. A VoiceXML based plug and play solution

SesaME, see Fig. 1, is a generic, task-oriented dialogue manager specially designed for the human-centered approach and for mobile environments. SesaME features an event based dialogue management and a central blackboard based architecture. SesaME relies on the Atlas generic speech technology platform (Melin, 2001). The Atlas platform provides high-level primitives for basic speech I/O, but access to low-level data is also facilitated. The ATLAS platform includes support for

the ACE speech recogniser (Seward, 2000), which features dynamical adaptation of the language model. One of the key issues in the SesaME architecture is the dynamic plug and play functionality of the dialogue management capabilities.

In the SesaME, the most of the operations related to the plug and play functionality are carried out by the *Dialogue Engine* (DE). However, synchronisation and communication with the mobile service environment is taken care by the *Application Interface*. Whenever new changes occur in the service environment (e.g. new services become available or existing services disappear etc.) the Application Interface dynamically updates the *Dialogue Description Collection* (DDC). All currently available dialogue descriptions are stored in the DDC. Beside the different task and domain specific dialogue descriptions DDC also contains resident application independent dialogue descriptions used for error handling or for meta dialogues necessary for providing information on available services etc.

For interoperability reasons, the application specific data including the dialogue descriptions has to be described in some standardized way. At the first development stage VoiceXML (W3C, 2001) has been chosen as the dialogue description language. The main reason is the fact that VoiceXML is a standardized markup language which shields the developers from low-level implementation details facilitating rapid application development. VoiceXML provides also support for simple grammars.

Procedurally, the internal plug and play functionality can be divided into three main parts:

- *identification* of the correct dialogue description,
- *activation* of the chosen dialogue description,
- the actual *dialogue management*.

At the current stage, the identification of the correct dialogue description is based on keywords extracted from the currently available VoiceXML document's $< meta >$ element. However context models, user models ore some planning mechanism could also be used for this purpose.

During the activation, the appropriate VoiceXML description is translated into internal data structures appropriate for the frame based dialogue management. This process is performed through JAXB (Sun-Microsystems, 2002a), the Java Architecture for XML Binding. JAXB provides a fast way to create a two-way mapping between XML documents and Java objects. Essentially JAXB can generate Java classes

based on DTD schema (XML Document Type Definition) without using computationally demanding XML parsing. This process is denoted unmarshalling and relies on specially developed translation schemes. Every DTD has an associated translation scheme which is used to generate precompiled Java classes. During the unmarshalling it is not necessary to perform dynamic interpretation and thus the used solution is domain and application independent. JAXB is an early access technology, and is still under development[1].

During the unmarshalling process different VoiceXML dialogue structures and related data such as prompts etc. are preserved and data structures appropriate for frame based dialogue management are generated. Moreover, two additional parallel data structures are also generated. One of these structures is dedicated to the user and context specific data, while the other is dedicated to the application specific data.

When, for the dialogue management necessary, data structures are generated the dynamic "in-plugging" of the dialogue management capabilities is finished. The actual dialogue interpretation do not follow the VoiceXML specifications. Accordingly, VoiceXML is only used as a dialogue description language. SesaME is not intended to be another VoiceXML interpretator.

During the dialogue interpretation process, the additionally generated parallel data structures can be accessed either by the *User Manager* and the *Context Manager* or respectively by the *Domain Knowledge Manager*. These structures may be updated with suggested pieces of information which can be used to adapt the interaction to the situation and to the user.

Application independent features of the dialogue management are handled by the *Interaction Manager* (IM). The IM handles also error prevention and detection, planning, keeps track of dialogue history and coordinates the different knowledge sources. A central, shared information storage, a blackboard, and a collection of autonomous software agents are the main components of the IM. The detailed description of the IM's functionality is, however, behind the scope of this paper.

4. Application

Before evaluating SesaME as a generic plug and play dialogue manager in mobile multi-domain environment, it is necessary to evaluate it as a domain dependent dialogue manager. The first evaluation of the plug

[1]Similar technologies and Java-to-XML data binding tools are provided by the Castor (http://castor.exolab.org/index.html) and the Zeus (http://zeus.enhydra.org/) projects.

and play functionality is conducted within the framework of the PER (Prototype Entrance Receptionist) project (Pakucs and Melin, 2001).

