

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERINIŲ TINKLŲ KATEDRA

Ivan Kovaliov

VERTIMO TECHNOLOGIJOS KONCEPTUALUS MODELIS IR JO
TAIKYMAS KOMPIUTERIZUOTO VERTIMO SISTEMOSE

Magistro darbas

Darbo vadovas:
doc. dr. Bronius Tamulynas

KAUNAS, 2006

TURINYS

1. ĮVADAS	4
2. VERTIMO TECHNOLOGIJŲ TIPAI	5
2.1 Termino išskyrimas ir terminologijos analizė	5
2.2 Automatinė terminologijos paieška	5
2.3 Terminologijos atitikimo patikrinimas	5
2.4 Teksto segmentacija	6
2.5 Kalbinių porų paieška vertimų atmintyje	6
2.6 Mašininis vertimas.....	6
2.7 Segmentų, formato ir gramatikos pilnumo patikrinimas	6
3. VERTIMŲ ATMINTIS (TRANSLATION MEMORY)	8
3.1 Vertimo atminties panaudojimo aspektai	8
3.2 Translation memory sistemos	8
3.2.1 Trados translator's workbench	9
3.2.2 Transit firmos Star	9
3.2.3 SDLX firmos SDL	9
3.2.4 DÉJÀ VU INTERACTIVE firmos Atril	9
3.2.5 IBM Translationmanager.....	10
4. KONCEPTUALUS MODELIS IR JO TAIKYMAS	11
4.1 Pagrindiniai darbo principai	11
4.2 Galimybių praplėtimo keliai	13
4.3 Daugialygis vertimų atminties modelis	14
4.3.1 Duomenų pristatymas	14
4.4 Paieška ir pridėjimas.....	16
4.5 Kalbinių porų susikirtimo skaičiavimai	18
4.6 Nuo kalbinių porų prie kalbinių žvaigždžių	20
4.7 Trumpas kalbos UNL aprašymas	21
4.8 Kalbinių porų persipynimo izomorfizmo problema	22
5. PROJEKTINĖ REALIZACIJA	26
5.1 Sistemos projektas	26
5.1.1 Reikalavimų projektuojamai sistemai specifikacija.	26
5.1.2 Funkciniai reikalavimai	26
5.1.3 Nefunkciniai reikalavimai	26
5.2 Kuo ypatinga programa „Transmem“?.....	27
5.3 Sisteminiai reikalavimai	27

5.4 Nustatymas	27
5.5 Apie Java Runtime Environment (Java Išpildymo Terpę)	27
5.6 Java Runtime Environment Nustatymas (Java Išpildymo Terpės).....	28
5.7. Programos „Transmem“ nustatymas	29
5.8 Svarbūs pastebėjimai	30
5.9 Vartotojo interfeisas	30
5.10 Segmentavimas	31
5.11 Vertimo procesas	31
5.12 Redagavimo funkcijos	32
5.13 Judėjimas tekstu.....	32
5.14 Pilni ir netikslūs sutapimai	32
5.15 Paieška pagal esminius žodžius ir tiksli paieška	33
5.15.1 Paieška pagal esminius žodžius.....	33
5.15.2 Tiksli paieška.....	33
5.16 Vertimo išsaugojimas, patikrinimas ir palyginimas	34
5.17 Glosarijus.....	35
5.18 Verčiamų failų formatai.....	35
5.19 Vertimų atminties ir glosarijaus failų formatai.....	36
5.20 Projekto failai ir direktorijos	36
5.20.1 Failų ir direktorių projekto aprašymas	36
5.20.2 Failų ir projekto direktorių buvimo vietos pakeitimas	37
5.21 Klavišų kombinacijos	38
5.22 Skirtingų šriftų ir abėcėlių panaudojimas.....	38
5.23 Darbas su openoffice.org ir staroffice failais.....	38
5.24 Bendri patarimai ir teiginiai.....	40
5.24.1 Microsoft Office failai	40
5.24.2 Tuščias projektas? (Microsoft Office II failai).....	40
5.24.3 Tik vienas projektas!	40
5.24.4 Dingo segmentas... ..	40
5.24.5 Ženklo įstatymas.....	40
6. IŠVADOS.....	42
7. LITERATŪRA	43
8. SANTRAUKA.	45
9. TERMINŲ IR SANTRUMPŲ ŽODYNAS	46
10. PRIEDAI	47

10.1 CD diskas.....	47
10.2 Sistemos vartotojo vadovas	47
10.2.1. Pradedant darbą – projekto pavyzdys.....	47

1. ĮVADAS

Sekundėlei atsitraukus nuo visų subtilybių, egzistuojančių būdų išversti tekstą iš vienos kalbos į kitą galima tvirtai teigti, kad yra tik trys vertimo tipai: žmoniškasis, mašininis ir žmoniškasis – mašininis. Pirmasis tipas, be jokių abejonių, reikalauja daugiausiai pastangų, tačiau iš pirmo žvilgsnio jis atrodo labiausiai patikimas, kas kitas, jei ne žmogus, gali adekvačiai perteikti teksto reikšmę. Tuo labiau, kad gilesnis žvilgsnis į problemą leidžia nustatyti, kad žmogaus vertimas labiausiai vertinamas arba, jei norite, neįkainojamas, tik grožinėje literatūroje ir, iš dalies publicistikoje, kur svarbūs faktoriai yra įvairiapusiškumas ir kūrybingumas. Tuo tarpu moksliniai ir techniniai tekstai reikalauja griežtų formuluočių ir tikslaus terminologijos laikymosi, kas, sutikite, visada žmogui yra problema. Ideali techninio vertimo priemonė galėtų būti kompiuteris, bet artimiausiais metais grynai mašininis vertimas vargu ar ras rimtą praktinį pritaikymą dėl sudėtingumo, įvairiapusiškumo ir nepakankamo bendrinės kalbos „formalizavimo“. Išėitis galėtų būti kombinuotas, žmoniškasis – mašininis vertimas, atliekamas dominuojant žmogui, bet panaudojant kompiuterinės technikos galimybes.

Remiantis anksčiau pasakytu, sutelkiam savo dėmesį ties optimalaus žmoniškojo ir kompiuterinio techninio teksto vertimų procesų kombinavimu.

Šiame darbe apžvelgsime kompiuterinio vertimo pagrindinius principus, analizuosime alternatyvios tekstų vertimo į kitas kalbas technologijas ir pristatysime konceptualų hierarchinį kompiuterinio vertimo modelį.

Pateiksime kompiuterizuoto vertimo modelį, orientuotą į hierarchinę modulinę sistemos architektūrą, kurios atitikmuo artimas intelektualiujų ekspertinių sprendimų priėmimo lentos koncepcijai.

Sukursim programą, kuri naudos savyje „Translation Memory“ technologiją.

Darbas suskirstytas į tokias dalis: *antroje dalyje* apžvelgiami vertimo technologijų tipai bei atliekama jų išsami apžvalga, *trečioje dalyje* aprašomi vertimo atmintis (TM) bei vertimo atminties panaudojimo aspektai. Taip pat apžvelgsime visus TM produktus, pateiksime privalumus ir trūkumus, *ketvirtoje dalyje* sukursime vertimo technologijos konceptualų modelį, *penktoje dalyje* atliekama projektinė realizacija, remiantis analitinėje dalyje pasiūlytais metodais, aprašoma programinės įrangos projektinė dokumentacija. Taip pat apžvelgsime projektuojamos sistemos specifikacija, programuojamos sistemos architektūrą, *šeštoje dalyje* pateikiamos darbo išvados.

Magistro darbą sudaro 10 skyrių, 5 lentelės, 8 grafinės iliustracijos, 1 priedas, 55 puslapiai.

2. VERTIMO TECHNOLOGIJŲ TIPAI

Šiuolaikinėse profesionaliose vertimo terpėse kompiuterinės technikos galimybės naudojamos skirtinguose etapuose ir lygiuose. Iš viso galima išskirti aštuonis kompiuterio panaudojimo būdus vertimo metu (Lentelė 1).

Lentelė 1. Aštuoni kompiuterio panaudojimo būdai vertimo metu

	<i>Terminų lygis</i>	<i>Segmentų lygis</i>
<i>Iki vertimo</i>	<ul style="list-style-type: none">• <u>Terminų išskyrimas</u>• <u>Terminologijos analizė</u>	<ul style="list-style-type: none">• <u>Teksto segmentacija</u>
<i>Vertimo metu</i>	<ul style="list-style-type: none">• <u>Automatinė terminologijos paieška</u>	<ul style="list-style-type: none">• <u>Kalbinių porų paieška vertimų atmintyje</u>• <u>Mašininis vertimas</u>
<i>Po vertimo</i>	<ul style="list-style-type: none">• <u>Terminologijos atitikimo patikrinimas</u>	<ul style="list-style-type: none">• <u>Segmentų, formato ir gramatikos pilnumo patikrinimas</u>

2.1 Termino išskyrimas ir terminologijos analizė

Šiame etape vyksta teksto tyrimas, norint išsiaiškinti, kokie žodžiai ar žodžių junginiai gali būti paimti kaip terminai. Po to, kai nustatytas terminas pirminėje kalboje, vyksta terminologijos analizė, kokį terminą tikslinėje kalboje reikia panaudoti reikalingo koncepto nustatymui. Pavyzdžiui, jeigu pirminėje kalboje pasitaikė žodžių junginys „operacinė sistema“, tai programa turi išanalizuoti jį kaip galimą terminą, netgi jeigu sistemoje jau nustatyti terminai „operacinė“ ir „sistema“.

2.2 Automatinė terminologijos paieška

Šis procesas gali būti palygintas su mašininio vertimu atskirų terminų lygyje. Jo esmė yra ta, kad darbo su tekstu procese vertėjas turi galimybę matyti vertimo variantus kiekvienam terminui ir greitai įterpti reikalingą vertimą į tekstą tiksline kalba, nerizikuojant padaryti klaidą.

2.3 Terminologijos atitikimo patikrinimas

Po to, kai vertimas padarytas, kompiuteris patikrina tai, kad visi kiekvieno terminai būtų išversti vienodai. Pavyzdžiui, jei terminas „operacinė sistema“ per savo pirmąjį įėjimą buvo pakeistas į „operatingsystem“, o per antrąjį į „operationalsystem“, tai turi atsirasti tam tikras perspėjimas apie terminologijos vieningumo pažeidimą.

2.4 Teksto segmentacija

Teksto išskyrimas į segmentus yra svarbus pilno arba dalinio teksto vertimo automatizacijos parengiamasis etapas. Segmentai turi pagal galimybę turėti teksto fragmentus, gramatiškai nepriklausančius vienas nuo kito. Kitaip sakant, turi būti užtikrinta korektiško kiekvieno segmento vertimo galimybė nepriklausomai nuo kitų. Dažniausiai išskyrimas į segmentus vykdomas tam tikra seka.

2.5 Kalbinių porų paieška vertimų atmintyje

Automatinė vertimų atmintis arba tiesiog vertimų atmintis (TranslationMemory), reiškia, visų pirma, anksčiau išverstų tekstų peržiūrą. Ji lygina tuo metu verčiamą tekstą su tuo, kuris saugomas bazėje, „atsimena“ segmentus, kurie nežymiai pasikeitė ir pasiūlo naudoti jų vertimą pakartotinai. Aišku, segmentų panašumo kriterijai gali būti įvairūs, jie atlieka labai svarbų vaidmenį plečiant vertimų atminties galimybes.

2.6 Mašininis vertimas

Šis vertimo būdas remiasi algoritminiu pirminio teksto apdorojimu, kurio metu vyksta segmentų nagrinėjimas, išskiriami atskiri terminai ir santykiai tarp jų, po ko vyksta visų terminų kaita į atitinkamus tikslinės kalbos terminus reikalinga forma ir savitarpio išdėstyme. Mašininis vertimas (MachineTranslatio) gali būti panaudotas tik labai siauru kontekstu ir reikalauja reikšmingo išversto teksto postredagavimo.

2.7 Segmentų, formato ir gramatikos pilnumo patikrinimas

Šitie veiksmai atliekami baigiant vertimą ir turi tikslą patikrinti, ar visi segmentai liko savo vietose, ar išliko formatuojama informacija, ir ar korektiškas gautas tekstas, žiūrint iš tikslinės kalbos gramatinės pusės.

Tarp paminėtų technologijų daugiausia dėmesio sulaukia terminologiniai žodynai ir vertimų atmintis, nes būtent nuo jų efektyvumo priklauso vertimo greitis ir kokybė. Terminologinių žodynų sudarymo technologija pakankamai atidirbta ir pagrįsta principais, analogiškais tiems, kurie naudojami įprastuose dvikalbiuose žodynuose. Teksto išskyrimas į terminus dažniausiai atliekamas „tarpo“ pagalba, su papildomu kaip kurių morfologinių analizių panaudojimu.

Sunkiau su vertimų atminties organizavimu. Kartu su kalbinės poros paieškos trivialine užduotim, įtraukiančią segmentą, identišką užduotam, vertimų atmintis turi užtikrinti segmentų,

panašių į turimą pagal tam tikrą kriterijų, paieškos galimybę. Tokiu atveju, klasikinės vertimų atminties centrinė problema yra analizatoriaus sudarymas tokiu „netiksliu sutapimu“ (fuzzymatches), kurio charakteristikos ir nustato kiekvienos konkrečios profesionalaus vertimo sistemos pranašumus ir trūkumus.

3. VERTIMŲ ATMINTIS (TRANSLATION MEMORY)

Vertimų atmintis naudojama dokumentų iš vienos kalbos į kitą vertimo metu. Ji padeda vertėjui, išsaugodama teksto nuotrupas duomenų bazėje ("vertimų atmintis") taip, kad jei tokia pati nuotrupa arba jos dalis vėl pasitaikys, galima bus iš karto pamatyti, kaip ji buvo išversta anksčiau.

