



Experiences with Commercial Telephone-based Dialogue Systems

Erfahrungen mit kommerziellen Telefon-Sprachdialogsystemen

Elmar Nöth, Lehrstuhl für Mustererkennung, Universität Erlangen-Nürnberg,
Axel Horndasch, Florian Gallwitz, Jürgen Haas, Sympalog Voice Solutions GmbH

Summary In this publication experiences with commercial spoken dialogue systems are discussed and guidelines for achieving high usability are pointed out. Different from most commercially deployed IVR (Interactive Voice Response) systems, the systems discussed in this paper belong to a new generation of real mixed-initiative spoken dialogue systems, i. e., the user may take the initiative, using full sentences, at virtually any point in time during the dialogue. We use three commercially deployed systems as example applications: the automated switchboard of a large German company, the movie information system operated by Germany's largest multiplex cinema, and a football Bundesliga information system operated by a German media company. ▶▶▶ **Zusammenfassung** In diesem Beitrag werden Erfahrungen mit kommerziellen Sprachdialog-

systemen diskutiert und wesentliche Aspekte herausgearbeitet, die für die Benutzerfreundlichkeit und letztlich für den Erfolg eines solchen Systems entscheidend sind. Anders als bei den meisten im kommerziellen Einsatz befindlichen IVR-Systemen (Interactive Voice Response) handelt es sich bei den in diesem Beitrag beispielhaft diskutierten Systemen um eine neue Generation von echten mixed-initiative Sprachdialogsystemen, d. h. der Benutzer kann praktisch zu jedem Zeitpunkt und durch Verwendung ganzer Sätze die Dialoginitiative übernehmen. Als Beispiele dienen die automatisierte Telefonzentrale eines deutschen Großunternehmens, das Kinoprogramm-Informationssystem des größten deutschen Kino-Komplexes sowie ein von einem Medienkonzern betriebenes Informationssystem für Fußballbundesliga-Ergebnisse.

KEYWORDS I.2.7. [Natural Language Processing] Conversational Dialogue Systems, Interactive Voice Response, IVR, Usability

1 Introduction

Practically all commercially available dialogue systems are either information retrieval systems (the user wants some information like a flight connection from a database) or transaction systems (the user wants some transaction to take place like order a product, talk to a person, or transfer money). The domain of automatic dialogue systems is in most cases very restricted, i. e., one system can often only interact with the user for one application. Negotiation dialogues where the system has to give advice to the user rarely exist.

The interaction with such a system is usually very short (often less than 5 minutes), so the system has very limited possibilities to adapt to the speech of a single user.

Another important property of dialogue systems is whether the system has the chance to hand the dialogue over to a human operator or not. While for a fully automatic system the interaction can fail for good in case of severe misrecognitions, a semi-automatic system is intended to "solve the easy tasks" like answer "frequently asked questions" or to gather necessary information from the user so that the human agent can

shorten his interaction with the user (first level support in a call center). In the case of one or more misrecognitions the user is immediately transferred to the human operator.

In this paper we want to report on our experiences with conversational dialogue systems. In the next section, we will describe three Sympalog¹ systems which will be used to explain important aspects of

¹ The Chair for Pattern Recognition at the University of Erlangen-Nuremberg has done research on conversational dialogue systems since 1978. In 1992 we connected the German demonstrator of the EU funded SUNDIAL project [5] as the worldwide first conversational dialogue system to the pub-

dialogue and system design in Section 3, and we briefly outline the major differences between Sympalog's conversational dialogue technology and today's standard IVR (*Interactive Voice Response*) technology. At the end a conclusion and outlook to future work is provided.

2 System Characteristics

The experiences reported in this paper have been made with various different commercial spoken dialogue systems implemented by Sympalog since 2000. For the pur-

lic telephone. In 2000, "Sympalog" was founded as a spin-off company of the chair and the Verbmobil project [6] (funded by the German Federal Ministry of Education, Science, Research, and Technology, *BMBF*). Sympalog is now a privately owned speech technology and solution company with a focus on conversational spoken dialogue systems.

poses of this paper, we will focus on three applications which we have selected in order to illustrate different aspects of dialogue design, system integration, and user behavior:

The **Sixt switchboard** application, handling all incoming calls (approx. 1000 per day) to the Sixt AG's central phone number, i. e., the number listed in the phone directory. Sixt AG is Germany's largest car rental and leasing company and people call for a variety of different reasons. 90% of the calls are transferred automatically to the correct person, the rest of the calls is handed on to a human operator. The system knows more than 1000 employee names, approx. 350 different notations which refer to one of the Sixt departments, and a large variety of potential reasons why people call the company. In a number of cases, there is a clarification dialogue to

more precisely determine the correct contact person.