4.1 The PER demonstrator

PER, see Fig. 2, is an animated agent based automated receptionist located at the entrance of our department. Originally, the system has been developed to allow fast and robust access for employees. The application's functionality is stream-lined for this purpose. PER features a multilingual speaker verification system for Swedish and English. An employee, when approaching the gate, is expected to speak his password, which consists of the employee's name and a random digit sequence displayed on the screen. A set of sensors allow PER to detect the presence of people and to differentiate between people leaving or entering the premises. In this way PER can take initiative when someone arrives at the gate.

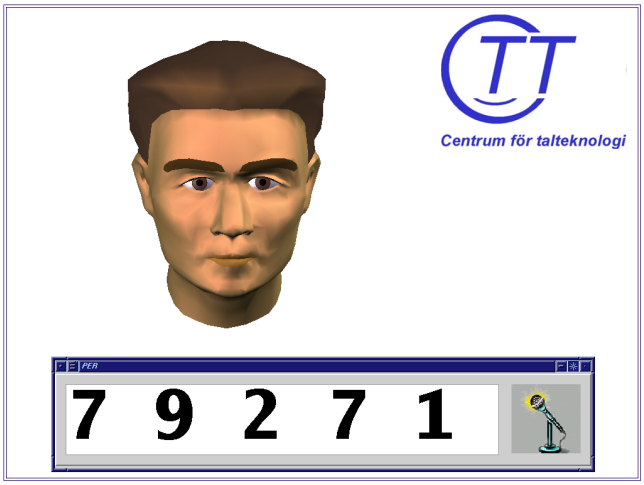

Figure 2. A screen shot of PER, as it meets the users.

4.2 Using VoiceXML

The functionality of PER is currently being extended to allow handling of external visitors. The interaction with the visitors relies on the SesaME dialogue manager and makes use of VoiceXML-based dialogue descriptions. PER features several different VoiceXML documents associated with different types of visitors and visitor goals such as expected personal guests, seminar visitors, students attending lectures. The handling of the VoiceXML descriptions, the identification of the correct

dialogue descriptions, the activation of these dialogue descriptions and the dialogue management, is done according to the description provided in the previous section.

The plug and play functionality allows us to incrementally update the system with new tasks even if the potential of dynamically updating the DDC with new dialogue descriptions at runtime is not used. For instance adding dialogue descriptions which helps to guide visitors in the building is planned. The plug and play functionality allows the extension of the system with support for new languages. Adding support for English speaking visitors is for instance planned.

The domain dependent data necessary for the dialogues is stored in an external database and is made available for manipulation through the web. Thus, employees can easily add to the database expected visitors, detailed information on seminars or lectures etc. or they can change the parameters of existing entries. In this way the data upon which PER operates is always kept up to date.

4.3 User studies

A pilot study was conducted to determine the appropriateness of the VoiceXML based receptionist services. Using VoiceXML dialogue descriptions a parallel, telephone-based, automated receptionist service was developed. A user study has been carried out with help of the VoiceXML based proprietary SpeechWeb platform (PipeBeach, 2002). Totally 6 different realistic scenarios where used. Twenty test subjects were asked to act as visitors in two different scenarios each.

The results proved that it is possible to use VoiceXML for an automated application. 93% of the subjects found the automated receptionist helpful. 87% of the interactions were successful. The user satisfaction was generally high. 75% of the users were positive about using such an automated receptionist service. The results of the user study are also intended to be used in a comparative study together with a user study of the SesaME based PER application.

5. Discussion

Some parts of the SesaME dialogue manager are still under development, but the VoiceXML-based plug and play functionality has been proved to be fast and useful. This paper demonstrates that, in spite of the shortcomings of the VoiceXML standard, it is possible to provide a dynamic plug and play solution for generic and flexible dialogue management using VoiceXML dialogue descriptions. The poorish support

for dialogue grammars in the VoiceXML specification is one, such a, shortcoming.

The total absence of the support for multi-modal interactions is another deficit of the VoiceXML standard. This is, however, not a limitation for the SesaME architecture. The PER application for instance features a 3D-animated talking head for multi-modal response generation. Additionally, the *Multi Media Manger* provides for instance means for generating XML based descriptions of the available services or the dialogue parameters currently being under discussion. By applying a generic output model for multi-modal response generation, such as GESOM (Edlund et al., 2002), it is possible to present the output on any kind of graphical output device that supports the output model.

