Tehnološka izvedba sodobnega digitalnega slovarja

Bojan Klemenc, Marko Robnik-Šikonja, Luka Fürst, Ciril Bohak in Simon Krek

Abstract

An important component in a state-of-the-art digital Slovenian language dictionary is its technological framework, which is briefly presented in this paper. We view the dictionary as a multi-tier architecture, with a presentation tier, a middle application tier (a back-end application system with a component for semi-automatic data extraction) and a data tier. In its natural form, language data is multidimensional. In a printed dictionary, there is just the presentation tier, and many of the relations between the underlying data are difficult to access or may even be lost. In electronic dictionaries, however, there are no such restrictions. The data can be preserved in all its complexity and presented in various ways because there is a distinction between the data and its presentation. This separation is the key factor in integrating the various data sources (different corpora and external databases) into a unified database. Various users or programs can then query different parts of the database based on their interests and the presentation tier displays or returns the data on different levels of granularity. For each tier we present the structure and review some of the technological considerations which ensure that the extensibility, reliability and adaptability of the final solution are to a high standard.

Keywords: digital dictionary, multi-tier software architecture, presentation layer, relational database, data extraction

Ključne besede: digitalni slovar, večdelna programska arhitektura, predstavitveni nivo, relacijska baza podatkov, luščenje podatkov

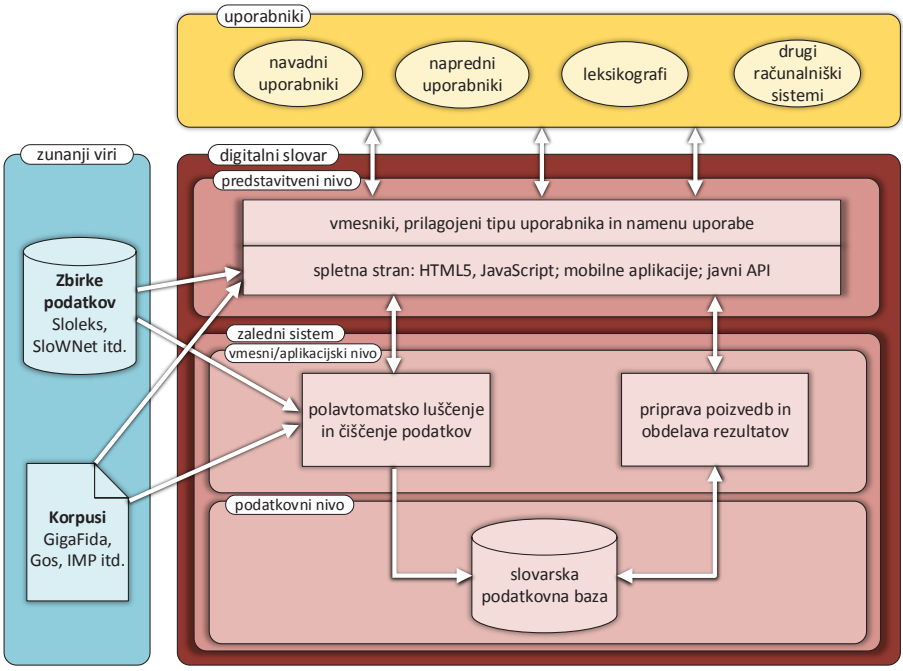
1 Uvod

Sodoben digitalni slovar slovenskega jezika bo imel poleg svoje vsebinske ravni tudi tehnološko, ki jo predstavljamo v tem prispevku. Najprej opišemo poglavitne komponente, ki so del tehnološke izvedbe takšnega slovarja, v nadaljevanju pa se prispevek dotakne tudi smernic za implementacijo takšnega slovarja. Pri tehnološki zasnovi slovarja je zelo pomembno, da so uporabljeni koncepti in tehnologije premišljeno izbrani z namenom trajnosti, razširljivosti, prilagodljivosti in zanesljivosti končne implementacije.

