Spoken Dialog Access to Web Content in the Car

Stefanie Schachtl[1], Manfred Gehrke[1], Matthias Schuster[1], Hans-Ulrich Block[1], André Berton[2], Peter Regel-Brietzmann[2]

[1]*Siemens AG Corporate Technology, 81730 Munich, Email: matthias.schuster@siemens.com*
[2]*DaimlerChrysler Research and Technology, 89013 Ulm, Email: andre.berton@ daimlerchrysler.com*

Introduction

The internet offers a lot of information that would be useful for the driver of a car. However, the driver may under no circumstances be distracted from driving while trying to access this information. In the SmartWeb project[1] Siemens, DaimlerChrysler and Fraunhofer FIRST work on a research prototype that allows the access of internet information in the car by speech[2]. The web information that is available to the driver on board is generated from two different sources in the web. One source are HTML tables, collected by a web crawler and automatically compiled into a speech dialog application, and the other source are Web services which offer online information on yellow pages, parking facilities, weather reports etc.. The speech dialogue applications for the web information in the onboard system are managed by a natural language DIAlogue engiNE (DIANE). DIANE communicates with the so called Meta Dialogue Manager of the infotainment system which controls the onboard ASR and TTS and Multimedia - Infotainment - UI (see the overview of the onboard system in Figure 1). In the following it will be explained, how the two kinds of natural dialogue applications for internet information are realized.

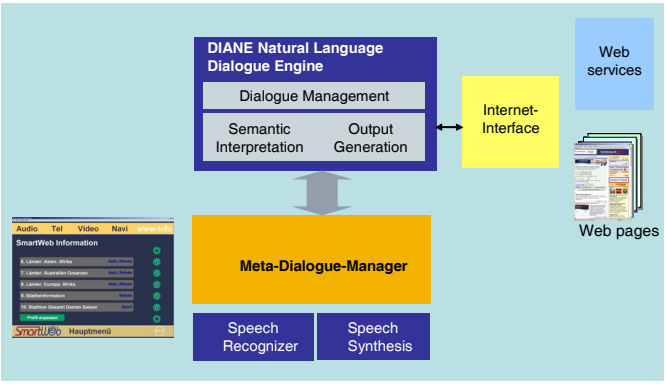

Figure 1: Client - Server - Architecture for Web Access in the

Access to HTML Table Content

All information in the internet is either structured or non-structured. Structured information which is described in tables is generally much quicker to understand than information hidden in continuous text. In many cases the reader is able to get the gist of the contents of a table in one glance, by scanning its title, the headlines of the columns and a sample of its values. Another important characteristic of tables is that they make it easy for the reader to compare between values, cf. Table 1:

Petrol Prices for Bonn and Surroundings

Station	Fuel	City	Code	Address	Price/l
Name 1	Normal	Bonn	53115	Street 1	1,189
Name 2	Normal	Siegburg	53721	Street 2	1,199
Name 3	Normal	Rheinbach	53359	Street 3	1,179
Name 1	Super	Bonn	53115	Street 1	1,239

Table 1: Example table for Petrol Prices

The basic idea of the Tablecompiler in the SmartWeb project was to use these characteristics of tables, in order to automatically generate speech dialogue applications from them. The first task in this process is to collect HTML-tables in the web, check them by heuristic methods in order to sort out those only used for formatting etc.. and then normalize the found tables, by unfolding multidimensional ones, resolving linked values in them and giving all tables the same orientation. After this, the linguistic content of the table is parsed in order to generate the basis for the natural language dialogue application. From this content the following units are generated: introduction, possible questions and possible answers. The introduction is necessary to tell the user what the speech application is about. Possible questions by the user must be modeled beforehand in order for the ASR to be able to recognize what was said, and possible answers must be modeled such that the TTS output is likely to be easily grasped by the user, giving sufficient and correct information on the contents .

The automatic generation of the introduction uses the title of the table, the headlines of the columns and a reference value from the table. Several heuristics have been modeled to produce the required information, even if it is deeper hidden in the table and not explicitly tagged. The gathered information is inserted into a special pattern that combines it to a natural phrase. From Table 1 above the following introduction would be generated: "*Petrol Prices in Bonn and surroundings, with information on Station, Fuel, City, Code, Address and Price per liter. You can ask me for example, what do you know about Name 1*?".The pattern has to be universal enough to fit for sport tables as well as for tables on financial information etc.

