ISSN 1392 – 1215 ELEKTRONIKA IR ELEKTROTECHNIKA. 2005. Nr. 3(59)

T 180 TELEKOMUNIKACIJŲ INŽINERIJA

Speech in Call and Web Centers

A. Rudžionis, K. Ratkevičius,
Speech Research Laboratory, Kaunas University of Technology
Studentų str. 65, LT-51369, Kaunas, Lithuania; phone: +370 37 35419; e-mail: alrud@mmlab.ktu.l, karat@mmlab.ktu.lt
V. Rudžionis,
Dept.of Informatics, Kaunas Humanities Faculty of Vilnius University
Muitinės str. 8, LT-44280 Kaunas, Lithuania; phone: +370 37 354191; e-mail: vyrud@mmlab.ktu.lt

Introduction

Described recently by Intel Corp. co-founder Gordon Moore as the next transforming technology, speech interaction without question provides extraordinary value to us all —even a 4-year-old child can understand the power of speech. Over the past two decades, significant progress has been made advancing speech technology as a mainstream modality for human/machine interaction. However, the industry has yet to bridge the gap between what people want and what it can deliver [1,2].

Reducing the speech recognizer's error rate in all environments remains the greatest challenge to making speech mainstream. Over the past decade, the speech research community maintained a relative error-reduction rate of 10 percent annually. A number of benchmarking studies indicate that there will be a speech recognizer that is almost as accurate as the human ear in the next 10 to 40 years [1].

Table 1 summarizes some recent studies on understanding the gap between people's expectations and the industry's capabilities. The most striking observation is that it may take the industry another 40 years to achieve simple digit-string recognition. Considering that neither human nor machine has any high-level context information for digits, the industry still has a long way to go to improve basic speech technology. On the other hand, we can see that the machine may reach the level of human performance in complicated tasks such as speaker-dependent dictation much earlier than digit-string tasks [1].

It is clear that speech recognition is a problem that has not yet been solved. Although reducing the speech recognition error rate is an ongoing issue for speech researchers, the technology is already useful in addressing other interaction challenges. Telecommunications carriers already use speech recognition for directory assistance. Speech recognition is employed in many luxury cars to control non-critical functions. PC users can dictate directly into Microsoft Word XP. Text to speech (TTS) quality has also reached the level that is not only intelligible but also close to be natural. Unquestionably, speech technologies have had a positive impact on businesses and consumers. The analyst firm Datamonitor PLC estimates that the supply-side market for speech technologies and services will grow to more than $4 billion over the next three years [1].

Speech technologies can be used by enterprise customers to reduce costs, enhance revenues and improve business agility [3]. There are incredible success stories of companies implementing speech solutions. For example, United Airlines saved $25 million when it introduced speech recognition to replace its interactive voice response (IVR) system [1]. A leading financial services provider that manages funds in excess of $77 billion and serves over 525,000 401(k) customers was using a touch-tone IVR system that was not providing an easy and quick way for customers to access and update personal account information. The company decided to implement speech on top of their existing IVR system and soon realized significant cost savings. The automated speech system reduced the average cost of an agent-assisted call from $4.23 to 36¢. Additionally, average call time decreased from 12 minutes per call to less than 2 minutes [3].

Table 1. Machine's and human's error rates for some recognition tasks and the number of years for machines to catch up with humans

Tasks	Machines' error rate today	Humans' error rate	Number of years for machines to catch up with humans
Freestyle speech transcription	30 percent	4 percent	19 years
Digit strings	0.7 percent	0.009 percent	41 years
Alphabet letters	5 percent	1 percent	15 years
Newspaper speech transcription	3 percent	0.9 percent	11 years

With multiple locations and mobile employees, companies are demanding anytime, anywhere access to information. Enterprises can improve employee productivity and agility via a unified Web and voice portal that enables employees to access information through not only a PC, but also a wireless personal digital assistant (PDA) or telephone. This will be one of the key factors driving the growth of the speech industry, since speech is the only modality that can provide a consistent user interface across all devices.

