GOVOREC — SISTEM ZA SLOVENSKO GOVORJENJE RAČUNALNIŠKIH BESEDIL

Jurij Leskovec
Odsek za inteligentne sisteme
Inštitut Jožef Stefan
Jamova 39, 1000 Ljubljana, Slovenia
e-mail: Jure.Leskovec@ijs.si
http://ai.ijs.si/govorec

POVZETEK

Članek opisuje sistem *Govorec* — sintetizator slovenskega govora. *Govorec* je zmožen branja računalniškega besedila v slovenskem jeziku. Sistem uporablja grafemsko-fonemski slovar s pol milijona gesli, grafemsko-fonemska pretvorba pa je podprta še z množico pravil. Prozodične parametre nastavljamo na treh nivojih: na nivoju stavka, besede in fonema. Osnovne govorne enote so difoni, ki jih sistem združuje s postopkom TD-PSOLA. Govorec je združljiv z Microsoftovim standardom Speech API, kar mu omogoča enostavno uporabo v okolju Windows. Zaradi standardnih vmesnikov, ki jih implementira *Govorec*, je sistem mogoče uporabiti na veliki množici že napisanih aplikacij. Posledica tega pa je, da Windows govorijo slovensko.

1. UVOD

Sistemi za sintezo govora in avtomatsko branje besedil so v zadnjih letih doživeli skokovit razvoj. Kaže se vedno večja potreba po prijaznejših uporabniških vmesnikih, pomemben del tega pa postaja tudi govor. Z razvojem mobilne telefonije se odpirajo nove možnosti uporabe sintetizatorjev kot vmesnikov pri komunikaciji človeka s strojem.

Sistem je bil razvit z mislijo na potrebe slepih in slabovidnih uporabnikov računalnikov. Ti uporabniki najlaže zaznavajo informacije, ki prihajajo iz računalnika, v zvočni obliki. Zaradi obilice tekstovnih informacij na internetu obstaja velika potreba po sistemu za branje besedil v slovenskem jeziku.

Slepi in slabovidni uporabljajo množico specializiranih njim namenjenih programov, ki jim olajšajo delo na računalniku. Pomemben del te množice so programi, ki omogočajo branje besedil z zaslona (Jaws, Narrator,…)[8].

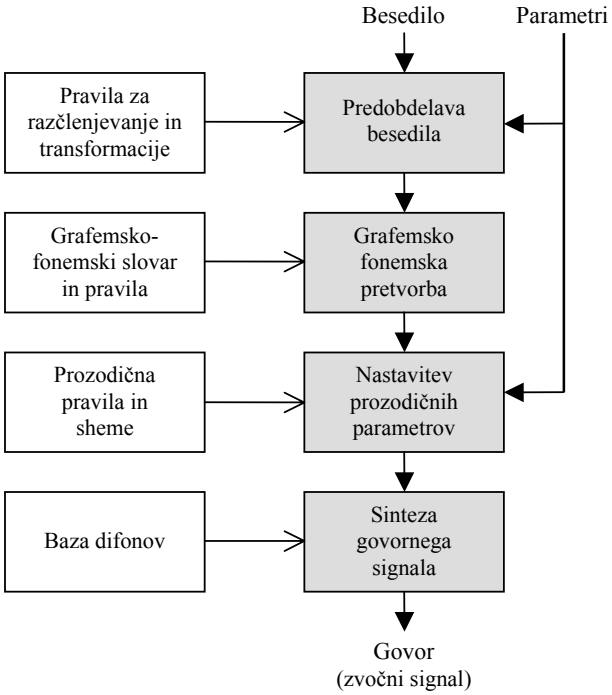
Govorec je sintetizator govora, združljiv z Microsof–tovim Speech API, kar mu omogoča enostavno integracijo v obstoječe programe za branje. Po namestitvi *Govorca* že nameščeni programi dobijo novo možnost branja v slovenskem jeziku, kar je za uporabnika najudobneje.

V nadaljevanju članka opisujemo delovanje posameznih modulov sistema *Govorec*. Članek vsebuje tudi pregled arhitekture in vmesnikov standarda Microsoft Speech API (SAPI) ter nove možnosti in načine integracije *Govorca*, ki jih prinaša združljivost s standardom.

