Computing and Informatics, Vol. 26, 2007, 577–603

VOICE OPERATED INFORMATION SYSTEM IN SLOVAK

Jozef Juhár, Anton Čižmár

Department of Electronics and Multimedia Communications
Technical University of Košice
Košice, Slovakia
e-mail: {Jozef.Juhar, Anton.Cizmar}@tuke.sk

Milan Rusko, Marián Trnka

Institute of Informatics, Slovak Academy of Sciences
Bratislava, Slovakia
e-mail: {Milan.Rusko, Trnka}@savba.sk

Gregor Rozinaj

Department of Telecommunications
Slovak University of Technology
Bratislava, Slovakia
e-mail: Gregor.Rozinaj@stuba.sk

Roman Jarina

Department of Telecommunications
University of Žilina
Žilina, Slovakia
e-mail: jarina@fel.uniza.sk

Manuscript received 16 November 2006; revised 14 June 2007
Communicated by Petr Polák

Abstract. Speech communication interfaces (SCI) are nowadays widely used in several domains. Automated spoken language human-computer interaction can replace human-human interaction if needed. Automatic speech recognition (ASR), a key technology of SCI, has been extensively studied during the past few decades. Most of present systems are based on statistical modeling, both at the acoustic and linguistic levels. Increased attention has been paid to speech recognition in adverse conditions recently, since noise-resistance has become one of the major bottlenecks for practical use of speech recognizers. Although many techniques have been developed, many challenges still have to be overcome before the ultimate goal – creating machines capable of communicating with humans naturally – can be achieved. In this paper we describe the research and development of the first Slovak spoken language dialogue system. The dialogue system is based on the DARPA Communicator architecture. The proposed system consists of the Galaxy hub and telephony, automatic speech recognition, text-to-speech, backend, transport and VoiceXML dialogue management modules. The SCI enables multi-user interaction in the Slovak language. Functionality of the SLDS is demonstrated and tested via two pilot applications, "Weather forecast for Slovakia" and "Timetable of Slovak Railways". The required information is retrieved from Internet resources in multi-user mode through PSTN, ISDN, GSM and/or VoIP network.

Keywords: Information system, dialogue system, Galaxy, VoiceXML, MobilDat, speech recognition, speech synthesis

1 INTRODUCTION

Due to the progress in the technology of speech recognition and understanding, the spoken language dialogue systems (SLDS) have emerged as a practical alternative for the conversational computer interface. They are more effective than the Interactive Voice Response (IVR) systems since they allow for more free and natural interaction and can be combined with the input modalities and visual output.

The above statement is true for many languages around the world, not just for Slovak. In this paper we describe the development of the first SLDS which is able to interact in the Slovak language. The system has been developed in the period from July 2003 to June 2006 and is supported by the National Program for R & D "Building of the information society". The main goal of the project is in the research and development of a SLDS for information retrieval using voice interaction between humans and computers. The SLDS has to enable multi-user interaction in the Slovak language through telecommunication networks to find information distributed in computer data networks such as the Internet. The SLDS will also be a tool for starting research in the area of native language technologies in Slovakia.

Contractors of the project are the Ministry of Education of the Slovak Republic and the Technical University of Košice. Collaborating organizations are the Institute of Informatics, the Slovak Academy of Sciences Bratislava, the Slovak University of Technology in Bratislava and the University of Žilina.

The choice of the solution emerged from the state-of-the-art in the topic, the experiences of the partners involved in the project, and from availability of free resources. As described further, the solution is based on the DARPA Communicator architecture based on the Galaxy hub, a software router developed by the Spoken Language Systems group at MIT [1], subsequently released as an open source package in collaboration with the MITRE Corporation, and now available on SourceForge [2]. The proposed system consists of the Galaxy hub and six modules (servers). Functionality of the SDS is demonstrated and tested via two pilot applications – "Weather forecast for Slovakia" and "Timetable of Slovak Railways" retrieving the required information from internet resources in multi-user mode through telephone.

The paper is organized as follows. The second section gives a brief overview of the system architecture. Automatic speech recognition (ASR) block is described in Section 3. A great effort was directed to Slovak speech database development. The speech database which was used for ASR training, with its extensive evaluation, is introduced in Sections 4 and 5. Text-to-speech (TTS) synthesis, dialog manager, i/o audio interface, and backend information modules are described in Sections 6, 7, 8 and 9, respectively. Section 11 addresses two pilot applications. Overall system performance was evaluated from a user point of view. This user-centered subjective evaluation is summarized in Section 12.

2 SYSTEM ARCHITECTURE

The architecture of the developed system is based on the DARPA Communicator [1, 2]. The DARPA Communicator systems use a 'hub-and-spoke' architecture: each module seeks services from and provides services to the other modules by communicating with them through the central software router, the Galaxy hub. Java along with C and C++ are supported in the API of the Galaxy hub. Substantial development based on the Communicator architecture has been already undertaken at Carnegie Mellon University and at the University of Colorado [3, 4].

Our system consists of the hub and of six system modules: telephony module (Audio server), automatic speech recognition (ASR) module, text-to-speech (TTS) module, transport module, backend module (information server) and of dialogue management module. The communication among the dialogue manager, the Galaxy hub, and the other system modules is represented schematically in Figure 1.

The telephony module connects the whole system to a telecommunication network. It opens and closes telephone calls and transmits the speech data through the Broker Channel to/from the ASR/TTS modules. The automatic speech recognition server performs the conversion of incoming speech to a corresponding text. Con-

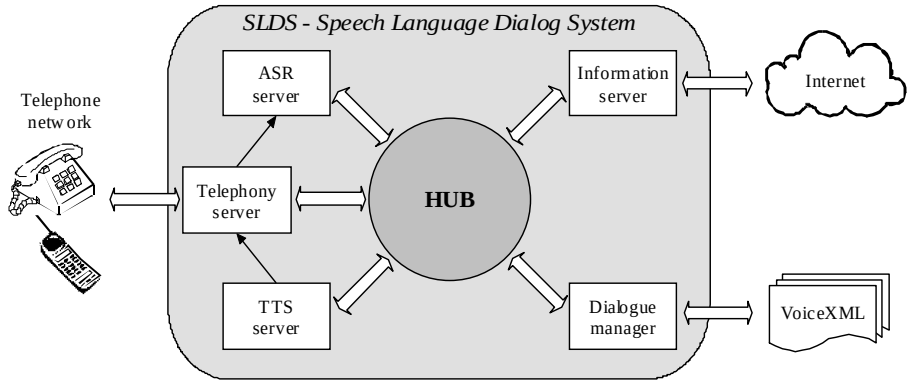

Fig. 1. The architecture of the Galaxy/VoiceXML based spoken Slovak dialogue system

text dependent HMM acoustic models trained on SpeechDat-Sk and MobilDat-Sk speech databases and ATK/HTK and Sphinx IV based speech recognition engines were used in this task. The dialogue manager controls the dialogue of the system with the user and performs other specified tasks. The heart of the dialogue manager is the interpreter of VoiceXML mark-up language. The information server connects the system to information sources and retrieves information required by the user. The server of text-to-speech (TTS) synthesis converts outgoing information in text form to speech, which is more user friendly.