There are two alternatives for the developing and deploying of speech-enabled applications: Microsoft Speech Server (MSS) and IBM WebSphere Voice Server [4]. The IBM WebShpere Voice Server is a VoiceXML 2.0 (Voice eXtensible Markup Language)-enabled speech environment [5]. The VoiceXML is aimed at developing telephony-based applications, and takes the advantages of Web-based applications delivery to IVR applications. Being different from IBM, Microsoft is using SALT 1.0 (Speech Application Language Tags) within Microsoft Speech Server [6]. SALT is a set of light-weight extensions to XML, adding speech-enabled telephony to Web-based applications and bringing them into a multimodal model. SALT targets speech-enabled applications across all devices such as telephones, PDAs, tablet PCs, and desktop PCs. The VoiceXML focuses on telephony application development whereas SALT is focused on multimodal speech applications that can be accessed by the whole device [4].

Futher we will overlook both alternatives for the developing of speech-enabled applications and will analyze our choice - Microsoft Speech Server (MSS) in detail.

IBM WebSphere Voice Server

The IBM WebSphere Voice Server and SDK/Toolkit is a member of the IBM WebSphere software family that is a SUN Java Framework-based Web application environment. You have to install the SUN Java Framework in both the Voice Server and development environment. The IBM Voice Toolkit for WebSphere Studio enables developers to create voice applications in less time, by using a VoiceXML application development environment, which includes a VoiceXML editor, grammar editor, and a pronunciation builder, and allows application developers to easily add voice technology to middleware applications. You can use Java2, Servlets, JSP, CGI, JDBC, JavaScript, and so forth to develop your speech application. The IBM WebSphere Voice ToolKit is integrated with the IBM WebShpere Studio development platform [4].

The IBM WebSphere Voice Server for Multiplatforms V4.2 includes VoiceXML voice browser, IBM Speech Recognition Engine, IBM TTS Engine, telephony and media component, and so forth. It can connect with many telephony platforms, including WebSphere Voice Response for AIX/Windows, Intel Dialogic, Cisco or Siemens HiPath, and Voice Server Speech Technologies for Windows and Linux [4]. The IBM WebSphere Voice Server is scalable, starting from basic analog telephony boards to high-density digital solutions with a T1/E1 interface, including Intel Dialogic D/120JCT, D/240JCT, D/480JCT, D/300JCT, and D600JCT [4].

Microsoft Speech Server

Microsoft Speech Server is comprised of two main components, Speech Engines Services (SES) and Telephony Application Services (TAS). SES provides engine services on the server side that perform interpretation of the SALT and speech processing, including a speech recognition engine, prompt engine, and TTS engine for speech-enabling IVR customers through TAS. The Speech Recognition Engine component works on handing a caller's speech input; the Prompt engine joins pre-recorded prompts from a prompt database and plays them back to the caller; the TTS engine works Text-to Speech to synthesize audio output from a text string. TAS serves as a connection proxy between PBXs and SES by managing a set of SALT interpreters, calling control Call control (establishing, managing, and terminating the voice connection) and telephony interface managers[4].

TAS is comprised of both telephony hardware and a software interface. So far, the telephony hardware that can currently work with a TAS server include Intel Dialogic boards D41JCT, DMV160LP, which have 4 and 16 analog channels and boards DM/V480, DM/V960,. which have 48 and 96 digital voice ports. Normally, TAS must work with third-party Telephony Interface Manager (TIM) software that is an interface between TAS and telephony hardware; right now, for TAS software, Intel NetMerge CallManager and InterVoice TIM exist in the marketplace [4].

Microsoft Speech Application Software Development Kit (SASDK)

The Microsoft Speech Server and Speech Application SDK (SASDK) are based on the MS .NET Framework. You have to install the .NET Framework and ASP.NET Speech Controls modules in the speech server (SES/TAS) development machines as well as the Web server. SASDK is being seamlessly integrated with MS .NET Visual Studio 2003; when you install the SASDK, all controls of the SASDK will be appear on the Visual Studio 2003 development environment toolbar [2]. When programming under ASP.NET, on the server side you can code in C#.NET or VB.NET and use JScript or VBScript to code on the client side. Plus, you are able to use ADO.NET to implement database access and transactions [4].