2. ZGRADBA SINTETIZATORJA GOVORA

Sintetizator slovenskega govora *Govorec* je sestavljen iz več medsebojno neodvisnih modulov [4], ki so povezani v hierarhično strukturo (slika 1).

Vhod v sistem je zaporedje znakov, ki predstavlja besedilo. Besedilo nato v zaporednih korakih pretvarjamo in obdelujemo do faze, ko vsakemu fonemu v besedilu priredimo dovolj podatkov, da se lahko lotimo same sinteze govornega signala. Izhod iz sistema je digitalni zvočni signal, ki ga lahko predvajamo na zvočni kartici, telefonu ali ga zapišemo v datoteko.



Slika 1: zgradba sintetizatorja govora Govorec

2.1 PREDOBDELAVA BESEDILA

V postopku predobdelave besedila se znebimo vseh odvečnih simbolov, ki ne vplivajo na izgovorjavo besedila: presledki, znaki in elementi, ki v besedilo ne sodijo (HTML ukazi, kontrolne značke). Zaradi združljivosti s SAPI *Govorec* podpira tudi kontrolne značke, ki omogočajo vnos

ukazov v besedilo. Z njimi lahko izboljšamo prozodične parametre.

Pretvorimo tudi posebne skupine znakov v njihov grafemski zapis: razvijemo kratice in krajšave, pretvorimo številke, številke in datume v besede, posebej obravnavamo elektronske naslove, telefonske številke in matematične izraze. Pri pretvorbi poskušamo ugotoviti tudi pravo obliko (spol, sklon) razvite besede. Besedilo za tem razbijemo na stavke, stavke na besede in besede na zloge. Vsaki besedi s pomočjo naglasnih shem in pravil določimo mesto naglasa.

Na koncu predobdelave je celotno besedilo razdeljeno na stavke, stavki pa na besede. Besede tako vsebujejo le 25 grafemov slovenske abecede in informacijo o ločilu ter naglasu.

2.2 GRAFEMSKO-FONEMSKA PRETVORBA

V tem koraku postopka sinteze govora preslikujemo vsak grafem v enega ali več od 33 možnih fonemov. *Govorec* uporablja slovar s 194.000 vnosi [9]. To je 18.000 osnovnih besed v vseh možnih oblikah. Slovar vsebuje pare: [beseda zapisana v grafemski obliki, izgovorjava besede (fonemski zapis)]. Velikost slovarja nam omogoča, da najdemo večino besed ter tako ne delamo napak pri pretvorbi, kar se odraža v naravnejši izgovorjavi.

Če besede ne najdemo v slovarju, uporabimo množico pravil za pretvorbo [3, 4]. Pravila se vedno nanašajo na pretvorbo danega grafema v ustrezne foneme, pri pretvorbi vedno upoštevamo še okolico grafema, kar prispeva k natančnejši pretvorbi. Pravila se uporabljajo hierarhično — od najbolj specifičnih k splošnejšim.

Rezultat grafemsko-fonemske pretvorbe je množica stavkov, razdeljenih na besede, ki so zapisane v fonemski obliki in vsebujejo mesto ter tip naglasa.

2.3 DOLOČANJE PROZODIČNIH PARAMETROV

Ta modul vsakemu fonemu določi trajanje in višino osnovnega tona glede na naglas in mesto naglasa v besedi. Na začetku vsakemu fonemu priredimo osnovno trajanje in frekvenco, nato pa ju postopoma spreminjamo do končne vrednosti.

Govorec uporablja superpozicijski model za določanje osnovne frekvence. Potek osnovnega tona v intonacijskem segmentu je definiran kot vsota globalne (nosilne komponente) in posameznih lokalnih komponent [5]. Globalna komponenta je določena z vrsto intonacijskega segmenta in položaja ter dolžine intonacijskega segmenta v sestavljenih oblikah povedi. Lokalna komponenta je definirana za poudarjene besede (naglašene zloge) v intonacijskem segmentu. Pri tonemskem naglaševanju ton znotraj naglašenega zloga narašča, po dosegu tonskega vrha pa začne padati v odvisnosti od vrste besede (bariton/oksiton) in vrste naglasa (akut/cirkumfleks) [2].

