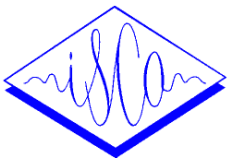

ITRW on Multi-Modal Dialogue
in Mobile Environments
Kloster Irsee, Germany
June 17-19, 2002

Connecting Mobile Dialogue Systems to the World Wide Web

René Schneider, Paul Heisterkamp
DaimlerChrysler AG

Abstract: We outline a dialogue system for the online reservation of hotels while driving a car. We describe the general system requirements and discuss the interdependencies that exist between dialogue, retrieval, and extraction modules as well as its architecture before illustrating the system at work.

Key words: mobile dialogue systems, world wide web, information extraction

1. INTRODUCTION[1]

Internet resources and services offer an ideal field of activity for speech and dialogue systems. They may not only be used in a home or office environment, but also in situations where keyboard or mouse are not within reach and hands must be used for other activities, e.g. while driving a car, where purely graphical user interfaces create a severe risk of driver distraction. Bearing in mind that - similar to broadcasting news - the in-car internet will become an integral part of the mobile society (Cairncross 1997), we have built an in-car based dialogue system that allows guided surfing the internet as available through telematics systems without interfering with the driving task or causing a traffic hazard.

The main advantage of these mobile dialogue systems that are connected to the internet is the fact that they give the driver access to the current or dynamic data base called world wide web as opposed to static databases like CD-ROMs (Abiteboutl 1997). One of the big advantages of the web is the

[1] The work described in this paper has in part been funded by Darpa as a part of the 'Communicator' program under contract N66001- 01-1-8901.

fact that it does not only supply the driver with up-to-date information but simultaneously supports him in his everyday activities and work.

2. SYSTEM REQUIREMENTS

Mobile dialogue systems face several very distinct problems, covering general aspects of human machine interaction, natural language processing and communication technology.

Firstly, the dialogue has to be intuitive and easy to follow, though the driver is in a situation where a good part of his concentration is needed to cope with the traffic. Thus, user friendliness is not only “nice to have,” but an indispensable must. As a consequence, the user should be able to chose between several modes to communicate with the system.

Secondly and as regards content, two major modules from the field of natural language processing have to be added: one for the interaction with the world wide web and a second one for extracting the retrieved information, i.e., methods from both information retrieval and information extraction build the coupling or cognitive interface between the dialogue system and the internet. Nevertheless, the system components implemented for this task both have to be regarded as subsidiaries to the speech and dialogue modules. This means that despite the considerable results that can be achieved with retrieval and extraction methods, the system builder should not forget that the results are not presented neither directly nor visually to the user, but via the speech dialogue. This should not be confronted with too much information it may not process or integrate appropriately. Furthermore these two components have to be accessible from a number of cars for several applications, so they should fulfil the requirements for a multi-user, multi-task and multi-thread system.

Thirdly and as regards the information technology, the data transfer between the in-car situated speech and dialogue system and the local server based retrieval and extraction component has to be ensured. So far this connection is done via GSM (Global System for Mobile Communications). Under real-world conditions, these connections are not reliably persistent but vulnerable to interruptions. Therefore every message that is sent between car and server is transmitted through a fresh connection which after successful transmission is shut down immediately.

Finally, we would like to highlight some problems concerning the connection between the local server or proxy on which the retrieval and extraction module are running, and the internet. Still too often, servers are not available. For this reason it is not recommended to rely on one single server only. This applies both to the servers being accessed from the cars via

GSM as to the internet servers and TCP/IP. Consequently it is advisable on the one hand to connect to two independent proxies and on the other hand to send the information request to two or more different www-servers with the extraction module being able to process to the output of the different servers.

3. SYSTEM ARCHITECTURE

In general, the architecture of the system can be divided into three major components as illustrated in Figure 1.

On-board the car, the complete speech recognition and synthesis systems as well as the dialogue system are installed. The dialogue system consists of a generic dialogue engine running as a stand-alone process, which interprets a dialogue model based on semantic object space states. The dialogue process communicates with the input and output components (recognizer, synthesis, applications) via a development environment that takes over the role of a resource manager for development, debugging and demonstration purposes. In real-life environments, a central system component that takes over these tasks.