Prve generacije digitalnih oz. digitaliziranih slovarjev so bile z vidika podatkovnega modela zgolj preslikava obstoječih slovarjev v papirni obliki (prim. Urdang 1984, Boguraev in Briscoe 1989, Hajnšek-Holz 1993, Krek 2014): geselski članki so se s svojo hierarhično organizacijo in oznakami hranili v datotekah, npr. v datotekah XML (angl. *eXtensible Markup Language*) ali pri spletnih slovarjih neposredno v HTML (angl. *HyperText Markup Language*). V slednjem primeru sta logična struktura geselskega članka in njegova prikazna oblika (izgled) združeni, pri geselskih člankih v XML-u pa je podana struktura članka, medtem ko se prikazna oblika ustvari s pomočjo ustreznih transformacij po predlogi, kot je npr. CSS (angl. *Cascading Style Sheets*). V tem primeru že pridemo do osnovnega ločevanja med podatki in njihovim prikazom. Besedilo geselskega članka lahko vsebuje tudi reference na druge geselske članke oz. njihove sestavne dele. Poizvedbe, ki jih lahko izvajamo v takšnem slovarju, so tipično omejene na iskanje iztočnic, iskanje po določenih elementih (ki so tipično opredeljeni v XML-u) in splošno iskanje po besedilu geselskega članka. Prikaz rezultatov je v takšnih slovarjih vselej enak: geselski članek, morebiti z obarvanimi rezultati iskanja. Poizvedbi oz. iskanemu rezultatu bolj prikrojenih prikazov pa ne moremo dobiti, saj so podatki strukturirani za določeno število vnaprej definiranih prikazov in zato oblikovanje prikaza glede na trenutno povpraševanje ni mogoče. Takšna ureditev podatkov (geselskih člankov) je naravna, kadar imamo opravka z medijem, kot je papir, kjer morajo biti podatki že organizirani oz. shranjeni v svoji končni predstavitveni obliki. Digitalno zasnovani slovarji namreč nimajo te fizične omejitve, pri njihovem načrtovanju pa moramo preseči tako »nivo papirja« kot tudi statične podatkovne strukture. Podatke je potrebno hraniti v njihovi naravni večrazsežni obliki ter jih na podlagi želenih poizvedb ustrezno filtrirati, preurediti in prikazati.

Za izvedbo digitalnega slovarja je torej ključna ločitev predstavitve podatkov od podatkov samih. Podatke na ta način lahko hranimo v vsej njihovi kompleksnosti, predstavitev podatkov pa je glede na vso kompleksnost hranjenih podatkov mogoča z različnih zornih kotov in omogoča tako spremembo gledišča kot stopnjo podrobnosti predstavitve. Tehnološko gledano je pri tem glavna delitev na predstavitveni nivo in podatkovni nivo. Uporabniku podatkovni nivo ni viden in

do teh podatkov dostopa zgolj preko predstavitvenega nivoja. Predstavitveni nivo uporabniku prikazuje podatke in sprejema uporabnikove »poizvedbe« (klike, iskanja). Vez med obema nivojema predstavlja t. i. aplikacijski oz. vmesni nivo. Njegova naloga je, da poizvedbe predstavitvenega nivoja pretvori v obliko, s katero lahko od podatkovnega nivoja pridobi ustrezne podatke. Pridobljene podatke nato ustrezno prečisti in preoblikuje ter posreduje predstavitvenemu nivoju.



Slika 1: Shema arhitekturne delitve digitalnega slovarja na tri nivoje: predstavitveni, vmesni aplikacijski in podatkovni nivo. Uporabniki vidijo predstavitveni nivo (spletna stran, mobilne aplikacije), ki jim prikazuje ustrezno izbrane in obdelane podatke iz podatkovnega nivoja. Naloga vmesnega nivoja je, da povezuje podatkovni in predstavitveni nivo ter omogoča polnjenje slovarske baze iz zunanjih virov.

Tako dobimo trinivojsko arhitekturo (Slika 1), kjer je uporabniku viden zgornji predstavitveni nivo (angl. *presentation tier,* tudi *front end*), vmesni aplikacijski nivo in spodnji podatkovni nivo pa nista vidna, imenujemo ju tudi zaledni sistem (angl. back end). Plastovitost arhitekture omogoča, da so posamezni deli relativno neodvisni eden od drugega – višji nivoji preko vnaprej definiranih programskih vmesnikov dostopajo do nižjih nivojev. Posledično lahko zamenjamo posamezni nivo, ne da bi to negativno vplivalo na ostale nivoje. Ločevanje predstavitvenega nivoja od baze podatkov na podatkovnem nivoju omogoča integracijo slovarjev

in virov, ki so bili do sedaj ločeni, saj imamo enotno podatkovno bazo, iz katere črpajo različni »pogledi« predstavitvenega nivoja, da prikažejo podmnožico baze (npr. pisni jezik, govorjeni jezik, sodobni jezik, arhaični jezik, pokrajinske različice, kombinacije omejitev itd.).