Najprej se lotimo problema na nivoju besede. Upoštevamo število zlogov, vrsto besede (bariton, oksiton) ter mesto in vrsto naglasa (akut, cirkumfleks). Glede na relativni položaj posameznega zloga glede na mesto naglasa posebej obravnavamo prednaglasni, naglasni in ponaglasni zlog [7].

Stavčno intonacijo oblikujemo glede na vrsto povedi (povedna, vprašalna, vzklična). Uporabimo enega od treh značilnih potekov ovojnice osnovne frekvence [4, 7].

Nazadnje določimo še premore v besedilu. Premori so na mestih ločil, na mestih ritmičnih delitev ter pred vezniki.

2.4 ZDRUŽEVANJE DIFONOV

Osnovne govorne enote, ki jih uporablja *Govorec*, so difoni. Difon je sklop dveh sosednjih fonemov. Sestavljen je iz druge polovice prvega in prve polovice drugega fonema. Difon vsebuje informacijo o glasovnem prehodu prvega fonema v drugega. Baza vsebuje 1156 difonov. Vsak difon ima označeno sredino prehoda, na zvenečih delih pa so označene tudi periode osnovne frekvence.

Difone združujemo s postopkom TD-PSOLA (Time Domain Pitch Synchronous Overlap-Add synthesis) [1], ki temelji na razčlenjevanju signala v zaporedje prekrivajočih se kratkotrajnih signalov. Metoda dovoljuje kvalitetne spremembe frekvence in trajanja govornega signala neposredno na časovnem signalu in je računsko poceni.

Lepljenje osnovnih period signala poteka s pomočjo Hanningovega okna, ki ima vrh na mnogokratniku osnovne periode signala. Okna so vedno daljša od osnovne periode, tako se sosednja okna prekrivajo in pri lepljenju signalov seštevajo.

Ob povezovanju prav tako upoštevamo spremembo trajanja in osnovne frekvence signala. Signal podaljšamo tako, da vrinemo zadostno število osnovnih period, če je signal periodičen, v nasprotnem primeru pa vrinemo šum (*š*) ali tišino (*p, t, k*). Frekvenco spremenimo tako, da dolžino periode ustrezno spremenimo.

Pomanjkljivost algoritma so spektralne nezveznosti na mestih lepljenja signala. Nezveznosti poskušamo odpraviti z linearno interpolacijo signala v časovnem prostoru.

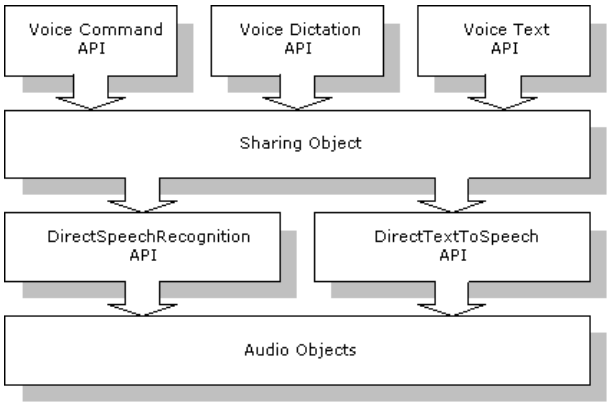
3. MICROSOFT SPEECH API (SAPI)

Microsoft Speech API (Application Programming Interface) definira standardno pot za vključitev govora (sinteza, razpoznavanje, narekovanje) v aplikacijo [6]. SAPI sledi arhitekturi OLE ter COM (Component Object Model). Standard definira skupek vmesnikov (interfaces), ki jih mora podpirati sintetizator ali razpoznavalec govora, združljiv s SAPI. Preko teh vmesnikov se aplikacija in sistem sporazumevata, si izmenjujeta sporočila ter se odzivata na dogodke.