This system includes a module for enabling data transfer to a local server which acts as a server to all in-car systems, and as a client towards the internet. Therefore it may be classified as a proxy. As specified in Section 2, the proxy includes two modules: one to simulate user behaviour while surfing the internet – the so called user agent – and one to adequately extract the responses received from the servers. The user agent's task consists mainly in: a) mapping the client's request to the CGI parameters of the servers selected and b) forwarding the servers' usable responses over to the extraction module. Another major task is to handle all server responses appropriately. This ranges from redirecting the query to another server if the first one is not available to shortening the dialogue steps, e.g. providing the server with the correct zip code for ambiguous city names.

The extraction module provides the system with several matching techniques for extracting the relevant information from the HTML output (consisting mainly of textual and iconic information) and storing them in domain specific templates under the corresponding ID number on the proxy. So far, the matching techniques used in the extraction module are based on regular expressions and thus quite brittle, i.e. that small changes in the HTML output require an adaptation of the extraction code. In the near future we plan to improve this by automatically adapting the matching expressions to changes in the output.

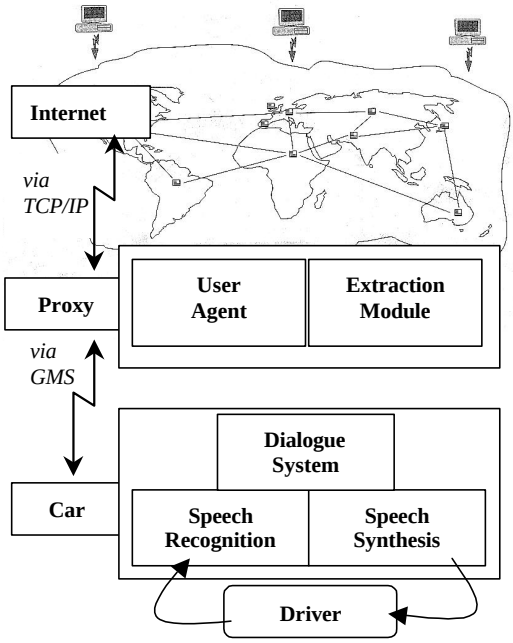

Figure 1. System Architecture

Communication between the car and the local server is realized as a client-server model with the restriction that the initiative for interaction comes only from the client, i.e. that the server responses are stored until a further request is received from the server (see Section 2).

4. THE SYSTEM AT WORK

This section shows an example of how the system works. As mentioned above, the task was to implement a system for the online reservation of available hotels. After the system has been started with the push-to-activate button located on the steering wheel, the user must choose the application s/he wants to use. This can be done explicitly by naming a system application, as shown in the example below, or implicitly, by making a request that is semantically mapped onto one of the applications. This is the way the system deals with mixed initiative.

The system prompts are graded in the amount of direct user direction they provide. The grades range from a help mode via a novice user and an

‘occasional’ user mode to an expert mode. The example below is in the ‘occasional’ user mode.

System: Welche Funktion*? (Telephone call or hotel reservation?*)
User: Hotelreservierung *(Hotel reservation).*
S: Hotelreservierung. In welcher Stadt? (*Hotel reservation. In which city?*)
U: Ich brauche ein Zimmer für zwei Personen in Dresden. *(I need a room for two people in Dresden.)*
S: Ein Doppelzimmer in Dresden? *(A double room in Dresden?)*
U: Ja. *(Yes.)*
S: Für welchen Tag? *(For which day?)*
U: Von Donnerstag bis Samstag. *(From Thursday to Saturday.)*
S: Von Donnerstag bis Samstag, also für zwei Nächte, ja? *(From Thursday to Saturday, so for two nights, right?)*
U: Ja. *(Yes.)*
S: Suche jetzt ein Doppelzimmer in Dresden von Donnerstag bis Samstag. *(Now searching a double room in Dresden from Thursday to Saturday.)*

The initial dialogue finishes with the confirmation of all relevant data. Afterwards, a compact string with this information is sent via GSM to the proxy. From there the internet request is started after the search parameters have been fitted to the parameters of the CGI programs or the servlets.