3.1 Vertimo atminties panaudojimo aspektai

Remiantis aukščiau pateikta medžiaga, bet kokios vertimo atmintis sistemos funkcionavimo pagrindas yra anksčiau išversti tekstai. Daugelis šių tekstų pastoviai papildomi naujais vertimais, kuo remiantis, automatiškai išverstų segmentų procentas pastoviai auga. Tai reiškia, kad efektyviam vertimo atminties panaudojimui, visi tekstai turi turėti pakankamai panašių frazių. Toks daiktų išdėstymas turi vietą skirtingų rūšių produktų dokumentacijoje. Tai pagrįsta dviem faktoriais. Pirma, dokumentaciją priimta sudaryti maksimaliai paprasta kalba, lakoniškai ir griežtais terminais. Antra, pasirodant naujoms tiekiamo vartotojams produkto versijoms ir modifikacijoms dokumentacijos turinys keičiasi nežymiai. Vertimų atmintis panašiose situacijose išvaduoja vertėją nuo būtinybės kelis kartus versti identiškus teksto fragmentus, esančius skirtinguose dokumentuose.

Tuo pat metu, vertimų atminties panaudojimas reikalauja iš vertėjo specialaus pasirengimo, o taip pat atitinkamo aparatinio ir programinio aprūpinimo. Kitas neigiamas faktorius yra tas, kad užtikrinti laukiamą efektą visi vertimai turi būti padaryti vienoje terpėje arba terpėse, atitinkančiose pagal formatą pateiktus duomenis. Naudingas vertimų atminties efektas atsiranda su ženkliu atidėjimu laike, reikalaujantis pradžioje papildomų investicijų.

Reziumuojant aukščiau pateiktą, galima išskirti tris tiriamos technologijos panaudojimo sąlygas:

1. Didelė vertimo apimtis.
2. Verčiamų tekstų panašumas.
3. Pasirengimas atidėtų investicijų grąžinimui.

3.2 Translation memory sistemos

Vertimų atmintis yra duomenų bazės, saugančios kalbų poras ir atitinkamą paieškos mechanizmą. Nežiūrint į tai, kad skirtingos profesionalios vertimų terpės, tokios kaip "Translator's Workbench" firmos Trados, "Transit" firmos Star, "DejaVu" firmos Atril ir kitos turi, kaip matoma, skirtingą šio mechanizmo realizaciją.

Produktai rinkoje. Šiuo metu rinkoje, naudojantis vertimų atmintį, yra žinomi tokie produktai.

- Déjà Vu firmos Atril
- EPTAS TM firmos ALPNET Technology
- SDLX firmos SDL
- Transit firmos STAR
- Translation Manager firmos IBM
- Translator's Workbench firmos Trados
- ZeresTrans firmos Zeres
- TRANS Suite 2000 firmos Cypresoft

Aukščiau išvardintos profesionalios vertimų terpės, visos jos turi skirtingą mechanizmo realizaciją, bet idėja yra bendra.

Toliau mes pabandyšim duoti trumpa sistemų aprašymą, tokių kaip „Translator's Workbench“ firmos Trados, „Transit“ firmos Star, „DejaVu“ firmos Atril, „SDLX“ firmos SDL ir „Translation Manager“ firmos IBM.

3.2.1 Trados translator's workbench

Šitoje programoje kaip teksto apdorojimo programa naudojamas MS Word. Prie to reikia pasakyti kad Microsoft įsigijo akcijos kompanijos „Trados“ ir ruošiasi naudoti ją kaip vidinę lokalizuotų komponentų saugyklą. Todėl programa Translation's Workbench turi gerus šansus tapti tarptautiniu standartu.

3.2.2 Transit firmos Star

Translation Memory firmos Star susideda ne tik iš atminties saugyklų, bet ir iš taip vadinamo asociatyvaus tinklo. Greitas duomenų apdorojimas net ir kai duomenų saugykla yra labai didelė.

3.2.3 SDLX firmos SDL

Tai produktas, kuris pasirodė neseniai ir greitai vystosi. SDL – viena iš didžiausių lokalizavimo bendrovių Didžiojoje Britanijoje. SDLX –Amprano tolimesnis vystymas. Susideda iš daugelio sujungtų programų.

3.2.4 DÉJÀ VU INTERACTIVE firmos Atril

Lyginant su kitais, tai nebrangus ir turintis daugybę (daugiausiai) funkcijų produktas. Vertėjai, kurie jį naudoja, paduoda paraiškas, kad būtų įdėta kai kurios papildomos naudingos funkcijos, ir tai operatyviai padaroma.

3.2.5 IBM Translationmanager

Tai visų TM sistemų „Motina“. Prarado savo lyderio pozicijas, dėl:

- Lėtas vystymas
- Silpnas marketingas

Kaip sako specialistai – tai efektyviausia programa. Bet paprasta, turinti prastą interfeisą (nepatogu su ją dirbti).

Tai vienintelė programa, palaikanti daugybę tekstinių formatų:

- MS Word
- WordPerfect
- Ami Pro
- HTML
- RTF
- Ventura Publisher
- Interleaf
- FrameBuilder
- BookMaster
- IPF (Information Presentation Facility)
- ASCII

Tūri labai patogi funkcija: kokioj aplinkoj yra pirminė kalba, tokioje saugomas ir išverstas tekstas.

Visi TM produktai turi trūkumus:

1. Verčiamų tekstų ribojimas (pvz. galima versti techninius tekstus, bet negalima versti grožinės literatūros).
2. Didelė TM produktų kaina (virš 1000 EUR).
3. Nėra bendro standarto. Pavyzdžiui vertėjas įsigijo vieną ar kitą produktą ir sumokėjo didelę pinigų sumą, bet tai nereiškia, kad jis galės įsidarbinti, todėl, kad skirtingi vertimo biurai naudoja skirtingus produktus.
4. Visi TM produktai reikalauja didelių monitorių, mažiausiai 17 colių diametro.

4. KONCEPTUALUS MODELIS IR JO TAIKYMAS

4.1 Pagrindiniai darbo principai

Tegul pirminiam tekste yra šios frazės:

- *"Temperatūra reguliuojasi rankenos pasukimu."*
- *"Temperatūra reguliuojasi rankenos pasukimu pagal laikrodžio rodyklę."*
- *„Vandens srovė reguliuojasi rankenos pasukimu pagal laikrodžio rodyklę.“*

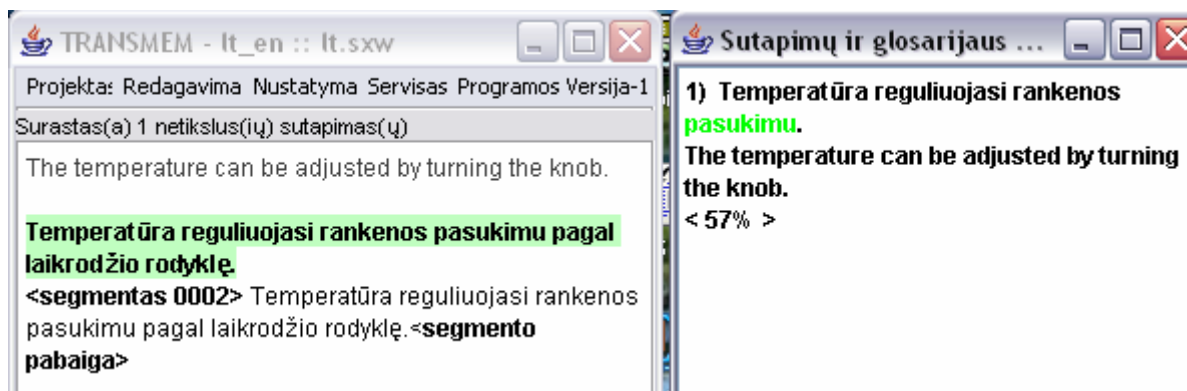
Jeigu segmentacija vykdoma sakiniiais, tai kiekviena iš pateiktų frazių patenka į skirtingą segmentą. Tegul pirmas segmentas buvo išverstas žmogaus, šiuo atveju:

"The temperature can be adjusted by turning the knob."

Kalbinė pora, susidedanti iš pirminio ir išversto segmentų, įvedama į vertimų atmintį. Kada vertėjas prieina iki antros pavyzdžio frazės, sistema nustato panašumą ir išveda į ekraną šią informaciją: lentelė 2.

Lentelė 2

Dabartinis segmentas	<i>Temperatūra reguliuojasi rankenos pasukimu</i> <small><i>pagal laikrodžio rodyklę</i></small>
Surastas segmentas	<i>Temperatūra reguliuojasi rankenos pasukimu</i>
Vertimas	<i>The temperature can be adjusted by turning the knob</i>
Panašumo laipsnis	~57%



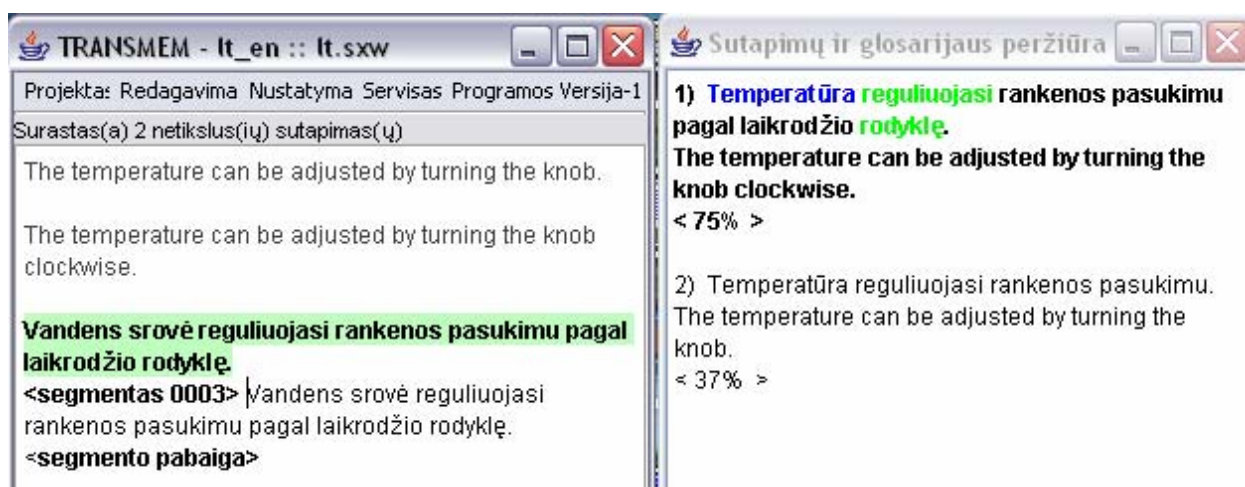
Dabar vertėjas turi galimybę iš dalies pasinaudoti jau padarytu vertimu, turėdamas omeny skirtumus:

"The temperature can be adjusted by turning the knob clockwise."

Po to, kai segmentas, atitinkantis antrą pavyzdžio frazę pažymimas kaip išverstas, vertimų atmintyje atsiranda dar viena kalbų pora. Tuo pačiu, kai prieiname prie trečios frazės sistema jau turi galimybę parodyti vertėjui du panašius variantus: lentelė 3.

Lentelė 3

Dabartinis segmentas	<i>Vandens srovė reguliuojasi rankenos pasukimu pagal laikrodžio rodyklę</i>
Rasta kalbinė pora 1	<i>Temperatūra reguliuojasi rankenos pasukimu pagal laikrodžio rodyklę</i>
	<i>The temperature can be adjusted by turning the knob clockwise</i>
Panašumo laipsnis	~75%
Dabartinis segmentas	<i>Vandens srovė reguliuojasi rankenos pasukimu pagal laikrodžio rodyklę</i>
Rasta kalbinė pora 2	<i>Temperatūra reguliuojasi rankenos pasukimu</i>
	<i>The temperature can be adjusted by turning the knob</i>
Panašumo laipsnis	~37%



Pasinaudojus, pavyzdžiui, pirmu iš pateiktų variantų vertėjas greitai susidoroja su likusia frazės dalimi:

"The water head can be adjusted by turning the knob clockwise."

Vertimų atminties darbo efektyvumas dažniausiai priklauso nuo to, kiek sėkmingai išspręstos šios užduotys:

1. Segmentacija;
2. Specialių simbolių ir formatuojančios informacijos apdorojimas.

Akivaizdu, kad augant segmentų dydžiui, mažės naudingų sutapimų kiekis (ir išaugs dalinių kiekis), kas stipriai padidins paieškos procedūrų resursų tūrį ir pareikalaus iš vertėjo didelių pastangų studijuojant pateikiamos jam kaip vertimo variantai kalbinės poros. Iš kitos pusės, segmentų apimties mažėjimas padarys juos mažiau naudingus pakartotinam naudojimui, nes stipriai išaugs konteksto įtaką vertimui. Optimaliu segmentacijos vienetu dažniausiai yra sakinio fragmentas, išskirtas skyrybos ženklais. Norėdami išvengti klaidingos segmentacijos pagal taškus abreviatūrų viduje ir kitais panašiais atvejais naudoja reguliarius išsireiškimus ir išimčių sąrašus.

Antra problema pagrįsta tuo, kad tekste be raidžių dažnai kiti simboliai, kaip: įdiegtų į dokumentą objektų markeriai, skirtukai, kryžminės nuorodos, šrifto savybių jungikliai. Visi šie skirtingi elementai kartais gali paveikti vertimą. Pavyzdžiui, išskirtas kursyvus žodis verčiant gali būti paimtas į skliaustus ir papulti į galutinį tekstą nepakeistas. Analizatoriaus elgesio valdymui tokiose situacijose daugelyje programinių produktų yra specialūs nustatymai, taip pat atsiradusios panaudojant reguliarius posakius.

4.2 Galimybių praplėtimo keliai

Kadangi vertimų atminties funkcijos yra išverstų užduotų segmentų fragmentų paieška duomenų bazėje, tai jos galimybių riba yra, akivaizdu, atranka, maksimaliai apimanti pirminį segmentą ir neturintį jokios papildomos informacijos. Pabandykim išskirti vertimų atminties kokybės padidėjimo galimus variantus, remiantis anksčiau duotu pavyzdžiu. Išrinksim ir peržiūrėsime dvi kalbines poras: lentelė 4.

