

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 771

**SUSTAV ZA AUTOMATSKU GOVORNU
NAJAVU DOGAĐAJA U ZRAČNOM
PROMETU**

Miłosz Chmura

Zagreb, lipanj 2009.

<IZVORNIK>

*Zahvaljujem Martini,
za njezin glas.*

Sadržaj

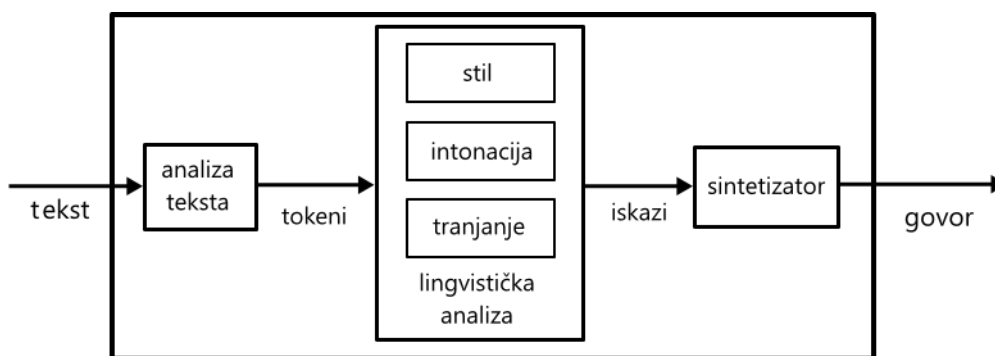
1. Uvod u sintezu govora	1
2. Postavljanje problema i pristup njegovom rješenju	4
3. Stvaranje novog glasa	6
Pregled domene rasporeda letova	7
Glavna ASP.NET aplikacija – prvi dio	8
Prikupljanje riječi iz domene	9
Slaganje rečenica koje opisuju let	9
Tokenizacija leta	11
Priprema rečenica za snimanje	12
Snimanje rečenica	13
Generiranje sintetizatora u Festivalu	14
Ukratko o Festival sustavu	14
Postavljanje okoline za generiranje sintetizatora	14
Generiranje sintetizatora	15
Stvoreni sintetizator	19
4. Sustav za interaktivan prikaz rasporeda	20
Prilagodba i povezivanje sintetizatora s vanjskim svijetom	21
Prilagodba	21
Povezivanje	22
Glavna ASP.NET aplikacija – drugi dio	24
Korsničko interaktivno sučelje	28
Pregled komponenata	28
Način rada sučelja i njegova funkcionalnost	29
5. Rezultat rada	34
Primjer korištenja	34
Dodavanje novog glasa	35
Problemi	36
Poboljšanja	36
6. Zaključak	37
7. Literatura	38
Sažetak	39
Abstract	39
Dodatak A: Lista rečenica snimljenih za sintetizator	40

1. Uvod u sintezu govora

Sinteza govora je umjetna proizvodnja glasa. Računalni sustavi koji se koriste za tu svrhu zovu se sintetizatori, a mogu biti implementirani ili softverski ili hardverski.

Tekst-u-govor sustav (eng. *Text-to-speech* ili *TTS*) pretvara nizom postupaka običan tekst, napisan u nekom jeziku, u ljudski govor, što je obrnuti proces od prepoznavanja govora. Jednostavna shema TTS sustava prikazana je na shemi *Shema 1.1*. Sustav se sastoji od:

- analize teksta i njegove pretvorbe na tokene koji se nalaze u bazi podataka ili leksikonu,
- lingvističke analize: stil pisanja (eng. *phrasing*), intonacija i trajanje; izlaz te analize su govorni iskazi koji se mogu sintetizirati u sintetizatoru,
- sintetizatora koji generira zvučne valove.



Shema 1-1: Proces pretvaranja teksta u govor

Postoji pet podjela s obzirom na način sinteze glasa:

1. konkatencijska metoda: u njoj su jedinice glasa različitih veličina snimljenog teksta pohranjene u bazi podataka. Te jedinice spajaju se s mogućim preklapanjem stvarajući govorni signal. Konkatenacijsku metodu možemo podijeliti na:
 - sintezu s odabirom jedinica (eng. *unit selection*): jedinice mogu biti samostalne ili kombinacija: fonema (najmanja jedinica govora), difona (dva uzastopno spojena fonema), silaba, riječi, fraza i rečenica.
 - difonsku sintezu: poseban slučaj prethodne sinteze, gdje je jedinica sinteze difon. Smatra se da je difon, a ne fonem, najmanja smisaona jedinica glasa nekog jezika, zbog različitog konteksta u kojem se javlja neki fonem.
 - sintezu ograničene domene: poseban slučaj sinteze s odabirom jedinica, gdje je jedinica riječ i/ili cijela fraza.
2. formantna metoda: koristi akustički model sa sljedećim parametrima: (i) fundamentalna frekvencija F_0 ili visina glasa, (ii) fonacija ili oscilacije dijelova grla koje proizvode foneme, (iii) šum. Navedeni parametri na osnovi pravila stvaraju zvučne valove.
3. artikulacijska metoda: sinteza je osnovana na modelu filtera ljudskog govornog trakta koji može biti opisan u Z-domeni formulom: $Govor(z) = Uzbuda(z) * VokalniTrakt(z) * Usne(z)$.
4. metoda bazirana na skrivenim Markovljenim modelima (eng. *Hidden Markov Model* ili *HMM*): koristi parametre: (i) frekvencijski spektar, (ii) F_0 , (iii) prozodija (ritam, akcent, intonacija) govora.
5. *sinewave* metoda: u sintezi su glavni pojasevi energije formantata (određene frekvencije koje karakteriziraju pojedine glasove) zamijenjeni sa čistim tonovima.

Procjenu sintetiziranog glasa možemo podijeliti na:

- kvalitetu govora: razumljivost (eng. *intelligibility*) i prirodnost ili sličnost sintetiziranog govora sa stvarnim izgovorom,
- cijenu sinteze: kompleksnost i zauzeće u memoriji,
- kašnjenje: izvođenje sinteze u stvarnom ili *off-line* vremenu.

U povijesti su neki pokušaji izgradnje sustava za reprodukciju čovjekovog glasa počeli već u XVIII. stoljeću, ali prvi veći uspjeh možemo datirati na 1930. godinu, kada je Bell Labs izgradio *VOCODER* i na 1936. godinu, kada je poboljšan *VODER* bio predstavljen na svjetskom sajmu u New Yorku.

1950tih godina bili su poduzeti raznoliki pokušaji izgradnje sintetizatora, pomoću analognih mehaničkih filtara, međutim tek s pojavom digitalnih računala 1960tih, ovo područje počelo se ozbiljno razvijati. Govorni sustavi koristili su uglavnom artikulacijsku metodu, a tek kasnih 1990tih godina, sa značajnim porastom snage računalnih resursa, počelo se prebacivati na konkatencijsku metodu.

2. Postavljanje problema i pristup njegovom rješenju

Zadatak završnog rada je programsko ostvarenje sintetizatora hrvatskog jezika za izgovaranje rasporeda avionskih letova na aerodromu Zagreb-Pleso. Uz sintetizator želi se ostvariti i sustav koji bi omogućio, u stvarnom vremenu, interaktivan prikaz rasporeda sa zvučnim iskazima svim korisnicima bez obzira na mjesto u kojem se nalaze i na platformu koju koriste.

Završni rad riješen je u dva glavna koraka: prvi je izgradnja sintetizatora (poglavlje 3. *Stvaranje novog glasa*), a drugi izgradnja sustava za interaktivni prikaz rasporeda (poglavlje 4. *Sustav za interaktivan prikaz rasporeda*).

Sintetizator za novi glas napravljen je na Linux operacijskom sustavu u programskoj okolini *The Festival Synthesis Speech System*¹ korištenjem *Festvox* skripti². Razlog korištenja tih alata je otvorenost sustava za modifikacije i besplatne licence za korištenje. Za generiranje sintetizatora rasporeda avionskih letova koristi se postupak opisan u *Festvox* dokumentaciji, u poglavlju 5. *Limited domain synthesis* [4], koji se primjenjuje za ograničenu domenu.

Napisane su dvije aplikacije u *C#* programskom jeziku: (i) aplikacija za prikupljanje svih riječi iz domene, (ii) aplikacija za provjeru pokrivenosti riječi u snimanim rečenicama i stvaranje datoteke sa tekstualnim iskazima. U tu svrhu koristen je *Visual Studio 2008* ili *VS2008*, jer se u njemu može brzo, lako i pregledno napisati objektno-orijentiran programski kôd.

Za snimanje rečenica korišten je besplatan program *Audacity*³ koji je lako upotrebljiv i prilagođen za brzo snimanje i spremanje više rečenica.

Sljedeći alat korišten za stvaranje novog glasa je *Emu*⁴ koji služi za ručno ispravljanje labela na snimljenim rečenicama, a korišten je zbog besplatne licence.

¹ <http://www.cstr.ed.ac.uk/projects/festival/>

² <http://festvox.org/>

³ <http://audacity.sourceforge.net/>

Sustav za interaktivan prikaz koristi više programskih alata koji su povezani internetskom vezom, običnim *HTTP* protokolom. Glavni sustav implementiran je u VS2008 kao *ASP.NET* aplikacija. Za povezanost *ASP.NET* servera i Linux servera, na kojim se nalazi sintetizator, koriste se *.scm* skripte te *.php* skripte. Budući da sintetizator u Festivalu generira nekomprimirane *.wav* snimke, na Linuxu se koristi besplatan alat *lame*⁵ za komprimiranje u približno deset puta manje *.mp3* datoteke.