We have designed the system to support "Windows-only" as well as mixed Windows/Linux platform solutions. In the second case a Transport Server managing file transmissions between platforms, is active.

3 AUTOMATIC SPEECH RECOGNITION SERVER

Several speech recognizers freely available for nonprofit research purposes were checked for their reliability and speed. We have adapted two well-known speech recognizers as the ASR module for our system. The first is ATK [5], the on-line version of HTK [6]. The ATK based ASR module was adapted for our SDS running on a Windows-only platform and on a mixed Windows/Linux platform as well. In the second case the ASR module runs on a separate PC with Linux OS. The second speech recognizer, which we have adapted for our system is Sphinx-4 written in Java [7, 8]. Our preliminary experiments have shown that both of these ASR modules give similar results.

SpeechDat-E SK [9] and MobildDat-SK [10] databases were used for training HMMs. Context dependent (triphone) acoustic models were trained in a training procedure compatible with "REFREC" [11]. Dynamic speech recognition grammars and lexicons are used in the speech recognizers.

4 MOBILDAT-SK SPEECH DATABASE

The MobilDat-SK is a new speech database containing recordings of 1 100 speakers recorded over a mobile (GSM) telephone network. It serves as an extension to the SpeechDat-E Slovak database and so it was designed to follow the SpeechDat specification [12] as closely as possible. It is balanced according to age, regional accent, and sex of the speakers. Every speaker pronounced 50 files (either prompted or spontaneous) containing numbers, names, dates, money amounts, embedded command words, geographical names, phonetically balanced words, phonetically balanced sentences, Yes/No answers and one longer non-mandatory spontaneous utterance.

Every speaker called only once; from one acoustic environment. The required minimum number of calls from different environments was specified as 10 % of the whole database for each environment. The home, office, public building, street and vehicle acoustic environments were chosen.

We decided to use the database content adopted from SpeechDat-E database; however, according to assumed practical applications some new items were added:

O4 – sentence expressing a query on departure or arrival of a train, including names of two train stations from a set of 500.

O6 – name of the town or tourist area from a set of 500 names.

O9 – web domain or e-mail address from a set of 150 web domains and 150 e-mail addresses

R1 – One non-mandatory item - a longer spontaneous utterance was added at the end of the recording. The caller had to answer a simple question randomly chosen from a set of 25 such as: "How do you get from your house to the post-office?". This item should considerably extend the spontaneous content of the database. Some trimmings were made in the definition in comparison to the SpeechDat-E.

We decided to let out one item, namely

O5 – most frequent company/agency.

The number of companies is too big to be covered reasonably; moreover, many companies included in SpeechDat-E SK changed their names or do not exist any more. Therefore the use of this item seemed to be questionable.

4.1 Annotation and Transcription

The transcription used in this database is an orthographic lexical transcription with a few details included that represent audible acoustic events (speech and non-speech ones) presented in the corresponding waveform files. The transcription is intended to be a rough guide that users of the database can further examine for details.

The transcriptions were made using our Label 2.0 transcription tool (see Figure 2). LABEL 2.0 performs various statistical analyses of the actual corpus.

The actual count of annotated male and female speakers, people from each region, from age groups and other statistical values such as phonetic (phoneme) coverage can be calculated.

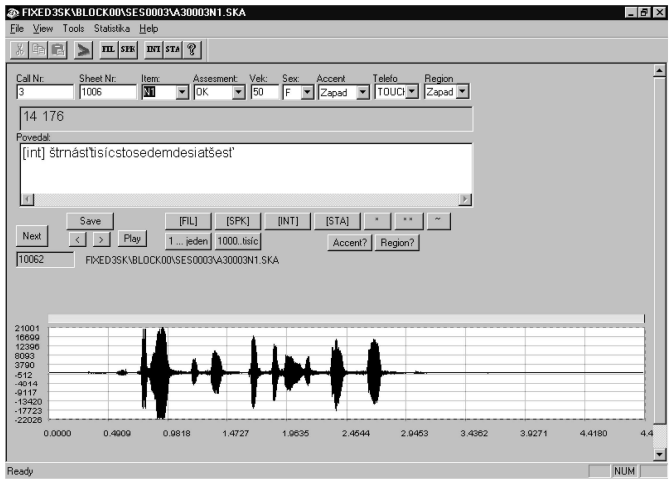

Fig. 2. A typical display of our Label 2.0 annotation tool

5 TESTS ON SUITABILITY OF THE AVAILABLE SPEECH DATABASES FOR TRAINING ACOUSTIC MODELS

Quality and accuracy of a speech recognition system based on HMM strongly depends on the amount of the speech material used for training acoustic models. Another important aspect is represented by environment conditions during recording which have to be similar to the conditions assumed in the real use of speech recognition system. Building such a speech database is time consuming and expensive.

First professional Slovak telephone speech database was built in 1999 within the SpeechDat-E initiative. The second database, called MobilDat-SK, was created within our research project in 2005.

A comparison of both databases was made in the sense of their application in speech recognition systems. The HMM based speech recognition system employed for experiments was using context independent and context dependent phoneme acoustic models.

Basic characteristics and differences of both databases are presented in 5.1. Training and testing processes are described in subsections 5.2 and 5.3. Special cross-test was performed in order to identify differences between both databases in the sense of acoustic space coverage. Results of experiments are presented in subsection 5.4.

5.1 Differences Between Databases

The above-mentioned databases are almost identical in terms of overall structure and number of speakers. Format and structure of both databases were adopted from the SpeechDat conventions [12]. 1 000 speakers (in SpeechDat) and 1 100 speakers (in MobilDat) were recorded using telephone interface. SpeechDat's recordings were collected in public switched telephone network (PSTN) accepting phone calls made from home or office with fairish signal to noise ratio. The MobilDat database contains recordings made using cellular phone in various acoustic conditions like in the street, in a public building or in a car, which results in lower signal to noise ratio.

The databases contain dictionaries with phonetic transcription of all recorded words. Phonetic transcription was made using Slovak SAMPA alphabet [13]. Although both dictionaries contain pronunciation alternatives, SpeechDat's dictionary contains alternatives only in two cases in contrast to MobilDat with more than thousand alternative spellings.

5.2 Acoustic Models Training

Phoneme acoustic models were built following REFREC 0.96 training procedure [10] which employs HTK tools [6] and is adapted to the structure of SpeechDat database.

The acoustic features were conventional 39-dimensional MFCCs, including energy and first and second order deltas. Standard 3-state left-to-right HMMs with no skip transitions for every phoneme were used. Distribution of speech parameters in acoustic space was modeled by continuous Gaussian mixtures, re-estimated up to 32 components and in special cases up to 1024.

The training process was started with "flat start" boot-strapping procedure with context independent models as product. In further phase of training word-internal context dependent phonetic HMMs (triphones) were created.

HMM parameters of context dependent models were shared for acoustically and phonetically similar contexts of a phoneme. This was done by tree-based clustering algorithm using broad classification of phonemes and allophones. Parameter sharing (tying) is efficient for rare phonemes represented by small amount of speech material. Clustering was applied on individual states in each HMM which led to a smaller number of parameters to store. The procedures are described in detail in [13, 14].

