# Marvina – A Norwegian Speech-Centric, Multimodal Visitors' Guide

**Ole Hartvigsen[1], Erik Harborg[2], Tore Amble[1] and Magne Hallstein Johnsen[1]**
[1] Norwegian University of Science and Technology (NTNU), N-7491 Trondheim, Norway
[2] SINTEF ICT, N-7465 Trondheim, Norway
hartvigs@stud.ntnu.no, Erik.Harborg@sintef.no, toreamb@idi.ntnu.no, mhj@iet.ntnu.no

## Abstract

This paper describes the development and testing of a multimodal visitors' guide service for guests to the city and university in Trondheim. The system is under continuous development. At the present state it serves as a help for visitors to Trondheim aiming at meeting people at the university. Using a natural speech interface with a mobile phone it provides help in finding the right bus connection, how to find your way on the campus, and finally how to find the office inside the building where you are going to meet. Information is provided to the user in the form of speech and graphics (maps, illustrations). It is also illustrated how a robot-guide can be used in helping the visitor in finding his way inside a building. Presently, the user end of the demonstrator is implemented on a standard PC, using IP-based telephony (Skype Out). However, in order to utilize all aspects of the system, a practical implementation would require a type of PDA-based phone.

## 1 Introduction

Spoken dialogue information systems over the telephone line have enjoyed growing commercial interest over recent years, and a large number of such systems have been developed and tested, e.g. (Gupta et al., 2006). The systems vary a lot in complexity, not only due to the variation in tasks and design techniques, but also because of the difference in targeted user friendliness.

The introduction of new PDA-based mobile phones has increased the interest in developing phone-based services utilizing additional modalities to speech, e.g. graphics through the enhanced screen on those units, e.g. (Bühler and Minker, 2005). Our work has been performed as a part of the collaborative, multidisciplinary BRAGE-project[1]. One of the main tasks within this project has been the development of spoken dialogue systems. The work presented here, is our first attempt of a multimodal add-on to a speech-only based system. It represents a merge and enhancement of previous work in various areas.

## 2 Human-machine dialogue systems

### 2.1 Human-human versus human-machine dialogues

It is a long way to go before human-machine dialogue systems can emulate real human-human spoken dialogues. When humans talk with each other the spoken dialogues are characterized by spontaneous speech with an "infinite" vocabulary, unfulfilled sentences with incorrect syntax, interrupts, corrections, filled pauses, false starts, repetitions, topic changes, change of dialogue initiative, complex reasoning, and use of "world" knowledge. In addition, a face-to-face dialogue between humans also applies meta-information such as gestures, mimics and voice mode to communicate the meaning of the spoken utterances.

In contrast, current human-machine dialogue systems are limited to a specific task, a finite vocabulary, moderate reasoning, and usually have no knowledge of the "world" outside the task. Further, the systems have only a limited ability to handle

[1] BRAGE homepage: *http://www.iet.ntnu.no/projects/brage/*.

**Joakim Nivre, Heiki-Jaan Kaalep, Kadri Muischnek and Mare Koit (Eds.)**
**NODALIDA 2007 Conference Proceedings, pp. 297–304**

spontaneous speech, interrupts, topic changes, and discourse.

### 2.2 Human-machine dialogue structures

A query system is a degenerate dialogue structure as the input must be a grammatically correct, complete request to which the system can respond in a single turn.

On the other end we find the so-called system driven dialogue. This dialogue is characterized by a predetermined sequence of system-initiated turns; i.e. questions which the user must respond to accordingly. Usually only a single semantic entity is asked for in each turn.

A more human-human like (and thus user-friendly) dialogue structure is used in so called mixed-initiative systems, where both the user and the system can take control of the dialogue flow.

Some tasks are complex to solve even for human-human dialogues. These cases are often termed problem-based dialogues. Developing automatic systems which can handle problem-based dialogues is a huge challenge and a current research topic within the field dialogue theory and formalisms.

### 2.3 Text-based versus speech centric systems

The Internet offers a variety of text-based information systems. Many of these are query based. Further, text-based system driven services (booking, bank etc.) are more or less used by everyone.