Glavni vmesniki SAPI (slika 2):

- *Voice Command*: visokonivojski API za razpoznavanje govora. Omogoča izvrševanje ukazov v Oknih na podlagi govora.
- *Voice Dictation*: visokonivojski API za narekovanje aplikacijam. Uporabniku omogoča narekovanje besedila in njegovo prikazovanje.
- *Voice Text*: enostaven visokonivojski API za sintezo govora (text-to-speech).
- *DirectSpeechRecognition*: nižjenivojski vmesnik, ki omogoča popoln nadzor nad sistemom za razpoznavanje govora

- *DirectTextToSpeech*: nižjenivojski vmesnik za popolno kotrolo sintetizatorja govora. Omogoča dostop na najnižjem nivoju in nadzor nad predvajanjem, stilom govora in kvaliteto glasu.
- *Audio Objects*: vhodno-izhodni vmesniki za kontrolo mikrofonov, slušalk, zvočnikov in datotek.



Slika 2: arhitektura SAPI

Sintetizator govora v zgornjem diagramu predstavlja škatla *DirectTextToSpeech*. Sintetizator ima povezovalno vlogo med višjenivojskim Voice Text API in objekti, ki omogočajo predvajanje sintetiziranega govora. Slika 3 prikazuje položaj sintetizatorja v shemi ter pot komunikacije med objekti.

3.1 ŽIVLJENSKI CIKEL SAPI SINTETIZATORJA

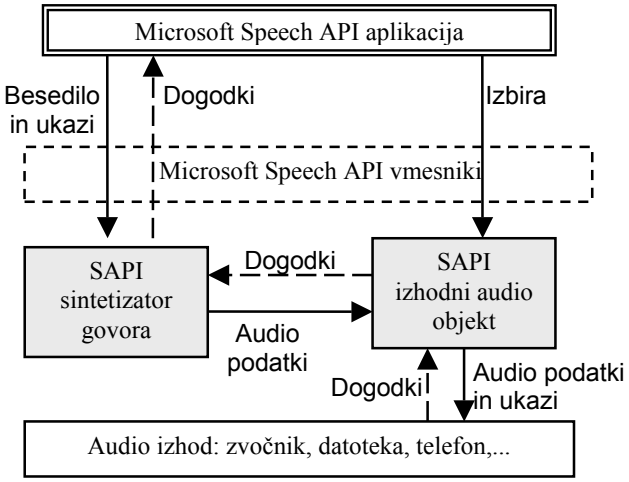
Aplikacija, ki želi uporabljati sintetizator govora, združljiv s standardom SAPI [6], mora najprej ustvariti zvočni objekt (*audio-destination*), ki bo sprejemal sintetizirani govorni signal. Ponavadi je to objekt, ki pošilja podatke do zvočne kartice, vendar to v splošnem ni nujno.

Program nato preko naštevnega objekta ugotovi prisotnost vseh sintetizatorjev govora v sistemu. Na podlagi tega potem glede na jezik in ostale atribute sintetizatorja izbere željeni sistem za sintezo govora. Sedaj aplikacija ustvari instanco željenega sistema in mu poda zvočni objekt.

V naslednjem koraku se sintetizator sporazume z zvočnim objektom o obliki kodiranja digitalnega zvočnega signala. Sistem za sintezo govora hkrati kreira še objekt *audio-destination-notification-sink*, preko katerega *audio-destination* sporoča dogodke in svoje stanje sistemu za sintezo govora.

Program lahko registrira tudi *main-notification-sink*, preko katerega od sintetizatorja prejema podatke o trenutnem stanju zvočnega objekta ter o položaju ustnic, kar lahko uporabimo za animacijo govorečih ust. Aplikacija je sedaj pripravljena in lahko sistemu pošlje besedilo, sintetizator generira zvočni signal in ga pošilja objektu *audio-destination*.

Aplikacija lahko registrira še drugi obveščevalni objekt *buffer-notification-sink* za vsako besedilo posebej. Preko njega sintetizator sporoča aplikaciji trenutno govorjeni fonem. Prav tako lahko v besedilo vrinemo zaznamke, ki se uporabljajo za sinhronizacijo med aplikacijo in sintetizatorjem. Takoj ko sistem začne govoriti besedilo na mestu zaznamka, o tem obvesti aplikacijo.