Berti, a football Bundesliga information system now commercially operated by a large German media company on a pay-per-call basis. Berti's service number 0900 3100748 is only available from Germany. The system is a significantly advanced version of the system that was recently evaluated in an independent Fraunhofer usability study for commercial IVR systems, where it was one of the two systems that ranked best, outperforming all other systems in the evaluation criteria "navigation/dialogue" and "error management" [4].

Filmtips, the movie information system operated by a Cinema company in the Nuremberg region. Among other cinemas, it includes Germany's largest multiplex and IMAX cinema.

System: *Sixt AG, head office, automatic switchboard operator. Please say the name or department of your contact person or the reason of your call, e.g. "I want to rent a car".*
Okay, how may I help you?

User: *Well, um, I have a question concerning an invoice I received from Sixt yesterday.*

System: *Does this concern a rental car or a leasing car?*

User: *This was about a rental car.*

System: *I guess the customer service for rental cars could help you. Correct?*

User: *Yes, I think so.*

System: *I'll connect you right now...*

System: *Sixt AG, head office, automatic switchboard operator...*

User: *[barging in] I would actually like to speak to Mr. Erich Sixt, please.*

System: *Your contact person is our CEO Mr. Erich Sixt, correct?*

User: *That's right.*

System: *I'll connect you right now...*

Figure 1 Example dialogues for the Sixt switchboard application.

System: *Hello, this is Berti, the football Bundesliga information portal. How may I help you?*

User: *Okay, what's the current score in the Franken Stadium?*

System: *1. FC Nuremberg is leading 1-0 against VfB Stuttgart. Do you need further information?*

User: *Who scored?*

System: *I'm sorry, but currently I do not have any information about scorers...*

User: *[barging in] Well, then, I'd like to know what happened in Dortmund yesterday?*

System: *Borussia Dortmund lost at home to Bayern Munich 3-1...*

User: *[barging in] And at what position are they now in the current table?*

System: *Borussia Dortmund is now in 10th position. Do you ...*

User: *[barging in] And Bayern?*

System: *Bayern Munich is now in 2nd position. ...*

User: *[barging in] And who's first?*

System: *Werder Bremen*

User: *[barging in] Thanks, Berti, bye bye!*

Figure 2 Example dialogue for the Berti football information system.

System: *Hello, this is the Filmtips movie infor...*
User: *[barging in] I would like to see the new Bond movie in Nuremberg.*
System: *You want to see "James Bond — Die Another Day" in Nuremberg. At what time?*
User: *Well, around 8 in the evening would be great.*
System: *Around 8 pm?*
User: *Yes, that's right.*
System: *"James Bond — Die Another Day" is shown in the Cinecitta in Nuremberg today at 8:20 pm. Do you need further information?*
User: *Can I also see it in Erlangen, tomorrow afternoon, let's say around 5 o'clock?*
System: *"James Bond — Die Another Day" is shown in the Manhattan theatre in Erlangen tomorrow at 5:30 pm...*
User: *[barging in] Thanks a lot, bye bye!*

Figure 3 Example dialogue for the Filmtips movie information system.

2.1 Functionality

The functionality of the three systems is most easily shown by example dialogues. They can only cover a small subset of the full functionality of the systems, but they have been selected to illustrate typical user requests and interaction patterns.

Example dialogues for the three systems are given in Figs. 1, 2 and 3.²

2.2 Main differences to other commercial IVR systems

In today's commercial IVR systems, the dialogue structure is typically either implemented using graphical call flow editors, or on the basis of call flow description languages, such as VoiceXML. In complex applications which strongly interact with back-end systems, the dialogue flow is often partly or completely implemented in programming languages such as C++ or Java.

In many cases, application-independent modular units are used to encapsulate individual dialogue steps (e. g., yes/no, time, date). As a consequence, the user is typically forced to provide the information he is prompted for in a way that corresponds to the subgrammar which is associated to that specific dialogue state. Thus, even simple correction utterances such as *no, in Munich* cannot be processed properly, because the word *Munich* is not

part of the application-independent yes/no-subgrammar.