Na predstavitvenem nivoju lahko prikazujemo različne vmesnike tudi glede na vrsto uporabnika – ena skupina uporabnikov slovarja lahko vidi/izbere drugačen vmesnik od druge, tako ima npr. srednješolec, ki uporablja slovar za pisanje eseja, popolnoma drugačen prikaz z drugimi in drugače hierarhiziranimi podatki kot jezikoslovec ali leksikograf. Vsi sicer dostopajo do iste baze podatkov, se pa razlikuje nivo podrobnosti vrnjenih podatkov oz. možnost vnašanja podatkov. Na primer leksikograf lahko podatke tudi spreminja, medtem ko jih drugi uporabniki ne morejo.

Posodabljanje slovarske baze se lahko izvaja tako s pomočjo ročnega dela leksikografa ali z množičenjem (angl. *crowdsourcing*) (prim. Kosem et al. 2013a; 2013b), sistem pa omogoča, da se podatki pripravljajo avtomatsko z luščenjem iz zunanjih virov (npr. korpusov). Luščenje ni zgolj enkratno opravilo, saj se jezik in posledično korpusi spreminjajo, tako da gre za ponavljajoč se proces. Vmesni aplikacijski nivo ima tako poleg naloge, da služi povezovanju predstavitvenega in podatkovnega nivoja, tudi nalogo, da se povezuje z zunanjimi viri in omogoča začetno avtomatsko luščenje podatkov.

Po tehnološki plati lahko slovar tako razdelimo na štiri poglavitne komponente, ki so na kratko predstavljene v nadaljevanju.

1. **Podatkovna baza** kot glavna komponenta podatkovnega nivoja je realizirana v obliki enotne relacijske podatkovne baze, ki je namenjena hrambi jezikovnih podatkov in iz korpusov izluščenih informacij.
2. **Zaledni aplikacijski sistem** oz. vmesni aplikacijski nivo je namenjen integraciji celotne rešitve in vsebuje programske vmesnike za dostop predstavitvenih modulov (spletna aplikacija, mobilne aplikacije) in programsko kodo za dostop do podatkovne baze.
3. **Komponenta za avtomatsko luščenje podatkov**, ki je v bistvu del vmesnega aplikacijskega nivoja in skrbi za polnjenje in ažurno obnavljanje baze podatkov iz zunanjih besedilnih korpusov in baz. Zaradi kompleksnosti jo bomo kot komponento obravnavali ločeno od preostalega aplikacijskega nivoja. Kot del leksikografskega procesa je avtomatsko luščenje podatkov predstavljeno tudi v Gantar et al. (2015).
4. **Predstavitveni nivo** v obliki spletnega portala s predstavitvijo vseh jezikovnih podatkov za različne tipe uporabnikov in mobilne aplikacije za različne mobilne platforme (npr. Android, Apple iOS in Windows Phone), ki omogočajo iskanje in brskanje po jezikovnih podatkih ter v opisanem nadzorovanem leksikografskem procesu (ibid.) tudi sodelovanje

pri popravljanju in dopolnjevanju jezikovnih podatkov. Uporabniki niso nujno samo ljudje, zato preko predstavitvenega nivoja izpostavimo tudi programski vmesnik, preko katerega lahko drugi računalniški sistemi dostopajo do slovarja.

Tehnološko izvedbo slovarja je smiselno v večji meri zasnovati na odprtokodnih rešitvah, ki so danes že dovolj zmogljive, da podpirajo tudi zahtevne operacije in veliko število uporabnikov. Pri izbiri tehnologij nam delitev na nivoje omogoča, da na vsakem nivoju izberemo najustreznejše tehnologije oz. jih po potrebi zamenjamo. Enako načelo velja tudi za posamezne komponente. Na primer: komponenta za avtomatsko luščenje podatkov je ločena od ostalih komponent na aplikacijskem nivoju; z njimi po potrebi komunicira preko programskih vmesnikov.

