

Uporaba pomnilnika prevodov pri prevajanju uporabniških vmesnikov v prostih programih

Primož Peterlin*, Andraž Tori†, Marko Samastur†, Roman Maurer‡, Aleš Košir‡

*Inštitut za biofiziko, Medicinska fakulteta, Univerza v Ljubljani
Lipičeva 2, SI-1000 Ljubljana, Slovenija
primoz.peterlin@biofiz.mf.uni-lj.si

†Univerza v Ljubljani,
andraz.tori1@guest.arnes.si, markos@elite.org

‡Hermes SoftLab
Litijska 51, SI-1000 Ljubljana, Slovenija
{roman.maurer, ales.kosir}@hermes.si

Povzetek

Prosto programje, kakršno je sistem Linux z namizjema KDE ali GNOME in programi iz projekta GNU, prevajajo skupine prostovoljcev. Navadno vsak prostovoljec prevaja posamične, logično povezane dele večje celote. Ker imajo prostovoljci različno jezikovno, a dobro programersko znanje, hkrati pa je tehnični jezik prostega programja neuveljavljen in neenoten, je pri teh projektih pomoč v obliki programov za zagotavljanje usklajenosti prevodov še pomembnejša kot pri krajevnih prilagoditvah komercialnih programov, kjer navadno delujejo manjše, povezane, strokovno izenačene skupine znotraj podjetij. Skupina za slovenjenje Linuxa si pomaga z lastnim izdelkom, programom Smart. Program, namenjen namigom pri prevajanju novih fraz ter usklajevanju obstoječih prevodov, iz dvojezičnega angleško-slovenskega poravnane korpusa obstoječih prevodov sestavi podatkovno zbirko, ki jo podpira strežnik SQL, in jo uporabi kot pomnilnik prevodov.

1. Uvod

Proti koncu osemdesetih let, predvsem pa v devetdesetih letih, so postala tržišča izven angleškega govornega območja za industrijo programske opreme dovolj pomembna, da so večje programerske hiše začele razmišljati o krajevnem prilagajanju (lokalizaciji) programja. Sredi devetdesetih let so se temu trendu pridružili tudi pisci prostega programja. Pri projektu GNU, ki je bil od sredine osemdesetih let dalje eden glavnih zagovornikov zamisli o prostem programju (Stallman, 1985), je prvo pobudo za prilagajanje dal poleti 1994 Patrick D’Cruze (D’Cruze, 1994). Od nekaj predlaganih prototipnih izvedb krajevnega prilagajanja se je kot najobetavnejši pokazal sistem GNU Gettext (Drepper, 1996).

Prevajanje komunikacije programov z uporabnikom je iz več razlogov primerno za uporabo računalniško podprtega prevajanja (Computer Assisted Translation, CAT). Po eni strani nas v to silo potreba: terminologija v večjem programskem sklopu, kot je na primer namizje, mora biti usklajena, saj uporabnikov ne zanima, kdo, kdaj in kako je prevajal posamezne sestavine namizja. Posebej pri ohlapno povezanih skupinah prostovoljcev, ki navadno prevajajo prosto programje, je zato programska podpora usklajenosti še kako dobrodošla.

Po drugi strani pa so tovrstna besedila tudi iz tehničnih razlogov posebej primerna za računalniško podprto prevajanje. Zaradi tehnične izvedbe – kot zgled obdelamo sistem GNU Gettext, velja pa splošneje – prevodi že sami po sebi predstavljajo dvojezični korpus, večinoma poravnani na ravni povedi (Erjavec, 1999). Ker gre za besedila tehnične narave z omejenim enoznačnim besediščem, mnogimi po-

navljajočimi se termini in besednimi zvezami, pa je velikost, pri kateri takšen korpus že postane uporaben, bistveno manjša kot pa pri splošnonamenskih jezikovnih korpusih (Webb, 2000).

Ustaljeno poimenovanje za takšen dvojezični poravnani korpus je *pomnilnik prevodov* (angl. Translation Memory, TM) (Vintar, 1999a; Vintar, 1999b). Po definiciji skupine EAGLES je to “večjezični besedilni arhiv, ki vsebuje (segmentirana, poravnana, razčlenjena in klasificirana) večjezična besedila ter dovoljuje shranjevanje besedil in iskanje po njih glede na različne pogoje” (EAGLES, 1996).