Lentelė 4

Kalbinė pora 1	<i>Temperatūra reguliuojasi rankenos pasukimu pagal laikrodžio rodyklę</i>
	<i>The temperature can be adjusted by turning the knob clockwise</i>
Kalbinė pora 2	<i>Vandens srovė reguliuojasi rankenos pasukimu pagal laikrodžio rodyklę</i>
	<i>The water head can be adjusted by turning the knob clockwise</i>

Segmentų panašumas pirminėje kalboje leidžia daryti prielaidą, kad jų vertimai, tai yra segmentai tiksline kalba taip pat turi būti panašūs. Jeigu tai tiesa, iškyla noras išskirti iš dviejų pateiktų kalbinių porų bendrą dalį ir pateikti ją naujos kalbinės poros pavidalu. Įvykdžius nesudėtiną eilučių susikirtimo operaciją, gauname šį rezultatą: lentelė 5.

Lentelė 5

Kalbinė pora 3	<i>Reguliuojasi rankenos pasukimu pagal laikrodžio rodyklę</i>
	<i>can be adjusted by turning the knob clockwise</i>

Dabar kiekvienam segmentui, turinčiam fragmentą „*reguliuojasi rankenos pasukimu pagal laikrodžio rodyklę*“, gali būti išrinkta kalbinė pora nr. 3, turinti tik būtiną fragmentui vertimą.

Vis dėlto, ne visada viskas taip gerai. Šiek tiek įdėmesnis žvilgsnis į šį pavyzdį iš karto priverčia mus pripažinti, kad tokių „sutrumpintų“ kalbinių porų sudarymas ekvivalentiškas segmento dydžio mažėjimui, o tai nėra gerai. Mažas teksto fragmentas, ypatingai, jeigu jis neapribotas jokiais skyrybos ženklais, vos gali būti teisingai išverstas be konteksto. Tai yra,

išskiriant bendras dalis dviejuose jau naudojamose kalbinėse porose būtina vadovautis tokiais pačiais principais, kaip ir pradinėje pirminio teksto segmentacijoje.

Taip pat, nereikia pamiršti, kad segmentų susikirtimas pirminėje kalboje nebūtinai izomorfiškas segmentų susikirtimui tikslinėje kalboje. Tai susiję su gramatikos taisyklių skirtumais skirtingose kalbose, žodžių eiliškumu, žodžių reikšmių atitikimu. Todėl prasminga tikslinio segmento kalbinės poros reikšmė, atsirado susikirtimų, galima būti tik tais atvejais:

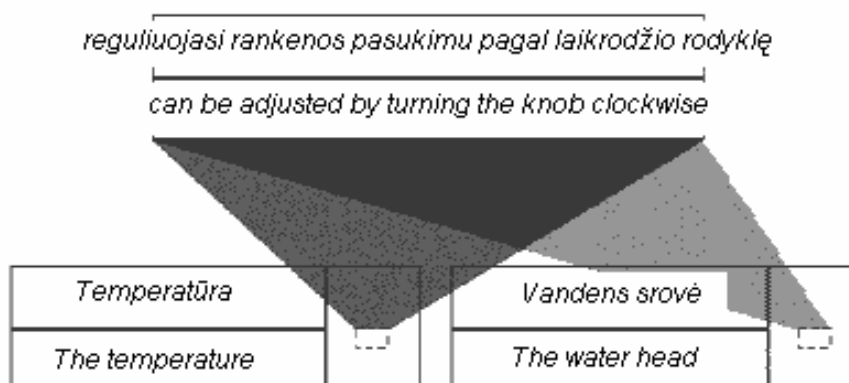
1. Reikšmingas abiejų segmentų dydis iš naujo susikūrusios kalbinės poros;
2. Evristiškai nustatytame segmentų susikirtimo izomorfizme pirminėje ir tikslinėje kalbose (pavyzdžiui, jei susikirtimas padarytas skyrybos ženklų pagalba);
3. Morfologinėje ir sintaksinėje rezultatų susikirtimo analizėje su mašininio vertimo technologijos panaudojimu.

Dar viena svarbi užduotis realizuojant aprašytą manipuliacijų mechanizmą su kalbinėmis poromis yra nedidelio formalizmo sudarymas, leidžiančio vienareikšmiškai nustatyti, kokios būtent poros turi būti apdorojamos, kaip būtent turi formuotis rezultatas, kaip turi vykti segmento paieška ir kokie segmentų palyginimo kriterijai vykdant paiešką.

4.3 Daugialygis vertimų atminties modelis

4.3.1 Duomenų pristatymas

Šiuo metu realiai naudojamų vertimų atminties realizacijų struktūra yra vienalygė ir turi savyje kalbinių porų einančio pagal eilę sąrašo. Įvedus bendrų dalių porose išskyrimo mechanizmą panašus duomenų organizavimas bus nepatogus. Iš tikrųjų, bet kokioms dviems susikertančioms kalbinėms poroms bazėje bus sudaromas papildomas elementas, kurio turinys pilnai dubliuos abiejose kalbinėse porose. Pertekliaus pavyks atsikratyti, jei pavyks ištrinti parodytą bendrą dalį iš abiejų porų, o į jos vietą pastatyti nuorodą į naujai sukurtą kalbinę porą (Pav. 1).

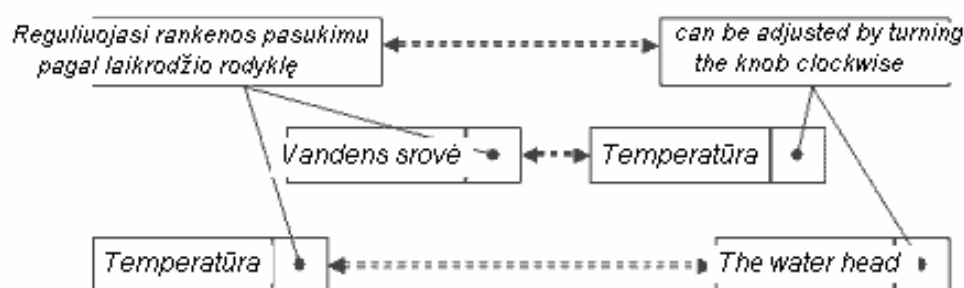


Pav. 1

Keičiant šią procedūrą kiekvieną kartą, kai aptinkamas eilinis susikirtimas, mes, galų gale, gausim nukreiptą grafą, kurios mazgai yra kalbinės poros, o lankai – įjungimo santykiai.

Dar viena optimizacija yra pirminio ir tikslinio segmentų kiekvienos kalbinės poros išskyrimas. Tai turi prasmę todėl, kad ne taip jau retai būna tokių atvejų, kai ta pati pirminė frazė verčiama į tikslinę kalbą skirtingai, kas pagimdo dvi kalbines poras su vienodu pirmu elementu. Be to, pirminių ir tikslinių segmentų susikirtimai ne visada izomorfiniai, ir jų rezultatai nebūtinai sudaro kalbinę porą.

Todėl išskirsim kalbines poras, ir gausim dvi orientuotas grafus, „synchronizuotus“ tarpusavyje (Pav. 2). Tuo, santykiai, jungiantys įvairių grafų viršūnes, gali būti skirtingi, bet būtinai turi simetrines savybes.



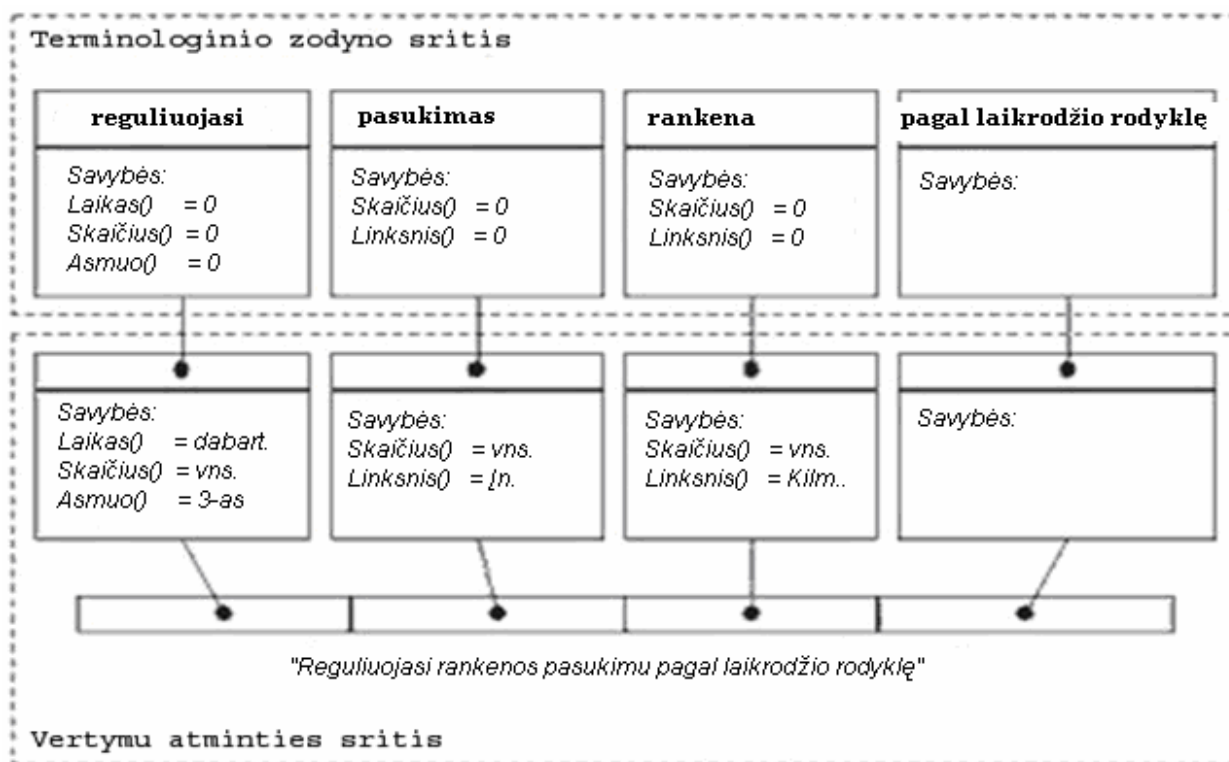
Pav. 2

Toliau vystant mintį, atsiminsime, kad mes jau turim aparatą, leidžiantį formalizuoti pateiktą schemą. Tai – objektiškai orientuotas priėjimas. Iš tikrųjų, kiekvienam segmentui bazėje gali būti sulyginata atskira klasė, įjungimo santykis pagal savo savybes identišką siekimo santykiui, o horizontalūs ryšiai, formuojantys kalbines poras gali būti užduoti asociacijos mechanizmo, arba įvedimo į klasę Translate („išversti“) metodu.

Tuo tarpu, pasiūlytas modelis kol kas neaprašo vidinės segmentų struktūros, kurią būtina analizuoti sudarant naują kalbinę porą egzistuojančių susikirtimo pagrindų. Norint išspręsti šią užduotį, išskaidysim visus segmentus į atskirus žodžius (su tarpais). Dabar kiekvienas segmentas gali atstovauti klasę, kuri yra visų žodžių išvestinė, kuri sudaro segmento tekstą. Todėl visiškai nebūtina turėti kiekvieno žodžio vertimo, nes tai atsispindės tik atitinkamų ryšių nebuvimu tarp grafų mazgų, o Translate metodas gražins „nulinę“ reikšmę.

Prisimenant tai, kad į vertimo terpės sudėtį be vertimų atminties įeina ir terminologinis žodynas, segmentų dalybą galima vykdyti ne pagal žodžius, o pagal terminus. Terminologinio žodyno buvimas duoda dar vieną galimybę. Kiekvienam termino į segmentą įvedimui galima nustatyti jo pradinę formą, tai yra tą, kurioje jis įeina į žodyno bazę. Ši pradinė forma tarnaus kaip abstrakti bazinė klasė kiekvienam termino įėjimui, o konkretus įėjimas turės reikšmių apibūdinimo užduotus bazinėje klasėje atributus: giminę, skaičius, linksnis ir t.t.

Paskutinė naujiena priartina mus prie minties apie tai, kad terminologinį žodyną galima suvienyti vertimų atminties pagalba, įsivaizduokim, tuo pačiu, visus vertėjo resursus universalaus modelio pavidalu (Pav. 3).



Pav. 3

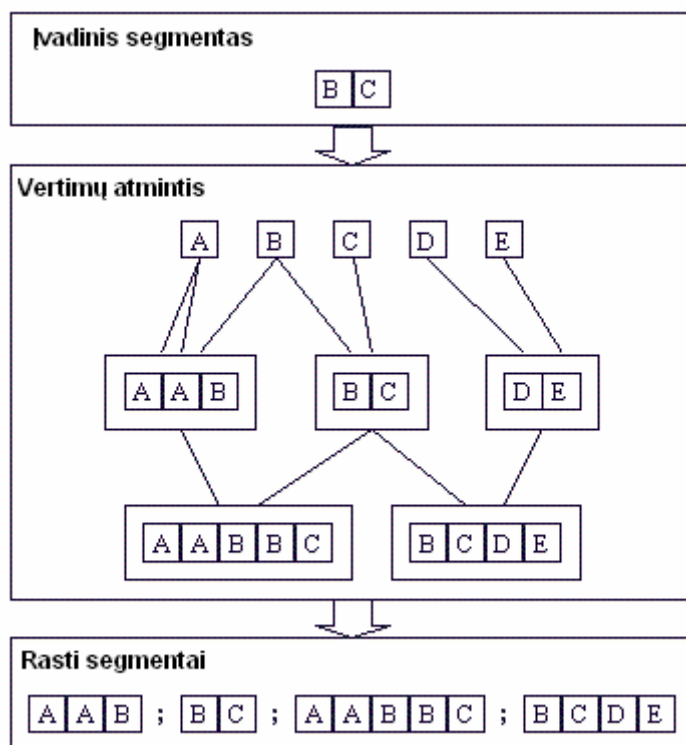
4.4 Paieška ir pridėjimas

Iki to, kol vertimų atmintis buvo linijinė, segmentai nedalomi, o sulyginimas griežtas, paieškos užduočių sprendimas ėjo prie griežtos leksikografinės tvarkos virš daugelio pirminės kalbos segmentų santykių įvedimų. Kitaip sakant, buvo nustatomas operatorius „mažiau“, kurio pagrindu buvo galima įgyvendinti paprastą dvejetainę paiešką ir lygybės patikrinimą. Su operatoriaus „netikslaus sutapimo“ įvedimu, kuris leido įvertinti bet kokių dviejų segmentų sutapimo laipsnį, problemos paieškos sprendimas staigiai susikomplicavo ir, be papildomų gudrybių su skirtingo lygio indeksacijomis, tapo ekvivalentiškas visiško perrinkimo reikšmei. Pasiūlytas daugialypis vertimų atminties modelis, tiesą sakant, parodo kažkokį nelygios indeksacijos mechanizmą: kiekvienas įeinantis į segmentą žodis, pagal esmę, identifikuoja vertimų atminties orientuoto grafo kažkokią dalį, susidedančią iš mazgų, kuriuos galima pasiekti pradėjus apėjimą nuo mazgo, atitinkančio išrinktą žodį.