In order to guess the possible questions a hearer might ask after such an introduction, e.g. "*Where do I find a Shell station in Bonn*?" or "*How much is super in Bonn?*", several mechanisms are deployed. First, the system tries to identify the type of the column automatically. If this is successful, it assigns the standard grammars that belong to this type. E.g. the heading "*Price per liter*" followed by lots of numerical

[1] see Wahlster, W. (2004) for more information on the SmartWeb project.
[2] see Berton et al. (2006) for more information on the SmartWeb Car project.

values is likely to be of type price and would be assigned the standard grammars for asking for the price of something, i.e. phrases like "*How much is*", "*what's the price of*" etc. If no type can be identified, the necessary grammars are generated from the values themselves, by putting a "*which*" in front of the column headline, e.g. "*which city*", and enumerating the values, e.g. "*Bonn*" etc.. By these mechanisms e.g. the possible question "*Where do I find a Shell station in Bonn?*" is captured. Finally the typical phrases for comparison are added wherever a numeric value is found, adding "*cheapest*", "*cheaper than*" etc. to the vocabulary, so that the ASR can understand questions like: "*Where can I get the cheapest petrol in Bonn*?".
The automatically generated answer finally must confirm what was understood before giving the answer found in the table, e.g. with Table 1 a possible answer would be: "*As answer to your question about address, the cheapest price, Bonn, I found: Hauptstraße 14.*"
The resulting natural language dialogue system is a one shot question answer system, with the table constituting a form and the columns its fields. The contents of the table are stored in a serialized file to feed the answers, some algorithms are added in order to allow cross comparison of the numeric values. This means that all information is already contained in the system, so that, once generated from the internet, the application is self-contained, it does not need to be online. Only when the contents of the table change it has to be updated (see Berton et al. (2006) for a description of how the dialogue applications were transmitted to the car via broadcast in the SmartWeb Scenario).

Access to Web service Content

If, however, the car environment offers direct access to online information, e.g. via UMTS, direct voice access to Web services can be realized. A web service is a data base or a collection of data bases for which a description of its fields and access methods has been published in the WSDL format. Web services offer access to all kinds of information, e.g. to yellow pages (e.g. www.dialo.de), event calendars (e.g. www.eventful.com) or weather reports (e.g. www.wetteronline.de). If they are to be accessed by speech the necessary information has to be gathered from the user in a dialogue, cf. the following example dialogue:

User: Are there any concerts in Munich?
System: Concerts in Munich, on which day?
User: Tomorrow.
System: Tomorrow, Friday, tenth of April?
User: Yes
System: One moment please.
System: I found seven concerts. Alban Berg Quartett Haydn, Schönberg, Beethoven. Herkulessaal der Residenz. Start time 8 p.m.
System: More Than Soul. Nightclub Bayerischer Hof. Start time 8 p.m.
System: …

We programmed an interface layer for the dialogue application that corresponds to the API of the web service and transforms the question of the user into a correct WSDL query to the data base. The answer that is sent by the web service usually consists of a long table of all entries, which matched the query. This list must be processed so that it can be well read by the TTS and understood by the driver. We experimented with two different procedures to achieve this task. In the first procedure, the items on the answer list are read one after the other and the user can barge in anywhere with commands like "details", in order to hear more information on this specific item, or "drive me there" in order to enter an intermediate destination into the navigation system. In the second procedure we transformed the web service answer into a table, and then automatically generated a dialogue application from it. This answer application then allows the user to directly ask for items, which other-wise would have been out of vocabulary, e.g.: "Are there any concerts with Alfred Brendel in Munich next week?".

Conclusion

The internet offers a lot of useful information for the driver of a car. Natural language dialogue and speech technology show promise to make this information accessible in a most natural way - just like making a phone call. The developed approaches and technologies to generate speech dialog applications with internet content automatically or semi automatically are very convincing and effective.

This work was partially funded by the German Ministry of Education and Research (BMBF) under grant 01IMD01K.

References

[1] Berton, A., Regel-Brietzmann, P., Block, H.-U., Schachtl, S: Integration of scalable dialogues in Cars. *Proceedings. of ESSV 2006*, Freiberg, Germany (2006),

[2] Beveridge, H.A., Carlyon, R.P.: Effects of aspirin on human psychophysical tuning curves in forward and simultaneous masking. Hear. Res. 99, 110-118

[3] Manley, G. A., Klump, G. M., Köppl, C., Fastl, H., Oeckinghaus, H,: Advances in Hearing Research. World Scientific, London (1995)

[4] Wahlster, W.: SmartWeb - Mobile applications of the semantic web, in: P.Dadam and M. Reichert, editors, GI Jahrestagung 2004, Springer

[5] Yin, T.C.T. and Chan, J.C.K.:) Neural mechanisms underlie interaural time sensitivity to tones and noise. In: W.E. Gall, G.M. Edelman and W.M. Cowans (Eds.), *Auditory Function: Neurobiological Bases of Hearing*. John Wiley, New York, pp. 385-430