Slika 3: delovanje SAPI

3.2 VMESNIKI SAPI

Govorec je združljiv s Microsoftovim standardom SAPI. V osnovi je to komponenta COM, ki implementira zahtevane vmesnike in preko njih ponuja možnost sinteze govora.

Vsi vmesniki, ki imajo opraviti z besedilom, obstajajo v dveh inačicah: Unicode in ASCII. Obe verziji imata enake metode s to razliko, da so pri eni znakovni nizi tipa Unicode (16 bitov na znak), pri drugem pa tipa ASCII (8 bitov na znak). Implementacija obeh mora biti ločena.

Sledi kratek opis večine pomembnejših vmesnikov in njihovih metod ter funkcionalnosti, ki jih ponuja *Govorec* na razpolago zunanjim aplikacijam.

3.2.1 ITTSCentral

ITTSCentral je osrednji vmesnik, preko katerega aplikacija nadzoruje in upravlja s sintetizatorjem. Metode omogočajo aplikaciji pošiljanje besedila, ki ga želimo prebrati. Prav tako je mogoče vrivati kontrolne značke medtem, ko se besedilo bere. To prinaša aplikaciji neposreden nadzor nad lastnostmi branega besedila, ko je le-to že v procesu sinteze govora.

ITTSCentral ponuja tudi metode za nadzor predvajanja sintetiziranega govora (start, stop, pavza). Z metodami vmesnika lahko dano besedilo pretvorimo v mednarodno fonetično besedo Unicode IPA (International Phonetic Alphabet) in ugotovimo lastnosti ter stanje sistema.

Program lahko z metodami, ki jih ponuja vmesnik, zahteva registracijo vseh objektov, namenjenih obveščanju aplikacije o stanju branega besedila, položaju ust, doseženih zaznamkih in o času predvajanja določenega fonema.

3.2.2 ITTSAttributes

Metode vmesnika *ITTSAttributes* ponujajo dostop do atributov sintetizatorja. Omogočajo nastavljanje višine, hitrosti in glasnosti govora. Hitrost in višina sta lastnosti sintetizatorja, glasnost pa objekta *audio-destination.*

3.2.3 ITTSDialogs

Vmesnik omogoča sintetizatorju, da prikaže okna s podrobnejšimi nastavitvami in informacijami, ki so odraz posameznega sistema za sintezo govora.

3.2.3 ITTSNotifySink in ITTSBufNotifySink

ITTSNotifySink obvešča aplikacijo o dogodkih, povezanih s procesom predelave besedila v govor: o spremembi katerega od atributov (*ITTSAttributes*), času začetka/konca predvajanja govora, o trenutno govorjenem fonemu in drugih informacijah, ki služijo za animacijo premikanja ustnic.

Vmesnik *ITTSBufNotifySink* sporoči aplikaciji začetek in konec obdelave besedila ter trenutno govorjeno besedo. V primeru, da besedilo vsebuje zaznamke, se ob začetku branja zaznamovanega besedila sproži dogodek. Tako lahko aplikacija vedno ugotovi, kateri je trenutno govorjeni fonem. Zaznamki so mehanizem za sinhronizacijo, saj imata tako sintetizator kot *audio-destination* objekt svoj medpomnilnik.

Govorec sintetizira pet besed vnaprej in tako zagotovi nemoten dotok podatkov v zvočni objekt, ki ima tudi svoj medpomnilnik, v katerem je prostora za sekundo govora.

3.2.4 IAudio, IAudioDest in IAudioDestNofiySink

Vmesnik *IAudio* omogoča nadzor nad zvočnim izhodom. Ponuja funkcije za začetek in konec prenosa ter nastavljanje glasnosti in formata zvočnega zapisa.

V *IAudioDest* sintetizator pošilja zvočni signal. Vmesnik ima metode za pošiljanje podatkov, vstavitev zaznamka in ugotavljanje zasedenosti zvočnega medpomnilnika.

IAudioDestNofiySink uporabimo za sporočanje o začetku ali koncu predvajanja govora. Sintetizator dogodek posreduje do aplikacije. Vmesnik ima tudi metodo, ki se proži v rednih presledkih in obvešča sintetizator o prostem notranjem medpomnilniku objekta *IAudioDest;* tako zagotovimo tekoče branje, saj zvočnemu objektu nikoli ne zmanjka podatkov.