Vartojant vertimų atminties išrinktos struktūros ypatybę, segmentų paieškos užduotį, panašius į užduotą, galima išspręsti šių veiksmų sudarymo keliu (Pav. 4):

1. Išskirti užduotą segmentą į žodžius;

2. Surasti vertimo atmintyje visus mazgus, atitinkančius šiuos žodžius;
3. Leidžiantis grafu paveldėjimo santykiais, įdėti į surastų segmentų sąrašą visus sutinkamus mazgus.



Pav. 4

Priežastinis klausimas yra tas, kokia tvarka reikia vaizduoti surastus segmentus vertėjui: gi pateikta paieškos procedūra išrinks iš atminties visus segmentus, susikertančius su duotu, nors bent pagal vieną žodį. Kokios rastų segmentų filtracijos ir rūšiavimo taisyklės?

Atsakymas į šį klausimą yra už išrinkto formalizmo ribų, vis dėlto tai nėra baisu. Reikalas tame, kad paieškos rezultatas yra klasikinė vienalygė vertimų atmintis, kurios analizė gali būti atlikta metodais, formalizuotais egzistuojančių vertimų terpių ribose. Paieškos efektyvumo aprūpinimui tikslinga vykdyti segmentų „tinkamumo“ vertinimą jų buvimo lygyje. Pavyzdžiui, jei kai kuris segmentas visiškai sutampa su etalonu, tai visi jo palikuonys grafe gali būti automatiškai išmesti iš paieškos.

Akivaizdi procedūros korektiškumo pridėjimo sąlyga yra sėkmingos paieškos aprūpinimas. Išeina, kad pridedamas segmentas turi turėti tarp savo protėvių (nebūtinai tiesioginių) visus sudarančius jį žodžius. Sekdami optimizacijos tikslą, galima padaryti išvadą, kad tarp daiktų turi būti tokie grafo mazgai, turintys duoto segmento fragmentus. Kitaip sakant, jei vertimų atmintyje yra segmentai „AB“ ir „CD“, tai segmentas „ABCD“ turi tapti šių dviejų segmentų palikuoniu. Analogiška, jei atmintyje yra segmentas „ABCD“, tai pridedamas segmentas „AB“ turi tapti jo protėviu. Bendru atveju segmento pridėjimo metu į vertimų atminties grafą gali egzistuoti alternatyviniai paveldėjimo variantai. Tokioje situacijoje pridėjimo

schema ryškiai pasunkės. Bet kokių atveju, optimalių klasių hierarchijos sudarymo problema sprendžiama objektiškai orientuoto priėjimo rėmuose.

4.5 Kalbinių porų susikirtimo skaičiavimai

Kadangi dviejų segmentų bendros dalies išskyrimas – svarbus pasiūlytos vertimo technologijos etapas, išnagrinėsime šį klausimą detaliau. Tuo pačiu, prisimenant tai, kad segmentas vertimo atmintyje veikia, dažniausiai, ne atskirai, o kaip kalbinės poros elementas, peržiūrėsime būtent porų perskaičiavimą.

Pradžiai apibūdinsime segmentų persikirtimą. Ir taip, segmentų A ir B persikirtimas – tai segmentų, C_i , dauguma, tokių kaip:

1. Kiekvienas iš C_i yra ir A, ir B;
2. Jokie du C_i neturi vienodų fragmentų;
3. Neegzistuoja toks segmentas D, kad ir A, ir B turi D, ir D turi vieną iš segmentų C_i .

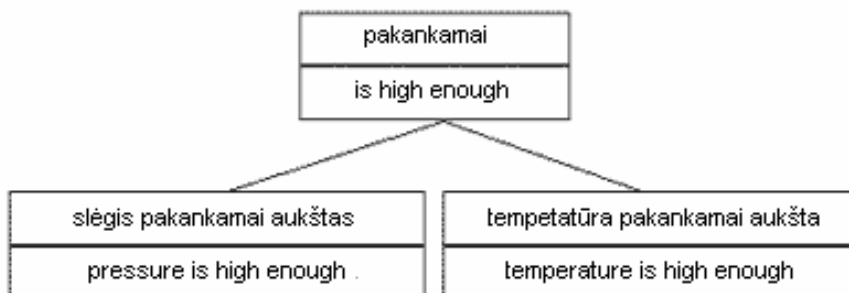
Pateiktas apibrėžimas nereiškia išskyrimo iš segmento A ir B visų bendrų fragmentų. Tai padaryta tam, kad būtų galima panaudoti įvairaus sudėtingumo persikirtimo realizacijos algoritmus.

Pereisime prie kalbinių porų išvardinimo. Kaip jau minėta aukščiau, labai svarbu nustatyti, ar persikirtimas yra izomorfinis, kitaip sakant ar galima pirminio ir tikslinio segmento persikirtimo rezultatus laikyti kalbine pora. Du pavyzdžiai tai parodo (Pav. 5). Pirmu atveju segmento pora „*pakankamai aukštas*“ ir „*ishighenough*“ turi prasmę įdėti į vertimų atmintį, kuris gali pakartotinai panaudotas vertėjo. Antru atveju – tai pakankamai akivaizdu – segmentui „*pakankamai*“ nereikia sutapatinti segmentą „*ishighenough*“, todėl kad duota kalbinė pora bus nekorektiška.

4) **Susikirtimas gali būti kalbinė pora**



6) **Susikirtimas nėra kalbinė pora**

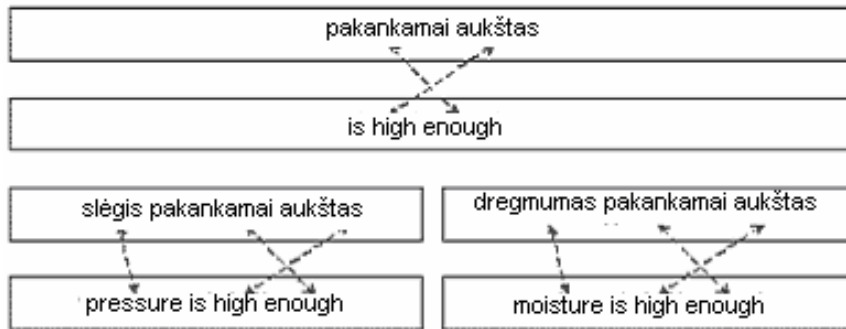


Pav. 5

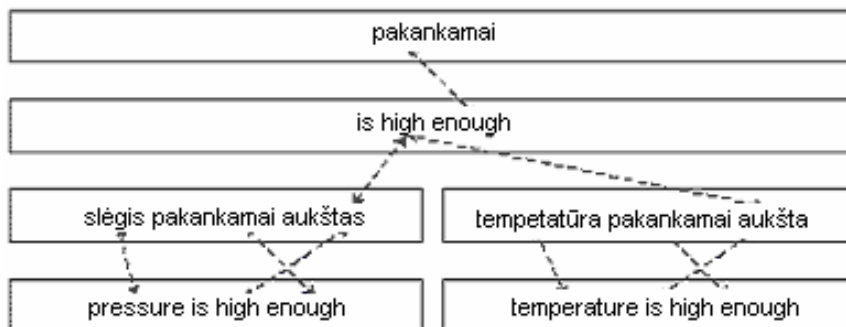
Izomorfizmo perkirtimų patikrinimui galima panaudoti priėjimą, pagrįstą mašininio vertimo technologija. Jo esmė terminų sutapatinime, sudarančių pirminius ir tikslinius segmentus. Tam būtina padaryti gramatinį segmentų nagrinėjimą terminų išskyrimo ir sintaksinių ryšių tarp jų tikslu. Po to galima pasinaudoti terminologiniu žodynu tam, kad apibrėžti kokį terminą tiksliniam segmente atitinka užduotas terminas pirminiam segmente. Kitaip tariant izomorfizmą galima apibrėžti pagal sekantį kriterijų (Pav. 6).

- Susikirtimas yra izomorfinis, jeigu visiems pirminio segmento terminam prilyginami terminai jo tikslinio segmento ir sintaksiniai ryšiai tarp jų identiški tiems, kurie yra segmentuose iš kurių buvo gautas susikirtimas.

3) **Susikirtimas gali būti kalbinė pora**



4) **Susikirtimas nėra kalbinė pora**



Pav. 6

Bendrai tariant, izomorfizmo įvertinimui galima tikrinti ne tik atskirus terminus (esminiai šakniniai vertimo atminties grafo mazgai), bet ir pirminiai visų lygių segmentai. Tai padidins įvertinimo patikimumą, sumažinus neteisingo sintaksinių ryšių apibrėžimo riziką.

Reikia atkreipti dėmesį į tą faktą, kad siūlomajame modelyje mašininis vertimas naudojamas tik gramatinei teksto analizei, sudarančiai segmentą. Mašininio vertimo sistemų silpnoji vieta yra segmento terminų vertimo pasirinkimas ir būtent šita užduotis sprendžiama labiau patikimu būdu – vertimų atminties pagalba.

4.6 Nuo kalbinių porų prie kalbinių žvaigždžių

Dažna situacija, kai vertimą tenka atlikti ne tik iš kalbos A į kalbą B, bet ir atvirkščiai, iš kalbos B į kalbą A. Viena ir ta pati vertimų atmintis bus vienodai naudinga abiem atvejais, kadangi joje yra maksimaliai sinchronizuoti kalbos A ir kalbos B segmentų grafai. Tačiau mums verta pasunkinti užduotį ir pasiūlyti vertimo būtinybę tarp kelių kalbų, kai vieningos vertimo atminties naudingumas žymiai krenta. Iš tikrųjų, jei vertimas atliekamas iš kalbos A į kalbą B ir C tai atmintyje nebus išsaugomi atitikmenys tarp kalbos B ir C segmentų. Kaip užtikrinti panašią galimybę?

Protingu sprendimu galėtų būti kažkokios tarpinės kalbos X naudojimas, į kurią būtų atliekamas vertimas, o paskui, antru etapu, būtų atliekamas vertimas iš kalbos X į tikslinę kalbą. Panašiu atveju visos kalbinės poros vertimų atmintyje susidėtų iš kalbos X segmento ir vieno iš tikslinės (arba pirminės) kalbų segmento. Tačiau čia yra kliūčių. Visų pirma, kaip mes jau įsitikinome, kalbinių porų susikirtimas ne visada būna izomorfiniu, atitinkamai ne visos kalbinės poros vertimų atmintyje turės vertimą į kalbą X. Aišku, tokios poros bus bevertės. Visų antra, verčiant visada yra prasmės iškreipimo pavojus: dvigubas vertimas žymiai padidina šį pavojų.

Kokia gi turėtų būti šita hipotetinė tarpinė kalba X, kad ja būtų galima atitinkamai naudotis? Jos savybės išplaukia iš dviejų įvardintų problemų. Visų pirma, šita kalba turi užtikrinti izomorfinį susikirtimą su bet kokia kita kalba. Izomorfizmo pažeidimų (kraštutiniu atveju, giminingose kalbose) priežastis atitinkamam lygy yra sintaksinių taisyklių skirtumas, privedantis prie skirtingos sakinio narių tvarkos, o taip pat prie vieno ir to paties žodžio formų skirtumų. Iš to seka, kad kalba X turi būti invariantinė žodžių tvarkai ir kaip nors atkreipti dėmesį į formas tikslinėje kalboje. Antra, ji turi galėti perduoti frazės esmę bet kuria kalba, atitinkamai turėti specifines visų egzistuojančių kalbų reikšmes.

Jeigu bus atrasta tokia universali kalba, tai bus galima organizuoti vertimų atmintį kalbinių porų pagrindu, o kalbinių žvaigždžių pagrindu, kur centre yra kalbos X segmentas, spinduliuose – jo vertimo į kitas kalbas variantai. Didelės apimties vertimo iš ir į daugelį kalbų atveju papildomos dvigubo vertėjo darbo pastangos pilnai atsipirks lanksčiu vertimų atminties mechanizmu, žymiai supaprastinančiu daugiakalbį vertimą.

Liko tik surasti kalbą X. Ir tokia kalba egzistuoja! Tai universali tinklo kalba UNL (UniversalNetworkingLanguage), pasiūlytas Jungtinių Tautų Universiteto (UnitedNationsUniversity- UNU) Studijų Vystymo Instituto (InstituteofAdvancedStudies- IAS). Juo mes ir pasinaudosime tolimesniame vertimų atminties modulio vystyme.