Only a few text-based mixed-initiative dialogue systems are public available and rarely any systems deal with tasks which need a problem-based strategy.

In many everyday situations, people need instant information without having access to a PC (keyboard). By enabling access to the information by a phone (or a PDA with GSM, GPRS or UMTS), this problem can be solved. This, however, implies that the users must be allowed to carry out the dialogue by using their voice and also receive the information by speech. A speech only dialogue system is shown in Figure 1, and typically consists of several modules; automatic speech recognition (ASR) including a semantic extractor, text-to-speech synthesis (TTS), a dialogue manager including a reasoning module, a database and a response generator. The last three are needed also for a corresponding text-based system.

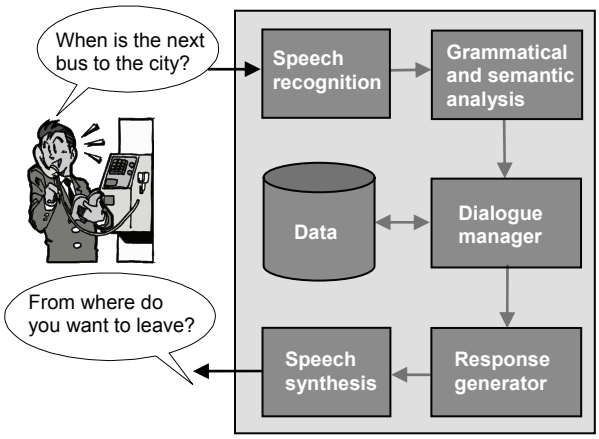


*Figure 1. A speech only based dialogue system.*

Note, however, that the ASR performance in a dialogue context means semantic instead of word accuracy; i.e. the goal is to extract the user's meaning or intention correctly. Further, the success of any automated service is strongly correlated to user friendliness. For dialogue systems this calls for a mixed-initiative control between a user and the system. This implies that the system should accept partial information, corrections, change of (sub)task, etc. Finally, it is mandatory that the user should not be restricted to speak in a "read-text" mode; i.e. natural, spontaneous speech must be accepted.

### 2.4 Speech centric multimodal dialogue systems

The convergence between mobile phones and PDAs is accelerating. Thus the display as a second input-output channel has now become an option. For these terminals we believe that multimodal inputs in the form of "tap and talk" will be useful. (Kvale et al., 2005; Almeida et al., 2002) The tap-option can replace or enhance the speech-option and thus reduce the ASR-complexity. Further, the corresponding outputs will have a richer and more compact form; i.e. a combination of graphics, text and speech. Thus, this type of multimodality will result in a simple and user-friendly interface and thus also opens for solving more complex tasks for the users in a convenient manner.

## 3 The modules of the Marvina system

Marvina is the result of a continuous activity and collaboration between the project partners going on for several years. This has led to the development of several subsystems, which for some parts have been previously reported. In this section we present a walkthrough of the various pieces which now has merged into the Marvina system.

### 3.1 Speech I/O

For speech input, an Automatic Speech Recognition (ASR) system is needed. Typically, a speech detector and a feature extractor forms the front-end of the recogniser. The feature extractor performs a sequence of short time-frequency conversions (typically every 10 msec.) and processes the frequency information in a perceptual way (mimics parts of the ear and brain). The output in the form of a sequence of feature vectors is used by the decoder in order to obtain a recognized text string. In this process, acoustic models, a pronunciation lexicon and a language model is used.

The acoustic models are a moderately sized set of phoneme based HMMs[2], and are thus responsible for the core conversion between the acoustic signal and the linguistic representation. The pronunciations lexicon gives the mapping between a phoneme/HMM sequence and a corresponding word. The language model is tailored to the dialogue task, and defines the allowable utterance (word sequence) structures. Thus the decoder maps the incoming feature sequence to the most likely sequence of phoneme HMMs corresponding to a legal word sequence. In some cases, several (ranked) utterance hypotheses can be produced. Further, the likelihood scores are often used to calculate a confidence score for the (best) proposal. The system uses this confidence score and dialogue knowledge to accept or reject the sentence hypothesis (or parts of it). If so, the system can ask the user to repeat the corresponding rejected utterance.