Each database is split into train and test sets. Test set represents 20 % of all speakers. The speech files with undesired or corrupted content were removed during the training. Important statistics of training process for SpeechDat-E SK (SD) and MobilDat-SK (MD) is shown in Table 1. FullDat (FD) is the working nomenclature for the database created by joining both SpeechDat-E SK and MobilDat-SK.

5.3 Test Setup

Speech recognition performance was evaluated on six different test scenarios, as specified in [10]. Recordings with following corpus codes were used:

A – applications words (< 50),

I – isolated digits (10),

Q – answers to yes/no questions (2),

O – proper nouns ($> 1\,000$),

W – phonetically rich words ($> 1\,000$),

BC – sequence of digits – connected word loop recognition (10).

The number in brackets represents vocabulary size. Complete testing statistics are shown in Table 2. The simplest test set had only 2 words in the vocabulary, the most difficult one had 2 586 words.

Number of	SD	MD	FD
Sessions (speakers)	800	880	1 680
Training utterances	32 855	36 742	69 588
Lexicon words	14 907	15 942	19 313
Lexicon pronunciations	14 909	17 786	27 954
Triphones in training set	9 121	10 466	11 755
Triphones in lexicon	9 417	10 915	12 182
States before clustering	27 390	31 422	35 289
States after clustering	3 666	4 088	5 701
Clustering reduction	13.4 %	13 %	16.2 %

Table 1. Training statistics for SpeechDat-E SK, MobilDat-SK and FullDat

Test	vocabulary size		average word length in phonemes		number of tested recordings	
	SD	MD	SD	MD	SD	MD
A	30	30	7.47	7.47	1 166	1 295
I	10	10	4.10	4.10	186	216
Q	2	2	2.50	2.50	340	300
O	1 794	1 296	9.57	9.25	1 152	1 063
W	2 586	2 311	7.13	9.02	784	850
BC	10	10	4.10	4.10	788	899

Table 2. Test statistics

As can be seen in Table 2, vocabulary differences were present only in O and W tests. Vocabulary was generated from all words which belong to category O or W in particular database.

The Word Error Rate (WER) parameter was used for tests evaluation and is defined as:

$$\mathrm{WER} = \left(1 - \frac{N - D - S - I}{N}\right) \cdot 100\,\% \tag{1}$$

where N is the number of all tested words, D is the number of deleted words, S is the number of substituted words, and I the number of inserted words.

5.4 Experimental Results

5.4.1 Standard Tests

In the first phase of experiments both databases were tested individually. Two sets of acoustic models were created on each database and used for standard test procedure.

Tests were performed for a number of HMM sets which were created during the training. As training is an iterative process, HMM sets were marked based on the stage of training as follows:

mini – context independent models trained using flat start,

mono – context independent models created using training with estimated duration information obtained by forced alignment on "mini" models,

tri – context dependent models created from "mono" models which were trained in different contexts,

tied – context dependent models with tied (shared) states.

Each name of HMM set contains a number (power of two) which indicates Gaussian mixtures count.

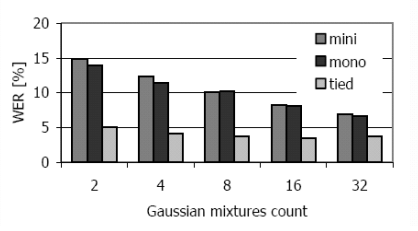

Fig. 3. WER for different HMM sets

Progress of recognition performance based on training stage for SpeechDat test is shown in Figure 3. As can be seen from the figure, WER for context independent models (mini and mono) is higher than for context dependent models with tied states, but it is more decreasing with increasing mixtures count. We took this fact into account and increased mixtures count up to 1 024. Results for this scenario are presented in subsection 5.3.

Context dependent models with 16 Gaussian mixtures and shared states (tied_16) were used in the comparison of the acoustic models sets because they gave the best results. The results are presented in Table 3.

Database training set/test set	WER (Word Error Rate) [%]						
	A	I	Q	O	W	BC	avg.
SD/SD	0.43	0.54	0.00	8.16	10.46	1.32	3.49
MD/MD	1.85	0.93	0.00	9.04	9.32	3.36	4.08

Table 3. Speech recognition performance

5.4.2 Cross Tests

In this setup acoustic models were trained on one database and used for recognition tests with the recordings from another database [15]. The main objective was to identify the differences between both databases relevant for acoustic modelling. Considering the fact that MobilDat was recorded in GSM environment which uses different speech coding scheme and is noisier, we wanted to know how big impact it will have. The results for models tied_16 are presented in Table 4. The first test (SD/MD) represents the situation when acoustic models trained on SpeechDat (SD) were used for recognition of MobilDat (MD) recordings, and the second test (MD/SD) represents the opposite situation.

Database training set/test set	WER (Word Error Rate) [%]						
	A	I	Q	O	W	BC	avg.
SD/MD	1.62	4.17	0.33	15.20	22.58	5.19	8.18
MD/SD	1.29	2.15	0.00	11.63	21.84	2.05	6.49

Table 4. Cross test results

As can be seen from Table 4, WER is nearly doubled compared to Table 3. We got better results with acoustic models trained on MobilDat. Possible reason for better acoustic space covering by MobilDat acoustic models is the variety of recording environments compared to almost "clean" recordings of SpeechDat. The results show that for calls coming from noisy environment acoustic models are not trained sufficiently in SpeechDat-E. Similar situation occurs when MobilDat acoustic models are used and calls are coming from PSTN. In these cases WER increases significantly.

5.4.3 FullDat Test

Although acoustic models were trained on both training sets, tests were performed separately for both test sets. Results obtained with FullDat's (FD) acoustic models for SpeechDat (SD) and MobilDat (MD) test sets are presented in Table 5.

By joining both databases we can avoid problems with calls coming from different telephone environments. Created acoustic models are universal and give comparable speech recognition performance to that of acoustic models trained and tested individually on SpeechDat or MobilDat.

Database training set/test set	WER (Word Error Rate) [%]						
	A	I	Q	O	W	BC	avg.
FD/SD	0.69	0.54	0.00	6.68	11.10	1.44	3.41
FD/MD	1.31	1.39	0.00	9.03	9.18	3.61	4.09

Table 5. FullDat results

As mentioned before, we decided to increase the number of mixtures for context independent models up to 1 024. This experiment was performed during the FullDat training and was applied to "mono" models. Results for both test sets are summarized in Tables 6 and 7.

mix. count	WER (Word Error Rate) [%]						
	A	I	Q	O	W	BC	avg.
32	3.60	0.00	0.00	16.49	21.05	3.77	7.49
64	2.92	0.54	0.29	13.63	18.37	3.39	6.52
128	1.80	0.00	0.00	11.63	15.82	2.88	5.36
256	1.63	0.00	0.00	10.42	15.43	2.69	5.03
512	0.60	0.00	0.00	9.46	16.96	2.28	4.88
1 024	0.60	0.00	0.00	9.81	22.19	2.25	5.80

Table 6. Increasing number of mixtures – SpeechDat test

mix. count	WER (Word Error Rate) [%]						
	A	I	Q	O	W	BC	avg.
32	3.86	2.78	0.67	19	16.14	7.64	8.35
64	3.47	1.85	0.67	18.34	15.18	7.22	7.79
128	2.78	1.39	0.67	17.03	14.02	6.71	7.1
256	2.32	1.39	0.33	14.77	13.78	6.37	6.49
512	2.47	1.39	0.33	13.64	13.53	6.33	6.28
1 024	2.32	1.85	0.33	13.36	14.71	6.17	6.45

Table 7. Increasing number of mixtures – MobilDat test

The results show that the increase in the number of Gaussians in mixtures for mono models leads to the WER decrease and speech recognition performance is approaching that of context dependent models. This progress can be seen up to number 512. The main disadvantage is high time consumption of training procedure of the models with higher mixture count. Recognition using such models requires also higher computational capacity. For example standard W test ran 17 minutes for tied_16 models and 62 minutes for mono_512 models.