Komunikacija med posameznimi nivoji poteka po modelu odjemalec–strežnik. Odjemalec pošlje zahtevo strežniku, ta pa pošlje ustrezen odgovor. Odjemalci imajo v primeru slovarja lahko zaradi tega manjše procesorske in pomnilniške zahteve, saj se podatki v veliki večini hranijo na strežniku in se tam tudi obdelujejo, odjemalcu pa se pošljejo le podatki odgovora, ki jih odjemalec (predstavitvenega nivoja) potem ustrezno prikaže. Manjše procesorske in pomnilniške zahteve pomenijo manjšo porabo energije, kar omogoča uporabo slovarjev na manj zmogljivih mobilnih napravah pod pogojem, da imamo podatkovno povezavo do strežnika. Podatki v slovarski bazi se tako v procesu izdelave in tudi kasneje redno spreminjajo, zato je takšna arhitekturna rešitev primerna, saj imajo uporabniki vedno dostop do najbolj ažurne različice podatkovne baze. Vendar pa takšna arhitekturna rešitev ne pomeni, da morajo biti odjemalci in strežniki strogo nameščeni na različnih napravah, ampak so lahko tudi fizično na isti napravi. V tem primeru pride do replikacije (dela) baze, kar pomeni, da je treba poskrbeti, da so posamezne kopije baze ustrezno sinhronizirane (tipično z eno od kanoničnih kopij baze). Primer koristnosti takšne rešitve je, da lahko (tudi na mobilnih napravah) uporabljamo slovar brez povezave z internetom.

Večnivojska in modularna zgradba nam omogočata, da posamezne dele slovarja gradimo, evalviramo in testiramo vzporedno. Predpogoj za to pa je, da so povezave med posameznimi nivoji, npr. programski vmesnik, vnaprej dobro definirane.

2 Podatkovni model in baza podatkov

Enotna podatkovna baza in ločen predstavitveni nivo omogočata integracijo slovarjev in virov, ki so bili do sedaj ločeni. Da lahko zgradimo ustrezno enotno podatkovno bazo, je na eni strani treba definirati ustrezen podatkovni model, ki bo lahko hranil integrirane podatke iz različnih obstoječih in novonastalih baz. Poleg tega mora omogočati širši nabor poizvedb, da pokrije tiste, ki so se že izvajale na

obstoječih bazah, in omogoči nove na integriranih podatkih. Na drugi strani pa se z integracijo oz. enotno bazo poveča količina hranjenih podatkov, ki morajo biti še vedno hitro dostopni.

Tabela 1 za posamezne jezikovne podatke, ki bodo prikazani v uporabniškem vmesniku, prikazuje vir podatkov, predvideno umestitev v enotno podatkovno bazo in obstoječ trenuten format podatkov. Pri vključenosti v bazo je navedena umestitev neposredno v bazo ali referenca na zunanje vire, npr. korpuse. Več o povezanih slovarskih in korpusnih virih v Krek et al. (2013b).

Tabela 1: Prikazani podatki, njihovi viri, način vključenosti v podatkovno bazo in trenutni format. Oznake formatov so naslednje: TEI (Text Encoding Initiative), LMF (Lexical Markup Framework) in LBS (Leksikalna baza za slovenščino).

Prikazani podatki	Vir podatkov	Vključenost podatkov v bazo	Trenutni format
besedne zveze	izluščeni podatki	DA, kot leksikon	XML LBS
besedne zveze - konkordance	Gigafida (Korpus slovenskega jezika Gigafida)	NE, povezava na konkordančnik	-
besedne oblike	Sloleks (Slovenski oblikoslovni leksikon Sloleks)	DA, kot leksikon	XML LMF
sinonimi in prevodi v izbrane tuje jezike	sloWNet (Slovenski semantični leksikon sloWNet)	DA, kot leksikon	XML DEBDIC
zgodovina, besede	IMP (Korpus starejše slovenščine IMP)	DA, kot leksikon	XML TEI
zgodovina - konkordance	IMP (Korpus starejše slovenščine IMP)	NE, povezava na konkordančnik	-
govor, besede	Gos (Korpus govorjene slovenščine Gos)	DA, kot leksikon	(XML TEI - izvedba v projektu)
govor - konkordance	Gos (Korpus govorjene slovenščine Gos)	NE, povezava na konkordančnik	-
vizualizacija relacij	izluščeni podatki	DA	XML LBS
multimedija	WikiMedia, ...	DA, tudi kot zunanji viri	različni multimedijski formati
jezikovna statistika	Gigafida (Korpus slovenskega jezika Gigafida)	DA	-