In our particular design a flexible speech recogniser was implemented based on an improved version of the procedure developed in COST Action 249, Continuous Speech Recognition over the Telephone (Johansen et al., 2000). The recognition engine was the HAPI recogniser (Odell et al., 1999). The Norwegian SpeechDat database (Höge et al., 1999) was used for training. For Norwegian, this database was restricted to 1000 speakers, which was recorded over ISDN-based fixed lines, i.e. the resulting acoustic models are not specifically adapted for mobile phone callers. The acoustic models consist of a context dependent phone set (triphones) with a relatively strong degree of state tying. In addition models for the SpeechDat defined noise labels (man-made and background) were trained. Finally, to cope with Out-Of-Vocabulary (OOV) words, a lexical based filler model was designed from a set of monophones, according to a simplified version of the procedure presented in (Méliani and O'Shaughnessy, 1996). Thus the user is allowed to speak in a natural language.

System prompts are transformed into speech by a text-to-speech system (TTS-synthesis). TTS for Norwegian is an ongoing research topic for one of our partners in another project[3]. For the time being we are using a commercially available TTS engine; RealSpeak from Nuance[4].

### 3.2 The BusTUC system

BusTUC (Amble, 2000) is a text-based question answering system about bus transportation. The NLP module is based on a complex set of rules, and is implemented in Prolog. It is versatile with respect to understanding and answering a variety of alternative formulations requesting the same kind of information. This question-answering system consists of three modules. The bottom module, BusLOG, includes the bus route database, the bus-stop list of names (including mappings from area descriptions to bus stops), and a route analyzer/planner, which finds the shortest/best route between two given bus stops. Bus transfer is handled if there is no direct route. The second module is a general text understanding module (Text Understanding Computer - TUC) which performs a rule based grammatical and semantic parsing. The third and main module integrates TUC and BusLOG, and tailors the system to process a complete inquiry in a single sentence. In fact this is the part

[2] HMMs - Hidden Markov Models are well suited for representing statistical sequences.

[3] The FONEMA project homepage: *http://www.iet.ntnu.no/projects/fonema/*

[4] Information about the Nuance RealSpeak TTS can be found at: *http://www.nuance.com/realspeak/telecom/*

which is called BusTUC, as it is the part specifically made for the question-answering mode. Thus one could call this module for a degenerate dialogue manager, i.e. with only one turn. BusTUC will thus perform a full sentence analysis. However there is no memory as there is no dialogue; i.e. every question is concerning a new, independent inquiry and must contain all the semantic entities necessary to provide an answer from the bus route database. This can be regarded as an extreme variant of a so-called 'User-initiative' system. Thus, BusTUC will typically understand and respond to sentences like:

- *I would like to travel from Ila to Saupstad in about one hour from now.*
- *When is the next bus from the City Centre to Dragvoll?*

The BusTUC system has now been commercialized, and is publicly available as a service from the bus company in the city of Trondheim. A web-service[5] has been operational since 1998 and a SMS-service since 2002. In 2006 about 678.000 inquiries was made on the web, and about 100.000 on SMS. This is a considerable amount for a city of about 160.000 inhabitants. Inquires are logged and used to continuously improve the performance of the system.

### 3.3 The BUSTER system

BUSTER (Johnsen et al., 2003) was originally developed as a text based and mixed initiative version of the inquiry system BusTUC, based on an existing system driven approach (Johnsen et al., 2000). It was designed as a first step towards a mixed initiative spoken dialogue system. Thus the system is made robust with respect to inputs which reflect recognition errors in a corresponding speech based system. To accomplish this, BUSTER gracefully degrades the dialogue towards a system driven approach.

As BUSTER includes BusTUC and thus allows inputs with more than one semantic entity, both BusLOG and TUC were needed. In order to allow mixed-initiative turn-taking, a complex dialogue structure was implemented, based on a dialogue grammar using a slot-filling formalism.