6 TEXT-TO-SPEECH SYNTHESIS

The quality of speech synthesis is determined by two factors – intelligibility and naturalness. High intelligibility of TTS is an inevitable condition for the proper function of a voice driven information system. However, it is very important for the user's comfort to have a synthesized speech that sounds naturally and human-like. Thus, the main challenge in TTS within a SLDS is then the trade-off between the intelligibility and the naturalness.

A diphone synthesizer is capable of producing intelligible speech from a variety of different kinds of texts. It is flexible and totally domain-independent. It is computationally inexpensive and has a small memory-footprint. However, this type of synthesizer sounds a bit robot-like and tedious.

Much better naturalness can be achieved by a unit-selection synthesis from a large speech-database, which, on the other hand, can encounter some problems with intelligibility. The easiest way to overcome them is to create a limited domain speech database covering mainly the specific speech material concerning the particular application. Long parts of the utterances to synthesize are then contained in the database and can be "played" in a whole, which leads to a very natural speech signal.

Therefore we have decided to develop two TTS modules using two different approaches – diphone and unit-selection synthesis.

6.1 The Diphone Concatenative Synthesizer

This speech synthesizer [16] is based on concatenation of small elements taken from the pre-recorded speech signal, mainly diphones. An original algorithm similar to the Time Domain Pitch Synchronous Overlap and Add (TD-PSOLA) was used for concatenation.

The pronunciation is controlled by a block of orthographic-to-orthoepic (grapheme to phoneme) conversion based on a sophisticated set of rules supplemented by a pronunciation vocabulary and a list of exceptions. This elaborated unit has proven to be more reliable than our similar data driven system based on CART trees [17].

6.2 Unit-Selection Synthesizer

The second method – unit-selection synthesis is based on concatenation of suitable acoustic units selected from a speech database. Concatenated units may be of non-uniform length. The advantage of unit-selection synthesis method is minimizing the number of concatenation points in synthesized speech and thus reducing the need for speech processing causing the generated speech sound being artificial.

The most critical phase of this synthesis is the selection of appropriate units [18]. Two cost functions are used for evaluating the optimal unit sequence: target cost defined as the difference between the desired unit and each candidate unit in the

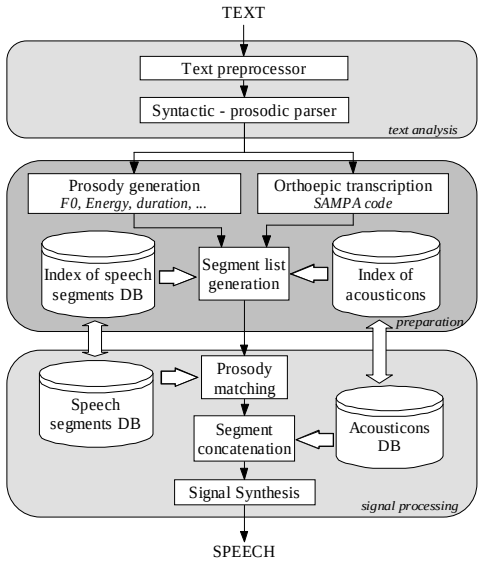

Fig. 4. Scheme of the diphone concatenative synthesizer

database and concatenation cost defined as the difference between two elements at their concatenation point. The desired unit (target) can be represented by various parameters characterizing its prosodic and phonetic features. Since these features are also used as the representation of other units in the database, the cost functions are computed as distances between these parameters. Apparently, the final selected unit sequence is determined by the lowest values of the costs. However, there are a couple of ways how to take the costs into account. One of them is to select those units that minimize the equation:

$$C(t_i^n, u_i^n) = \sum_{i=1}^{n} C^t(t_i, u_i) + \sum_{i=2}^{n} C^C(u_{i-1}, u_i) \tag{2}$$

where n stands for number of units in the sequence, u_i for the i^{th} unit in the sequence, t_i for the i^{th} target, and C^t and C^C for target and concatenation cost, respectively. Each cost is represented by several subcosts that contribute to the final cost by different degrees and so these subcosts must be weighted. The minimum sums in (1) are then effectively computed by Viterbi search.

The examination of the database [19] showed that not all types of diphones were present in it. As the TTS has to be able to synthesize any input, not just the intended domain dialogue, the unit selection algorithm was enriched by a method of substituting missing diphones by phonemes. The method introduces 5 types of basic synthesis units, which are combined in the final target specification for the selection algorithm [20]. The advantage of using the defined types is that some

missing diphones may be substituted by expansion of borders of neighboring diphones. This may save a couple of concatenations and so enhance the synthesis quality, particularly at word and phrase boundaries.

An automatic procedure within the ATK system based on triphone HMMs was used for the database annotation.

One of the drawbacks of the realized corpus based synthesis is its long response time when synthesizing long phrases. That's why we had to implement a cache for these. Each long phrase target specification is first looked up in the cache; if it is found, no synthesis takes place, just samples are copied from the cache file.

The flow of TTS process within our SLDS can be seen in Figure 5. It starts with a request for the synthesis coming from the HUB server. The request contains the text to be synthesized. First, the text is processed by a module of high level synthesis which substitutes all non-standard characters and transcribes the text to phonetic notation. The phonetic transcription is done either by dictionary or data driven rules. The phonetic notation (SAMPA alphabet) is the target synthesis specification for the low level synthesis module. In the actual version of low level synthesis module the best speech unit sequence is given by minimum number of concatenations only [21]. In case of equal number of concatenations, the lowest distortion of F0 or CLPC coefficients at the concatenation points is decisive. Finally, the synthesized signal is stored to an audio file and sent via the broker channel to the audio server that plays it to the user.