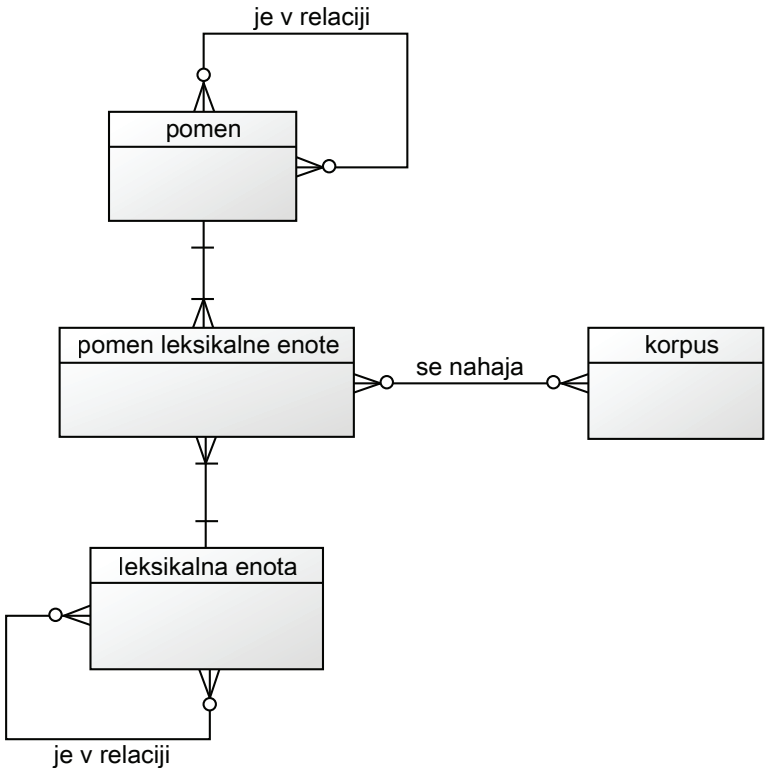
Če se osredotočimo na tekstovne oblike virov, so večinoma v zapisu XML ali v običajnih tekstovnih datotekah. XML poleg vsebinskih podatkov vsebuje tudi podatke o strukturi. Struktura podatkov v različnih virih ni enaka (tudi zaradi vsebine, ki jo pokrivajo), zato strukture XML v bazi ni smiselno ohranjati (obstajajo tudi izjeme, kjer je smiselno ohraniti manjše dele, npr. poudarki pri opisih). XML je po naravi hierarhična oblika hrambe podatkov, ki pa ni najbolj primerna za hranjenje podatkov, ki niso hierarhične narave (kot v primeru slovarjev). Vendar pa je oblika XML zaradi te hierarhičnosti precej primerna za serializacijo, poleg tega pa sama datoteka XML vsebuje podatke o strukturi podatkov. Zaradi teh dveh razlogov je primerna za izmenjavo podatkov (z zunanjimi viri in z zunanjimi aplikacijami, ki preko programskih vmesnikov dostopajo do slovarja).

Pri podatkih v slovarski bazi so pomembne medsebojne relacije med posameznimi zapisi. Za modeliranje teh relacij so primerne grafovske podatkovne baze in relacijske podatkovne baze. Glede na zmogljivosti (Vicknair et al. 2010) sta obe ti vrsti podatkovnih baz primerni za velike količine podatkov, ki se pojavljajo pri slovarjih. Obe vrsti imata definirane poizvedovalne jezike: pri grafovskih bazah npr. SPARQL (SPARQL Query Language for RDF) in več nestandardnih rešitev (Wood 2012, Haase et al. 2004), pri relacijskih podatkovnih bazah pa sta uveljavljena standarda SQL in SQL/PSM. Grafovske baze so precej fleksibilne, saj nimajo eksplicitno definirane strukture. Primerne so za podatke, ki imajo zelo variabilno strukturo. Relacijske podatkovne baze imajo eksplicitno definirano strukturo, zato je potrebno vnaprej dobro definirati podatkovni model. Hkrati tako tudi vnaprej načrtujemo, katere poizvedbe nad bazo so možne in katere ne. Vseeno lahko tudi relacijski model prilagajamo tako, da del strukture hranimo kot podatke (Newman 2007).

Multimedijski viri se hranijo kot referenca in pred vključitvijo podatkov v bazo ustrezno tekstovno označijo, tako da jih lahko lažje preiskujemo.

Slovarska baza je zaradi zrelosti tehnoloških rešitev načrtovana kot relacijska podatkovna baza. Poenostavljen konceptualni model jedra podatkovne baze je prikazan na Sliki 2. Leksikalna enota nosi en pomen ali več pomenov. Pomeni so lahko med seboj v različnih relacijah. Leksikalne enote so lahko leksemi, stalne zveze ali fraze, tudi deli besed in so lahko med seboj v različnih relacijah. Za leksikalne enote z določenim pomenom hranimo (agregirane) podatke o tem, v katerih virih smo jih našli.