Zamisel o računalniški podpori prevajanju uporabniškega vmesnika programov ni nova ne med prostim programjem ne sicer. Projekt prosto dostopnega namizja KDE vključuje razvoj programa Kbabel (Kiefer, 2000a), ki se v mnogočem zgleduje po nekaj starejšem Ktranslatorju (Rizzi, 1999), ter večjezični spletni glosar Kdedict (Kiefer, 2000b). Pri projektu namizja GNOME pa za računalniško podporo prevajanju razvijajo orodja Gtranslator (Fatih, 2000), Muli (Reichl, 2000) in Gnopo (Perseus, 2000). Ne smemo pozabiti tudi na starost med orodji za računalniško podprto prevajanje, to je na urejevalniški način po za urejevalnik Emacs. Način je standardni del paketa GNU Gettext.

Podobne, vendar komercialne rešitve obstajajo tudi v drugih okoljih; omenimo naj le Atril Déjà Vu (<http://www.atril.com/>), Trados Workbench (<http://www.trados.com/>) in IBM Translation Manager (<http://www-4.ibm.com/software/ad/translat/tm/>), celovitejši in natančnejši pregled pa je na voljo v (Vintar, 1999a).

2. Program Smart

V prispevku predstavljamo pri nas razvit program Smart (Tori, 2000) za pomoč pri prevajanju in preverjanje skladnosti prevodov. Iz poravnanih parov fraz izvornika in prevoda Smart sestavi relacijsko zbirko podatkov. Trenutno fond fraz obsega blizu 30.000 poravnanih fraz v angleščini in slovenščini (tabela 1). Slednje izvirajo večinoma iz namizij KDE in GNOME ter programov GNU, ki jih poganjamo v ukazni vrstici.

Ko je zbirka sestavljena, lahko zbrane prevode uporabimo pri prevajanju novih. Program Smart (Tori, 2000) prečeše datoteko z neprevedenimi frazami in ob vsaki neprevedeni frazi poskusi poiskati ustreznico v zbirki. Algoritem ujemanja pametno razlikuje med velikimi in malimi črkami, upošteva prazne prostore in druge posebne znake, ki se uporabljajo kot bližnjice v okoljih.

Predstavljena rešitev ponuja v primerjavi s prej naštetimi boljšo podporo za slovenščino, je pa tudi od vseh naštetih najbolj neodvisna od strojnega in programskega okolja. Od splošnonamenskih programov predstavlja boljšo podporo za prevajanje sporočil v prostih programih, njena prednost pa je tudi ta, da obvlada tako posebnosti programov namizja KDE kot tudi programov namizja GNOME.

3. Uporabljena orodja

Predlagana rešitev uporablja vrsto orodij, prosto dosegljivih v internetu. To so GNU Gettext, knjižnica in nabor namenskih programov za krajevno prilagoditev; predmetni relacijski podatkovni strežnik PostgreSQL, programski jezik Perl ter Perl DBI, vmesnik med Perlom in podatkovno zbirko.

3.1. GNU Gettext

GNU Gettext (Drepper et al., 1998) je bil prvi uporabni prosti sistem za krajevno prilagajanje, zato ga danes uporablja velika večina proste programske opreme, njegova uporaba pa se zaradi dobrih lastnosti sistema vse bolj širi tudi na komercialna orodja in komercialne programske pakete.

Programsko opremo za njegovo uporabo napišemo tako, da jo internacionaliziramo, kar pomeni, da posplošimo vse njene kulturne, geografske in jezikovne lastnosti do te mere, da programi sami ne vsebujejo teh krajevnih podatkov. Namesto tega operacijski sistem po nastavitvi uporabnika poskrbi za čimvečjo lokalizacijo teh metod na uporabnikovo geografsko in kulturno območje. Sporočila, s katerimi se sporazumevamo s programom, pa so izvedena s sistemom GNU Gettext.

Sistem GNU Gettext pričakuje prevode sporočil v jeziku uporabnikov tako, da program uporabi posebne datoteke PO (Portable Object), ki obsegajo pare fraz v izvorniku in prevodu, denimo:

```
msgid "error while reading %s"
msgstr "napaka pri branju %s"
```

Pri tem zapis `msgid` označuje izvorno, `msgstr` pa prevedeno frazo. Oznaka `%s` pomeni, da program na tisto mesto vrine niz znakov, ki jih dobi funkcija za izpis kot svoj argument. V danem zgledu je to ime datoteke. Oznake v

frazah so enake, kot jih določa standard POSIX za formatna določila funkcije `printf(3)` v programskem jeziku C.