Korisnička aplikacija koristi statičan HTML kostur, zajedno sa *.css* datotekom, koja definira izgled stranice. Povezanost *ASP.NET* severa sa korisničkom HTML stranicom i interaktivnost stranice ostvaruje se pomoću *JavaScript* tehnologije. *JavaScript* ostvaruje korisniku nevidljive pozive za prikupljanje novih podataka te ažurira i kontrolira interaktivnu tablicu rasporeda.

Sljedeći neophodan alat je *SoundManager*⁶ izveden u *Adobe Flash* i *JavaScript* tehnologiji, koji služi za reproduciranje *.mp3* snimki. Razlog korištenja tog alata je njegova otvorenost za modifikacije i kontrolu preko *JavaScript* sučelja te besplatna licenca.

Zadnji bitan alat, koji značajno olakšava pisanje HTML, CSS i *JavaScript* kôda, je *Notepad++*⁷, za koji također nije potrebno kupiti licencu.

⁴ <http://emu.sourceforge.net/>

⁵ <http://lame.sourceforge.net/>

⁶ <http://www.schillmania.com/projects/soundmanager2/>

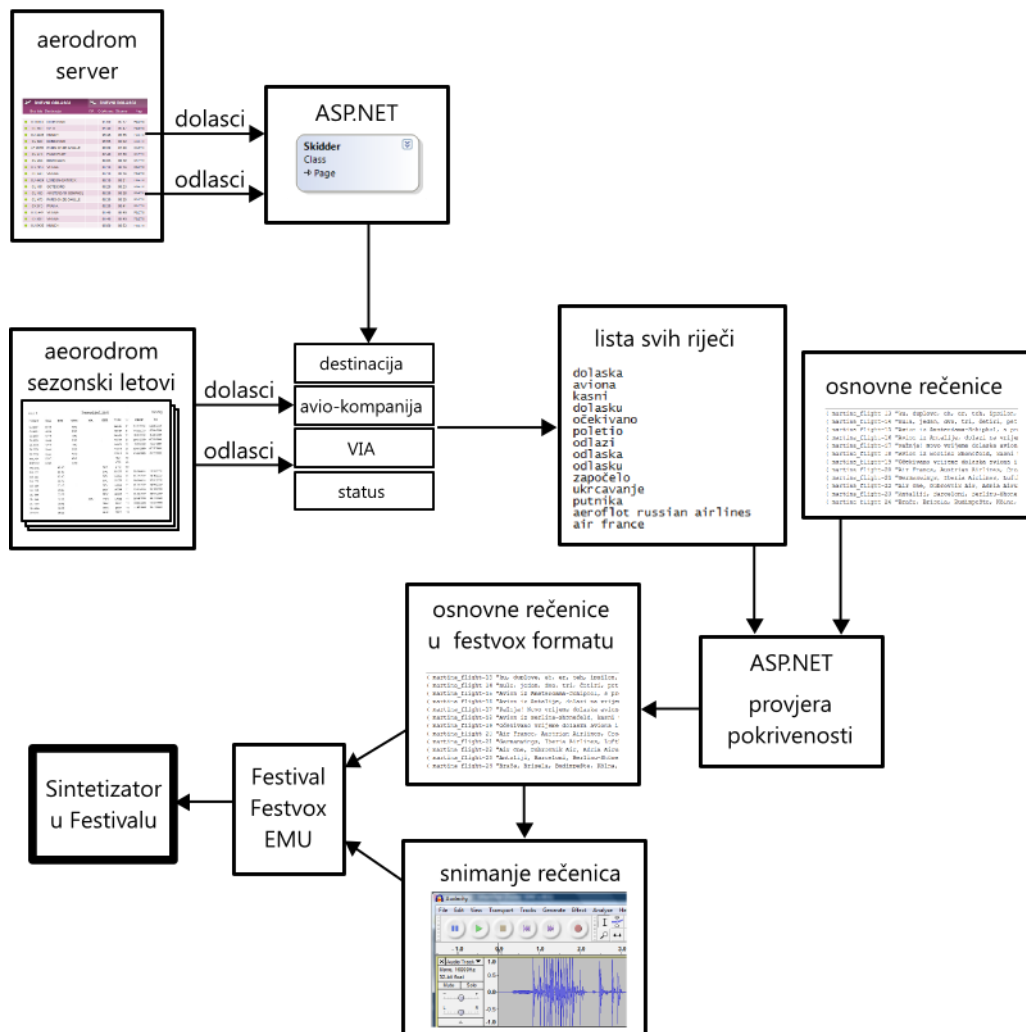
⁷ <http://notepad-plus.sourceforge.net/>

3. Stvaranje novog glasa

Kod stvaranja novog sintetizatora potrebno je proći kroz nekoliko faza.

Pojedine faze su vidljive na shemi *Shema 3.1*:

- prikupljanje riječi iz domene,
- smišljanje rečenica za čitanje rasporeda,
- priprema rečenica za snimanje,
- snimanje rečenica,
- razvoj glasa.



Shema 3-1: Stvaranje sintetizatora za raspored avionskih letova

Pregled domene rasporeda letova

Avionski raspored sastoji se od tablice dolaznih i odlaznih avionskih letova koji su opisani: (i) brojem leta, (ii) destinacijom, (iii) presjedanjem ili *via*, (iv) očekivanim vremenom dolaska/odlaska, (v) stvarnim vremenom dolaska/odlaska, (vi) statusom leta. Primjer rasporeda dolaska, s nekim izmišljenim kombinacijama, nalazi se u tablici *Tablica 3.1*.

Tablica 3.1: Primjer rasporeda avionskih letova

Broj leta	Destinacija	VIA	Očekivano	Stvarno	Status
AF 2055	PARIS-CH. DE GAULLE	ZAD	06:40		
OS 7052	VIENNA		17:10	17:50	KASNI
OU 484	GOTEBORG		08:23	10:23	NEW TIME
OU 4494	LONDON- GATWICK		10:22	10:22	BOARDING
TK 9932	ISTANBUL		03:32	03:25	SLETIO

Broj leta sastoji se od: (i) troslovnog *IATA*⁸ ili četvoroslovnog *ICAO*⁹ kôda, koji jedinstveno identificira avionsku kompaniju, (ii) nekog broja.

Destinacija je na engleskom jeziku, a sastoji se od imena grada, npr. *ISTANBUL*, a ponekad i dodatno od imena aerodroma odvojen crticom, npr. *PARIS-CH. DE GAULLE*.

Polje *VIA* je označeno troslovnim *IATA* kôdom, koji jedinstveno određuje aerodrom. To polje može ostati prazno ako nema presjedanja.

Očekivano vrijeme je vrijeme u rasporedu u kojem bi avion trebao sletiti/poletiti, a stvarno vrijeme je vrijeme koje pokazuje kad je avion zapravo sletio/poletio ili koje mu je novo vrijeme u rasporedu. Kada stvarno vrijeme nije još poznato, to polje ostaje prazno.

⁸ International Air Transport Association

⁹ International Civil Aviation Organization

Postoji ograničen broj statusa leta: (i) SLETIO, (ii) POLETIO, (iii) NEW TIME ili novo vrijeme, (iv) NA VRIJEME, (iv) BOARDING ili početak ukrcavanja putnika u avion, (v) KASNI.

Glavna ASP.NET aplikacija – prvi dio

Napisana ASP.NET aplikacija *FlightToSentence*, služi za dvije svrhe: (i) aplikacija za prikupljanje podataka opisana u ovom poglavlju – *Skidder.aspx*, (ii) servis za dohvaćanje novih/sljedećih avionskih letova zajedno sa njezinim sintetiziranim govornim iskazima, koji su opisani u sljedećem poglavlju – *more.aspx*.

Oba dijela aplikacije koriste zajedničke klase koje se nalaze u folderu *Parser*, a to su: *DataGetter.cs*, *Fligth.cs*, i *Common.cs*.

U klasi *Common* nalaze se zajednički postupci koje koriste klase *DataGetter* i *Flight* za dohvaćanje i parsiranje stranice.

Klasa *DataGetter* služi za: (i) skidanje odgovarajuće stranice s trenutnim rasporedom sa servera aedroma¹⁰, (ii) parsiranje redaka koji opisuju pojedini let, (iii) stvaranje liste *Flight* klase ili *List<Flight>*.

U klasi *Flight* u konstruktoru odvija se daljnje parsiranje predanog retka, i podaci se zapisuju u *string* i *DateTime* polja. Ta klasa ima i dodatna polja: (i) *string soundID* za identifikaciju govornog iskaza, (ii) *bool isArrival* za razliku između dolaska i odlaska, (iii) *bool isError* za označavanje pogrešno sparsiranog leta. Budući da u rasporedu ne postoji informacija o datumu, dodatno se, tijekom parsiranja, pronalazi trenutni datum koji je u Zagrebu, a taj podatak se sprema uz vrijeme leta. Na aerodromskom serveru raspored letova potpuno se briše svaki dan nakon ponoći.

¹⁰ <http://www.zagreb-airport.hr/>

Prikupljanje riječi iz domene

Da bi se mogao napraviti sintetizator ograničene domene, potrebno je saznati sve riječi koje će se pojavljivati tijekom korištenja sintetizatora zbog toga što će se jedino riječi iz domene moći sintetizirati.

Prikupljanje novih riječi napravljeno je na dva načina: (i) iz dobivenih ispisa svih letova aerodroma u ljetnoj sezoni, (ii) pomoću ASP.NET aplikacije *Skidder.aspx*, koja skida, parsira te sprema avionske letove sa servera aerodroma.