We have added a speech interface to BUSTER, as described in Section 3.1. The size of the vocabulary is about 800 words in this case, where around 700 contain names of bus stops and area description in the city of Trondheim. Based on logs from real use of the text-based system, Wizard-of-Oz (WoZ) experiments and online use of the speech based system, we have enhanced the system to its present state. Also, evaluation reports from the WoZ-callers have been used for this purpose. The system has now been publicly available for about one year.

### 3.4 The DAter system

The text-based Directory Assistance system, DAter is based on the same technology as BUSTER, and illustrates the portability of that design to another domain. It covers all employees at the Norwegian University and other cooperating institutions connected to the same telephone central, summing up to about 5000 names. You may ask questions about the following information:

- Telephone numbers
- Employee names
- Job position
- Street address
- Office location (room number)
- Associated institution
- Email-address

Also for this system a speech interface has been added in order to facilitate usage over the phone. However, the present version is restricted to a subset of the employees for demonstration purposes and due to limitations in the speech recognition system. For this application the vocabulary includes about 350 words, covering all the 250 employees at the dept. of Computer and Information Science at NTNU.

The DAter system was originally developed as a question answering system based on the TUC framework. However, as for BUSTER, it has been extended to deal with dialogue handling.

### 3.5 The dialogue handling system

The BUSTER system is built upon BusTUC using a generic dialogue handling system. The dialogue system described here is common to BUSTER and DAter.

In general terms, the following functions are new when added to a working question answering system:

[5] *http://www.team-trafikk.no/asttweb/bussorakel2.asp*

- *Decomposition:* Dialogue systems make it easier to convey complex information between the participants, because the information can be broken down into smaller components, and conveyed separately.
- *Anaphoric references:* Decomposition makes it natural to refer to earlier elements by anaphoric references of various kinds.
- *Elliptic references:* Incomplete parts of a sentence that is supposed to be supplemented by the use of earlier text.
- *Augmentation*: The user can on his own initiative add more constraints to a previous query, using ellipsis or otherwise.
- *Modification:* The user can modify the last query, using ellipsis or otherwise.
- *Additional information:* The user may need extra information about something that has come up in the dialogue.
- *Confirmation:* The user may need confirmation of something the system has uttered. Either because the output is ambiguous or unclear, or (in a speech environment) because he is uncertain if he has misheard.

The input from the user is broadly classified into 4 groups of "User Speech Acts":

- *Question:* A question. E.g. "when does the bus leave?"
- *New:* A complete non-question sentence. E.g. "I want to go to Lade".
- *Item:* A single item, not a complete sentence. E.g. "NTH", "Lade allé 80".
- *Modifier:* An elliptic utterance, e.g. "to NTH", "from Lade to NTH".

**User dialogue terminals**

The terminals for user speech acts are based on these classes. We defined separate terminals based on where in the dialogue the speech acts occurred, because the speech acts should be treated differently based on this.

**Dialogue grammar**

With these terminals as building elements, the whole dialogue can be modelled by a dialogue grammar. A dialogue grammar is analogous to a sentence grammar, but the nodes in the dialogue grammar are phrases, annotated with their corresponding dialogue terminal types.

```
Dialog    → UserQs, [dialogerror], SystemDialog
UserQs    → UserQ, UserQs | [ ]
UserQ     → [uin], UiqRepl | [uiq], UiqRepl
UiqRepl   → [sant] | Askrefs, Askfors, UiqRepl2
Askrefs   → Askref, Askrefs | [ ]
Askref    → [sqd], SqdRepl
Askfors   → Askfor, Askfors | [ ]
Askfor    → [sqt], UserQs, SqtRepl
UiqRepl2  → [sat], Modify | [sal], Modify |
              [relax], sat, Modify | [saf]
Modify    → [uim], UiqRepl | [ ]
SqdRepl   → [uadi] | [uadm] | [uadn] | [uadq]
SqtRepl   → [uati] | [uatc] | [uatm] |
              [uatn] | [uatg] | [uatf]
```

*Figure 2. Generic Dialogue Grammar.*

Figure 2 presents a flavour of this grammar, which is further explained in (Fledsberg and Bjerkevoll 1999).