Izkušnje kažejo (tabela 1), da fraze večinoma niso daljše od ene vrstice – povprečje na našem vzorcu je približno 1,25 vrstice na frazo – seveda pa datoteke PO dopuščajo tudi fraze, ki se raztezajo prek več vrstic. V tem primeru je niz pri `msgid` prazen, besedilo izvornika pa sledi v vrsticah do oznake `msgstr`, kot kaže zgled:

```
msgid ""
"kfontmanager Version 0.2\n"
"Copyright 1997\n"
"Bernd Johannes Wuebben\n"
"wuebben@math.cornell.edu\n"
msgstr ""
"kfontmanager, različica 0.2\n"
"Avtorske pravice 1997\n"
"Bernd Johannes Wuebben\n"
"wuebben@math.cornell.edu\n"
```

Oznake `\n` v besedilu določajo, kje naj se vrstice s sporočili pri izpisu v programu zaključijo. Oblika datotek PO ob frazah dopušča še komentarje, bodisi komentarje prevajalca ali avtorja programa, bodisi strojno generirane.

Zaradi togosti osnovne različice sistema GNU Gettext, ki ni dovoljevala zamenjave vrstnega reda argumentov pri izpisu sporočila, je bil sistem dopolnjen. Druga izdaja namizja KDE uporablja razširjeno obliko datotek PO, kjer določimo zaporedno številko argumenta, ne pa njegove vrste. Oznaka `%1` tako pomeni prvi argument v izvorniku, `%2` drugega in tako dalje. Ta razširitev omogoča, da se vrstna reda argumentov v izvorniku in prevodu razlikujeta, kot prikazuje zgled:

```
msgid ""
"File %1 cannot be opened by tool %2."
msgstr ""
"Orodje %2 ne more odpreti "
"datoteke %1."
```

Druga razširitev, ki so jo v obliko datotek PO vnesli pri KDE, pa je pojasnilo sporočila. Niz `_:` na začetku izvorne fraze pomeni pojasnilo, ki se pri prevodu preskoči. Služi za boljše razumevanje in razločevanje konteksta, v katerem je sporočilo. S to razširitvijo je premagana ena glavnih pomanjkljivosti sistema GNU Gettext.

```
msgid ""
"_: forward call\n"
"&Forward"
msgstr "&Posreduj"
```

Pomen znaka `&` bomo opisali v razdelku o bližnjicah.

Sistem GNU Gettext sestavlja programska knjižnica `libintl` ter nabor namenskih programov: `gettext`, s katerim lahko izpisujemo prevode sporočil iz kataloga sporočil; `msgcmp`, s katerim primerjamo dve datoteki PO; `msgcomm`, s katerimi poiščemo fraze, skupne dvema datotekama PO; `msgfmt`, s katerim prevedemo datoteko PO v strojno berljivo obliko MO (Message Object); `msgmerge`, s katerim združimo dve datoteki PO; `msgunfmt`, s katerim sporočila pretvorimo iz prevedene oblike MO v človeku

berljivo obliko PO, in `xgettext`, s katerim samodejno izluščimo iz izvorne kode programa vsa sporočila.

Kako deluje mehanizem `Gettext`? Ob prevajanju prevajalnik predmetno kodo programa poveže s knjižnico `libintl` in v program shrani podatke o tem, kje naj išče kataloge prevedenih sporočil. Ti so shranjeni v prevedeni obliki MO v podimenikih systemskega imenika `/usr/share/locale/`. Ob izvajanju poskusi program glede na nastavitve spremenljivk okolja `LC_MESSAGES`, `LC_ALL` ali `LANG` poiskati katalog sporočil za želeni jezik. Vsebino teh spremenljivk nastavlja uporabnik ali skrbnik sistema. Če je šlo iskanje in nalaganje datotek s sporočili brez napak, program nadomesti angleška sporočila s krajevno prilagojenimi. Če iskanega kataloga sporočil ni ali če je nepopoln, pa kot zasilna različica vselej ostaja izvorno angleško sporočilo.

3.2. PostgreSQL

Program `Smart` zaradi hitrega dostopa do zapisov te hrani v relacijski zbirki `PostgreSQL`. Predmetni relacijski podatkovni strežnik `PostgreSQL` ima dolgo zgodovino. Njegove korenine segajo do relacijske podatkovne zbirke `Ingres` (Stonebraker et al., 1976), ki so jo razvijali na kalifornijski univerzi v Berkeleyju konec sedemdesetih in v prvi polovici osemdesetih let. Razvoj `Ingres` je do tržnega izdelka nadaljevalo podjetje `Relational Technologies/Ingres Corporation`, ki je pozneje postalo del velikana `Computer Associates`.