4.7 Trumpas kalbos UNL aprašymas

UNL kalba reiškia pasisakymus taip vadinamų universalių žodžių daugumos pavidalu, susietų atitinkamo tipo binariniais ryšiais. Universalus žodis reiškia, kažkokios reikšmės esmę ir priskiriamas atitinkamos reikšmės vardas (dažniausiai anglų kalba), grupė pagalbinių atributų (skaičius, laikas, nuosaka ir t.t.) ir kai kuriais semantikos apribojimais pateikiamais kitų universalių žodžių ir santykių pagalba. Štai universalių žodžių pavyzdžiai:

"žmogus" - man(icl>person)

"žmonės" - man(icl>person).@plural

"kepurė" - hat(icl>thing)

Binariniai ryšiai sukuria tarpusavio ryšius tarp reikšmių tipą. Pavyzdžiui, žodžių junginyje „žmogus eina“ naudojamas santykis „agt“ (agent) reiškiantis ryšį tarp veiksmo subjekto ir pačio veiksmo. Žodžių junginyje „nešti vėliavą“ naudojamas santykis „obj“ (objekt), reiškiantis veiksmo nukreipimą į objektą. UNL sintaksėje šitie pavyzdžiai rašomi taip:

```
"žmoguseina" - agt(walk(icl>do), man(icl>person))
"neštivėliavą" - obj(carry(icl>do), flag(icl>thing))
```

Bet kuri tokių porų dauguma gali būti apjungta į vieną sudedamąjį universalų žodį specialių atžymų pagalba. Pavyzdžiui, žodžių junginys „žmogus, nešantis vėliavą“ pateikiamas sekančiais:

```
agt:01(carry(icl>do):02, man(icl>person))
obj:01(carry(icl>do):02, flag(icl>thing))
```

kad parodyti tą faktą, kad keletas vieno ir to pačio universalus žodžio įvedimas reiškia vieną objektą, visi įvedimai žymimi viena ir ta pačia žyme, kaip tai padaryta žodžio „carry“ atveju.

Sudėtinis žodis, taip pat kaip ir paprastas universalus žodis gali būti binarinio ryšio elementu. Frazė „aš matau žmogų, nešantį vėliavą“ užrašomas taip:

```
agt:01(carry(icl>do):02, man(icl>person))
obj:01(carry(icl>do):02, flag(icl>thing))
agt(see(icl>do):03, I)
obj(see(icl>do):03, :01)
```

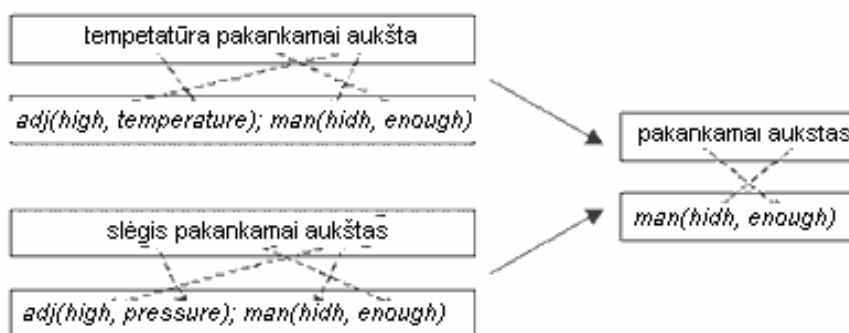
Kaip matoma iš pavyzdžių, kiekvienas žodis, paprastas ir sudėtinis kalboje reiškia apibrėžtą reikšmę ar „konceptą“. Atitinkamai, UNL operuojame ne žodžiais, o būtent konceptais. Iš kitos pusės sakinyje UNL kalba reiškia nepaeiliui sudėtą konceptų susietų binariniais ryšiais daugumą. Iš to seka, kad verčiant į UNL išmestas izomorfizmo pažeidimas, kurį sukelia skirtinga žodžių tvarka.

4.8 Kalbinių porų persipynimo izomorfizmo problema

Taigi, mes nustatėme, kad vaizdavimas UNL kalboje leidžia visiškai išsaugoti esmę (kadangi leksikinėmis dalelėmis yra vienareikšmės reikšmių reikšmės) ir garantuoja kalbinių

porų izomorfizmo persipynimo nepriklausomybę nuo žodžių tvarkos sakiniuose. Tačiau, liko ir padidėjo izomorfizmo pažeidimo problemos, išskuktos vieno ir to pačio žodžio formų skirtinguose sakiniuose skirtumo. Iš tikrųjų, konceptas UNL, nekeičia savo formų, kad ir su bet koku konceptu jis būtų susijęs. Tuo pačiu vienas ir tas pats žodis bendrinėje žmogaus kalboje gali keistis.

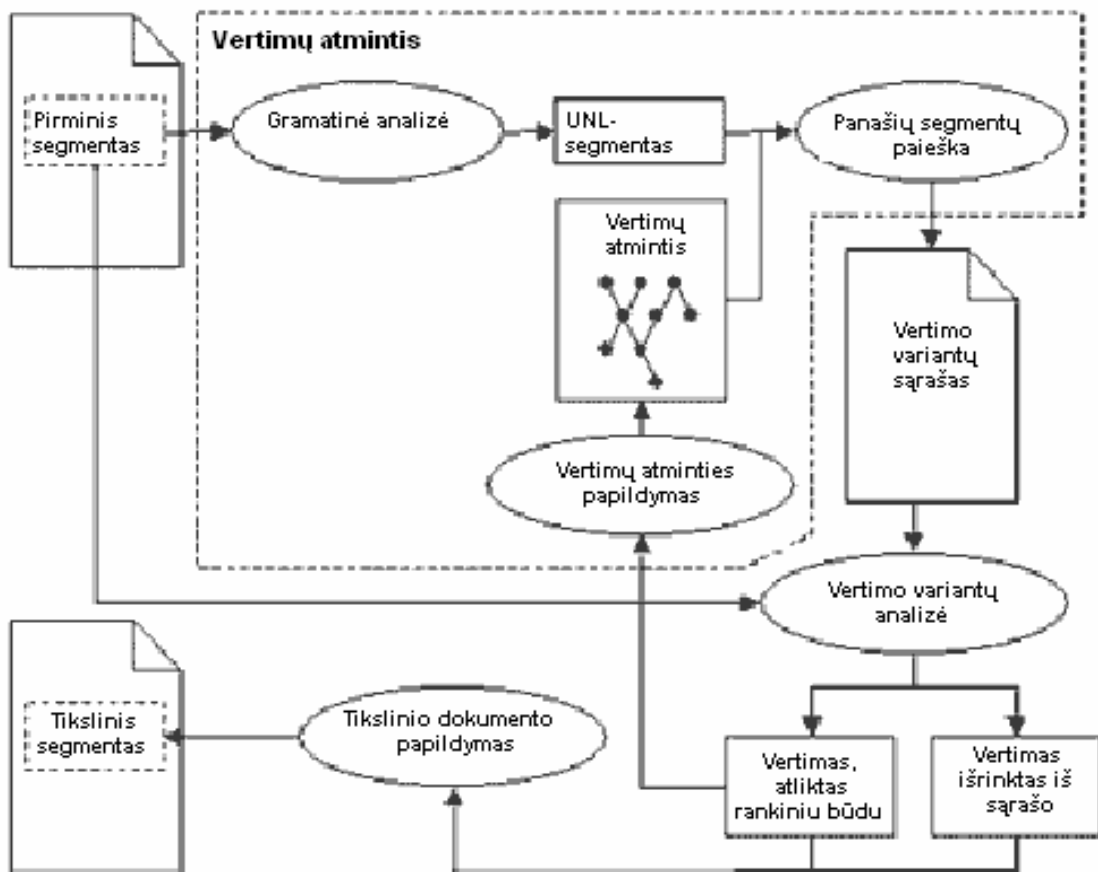
Analizuojant šią problemą susimąstom: ar iš tikrųjų mums reikia išskaičiuoti pirminių segmentų susikirtimus? Ar negalima, išskaičiavus tikslinių segmentų susikirtimus (t.y. UNL – salinių), automatiškai suformuluoti jam atgalinį į pirminę kalbą vertimą? Teigiamą atsakymą į šiuos klausimus galima duoti, jeigu vėl pasinaudotume mašininio vertimo technologija. Iš tikrųjų, visi konceptai turi savo vertimą į pirminę kalbą, atitinkamai, mašininio vertimo silpnosios vietos – leksikos parinkimo – pavyks išvengti. Viskas, ko bus reikalaujama iš kompiuterio – tai pirminiame segmente išskirti tuos žodžius ir sintaksinius ryšius, kurie pateko į UNL sakinių susikirtimų sudėtį, ir suformuoti naują žodžių junginį, reikiamai pakeitus žodžių formas (pav. 7).



Pav. 7

Kaip greitai mes patikėjom mašininio vertimo sistemai sintaksinę ir morfologinę segmento analizę, kai vertinome kalbinių porų izomorfizmo persikirtimus be UNL panaudojimo, patikėsime jai padaryti tą patį segmento vertimų atmintyje paieškos organizavimui. Iš tikrųjų, kodėl gi nesudaryti pirminį segmentą UNL sakinyje ir neatlikti paieškos segmentų grafe, saugojančiame tekstą UNL kalba? Taip pasielgus, mes visiškai išvengsime būtinybės atlikti paieškos operacijas ir pridėjimo prie segmentų grafo, saugojančio tekstą tikrinėje kalboje. Visos operacijos bus atliekamos UNL sakinių grafe. Dabar vietoj kelių grafų (po vieną kiekvienai kalbai) vertimų atmintis naudos vieną vienintelį grafą, kurio kiekvienas mazgas vaizduos kalbinę žvaigždę su UNL sakiniu centre ir vertimo variantais spinduliuose.

Visas vertėjo darbo procesas su siūloma sistema aprašomas schema, pavaizduota 8 paveiksle.



Pav. 8

Svarbiu faktoriumi yra tai, kad klasikinės vertimų atminties darbas aprašomas tokia pat schema. Tai reiškia, kad siūlomo modelio realizacija gali būti lengvai įstatyta į egzistuojančias sistemas.

Mes sudarėme vertimų teorijos modelį, kurio pagrindas yra orientuotas santykių paveldėjimo grafas, kurio mazguose yra reikšmės (konceptai) skirtingo konkretumo laipsnio. Tuo šakniniuose (neturinčiuose protėvių) grafo mazguose yra labiau bendri abstraktūs konceptai, atitinkantys terminologinio žodyno elementus. Daugkartinio paveldimumo būdu iš jų gimsta sudėtiniai konceptai, atitinkantys konkretesnes reikšmes. Su kiekvienu grafo konceptu susieti jo vertimo į įvairias kalbas variantai. Ne kiekvienam konceptui galimas vertimas į užduotą kalbą. Iš kitos pusės, kai kuriam konceptui gali būti priskirta keletas vertimo į vieną ir tą pačią kalbą variantų.

Tai buvo trumpas siūlomo priėjimo techninės pusės reziumė. Bet daug svarbesnis yra technologinis aspektas. Ilgą laiką mašininio vertimo ir vertimų atminties sistemos atstovavo dvi konkuruojančias kryptis ir niekada nebuvo analizuojamos kartu, nebent tik kaip priešingybės. Šiai dienai požiūriai keičiasi, ir nors firmos nepriduoda savo naujienoms didelio viešumo, pastebėtina tendencija bendram abiejų technologijų vartojimui kai kuriose sistemose. Siūlomas modelis demonstruoja vieną iš galimų tokios integracijos variantų. Dar daugiau, ji reiškia

bandymą parodyti, kad mašininiam vertimui ir vertimų atminčiai galima suteikti bendrą pagrindą, ir sukurti tokią profesionalaus vertimo sistemą, kurioje abu mechanizmai veikia kaip viena visuma.

5. PROJEK TINĖ REALIZACIJA

Projektinės realizacijos dalyje aprašoma programinės įrangos projektinė dokumentacija. Projektinėje dalyje pateikiamas realizacijos kelias, pasirinktas analizės dalyje. Joje taip pat turi būti apžvelgta projektuojamos sistemos specifikacija, programuojamos sistemos architektūra, siūlomas programos prototipinis modelis, programinių modulių ir objektų specifikacijos.

5.1 Sistemos projektas

Šiame skyriuje pateikiamas sistemos projekto aprašymas.

5.1.1 Reikalavimų projektuojamai sistemai specifikacija.

Užsakovo pateikti reikalavimai sistemai:

- Sistema turi būti naudojama daugelyje platformų.
- Sistema turi būti lengvai modifikuojama.
- Sistema turi palaikyti šių formatų failus:
 - neformatuotas tekstas;
 - html;
 - xhtml;
 - Java resursų failai (su .properties vardų išsiplėtimais);
 - OpenOffice.org 1.0/StarOffice 6.0 arba aukščiau.
- Paprastas ir aiškus sistemos įsisavinimas ir eksploatavimas.

5.1.2 Funkciniai reikalavimai

- Sistema turi, kaip ir kitos, aukščiau išvardintos sistemos, naudoti TM technologiją. Ir palaikyti daugelis kalbų.

5.1.3 Nefunkciniai reikalavimai

Naudojamumas

- Aiškus ir suprantamas vartotojo interfeisas.
- Neilgas išmokymo naudotis programa laikas.

Patikimumas ir kokybė

- Nepatekti į kritines situacijas.
- Sistema turi būti labai patikima

Mobilumas

- Lengvas sistemos perkėlimas iš vieno kompiuterio į kitą.

Saugumas

- Programa neturi kenkti sistemos, kurioje vykdoma, stabilumui.

Programa „*Transmem*“ apdorota Windows XP Professional platformoje. Ši dokumentacija skirta vartotojams visose platformose.

5.2 Kuo ypatinga programa „*Transmem*“?

Šiandien rinkoje daug programų, naudojančių vertimų atmintį. Programos „*Transmem*“ ypatybės yra šios:

- ji veikia su Java Runtime Environment, ir gali būti naudojama daugelyje platformų, įskaitant Windows, Linux, ir Mac OS X;
- jos apdorojime pagrindas yra vertimų greitis, o ne sudėtingo tekstų formatavimo palaikymas ar daugelis failų formatų;
- ji betarpiškai palaiko OpenOffice.org failų formatą.