Slaganje rečenica koje opisuju let

Kod izgovora avionskih letova postoji više kombinacija pojavljivanja polja via, vremena i statusa. Osnovna zamisao izgovora nekog leta je da se izgovori cijela informacija sadržana u retku rasporeda. Kod generiranja rečenica treba paziti i na vremena i padeže izgovaranih destinacija.

Sljedeći dijelovi teksta pojavljuju se u rečenici ako postoje odgovarajući podaci:

- `viaPart`: „s presjedanjem u:” + destinacija u Lokativu
- `airlinesPart`: „avionske kompanije:” + avionska kompanija u Nominativu
- `flightNumberPart`: „s brojem leta:” + slovčani broj leta

Za svaki let mora postojati vrijeme: očekivano ili stvarno. Za izgovor rečenice prvenstveno se koristi stvarno vrijeme jer je ono bitnije od onog očekivanog.

S obzirom na statuse letova generiraju se sljedeće rečenice za dolazak:

- SLETIO: „Avion iz:” + destinacija u Genitivu + `viaPart` + `airlinesPart` + `flightNumberPart` + „sletio je u:” + deklinirano vrijeme
- NA VRIJEME: „Avion iz:” + destinacija u Genitivu + `viaPart` + `airlinesPart` + `flightNumberPart` + „dolazi na vrijeme u:” + deklinirano vrijeme

- NEW TIME: „Pažnja! Novo vrijeme dolaska aviona iz:” + destinacija u Genitivu + viaPart + airlinesPart + flightNumberPart + „je:” + vrijeme u Nominativu
- KASNI: „Avion iz:” + destinacija u Genitivu + viaPart + airlinesPart + flightNumberPart + „kasni u dolasku. Novo vrijeme dolaska je:” + vrijeme u Nominativu
- bez statusa i za nepoznati status: „Očekivano vrijeme dolaska aviona iz:” + destinacija u Genitivu + viaPart + airlinesPart + flightNumberPart + „je:” + vrijeme u Nominativu

S obzirom na statusse letova generiraju se sljedeće rečenice za odlazak:

- SLETIO: „Avion za:” + destinacija u Akuzativu + viaPart + airlinesPart + flightNumberPart + „poletio je u:” + deklinirano vrijeme
- NA VRIJEME: „Avion za:” + destinacija u Akuzativu + viaPart + airlinesPart + flightNumberPart + „odlazi na vrijeme u:” + deklinirano vrijeme
- NEW TIME: „Pažnja! Novo vrijeme odlaska aviona za:” + destinacija u Akuzativu + viaPart + airlinesPart + flightNumberPart + „je:” + vrijeme u Nominativu
- KASNI: „Avion za:” + destinacija u Akuzativu + viaPart + airlinesPart + flightNumberPart + „kasni u odlasku. Novo vrijeme odlaska je:” + vrijeme u Nominativu
- BOARDING: „Započelo je ukrcavanje putnika u avion za:” + destinacija u Akuzativu + viaPart + airlinesPart + flightNumberPart + „Vrijeme odlaska je:” + vrijeme u Nominativu
- bez statusa i za nepoznati status: „Očekivano vrijeme odlaska aviona za:” + destinacija u Akuzativu + viaPart + airlinesPart + flightNumberPart + „je:” + vrijeme u Nominativu

Zbog padeža koji se koriste u rečenicama potrebno je, uz prikupljene riječi, napisati potrebne padeže za svaku nađenu destinaciju. Za prikupljene IATA i ICAO

kôdove iz broja leta potrebno je pronaći naziv avionske kompanije. Te informacije spremljene su u sljedeće datoteke:

- *AirportCases.txt*: svaki redak sadrži destinaciju koja se pojavljuje u rasporedu, njezin IATA kôd i sve potrebne padeže.
- *IATAAirports.txt*: svaki redak sadrži troslovni IATA kôd svakog aerodroma i njegov engleski naziv.
- *airlines.iata.txt* i *airlines.icao.txt*: svaki redak sadrži dvoslovni IATA kôd ili troslovni ICAO kôd avionske kompanije i naziv avionske kompanije.

Tokenizacija leta

Sintetizator za ograničenu domenu očekuje samo riječi iz te domene, takozvane tokene i jedino te je u stanju pretvoriti u govorni iskaz. Metoda `Flight.ToSentence()` koja tokenizira jedan let je smještena u ranije spomenutoj klasi `Flight`. Ova metoda vraća `string`, u kojem se nalaze jedino tokeni iz ograničene domene, koji se direktno predaje sintetizatoru.

Za generiranje i tokenizaciju ranije opisanih rečenica koristi se niz metoda:

1. `DestinationNameToCasedSentence()`: u *AirportCases.txt* pronalazi se destinacija i vraća odgovarajući padež. Ako destinacija nije pronađena, sva slova koja ona sadrži pretvaraju se u oblik kako se izgovaraju te se vraćaju odvojena zarezom.
2. `DestinationCodeToCasedSentence()`: u *AirportCases.txt* pronalazi se IATA kôd i vraća odgovarajući padež. Ako destinacija nije pronađena, sva slova koja kôd sadrži se pretvaraju u oblik kako se izgovaraju te se vraćaju odvojena zarezom.
3. `FlightNumberToAirlinesName()`: parsira se IATA/ICAO kôd, pretražuje se *airlines.iata.txt/airlines.icao.txt* datoteku i vraća se naziv avionske kompanije. Ako naziv nije pronađen, ignorira se dio rečenice `airlinesPart`.

4. `FlightNumberToSentence()`: pretvaraju se brojevi i slova u oblik kako se izgovaraju te ih se vraća odvojene zarezom.
5. `TimeToSentence()`: vrijeme se pretvara na način na koji se izgovara u odgovarajućem padežu. Razlika u padežu je samo kod izgovora „jedna minuta” i „jednu minutu”.

Svaki dio rečenice odijeljen je sa zarezom, a prije izgovora destinacije, avionske kompanije, broja leta ili vremena stavlja se dvotočka. Svrha tih simbola je označavanje pauze za sintetizator.

Primjer rečenice koja se šalje direktno sintetizatoru je: *„Avion iz: Verone, s presjedanjem u: Puli, avionske kompanije Croatia Airlines, s brojem leta: oh, uh, pet, tri, tri, dva, kasni u dolasku. Novo vrijeme dolaska je: petnaest sati, i pedeset jedna minuta.”*.

Priprema rečenica za snimanje

Ukupan broj riječi koje su prikupljene uljučujući sve moguće padeže je 303. Za sintetizator je potrebno smisliti rečenice kod kojih su izgovarane riječi u istom kontekstu kao i u stvarnim rečenicama, kako bi se osjetila što manja razlika u prozodiji kod stvarne sinteze. S druge strane potrebno je smisliti što manje rečenica koje bi pokrivale sve te riječi jer se snimanje treba napraviti odjednom. Razlog tome je dosta uočljiva različitost kod sinteze glasa između dva kruga snimanja. Dužina snimanja također ne smije biti preduga jer se glas govornika s vremenom umara, što opet može imati za posljedicu promjenu u prozodiji.

U ovom radu rečenice za snimanje pokrivaju izmišljene rečenice za: (i) vrijeme, (ii) brojeve, (iii) slova, (iv) dolazne rečenice, (v) odlazne rečenice, (vi) sve ostale rečenice u kojima se nalazi po nekoliko naziva avionskih kompanija i destinacija u svim potrebnim padežima odvojenim pauzom.

Rečenice su napisane ručno, a za provjeru pokrivenosti cijele domene, te generiranje datoteke u formatu pogodnu za generiranje sintetizatora, koristi se C# aplikacija *ToFestivalFormat*. Lista svih 44 osnovnih rečenica nalazi se u privitku *Dodatak A: Lista rečenica snimljenih za sintetizator*.

U izgeneriranoj datoteci s osnovnim rečenicama svaki redak sadrži osnovnu rečenicu u formatu identifikator i rečenica:

```
( martina_flight-01 "Prva recenica" )
```

Bitno je napomenuti da su bitni razmaci između pojedinih komponenti.

Snimanje rečenica

Za svaku osnovnu rečenicu potrebno je snimiti `.wav` datoteku s parametrima: mono, 16kHz, 32-bit float. Snimljenu rečenicu potrebno je spremiti u formatu: `ŠIDENTIFIKATOR.wav`, gdje je `ŠIDENTIFIKATOR` jednak onome u izgeneriranoj datoteci s osnovnim rečenicama. Bitno je napomenuti da `.wav` datoteke ne smiju sadržavati meta podatak, jer ih Festival neće biti u stanju pročitati.

Rečenice su bile snimane ženskim glasom (Martina) pomoću slušalica s običnim mikrofonom s funkcijom brisanja buke (eng. *noise cancelling*). Snimanje se odvijalo na laptopu isključenom iz vanjskog napajanja zbog smanjenja elektromagnetskog utjecaja. Da bi se što više smanjila pozadinska buka, snimanje je obavljeno u jedan sat ujutro, na tihom mjestu unutar automobila. Snimanje je trajalo oko sat i pol.

U stvarnosti je snimljeno oko 90 rečenica u kojima se vrlo često pojavljivala fraza „avion za“, „avion iz“ te „avionske kompanije“ za neke destinacije i nazive avionskih kompanija. Ponavljanje fraza loše je utjecalo na izradu sintetizatora, koji je preklapanjem riječi stvarao iritantnu jeku. Zbog toga su rečenice bile pretvorene u

manji broj rečenica, koje su sadržavale nekoliko destinacija ili naziva avionskih letova odvajanih pauzama.

