HOW TO ACCESS LARGE NAVIGATION DATABASES IN CARS BY SPEECH

André Berton, Sandra Mann and Peter Regel-Brietzmann

DaimlerChrysler Group Research and Advanced Engineering
andre.berton@daimlerchrysler.com

Abstract: Navigation applications are becoming increasingly complex, since databases and features are rapidly growing. Navigation databases include a growing number of points of interest (POI) and a more precise resolution at the street, cross-road and house number level. Human-Machine Interfaces (HMI) such as Speech Dialog Systems (SDS) need to be designed in order to allow comfortable access to data with as little driver distraction as possible. We discuss how the user can find the intended data in such a large database? Current approaches require browsing the hierarchical database structure through categories and sub-categories, such as *restaurant* and *vegetarian*. This paper presents a new interface that utilizes a context-free keyword search, and relies on a hierarchy of categories only for disambiguation purposes. This allows a user who does not completely know the hierarchy to find a destination by simply saying it. The user does not even have to know the precise name of the destination, since the proposed method generates wording variants by decomposing the name and recombining its parts. After presenting the new interaction design and dialog requirements, we derive a dialog system architecture which we implemented to fulfill these requirements. A short pre-evaluation indicates that users prefer the new interaction method to the interaction of the S class series model, because of its increased comfort, ease of use, general appeal, and in particular because of its voice control interface.

1 Introduction

Premium segment cars already offer speech dialog systems as standard accessory to provide assistance in operating audio, phone and navigation devices. Navigation applications are becoming increasingly complex due to their growing databases and increasing number of features. Navigation databases include a growing number of POI and a more precise resolution of street addresses and cross-roads. Some headunits even allow the user to include additional third-party data (POI) to the existing database.

Depending on the market, the number of city, street and POI names can become very large. That makes it difficult to obtain a high quality of Automatic Speech Recognition (ASR) under noisy car conditions. Table 1 summarizes the number of orthographically distinguishable city, street and POI names in order to estimate, which vocabulary sizes ASR systems must potentially cope with.

Market	City names	Street names	POI names
Germany	78,000	414,000	400,000 – 2,000,000
USA	60,000	1,849,000	1,900,000 – 13,300,000

Table 1 – Number of orthographically distinguishable names

State-of-the-art ASR systems in cars can handle a magnitude of around 200,000 vocabulary entries. The current benchmark for SDS in cars is the C class of Mercedes-Benz [1]. Its destination entry interface allows the user to utter the city name in a given country. It verifies the city name by presenting the user with a list of recognition results and asking the user to

enter the line number of the desired city within the list. An additional dialog step is included for ambiguous city names. After selecting the unique city name, the system dynamically enroles the street names for the selected city into the vocabulary much faster than any competitor product. The user is then asked to enter a street name. The maximum number of street names for a given city is about 50,000. Such a vocabulary size can be handled well by state-of-the-art ASR technology. After entering the street name, the user may optionally input a house number or crossroad. A more advanced approach, which asks for the street name directly after entering the city name, incorporates the street name and more information if required to unify the address [2]. However, the next big step, which reduces interaction steps even further and has already been implemented in prototype demonstrators, is the so-called one-shot destination entry, which allows the driver to enter the entire address in a single utterance [3].

POIs can only be selected by category at the moment. The user has to browse through categories, which can be inputted with voice control, but the POI name itself cannot be spoken. It must be selected by uttering the corresponding line number in the currently displayed list. This is due to the fact that the two major map suppliers, Teleatlas and Navteq, provide complete phonetic transcriptions for city and street names, but only very few phonetic transcriptions for POI names. Navteq offers no POI phonetics at all. Teleatlas offers phonetics for 12 out of 80 POI categories [4]. These categories contain airports, ferry terminals, frontier crossings, golf courses, parking areas, petrol stations, post offices, railway stations, rest areas and shopping centres. All these categories contain a limited number of entries which are significantly related to the driving task, but hardly change frequently. Larger categories which change quite dynamically are restaurants, hotels, shops, cash dispensers, doctors and car dealers. POI names belonging to all these categories are not phonetically transcribed yet. Overall, the number of POI names in the navigation database can add up to several millions (see Table 1).