5.3 Sisteminiai reikalavimai

Programos „*Transmem*“ darbui reikalinga Java Runtime Environment (Java išpildymo terpė). Kaip taisyklė, jeigu sistemoje yra Java Runtime Environment (taip pat vadinama J2RE), programa „*Transmem*“ gali funkcionuoti. Java Runtime Environment realizuota šiose platformose: Windows 95/98/2000/ME/XP, UNIX, Linux, Solaris (x86 ir Sun SPARC), ir Macintosh OS X.

Programa „*Transmem*“ neveikia Windows NT dėl defekto realizuojant Java NT.

5.4 Nustatymas

Nustatymas turi tris stadijas. Pirmiausia, nustatyti Java Runtime Environment (jeigu dar nenustatyta). Visų antra – nustatyti programą „*Transmem*“. Trečia – suderinti. J2RE programa „*Transmem*“ galima nustatyti bet kuria eilės tvarka. Nustatymas nėra labai svarbus, bet palengvina priedo paleidimą.

5.5 Apie Java Runtime Environment (Java Išpildymo Terpę)

Java Runtime Environment (J2RE) leidžia paleisti sistemoje programas, parašytas Java pagalba. Ją galima užkrauti kaip atskirą produktą ir Java apdorotojo instrumentarijaus sudėtyje.

Java taip pat pasiekama prijungiamas prie brauserio modulis. Ši Java forma neleis paleisti programą „*Transmem*“.

JavaScript – visai kitas produktas, techniškai visai nesusijęs su Java.

Jeigu jūs norite panaudoti programą „*Transmem*“, jūsų sistemoje būtinai reikia nustatyti Java Runtime Environment. Be jos programa „*Transmem*“ neveiks. Neverta programos „*Transmem*“ paleisti be Java Runtime Environment. Tai tuščias laiko gaišimas.

Jeigu jūs nežinote, ar nustatyta Java Runtime Environment jūsų sistemoje, paprasčiausiai pabandykite nustatyti ir paleisti programą „*Transmem*“ (programos „*Transmem*“ nustatymo nurodymai aprašyti žemiau). Jeigu nieko nevyksta arba pasirodo pranešimas apie klaidas, Java išpildymo terpė, greičiausiai, nenustatyta. Šiuo atveju ją reikia gauti ir nustatyti.

5.6 Java Runtime Environment Nustatymas (Java Išpildymo Terpės)

Vienintelė platforma, nereikalaujanti Java Runtime Environment, - tai MacOSX. MacOSX pristatoma su iškart nustatyta ir sukonfigūruota JRE.

Windows vartotojams: sužinoti, kokia Java Runtime Environment versija nustatyta, galima, paspaudus mygtuką *Start* / paspaudus mygtuką *Windows* ir pasirinkus *Settings>Control Panel*. Jeigu J2RE nenustatyta, bus matoma valdymo panelė su užrašu “Java(TM)2 Runtime Environment, Standard Edition 1.3.0_01”.

Linux vartotojams: paprasčiausiai įveskite *Java-version* komandos eilutėje. Bus sugrąžinta nustatyta versija (aišku, jeigu ji nustatyta sisteminiu lygiu).

J2RE galima užkrauti pagal nuorodą žemiau. Ji nemokama (atkreipkite dėmesį į licenzijos sąlygas), bet pavadinimo taisyklės gali šiek tiek sukomplicuoti reikalingos versijos paiešką. Reikės versijos aukščiausiai platformai (Linux, Windows ir t.t.). J2RE apimtis – apie 16 Mb.

Programa „*Transmem*“ turi veikti su bet kuria J2RE versija nuo 1.2 ir vėliau. Vis dėlto rekomenduojama gauti paskutinę J2RE versiją, kadangi ankstesnės Java versijos gali nepalaikyti kai kurių funkcijų.

Kaip J2RE alternatyvą galima nustatyti apdorotojo instrumentarijų. Apdorotojo instrumentarijus turi savo J2RE kopiją, o taip pat leidžia keisti programą „*Transmem*“.

Windows ir Linux vartotojai gali užkrauti Java Runtime Environment su tinklapiu:

- Sun (visos platformos)
- Blackdown (Linux)

Nustatykite J2RE pagal duotus nurodymus.

5.7. Programos „Transmem“ nustatymas

Visos platformos

Paprasčiausiai išpakuokite archyvo turinį į pasirinktą direktoriją. Archyvas gali būti *.zip arba *.tar.gzip formate. Linux ir Mac (Stuffit) vartotojai bet kokių atveju neturėtų turėti problemų; Windows vartotojai gali neturėti utilitų paskutiniame.

Tik Windows

Windows vartotojai gali užkrauti programą Windows sistemai naudojamam failo *Transmem_setup.exe* pavidalu, kuri paprasčiausiai reikia įjungti. Tokiu atveju programa sukurs nuorodą programos „Transmem“ įjungimui į meniu *Start > Programs*.

Suderinimas (Tik Linux)

Jeigu jūs naudojate Linux, bet nelabai gerai jį žinote, patarimas, kaip galima padaryti lengvesnį programos „Transmem“ įjungimą:

Sukurkite shell-skriptą, parašę tokio turinio tekstinį failą:

```
#!/bin/bash
cd
./java -jar Transmem.jar
```

kur – direktorija, turinti failą *Transmem.jar*, o – direktorija, turinti naudojamą J2RE failą (kuris vadinamas *java*). Jeigu nustatytas Java kelio kintamasis, tos direktorijos nereikia nurodyti, užtenka *java -jar Transmem.jar*. Nurodyti direktoriją reikia, jeigu nenustatytas Java kelio kintamasis, o taip pat jeigu sistemoje nustatyta keletas Java versijų, ir reikia užtikrinti, kad programa „Transmem“ būtų paleidžiama su reikiama versija.

Išsaugokite tekstinį failą kaip *TRANSMEM* ir padarykite jį naudojamu (pavyzdžiui, Konqueror kontekstinis meniu > *Properties > Permissions > Exec*). Dabar, kad įjungti programą „Transmem“, galima paprasčiausiai įvesti *Transmem* komandos eilutėje. Norint galima patalpinti ženkliuką ant ekrano, nustačius jį kaip failą skriptą.

Priedo Paleidimas

Macintosh OS X

Paspauskite failą *TRANSMEM.jar* failų valdymo programoje.

Windows

Jeigu programa „Transmem“ nustatyta Windows *.exe pagalba (žiūrėkite aukščiau), ją galima paleisti, paspaudus nuorodą “Transmem” meniu *Start > Programs > Transmem*. Taip pat galima paspausti failą *Transmem.exe*, *Transmem.bat* arba *Transmem.jar* failų vadove.

Linux

Įveskite *java -jar Transmem.jar* komandos eilutėje.

Jeigu Java nesuderinta rasti programos „Transmem“ failus, iš pradžių reikia pereiti į direktoriją, turinčią failą *Transmem.jar*.

Jeigu nenustatytas Java kelio kintamasis, reikia nurodyti visą kelią. Jeigu nepadaryta nieko iš nurodyto aukščiau, įjungimas turi būti maždaug toks:

```
cd /home/user/transmem
/home/user/j2re1.4.2/bin/java -jar transmem
```

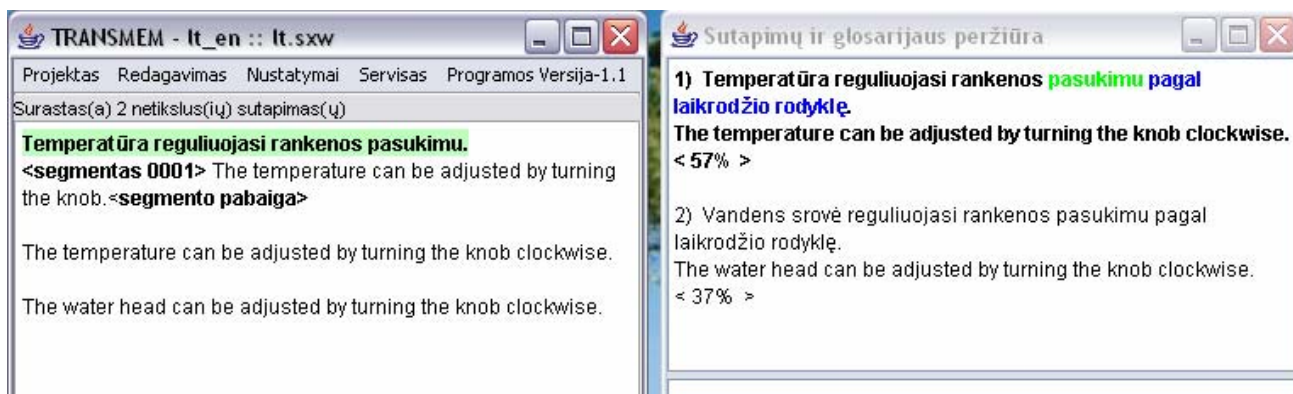
5.8 Svarbūs pastebėjimai

Programa „Transmem“ – patikima programa, nelinkusi gesti ir prarasti arba pažeisti duomenis. Vis dėlto naudojant programą „Transmem“ reikia rūpintis duomenų išsaugojimu – kaip ir naudojant bet kokią kitą programą.

- Naudokite programą „Transmem“ versti trumpus tekstus, kol nepriprasisit.
- Įdėmiai perskaitykite skyrių *Išsaugojimas, vertimo tikrinimas ir palyginimas*.

Taip pat reikia priminti, kad programa „Transmem“ sukuria rezervinę vertimų atminties kopiją faile *project_save.tmx.bak* antrinėje projekto *Transmem* direktorijoje kiekvieną kartą užsaugant ar lyginant projektą.

5.9 Vartotojo interfeisas



Programos „Transmem“ vartotojo interfeisas susideda iš pagrindinio programos lango ir sutapimų bei glosarijų apžvalgos lango.

Pagrindiniame programos „*Transmem*“ lange atsispindi verčiamas tekstas. Kiekvienam segmentui atsispindi arba galutinis tekstas, arba vertimas, priklausomai nuo to, išverstas segmentas, ar ne. Išimtis yra aktyvus segmentas, kuriame atsispindi ir galutinis tekstas, išskirtas pusstoriu pabrėžimu, ir vertimas, patalpinamas tarp <segmentas numeris...> ir <segmento pabaiga>.

Sutapimų peržiūros ir glosarijaus langas suskirstytas į dvi paneles. Sutapimų panelė parodo netikslius sutapimus (anksčiau išversti segmentai, turintys pakankamą panašumą į aktyvų segmentą). Glosarijaus panelė atspindi informaciją iš glosarijaus failo, jeigu jis yra.

Dažniausiai nėra būtinybės atidaryti kitų langų. Norint galima palikti atidarytą langą su projektų failų sąrašu kartu su segmentų kiekiu juose.

Visos funkcijos pasiekiamos per meniu pagrindinio programos „*Transmem*“ lango viršuje. Daugelį funkcijų galima iškviešti klavišų kombinacijų pagalba.

5.10 Segmentavimas

Programa „*Transmem*“ segmentuoja verčiamą tekstą pagal pastraipas.

Programa „*Transmem*“ skelia tekstą į segmentus atidarius projektą. Dirbant su „*Transmem*“, negalima keisti segmentų ribų. Jeigu jūs numanote, kad tekste yra nereikalingi eilučių tarpai, tikslinga redaguoti tekstą pirmiau, negu atidaryti projektą. Jeigu dirbant su programa „*Transmem*“ sutiksime netikslų segmentavimą (pvz., eilutės tarpai sakinio viduryje), galima ištrinti tarpą iš išeinamojo teksto (pavyzdžiui, tekstų redaktoriuje). Pakeitimai pasirodys programoje „*Transmem*“, bet vis dėlto, tik po projekto perkrovimo.

5.11 Vertimo procesas

Atidarykite projektą ir išspausdinkite vertimo tekstą virš galutinio teksto aktyviame segmente.

Paspauskite *Enter* arba *Ctrl+N*, kad patvirtinti tekstą ir pereiti prie sekančio segmento. Jeigu reikia naudoti *Enter* kitiems tikslams, pavyzdžiui, sudėtingų ženklų įvedimo sistemose, vietoj jo galima panaudoti *Tab*. *Tab* panaudojimui reikia aktyvuoti atitinkamą opciją per meniu *Suderinimas > panaudoti TAB perėjimui toliau*.

Atidarant projektą automatiškai aktyvuojasi pirmas segmentas. Greitam sekančio neišversto projekto segmento radimui, paspauskite *Ctrl+U* arba pasirinkite meniu punktą *Redagavimas > Sekantis neišverstas segmentas*.

Verčiant visus segmentus, kompiliuokite, pasirinkę meniu punktą *Projektas > Kompiluoti vertimą*.

Failai, gauti kompiliavus, patalpinami į projekto antrinę direktoriją */target*, kur jas galima pasižiūrėti. Jeigu rezultatai nepatenkinami, paprasčiausiai pakartokite vertimą pagal nurodytą procedūrą.

5.12 Redagavimo funkcijos

Programos „*Transmem*“ redagavimo funkcijos naudojamas redaktorius lauke, kuris yra pagrindinis aktyvaus segmento tekstas. Redagavimo operacijos ribojamos tuo lauku, t.y. bet kuriuo laiko momentu galima redaguoti tik tikslinį tekstą aktyviame segmente. Tekstas bet kuriame tekstiniam lauke ne redaktorius lauke, vis dėlto, galima išskirti ir nukopijuoti buferį sekančiam įdėjimui.

Programa „*Transmem*“ patvirtina įprastas operacijas pele, kursoro valdymo klavišais ir kursoro valdymo klavišų kombinacijomis, o taip pat klavišų kombinacijomis išskyrimo, iškirpimo ir įdėjimo operacijoms atlikti. Jie, greičiausiai, skirsis įvairiose operacinėse sistemose.

Veiksmo nutraukimo ir pakartojimo operacijas galima atlikti kombinacijas *Ctrl-Z* ir *Ctrl-Y* lygiagrečiai.