Unlike most dialogue systems on the market, Sympalog systems allow for mixed-initiative dialogues, i. e., an unexperienced user is guided through the dialogue by the system which asks for the necessary pieces of information (like *place of departure, place of arrival*) step by step. An experienced user can formulate his intent freely, provide all the necessary information in one utterance and thus can shorten the interaction time with the system significantly.

Although VoiceXML provides some mechanisms for handling natural language utterances, such as 'form-filling' grammars to implement 'speech shortcuts' in (tree-structured) dialogues, the underlying idea of hard-coding the dialogue on a call flow level appears hardly suited for conversational dialogue systems.

We believe that for the efficient implementation of such systems, a dialogue manager component to determine the optimal system response is absolutely necessary, which also separates the speech recognition task from the speech understanding task. Different from call scripting approaches, such a dialogue manager can also incorporate a large number of generic interaction patterns, and thus allows efficient implementation of mixed-initiative dialogue systems for new applications. Sympalog's dialog manager SymDialog will be

described in some more detail in the following section.

For configuring the speech recognizer, we found that the use of statistical language models is essential for achieving the recognition accuracies required for successful conversational dialogues. We also found that by means of written example utterances the effort for configuring the recognizer for a new application is a fraction of the effort that is required for explicitly coding a context free grammar for the same subset of the natural language. Finally, iterative tuning of a statistical grammar can be performed in a straightforward manner and semi-automatically.

2.3 Dialogue Management

All three applications are conversational, mixed-initiative dialogue systems. The call flow is not explicitly scripted, but determined dynamically at run time. For this purpose, Sympalog's dialogue manager engine SymDialog takes into account the current user utterance, the dialogue history, and the current status or content of the back-end systems.

SymDialog is configured for each application using a GUI (*Graphical User Interface*) or ASCII configuration files. For example, the configuration of the Filmtips movie information system contains information about the slots "time", "date", "movie title", "city", "cinema name", etc. For each of the slots, a number of parameters is defined, e. g., grammar rules

² The three applications have been deployed for the German language; examples for system prompts and user utterances given in this paper have been translated to English.

to extract the information from the word sequence hypothesized by the speech recognizer, prompt wordings, and synonym tables. Furthermore, a number of parameters for defining specific properties of the individual slots can be provided. Among them are a variety of confirmation strategies.

SymDialog uses generic, language-independent knowledge about spoken dialogue interaction as well as application-specific dialogue goals to determine the optimal system prompt for each dialogue step. SymDialog also incorporates a number of mechanisms for coping with the intrinsic uncertainty of the speech recognizer's output. When compared to its ancestor at the University of Erlangen-Nuremberg, the dialogue manager used for the EVAR³ train timetable inquiry system [3] (which itself was a further development of the dialogue manager developed within the SUNDIAL project), SymDialog is far more flexible, configurable, and scalable, far more robust to speech recognizer errors, and by orders of magnitude more efficient in terms of computation time.

If desired, the flexibility of the dialogue can be restricted gradually. Thus, on the far end of the scale, tree-structured, menu-based dialogues represent a subset of all dialogue flows that can be configured using SymDialog.

2.4 Speech Recognition

The three systems presented here are all based on statistical n -gram language models with a vocabulary size of approximately 2000 (Berti football information) to 7000 words (Sixt switchboard). All systems mentioned in this publication use Sympalog's SymRec recognition engine, which is HMM-based⁴, optimized for spontaneous speech phenomena, and which does not require a phonetic dictionary.

³ EVAR: Erkennen – Verstehen – Antworten – Rückfragen (Recognize – Understand – Reply – Ask back)

⁴ HMM: Hidden Markov Model

2.5 Speech Output and Barge-In

All systems presented here support barge-in. Furthermore, the systems are optimized for minimal response time. Under normal acoustic conditions, the users hear the system response approx. 0.6 to 0.9 seconds after they stop speaking, irrespective of the length of their own utterance.

As speech output, all systems mentioned here use pre-recorded sound files which are concatenated dynamically. The Berti football information system adds a muted, continuous background noise that gives the impression that Berti is actually reporting live from a football stadium.

3 Guidelines for Successful Systems

In this section the authors propose five guidelines, which they consider to be crucial, when building and operating telephone-based dialogue systems. All applications described in the previous section were built according to these guidelines.