Skupina pod vodstvom Michaela Stonebrakerja na isti univerzi pa je nadaljevala razvoj predmetno-relacijskega podatkovnega strežnika `Postgres` (Stonebraker and Rowe, 1986; Stonebraker et al., 1990). Tega je v tržni izdelek razvilo podjetje `Illustra`, ki je pozneje postalo del podjetja `Informix`. Dva podiplomska študenta profesorja Stonebrakerja, Jolly Chen in Andrew Yu, sta med letoma 1994 in 1996 nadomestila jezik `PostQuel`, ki ga je uporabljal `Postgres`, s standardnim sestavljenim jezikom za poizvedbe (`Structured Query Language, SQL`). Tega so začeli v začetku sedemdesetih let razvijati pri `IBM` (Codd, 1970). Tako dopolnjeni `Postgres` sta poimenovala `Postgres95` (Yu and Chen, 1995). Chen in Yu sta zapustilay Berkeley, vendar pa je Chen nadaljeval z razvojem in poleti 1996 je vzdrževanje in razvoj projekta prenesel na skupino prostovoljcev z interneta. Da bi poudarili podporo za jezik `SQL`, so ime pozneje spremenili v `PostgreSQL` (Lockhart, 2000).

Danes ožje jedro razvijalcev in vzdrževalcev podatkovnega strežnika `PostgreSQL` predstavlja ob šestčlanskem svetu še sedemnajstčlanska ekipa programerjev s treh celin. Spletna stran projekta je <http://www.postgresql.org/>. V času pisanja tega prispevka je zadnja izdaja nosila številko 7.0.2.

3.3. Perl

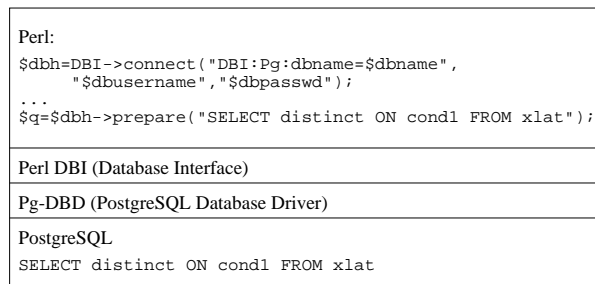
Programski jezik `perl` (`Practical Extraction and Report Language`, priročni jezik za izvlečke in poročila) in tolmač zanj je napisal Larry Wall leta 1989 (Wall et al., 2000). Po avtorjevih besedah je poskusil v njem združiti najboljše lastnosti lupinskih skriptov, programskega jezika `C` ter orodij `Sed` in `Awk`.

Perl se še posebej odlikuje po razširjenem naboru *regularnih izrazov* (Friedl, 1998), zato je izjemno močno orodje pri delu z besedili.

V času tega prispevka zadnja izdaja Perla nosi številko 5.6. Informacije o Perlu najdemo v spletu na naslovu <http://www.perl.org/>.

3.4. Perl DBI

Prvo različico vmesnika med `perlom` in podatkovnimi strežniki (Descartes and Bunce, 2000) je leta 1994 napisal Tim Bunce. `Perl DBI` (`Database Interface`) definira skupino postopkov, spremenljivk in dogovorov, ki skupaj sestavljajo enoten programerski vmesnik do različnih podatkovnih strežnikov. Bunce je začel s podporo za `Oracle`; danes pa so na voljo gonilniki (`Database Driver, DBD`) za številne podatkovne strežnike, med drugimi `Adabas`, `DB2`, `Empress`, `Illustra`, `Informix`, `Ingres`, `InterBase`, `mSQL`, `MySQL`, `ODBC` (`Microsoft SQL Server`), `Oracle`, `PostgreSQL`, `Sybase` in `XBase`.



Slika 1: Shema povezave med Perlom in zbirko podatkov.

Slika 1 shematično prikazuje odnos sestavnih delov, ki sestavljajo sistem. `Perl` je splošno namenski programski jezik, katerega posebna odlika so močni ukazi za delo z nizi znakov. `Perl:DBI` je splošni vmesnik do zbirk podatkov, ki mora biti podprt še z gonilnikom za izbrano podatkovno zbirko (v našem primeru `DBD:Pg`). Ta ukaze prevede v poizvedbeni ukaz strežniku `PostgreSQL`. Slednji prevzame skrb za evidenco o razporeditvi podatkov na disku, zato se programerju eno raven više s tem ni treba ukvarjati.

Z vmesnikom do zbirk podatkov ob že sicer močnih ukazih za delo z besedili postane `Perl` še močnejši.

4. Razširjeno ujemanje nizov

Če se povrnemo k opisu datotek `PO`, vidimo, da so v njih vsebinske in skladišne prvine precej prepletene. Ker je na voljo razmeroma skromni nabor prevodov, bi bilo ob iskanju ujemačnega se niza potratno zaradi morebitnih manjših skladišnih razlik zavreči celotno vsebino prevoda. Program ima zato vgrajenih nekaj mehanizmov, ki poskušajo v čim večji meri ločiti vsebino od oblikovnih in skladišnih prvin, in primerjati vsebino fraz.