Verčiant tekstą negalima naudoti formatavimo. Failo formatavimas (dabar palaikomi StarOffice/OpenOffice.org ir HTML formatai), išsaugojamas, ir egzistuoja ribotas jo kontrolės lygis.

Kad panaudotumėte formatavimą, įstatykite simbolį, pavyzdžiui "###" ir suformatuokite tekstą kitoje programoje, pavyzdžiui, OpenOffice.org po kompiliavimo.

5.13 Judėjimas tekstu

Kad pereitumėte į:

- sekantį segmentą, spauskite *Enter*, *Ctrl+N* arba *Tab*, jeigu padarytas atitinkamas pasirinkimas meniu *Suderinime*.

- Ankstesnis segmentas, spauskite *Ctrl+P*

- Sekantis neišverstas segmentas, spauskite *Ctrl+U*

Perėjimui prie kito failo tame pačiame projekte, iškvieskite projekto failų sąrašą (*Projektas > Parodyti failų sąrašą*). Paspauskite pele norimo failo vardą pereiti į jo pradžią.

5.14 Pilni ir netikslūs sutapimai

Atidarant segmentą, programa „*Transmem*“ ieško sutapimų dabartiniame tekste, o taip pat vertimų ir direktorijos */tm* atminties failuose. Jeigu randamas 100%-tinis sutapimas ir jis

tikslus, eilutė į eilutę atitinka verčiamą tekstą, juo automatiškai keičiamas tekstas tiksliniame aktyvaus segmento lauke (tai reiškia, kad nors „Labas“ ir „Labas!“ yra beveik 100%-tiniai sutapimai, automatinis pakeitimas neįvyks todėl, kad eilutės nevienodės). Netikslūs sutapimai atsispindi sutapimų ir glosarijaus peržiūros lange.

Sutapimų ir glosarijaus peržiūros lange atsispindi iki penkių pilnų ir netikslų sutapimų. Spauskite *Ctrl+1*, kad išrinktumėte pirmą sutapimą, *Ctrl+2*, kad išrinktumėte antrąjį, ir t.t..

Kada sutapimas pasirinktas, jį galima įstatyti į aktyvų segmentą. *Ctrl+I* leidžia įstatyti sutapimą į kursoro poziciją, *Ctrl-R* keičia aktyvaus segmento tekstą sutapimu.

Sutapimo procentas, rodomas programoje „*Transmem*“, yra grynai techninio pobūdžio. Jis negali parodyti, kiek darbo jau padaryta už jus!

5.15 Paieška pagal esminius žodžius ir tiksli paieška

Programa „*Transmem*“ turi galingą paieškos funkciją.

Iškvieskite paieškos funkciją, paspaudę *Ctrl+F*, ir įveskite ieškomą žodį arba frazę lauke „Rasti“.

Kitas būdas – išskirti žodį arba frazę į redagavimo lauką ir paspausti *Ctrl+F*. Šiuo atveju žodis ar frazė automatiškai įstatoma į lauką „Rasti“.

Pasirinkite paieškos tipą selekcinio mygtuko pagalba ir įjunkite būtinas opcijas.

Kiekvienam paieškos tipe žodžių registras nenumatomas, t.y., „abc“ paieška leidžia rasti abc, ABC, Abc ir t.t..

5.15.1 Paieška pagal esminius žodžius

Išskirkite „Paieška pagal esminius žodžius“ ir įveskite bet kokį žodžių bet kokia tvarka skaičių, ir programa „*Transmem*“ parodys visų segmentų, kuriuose sutinkami visi žodžiai, sąrašą. Paieška pagal esminius žodžius analogiška opcijai „Ieškoti visų žodžių“ paieškos sistemose, tokiose kaip Google, Internete.

Paieška pagal esminius žodžius leidžia rasti tik pilnus žodžius, o ne žodžių dalis ar neužbaigtus žodžius.

5.15.2 Tiksli paieška

Tiksli paieška leidžia rasti frazę, t.y., jeigu įvesti keli žodžiai, jie bus rasti tik tuo atveju, jeigu jie yra tekste nurodyta tvarka. Tiksli paieška analogiška opcijai „rasti tikslią frazę“ paieškos sistemose, tokiose kaip Google, Internete.

Tikslią paiešką galima vykdyti:

- galutinio teksto ir dabartinio projekto segmentuose;
- galutinio teksto ir dabartinio projekto vertimo segmentuose bei vertimų atminties failuose (failuose direktorijoje */tm*);
 - bet kuriuose failuose, kurių formatą programa „*Transmem*“ gali perskaityti (.txt, .html, .xhtml или .sxw), bet kurioje pasirinktoje direktorijoje ir pagalbinėse direktorijose. Vykdamt tikslią paiešką galima panaudoti simbolius '*' ir '?'.
 - Simbolis '?' keičia vieną bet kurį simbolį.
 - Simbolis '*' keičia bet kurį simbolį arba simbolių seką.

5.16 Vertimo išsaugojimas, patikrinimas ir palyginimas

Išsaugojimas

Norint išsaugoti darbą bet kuriuo momentu, išskirkite *Projektas > Išsaugoti* arba paprasčiausiai spauskite *Ctrl+S*.

Tokiu būdu vertimas bus išsaugotas ne kaip dokumentas, o kaip tarpinė vertimų grandis faile */TRANSMEM/project_save.tmx*.

Patikrinimas

Norint patikrinti tekstą, lyginant vertimą ir originalą, paprasčiausiai pereikite nuo segmento prie segmento, spausdami *Enter arba Ctrl+N*. Vertėjai, pažįstantys kitas programas, pagrįstas vertimų atmintimi, įvertins šį patogų projektų patikrinimo metodą.

Kompiliacija

Verčiant programos „*Transmem*“ pagalba neįvyksta pirminio teksto pakeitimų, kiekvienos pastraipos vertimai bus išsaugoti atskirame faile. Kompiliacija – tai paprasčiausiai procesas, kurio metu programa „*Transmem*“ tame faile keičia teksto segmento originalą vertimais. Programa taip pat sukuria vertimų atmintį galutinėje formoje, tinkamoje pakartotiniam naudojimui. Belyginant nereikia laukti, kol bus išverstas visas tekstas, palyginimą galima vykdyti kiek nori kartų.

Patarimai

- Reguliariai darykite failo */TRANSMEM/project_save.tmx* kopijas. Prie kopijos vardo pridėkite */TRANSMEM/project_save.tmx-01* arba */TRANSMEM/project_save.tmx-04-05-04*. Šį failą galima pernešti į kitą nešėją, pavyzdžiui, į disketę, kuri yra patogus rezervinio kopijavimo būdas. Iš esmės, pervedant failus su formatavimu (StarOffice/OpenOffice.org, XHTML arba HTML), greičiausiai, failas *project_save.tmx* bus daug mažesnis negu originalo arba vertimo tekstas.

- Kol neįsisavinsite programos „*Transmem*“, darykite kompiliaciją reguliariai ir tikrinkite, kad sukompiliuotas failas turėtų paskutinę vertimo versiją.
- Būkite ypatingai atsargūs darant pakeitimus direktorijoje */source* į failus projekto eigoje. Kartais reikia pakeisti pradinius failus, pavyzdžiui, norint ištrinti eilučių nuotrupas programos viduryje. Vis dėlto jeigu pradinio failas pakeistas, kada jau pradama versti, programa „*Transmem*“ gali nerasti jau išversto segmento. Visada darykite atsarginę pačio pradinio failo kopiją, arba viso projekto, pirmiau, nei keisti pradinį failą.
- Jeigu atidarytas bet koks projektas, nereikia kurti naujo arba atidarinti kito. Uždarykite atvirą projektą ir esant būtinybei perkraukit programą „*Transmem*“, pirmiau, nei atidarysit kitą projektą.

5.17 Glosarijus

Glosarijus funkcija veikia automatiškai. Norint panaudoti esamą glosarijų, paprasčiausiai patalpinkite jį į direktoriją */glossary* po projekto sukūrimo. programa „*Transmem*“ automatiškai atpažįsta glosarijus failus šioje direktorijoje atidarius projektą.

Terminai aktyviame segmente, kuriuos programa „*Transmem*“ randa glosarijus faile, atsispindi sutapimų ir glosarijus peržiūros lange.

Glosarijus funkcija ieško tik tikslių sutapimų su glosarijus straipsniu.

Naujus terminus galima rankiniu būdu įvesti į glosarijus failą vertimo metu, bet vėl pridėti terminai bus atpažįstami tik sekantį kartą atidarius projektą.

5.18 Verčiamų failų formatai

Program „*Transmem*“ palaiko šių formatų failų vertimą:

- neformatuotas tekstas;
- html;
- xhtml;
- Java resursų failai (su *.properties* vardų išsiplėtimais);
- OpenOffice.org 1.0/StarOffice 6.0 arba aukščiau.

Programa „*Transmem*“ automatiškai nustato failo formatą failo vardo platesnės reikšmės pagrindu, todėl failai turi turėti platesnes reikšmes *.txt*, *.html* ir *.sxw* neformatuotam tekstui, *html* ir *OpenOffice.org/StarOffice* atitinkamai. Platesnių reikšmių ir atitinkamų formatų palyginimas aprašytas faile *omega.prefs* direktorijoje, kur nustatyta „*Transmem*“. Redaguojant tą failą, galima priversti programą „*Transmem*“ atpažinti failus su kitokiomis platesnėmis vardų reikšmėmis (bet ne formatais!).

OpenOffice.org turi puikius filtrus failų eksportui į kitus formatus, įskaitant MS Word, Excel ir Powerpoint.

5.19 Vertimų atminties ir glosarijaus failų formatai

Vertimų atminties failai, importuojami, eksportuojami ir naudojami „*Transmem*“ programos, atitinka standartinį formatą TMX. Dėl to, vertimų atminties failai, sukurti „*Transmem*“, gali naudoti kitas CAT programas, palaikančias šį standartą. Programa „*Transmem*“, savo ruožtu gali importuoti ir naudoti atminties failus, sukurtus tų programų, jeigu jie išsaugoti TMX formate.

Programa „*Transmem*“ negali naudoti vertimų atminties failų, sukurtų kitų CAT programų proprietariniuose formatuose.

Glosarijaus failai – tai tekstiniai failai su praplatėjimu *.tab, turintys trijų eilių sąrašus, atskirtus tabuliacijomis, kur pirmoje eilutėje yra pradiniai terminai, o antroje – jų vertimai. Trečia eilutė gali turėti papildomą informaciją.

Glosarijaus failai gali būti ne tik sisteminėje koduotėje, bet ir UTF-8.

Glosarijus, sukurtas Multiterm programos iš Trados paketo teksto eksporto funkcijos pagalba, galima naudoti kaip „*Transmem*“ programos be papildomų pasikeitimų glosarijus, jei jiems suteikti failo vardą su praplėtimu *.tab.

5.20 Projekto failai ir direktorijos

5.20.1 Failų ir direktorių projekto aprašymas

Programos „*Transmem*“ projektas turi šiuos failus ir direktorijas.

Pagrindinė projekto direktorija. Ši direktorija sukuriamas, kai sukuriamas projektas. Jos vardą sugalvoja vartotojas.

- *TRANSMEM.project.* Kada sukuriamas projektas, programa „*Transmem*“ automatiškai sukuria šį failą ir patalpina jį į šią direktoriją. Norint atidaryti programos „*Transmem*“ projektą, reikia atidaryti jo failą *TRANSMEM.project.*

- *<projekto pavadinimas>.tmx.* Tai vertimų atminties failas, kurį eksportuoja projektas. Programa „*Transmem*“ sukuria jį automatiškai ir patalpina į šią direktoriją kiekvieną kartą, kai lygina projektą. Reikia prisiminti, kad šis vertimų atminties failas turi tik segmentus iš projekto paskutinio jo palyginimo metu. Į jį neimportuojami segmentai iš kitų vertimų atminties failų direktorijoje /tm.

- Direktoriija */source*. Patalpinkite failus vertimui į šią direktoriiją po projekto sukūrimo ir prieš jo atidarymą. Bet kurie failai šioje direktorijoje ir jos antrinėse direktorijose įtraukiamos į projektą vertimui, jeigu palaikomas jų formatas, ir jos turi atitinkamą failo vardo reikšmės praplatėjimą.

- **.txt*,
- **.xhtml*, **.html*,
- **.properties*,
- **.sxw*

Failai */source*, kurie neimportuoti į projektą, perduodami be pasikeitimų į */target* lyginant projektą. Tai leidžia sukurti pilnų web kilpų veidrodžius iš */source* į */target* kartu su užkraunamais jų paveikslais tokiuose formatuose, kaip GIF arba JPEG.

- Direktoriija */glossary*. Po projekto sukūrimo patalpinkite į tą direktoriiją glosarijų, kurį jūs norite panaudoti.

- **.tab*, **.utf8*: glosarijaus failai.

- Direktoriija */TRANSMEM*. Programa „*Transmem*“ kurdama automatiškai patalpina į tą direktoriiją šiuos failus:

- *project_save.tmx*: šis failas sukuriamas pirmą kartą išsaugojus projektą. Jis turi TMX formatą ir yra projekto vertimų darbinė atmintis.

- *word_counts*: šis failas turi du užbaigto projekto žodžių skaitiklius: vienas, sutampantis 100%-u, ir vienas, pašalinantis.

- Direktoriija */target*. Programa „*Transmem*“ lygindama patalpina į tą direktoriiją failus su vertimais.

- **.txt*, **.xhtml*, **.html*, **.properties*, **.sxw*: išversti failai.

- Neišversti failai perduodami iš */source* kompiluojant.

- Direktoriija */tm*. Po sukūrimo ir prieš projekto paleidimą įdėkite į tą direktoriiją *vertimų atminties failus*, kuriuos jūs norite panaudoti. Nepriklausomai nuo to, naudojami ar ne anksčiau sukurti vertimų atminties failai, programa „*Transmem*“ sukuria vertimų darbinę atmintį projektui, *project_save.tmx*, in */TRANSMEM* išsaugant projektą, o taip pat patalpina eksportuotą vertimų atmintį į pagrindinę direktoriiją projekto lyginimo metu.