3.1 Conversational systems will prevail

As already mentioned in Section 1 one possibility to offer automatic services over the telephone is the *menu-driven approach*. Since many people are used to that kind of interaction, because of their experiences with other electronic devices, this approach seems obvious. But designing menus to be used on the telephone is a very difficult if not impossible task: if there are too many alternatives presented in a system prompt, users tend to forget the choices offered at the beginning. Also the navigation within a hierarchical menu of a complex system can frustrate a caller, especially if something goes wrong and you have to go back to the "main menu" [4].

A more promising way to serve customers in an efficient way while at the same time creating high user-acceptance is the *conversational approach*. The prerequisites for such an approach are of course a speech recognition engine that can handle

spontaneous speech and an intelligent dialogue management. Another important point is to tune both modules in terms of shared knowledge (e.g., the last system prompt).

Conversational systems are characterized by human-like features when handling the input of the dialogue partner. The most important ones are mixed-initiative interaction, processing out-of-focus input, and the possibility for the caller to "over-answer" questions.

The evaluation of users interacting with the systems described in Section 2 shows, that callers appreciate this more human-like conversation and quickly adapt their way of communication. This has also been proven in an independent usability test of an air travel reservation system also built by Sympalog [1]. During the test four different tasks had to be performed by persons who in the beginning were not familiar with the application. And although the tasks got more complicated the average time to carry out the different transactions went down significantly. Another notable change was that on the average users answered the system's first question supplying more information the further they got into the test.

This shows that it is also essential that the system is prepared for users with different levels of experience:

- The caller used to a menu-based system who needs to be led throughout the dialogue.
- The new user who has never dealt with an automatic system and needs some help, but intuitively makes use of the features offered by the conversational system.
- The "power-user" who knows the system very well and wants to get the information within a few dialogue steps.

As it was mentioned in Section 2, the status of the back-end system is used for the dialogue management. For example the result of



a database request within the movie information system with “city” set to “Munich” will be rather large so it is necessary to gather more information, e.g., the “time”. However in a small town probably no more than three movies are shown on a certain day and so the system can produce a prompt containing information about all movies at once.

3.2 Building a bridge

Telephone-based dialogue systems can be looked upon as a very advanced middleware between a human and a data source with speech as the interface. The problem when developing an application for a completely new domain is, that even with a lot of research in advance and experience from other systems, it usually takes more than one iteration to build a system that will bridge the gap with the user on one side and the information on the other.

Apart from getting speech data for adapting the recognizer, a thorough evaluation of test cases is necessary to get a feeling for what people actually say. Of course it is possible to recycle atomic knowledge from other domains, for example the way users refer to a point in time (“at four thirty tomorrow afternoon”) or how they describe when their credit card expires (“zero eight, zero six” as opposed to “August two-thousand-six”). But in general it is necessary to go through some iterations before the system can go online. A typical approach, where after each step the dialogue system is optimized in all relevant aspects, is:

- Build an initial system
- Employees test the system (experienced, biased users)
- Friends and family testing (semi-experienced, unbiased users)
- Friendly user test(s) (unexperienced, unbiased, cooperative users)
- Go online (all kinds of users)
- Readjust the system if appropriate

Since the approach to building new conversational, mixed-initiative dialogue systems is user-oriented and not database-oriented as is the case for menu-driven systems, there are new problems we have to cope with during the design phase: because the user is not restricted to say certain things at a certain time, there will be questions or requests from the user that cannot be covered, sometimes simply because the information is not in the database. It is however desirable to react in a sensible way to requests like that. A good example is given in Fig. 2. The system does not have information about scorers, but it is prepared to react to that request.

To improve the performance of an application, it is sometimes necessary to put a lot of work into processing different wordings, that refer to the same value for a slot. Slots which can be especially complex are names. For the Sixt switchboard application (see Fig. 1) for example the database contains the approx. 350 Sixt employees it should connect to with their first and last names. Customers however do not always use the correct names as they are stored in the database. So the system has to know that a “Robert” is also referred to as “Bob” or that “Maier” and “Meyer” are pronounced in the same way. It also makes sense to route calls to the “head of sales” or to “someone in the marketing department”. The consequence is, that the system is able to understand about 1000 alternatives of all the names in the Sixt employee database and about 350 alternatives for the 40 Sixt departments. To solve these problems SymDialog uses flexible conversion techniques and relaxation strategies, if for example a name consisting of first and last name cannot be found in the database.

3.3 Speech output is what the user “sees”

The importance of the text-to-speech component in a dialogue system is often underestimated al-

though it plays a crucial role for user-acceptance. The authors of [4] even ask for a so called “esthetic consistency” with the following definition: “Voice, style, the audience addressed, domain, the company represented, purpose and content of the application – all of that has to go together and form a coherent picture to create customer-confidence”.