4.1. Male in velike črke

Prva skladišna težava je razlikovanje med malimi in velikimi črkami. Denimo, da imamo v zbirki prevodov že par:

Programski paket	Število datotek	Število fraz (msgid)	Število vrstic	Povprečno vrstic/fraza	Skupaj znakov
GNOME	36	10263	11195	1.0908	229 965
GNU	13	2841	5184	1.8247	217 954
KDE 1.1.2	91	6373	7761	1.2178	142 932
KDE 2.0	208	17719	21745	1.2277	473 586
Kdeadmin	8	1320	1495	1.1326	34 935
Kdebase	73	3459	4672	1.3507	136 414
Kdegames	23	1368	1501	1.0972	23 617
Kdegraphics	12	864	1067	1.2350	19 055
Kdemultimedia	9	724	1003	1.3854	21 300
Kdenetwork	22	2877	3724	1.2944	78 409
Kdepim	5	904	936	1.0354	12 426
Kdesdk	3	483	944	1.9545	30 713
Kdetoys	5	94	114	1.2128	2 233
Kdeutils	31	2028	2416	1.1913	48 736
Koffice	15	2254	2445	1.0847	41 126
Others	2	1337	1428	1.0681	24 622
Red Hat Linux 6.1	10	3875	10897	2.8121	542 309
Red Hat Linux 6.1	10	4202	11825	2.8141	594 479

Tabela 1: Statistika prevedenih programskih paketov. Namizje KDE 2.0 je podrobno razčlenjeno po tematskih sklopih.

```
msgid "About GNOME"
msgstr "O namizju GNOME"
```

Ko pri prevajanju novih fraz naletimo na različico sporočila v izvorniku "About Gnome", seveda želimo, da se uporabi že obstoječi prevod fraze iz zbirke. Enakovredno obravnavanje velikih in malih črk samo po sebi ni težavno. Vendar pa nismo zadovoljni s tem, da omenjeni niz prevedemo preprosto z nizom "O namizju GNOME". Verjame, da je avtor namenoma izbral male črke, zato se poskušamo oblikovno čim bolj približati izvorniku, ki ga prevajamo. V danem primeru torej besedo izpišemo z veliko začetnico, sicer pa z malimi črkami: "Gnome".

Pravila, ki se jih Smart drži pri obravnavanju velikih črk, so naslednja:

1. Če je izvorna fraza izpisana s samimi velikimi črkami, bo takšen tudi prevod.
2. Če je izvorna fraza izpisana z veliko začetnico, bo prevod tudi.
3. V izvorniku poišče vse besede, ki so izpisane s samimi velikimi črkami (kratice), in enako izpiše tudi prevod.

4.2. Bližnjice

Bližnjice imenujemo kombinacije tipk, s katerimi lahko hitreje dosežemo izbire v menutih. Navadno so v meniju označene bližnjice tako, da je črka podčrtana: izbiramo "Šhrani" na primer dosežemo tako, da hkrati pritisnemo tipki Alt in S.

Težava nastopi, ker namizji KDE in GNOME interno uporabljata v sporočilih različni oznaki za bližnjice: KDE jih označuje z znakom &, GNOME pa z znakom _ . V resnici seveda te razlike ne želimo. Denimo, da imamo v zbirki prevodov par:

```
msgid "&Save"
msgstr "&Shrani"
```

V tem primeru seveda želimo, da program uporabi ta prevod, ko naleti na izvorni niz "_Shrani". Enako seveda želimo tudi pri tistih redkih programih namizja KDE, ki uporabljajo posebni način zapisa bližnjic v obliki Šhrani (&S).

Težavo smo rešili tako, da oba posebna znaka ob sestavljanju podatkovne zbirke nadomestimo z novim posebnim in redkim "metaznakom" ^, ob prevajanju pa se ta nadomesti z enakim znakom, kot je v izvorniku fraze.

4.3. Presledki in nečrke

Podobne težave kot z znaki, ki označujejo bližnjice, so tudi s presledki in ostalimi oblikovnimi elementi. Pogosti so uvodni ali zaključni presledki ter zaključno dvopičje ali tropičje. Pri iskanju prevoda želimo, da so naslednji trije nizi pomensko enakovredni.

```
msgid "Search"
msgid " Search "
msgid "Search:"
```

Zato pred iskanjem ujemajočega niza v zbirki niz otrebimo oblikovnih oznak in primerjamo vsebino nizov. Zaporedne stopnje pri iskanju ujemajočega niza ponazarja slika 2.