3.2.5 ITTSEnum in ITTSFind

Vmesnika ustvarimo na začetku, ko aplikacija izbira med sintetizatorji na sistemu. Pričakuje se, da so sintetizatorji, združljivi s SAPI, vpisani v register. Operacijski sistem zbudi vsak sintetizator posebej, ki preko *ITTSEnum* pove svoje lastnosti, jezik in obliko zvočnega zapisa, ki jih podpira. Ko aplikacija najde željeni sintetizator, začne z ustvarjanjem in inicializiranjem ostalih vmesnikov SAPI.

3.2.5 Kontrolne značke in zaznamki

Kontrolne značke so posebni kontrolni deli besedila, npr.: `\Ctx=e-mail\`. Vsebujejo ukaze in podrobneje opisujejo način izgovorjave besedila (prozodijo). Omogočajo spreminjanje hitrosti, glasnosti in barve branja.

Zaznamek je posebna značka oblike `\Mrk=število\`. Sistem začne brati zaznamovani del besedila in to preko vmesnika *ITTSBufNotifySink* takoj sporoči aplikaciji.

4. ZAKLJUČEK

Predstavili smo *Govorec,* sistem za branje slovenskih besedil.

Sinteza govora v Govorcu poteka po bolj ali manj standardnih postopkih. Za večino transformacij in nastavitev uporabljamo hierarhična pravila. Uporaba grafemsko-fonemskega slovarja je močno izboljšala razumljivost sintetiziranega govora, saj ni več napak pri naglaševanju ter izbiranju različnih oblik samoglasnikov (ozki, široki). Sistem sicer še vedno dela napake, vendar so napake stalne, tako da se poslušalec kmalu privadi in lažje razume umeten govor.

Govorec je združljiv z Microsoftovim standardom Speech API, zato je mogoče sistem brez težav vključiti v množico že napisanih aplikacij. Aplikacije lahko »prisilimo«, da berejo slovensko, čeprav tega v osnovi ne znajo. Ta lastnost poenostavi uporabo sistema slepim in slabovidnim uporabnikom, saj lahko sistem enostavno integrirajo v programe, ki jih uporabljajo [8]. Tako imajo možnost poslušati svoj priljubljeni elektronski časopis v slovenskem jeziku.

Literatura

[1] E. Moulines, F. Charpentier. *Pitch-Synchronous Waveform Processing Techniques for Text-to-Speech Synthesis Using Diphones*. Speech Communication 9, pp. 453–467, 1990.

[2] J. Gros, N. Pavešić, S. Dobrišek, M.Erpič, B. Gorenc, A. Rakar, T.Šef, V. Vračar, F. Mihelič. *Sistem za sintezo slovenskega govora*. Proc. ERK'95, Slovenia Section IEEE Conference, Portorož, Slovenia, 1995.

[3] J. Toporišič, *Slovenska slovnica.* Založba Obzorja, Maribor. 1984.

[4] T. Šef, *Sistem za govorno posredovanje obvestil o prostih delovnih mestih*. Magistrska naloga, Fakulteta za elektrotehniko in računalništvo, Univerza v Ljubljani, 1998.

[5] H. Fujisaki, S. Ohno, *Analysis and Modeling of Funda–mental Frequency Contour of English Utterances*. Proc. EUROSPEECH'95, vol. 2, pp. 634–637, Philadelphia, 1996.

[6] Microsoft, *Speech API*, *http://www.microsoft.com/speech*

[7] J. Gros, *Samodejno tvorjenje govora iz besedil.* Doktorska disertacija, Fakulteta za elektrotehniko in računalništvo, Univerza v Ljubljani, 1997.

[8] Henter-Joyce, *Jaws*. *http://www.hj.com/JAWS/JAWS.html*

[9] T. Šef, *Analiza besedila v postopku sinteze govora*. Doktorska disertacija, Fakulteta za elektrotehniko in računalništvo, Univerza v Ljubljani, 2000.