Generiranje sintetizatora u Festivalu

Ukratko o Festival sustavu

Festival nudi općeniti *framework* za razvoj novih sintetizatora glasa i sintezu rečenica. Osnovna jedinica Festivala je govorni iskaz (eng. *utterance*) koji se sastoji od više komponenti kao što su: tekst, token, riječ, silaba, segment, intonacija, signal, te relacija između tih komponenata. Osnovni proces sinteze u Festival sustavu je primjenjivanje njegovih modula na govorne iskaze sve do primjene modula za generiranja zvučnog `.wav` signala.

Prednost Festivala je korištenje *SOID*¹¹ interpretatora skriptnog jezika *Scheme*¹², što omogućava podešavanje njegovih parametara bez ingerencije u izvorni kôd, na način da se primjenom postojećih modula može lakše stvoriti novi sintetizator glasa.

Postavljanje okoline za generiranje sintetizatora

U ovom radu korišten je virtualni stroj *VMware player*¹³ instaliran na Windows Vista operacijskom sustavu, na njemu je instaliran Linux *Centos-4.6-i386-server* operacijski sustav. Na tom sustavu pomoću *yum* programa instalirani su: *gcc*, *Festival*, *speech_tools* ili alati za zvuk od Festivala, te *Festvox*. Na ovakav način namješten Centos spreman je za postupak generiranja novog sintetizatora.

Da bi se Festival mogao lakše koristiti na virtualnom stroju potrebno je uključiti zvuk u Linuxu pomoću *amixer* programa. Za provjeru rada zvuka može se utipkati naredbu: `festival -b '(SayText "Hello world")'`, koja pokreće

¹¹ Scheme in One Defun

¹² funkcijski jezik, dijalekt Lispa

¹³ <http://www.vmware.com/>

Festival te sintetizira i reproducira govorni iskaz *default* glasom. Na disku priloženom ovom završnom radu nalazi se VMware *image* s postavljenim, prethodno opisanim, postavkama.

Sljedeći program koji je potreban tijekom daljnje opisanog postupka generiranja novog sintetizatora je EMU, koji se može skinuti ili za Windows ili za Linux. U ovom radu on je instaliran na Windows Vista operacijskom sustavu jer virtualni stroj Centos ne podržava grafičko sučelje koje je neophodno za njegovu uporabu.

Zadnja stvar, koju je potrebno napraviti prije što se krene na postupak generiranja, je povezivanje virtualnog stroja s Vistom radi razmjene podataka. Najjednostavniji način da se to ostvari je dijeljenje Vista foldera preko *Ethernet* mreže na sljedeći način:

- uključi dijeljenje nekog foldera u Visti, npr. folder *SHARED*
- u virtualnom stroju utipkaj sljedeće naredbe:

```
➤ mkdir /mnt/shared
➤ mount -t smbfs -o
  username=<VISTA_USERNAME>,password=<VISTA_PASSWORD>
  //$VISTA_IP/SHARED /mnt/shared
```

gdje je `$VISTA_IP` najvjerojatnije 192.168.8.1

- `/mnt/shared` je sada dijeljen sa *SHARED*

Generiranje sintetizatora

Sintetizator se generira postupkom opisanim u Festvox dokumentaciji , u poglavlju 5. *Limited domain synthesis* [4]. Za stvaranje sintetizatora potrebni su Festvox programi i skripte već ranije instalirane na Centos sustavu. Opisani postupak generira sintetizator s odabirom jedinica, koji su engleski difoni jedinstveno vezani uz pojedinu riječ. To označava da se, kod sinteze, neki difon iz jedne riječi ne smije upotrijebiti u drugoj riječi.

Kod stvaranja sintetizatora s odabirom jedinica, neophodno je korektno označavanje vremena u kojima se pojavljuje pojedina jedinica u snimci. Navedeni postupak opisuje način u kojem se labeliranje difona provodi automatski, na način da se prvo sintetizira najbližim difonskim glasom zadane osnovne riječi pa se pokušava pronaći preslikavanje između sintetiziranih i stvarno snimljenih glasovnih iskaza. Međutim, za dobivanje boljih rezultata neophodno je ručno ispravljati automatski označene labele.

Druga točka tog postupka je označavanje perioda visine glasa (eng. *pitch marking*). Za tu svrhu opet se koristi automatsko labeliranje, koje uz dobro namještene parametre zadovoljavajuće označava svaki početak perioda.

Za sljedeći postupak potrebna je datoteka s osnovnim riječima izgenerirana ToFestivalFormat programom te snimljene `.wav` datoteke:

1. namjesti varijable okoline za Festfox i Speech Tools:

- `export FESTVOXDIR=/root/festvox`
- `export ESTDIR=/root/speech_tools/`

2. prenesi se u direktorij koji se dijeli s Vistom:

- `cd /mnt/shared`

3. stvori novi direktorij i prenesi se u njega:

- `mkdir martina_flight`
- `cd martina_flight`

4. instaliraj sve potrebne skripte za Festvox:

- `bash $FESTVOXDIR/src/ldom/setup_ldom $INST $DOMAIN $SPEAKER`

gdje su: `$INST=fer, $DOMAIN=flight, $SPEAKER=martina`

5. u Visti: u dijeljenom direktoriju `SHARED\martina_flight\etc` kopiraj datoteku s osnovnim rečenicama i preimenuj ju u `flight.data`

6. prebacivanje snimljenih `.wav` datoteka:

- prebaci sve `.wav` datoteke u direktorij: `SHARED\martina_flight\wav`

7. u virtualnom stroju: sinteziraj rečenice:

- `festival festvox/build_ldom.scm`
`festival>(build_prompts "etc/flight.data")`

stvoreni su sljedeći direktoriji za sintetizirane rečenice: *prompt-wav* koji sadrži sintetizirane *.wav* datoteke, *prompt-lab* koji sadrži labelirane *.lab* datoteke, *prompt-utt* koji sadrži govorne iskaze za Festival ili *.utt* datoteke

8. pokreni automatsko labeliranje snimljenih *.wav* datoteka:

- `/make_labs prompt-wav/*.wav`
stvoreni su sljedeći direktoriji: *cep* koji sadrži *.cep* datoteke koje sadrže informaciju o spektru signala (eng. *cepstrum*), *prompt-cep* koji sadrži *.cep* datoteke za sintetizirane snimke, *lab* koji sadrži labelirane *.lab* datoteke

9. u Visti: ručno podesi labele pomoću EMU¹⁴ alata:

- kopiraj datoteku *SHARED\martina_flight\etc\emu_lab.tpl* u *C:\.emu\Emu*
- u kopiranoj datoteci podesi imena putanja za direktorije *lab*, *lab_hlb* i *wav* tako da pokazuju na odgovarajuće direktorije u *SHARED\martina_flight*
- otvori EMU, učitaj *emu_lab.tpl*, te duplim klikom otvori snimku
- ručno podesi labele i spremi datoteku. U ovom radu kod ručnog podešavanja bitno je bilo podesiti difone za riječi koje su se ponavljale zbog preklapanja tijekom sinteze. Međutim za sve riječi koje su se pojavljivale samo jednom, kao npr. destinacije, dovoljno je bilo točno odvojiti pauze između riječi, jer je svaki difon jedinstveno vezan za pojedinu riječ

10. u virtualnom stroju: pokreni stvaranje cijele strukture govornih iskaza:

- `festival festvox/build_ldom.scm`
`festival>(build_utts "etc/flight.data")`

¹⁴ u Festvox dokumentaciji spominje se izvršni program *emulabel*, koji više ne postoji

11. pokreni izvlačenje znački perioda visine glasa:

- u datoteci *bin/make_pm_wave* podesi parametre za ženski glas:
-min 0.0033 -max 0.0875 -def 0.005
- `cp etc/flight.data etc/txt.done.data`
- `bin/make_pm_wave wav/*.wav`
stvoren je direktorij *pm* koji sadržava *.pm* datoteke s izvučenim značkama
- `bin/make_pm_fix pm/*.pm`
dodatno se ispravljaju *.pm* datoteke
- `/make_mcep wav/*.wav`
stvoren je direktorij *mcep* koji sadržava *.mcep* datoteke. One sadržavaju *MEL* skaliranu informaciju o spektru signala (eng. *Mel-frequency spectrum*)

12. pokreni generiranje sintetizatora baziranom na klaster jedinicama (eng.

cluster unit):

- `festival festvox/build_ldom.scm`
`festival>(build_clunits "etc/flight.data")`
- stvorene su datoteke *festival/clunits/fer_flight_martina.catalogue* i *festival/trees/fer_flight_martina.tree*

Stvoreni sintetizator

Izgenerirane datoteke za sintetizator nalaze se u ranije spomentim folderima *festival/clunits* i *festival/trees*. U prvom direktoriju nalazi se datoteka *fer_flight_martina.catalogue*, koja sadržava listu difona i snimljenih datoteka u kojima se oni nalaze, zajedno sa njihovim pozicijama. U drugom direktoriju nalazi se *CART*¹⁵ *.tree* datoteka *fer_flight_martina.tree*, koja sadrži izgeneriranu listu odluka.

Za rad sintetizatora potrebne su sljedeće direktorije: *festival*, *festvox*, *mcep* i *wav*. Njihovo ukupno zauzeće u memoriji je 21MB.