Model je zasnovan dovolj splošno, da omogoča širitev obsega hranjenih podatkov na več jezikovnih zvrsti in jih obravnava enakovredno. Poleg tega odločitve pri gradnji podatkovnega modela določajo tudi stopnjo granularnosti podatkov (manjša



Slika 2: Poenostavljen konceptualni model jedra podatkovne baze v Martinovi notaciji, ki služi kot izhodišče za načrtovanje celotne baze.

stopnja granularnosti pomeni, da hranimo več agregiranih podatkov ali manj natančne podatke, posledično na določene poizvedbe ne bomo mogli odgovoriti). Granularnost je pomembna pri luščenju podatkov in polnjenju baze, ker določa, kaj vse je potrebno izluščiti in kakšno dodatno delo bo potrebno opraviti, npr. pri množičenju in končni leksikografski obdelavi. Na primer: če se pri luščenju podatkov za posamezne leksikalne enote ne beleži podatkov o časovnem razponu pojavljanja (recimo na podlagi pojavitev v korpusih v določenem obdobju), potem ne bo možno omejiti poizvedb na besedišče iz določenega obdobja.

Pri postavitvi podatkovne baze imamo na izbiro več sistemov za upravljanje s podatkovnimi bazami. Ker so relacijske podatkovne baze precej uveljavljene, obstaja več odprtokodnih rešitev, vendar vse nimajo potrebnih funkcionalnosti. Sistem za upravljanje s podatkovno bazo mora podpirati tudi t. i. rekurzivne poizvedbe in SQL/PSM (v bazi shranjene procedure), zato smo omejeni na sisteme, ki le-te podpirajo (npr. PostgreSQL[1]).

1 www.postgresql.org (dostop 12. 6. 2015).

3 ZALEDNI APLIKACIJSKI SISTEM

Zaledni aplikacijski sistem predstavlja vmesni sloj med podatkovnim nivojem in predstavitvenim nivojem. Avtomatsko luščenje podatkov je sicer del zalednega aplikacijskega sistema, vendar ga zaradi obsežnosti obravnavamo v posebnem razdelku. Naloga sistema je, da zahteve po podatkih, ki jih prejme od predstavitvenega nivoja, ustrezno (pre)oblikuje in pošlje podatkovni bazi oziroma zunanjim virom (korpusom, zunanjim bazam). Njihove odgovore ustrezno obdela, prečisti in posreduje predstavitvenemu nivoju. Tukaj je pomembno ločiti med podatki in dodatnimi omejitvami in pravili, ki jih definiramo nad podatki, vendar pa se lahko te omejitve in pravila s časom tudi spreminjajo. Na primer, kolokacije, ki se pojavljajo pri posamezni leksikalni enoti, lahko beležimo čez daljše obdobje. Če želimo privzeto izpisovati samo kolokacije, ki so se ob leksikalni enoti pojavljale v določenem časovnem obdobju, npr. od leta 2005 do 2015 (zadnjih 10 let), potem je definicija tega kolokacijskega obdobja pravilo – tekom časa se kolokacije spreminjajo. Naloga aplikacijskega nivoja je, da omogoča definiranje takšnih pravil in ustrezno formulira poizvedbe na podatkovni bazi glede na dana pravila in omejitve. To ne pomeni, da smo omejeni na obdobja, ki jih določajo omejitve; preko uporabniškega vmesnika lahko eksplicitno določimo obdobje (ali pa ustrezno označimo ostale kolokacije pomena).

Aplikacijski sistem svoje storitve ponuja v obliki programskega vmesnika. Prednost ločenih nivojev je, da se lahko programska koda aplikacijskega nivoja spreminja (dopolnjuje, popravlja, izboljšuje), medtem ko ostane programski vmesnik enak in lahko odjemalci predstavitvenega nivoja (spletna aplikacija, mobilne aplikacije) brez težav dostopajo do storitev. Poleg odjemalcev predstavitvenega nivoja je potrebno omogočiti dostop tudi drugim računalniškim sistemom, ki bi želeli dostopati do podatkov, in omogočiti povezljivost v smislu semantičnega spleta (povezani podatki, angl. *linked data*).

Ker komunikacija med predstavitvenim nivojem in aplikacijskim nivojem deluje po principu odjemalec-strežnik, je pomembna naloga aplikacijskega nivoja, da se podatki pripravijo tako, da odjemalci dobijo le tiste podatke, ki jih nujno potrebujejo, in ni nepotrebnih prenosov.