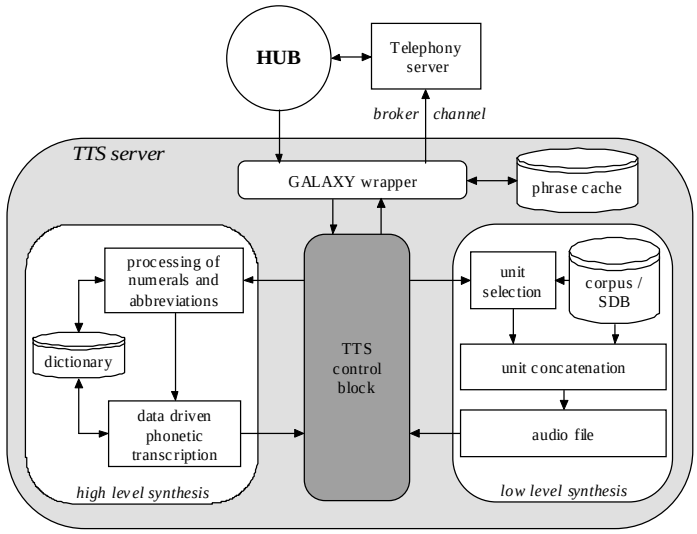

Fig. 5. Schematic diagram of the Unit-selection TTS server

7 DIALOGUE MANAGER

There are many approaches to how to solve a dialogue manager unit and also many languages for writing a code for it. Voice Extensible Markup Language (VoiceXML) is a markup language for creating voice user interfaces that use automatic speech recognition (ASR) and text-to-speech synthesis (TTS). Many commercial VoiceXML-based speech applications have been deployed across a diverse set of industries, including financial services, government, insurance, retail, telecommunications, transportation, travel and hospitality. VoiceXML simplifies speech application development, enables distributed application design and accelerates the development of interactive voice response (IVR) environments. For these reasons, VoiceXML has been widely adopted within the speech industry, and for these reasons we decided that the dialogue manager unit should be based on VoiceXML interpretation.

Figure 6 shows the main components of our dialogue manager based on the VoiceXML interpreter [22]. Its fundamental components are VoiceXML interpreter, XML Parser and ECMAScript unit. The VoiceXML interpreter interprets VoiceXML commands. They are not interpreted sequentially top down, but their executing is exactly described by the VoiceXML algorithms: Document Loop Algorithm, Form Interpretation Algorithm, Grammar Activation Algorithm, Event Handling Algorithm and Resource Fetching Algorithm. XML Parser creates hierarchical image of VoiceXML application from the VoiceXML code and parses XML grammars and XML content from internet locations. ECMAScript unit is important to support ¡script¿ element of VoiceXML and enables writing of ECMAScript code in VoiceXML. The unit is based on SpiderMonkey (JavaScript-C) engine.

The additional components of the dialog manager are: input interface unit, managing user's spoken input, time intervals and input events (noinput, nomatch); grammar's handling unit, implementing grammar activation algorithm, taking care about scope of grammars, creating support for several grammar formats, generating grammars and doing conversion between grammars formats; output interface unit, generating the prompts to the user, doing conversion between code tables and implementing prompt selection algorithm; setup manager, responsible for managing of system setup parameters; logging interface, logging all events that occur during a dialog into "log" files and serving as monitoring tool.

The dialogue manager is written in C++. We have started from scratch and in its actual state the interpreter performs all fundamental algorithms of VoiceXML language and service functions for all VoiceXML commands, i.e. Form Interpretation Algorithm, Event Handling, Grammar Activation and Resource Fetching Algorithms. It supports the full range of VoiceXML 1.0 functions. Basic components of the current VoiceXML based dialogue management unit are shown in Figure 6. Our future goal is a full support of VoiceXML v2.0.

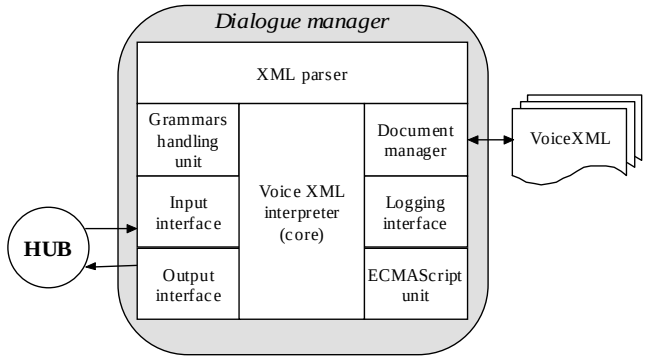

Fig. 6. Basic components of VoiceXML based dialogue management unit

8 AUDIOSERVER

The Audioserver is in charge of providing the whole information system with reliable multiuser connection to the telephone networks. The telephony server is written in C++ and supports telephone hardware (Dialogic D120/41JCT-LSEuro card). The ports are connected to a PABX switch that enables interconection with any common communication network (see Figure 7). Thus the audioserver can be accessed from the outside communication networks by using various telecommunication terminals such as mobile phones, ISDN/PSTN fixed-telephone, and VoIP.

Almost all of the communication among the various servers in the system is routed through the Hub. However, speech prompts, which are high data-rate messages, are brokered by the hub instead, in order to reduce the network load. The direct (broker) connection between audio server and ASR server or TTS server is established to transmit speech data and this way to reduce the network load. In the receiving mode, the audio server sends to the hub a token informing it of an impending audio data. In accordance with the hub script, the hub consults the existing ASR servers, to determine if there are any ASRs free to receive data. In the transmitting mode the TTS server sends to the hub a token to initiate establishing the broker channel between TTS and audio servers.

The telephony board implemented into the system is equipped with useful functionalities such as Advanced Signal Processing module, dual-tone multifrequency (DMTF) detection and barge-in detection (Dialogic firmware). An alternative touch-tone input is particularly convenient in noisy environment or when a higher level of secure communication is required. Barge-in function allows an interruption of the data flow during playing a prompt from TTS server and switching automatically to the receiving mode if any speech data are present at the input (i.e. any time when a user starts speaking).

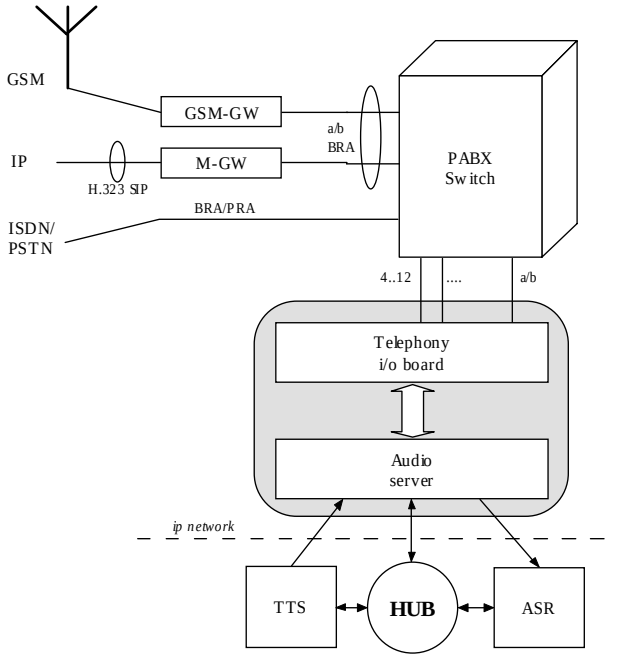

Fig. 7. Connection of audio server to the communication network (PRA – Primary rate access, BRA – Basic rate access, M-GW/GSM-GW – Media/GSM gateway)

9 BACKEND MODULE

Having analyzed various existing approaches of information retrieval from the web, and the task to be carried out by the information server in our pilot applications, we came to the decision that a simple retrieval technique, such as a rule based ad-hoc application searching only several predefined web-servers with a relatively well known structure of pages is sufficient to fulfill the task. Selection of eligible web servers was preceded by proper examination for stability and information reliability during one month period.