Another major issue is usability: How can the user interact with large databases, if s/he does not know the precise names of the entries? We investigated a first approach for accessing the address book, where the user was allowed to enter either the first or the last name of the entry in question first. Some navigation systems already allow fault-tolerant manual input, such as Route 66 with FuzzySearch [5]. Misspellings are not a major SDS issue, since fault-tolerant spell matchers have been available for a long time. A major challenge for SDS is long POI names, which contain several constituent parts, such as "*Hahn + Lang Automobile GmbH + Co. KG Niederlassung Stuttgart*". Even if the SDS had the phonetic transcription for the complete entry, how would the user address such an entry? It is very likely that the user would not remember the precise name, but instead come up with a shorter name (wording variant), that might recombine constituent parts of the entire name. This paper proposes a decomposition and recombination method that makes several wording variants available to the user for each POI name, so that the user is able to utter wording variants and find the desired POI or address.

This paper addresses the problem of how to handle large and dynamic navigation databases that might not even contain the phonetic transcriptions for their entries. Section 2 discusses a dialog concept of how to access the contents of such large databases, where the user might not even know the precise name of an entry. Section 3 describes the prototype architecture which is needed to implement the specified dialog concept. Section 4 contains some results of a short usability evaluation of the prototype demonstrator. Finally, Section 5 summarizes and proposes future work.

2 A Speech Dialog Concept for accessing data of large navigation databases

Since the user is already used to enter the destination by uttering city and street name, the user will expect the same thing for POI names. At the moment the user can browse the POI database by first entering the category name and then choosing the desired POI name. The POI databases of map suppliers are well-structured as a hierarchy of categories and sub-categories, such as *restaurant* and *vegetarian*. This hierarchy of categories makes it possible to design a search interface which allows the user to browse the hierarchy (basic interaction).

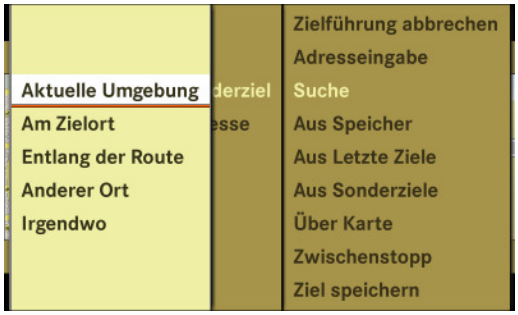
Unfortunately the number of POI categories is quite large. Can we expect the user to be aware of the complete hierarchy of some 80 different categories and some even further structured into sub-categories? Will the user precisely know which category the desired POI is in? A difficult example is, for instance, distinguishing between the two categories: *Rent-A-Car Parking* and *Parking Garage*. Therefore we propose an HMI with a second input functionality particularly for users, who know their desired POI name and do not want to browse through the category names, and for users, who are not sure which category their desired POI belongs to. This input functionality allows the user to directly enter a search string, like in Google, and let the system come up with all matching database entries. The search must be fault-tolerant, since the user might misspell the search string. We dub this input functionality "*POI search.*"

Our proposed HMI includes both: the hierarchical browse and the POI search function. The HMI considers all input and output modalities available in the car. In our case, we offer manual and speech input and graphical and speech output. How do we design the speech input modality for such a POI search? The user should be allowed to directly say the desired POI name. A spell matching input would only be acceptable as a fall-back solution, since the users are already used to entering destination addresses by whole-words.