4 AVTOMATSKO LUŠČENJE PODATKOV

Podatki v slovarski bazi so izluščeni iz različnih zunanjih virov, ki jih prikazuje Tabela 1. Pri luščenju podatkov se srečamo z dvema poglavitnima problemoma: količina podatkov, ki predstavlja vir, iz katerega luščimo (npr. korpus Gigafida

vsebuje približno 1,2 milijardi besed), in kako zagotoviti kakovost izluščenih podatkov. Proces luščenja zaradi časovne dinamičnosti jezika ni zaključen z objavo slovarja, ampak je trajen proces. Zaradi zgornjih zahtev se luščenje izvaja v prvem fazi avtomatsko, rezultate te faze ustrezno ovrednotimo in tiste z veliko stopnjo zanesljivosti vpišemo v bazo, rezultate z manjšo zanesljivostjo pa pošljemo v fazo čiščenja in ročnega obdelovanja.

Pri avtomatski fazi izhajamo iz metod luščenja podatkov, ki so bile razvite za potrebe sestavljanja Leksikalne baze za slovenščino v okviru projekta Sporazumevanje v slovenskem jeziku (Gantar 2009; Gantar in Krek 2011) in jih nadgradimo z novimi spoznanji in tehnološko izboljšanimi orodji. Za celotno besedišče, ki bo vizualizirano, se lahko strojno pridobi naslednje podatke: iztočnico v osnovni obliki, besedno vrsto, podatek o pogostosti v korpusu, slovnične relacije, ki se v bazi prepišejo v vzorce, ter pripadajoče kolokacije in njihovi zgledi. Za postopek avtomatizacije je že bila izdelana t. i. slovnica besednih skic, ki deluje v orodju Sketch Engine.[2] S pomočjo prilagojene programske skripte, ki vsebuje opise vseh relevantnih slovničnih relacij za luščenje kolokacij, t. i. konfiguracije GDEX (okrajšava za angl. *Good Dictionary Examples*), ki opredeli lastnosti dobrih zgledov, lahko iz korpusov avtomatsko pridobimo čim boljše kandidate za primere uporabe posameznih iztočnic v realnem besedilnem okolju (Kosem et al. 2011).

V drugi fazi se podatki pred vključitvijo v slovarsko bazo ročno pregledajo. Delo se opravlja s pomočjo množičenja, kjer uporabniki označujejo, ali so v rezultatih anomalije oz. napake. Na koncu podatke preuredi in potrdi leksikograf. Potrjene napake, ki so posledica avtomatskega luščenja, se označijo in vračajo kot informacija nazaj v sistem luščenja, ki se iz njih uči z uporabo tehnik strojnega učenja in tako izboljšuje svoje delovanje.

Avtomatsko luščenje podatkov spada v zaledni sistem. Delno in končno obdelane podatke zapisujemo v slovarsko podatkovno bazo. Vsi podatki, ki niso dokončno obdelani, so v bazi ustrezno označeni, kar pomeni, da jih lahko na predstavitvenem nivoju bodisi prikažemo bodisi ne prikažemo. Na primer, leksikograf in splošni uporabnik dostopata do iste baze, vendar bo leksikograf poleg drugačnega uporabniškega vmesnika videl tudi podatke, ki niso dokončno obdelani, in jih lahko ustrezno obdeloval. Tudi uporabniki, ki sodelujejo v množičenju, imajo svoj pogled na podatke. Za množičenje se lahko uporabijo obstoječe platforme, kot je npr. PyBossa[3], ki omogočajo preprostejšo izdelavo aplikacij za množičenje (prim. Fišer et al. 2015).

2 http://www.sketchengine.co.uk/ (dostop 12. 6. 2015).
3 http://pybossa.com/ (dostop 12. 6. 2015).

5 PREDSTAVITVENI NIVO: SPLETNI PORTAL IN MOBILNE APLIKACIJE

Predstavitveni nivo mora zaznamovati uporabniška izkušnja in s tem posledično ustrezen uporabniški vmesnik aplikacij, kar ima velik vpliv na njihovo uspešnost. Zelo pomembna je tudi celostna grafična podobna aplikacij. Namen predstavitvenega nivoja je tudi v kar najbolj podobni obliki prikazovati informacije na spletnih straneh in priljubljenih mobilnih platformah.