The information server (backend server) is capable of retrieving the information contained on the suitable web-pages according to the Dialogue Manager (DM) requests, extracting and analyzing the desired data. If the data are taken for valid, it returns them in the XML format to the DM. If the backend server fails to get valid data from one web source, it switches to the second wrapper retrieving the information from a different web-server.

The information server communicates with the HUB via the GALAXY interface. This module accomplishes its own communication with HUB, receives input requests, processes them and makes decisions to which web-wrapper (WW) the re-

quest should be sent. Then it receives an answer and sends it back to the HUB. Information server (Backend) architecture is shown in Figure 8.

The web-wrapper is responsible for the navigation through the web-server, data extraction from the web-pages and their mapping onto a structured format (XML), convenient for further processing. The wrapper is specially designed for one source of data; thus to combine data from different sources, several wrappers are required.

The wrappers are designed to be as robust as possible against changes in the web-page structure. Nevertheless, in the case of substantial changes in the web-page design, the adaptation of the wrapper would probably be inevitable.

An automatic periodic download and cashing of the web-page content is performed to speed up the system (to eliminate the influence of long reaction times of the web-pages) and to ensure robustness against drop-out while simultaneously keeping the information as current as possible.

The server allows for extension with additional web-wrappers; thus it is open for future applications and services. The information on the wrappers currently accessible is stored in the system configuration file.

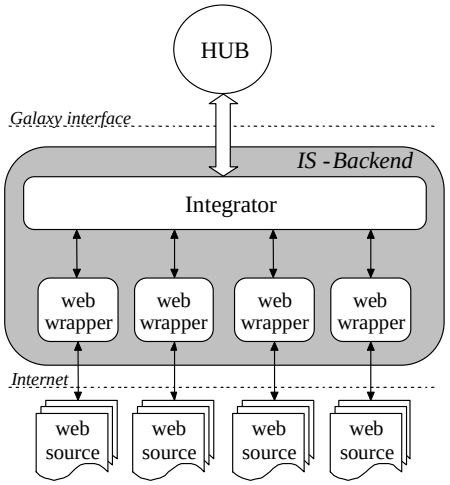

Fig. 8. Information server (Backend) architecture

10 PILOT APPLICATIONS

10.1 Dialogue Design Methodology

The methodology we used for building dialogues for the pilot applications, "Weather forecast in Slovakia" and "Timetable of Slovak Railways", consisted of five steps [23]:

- Database analysis. Information contained in the domain database is described by an Entity-Relationship diagram (E-R), that shows a semantic representation

of the data. In this diagram, the main entity sets, their attributes and keys and entity set relationships have to be defined.

- Dialogue alternatives design by intuition. A "brain-storming" on the E-R is carried out for proposing different dialogue alternatives. These proposals are concerned with the goals to provide (services such as timetable information, reservations, fares, etc.), the sequence of offering them and the items that are needed to satisfy each goal.
- Observation. The design by observation consists of analyzing user-operator dialogues in a similar service and tracking off different events observed.
- System simulation with a Wizard of Oz method in order to learn the specific characteristics of human-system interactions. In this simulation, we focused on the design of the dialogue flow.
- Interactive improvement. This step consists of an iterative process in which the system is tested and modified at the same time.

10.2 Weather Forecast Service

Based on the available web page `http://www.meteo.sk/` the proposed telephone-based Weather Forecast Service enables to get weather forecast for about 80 Slovak district towns and the most popular tourist localities. Callers can access continuously updated weather information simply by uttering the names of cities or holiday localities and date.

It was necessary for every entity to define their keys or key words, i.e. main obligatory items, which must be obtained from the user (the data necessary to query the database and to obtain the information requested):

Place: District town or holiday locality

Date: relative date/accurate date

10.3 Timetable of Slovak Railways

The second service running on our SLDS provides information about Slovak railways timetable. Callers have direct access to online timetable on webpage during this service.

In the first step of the dialogue design we specified the necessary domain database items. These items represent the required information which needs to be obtained from the user:

Starting place: a railway station in Slovakia

Destination place: a railway station in Slovakia

Date: relative date (today, tomorrow etc.)/absolute date ("the twentieth of December", etc.)

Time: departure time (hour, minute).

10.4 Implementation of Services

Both services are available at one access point, which is accessible from several telephone numbers. One can try them by calling +421 55 602 2297 (T-Com), +421 911 650 038 (T-Mobile), +421 918 717 491 (Orange).

At the beginning, the system provides an initial message, where it welcomes the user, introduces itself and notices the user about the key words/expressions and its use. The help service is activated when one of the following three different situations is detected: the user keeps quiet; the user says something that is not understood by the system, or the user explicitly asks for help.

Afterwards, the user is prompted to choose one of the available services – Weather forecast or Railway timetable. Then the dialog switches to the particular sub-dialog of the selected service.

A typical dialogue between the user (U) and the system (S) looks as follows:

S: Welcome to the IRKR portal. Would you like to play the introduction?

U: No.

S: Choose one of the services: Weather forecast or Railway's timetable.

U: Weather forecast

S: Please, name a city and assign a day, for which you want to get the weather forecast.

U: Bratislava, tomorrow.

S: Did you say Bratislava, tomorrow?

U: Yes

S: The weather forecast for Bratislava for tomorrow is: sunny, 32 degrees Centigrade...

Notice: IRKR means "Inteligentné rečové komunikačné rozhranie", which is a Slovak expression for Intelligent Speech Communication Interface.

11 EVALUATION OF THE PERFORMANCE

Obviously a final goal of a HCI (Human-Computer Interaction) system design is a user satisfaction with the system. In our case that means the developed speech user interface can only be effective if users can successfully interact with it. Objective measures of usability (e.g. error rate, response delay, etc.) do have some impact on user satisfaction, but it does not fully correlate with user's subjective perception. Hence subjective evaluation still remains a cornerstone in HCI evaluation [24]. Questionnaires and interviews remain the main tools for gathering information on user satisfaction [25, 26].

Recent efforts in subjective evaluation of speech based interfaces have resulted in development of the questionnaire standards such as BT-CCIR [27] and SASSI (Subjective Assessment of Speech System Interfaces) questionnaire [28].

To evaluate our SLDS we have proposed a questionnaire following the main ideas of BT-CCIR and SASSI. The questionnaire consists of

- six factual questions to capture demographic information like age, gender, experience, education, communication environment (office, street, vehicle, ...), and terminal type;
- six core questions to capture user's opinion and experience during interaction with the system. The answers are scored in Likert-like scale [29]. The following are two examples of questions with 2-degree and 5-degree Likert scales, respectively:
  - Did you get the information you were looking for? Yes/No.
  - Did the system understand what you uttered? Always/Almost everything/ Sometimes/Hardly anything/Never.

  The questionnaire concluded with two open-ended questions as follows:
  - What do you like/dislike about this service?

Usability testing was performed by 183 users. Distribution of users according to gender, age, experience, as well as test conditions are shown in Table 8. 91 % of the users used mobile phone over GSM network, 9 % called from fixed-line network or via VoIP. 31 % of them interacted with the system in public, often noisy, environment (street, public building, vehicle), and the rest at home or office. The users were asked to perform a specific task and fill in the questionnaire after interaction with the system. Examples of the tasks are as follows:

- Find departure of the first train from XXX to YYY after 5 am tomorrow.
- What is the weather forecast for this weekend in ZZZ?