**Grammar execution**

The BUSTER grammar is interpreted by a grammar engine. The grammar engine stores the state of the analysis in a stack of frame nodes containing the following information:

- The name of the current non-terminal.
- A focus-structure defining the focus of the dialogue at this point in the dialogue.
- A frame node containing the values of the required slots.
- A list of last mentioned referents.

### 3.6 The Telebuster system

Telebuster is a unified system for BUSTER and DAter. It uses the same generic dialogue handling system that was used for BUSTER and DAter.

The first step was to make the one and same system to handle two different kinds of dialogues, one for buses and one for directory assistance. The next step was to handle both domains integrated in the same dialogue. As the examples will show (see Section 4.1), this is well on its way to be successful.

The Telebuster dialog system resides upon a complete question understanding system. One of the additional functions of the dialog handler is to maintain a context frame which is updated during the dialogue. This context will then contain all possible referents to anaphoric references that are extracted from the users' queries and from essential information of the answers.

Another function is to ask for missing information from the user in order to be able to formulate a meaningful query to its databases.

Implicit in the multi-domain dialogue handling is the decision, what type of question or dialogue-act it is confronted with. For example, for the question:

*"How do I get to Erik Harborg"*,

the system may decide to find and give information about Erik Harborg, and thereby make salient the information of his location. Thereafter, the location is extracted from the answer, and stored in the context frame.

The system may not automatically go on to find a bus route, but on a later question:

*"How do I get there"*,

the location is already available, and the query can be processed as a yet incomplete bus route query. This demands an origin of departure and time constraints, and will be prompted unless given earlier or decided by defaults.

Similarly, a later question :

*"What is his phone"*

is easily resolved because Erik Harborg is the last mentioned person in the dialogue.

The approach shows its generality in a common handling of the language analysis (both English and Norwegian), the same semantic based translation for two languages and two domains and a unified treatise of dialogues.

### 3.7 The Marvin system

Marvin (Hartvigsen, 2006) was the first attempt to create an "Intelligent Helper" system built on top of the previously Telebuster system. The original task was to create a text based natural language user interface for a virtual robot. The robot is a "guide" or "helper" for people visiting a certain floor in a building with an office environment. Marvin used the Telebuster system for semantic and syntactic analyses of input sentences, and for getting information about people and offices, however the bus route part of the system was not used in Marvin. Telebuster also returned a logical representation (TQL[6]) of the sentences, which Marvin used to interpret some ad-hoc requests that Telebuster would normally not understand. The movements of the virtual robot were simulated in real time in a Java program called WSU Khepera Robot Simulator[7]. The Marvin system would also show maps of the floor or surrounding buildings to supplement the textual answers.

## 4 Connecting the pieces – Marvina

### 4.1 A typical scenario

Marvina is a modular demonstrator, making it easy to change/substitute parts of the system and utilize parts of previous demonstrators. It utilizes both the BUSTER and DAter system in order to provide information about bus transportation in the city of Trondheim, and information about employees at the university and cooperating organizations. It uses the Marvin system in order to provide localization information within a building for employees at the university area. A speech interface has been added to this originally text-based system.

A typical scenario for the demonstrator will be a visitor to the city of Trondheim, who is going to meet an employee of the university at his office. The visitor stays at a hotel downtown and has no idea about how to get to the university and to the office of the person he is going to meet. He calls in to the University Visitors' Guide service from his PDA-based phone to get help. The dialogue might proceed as follows:

- *S:* Welcome to the Marvina Visitors' Guide. How may I help you?
- *U:* I am going to visit Tore Amble.
- *S:* Tore Amble's office is in the IT-building, room no. 312. The street address is Sem Sæland Street 9.
- *U:* How do I get there?
- *S:* You will go to Gløshaugen Syd. Where are you travelling from?
- *U:* I am staying at Hotel Augustin and have a meeting at 3 o'clock.
- *<A map of the city pops up on the PDA-screen showing the hotel and bus stop in question>*
  *S:* The bus stop closest to Hotel Augustin is Torget. You will go from Torget to Gløshaugen Syd before 15:00. Bus no. 5 passes Torget at 14:51 and Gløshaugen Syd at 14:56.
- *U:* What is his phone number?
- *S:* Tore Amble has phone no. 73594451.