SASDK provides ASP.NET Speech controls, Speech Control Editor, Speech Grammar Editor, Speech Prompt Editor, Speech Debugging Tools such as Telephony Application Simulator, Speech Debugging Console, Speech Debugging Console Log Player, Speech Add-in for Microsoft Internet Explorer, a speech application deployment service, and a broad set of grammar libraries [2].

Four types of speech controls are available. Basic, Dialog, and Application speech controls provide a server-side (Web server control) API that manages the speech interaction of your application. Call management controls are an abstraction of the Computer Supported Telecommunications Applications (CSTA) messages you'll use in your application [7].

The Basic speech controls are a very basic server-side API for the two core SALT elements, <prompt> and <listen>, as well as the full hierarchy of elements they support. The <listen> element is used for speech recognition, for audio recording or for both. The <prompt> element is used to specify the content of audio output.

The Dialog speech controls are the building blocks of dialogue in the SASDK. The SALT constructs required to implement the full interaction and management of the dialogue's state is wrapped up into this API to simplify and limit the code you need to write to wire up your presentation logic.

The Application speech controls are packaged dialogue components for collecting specific types of information. They accelerate development of common voice-only scenarios such as collecting credit card information. In fact, you can create your own Application speech controls for scenarios you might encounter in your applications. The Application speech controls are, in fact, compositions of Dialog speech controls.

Telephony call control

Call control describes the collection of functions responsible for establishing, maintaining, and terminating calls. The IVR applications require call control functions such as answering a call, transferring a call, conferencing, making a call, disconnecting a call, and so forth.

In IVR applications with an analog phone line, call signaling is accomplished in-band; that is known as channel-associated signaling (CAS) because the signaling is transmitted in the same channel as the voice [9]. Tones are also used for in-band signaling on digital connections. Alternatively, call control can be implemented as out-of-band signaling protocols; in this way, the signaling is communicated on a separate channel from the voice. Out-of-band signaling is more reliable and scalable than in-band signaling. For instance, in ISDN (Integrated Services Digital Network) a single 64 kpbs signaling channel can support 23 voice channels. The out-of-band signaling protocols include Signaling System 7 (SS7-ISUP), Session Initiation Protocol (SIP), H.323, and so on [9].

Actually, two kinds of call control, first-party and third-party call control, are usually used in a speech-enabled IVR application to describe the relationship between the application and the call [9]. In first-party call control, the IVR application is also a talking party on the call. This implies a direct connection between the caller and the application. In third-party call control, the IVR application is not necessarily a talking party. By using third-party call control, an application can simultaneously monitor several calls [8].

Both IBM WebSphere Voice Server and MS Speech Server can provide simple call controls such as transfer call, make call, answer call, and so on. In some cases, if you want to implement complex call controls functionality, you have to use the CCXML (Call Control eXtensible Markup Language) editor in IBM WebSphere Voice Server [10] and use CSTA (Computer Supported Telephony Application) data extension controls in MS Speech Server [7]. CCXML provides telephony call control that can be used in VoiceXML or SALT-based, speech-enabling applications. CCXML can provide the call management, event processing, conferencing, and such that VoiceXML and SALT lacked. Currently, MS Speech Server 2004 cannot support CCXML [9].

Telephony Call control with Microsoft Speech Server

As was mentioned above, MS Speech Server use CSTA data extension controls. CSTA is an ECMA (European Computer Manufacturers Association) standard with an extensive feature set and a comprehensive call model. CSTA supports basic first-party call control and advanced third-party call control with the same standardized model. CSTA exposes advanced communication platform features to telephony application developers without burdening them with underlying protocol specifics. CSTA specifies an Applications Interface and Protocols for monitoring and controlling calls and devices in a communications network. These calls and devices may support various media and can reside in various network environments such as IP, TDM, and mobile networks.