A decision that needs to be made for every application is, if pre-recorded sound files or a speech synthesizer should be used. All systems described in Section 2 use pre-recorded sound files. It is desirable to do this for all applications if possible, although there are some obvious problems, for example, if the content of the data source, which is of interest to the user, changes quickly. In that case it makes sense to use an automatic speech synthesizer for the dynamic parts of the system prompt. However it is still preferable to use sound files where appropriate. Callers do accept a mixture of different voices as long as the dynamic content and the basic system prompts are sufficiently separate. In the movie information system, the movie titles are presented by a different voice (not by synthetic speech though).

Another important point is, that there is a trade-off between the requirement that prompts should be short and the fact that a lot of information should be contained in them. A smart way to get around that problem is to make long pauses during prompts, to invite the user to barge in, if he already has made up his mind. Users who are still uncertain will wait and get more information from the system on what they can do. A very simple way to give the user a better idea of what the system is capable of understanding is, to sometimes attach extra information during the dialogue to system prompts: “What can I help you with? [attach:] You can for example ask me *Whats the score in Dortmund right now?*”.

It is also possible to avoid critical situations during a dialogue using proper prompting: Instructing the user very explicitly on how to interact with the system can have a very negative effect, if it is not done politely. For example a prompt telling the user to “speak loud and clearly” will often result in user complaints about the system (“I AM speaking very clearly, you just don’t understand me!”) or even a hang-up. A much better way is to react more subtle, almost human-like in a colloquial way, because this implicit information is easily understood, since people are used to saying or hearing it all the time: “Excuse me, I didn’t hear you?”, “What was that again?”.

To not annoy a caller very quickly by repeating a prompt over and over again it is well known, that there should be alternatives for prompts [2], especially general ones, e. g., prompts presented to the user when the system was not able to process the speech input in a meaningful way.

For some applications it may even be useful (and fun) to create an atmosphere during the call playing continuous sound files, like stadium noise (see Section 2) or to use “earcons” (audio icons) which stand for certain events.

3.4 Discover the optimal dialogue strategy

The examples in Figs. 2 and 3 point out that dialogues within different domains can look very different. Even if non-critical data is provided by the caller, it can be reasonable to get at least an implicit confirmation. That is the case in the movie information system, where up to four slots need to be set so the correct information can be retrieved. For transaction systems, which involve critical data like user names and PINs, explicit confirmation strategies may be inevitable, possibly even juristically necessary. Depending on the number of misunderstandings during a dialogue the system should also apply relaxation strategies con-

cerning the confirmations. If an application is very complex and dialogues tend to get rather lengthy, it is helpful to restrict the flexibility of the dialogue gradually. For example in the air travel reservation system, the dialogue manager does not completely adhere to the concept of “say what you want at any time you want to”: after the caller has chosen a flight, the first goal of the dialogue has been reached. In a second part, the user provides his personal data, including his billing address. Switching between the two parts of the dialogue is still possible with so called meta-utterances (“I want to change the date.”).

During the first iterations of application development, it is often the case, that dialogue strategies need to be more robust due to the sub-optimal accuracy of the speech recognizer. During the later stages of the process, when enough speech data has been collected for the adaptation of the recognizer, the task becomes a lot easier in terms of dialogue management and less restrictive dialogue strategies can be applied.

3.5 Speed is crucial

It is essential that a spoken dialogue system reacts fast to any kind of user input, no matter how long the user utterance is. We found that a delay of two seconds or more after the end of the user utterance is likely to confuse the user, because this often misleads the users to repeat or to reformulate their utterance. This desynchronizes the system-user interaction and may lead to a complete failure of the dialogue.

With today’s speech recognition engines and hardware, faster than real-time performance of the recognizer can easily be achieved, even if 30 or more lines in parallel and vocabulary sizes of several thousand words are processed on a single PC. In order to save hardware costs and, at the same time, to ensure optimal system response times, Sympalog’s recognition engine SymRec has a feature that adjusts the computa-

tional effort to the current system load, ensuring that the recognition result is always available within significantly less than 0.1 seconds after the turn end was detected. This may result in a minor decline in recognition accuracy under high system load, e. g., if all 60 speakers in a 60 line system speak at the same time, which is a highly improbable situation. This decline in recognition accuracy, however, is far less likely to confuse or annoy the user than a noticeable delay.