The tasks were chosen with the aim to test various forms of time and place specifications (relative and absolute time and date, one-word and multi-word city names, relative locations).

11.1 Evaluation Results

A majority of the users (82 %) was satisfied with call duration (time for retrieving information) and assigned it as very short, short or adequate.

Quality assessments of TTS, ASR, and Dialog manager blocks, as they were perceived by users, are shown in Figure 9. Intelligibility of the diphone speech synthesiser (TTS) was assessed as good or fair in 95 % of cases. Speech recognition accuracy was assessed as good or fair in 77 % of cases. Only 6 % of users rated it as very poor. A dialog with the system was assessed as objective and concise by 70 % of the users, only 5 % of the users judged it as very long and tedious.

Overall satisfaction with the system was evaluated by the question: Would you use this service/system again? We assume that the users who are satisfied and have

a good experience with the system interaction, are willing to use such system again in the future. Positive answer was given by 81 % of the users; this is an encouraging result. It is desirable to note that this number corresponds not only to quality of the system but it is also influenced by social and psychological aspects of the users. The results are shown in Table 8, where they are arranged according to the user types and test conditions.

Total number of users (respondents): 183

Gender	male	70 %	Communication interface	mobile phone:	83 %
	female	30 %		fixed line:	8 %
Age (years)	0–20	11 %	Environment	home/office:	69 %
	41–60	15 %		public building:	8 %
	21–40	72 %		street:	14 %
	61+	2 %		vehicle:	9 %
Education	Primary school:	5 %	Experience with computer	never use:	11 %
	Secondary school:	77 %		seldom use:	18 %
	University:	18 %		often use:	71 %

Table 8. Users (test persons) classified according to user types and test conditions

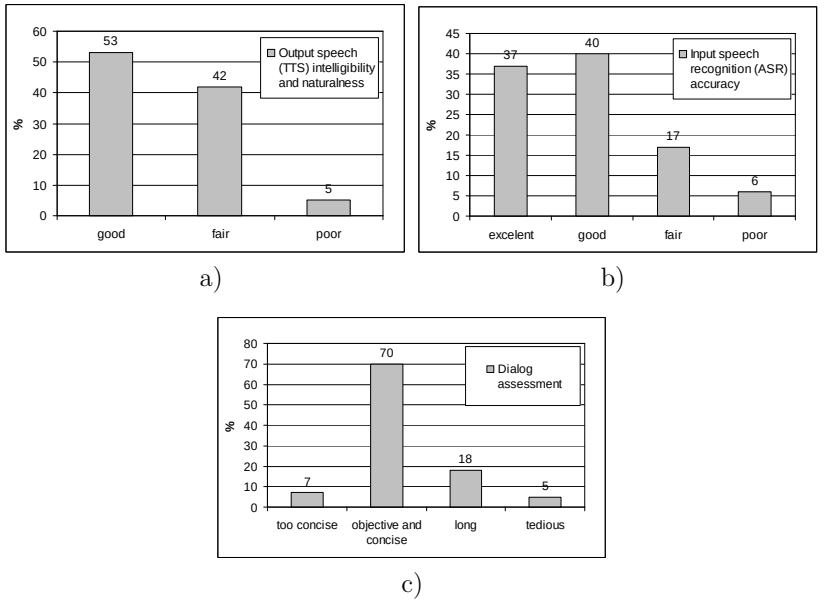

Fig. 9. Subjective evaluation of quality of a) TTS; b) ASR, c) DM

By detailed analysis of the questionnaires we found out that satisfaction with the system strongly correlates with the performance of the ASR block. That means

ASR is the most sensitive and challenging part of the system and thus the greatest effort hast to be oriented towards ASR improvements during system development.

12 CONCLUSIONS

In this paper we have described the development of the first Slovak spoken language dialogue system. Our main goal was to develop a dialogue system that will serve as a starting platform for further research in the area of spoken Slovak engineering. The work on this project was finished in 2006 [30]. We successfully combined up to date free resources with our own research into functional system that enables a multi-user interaction through telephone in Slovak language to retrieve required information from Internet resources. The functionality of the SLDS is demonstrated and tested by means of two pilot applications, "Weather forecast for Slovakia" and "Timetable of Slovak Railways". Applying the new findings we continue in further development and improvement of the system.

Acknowledgements

This work was supported by Slovak Grant Agency VEGA under grants No. 1/1057/04, 1/3110/06, 2/2087/22 and by the Ministry of Education of Slovak Republic under research project No. 2003 SP 20 028 01 03.

REFERENCES

[1] http://www.sls.csail.mit.edu/sls/technologies/galaxy.shtml.

[2] http://communicator.sourceforge.net/.

[3] http://fife.speech.cs.cmu.edu/Communicator/index.html.

[4] http://communicator.colorado.edu/.

[5] Young, S.: ATK: An Application Toolkit for HTK, Version 1.3. Cambridge University, January 2004.

[6] http://htk.eng.cam.ac.uk/.

[7] Lamere, P.—Kwok, P.—Walker, W.—Gouvea, E.—Singh, R.—Raj, B.—Wolf, P.: Design of the CMU Sphinx-4 Decoder. In Proc. Eurospeech 2003, Geneve, Switzerland, September 2003, pp. 1181–1184.

[8] Mirilovič, M.—Lihan, S.—Juhár, J.—Čižmár, A.: Slovak Speech Recognition Based on Sphinx-4 and SpeechDat-SK. In Proc. DSP-MCOM 2005, Košice, Slovakia, Sept. 2005, pp. 76–79.

[9] Pollák, P.—Černocký, J.—Boudy, J.—Choukri, K.—Rusko, M.—Trnka, M. et al.: SpeechDat(E) "Eastern European Telephone Speech Databases". In Proc. LREC 2000 Satellite workshop XLDB – Very large Telephone Speech Databases, Athens, Greece, May 2000, pp. 20–25.

[10] Rusko, M.—Trnka, M.—Darjaa S.: MobilDat-SK – A Mobile Telephone Extension to the SpeechDat-E SK Telephone Speech Database in Slovak. Proceedings of the XI. International Conference SPECOM 2006, St. Petersburg, Russia, 2006, pp. 449—454, (ISBN 5-7452-0074-X).

[11] Lindberg, B.—Johansen, F. T.—Warakagoda, N.—Lehtinen, G.—Kačič, Z.—Žgank, A.—Elenius, K.—Salvi, G.: A Noise Robust Multilingual Reference Recognizer Based on SpeechDat (II). In Proc. ICSLP 2000, Beijing, China, 2000.

[12] Winski R.: Definition of Corpus, Scripts and Standards for Fixed Networks. Technical report, SpeechDat-II, Deliverable SD1.1.1., workpackage WP1, January 1997, http://www.speechdat.org.

[13] http://ui.sav.sk/speech/index.htm.

[14] Lihan, S.—Juhár, J.—Čižmár, A.: Crosslingual and Bilingual Speech Recognition with Slovak and Czech SpeechDat-E Databases. In Proc. Interspeech 2005, Lisabon, Portugal, September 2005, pp. 225–228.