The Telephony Application Services (TAS) component of MSS runs speech-enabled Web applications containing both application and call control logic created by developers using the SASDK. While developers work directly with call control objects within the SASDK, under the covers CSTA XML messages serve as the communication link between TAS and the TIM. The SALT interpreter, a TAS component, establishes a communication conduit to the TIM for call control purposes. The SALT <smex> element is used as the wrapper for call control messages that pass back and forth along this simple communication channel. The CSTA content of the XML request, response, and event messages are defined in Standard ECMA-323. Figure 1 shows how the speech application makes service requests, and how the TIM responds with service request responses and call control events [11].

TIM performs the following functions:

- provides the interface to the Intel telecom board, handling incoming and outgoing calls made through the telecom board;
- maps outbound call requests from a SALT interpreter to available channels on the Intel telecom board;
- maps call control signaling events from the switch into CSTA XML messages that can be understood by the SALT interpreter;
- maps CSTA XML messages from the SALT interpreter into telephony requests that can be sent to the switch;
- handles media processing, such as recording audio;

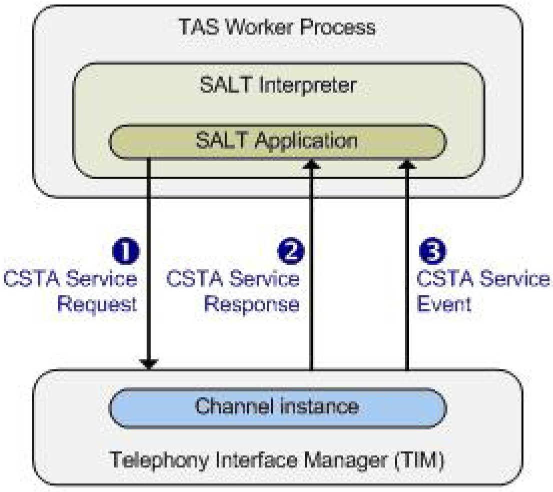

Fig.1. Call control communication between the application and the TIM

TIM handles incoming and outgoing calls in the following ways:

• when an incoming call arrives, TIM allocates it to a SALT interpreter instance. Each SALT interpreter instance is generated and managed by the TAS, that provides the link between the telephone system, and the Speech Engine Services (SES) and Web server. The SALT interpreter translates SALT-enhanced HTML script on speech-enabled web pages into speech processing requests;

• if a speech-enabled web page invokes an outgoing call, TIM maps the request from the SALT interpreter to an available channel on the Intel telecom board. The board passes the call request to the telephony network.

The SALT interpreter and Intel telecom board use different protocols to communicate with the TIM for all telephony events and requests. TIM maps these protocols so that the events and requests can be sent from the MSS environment to the telephony network environment and vice-versa. Figure 2 shows the different protocols supported [11].

For example, in Figure 2, when an incoming call arrives at the telecom board, the Call Manager (TIM) generates a CSTA XML DeliveredEvent message and sends it to the SALT interpreter. The SALT interpreter then passes a CSTA XML AnswerCall request to the TIM, and the Call Manager generates an "answer call" signaling request through the telecom board to the switch handling the call.

Call Management controls in SASDK

The SASDK includes basic call controls as well as extended call control capabilities via the SALT specification's Simple Messaging Extension (smex) element.

In speech-enabling IVR applications, normally its Web page starts with an AnswerCall control, and ends with a DisconnectCall control.

The AnswerCall control answers a telephone call. AnswerCall would be the first call control when you create an speech-enabling IVR application.