[6] TQL – TUC Query Language (Amble, 2000).

[7] Khepera info page: *http://carl.cs.wright.edu/reg//ksim/ksim/ksim.html*

- *<The user takes the bus to the Gløshaugen Syd bus stop, and calls in again>*
- *U:* How do I get from Gløshaugen Syd to Tore Amble's office?
- *<A map pops up on the PDA screen with a marked walking route>*
  *S:* Please follow the marked route on the map to the IT-building, and take the stairs to the 3rd floor.
- *<The user finds his way to the 3rd floor of the IT-building, and finds a visitors' guide robot in the hall. He addresses the robot and says:>*
  *U:* Where is Tore Amble's office?
- *S:* Tore Amble's office is in room no. 312. Do you want me to follow you there?
- *U:* Yes, please!
- *<The robot moves down the corridor and stops outside Tore Amble's office>*

Presently, the system is not implemented on a PDA, however a PC-based demonstrator is implemented. Also, the robot is not yet physically implemented, but simulated as a moving avatar on a floor map presented on the PC screen. This simulation is similar to the one used in the Marvin system (see Section 3.7).

### 4.2 Marvina architecture

Marvina runs across three different computers. A Linux server runs the speech recognition and a Windows server runs the text-to-speech synthesizer. Finally, the main Marvina application and Telebuster is running on another Windows system. Telebuster may also look up information about employees from an LDAP[8] database located elsewhere. The structure of the system is shown in Figure 3.

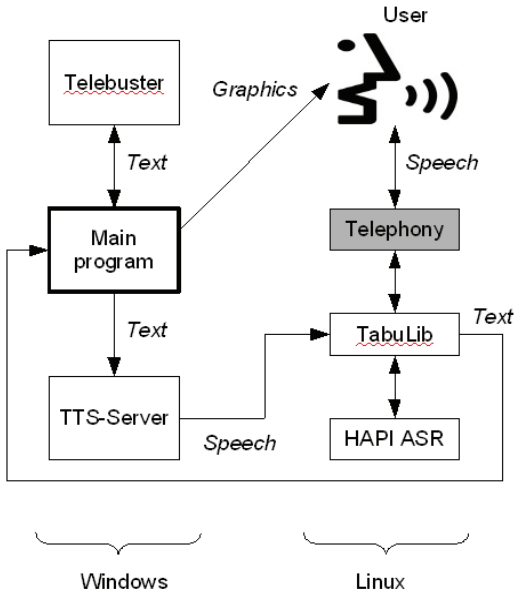


*Figure 3. Marvina architecture.*

The TabuLib program library (Knudsen et al., 2005) controls the ISDN telephony interface and the HAPI speech recognizer. Using this library we avoid low level programming of the I/O system.

The user calls up the speech recognizer via telephone (using Skype). The recognized sentence is sent to the main program as a text string, and the main program decides what to answer in cooperation with Telebuster. An output string is sent to the text-to-speech server, which synthesizes an audible speech output sent back to the user via telephony. The main program also outputs graphics to the user.

### 4.3 Merging the ASR language models

To make use of the complete Telebuster system, the language models for the BUSTER and DAter system had to be merged. The language model has also been extended to cope with special case sentences for the Marvina system. Such sentences include queries like "Where is the toilet?" and "I'm on the third floor of the IT-building". To keep the dictionary size at a reasonable level, names are limited to people working at the Department of Computer and Information Science at NTNU.

### 4.4 Dynamic answers

An important feature in Marvina is the ability to answer dynamically, based on the location of the user and the desired destination. A question about the location of a person's office will be answered in different ways by Marvina depending on the "closeness" of the user. For instance, a user located in a different suburb of the city than the destination will be advised to take certain buses that will bring him closer to the destination. Maps and spoken answers can, for instance, guide the user from his current location to the bus stop, from the bus stop to the desired building, from the building to the correct floor and from the entrance of a floor to the correct office.