[15] Lihan, S.—Juhár, J.Čižmár. A.: Comparison of Two Slovak speech Databases in Speech Recognition Tests. 33rd International Acoustical Conference ACOUSTICS High Tatras '06, Štrbské Pleso, Slovakia, October 4–6, 2006 (accepted paper).

[16] Rusko, M.—Trnka, M.—Darjaa, S.: Three Generations of Speech Synthesis Systems in Slovakia. Proceedings of the XI. International Conference SPECOM 2006, St. Petersburg, Russia, 2006, pp. 449–454 (ISBN 5-7452-0074-X).

[17] Černák, M.—Rusko, M.—Trnka, M.—Darjaa, S.: Data-Driven Versus Knowledge-Based Approaches to Orthoepic Transcription in Slovak. Proceedings of ICETA 2003, Košice (Slovak Republic), pp. 95–97, 2003.

[18] Conkie, A. D.: Robust Unit Selection System for Speech Synthesis. Joint Meeting of ASA, EAA, and DAGA, paper 1PSCB_10: Berlin, Germany, 15.–19. 3. 1999.

[19] Páleník, A.—Čepko, J.—Rozinaj, G.: Design of Slovak Speech Database for Limited Domain TTS within an IVR System. 48th International Symposium ELMAR-2006 focused on Multimedia Signal Processing and Communications, 7–9 June 2006, Zadar, Croatia.

[20] Čepko, J.—Rozinaj, G.: Corpus Synthesis of Slovak Speech. Proc. of the 46th International Symposium Electronics in Marine – ELMAR 2004, pp. 313–318, Zadar, Croatia. 2004.

[21] Čepko, J.—Turi Nagy, M.—Rozinaj, G.: Low-Level Synthesis of Slovak Speech in S2 Synthesizer. 5th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services, Smolenice, Slovak Republic, 2005.

[22] Ondáš, S.—Juhár, J.: Dialogue Manager Based on the VoiceXML Interpreter. In Proc. DSP-MCOM 2005, Košice, Slovakia, September 2005, pp. 80–83.

[23] Gladišová, I.—Doboš, Ľ.—Juhár, J.—Ondáš, S.: Dialog Design for Telephone Based Meteorological Information System. In Proc. DSP-MCOM 2005, Košice, Slovakia, September 2005, pp. 151–154.

[24] Polifroni, J.—Seneff, S.—Glass, J.—Hazen, T. J.: Evaluating Methodology for a Telephone-Based Conversational System. Proc. First Language Resources and Evaluation Conference, 1998, pp. 43–49.

[25] Steele, A.—Mehta, K.: Design and Evaluation of Voice User Interfaces. Proc. Midwes Software Engineering Conference MSEC2003, Chicago, IL, June 2003.

[26] Larsen, L. B.: Assessment of Spoken Dialogue System Usability – What are We really Measuring? Eurospeech '03, Geneva, 2003.

[27] Love, S.—Dutton, R. T.—Foster, J. C.—Jack, M. A.—Stentiford, F. W. M.: Identifying Salient Usability Attributes for Automated Telephone Services. Proc. Int. Conf. on Spoken Language Processing ICSLP '94, September 1994, pp. 1307–1310.

[28] Hone, K. S.—Graham, R.: Towards a Tool for the Subjective Assessment of Speech System Interfaces (SASSI). Natural Language Engineering, Vol. 6, 2000, No. 3/4, pp. 287–305.

[29] Larsen, L. B. ed.: Evaluation Methodologies for Spoken and Multi Modal Dialogue Systems. COST278 WG2 and WG3 report, Stockholm, May 2–4, 2003.

[30] Juhár, J.—Ondáš, S.—Čižmár, A.—Rusko, M.—Rozinaj, G.—Jarina, R.: Development of Slovak GALAXY/VoiceXML Based Spoken Language Dialogue System to Retrieve Information from the Internet. Proceedings of the Ninth International Conference on Spoken Language Processing (Interspeech 2006 – ICSLP), Pittsburgh, Pennsylvania, USA, 2006, ISSN 1990-9772, pp. 485–488.

Jozef Juhár graduated from the Technical University of Košice in 1980. He received his Ph. D. degree in radioelectronics from the same university in 1991, where he works as an Associate Professor at the Department of Electronics and Multimedia Communications. He is author and co-author of more than 70 scientific papers. His research interests include digital speech and audio processing, speech and speaker recognition, speech synthesis and spoken dialogue systems in telecommunication networks.

Anton Čižmár – Slovak University of Technology, Bratislava 1980; Ph. D. – Technical University of Košice 1986; Associate Professor – Technical University of Košice, 1990; Dr. h. c. – University of Oradea, Romania 1998; Full Professor – Technical University of Košice 1999; Rector of the Technical University of Košice 2006. Author and coauthor of several books and more than 140 scientific papers. His interests include telecommunication management, project management, broadband information and telecommunication technologies, multimedia systems, telecommunications networks and services, energy networks and data transmission, man-machine communications.

Milan Rusko graduated from the Slovak Technical University, Bratislava in 1994. Since 1993 he has been the Head of the Department of Speech Analysis and Synthesis of the Institute of Informatics of the Slovak Academy of Sciences. He is author and co-author of more than 50 scientific papers. His research interests include speech acoustics, speech corpora, digital speech and audio processing, speech recognition and speech synthesis.

Marián Trnka received his Ing. degree in technical cybernetics in 1994 from the Slovak Technical University in Bratislava. He has joined the Department of Speech Analysis and Synthesis at the Institute of Informatics of the Slovak Academy of Sciences in 1996. His research interests include unit selection and diphone speech synthesis systems, speech recognition, signal processing algorithms and speech databases.

Gregor Rozinaj received his M. Sc. and Ph. D. degrees in telecommunications from Slovak University of Technology, Bratislava, Slovakia in 1981 and 1990, respectively. He has been a lecturer at Department of Telecommunications of the Slovak University of Technology since 1981. In 1992–1994 he was with Alacatel Research Center in Stuttgart, Germany. In 1994–1996 he was employed as a researcher at University of Stuttgart, Germany. Since 1997 he has been the Head of the DSP group at Department of Telecommunications of the Slovak University of Technology, Bratislava. Since 1998 he has been an Associate Professor at the same department. He is the author of 3 US and European patents and of 1 Czechoslovak patent. His main research interest is oriented to fast algorithms for DSP and speech processing.

Roman Jarina received his Ing. and Ph. D. degrees from the University of Žilina, Slovakia in 1990 and 2000, respectively. Between 2000 and 2002 he was a Postdoctoral Fellow at Dublin City University, Ireland where he was involved in development of an advanced algorithms for content-based audio a video analysis. Currently he is an Assistant Professor and also the Head of the Digital Signal Processing group at the Department of Telecommunications of the University of Žilina. His research interests are in the areas of acoustics, digital audio and speech processing, recognition, and multimedia information retrieval. He is a member of the Signal Processing Society of the IEEE, of the Institute of Engineering and Technology (IET) and of the Audio Engineering Society (AES). He is also Slovak representative of the EU action COST292.