Da bi se stvoreni sintetizator učitao u Festival te sintetizirao i pročitao neki tekst iz domene, potrebno je u Linuxu utipkati naredbe:

```
➤ festival festvox/fer_flight_martina_ldom.scm
festival>(voice_fer_flight_martina_ldom)
festival>(SayText "Avion iz: Dubrovnika, sletio je u: ponoć.")
```

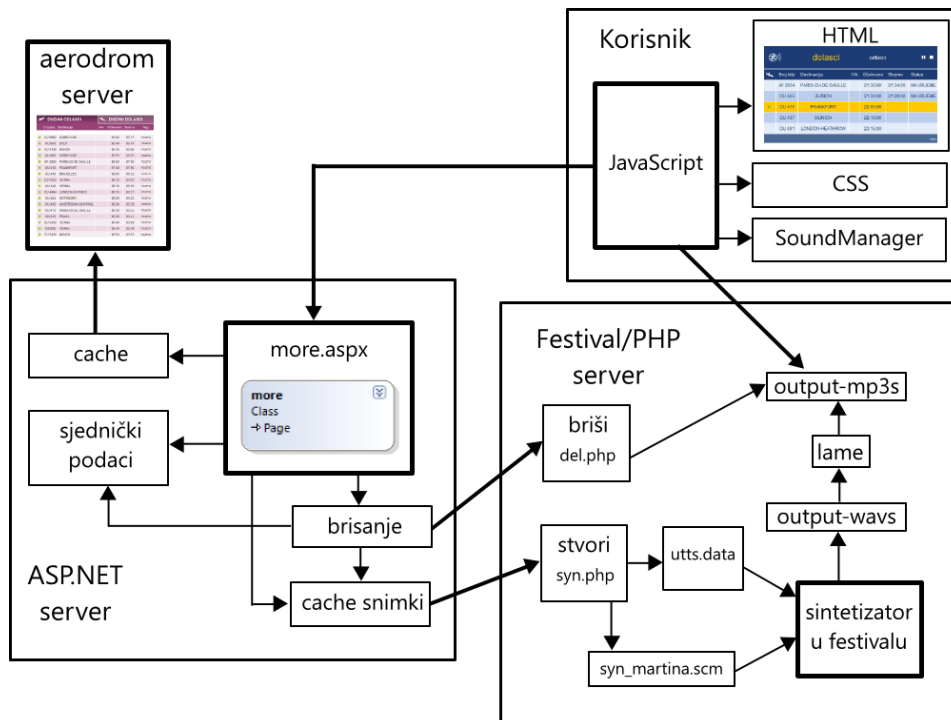
¹⁵ Classification and Regression Trees

4. Sustav za interaktivan prikaz rasporeda

Za interaktivan sustav prikaza rasporeda koristi se više komponenti koje su međusobno povezane. Osnovna ideja rada je da *korisnik* ili korisnički program preko HTTP protokola dohvaća statičku HTML stranicu s tablicom u koju će se dodavati retci s letovima. Letovi se dohvaćaju pomoću JavaScript komponente s ASP.NET servera koji služi za organizaciju i kontrolu sustava. ASP.NET dohvaća svježije podatke sa servera aerodroma i šalje upite serveru Festivala za sintezu i brisanje odgovarajućih rečenica, te u odgovoru korisniku šalje podatke o letovima, zajedno s identifikatorima govornih iskaza koji korisnik skida sa servera Festivala.

Pojedine komponente sustava, prikazane na shemi *Shema 4.1*, su:

- server aerodroma,
- Linux server Festivala,
- ASP.NET server,
- korisnik.



Shema 4-1: Sustava za interaktivan prikaz rezultata

Prilagodba i povezivanje sintetizatora s vanjskim svijetom

Budući da je sintetizator, koji je stvoren u prethodnom poglavlju, u Festivalu na Linuxu, a sustav za organizaciju i kontrolu podataka izveden je u ASP.NET na Windowsu, potrebno je povezati ta dva sustava. Najlakši način da se to napravi je preko postojeće Internet infrastrukture i HTTP protokola.

Prilagodba

Na početku se sintetizator prilagođava kako bi povezivanje bilo lakše ostvarivo. Prilagodba se može opisati u nekoliko koraka:

1. na Linuxu koji je spojen na Internet i ima instaliran *Apache* servis, stvoren je novi direktorij *public_html*, koji je vidljiv svim korisnicima koji pristupaju tom serveru preko HTTP protokola. U tom direktoriju datoteke moraju imati pravo pristupa `655`, a poddirektoriji pravo pristupa `755`, što je moguće namjestiti naredbom `chmod`. U ovom radu koristi se server kojem se može pristupiti na adresi <http://alley.zesoi.fer.hr/~milosz/>.
2. na Linuxu je instaliran program za kompresiju podataka – *lame*. Time se smanjuje količina sintetiziranih rečenica koje se šalju.
3. u *public_html* kopiran je direktorij izgeneriranog sintetizatora, *martina_flight*, s potrebnim folderima za njegov rad.
4. po početnim postavcima izgeneriran sintezator za riječi nepoznate domene koristi difonski engleski glas iz Festivala. Zbog toga je u datoteci *fer_flight_martina_ldom.scm* podešena fraza koja se izgovara u slučaju kada se u sintetiziranoj rečenici nalazi neka riječ izvan domene. U ovom radu ta fraza je prazna, što znači da se ignoriraju rečenice izvan domene.
5. dodatno je u *fer_flight_martina_ldom.scm* dodano pravilo za pauzu kada se pojavljuje znak „!” zbog izraza „Pažnja!”.

6. stvorena je nova datoteka `utts.data` u koju će se upisivati rečenice za sintezu odijeljene novim retkom u formatu (`$SOUNDID "rečenica"`), gdje je `$SOUNDID` jedinstveni identifikator rečenice. Dodatno su stvorena dva direktorija: `output-wavs` i `output-mp3s` u koje će se spremati sintetizirane i komprimirane datoteke. Pravo pristupa tim datotekama i direktorijima mora biti `777` , kako bi se preko `.php` skripte moglo manipulirati s njihovim sadržajima.
7. stvorena je nova datoteka `syn_martina.scm` koja pozvana iz Festivala:
 - učitava sintetizator:


```
(load "festvox/fer_flight_martina_ldom.scm")
```
 - uključuje Martinin glas: `(voice_fer_flight_martina_ldom)`
 - definira funkciju (`FileToWavs "filename"`)

koja za svaku rečenicu definiranu u datoteci `filename` u odgovarajućem formatu:

 - i. obavlja sintezu: `utt.synth`
 - ii. sprema `.wav` u `output-wavs` : `utt.save.wave`
 - iii. poziva lame koji komprimira `.wav` i sprema `.mp3` u `output-mp3s` :


```
lame output-wavs/SOUNDID.wav output-mp3s/$SOUNDID.mp3
```
 - iv. briše `$SOUNDID.wav` datoteku:


```
rm output-wavs/SOUNDID.wav
```
 - poziva funkciju (`FileToWavs "utts.data"`)

Povezivanje

Dvije osnovne operacije su potrebne za kontrolu sintetizatora i sintetiziranih datoteka: (i) sinteza rečenica, (ii) brisanje zastarijelih `.mp3` datoteka.

Sinteza rečenica ostvarena je pomoću `.php` skripte `syn.php` koja se nalazi u folderu `martina_flight` . Skriptu može pozvati svaki korisnik preko HTTP protokola. Jedina prividna zaštita koja služi za autentifikaciju korisnika je broj koji korisnik mora poslati u `POST` polju `check` . Korisnik koji želi da se obavi sinteza mora u `POST`

podacima poslati još dva polja: (i) `utts` – rečenice odvojene novim retkom, (ii) `files` – `$$SOUNDID` identifikatore rečenica odvojene novim retkom. Broj rečenica i identifikatora mora biti isti, a svaki redak rečenice odgovara retku identifikatora.

Skripta radi sljedeće:

1. otvara datoteku `utts.data` za pisanje.
2. sprema identifikatore i rečenice odvojene novim retkom u datoteku `utts.data`.
3. poziva ranije definiranu Festival skriptu za sintezu:
`festival -b syn_martina.scm`. Parametar `b` (eng. *batch*) kod poziva označava da će Festival izaći nakon što odradi skriptu.
4. zatvara datoteku `utts.data`.
5. ako sve je prošlo bez greške korisniku u odzivu ispisuje poruku "OK".

Bitno je napomenuti da se datoteka `utts.data` zatvara tek nakon obavljene sinteze, što spriječava njezino korumpiranje tijekom čitanja u Festivalu.

Brisanje rečenica ostvareno je pomoću `.php` skripte `del.php` koja se nalazi u direktoriju `martina_flight`. Kao i skripta `syn.php`, ona je dostupna svim korisnicima i koristi isti način autentifikacije korisnika. Korisnik koji želi pobrisati `.mp3` datoteke iz foldera `output-mp3s` mora poslati u POST podacima identifikatore `$$SOUNDID` odvojene novim retkom. Skripta za svaki identifikator pokušava pobrisati datoteku `$$SOUNDID.mp3` pomoću naredbe: `unlink("output-mp3s/SOUNDID.mp3")`. Ako je sve prošlo bez greške korisniku se u odzivu ispisuje poruka "OK".

Glavna ASP.NET aplikacija – drugi dio

ASP.NET aplikacija koristi, u prvom dijelu opisane, klase: `Common`, `DataGetter` i `Flight`. Njezina glavna klasa, koja je ujedno i servis za korisničke JavaScript asinkrone pozive, je `more.cs`.

Svaki put kada korisnik pristupa servisu, server šalje u odzivu redak s brojem koji jedinstveno identificira taj server. Ovo je uvedeno zbog korištenja besplatnog ASP.NET *hostinga*¹⁶ koji ponekad ne šalje očekivane podatke.