For the same reason, fast and reliable turn end detection is of extreme importance. Sympalog uses an algorithm which also incorporates prosodic information for this purpose. Only after the turn end has been detected, a semantic and pragmatic interpretation of the recognition result can be performed. The dialogue engine then creates a new system prompt, also taking into account the content of the back-end database (or, if possible and necessary, a cached version of the database). The last step is to generate the speech signal, which is then sent to the caller. The whole course of action after the recognizer has produced its result needs to be finished within a very short time, preferably less than 0.15 seconds.

Fast and reliable turn begin detection is equally important for the barge-in feature, and a tight coupling of the turn begin detection with the speech output is required to immediately turn off the system voice in case the user takes the turn.

VoIP protocols such as H.323 or SIP (*Session Initiation Protocol*) may have a disturbing latency, e. g., if the jitter buffer size is chosen too large. In our experience, any latency has a negative impact on the usability of spoken dialogue systems.

4 Conclusion

In this paper we reported on experiences made during the design, implementation, hosting, and support of various commercial conversational dialogue systems. The

systems we described are based on a commercial dialogue manager module which can be effectively adapted to new applications via a GUI or configuration files. The systems discussed allow for typical human dialogue behavior like “over-answering”. Current and future work include the integration of an “emotion module” which detects user frustration and anger based on acoustic and linguistic information. Thus the system can react appropriately in cases of misunderstanding and make yet another step towards a more human interaction with automatic dialogue systems.

References

- [1] A. Batliner, C. Hacker, S. Steidl, E. Nöth, and J. Haas. User States, User Strategies and System Performance: How to Match the One with the Other. In *Proc. of an ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems*, 2003.
- [2] W. Eckert. *Gesprochener Mensch-Maschine-Dialog*. PhD thesis, Universität Erlangen-Nürnberg, Germany, 1995.
- [3] F. Gallwitz, M. Aretoulaki, M. Boros, J. Haas, S. Harbeck, R. Huber, H. Niemann, and E. Nöth. The Erlangen Spoken Dialogue System EVAR: A State-of-the-art Information Retrieval System. In *Proc. of the 1998 Int'l. Symp. on Spoken Dialogue (ISSD 98)*, pp. 19–26, Sydney, Australia, 1998.
- [4] M. Peissner, J. Biesterfeldt, and F. Heidmann. Akzeptanz und Usability von Sprachapplikationen in Deutschland. Technical report, Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO), Stuttgart, Germany, 2004.
- [5] A. Simpson and N. Fraser. Black Box and Glass Box Evaluation of the SUNDIAL System. In *Proc. European Conf. on Speech Communication and Technology*, vol. 2, pp. 1423–1426, Berlin, Germany, 1993.
- [6] W. Wahlster, editor. *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer, 2000.



1



2



3



4

The authors work for the German speech technology and solutions company Sympalog (www.sympalog.com) and for the Chair for Pattern Recognition at the University of Erlangen (www5.informatik.uni-erlangen.de),

respectively. Sympalog is a spin-off company of the Verbmobil Project and the University of Erlangen-Nuremberg. The company is specialized on leading-edge, conversational spoken dialogue systems. The authors are happy to receive questions or comments regarding this publication directly.

1 Dr.-Ing. Elmar Nöth

Address: Lehrstuhl für Mustererkennung, Universität Erlangen-Nürnberg, Martensstr. 3, 91058 Erlangen, Germany, Tel.: +49-9131-8527888, Fax: +49-9131-303811, E-Mail: noeth@informatik.uni-erlangen.de

2 Dipl.-Inf. Axel Horndasch

Address: Sympalog Voice Solutions GmbH, Karl-Zucker-Str. 10, 91052 Erlangen, Germany, Tel.: +49-61661-0, Fax: +49-61661-20, E-Mail: horndasch@sympalog.de

3 Dr.-Ing. Florian Gallwitz

Address: Sympalog Voice Solutions GmbH, Karl-Zucker-Str. 10, 91052 Erlangen, Germany, Tel.: +49-61661-0, Fax: +49-61661-20, E-Mail: gallwitz@sympalog.de

4 Dr.-Ing. Jürgen Haas

Address: Sympalog Voice Solutions GmbH, Karl-Zucker-Str. 10, 91052 Erlangen, Germany, Tel.: +49-61661-0, Fax: +49-61661-20, E-Mail: haas@sympalog.de