An identity number is assigned to any request using the time stamp of the time the request has been started. Whenever a request to one of the servers has been succesful, the relevant information is extracted and reduced to an embedded pair list. The following example is an extract from our system using Prolog notation:

```
[status:'OK',hitcount:'8',
 resultview:'hotels',
 searchres:[
     ['hotel_name':'Elbschleife',
     'street':'Elbufer 36',
     'zip_code':'01067',
     'city':'Dresden ',
     'single_room':'0',
     'double_room':'195.00',
     'email':'none',
     'category':'4',
     'tel':'none',
     'fax':'none'],
...
    ['hotel_name':'Hotel Rabenauer Hof,
     'street':'Hofstr. 19',
     'zip_code':'01734',
     'city':'Rabenau',
     'single_room':'75',
     'double_room':'130',
     'email':'rabenauer-hof@emails.de',
     'category':'3',
     'tel':'0351/xxxxxxx',
     'fax':'0351/xxxxxxx',
     'tel_in_room':'yes',
     'non_smoking_rooms':'yes',
     'special_price_for_children':'yes',
     'carpark':'yes',
     'quiet':'yes',
     'historical_building':'yes',
     'tv_in_room':'yes',
     'wellness_facilities':'yes'],... ]]
```

The pair list or template is stored on the proxy under the specific ID number of the request and may be interrogated by the client.

As can be seen from the results given in example above, the search for hotels is not simply restricted to the city name, but is also coupled with the zip code of the area to increase the number of hits for areas with lower hotel densities. The example above shows one hit from each of two www-servers.

In the next step the hotels in the list presented to the driver, ordered by price, starting with the most expensive.

System: Acht freie Hotels in Dresden gefunden: *(There are eight hotels available in Dresden)*: Nummer 1: Elbschleife , Nummer 2: Rabenauer Hof, . . ., Hotel 8: . . . Möchten Sie mehr Information über eines dieser Hotels? *(Would you like more information about any of these hotels?)*

User: Das zweite. *(The second one.)*

S: Hotel Rabenauer Hof, Hofstraße 19, Preis für ein Doppelzimmer 130 Mark. *(Hotel Rabenauer Hof, Hofstraße 19, price for a double room 140 Marks).* Möchten Sie in diesem Hotel reservieren? *(Would you like to make a reservation?)*

The user decides whether s/he wants more information on any of these hotels and then either makes a reservation or terminates the dialogue. The system's booking facility has not been implemented for legal reasons, not because of any unsolved problems in algorithms and technology. In a real application the driver could make a reservation via phone, e-mail, or online.

5. CONCLUSIONS AND FUTURE WORK

In this paper we presented the implementation of a speech dialogue based, mobile, online reservation system which is part of one of our research projects. The system is fully implemented (except the final step of booking) and fulfils all the requirements described in Section 2. An English version of the system designed for hotel reservation in the United States has been set up as well.

A internal study for user acceptance has shown that there is a growing interest and need for in-car speech based internet applications. For this reason, future work in the domain outlined in this paper will deal with the extension of our local server platform and the dialogue system in order to enhance their versatility. The scientific focus will be directed towards both flexible search and generic extraction mechanisms to allow the quick adaptation to different internet resources and services, such as route and location adaptive tourist information, traffic news, municipal parking systems, etc.

6. REFERENCES

Abiteboul, V. and Vianu, V. (1997). Queries and Computation on the Web, *Proceedings of the 6th International Conference on Database Theory (ICDT-97)*, 262-275.

Cairncross, F. (1997). *The Death of Distance: How the Communications Revolution will Change our Lives*. 1997.

Fujisaki, H., Kameda, H., Ohno, S., Abe, K. Iijima, J., Suzuki, M. and Taketa, K. (1999). Principles and Design of an Intelligent System for

Information Retrieval over the Internet wich a Multimodal Dialogue Interface. *Proceedings of Eurospeech '99*, 2467-2470.