Korisnik može pristupiti servisu s *GET* parametrima koji mogu biti:

1. `type` s vrijednostima:
 - a. "A" za dolaske
 - b. "D" za odlaske
2. `getType` s vrijednostima:
 - a. "new" za početni dohvat samo novih promjena u rasporedu
 - b. "since" za početni dohvat letova od trenutnog vremena
 - c. "all" za početni dohvat letova od početka rasporeda
 - d. "more" za dohvaćanje sljedećih letova koje korisnik još nije dobio
3. `voice` s nazivom glasa. Po početnim postavcima je to "martina_flight"

Kada korisnik prvi put pristupa servisu za njega se stvara nova sjednica (eng. *session*). U sjedničkim podacima postoje polja `playedA` za dolaske i `playedD` za odlaske koja označavaju koji avionski letovi su odaslani korisniku. Ti letovi implementirani su kao lista letova ili `List<Flight>`.

Na početku svakog pristupa korisnika, poziva se metoda `getDownloadedFlights()` koja vraća trenutnu listu letova u polje `downloadedFlights`. Ova metoda koristi priručnu memoriju (eng. *cache*). Drugim

¹⁶ <http://www.aspspider.com/>

riječima, postoji globalno polje u koje se sprema raspored skinut sa servera aerodroma. Kod poziva metode `getDownloadedFlights()`, ako je to globalno polje starije od 5 minuta, dohvaća se lista letova sa servera aerodroma, inače se koristi priručna memorija. Na taj način smanjuje se promet u komunikaciji i povećava brzina izvođenja servisa.

U ovisnosti o tome koji `getType` je odabrao korisnik u polju `playedFlights` spremaju se odgovarajući letovi:

1. "new": svi letovi iz `downloadedFlights`.
2. "since": svi letovi stariji od trenutnog vremena umanjenog za 15 minuta. Treba primjetiti da se računa vrijeme koje je trenutno u Zagrebu jer su vremena u rasporedu u zagrebačkoj vremenskoj zoni.
3. "all": ni jedan let se ne dodaje, tj. polje sadržava nula letova.
4. "more": svi letovi iz korisničkog sjedničkog polja `playedA` ili `playedD`. Ovaj parametar označava da se nastavlja korisnička sjednica.

Nakon toga, bez obzira na vrijednost `getType` parametra, obavljaju se sljedeće faze:

1. provjerava se da li se slučajno prekinula sjednica s korisnikom. Ako je, šalje mu se poruku o pogrešci, kako bi mogao ponovno inicirati sjednicu.
2. u polje `synthesizedFlights` učitava se iz priručne memorije listu letova koja je već sintetizirana na serveru Festivala. Priručna memorija implementirana je kao datoteke, koje su različite za svaki glas, a nalaze se u direktoriju s istim nazivom kao traženi glas. Te datoteke sadržavaju, odvojene novim retkom, potpune informacije o svakom letu. Za parsiranje retka koristi se jedan od konstruktora `Flight` klase.
3. metodom `getAnotherFlights()` pronalazi se najviše 5 sljedećih avionskih letova, koje korisnik još nije odslušao. Povratna vrijednost sprema se u `newFlights` polje.

4. provjerava se koji letovi sadrže zastarjelu informaciju te se:
 - a. šalje upit korisniku: zastarjele letove se provjerava na način da se briše svaki let iz `playedFlights` koji nije nađen u `downloadedFlights`, a u odzivu korisniku se šalje redak s identifikatorom tog leta ili `$SOUNDID`. Jednakost dva leta se provjerava metodom `Flight.IsEqual()` po parametrima: `flightNumber`, `destination`, `via`, `status`, `expectedTime` i `realTime`.
 - b. briše privremena memorija sintetiziranih rečenica: svi letovi iz `synthetizedFlights`, koji se ne nalaze u `downloadedFlights`, se brišu, a istovremeno se sprema `$SOUNDID` identifikatore odijeljene novim retkom u `oldSoundIDs`. Tu varijablu predaje se metodi `deleteSounds()`, koja koristi `oldSoundIds` kao POST parametar u upitu za brisanje serveru Festivala, preko `del.php` priključka. Ne provjerava se uspješnost operacije.
5. provjerava se koje rečenice nisu jos sintetizirane:
 - a. u listu indeksa `newIndexes` spremaju se indeksi iz `newFlights` koji se ne nalaze u listi `synthetizedFlights`.
 - b. za svaki let koji nije sintetiziran stvara se njegov jedinstevni identifikator (`$SOUNDID`), te se priprema POST podatke s novim identifikatorima ili `newSoundIds` i s novim rečenicama ili `newUtts`, koje se stvaraju pomoću ranije opisane metode `Flight.ToSentence()`.
 - c. poziva se metodu `createSounds()`, koja šalje upit serveru Festivala, pomoću `syn.php` priključka. Metoda vraća listu uspješno sintetiziranih rečenica u obliku indeksa `newFlights` liste.
 - d. ako je povezivanje sa serverom Festivala neuspješno, korisniku se šalje pogrešku o nedostupnosti servisa.

6. svaki uspješno sintetiziran let se sprema u `synthetizedFlights` listu, koju se sprema u privremenu memoriju u odgovarajuće datoteke. Za spremanje jednog leta koristi se `Flight.ToString()` metoda.
7. jedino ako je uspjela sinteza svih letova:
 - a. korisniku se u odazivu šalje letove iz `newFlights` odvojene recima. Tijekom slanja letova korisniku, koristi se metoda `Flight.ToSendForm()` koja pretvara let s točkazarezom odvojenim parametrima, uključujući `$$SOUNDID`.
 - b. u `playedFlights` dodaju se letovi iz `newFlights`, te se to polje sprema u korisnički sjednički podatak `playedA` ili `playedD`.

Treba primjetiti da se, ako ne postoje nikakvi novi podaci ili ako sinteza novih letova nije uspjela u potpunosti, korisniku ne šalju nikakvi letovi. U tom slučaju korisnik neće javiti grešku, nego će pozvati servis kasnije.

Korsničko interaktivno sučelje

Osnovni zamisao interaktivnog prikaza rasporeda avionskih letova je, da se na intuitivan način prezentira korisniku trenutni dinamički raspored letova obogaćen govornim iskazima.

Pregled komponenata

Za korisničko sučelje korištene su sljedeće komponente koje se nalaze u direktoriju *Real* u FlightToSentence projektu:

1. statička HTML stranica *Default.aspx* koja se sastoji od: (i) glavne interaktivne tablice, (ii) tipki za prikaz odlaska ili dolaska, (iii) informacije o projektu
2. direktorij *imgs* sa svim potrebnim slikama
3. SoundManager dodatak za reproduciranje i kontrolu reproduciranja snimaka, koji se nalazi u *soundmanager* direktoriju
4. direktorij files s datotekama:
 - a. *flightToSentence.css* koja definira CSS izgled stranice:
 - i. glavne tablice, njezine retke, zaglavlje i podnožje
 - ii. klase retka `.odd` koja se koristi za prikaz neparnih redaka tablice
 - iii. klase retka `.highlighted` koja se koristi za prikaz trenutno aktivnog retka
 - b. *flightToSentence.js* sa svim potrebnim JavaScript funkcijama koje:
 - i. asinkrono pozivaju servis *more.aspx* za dohvaćanje novih letova
 - ii. prikazuju poruke o učitavanju podataka i o pogreškama
 - iii. kontroliraju prikaz podataka u interaktivnoj tablici, tj. stvaraju stare i brišu nove retke
 - iv. pokreću i zaustavljaju reproduciranje odgovarajućih govornih iskaza

Bitno je primjetiti da, kod korištenja asinkronih JavaScript poziva servisa glavne ASP.NET aplikacije, domena pozvane stranice mora biti ista kao i stranica na kojoj se izvodi skripta, takozvani *cross-domain* problem. To označava da se datoteka s korisničke stranice ili *Default.aspx* mora nalaziti na istom serveru i u istom direktoriju kao servis *more.aspx*. Zato se opisane komponente nalaze na glavnom ASP.NET serveru. Moguće je smjestiti korisničku stranicu i na drugom mjestu ako se *cross-domain* problem zaobiđe, npr. korištenjem *proxy* skripte koja spaja server na kojim se nalazi stranica sa serverom na kojim se nalazi servis.

Asinkronim pozivom servisa *more.aspx*, u valjanom odgovoru server šalje retke koji sadržavaju polja odijeljena točkazarezom. U ovisnosti o broju polja, pojedini retci označavaju:

- 2: pogrešku
- 1: brisanje retka
- 8: dodavanje retka

Način rada sučelja i njegova funkcionalnost

Način rada sučelja zajedno sa definicijom i glavnom funkcionalnošću potrebnih funkcija:

1. korisnik pokreće stranicu *Default.aspx* s mogućim dodatnim GET parametrima `getType` i `voice` u svom web pregledniku. Kod učitavanja stranice pokreće se JavaScript *flightToSentence.js* koja definira sve potrebne varijable i funkcije, te inicijalizira `SoundManager`.
2. nakon što se stranica učita:
 - a. pokreće se `onloadFlight()` funkcija koja dohvaća i sprema pokazivač na glavnu tablicu u varijablu `table`.
 - b. pomoću funkcije `turnA()` automatski se pokreće postupak dohvaćanja dolaznih letova.