However, there is a significant difference between entering a search string manually or through voice, since manual input requires for spelling parts of the string, while comfortable speech input allows the user to utter the entire POI name or wording variants of the search string. The following sample speech dialog illustrates our concept for search POI names. It shows the speech input of the user, followed by speech and graphical output:

User: Suche Sonderziel.
(*Find point of interest*)

System: Sonderzielsuche.
In welcher Umgebung?
(*Finding point of interest. Where?)*

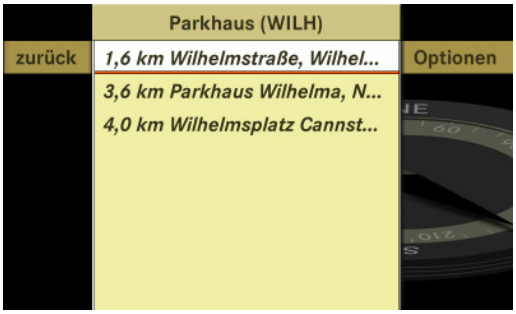


User: Hier.
(*Here.*)

System: In aktueller Umgebung.
Ihr Sonderziel bitte!
(*Local search.Which point of interest?)*

User: Wilhelma.
(*Wilhema*.)
System: Wilhelma.
Welche Kategorie?
(*Wilhelma. Which category*?)

User: Parkhaus.
(*Parking garage*.)
System: Parkhaus. Welches Sonderziel?
(*Parking garage.*
Which point of interest?)



User: Parkhaus Wilhelma.
(*Parking Wilhelma*.)
System: Parkhaus Wilhelma.
Zielführung starten?
(*Start route calculation*?)

User: Ja. (*Yes*.)
System: Die Route wird berechnet. (*The route is being calculated*.)

Figure 1 – Spoken dialog example for searching points of interest

The overall dialog concept is illustrated in Figure 1. It consists of the following features:

1. "*What you see is what you can say*": The user can say anything that is displayed. Therefore all POI names need to be utterable. Phonetic transcriptions are required for all the displayed data.
2. Shortcuts, such as "*find point of interest here*," to reduce the number of interaction steps to a minimum.
3. Direct access to related applications, such as "*start navigation"* or "*call*."
4. Disambiguation: If a keyword is member of several POI categories or if ASR delivers confident results for more than one POI category, the SDS cannot directly present the list of POI names found. A disambiguation of the POI category is advantageous. We propose not to present lists of mixed POI categories.
5. Combined input: The user can input the category and the POI name in one combined utterance.
6. Local restriction of search space: the search space should be locally restricted according to technical possibilities of ASR vocabulary size and disambiguation effort.
7. Partial name input: The user is allowed to utter significant wording variants of names, since the user might not remember the precise name completely.
8. Synonyms from domain ontologies: Synonyms for POI categories should be available. We require more than simple synonym mapping and propose incorporating ontologies

that model domains related to the POI. This allows the user to enter general requests, such as "*I am hungry*" as well as requests for particular POI related items, such as "*I am looking for flowers.*"

How can these features be realized? The first five features are part of current dialog design guidelines that should be included in any modern speech dialog. The last three features, namely the local restriction of the search space, the partial name input, and synonyms from domain ontologies, will be further described in the following sections:

2.1 Local restriction of the search space

A major technical problem at the moment is in-car ASR for large vocabularies. If embedded ASR engines are able to handle a magnitude of 200,000 vocabulary entries, the POI database must be broken down into clusters of that size. This can be achieved by reducing the search space to a particular environment, such as the local environment, the destination environment, the corridor along the route or any other particular area. We also propose to offer such a local restriction in order to limit result space. The larger the environment to be searched, the larger the result space and the greater the effort needed for disambiguation. Suppliers are already developing ASR engines that can handle vocabulary sizes up to a few millions, which will allow global POI search in next generation cars. Such a global search should be offered as additional possibility.

2.2 Synonyms from domain ontologies

A domain-specific ontology models concepts of a typical domain that are related to one another. These concepts have to be mapped to POI categories and sub-categories within the hierarchy. Figure 2 shows a part of the ontology for the food domain. It contains the following entries:

- POI category: *restaurant* can be directly mapped into the POI hierarchy.
- POI sub-category: *Italian*, *Chinese*, *fast food* and *vegetarian* can also be mapped directly into the POI hierarchy.
- Synonyms: *cafe* is effectively a synonym to *restaurant*, since there is no other matching entry in the POI hierarchy. The same holds for *pizza* and *burger*, which are synonyms for *Italian* and *fast food*, respectively.
- Unspecific input: *hungry* is non-specific since it does not define a specific POI category. The SDS should be able to handle such inputs by mapping them into the best matching POI categories.