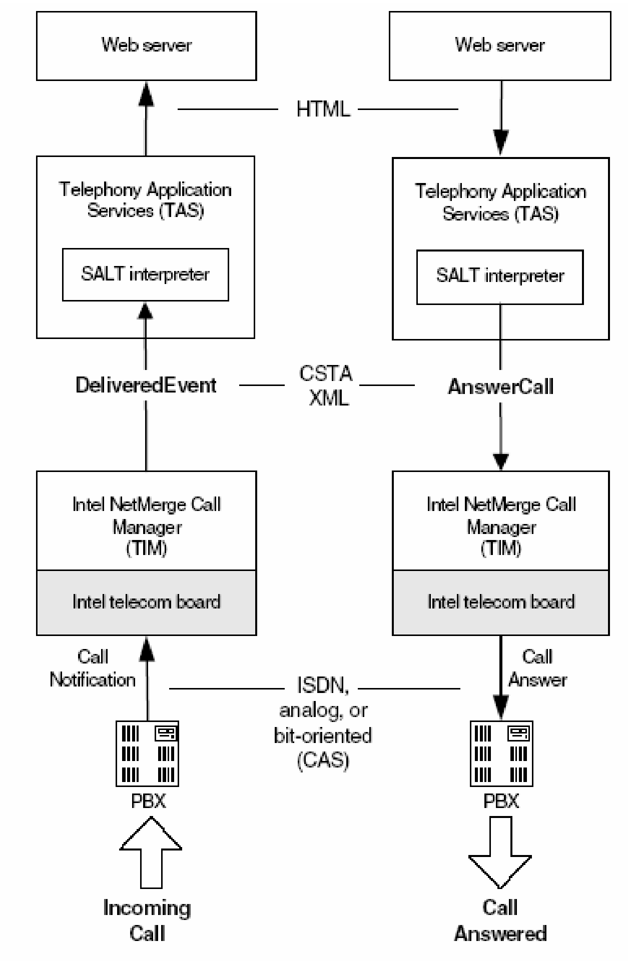

Fig. 2. Protocols used for call handling

The MakeCall control initiates a telephone call. The TransferCall control transfers the current telephone call to another phone. In a call center/IVR environment, typically TransferCall control transfers the IVR phone calls to a queue of PBX or an appropriate agent.

The SmexMessage control handles generic CSTA messages and events. When you customize call management controls in your applications, they should be derived from SmexMessage.

The DisconnectCall control disconnects a current telephone call. Typically, this control is used to terminate the IVR call [11].

CTI information in MS Speech Server

Computer-telephone integration (CTI) allows PC-based telephony applications to integrate with proprietary PBXs to retrieve the caller's information such as ANI, DNIS, and Entered Digits (for example, the caller's account number and PIN), and then implement a screen pop-in desktop agent according to the caller's account information that is retrieved from the data base source. At the same time, the application can instruct the PBX to transfer the call from the TAS to a queue of PBXs or an appropriate agent.

The Callinfo class of MS SASDK could provide information that might be used in a CTI application. For instance, CallID properties provide the call ID for the current active call; CalledDevice properties can get the called device information that is provided by the phone network; basically it is DNIS. CallingDevice properties provides the calling device information that is given by the phone network, normally called ANI; In some specific IVR/CTI applications, you may need to know the port number of telephony hardware boards for the current active call. You can retrieve the port number from CorrelatorData properties [9].

Normally, MS Speech Server works on a first-party call control model to perform call answering, play prompts, transfer a call, and disconnect a call. If you want to implement complex telephony applications such as contact center and CRM, you can consider using CTI middleware such as Intel NetMerge Call Processing Software (formerly CT Connect), Genesys CTI, Cisco ICM CTI, Avaya ASAI, and Nortel CTI Links [9].

PBX connects to speech-enabled IVR

A PBX is a privately owned, miniature version of a telephone company's central office (CO) switch. For businesses, the key advantage to owning a PBX is the efficiency and cost savings of sharing a specific number of telephone lines among a large group of users. Grouped with PBXs are key telephone systems (KTSS), generally a smaller version of of a PBX that provides direct access to telephone lines. PBXs offer users calling flexibility and numerous features, including direct inward dialing (DID). In addition, PBXs come equipped with digital telephones that offer dedicated function keys and ASCII display screens to access those features.