3. korisnik može linkom odabrati vrstu letova, što pokreće postupak dohvaćanja letova: za dolazak `turnaA()` ili za odlazak `turnD()`.
4. funkcija `turnA()` ili `turnD()`:
 - a. postavlja globalnu varijablu `type` na vrijednost "A" za dolaske ili "D" za odlaske.
 - b. prekida dretvu, pokrenutu za prethodni asinkroni poziv, ako ona postoji.
 - c. postavlja varijablu `playedSoundID` na `null`. U toj varijabli definiran je `$SOUNDID` identifikator trenutno reproducirane snimke za let.
 - d. poziva funkciju `callRequest(false)` koja po prvi put skida letove sa servisa.
5. funkcija `callRequest(isMore)`:
 - a. ako se poziva servis `more.aspx` po prvi put, parametar `isMore` je `false`:
 - i. briše se sadržaj cijele glavne tablice: `CleanTable()`
 - ii. pokazuje se poruka o dohvaćanju podataka u novom retku glavne tablice: `ShowMessage()`
 - iii. stvara se novi asinkroni poziv s odgovarajućim `get=getType` parametrom
 - iv. na serveru ASP.NET resetirat će se korisnička sjednica
 - b. inače, ako se žele dohvatiti sljedeći letovi, parametar `isMore` je `true`:
 - i. pokazuje se poruka o dohvaćanju podataka
 - ii. stvara se novi asinkroni poziv s `get=more` parametrom
 - c. dodatno se postavlja maksimalno vrijeme u kojem mora stići odgovor od servera. Ako server ne odgovori na zadano vrijeme, poziva se funkcija `connectionFailed()` koja prikazuje poruku o grešci i prekida postupak dohvaćanja letova.

6. kada je asinkroni poziv gotov, poziva se funkcija `reqFinished(isMore)` koja:
- a. prekida dretvu iz 5c.
 - b. ovisno o vrijednosti parametra `isMore`:
 - i. `false`: briše cijelu tablicu: `CleanTable()`
 - ii. `true`: briše zadnji redak s porukom: `deleteLastRow()`
 - c. ako je uspješno stigao odgovor sa servera, provjerava autentificira li ga prvi redak: varijabla `check2`:
 - i. ako nije, asinkroni poziv ponavlja se u eksponencijalno rastućem vremenu `timeOut`
 - ii. ako broj pokušaja `trialsCounter` prijeđe maksimalnu vrijednost `trialsLimit`, ispisuje se poruka o grešci, a postupak dohvaćanja letova se zaustavlja
 - d. ako se sljedeći dohvaćen redak sastoji od dva polja, server je poslao informaciju o grešci:
 - i. ako je prvo polje jednako `Session`, server informira korisnika da je resetirao njegovu sjednicu, pa korisnik ponovno započinje proceduru dohvata letova, opisanu u 4. koraku
 - ii. inače, asinkroni poziv ponavlja se u eksponencijalno rastućem vremenu `timeOut`
 - iii. ako broj pokušaja `trialsCounter` prijeđe maksimalnu vrijednost `trialsLimit`, ispisuje se poruka o grešci, a postupak dohvaćanja letova se zaustavlja
 - e. za svaki ostali redak koji ima 8 polja dodaje se novi redak u glavnu tablicu:
 - i. pomoću funkcije `document.createElement()` za svako polje osim zadnjeg stvaraju se `td` ćelije u koje se upisuju vrijednosti polja, koristeći parametar `innerHTML`.
 - ii. stvara se novi redak `tr` kojem se dodaju sve stvorene ćelije pomoću funkcije `appendChild()`.

- iii. identifikatoru retka `id`, pridružuje se vrijednost zadnjeg polja (`$SOUNDID`), te se stvoreni redak dodaje glavnoj tablici.
 - iv. u `SoundManageru` stvara se novi zvuk s identifikatorom jednakim identifikatoru retka (parametar `id`) i s putanjom koja pokazuje na sintetiziranu `.mp3` datoteku jednaku `mp3FolderUri+$SOUNDID+".mp3"` (parametar `url`).
 - v. kod stvaranja zvuka definira se događaj `onload` koji se pokreće prije nego što započne reproduciranje neke `.mp3` datoteke. U tom događaju provjerava se postojanje `.mp3` datoteke. Ako ona ne postoji, to znači da ju je server pobrisao pa je njezin pridruženi redak zastario. U tom slučaju redak se briše iz tablice i pokreće se reproduciranje sljedećeg.
 - vi. kod stvaranja zvuka definira se događaj `onfinish` koji se pokreće nakon što je reproducirana `.mp3` datoteka. U tom događaju poziva se funkcija `playNext()` koja pomoću `playedSoundId` varijable pronalazi `id` sljedećeg retka te pokreće njegovo reproduciranje. Ako ne postoji više redaka, pokreće se asinkroni poziv za dohvaćanje novih letova `callRequest(true)`.
 - vii. kod stvaranja zvuka dodatno se definiraju događaji: `onpause`, `onplay` i `onresume`, koji mijenjaju ikonice i semantiku tipki za kontrolu reprodukcije.
- f. za svaki ostali redak koji ima jedno polje briše se stari redak iz glavne tablice:
- i. iz servera je stigao `$SOUNDID` o starom letu
 - ii. korisnik poziva funkciju `deleteRow(id)` koja s `id=$SOUNDID` briše redak iz glavne tablice i uništava zvuk napravljen u `SoundManageru`
- g. prikazane retke glavne tablice uljepšava se sa stilom parni-neparni pomoću `makeStyledRows()` funkcije.

- h. ako su stigli novi letovi:
 - i. za prvi poziv pokreće se prvi govorni iskaz u tablici:
`playerPlay()`
 - ii. za sljedeće pozive pokreće se prvi novi govorni iskaz u tablici:
`playNext()`
 - i. ako nisu stigli novi letovi, asinkroni poziv ponavlja se nakon pet minuta.
7. svaki redak u glavnoj tablici može biti reproduciran kad se na njega klikne mišem. Za tu svrhu služi funkcija `play(id)`, gdje je `id=$SOUNDID`.
8. gore desno nalaze se tipke za kontrolu reprodukcije:
- a. `play`: pokreće se ili nastavlja trenutno reproducirani iskaz:
`playPlayer()`
 - b. `pause`: pauzira se reprodukcija: `pausePlayer()`
 - c. `stop`: zaustavlja se reprodukcija: `stopPlayer()`

5. Rezultat rada

U ovom završnom radu ostvaren je sintetizator rasporeda avionskih letova i sustav za interaktivan proces prikaza rezultata s korisničkom web aplikacijom prikazanom na slici *Slika 5.1*.

Iako je opisan postupak generiranja sintetizatora prvenstveno namijenjen za engleski jezik, on je dao veoma dobre rezultate i za hrvatski jezik.



Broj leta	Destinacija	VIA	Očekivano	Stvarno	Status
OU 415	FRANKFURT		22:05:00	21:55:00	NA VRJEME
OU 437	MUNICH		22:10:00	22:11:00	NA VRJEME
8P 337	LILLE		22:50:00	22:40:00	NA VRJEME
OK 814	PRAHA		23:10:00		
OU 667	DUBROVNIK		23:55:00		

Fakultet Elektrotehnike i Računarstva - Završni Rad 2009.
Sinteza govora rasporeda avionskih letova aerodroma Zagreb.
Alati: festival+festvox, asp.net, php, javascript, adobe flash.
Glas: Martina Petraš
Mentor: Davor Petrinović
Lipanj 2009., Miłosz Chmura

Slika 5.1: Korisničko web sučelje

Korisnička aplikacija napisana je u HTML jeziku i koristi CSS sa stiliziranje stranice. Za rad aplikacije u web pregledniku mora biti uključen JavaScript i instaliran AdobeFlash dodatak. Aplikacija je bila testirana i radi u popularnim web preglednicima: *Firefox 3*, *Internet Explorer 7*, *Google Chrome 2* i *Opera 9*.

Primjer korištenja

Sintetizator i cijeli sustav za interaktivan prikaz rasporeda nalazi se na dva servera koja su spojena na internet vezu. S korisničke strane aplikaciji se može pristupiti sa stranice <http://alley.zesoi.fer.hr/~milosz/index.php> koja preusmjerava na ASP.NET server.

Dodatno se aplikaciji može pristupiti s GET parametrima:

1. `getType`: definira drugačiji početni pristup podacima: (i) "new", (ii) "since", (iii) "all"
2. `voice`: definira glas za sintetizirane rečenice

Primjer poziva koji je ujedno poziv s početnim postavkama je:

http://alley.zesoi.fer.hr/~milosz/index.php?getType=since&voice=martina_flight.

Na otvorenoj stranici korisnik može klikom na retke rasporeda odslušati pojedini let ili klikom na odlasci/dolasci uključiti automatsko dohvaćanje i sviranje aktualnog avionskog rasporeda.

Dodavanje novog glasa

U stvorenom sustavu na jednostavan način moguće je dodati sintetizator s novim glasom. Da se to učini potrebno je napraviti sljedeće:

1. stvoriti direktorij s nazivom glasa na ASP.NET serveru u istom direktoriju u kojem se nalazi početni glas *martina_flight*
2. stvoriti direktorij s nazivom glasa na Linux serveru na kojem se nalazi Festival u istom direktoriju u kojem se nalazi početni glas *martina_flight*
3. u novo stvorenom direktoriju kopirati iz *martina_flight* datoteke: *utts.data*, *syn.php*, *del.php* i *syn_martina.scm*
4. u novo stvorenom direktoriju stvoriti foldere *output-mp3s* i *output-wavs* te im promijeniti prava pristupa na `777`
5. po potrebi modificirati `.scm` skriptu

Za poziv aplikacije sa glasom `novi_glas` potrebno je pokrenuti stranicu:

http://aspspider.info/zavrzni2009/Real/Default.aspx?voice=novi_glas.