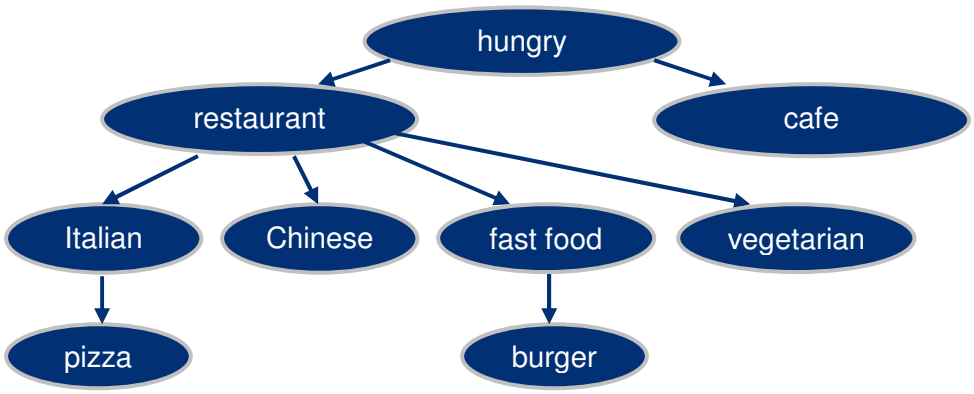


Figure 2 – Sample ontology for points of interest related to food

Using such ontologies allow the POI interaction interface to be modularly updated. The POI hierarchy remains the same, but if new names come up in everyday life, these names can be included in the HMI by updating the ontology, which can be a simple XML file.

2.3 Partial name input

The most complicated issue in searching POI names is modelling what a user would call a given POI. The databases from the suppliers contain a defined name for each POI and rarely alternative names. However, the user might not remember the precise name of a POI in the database, particularly for entries such as "*Südtank Station, Blaubeurer Straße, Ulm, Inhaber Peter Weber.*" We assume that the user might call that petrol station "*Südtank,*" "*Südtank Ulm,*" "*Südtank Blaubeurer Straße,*" or "*Südtank Peter Weber*". How can the SDS handle such user input? We propose a 4-stage filter and recombination method to allow for partial name input [6]:

1. Decomposing the name by filtering keywords: category names and their synonyms might be omitted in speech input as well as trademarks and city names:
    a. *Ristorante* Girasole → Girasole
    b. Mercedes-Benz *Autohaus* Cottbus → Mercedes-Benz Cottbus
    c. Auto Fricker *GmbH* → Auto Fricker
    d. St. Anna-Klinik *Bad Cannstatt* → St. Anna-Klinik
2. Morpho-syntactic analysis: closed word classes, such as articles, prepositions, and adjectives can be omitted in some nominal phrases:
    a. *Zum* Weinhof → Weinhof
    b. Gasthof *der* Zauberlehrling → Zauberlehrling
3. Syntactic-semantic analysis: secondary name constituents can be omitted:
    a. *Gottlieb*-Daimler-Strasse → Daimler-Strasse
    b. *Walter-Erich*-Schäfer-Weg → Schäfer-Weg
    c. Dr. *med dent. Heinz* Flieger → Dr. Flieger
4. Recombination of constituent parts to wording variants:
    a. Held&Ströhle Autohaus *Ulm* → Autohaus Held&Ströhle

This 4-stage method describes the name decomposition and recombination on the orthographic level only. In order to enable voice control, we need to provide the SDS with the corresponding phonetic transcriptions. Therefore we require suppliers to provide phonetic transcriptions for the POI names. These transcriptions include word boundaries so that the phonetic transcriptions can be decomposed and recombined according to their orthographic rendition.