Za razvoj mobilnih aplikacij je smiselno uporabiti t. i. hibridni pristop, ki predstavlja najboljši način za prenosljivost aplikacij med različnimi mobilnimi platformami pri čim višji ponovni uporabljivosti posameznih razvitih delov. Pri tem je trenutno za razvoj osnovnih funkcionalnosti smiselno uporabiti tehnologiji HTML5 in Javascript. Na tak način razvito jedro aplikacije lahko nato umestimo v aplikacijsko ogrodje posamezne podprte platforme. Takšen razvoj podpirajo številna odprtokodna orodja, npr. PhoneGap,[4] ki temelji na platformi Apache Cordova.[5] To olajša in pospeši razvoj aplikacij za vse podprte platforme, zagotavlja pa tudi enoten predstavitveni nivo na vseh platformah, kot tudi poenostavljeno posodabljanje aplikacij. Osnova tako razvite mobilne aplikacije je lahko osnova pri razvoju spletnega portala.

Z namenom prepoznavnosti in enotne uporabniške izkušnje pri uporabi aplikacij je smiselno zasnovati celostno grafično podobo uporabniškega vmesnika. Pomembno je, da zasnova upošteva standard WCAG 2.0 (Web Content Accessibility Guidelines 2.0), s čimer je omogočena tudi raba uporabnikom s posebnimi potrebami.

6 ZAKLJUČEK

Pri tehnološki izvedbi sodobnega digitalnega slovarja slovenskega jezika je ključna ločitev predstavitve podatkov od podatkov samih. Podatke na ta način lahko hranimo v vsej njihovi kompleksnosti, predstavitev podatkov pa je mogoča z različnih zornih kotov in omogoča tako spremembo gledišča kot stopnjo podrobnosti predstavitve. Arhitekturno je tehnološka izvedba zasnovana trinivojsko, kjer imamo predstavitveni nivo, vmesni aplikacijski nivo in podatkovni nivo. Naloga predstavitvenega nivoja je, da uporabniku prikaže podatke, ki so shranjeni na podatkovnem nivoju. Med obema nivojema je vmesni aplikacijski nivo, ki pretvori uporabnikove poizvedbe s predstavitvenega nivoja v obliko, ki je primerna za poizvedovanje direktno na podatkovnem nivoju (podatkovni bazi). Na drugi strani pa vmesna aplikacijska plast preoblikuje podatke v obliko, ki jo potrebuje predstavitveni nivo. Še ena funkcionalnost vmesnega aplikacijskega nivoja je

4 http://phonegap.com/ (dostop 12. 6. 2015).
5 https://cordova.apache.org/ (dostop 12. 6. 2015).

avtomatizirano luščenje podatkov iz korpusov in zunanjih zbirk podatkov. Ker se jezik razvija, je avtomatizirano luščenje podatkov stalen proces, pri katerem sodelujejo tudi leksikografi, ki dostopajo do podatkov preko ustreznih prikazov predstavitvenega nivoja. Ločitev med podatki in predstavitvijo je ključen dejavnik pri integraciji različnih virov (korpusov in zunanjih zbirk podatkov) v enotno podatkovno bazo. Različni uporabniki ali pa tudi zunanji računalniški sistemi nato s pomočjo poizvedb, posredovanih preko predstavitvenega nivoja, pridobijo želene podatke iz podatkovne baze, ki jih potem prikaže predstavitveni nivo.

Prednost nivojske arhitekture je neodvisnost posameznih nivojev, dokler je programski vmesnik, preko katerega višji nivoji dostopajo do nižjih, ustrezno definiran. Na vsakem nivoju lahko zato izberemo najustreznejše tehnologije za izvedbo in sprememba na enem od nivojev nima večjega vpliva na ostale nivoje, dokler se programski vmesnik ne spreminja. Tehnološko zasnovo slovarja lahko z upoštevanjem nivojske arhitekture razdelimo na 4 komponente: podatkovno bazo (podatkovni nivo), zaledni aplikacijski sistem s komponento za delno avtomatizirano luščenje podatkov (oboje spada v vmesni aplikacijski nivo, vendar obravnavamo komponento za luščenje podatkov ločeno zaradi njenega obsega in pomena za celoten sistem) in predstavitveni del, kjer imamo spletni portal in mobilne aplikacije (predstavitveni nivo).

Opisana tehnološka zasnova slovarja zagotavlja, da na njej sloneča rešitev služi kot osrednji interaktivni spletni jezikovni portal z opisom vseh ravnin besedišča slovenskega jezika. Komponente omogočajo trajnostni razvoj portala, namenjenega tako uporabnikom spleta kot mobilnih naprav in bi bile pod prosto dostopno licenco na voljo za nadaljnji razvoj.