Problemi

Tijekom izrade sustava ustanovilo se par grešaka koje imaju veze s reproduciranjem govornih iskaza:

1. kod snimljenih rečenica povremeno se čuje tiho kucanje, što najvjerojatnije ima veze sa kvalitetom korištenog mikrofona
2. kod sinteze riječi „jedan” čuje se jeka jer se ta riječ kod snimanja pojavila više puta i loše je spojena
3. kod izgovaranja „za Budimpeštu” čuje se „šišu” umjesto „štu”, što je rezultat snimanja tog grada u kontekstu „Avion za Budimpeštu”
4. ponekad, nakon reprodukcije određene rečenice, čuje se pucanje zvuka
5. kod nekih korisnika pucanje se čuje svaki put na prijelazu između reprodukcije jednog i drugog leta
6. zbog sadržanih svih informacija, neke rečenice postaju preduge, što smanjuje njihovo razumijevanje i cilj uspješnog prenošenja poruke

Poboljšanja

Poboljšanja sustava mogla bi uključivati:

1. izgradnju sintetizatora koji je temeljen na biranju riječi, umjesto biranja difona vezanih uz riječ,
2. izgradnju sintetizatora difonskog hrvatskog za izgovor nepoznatih destinacija, umjesto njihovog slovkanja,
3. optimizaciju ASP.NET aplikacije s obzirom na brzinu rada, prvenstveno u korištenim algoritmima za usporedbu letova i spremanje priručne memorije,
4. zaštitu od nepoželjnog upada na Festival server preko sučelja *del.php* i *syn.php*,
5. izgradnja alternativnog sučelja za korisnike koji nemaju uključen JavaScript ili instaliran Adobe Flash.

6. Zaključak

U ovom završnom radu uspješno je napravljen sustav za sintezu govora i interaktivan prikaz rasporeda avionskih letova.

Sinteza govora se često koristi kao zamjena za operatera u telefonskim centralama za podršku, za čitanje vijesti na web portalima i sve popularnija je za čitanje ostalog web sadržaja.

Gotov sustav može se, uz male modifikacije, primijeniti u stvarnosti, jer je zbog korištenja Internet infrastrukture i web korisničkog sučelja dostupan svima koji se mogu spojiti na Internet mrežu i imaju web preglednik koji podržava JavaScript i Adobe Flash. Napokon, kao svaki govorni sustav koji reproducira informacije u zvučnom obliku, izrađena aplikacija mogla bi biti od velike koristi slijepim, slabovidnim ili nepismenim osobama.

Miłosz Chmura

7. Literatura

[1] O'Shaughnessy D., *Modern Methods of Speech Synthesis*. IEEE Circuits and Systems Magazine, treći kvartal 2007., str. 6-23.

[2] Wikipedia, *Speech synthesis*,
http://en.wikipedia.org/wiki/Speech_synthesis, datum pristupa: 16.06.2009.

[3] Black A., Taylor P., Caley R., 17.06.1999., *The Festival Speech Synthesis System: System documentation*,
<http://www.cstr.ed.ac.uk/projects/festival/manual/>, datum pristupa: 30.03.2009.

[4] Black A., Lenzo K., 2007., *Festvox: Building Synthetic Voice*,
<http://festvox.org/bsv/>, datum pristupa: 10.04.2009.

Sustav za automatsku govornu najavu događaja u zračnom prometu – Sažetak

U ovom radu opisan je postupak generiranja sintetizatora za ograničenu domenu u Festival sustavu, koji se koristi za sintezu na hrvatskom jeziku rasporeda avionskih letova aerodroma Zagreb-Pleso. Dodatno je opisan postupak stvaranja sustava za interaktivan prikaz rasporeda u stvarnom vremenu koji je, korištenjem Internet infrastrukture, dostupan svim korisnicima preko web sučelja.

Ključne riječi: *sinteza govora, festival, festvox, sinteza s biranjem jedinica, ograničena domena, raspored letova*

System for automatic speech notification of air traffic events – Abstract

This work describes the process of creating a synthesizer for limited domain in Festival system, which is used to synthesize in Croatian language airplane timetable of Zagreb-Pleso airport. It also describes the process of creating the system for real-time interactive representation of the timetable, using the Internet infrastructure, which is available to all users through the web interface.

Keywords: *speech synthesis, festival, festvox, unit selection, limited domain, flight timetable*

Dodatak A: Lista rečenica snimljenih za sintetizator

01:Ponoć i dvadeset jedna minuta.
02:Jedan sat i trideset jednu minutu.
03:Dva sata i četrdeset dvije minute.
04:Tri sata i pedeset četiri minute.
05:Pet sati i šest minuta.
06:Sedam sati i osam minuta.
07:Devet sati i deset minuta.
08:Jedanaest sati i dvanaest minuta.
09:Trinaest sati i četrnaest minuta.
10:Petnaest sati i šestnaest minuta.
11:Sedamnaest sati i osamnaest minuta.
12:Devetnaest sati i trideset minuta.
13:ku, duplove, eh, er, teh, ipsilon, uh, ih, oh, peh, ah, es, deh, ef, geh, hah, jot, kah, el, zeh, iks, ceh, veh, beh, en, em
14:nula, jedan, dva, tri, četiri, pet, šest, sedam, osam, devet
15:Avion iz Amsterdama-Schiphol, s presjedanjem u Amsterdama-Schiphol, avionske kompanije Aeroflot Russian Airlines, s brojem leta iks ipsilon, sletio je u...
16:Avion iz Antalije, dolazi na vrijeme u...
17:Pažnja! Novo vrijeme dolaska aviona iz Barcelone, je...
18:Avion iz Berlina-Shönefeld, kasni u dolasku. Novo vrijeme dolaska je...
19:Očekivano vrijeme dolaska aviona iz Bolonje, je...
20:Air France, Austrian Airlines, Croatia Airlines, Czech Airlines
21:Germanwings, Iberia Airlines, Lufthansa Cityline, Malév Hungarian Airlines, Scandinavian Airlines System, Skyservice Airlines, Sun D'or, TAP Air Portugal, Tarom, Turkish Airlines, Wizz Air, Air Guinee Express, Pacific Coastal Airline
22:Air One, Dubrovnik Air, Adria Airways, Miniliner, Farnair Switzerland, Solinair, Nouvel Air Tunisie, Japan Airlines, Karthago Airlines
23:Antaliji, Barceloni, Berlinu-Shönefeld, Bolonji
24:Brača, Brisela, Budimpešte, Kölna, Kopenhagena, Dubrovnika, Frankfurta, Gerone, Göteborga, Istanbula, Lisabona
25:Braču, Briselu, Budimpešti, Kölnu, Kopenhagenu, Dubrovniku, Frankfurtu, Geroni, Göteborgu, Istanbulu, Lisabonu
26:Londona-Gatwick, Londona-Heathrow, Londona-Luton, Madrida, Monastira, Moskve-Sheremetyevo, Münchena, Osijek-Klisa
27:Londonu-Gatwick, Londonu-Heathrow, Londonu-Luton, Madridu, Monastiru, Moskvi-Sheremetyevo, Münchenu, Osijeku-Klisa
28:Pariza-Charles de Gaulle, Podgorice, Pule, Praga, Prištine, Rhodosa, Rijeke, Rima-Fiumicino, Sarajeva, Skoplja, Splita, Stockholma, Stuttgarta
29:Parizu-Charles de Gaulle, Podgorici, Puli, Pragu, Prištini, Rhodosu, Rijeci, Rimu-Fiumicino, Sarajevu, Skoplju, Splitu, Stockholmu, Stuttgartu
30:Tel Aviva, Beča, Zadra, Zürichu, Atene-Eleftherios, Sofije, Ankare, Ljubljane, Bergama, Genu, Helsinkija, Tokija-Narita, Toronto
31:Tel Avivu, Beču, Zadru, Zürichu, Ateni-Eleftherios, Sofiji, Ankari, Ljubljani, Bergami, Genui, Helsinkiju, Tokiju-Narita, Torontu
32:Bukurešta-Otopeni, Damascusa, Verone, Tarbes-Lourdesa, Ohrida, Rima-Ciampino, Beograda, Valladolida, Belfasta, Manchesteru, Badena
33:Bukureštu-Otopeni, Damascusu, Veroni, Tarbes-Lourdesu, Ohridu, Rimu-Ciampino, Beogradu, Valladolidu, Belfastu, Manchesteru, Badenu
34:Avion za Budimpeštu, poletio je u...
35:Novo vrijeme odlaska aviona za Barcelonu je...
36:Avion za Geronu, kasni u odlasku. Novo vrijeme odlaska je...
37:Započelo je ukrcavanje putnika u avion za Bolonju... Vrijeme odlaska je...
38:Očekivano vrijeme odlaska aviona za Moskvu-Sheremetyevo, je...
39:Avion za Antaliju, odlazi na vrijeme u...
40:Podgoricu, Prištinu, Rijeku, Atenu-Eleftherios, Sofiju, Ankaru, Pulu, Ljubljanu, Genu, Veronu
41:Amsterdam-Schiphol, Berlin-Shönefeld, Brač, Brisel, Köln, Kopenhagen, Dubrovnik, Frankfurt, Göteborg, Istanbul, Belfast
42:Lisabon, London-Gatwick, London-Heathrow, London-Luton, Madrid, Monastir, München, Osijek-Klisa, Pariz-Charles de Gaulle, Prag
43:Rhodos, Rim-Fiumicino, Sarajevo, Skoplje, Split, Stockholm, Stuttgart, Tel Aviv, Beč, Zadar, Zürich, Bergamo, Helsinki, Manchester
44:Tokio-Narita, Toronto, Bukurešt-Otopeni, Damascus, Tarbes-Lourdes, Ohrid, Rim-Ciampino, Beograd, Valladolid, Baden