However, users might want to update their POI database with data from third-party suppliers that do not contain phonetic transcriptions. So we have to rely on G2P conversion in order to use voice input. We offer a fall-back spelling mode and voice-enrolments to store the entire name in order to be able to handle phonetic exceptions, which cannot be transcribed correctly automatically. We are considering offering a user dictionary, in which the user can enter phonetically spelled alternatives for a POI name, which can then be correctly transcribed by G2P.

POI names must be available in several languages. At the moment we have 3 languages for North America and 15 for Europe. The SDS activates the phonetic transcriptions for both the target language and the interaction language of the user. The sight "*Piazza di San Marco*" can then be found by a German user by saying either "*Piazza di San Marco*" or "*Markusplatz*".

We implemented decomposition and recombination rules, optimized these rules manually, and tested them on a database for Baden-Württemberg and found that the vocabulary size becomes about three times as large.

3 Prototype architecture

A prototype system was built to verify the new concept of searching navigation destinations (addresses and POIs) by voice. Figure 3 shows the architecture of the prototype system which we derived from common SDS architecture approaches, such as [7]. It has two input and two output modes: Manual and speech input, and graphical and speech output.

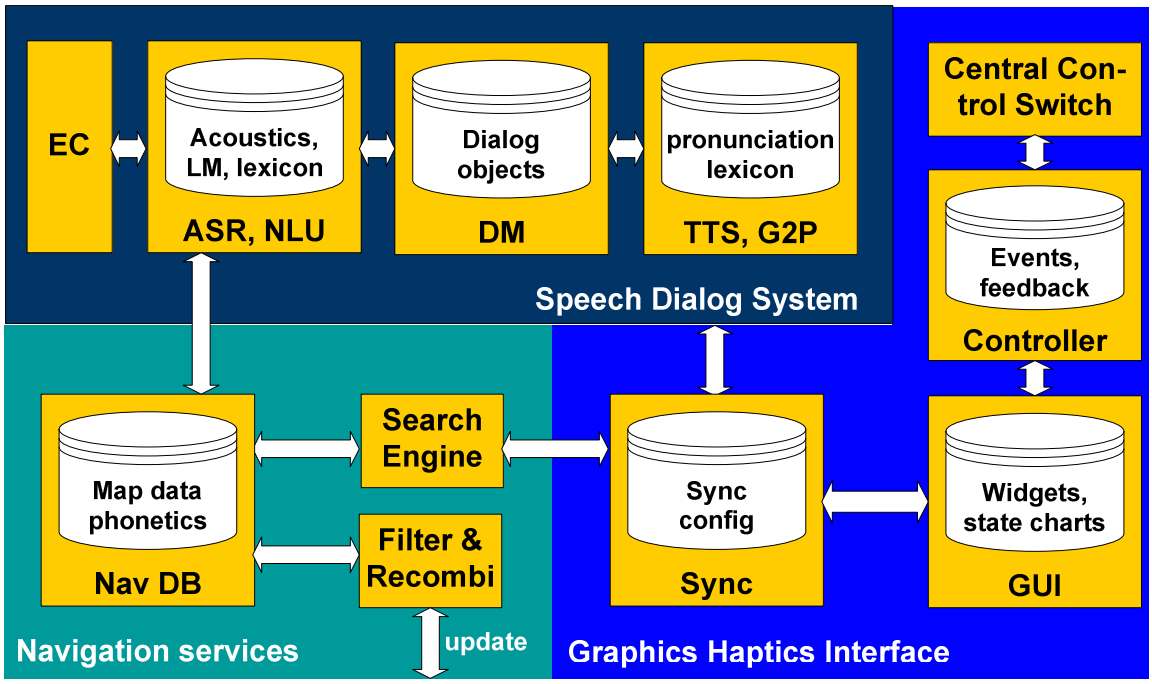


Figure 3 – Prototype system architecture for POI and address search

The SDS consists of a task-oriented dialogue manager (DM), a natural language understanding unit (NLU) containing a contextual interpretation, an automatic speech recognizer (ASR) and a text-to-speech component, which includes a grapheme-to-phoneme (G2P) converter. It also includes echo cancellation (EC) pre-processing.