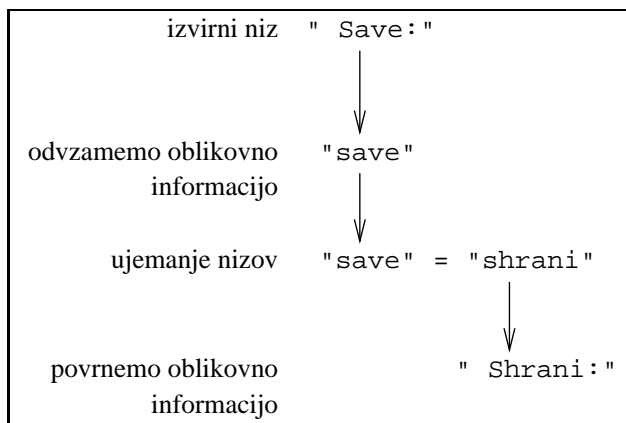
Drugi oblikovni elementi, ki se tudi pojavljajo v frazah in za katere je treba poskrbeti, so še pika (.), vejica (,), dvopičje (:), podpičje (;), vprašaj (?), klicaj (!), vezaj (-), narekovaji (' '), oglati oklepaji ([]) ter znaka za tabulator (\t) in novo vrstico (\n).

5. Izkušnje ob delu

Primerjava tabele 1 s tabelo 2 pokaže, da je prevedenega prostega programja še vedno dosti manj od komercialnega, vseeno pa delež ni zanemarljiv.

Programski paket	Število fraz	Povprečno znakov/fraza	Skupaj znakov
Microsoft Chat 2.1	1022	23.76	24 282
Microsoft Excel 97	14610	23.82	348 033
Microsoft Excel 2000	13650	23.69	323 385
Microsoft Internet Explorer 5.0	16823	30.12	506 747
Microsoft Internet Information Server 4.0	4033	52.58	212 067
Microsoft NetMeeting 2.1	4509	28.06	126 541
Microsoft Office 97	23488	21.84	513 068
Microsoft Office 2000	14217	20.25	287 829
Microsoft Outlook 97	6030	25.92	156 280
Microsoft Outlook 2000	13825	27.16	375 530
Microsoft Windows 95	22057	37.61	829 580
Microsoft Windows 98	46022	36.65	1 686 805
Microsoft Windows 98 Second Edition	37751	32.26	1 218 014
Microsoft Word 97	11147	23.77	265 012
Microsoft Word 2000	16592	23.18	384 641
Netscape Communicator 4.51	5992	42.48	254 529

Tabela 2: Primerjava: obseg sporočil v krajevno prilagojenih izdelkih podjetij Microsoft in Netscape.



Slika 2: Stopnje pri iskanju ujemajočega se niza v zbirki.

Prve izkušnje s programom Smart so – ne povsem nepričakovano – razkrile, da je obstoječa baza prevodov neuskaljena. Značilno situacijo denimo prikaže naslednji primer:

```

msgid "configure keys"
#s_msgstr "nastavi tipke"
#s_msgstr "nastavite tipke"
#s_msgstr "nastavitev tipk"
#s_msgstr "prikroji tipke"
  
```

Je na mestu velelnik v ednini ali v množini? Če besedilo razumemo kot ukaz računalniku, je verjetno bolj na mestu velelnik v ednini. Eden od prevajalcev se je dilemi izognil z uporabo glagolnika. In nazadnje najdemo še zanimiv prevod, ki poskuša rešiti težavo, da slabe pol ducata angleških izrazov (setting, setup, configuration, profile, preference) prevajamo z “nastavitev”.

Svojevrstnen rekord pa je dosegel prevod fraze “Case sensitive”, za katerega smo v zbirki našli nič manj kot sedem različnih slovenskih prevodov, v razponu od povsem

napačnih do zgolj nerodnih:

```

msgid "Case sensitive"
#s_msgstr "Loči velike in male črke"
#s_msgstr "Občutljiv na velikost črk"
#s_msgstr "Pazi na velikost črk"
#s_msgstr "Pomembnost velikosti črk"
#s_msgstr "Upoštevaj velikost črk"
#s_msgstr "Velikost črk ni pomembna"
#s_msgstr "Velikost črk pomembna"
  
```

Primer kaže še nekaj: med navedenimi sedmimi prevodi ni pravzaprav nobenega res dobrega: z “Razlikuj velike in male črke” takoj posekamo vse.