During the further development of SesaME, support for plugand playable recognition grammars, based on the ACE speech recognizer, and dynamic updating of the language model will be integrated. Employing and evaluating SesaME in a multi-doamain mobile environment is also planned.

6. Conclusions

This paper, motivates the necessity of a plug and play functionality for speech interfaces in mobile environments. A plug and play based solution appears to be appropriate for fine tuning the speech based interactions to the current user and to the actual situation.

Further, a VoiceXML-based dynamic plug and play dialogue management solution was introduced. This plug and play functionality is implemented in SesaME, a modular and highly flexible dialogue manager. The plug and play functionality is also useful for runtime reconfiguration of system capabilities with new features and for extending the dialogue system capabilities with new tasks. A VoiceXML based plug and play solution could also enhance the portability of the dialogue system to new domains and languages.

This paper demonstrates that it is possible to use VoiceXML-based dialogue descriptions for accessing locally available services in mobile environments through a dynamic plug and play solution and still allow a generic and flexible dialogue management. A dynamic plug and play based dialogue management is a small step toward true multi-domain and generic dialogue management.

Acknowledgments

This research was carried out at the CTT, Centre for Speech Technology, a competence center at KTH, supported by VINNOVA (The

Swedish Agency for Innovation Systems), KTH and participating Swedish companies and organizations. This work was also supported by GSLT, The Swedish National Graduate School of Language Technology.

References

Arbanowski, S., van der Meer, S., Steglich, S., and Popescu-Zeletin, R. (2001). The human communication space: Towards I-centric communications. *Personal and Ubiquitous Computing*, 5:34–37.

Bylund, M. (2001). *Personal Service Environments - Openness and User Control in User-Service Interaction*. Licentiate thesis, Computer Science Department, Uppsala University, Sweden.

Chung, G., Seneff, S., and Hetherington, L. (1999). Towards multi-domain speech understanding using a two-stage recognizer. In *Proceedings of Eurospeech '99*, pages 2655–2658, Budapest, Hungary.

Edlund, J., Beskow, J., and Nordstrand, M. (2002). GESOM (generic system output model): a model for describing and generating multi-modal output. In *Proceedings of IDS02, ISCA Tutorial and Research Workshop on Multi-Modal Dialogue in Mobile Environments*, Kloster Irsee, Germany. (This volume).

Melin, H. (2001). ATLAS: A generic software platform for speech technology based applications. *TMH-QPRS, Quarterly Progress and Status Report*, 2001(1).

Microsoft (2000). Understanding universal plug and play. Available at: http://www.upnp.org/resources/whitepapers.asp.

Pakucs, B. (2002). A human-centered approach to speech interfaces in mobile and ubiquitous computing environments. *TMH-QPRS, Quarterly Progress and Status Report*, 2002.

Pakucs, B. and Melin, H. (2001). PER: A speech based automated entrance receptionist. *Presented at the 13th Nordic Computational Linguistic Conference, NoDaLiDa 2001*. Available at: http://www.speech.kth.se/~botte/.

PipeBeach (2002). speechWeb. Available at: http://www.pipebeach.com/.

Quesada, J., Amores, J., Bos, J., Ericsson, S., Gorell, G., Knight, S., Lewin, I., Milward, D., and Rayner, M. (2001). Configuring linguistic components in a plug and play environment. Deleiverable 3.1, D'Homme project. Available at: http://www.ling.gu.se/projekt/dhomme/.

Rayner, M., Lewin, I., Gorrell, G., and Boye, J. (2001). Plug and play speech understanding. In *Proceedings of 2nd SIGdial Workshop on Discourse and Dialogue*, Aalborg, Denmark.

Seward, A. (2000). A tree-trellis n-best decoder for stochastic context-free grammars. In *Proceedings of 6th International Conference on Spoken Language Processing*, Beijing, China.

SunMicrosystems (2002a). Java architecture for XML binding (JAXB). Available at: http://java.sun.com/xml/jaxb/.

SunMicrosystems (2002b). Jini architectural overview. Available at: http://www.sun.com/software/jini/whitepapers/index.html.

W3C (2001). Voice extensible markup language (voicexml) version 2.0. Available at: http://www.w3.org/TR/2001/WD-voicexml20-20011023/.

Weiser, M. and Brown, J. S. (1996). Designing calm technology. *PowerGrid Journal*. Available from: http://www.ubiq.com/hypertext/weiser/UbiHome.html.