The implementation of the Graphics-Haptics Interface follows the model-view-controller paradigm. Models (state charts) and views (widgets) are described in the GUI module. The controller module contains the event management and the interface (CAN bus) to the central control switch, which can be pushed, slid and turned. Such a control switch is the typical control element in premium segment cars, such as Audi, BMW and Mercedes-Benz. The synchronization module (Sync) synchronizes the states of the SDS and the Graphics-Haptics Interface using real-time events and data transfer. It also functions as a common application (search) interface for both modes.

The navigation application consists of the navigation search engine, the navigation database, and the Filter-&-Recombination module. The navigation database is a large database containing all map and phonetic data for a certain market, such as Europe or North America. The database must be pre-compiled in order to ensure fast access. The database contains pronunciation variants for each entry (POI name). Among those variants are also wording variants produced by the Filter-&-Recombination module. The search engine contains a quick database search method for manual or voice spelling input. The Filter-&-Recombination module generates the wording variants for new POI and address data (which come in as updates) described in Section 2 and it populates the navigation database. It is connected to the G2P engine which provides missing phonetic transcriptions for new POI names. The ASR module has an efficient access to the database.

4 Evaluation

In a usability evaluation we compared our new search method with the current hierarchy browsing in the Mercedes-Benz S class. Thirty-three subjects were asked to solve nine search tasks with both systems. Twenty-nine out of the 33 subjects of the evaluation preferred the new prototype over the reference system, particularly in terms of comfort, ease of use, fast interaction and general appeal. They generally favoured speech to manual input, since this reduced distraction and interaction time.

5 Conclusion

This paper addresses the problem of how to interact with navigation applications that access large and dynamic navigation databases. First it describes the state-of-the-art method of browsing through the hierarchical category structure of a points-of-interest database. Then it proposes a new method for directly accessing any database entry without having to browse through a hierarchy. This keyword search not only allows users who do not know the complete hierarchy of categories to find their desired destination, but also generates wording variants for destinations, so that users who do not precisely remember the name of the destination can still find it in the database. These wording variants are generated by decomposing the names into their constituent parts and recombining them according to a set of manually optimized rules, allowing for partial name search. This approach is extended to allow database updates with entries which might not have their own phonetic transcriptions. The derived architecture for implementing the proposed method contains target country- and speaker-specific G2P modules to provide phonetic transcriptions in several languages, and the filter and recombination module to provide the system with the wording variants. A short pre-evaluation indicates that users prefer the interaction method of the new system to the one of the S class series model in terms of its comfort, ease of use, general appeal, and in particular due to the voice control interface. In future work we will further investigate cross-lingual interaction and extend our usability evaluation.

Acknowledgements

This work was partially funded by the German Ministry of Education and Research BMBF in the framework of the SmartWeb project [8] under grant 01IMD01K.

Literatur

[1] Mercedes-Benz: SDS reference manual for C class model, Stuttgart, 2007
[2] Berton, A.; Schreiner, O. and Hagen, A.: How to speed up voice-activated destination entry, In Proc. ESSV 2005, Prague, 2005
[3] Hunt, M.: Automatic Identification of Spoken Names and Addresses, Langtech, Berlin 2003
[4] Teleatlas: Multinet Product Documentation CD, Gent, 2007
[5] Route 66: Mobile 7 Betriebsanleitung, Version 1.0, 2006
[6] Berton, A.; Ehrlich, U. and Mann, S.: Aufarbeitung von Aussprachevarianten für Text-Enrolements von Sprachbediensystemen. Patent application filed, 2007
[7] McTear, M.F.: Spoken Dialog Technology – toward the conversational user interface, Springer, London, 2004
[8] Wahlster, W.: SmartWeb – Mobile applications of the semantic web, In P. Dadam and M. Reichert, editors, GI Jahrestagung 2004, Springer 2004