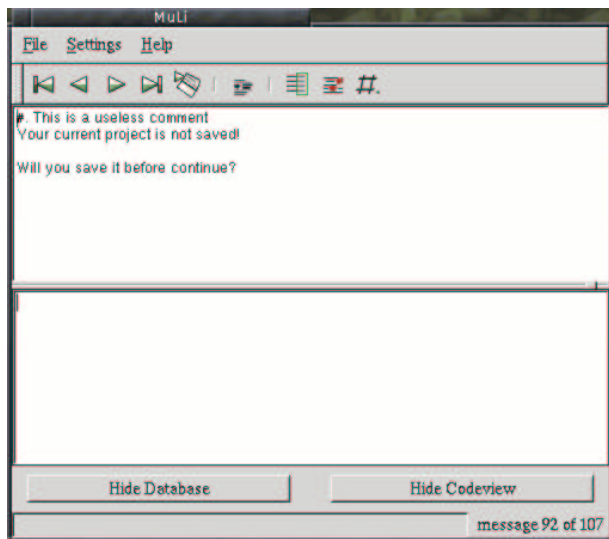
Navedeni primer je res bolj anekdotičen kot tipičen, nikakor pa ni edini. Zaradi neenotnosti gradiva v zbirki prevodov se je pokazala potreba po nekakšnem “kanoničnem” seznamu prevodov: ročno pregledan seznam z usklajenimi izrazi sproti dopolnjujemo, uporabljamo kot zgled za nove popravke in navzkrižno primerjamo obstoječe prevode.

6. Načrti za nadaljnje delo

Smart je že sedaj (v času pisanja tega prispevka je njegova zadnja različica 0.3) uporabno orodje pri prevajanju datotek PO. Vendar pa je treba v isti sapi priznati tudi, da vsi dosedanja uporabniki izvirajo iz tehnično-programerskega okolja, ne pa iz prevajalskega. Slednji bi si morda želeli prijaznejši uporabniški vmesnik.

Zato eden od načrtov zajema tudi prilagoditev Smarta za delo v tandemu s programom Muli (slika 3). Pri tem bi Muli z bolj dodelanim uporabniškim vmesnikom prevzel vlogo čelnega dela sistema, Smart z izvedenimi algoritmi za ujemanje pa zaledni del.

Poleg uporabniškega vmesnika želimo izboljšati tudi podporo za slovenščino. Zaenkrat program že vključuje preverjanje predlogov s/z in h/k (Košir et al., 1998). Dodali pa bi radi še natančnejše preverjanje jezika, pri čemer bi si



Slika 3: Glavno okno programa Muli.

lahko pomagali z rezultati projekta MULTEXT-East (Erjavec, 1998). Program bi radi dopolnili s seznamom regularnih izrazov, ki bi opisovali pogoste napačno ali slogovno oporečno rabljene besede ali besedne zveze, za vsako navedli pojasnilo, zakaj je slaba in kako jo popraviti ali izboljšati.

7. Dostopnost

V duhu zamisli odprtega programja je tudi program Smart prosto dostopen na internetnem naslovu <ftp://ftp.lugos.si/pub/lugos/localization/smart/>. Uporabniki ga lahko z navedenega naslova prepišejo na svoj disk, spreminjajo po lastnih željah in potrebah ter razširjajo dalje. Enako velja tudi za vse ostale programe, ki jih Smart uporablja.

8. Zaključek

Iz potrebe, da bi ljubiteljski prevajalci proste programske opreme poenotili tehnično izrazje in vpeljali stalne slovenske besedne zveze prevodov za najpogostejša sporočila programov v angleščini, je nastal prosto dostopni program Smart, ki svetuje uporabniku na osnovi poravnanih dvojezičnih korpusov obstoječih prevodov. Navzlic temu, da je na voljo več tovrstnih prosto dostopnih ali plačljivih orodij, ima program Smart to veliko prednost, da je namenjen prav delu s slovenskim jezikom in zato bo mogoče vanj vgrajevati po potrebah uporabnikov in zmožnostih programerjev tiste funkcije, ki se bodo ob delu pokazale za nujne ali priročne. Ker je njegova izvorna koda na voljo, si lahko obetamo hiter razvoj programa.

9. Zahvala

Avtorji se za skrbno branje rokopisa in podane popravke in pripombe, ki so pripomogle k njegovi razumljivosti in jasnosti, zahvaljujemo dr. Tomažu Erjavcu z IJS in anonimnim recenzentom, ki so sodelovali pri pripravi konference.