IVR allows customers to manipulate information in a computer database, such as retrieving an account balance and transferring funds from one account to another. These applications range from systems that deliver a selection of messages to transaction-based systems that let callers access accounts and update information on a LAN-based or host-based database with fax confirmation. One of the most valuable PBX integration features for IVR applications is the capability to retrieve caller profile information based on the caller ID (which trunk the call came in on) and customer identification number. Other valuable features include blind and supervised call transfers and positive disconnect supervision.

Call centers (both telemarketing and helpdesks) are partially or fully automated locations where a large number of agents or telephone operators process caller requests. With these types of applications, the system usually retrieves information about the callers and their requests before connecting them to agents. One of the most valuable PBX integration features for call center applications is the ability to quickly and easily retrieve caller profile information. Other valuable features include supervised call transfers and positive disconnect supervision.

A speech-enabled IVR application deployed in the enterprise environment can have many kinds of connection schemes among PSTN, IVR, and PBX, such as directly to the PSTN or sit behind or in front of a PBX. A PBX often supports multiple protocols simultaneously. In the MS Speech Server, normally the physical connection to the TAS/SES speech server can be analog lines or digital trunks by telephony interface boards installed in TAS server. Analog connections are deployed using interfaces similar to your home phone line. Digital circuit-switched networks use a time-division multiplexing (TDM) where a single voice channel occupies 64 Kbps of bandwidth. In North America, it called a T-1 that 24 voice channels are multiplexed into a single 1.544 Mbps bit stream. In Europe and most regions of Asia, E-1 is used; it has 30 voice channels with a 2.048 Mbps bit stream [9].

Examples of speech-enabled Web and telephony applications

To use above mentioned technologies for Lithuanian language, one must have a Lithuanian TTS and recognition engines installed. Two Lithuanian text-to-speech synthesizers "Aistis" and LtMBR are integrated to Microsoft Speech Programming Interface (SAPI) while Lithuanian speech recognizer compatible with SAPI so far is under development [12].

Some demonstrations of speech–enabled Web pages were prepared using SASDK (http://www.speech.itpi.ktu.lt): a) reading of input text by voice; b) filling of forms by voice c) virtual discotheque. Speech prompts by TTS and voice dialogues are used in these applications. User can control these multimodal applications by keyboard, mouse or by voice commands. To test this web page you need to install Windows'2000 or Windows'XP system and freely distributed SASDK. Lithuanian TTS engine is needful for TTS prompts in Lithuanian.

Another project applies IVR technology to provide long distance bus timetables between five largest towns in Lithuania. Automatic answer messages are generated from prerecorded phrases or words combining them. Lithuanian text-to-speech synthesizer of AISTIS system and recognition of voice commands through the telephone have been implemented in this project.

First attempts to master Microsoft Speech Server and Intel Dialogic board D41JCT were started.

Conclusions

There are two alternatives for the developing and deploying of speech-enabled applications: Microsoft Speech Server (MSS) and IBM WebSphere Voice Server. Since we are working for many years with Microsoft Visual Studio software and Microsoft Windows operation system, our choice was MSS. It combines Web technology with speech-processing services and telephony capabilities in a single system.

SALT technology allow to access the content of a webpage using text-to-speech engine, browse by speaking voice commands, fill in internet forms by speaking, leave voice messages and create entirely new possibilities for the customer.

The SASDK provides an integrated, comprehensive platform for building speech-enabled telephony and Web applications.

Some demonstrations of speech–enabled Lithuanian Web pages (http://www.speech.itpi.ktu.lt) and telephony services were prepared. First attempts to master Microsoft Speech Server and Intel Dialogic board D41JCT were started.