[8] LDAP – Lightweight Directory Access protocol.

## 5 Present experience and future work

At present, Marvina works as a fully functional demonstrator. The user may call in from any phone, however, in order to use the graphics capabilities, he must use a PC. The system seems to work quite well within the domain, though extensive user tests have not yet been performed.

Also, the speech interface seems to work quite nicely, however, we are close to a limit for the lexicon size and complexity of the language model in order to maintain acceptable speech recognition rates for the limited training speech data we have available, in particular when mobile phones are used (speech for ASR training is solely based on fixed network recordings). The robust grammar and semantic analysis are helpful in obtaining a graceful degradation when error occurs.

Obvious tasks for the future include performing extensive user tests/evaluation in order to verify/improve the usability of the system. Standard methods for evaluation of spoken dialogue systems have been suggested (Walker et al., 2000). Also, an implementation on a PDA-based phone is foreseen. At the Department of Computer and Information Science (NTNU), there are plans to implement the robot visitors' guide, which would serve as a natural extension of the present demonstrator.

## Acknowledgement

This work has mainly been financed by the Norwegian Research Council as a part of the BRAGE project.

We would like to thank our colleagues at NTNU, SINTEF and Telenor R&D for valuable and fruitful discussions and cooperation with the work within BRAGE.

## References

Luís Almeida et al.: "The MUST guide to Paris - Implementation and expert evaluation of a multimodal tourist guide to Paris," in Proc. ISCA Tutorial and Research Workshop on Multi-Modal Dialogue in Mobile Environments, Kloster Irsee, Germany, 2002.

Tore Amble: "BusTUC – A Natural Language Bus Oracle," in Applied Natural Language Processing Conference, Seattle, USA, April 2000.

Dirk Bühler and Wolfgang Minker: "Mobile Multimodality – Design and Development of the SmartKom Companion," International Journal of Speech Technology, vol, 8, pp. 193-202, 2005.

Øystein Fledsberg and Kim Bjerkevoll: *Buster - Robust dialogue management*, MSc thesis, Norwegian University of Science and Technology (NTNU), December 1999.

Narendra Gupta et al.: "The AT&T Spoken Language Understanding System," IEEE trans. on Audio, Speech, and Language Processing, vol. 14, no. 1, pp. 213-222, January 2006.

Ole Hartvigsen: *Marvin - Intelligent Corridor Guide*, MSc Thesis, dept. of Computer and Information Science, Norwegian University of Science and Technology (NTNU), June 2006.

Harald Höge et al.: "SpeechDat multilingual speech databases for teleservices: Across the finish line," in Proc. EUROSPEECH, Budapest, Hungary, pp. 2699–2702, September 1999.

Finn Tore Johansen et al.: "The COST 249 SpeechDat multilingual reference recogniser," in Proc. LREC, Athens, Greece, pp. 1351–1354, May 2000.

Magne Hallstein Johnsen et al.: "TABOR - A Norwegian Spoken Dialogue System for Bus Travel Information," in Proc. International Conference on Spoken Language Processing (ICSLP), Beijing, China, October 2000.

Magne Hallstein Johnsen et al.: "A Norwegian Spoken Dialogue System for Bus Travel Information", Telektronikk (2), 2003.

Jan Eikeset Knudsen et al.: *Tabulib Reference Manual*. Version 1.5.0, Telenor R&D, February 23, 2005.

Knut Kvale et al.: "Evaluation of a mobile multimodal service for disabled users," in Proc. MMUI, Gothenburg, Sweden, April 2005.

Rachida El Méliani and Douglas O'Shaughnessy: "New efficient fillers for unlimited word recognition and keyword spotting," in Proc. International Conference on Spoken Language Processing (ICSLP), Philadelphia, USA, pp. 590–593, October 1996.

Julian Odell et al.: *The HAPI Book*, Version 1.4, Entropic Ltd., January 1999.

Marilyn Walker et al.: "Developing and Testing General Models of Spoken Dialogue System Performance," in Proc. International Conference on Language Resources and Evaluation (LREC), Athens, Greece, pp. 189–192, May 2000.