10. Literatura

- Edgar F. Codd. 1970. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387.
- Patrick D’Cruze. 1994. Locale tutorial. <ftp://sunsite.unc.edu/pub/Linux/utills/nls/catalogs/Incoming/locale-tutorial-0.8.txt.gz>.
- Alligator Descartes in Tim Bunce. 2000. *Programming the Perl DBI*. O’Reilly & Associates, Sebastopol, CA.
- Ulrich Drepper, Jim Meyering, in François Pinard, 1998. *GNU gettext tools, version 0.10.35: Native Language Support Library and Tools*. Free Software Foundation, Boston, MA. <http://www.gnu.org/manual/gettext-0.10.35/>.
- Ulrich Drepper. 1996. Internationalization in the GNU project. V: *Proceedings of the First Conference on Freely Distributable Software, Cambridge, Massachusetts, February 3–5, 1996*, str. 47–62, Boston, MA. Free Software Foundation.
- EAGLES. 1996. Evaluation of natural language processing systems: Final report. Tehnično poročilo, Expert Advisory Group on Language Engineering Standards. <http://issco-www.unige.ch/projects/ewg96/>.
- Tomaž Erjavec. 1998. The MULTEXT-East Slovene lexicon. V: *Zbornik sedme Elektrotehniške in računalniške konference ERK '98, 24.–26. september 1998, Portorož, Slovenija*, zvezek B, str. 189–192. IEEE Region 8, Slovenska sekcija IEEE.
- Tomaž Erjavec. 1999. Slovensko-angleški korpus ELAN. *Slavistična revija*, 47(4):515–522.
- Demir Fatih. 2000. Gtranslator. Spletna stran. <http://gtranslator.sourceforge.net/>.
- Jeffrey E. Friedl. 1998. *Mastering Regular Expressions*. O’Reilly & Associates, Sebastopol, CA.
- Matthias Kiefer. 2000a. Kbabel. Spletna stran. <http://i18n.kde.org/tools/kbabel/>.
- Matthias Kiefer. 2000b. Kdedict. Spletna stran. <http://i18n.kde.org/tools/kdedict/>.
- Aleš Košir, Primož Peterlin, in Tomaž Erjavec. 1998. GNUsl: prosto programje in slovenščina. V: Tomaž Erjavec in Jerneja Gros, ur., *International Multi-Conference Information Society – IS'98, 6.–7. oktober 1998, Ljubljana / Multi-konferenca Informacijska družba – IS'98, Ljubljana, Slovenia. Jezikovne tehnologije za slovenski jezik : zbornik konference : proceedings of the conference*, str. 35–41, Ljubljana. Inštitut Jožef Stefan.
- Thomas Lockhart, 2000. *PostgreSQL Tutorial*. The PostgreSQL Development Team. <http://www.postgresql.org/docs/tutorial/>.
- Andreas Persenius. 2000. Gnopo. Spletni vmesnik do arhiva CVS. <http://cvs.gnome.org/bonsai/rview.cgi?cvs-root=/cvs/gnome&dir=gnopo>.
- Detlef Reichl. 2000. Muli : Multipliciter lingua. Spletna stran. <http://muli.sourceforge.net/>.
- Andrea Rizzi. 1999. Ktranslator. Spletna stran. <http://www.geocities.com/bilibao/>.
- Richard M. Stallman. 1985. The GNU manifesto. *Dr. Dobb’s Journal of Software Tools*, 10(3):30–??, March.

- Michael Stonebraker in Lawrence A. Rowe. 1986. The design of Postgres. V: Carlo Zaniolo, ur., *Proceedings of the 1986 ACM SIGMOD International Conference on Management of Data, Washington, D.C., May 28–30, 1986*, str. 340–355. ACM Press.
- Michael Stonebraker, Eugene Wong, Peter Kreps, in Gerald Held. 1976. The design and implementation of INGRES. *ACM Transactions on Database Systems*, 1(3):189–222.
- Michael Stonebraker, Lawrence A. Rowe, in Michael Hirohama. 1990. The implementation of Postgres. *IEEE Transactions on Knowledge and Data Engineering*, 2(1):125–142.
- Andraž Tori, 2000. *Smart User Manual*.
- Špela Vintar. 1999a. Računalniška orodja za prevajanje. *Mostovi*, 33(1):47–59.
- Špela Vintar. 1999b. Računalniške tehnologije za prevajanje. *Uporabna informatika*, 7(1):17–24.
- Larry Wall, Tom Christiansen, in Jon Orwant. 2000. *Programming Perl*. O'Reilly & Associates, Sebastopol, CA, 3rd izd.
- Lynn E. Webb. 2000. Advantages and disadvantages of translation memory: A cost/benefit analysis. Magistrsko delo, Monterey Institute of International Studies, Monterey, CA. <http://www.webbsnet.com/translation/thesis.html>.
- Andrew Yu in Jolly Chen, 1995. *The Postgres95 User Manual*. The POSTGRES Group, University of California, Berkeley CA.