References

1. **Xuedong Huang**. Making Speech Mainstream. Retrieved March 7, 2008, from http://www.microsoft.com/speech/docs/HuangSpeechArtfinal.html.
2. Voice Technology Goes Mainstream with Release of Microsoft Speech Server 2004. Retrieved March 7, 2008, from http://www.microsoft.com/presspass/features/2004/mar04/03-24SpeechServer.asp.
3. The Business Value of Speech. Retrieved March 7, 2005, from http://www.microsoft.com/speech/businessvalue/speech/default.mspx.
4. **Xiaole Song**. Comparing Microsoft Speech Server 2004 and IBM WebSphere Voice Server V4.2. Retrieved March 8, 2005, from http://www.developer.com/voice/article.php/b3381851.html.
5. VoiceXML forum. Retrieved March 8, 2005, from http://www.voicexml.org.
6. SALT forum. Retrieved March 8, 2005, from http://www.saltforum.org.
7. Computer Supported Telecommunications Applications (CSTA). Retrieved March 14, 2005 from http://www.ecma-international.org/activities/Communications/TG11/cstaIII.htm
8. Telephony Fundamentals: An Introduction to Basic Telephony Concepts. Retrieved March 14, 2005 from http://www.intel.com/network/csp/pdf/3146wp.htm.
9. **Xiaole Song** .Understanding Telephony Concepts for Interactive Voice Responses. Retrieved March 8, 2005, from http://www.developer.com/voice/article.php/3461361.html.
10. Voice Browser Call Control: CCXML Version 1.0. Retrieved March 8, 2005, from http://www.w3.org/TR/ccxml/.
11. **Kershaw Dan.** An Introduction to Telephony Call Control with Microsoft Speech Server 2004. Retrieved March 14, 2005, from http://www.microsoft.com/speech/evaluation/whitepapers/TelephonyCallControlDoc/.
12. **Rudzionis A., Ratkevicius K., Rudzionis V.** Speech recognition research and some speech technologies applications in Lithuania. In: Proc. of the first Baltic Conference HLT-2004. Riga, Latvia. – P.132 –138.

Pateikta spaudai 2005 03 09

A. Rudžionis, K. Ratkevičius, V. Rudžionis. Kalba skambučių ir interneto centruose // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2005. – Nr. 3(59). – P. 58-63.

Nagrinėjamos balsinės telefonijos ir interneto paslaugos. Pristatomos dvi balsinių telefonijos ir interneto paslaugų kūrimo platformos: Microsoft kalbos serveris ir IBM WebSphere. Aprašomas Microsoft balso programavimo priemonių paketas SASDK. Pateikiamos telefoninių skambučių valdymo priemonės, esančios Microsoft kalbos serveryje ir SASDK pakete.. Pristatoma demonstracinė su SASDK paketu paruošta lietuviškai kalbanti interneto svetainė (http://www.speech.itpi.ktu.lt) bei paruoštos telefoninės paslaugos. Il. 2, bibl. 12 (anglų kalba, santraukos lietuvių, anglų ir rusų k.).

A. Rudžionis, K. Ratkevičius, V. Rudžionis. Speech in Call and Web centers // Electronics and Electrical Engineering. – Kaunas: Technologija, 2005. – No. 3(59). – P. 58-63.

Paper deals with the speech-enabled telephony and internet applications. Two alternatives (Microsoft Speech Server MSS and IBM WebSphere) for the developing of speech-enabled applications are analyzed. Microsoft Speech Application Software Development Kit (SASDK) is described. Telephony call control with MSS and call management controls in SASDK are presented. Some demonstrations of speech–enabled Lithuanian Web pages (http://www.speech.itpi.ktu.lt) and telephony services are prepared. . Ill. 2, bibl. 12 (in English, summaries in Lithuanian, English, Russian).

А. Руджёнис, К. Раткявичюс, В. Руджёнис. Речь в телефонных и интернетных центрах // Электроника и электротехника. – Каунас: Технология, 2005. – № 3(59). – С. 58-63.

В статье анализируются речевые интернетные и телефонные сервисы. Представлены две платформы для создания речевых телефонных и интернетных услуг: Microsoft Speech server и IBM WebSphere. Представлен пакет для программирования речевых приложений Microsoft SASDK. Представлен набор средств для управления телефонными вызовами из речевого сервера Microsoft и пакета SASDK. Описан говорящий по литовский интернет-сервер (http://www.speech.itpi.ktu.lt), созданный с помощью SASDK, а также подготовленные телефонные услуги. Ил. 2, библ. 12 (на английском языке; рефераты на литовском, английском и русском яз.).