- **.tmx*: vertimų atminties failai, sukurti ankstesniuose projektuose. Failai **.tmx* turi būti TMX formate.

5.20.2 Failų ir projekto direktorijų buvimo vietos pakeitimas

Projekto direktorijos sukuriamos kuriant projektą. Pagal nutylėjimą jos grupuojasi į pagrindinę projekto direktoriją, bet joms galima išrinkti kitas buvimo vietas, paspaudus pele klavišą *Peržiūra*. Tokiu būdu galima laikyti translation memories apibrėžtose direktorijose, pavyzdžiui, ir išsirinkti norimą direktoriją kuriant naują projektą, vietoj to, kad kopijuoti vertimų atminties failą į vėl sukurtą direktoriją /tm.

5.21 Klavišų kombinacijos

<i>Ctrl+O</i>	Atidaryti egzistuojantį projektą
<i>Ctrl+S</i>	Išsaugoti vertimų atmintį
<i>Ctrl+Q</i>	Išeiti iš programos
<i>Enter, Ctrl+N, Tab*</i>	Pereiti prie sekančio segmento
<i>Ctrl+P</i>	Pereiti prie ankstesnio segmento
<i>Ctrl+F</i>	Vertimų atminties paieška
<i>Ctrl+R</i>	Keisti tekstą aktyviame segmente išsirinktu netiksliu atitikimu
<i>Ctrl+I</i>	Įstatyti išsirinktą netikslių atitikimą į aktyvų segmentą į kursoro poziciją
<i>Ctrl+1, 2, u m. n.</i>	Išsirinkti netikslių atitikimą Nr.1, netikslių atitikimą Nr.2 ir t.t..
<i>Ctrl+L</i>	Parodyti projekto failų sąrašą
<i>Ctrl+U</i>	Pereiti prie sekančio neišversto segmento

5.22 Skirtingų šriftų ir abėcėlių panaudojimas

Keisti ekrano šriftą galima pobūdžiu *Nustatymas > Šrifto pasirinkimas*. Galima pasirinkti baldą ir kėglį. Išsirinktas šriftas naudojamas tik išėjimui į ekraną, šriftai, naudojami dokumentuose su formatavimu, neliečiami. Pasirinkimui pasiekiami šriftai, nustatyti sistemoje.

Kad padaryti įmanomą vertimą tarp dviejų kalbų, naudojančių skirtingą abėcėlę, pasirinkite šriftą, turinti **both** abėcėlės simbolius.

Jeigu nei vienas atskiras šriftas neturi abiejų abėcėlių ženklų, galima išrinkti virtualų Java šriftą, ir nustatyti tą šriftą tokiu būdu, kad jis naudotų daugelį realių šriftų.

5.23 Darbas su openoffice.org ir staroffice failais

Openoffice.org ir Staroffice skirtumai

Plačiaja reikšme OpenOffice.org ir StarOffice – ta pati programa. OpenOffice.org – programa su atviru galutiniu kodu, kas reiškia, kad ji nemokama ir gali būti laisvai kopijuojama. StarOffice – komercinis produktas: reikia nusipirkti licenziją kiekvienai sistemai, kurioje planuojama ją naudoti. StarOffice turi apibrėžtus komponentus, kurių nėra OpenOffice.org; licenzijos gavimas taip pat duoda teisę techninei pagalbai.

Openoffice.org ir transmem

TRANSMEM – savarankiška programa, nereikalaujanti darbui OpenOffice.org arba StarOffice, MSWord, arba kitokio tekstų redaktoriaus.

Vis dėlto OpenOffice.org ir StarOffice žymiai praplatina programos „Transmem“ galimybes, leisdami dirbti su formatais, kurių šiaip nepalaiko. Iš dalies tai MS Word, MS Excel, MS Powerpoint ir RTF. Vertimo failams parengti tuose formatuose paprasčiausiai atidarykite juos OpenOffice.org programoje ir išsaugokite savo OpenOffice.org formate. Baigus vertimą, atidarykite vertimo tekstą OpenOffice.org programoje ir išsaugokite originalo formatu.

OpenOffice.org/StarOffice išplečia programos „Transmem“ galimybes. Klaidų tikrinimas ir segmentavimas pagal sakinius, o ne pagal pastraipas – dvi funkcijos, kurių neturi programa „Transmem“, bet pasiekiamos OpenOffice.org pagalba.

Naudokite su programa „Transmem“ OpenOffice.org 1.0/StarOffice 6.0 arba dar vėlesnes versijas. Ankstesnės versijos gali neveikti. StarOffice 5.2 neveiks.

Patarimai

Įsitikinkite, kad išsaugant failus MSWord formate nebuvo išsirinkta funkcija “greitas išsaugojimas”, kitaip OpenOffice.org greičiausiai negalės jų perskaityti.

Nereikia užkrauti programą „Transmem“ failų, išsaugotų su įjungta pakeitimų įrašymo funkcija, OpenOffice.org programoje.

Iš pradžių išjunkite šią funkciją ir patvirtinkite visus pakeitimus. Ši problema visiškai galima, kada jūs gaunate MSWord failus, išsaugotus su įjungta pakeitimų įrašymo funkcija, ką jūs žinote. OpenOffice.org palaiko šią funkciją, bet jos negalima naudoti darbui su programa „Transmem“.

Gaukite ir nustatykite klaidų taisymo funkciją kalboms, su kuriomis dirbate. Po to, prieš užkraunant projektą, turintį OpenOffice.org failus, atidarykite juos OpenOffice.org programoje ir pakeiskite teksto kalbą į tikslinę kalbą, išskyrę visą tekstą ir paspaudę *Ctrl-A*, po to *Format > Character > Language*. Po to galima naudoti OpenOffice.org klaidų taisymui teksto vertimui po palyginimo. Kalbą galima pakeisti ir po palyginimo, bet jeigu palyginimas daromas kelis kartus, kiekvieną kartą po palyginimo teksto kalba bus pradinė kalba.

Po teksto palyginimo OpenOffice.org formate paskutinį kartą patikrinkite, ar viskas išversta. Bet kuriuo atveju, reikės atidaryti failą OpenOffice.org programoje, patikrinti klaidas. Gali būti, kad programa „*Transmem*“ praleido kokias nors teksto dalis.

5.24 Bendri patarimai ir teiginiai

5.24.1 Microsoft Office failai

Dirbant su Microsoft Office failais iš pradžių perdarykite juos į OpenOffice.org formatą ir patalpinkite perdarytus failus į direktoriją */source*. Nereikia į */source* patalpinti pradinių failų formate MS Office. Jeigu būtina, sukurkite jiems naują direktoriją, pavyzdžiui, */original*. Kada vertimas baigtas, perdarykite failus iš direktorijos */target* atgal į formatą MS Office OpenOffice.org pagalba.

5.24.2 Tuščias projektas? (Microsoft Office II failai)

Jeigu sukurtas naujas projektas, ir failai vertimui nukopijuoti į direktoriją */source*, bet atsiranda pranešimas, kad, projektas tuščias, tai, galbūt pradiniai failai MS Word formate nebuvo perdaryti į OpenOffice.org formatą.

Jeigu reikia pamatyti teksto maketą vertimo metu, paprasčiausiai laikykite originalaus failo kopiją atidarytą tekstiniame procesoriuje atskirame lange.

5.24.3 Tik vienas projektas!

Nekurkite naujo projekto ir neatidarinėkite egzistuojančio projekto, jeigu dar vienas projektas „*Transmem*“ programoje jau atidarytas. Gali atsitikti taip, kad vidinė aktyvaus projekto vertimų atmintis bus perduota naujam projektui.

5.24.4 Dingo segmentas...

Gali pasirodyti, kad po perkėlimo į kitos didelio projekto dalies segmentą pasidarys neįmanoma nustatyti segmento, versto anksčiau, buvimo vietą, kad prie jo būtų galima grįžti. Kad be problemų rastume segmentą, reikia įvesti ženklą tame segmentą ir paspausti *Enter* prieš tai, kol pereisite prie kitos projekto dalies. Dar geriau pridėti prie ženklo segmento numerį, nes jei segmentas – vienas iš kelių vienodų, gali pasirodyti, kad bandant ieškoti ženklas pasirodys keliuose segmentuose.

5.24.5 Ženklo įstatymas

Jeigu reikia įstatyti ženklą į vertimą, pavyzdžiui, kad pažymėtume neišspręstą klausimą, nereikia naudoti simbolių „?“ arba „*“, kurie naudojami paieškoje kaip keitinio simbolis, ir todėl negalima jo rasti *Ctrl-F* pagalba.

6. IŠVADOS

1. Mes sudarėme vertimų teorijos modelį, kurio pagrindas yra orientuotas santykių paveldėjimo grafas, kurio mazguose yra reikšmės (konceptai) skirtingo konkretumo laipsnio. Tuo šakniniuose (neturinčiuose protėvių) grafo mazguose yra labiau bendri abstraktūs konceptai, atitinkantys terminologinio žodyno elementus. Daugkartinio paveldimumo būdu iš jų gimsta sudėtiniai konceptai, atitinkantys konkretesnes reikšmes. Su kiekvienu grafo konceptu susieti jo vertimo į įvairias kalbas variantai. Ne kiekvienam konceptui galimas vertimas į užduotą kalbą. Iš kitos pusės, kai kuriam konceptui gali būti priskirta keletas vertimo į viena ir tą pačią kalbą variantų.
2. Programos „*Trasmem*“ ypatybės, lyginant su kitais produktais yra šios:
 - a. ji veikia su Java Runtime Environment, ir gali būti naudojama daugelyje platformų, įskaitant Windows, Linux, ir Mac OS X;
 - b. jos apdorojime pagrindas yra vertimų greitis, o ne sudėtingo tekstų formatavimo palaikymas ar daugelis failų formatų;
 - c. ji betarpiškai palaiko OpenOffice.org failų formatą.
3. Kadangi sistemą ketinama naudoti daug metų, pagrindinį dėmesį nuspręsta sutelkti į galimybes lengvai modifikuoti, bei išplėsti sistemą.
4. Sistema gali būti panaudota kaip vertėjų pagalbininkas.

7. LITERATŪRA

1. Arnold, D.L., Balkan L. and others (1994) *Machine Translation: An Introductory Guide*.
2. Blekhman, M.S. (1998) *Machine Translation: Professional Experience*.
3. Dobrovolskis, B., Kniūkšta, P., Kučinskaitė, A. ir kt. (2000) *Lietuvių kalbos žinynas*, Kaunas: Šviesa.
4. Lynn E. Webb (1999): *Advantages and Disadvantages of Translation Memory: A Cost/Benefit Analysis*, master thesis, San Francisco State University
5. Martin Volk: *The Automatic Translation of Idioms. Machine Translation vs. Translation Memory Systems*. In: Nico Weber (ed.): *Machine Translation: Theory, Applications, and Evaluation. An assessment of the state of the art*. St. Augustin: gardez-Verlag. 1998.
6. Misevičius, G. (2002) *Interlingua principai kompiuterinio vertimo sistemose į lietuvių kalbą*// KTU konf. „Informacinės technologijos 2002“, p. 311-316.
7. Misevičius, G., Tamulynas, B. (2001). *Kalbos vertimo kompiuterizavimo alternatyvios technologijos*// Mokslinė–techninė konferencija "Garso korta 2001 05 04": [Kompaktinis diskas], Kaunas: Technologija, ISBN 9955-09-063-4.
8. Mitamura, T., Nyberg, E.H., Carbonell, J.G. (1991) *An Efficient Interlingua Translation System for Multi-lingual Document Production* // Proc. of Machine Translation Summit III, Washington, DC.
9. Pacevičius, P., Tamulynas, B. (2001) *Konceptualus virtualaus žodyno modelis kompiuterizuoto vertimo sistemose*// KTU konf. „Informacinės Technologijos 2001“, Kaunas, p. 122-124.
10. Sidorov, S., Tamulynas, B. (2001) *Konceptualus hierarchinis kompiuterinio tekstų vertimo modelis*// KTU konf. „Informacinės Technologijos 2001“, Kaunas, p. 91-95.
11. Seminararbeit für das Virtueller Campus Seminar "Mensch-Maschine-Interaktion in der Maschinellen Übersetzung". Leitung: Dr. Folker Caroli und Prof. Claus R. Rollinger
12. Tamulynas, B., Žemaitis M. (2002) *Sintaksinių sakinio grupių sudarymas ir jų panaudojimas kompiuterinio vertimo programose*. Kaunas, KTU konf. „Informacinės Technologijos 2002“, p. 317-320.
13. *Trados GmbH: Handbuch zu Trados Translator's Workbench 2*
14. *Translation Memory Technologie und Maschinelle Übersetzung im Vergleich*. Prieiga per internetą: www.heeg.de/uta/AK-Protokoll-TM-.html
15. *Translation Memories: Übersetzungshelfer mit Gedächtnis*. Prieiga per internetą: www.doculine.com/zeitsch/9910/transmem.htm
16. *Translation Memory*. Bulletin of the Institute of Translation and Interpreting. 5/99

17. Trujillo, A. (1999) Translation Engines: Techniques for Machine Translation: Springer.
18. Глазунов А. Г. „Концептно-ориентированная модель памяти переводов“. Prieiga per internetą: <http://www.citforum.ru/programming/digest/cotm.shtml#24>
19. Олег Сонин, **МТ или ТМ**// Компьютерная неделя N26-27(200-201).- М., 1999
20. Не делайте один и тот же перевод дважды - статья о машинном переводе и технологии Translation Memory (TM) Prieiga per internetą:
http://langust.ru/etc/twice_tr.shtml
21. Перевод. Prieiga per internetą: <http://langust.ru/trans.shtml>
22. Лаборатория компьютерной лингвистики. Prieiga per internetą:
<http://proling.iitp.ru/Deco/>
23. Трансинтер: все о переводах, все о переводах. Prieiga per internetą:
<http://www.transinter.ru/>
24. Translation tools for Ms Word. Prieiga per internetą: <http://www.wordfisher.com/>
25. InfoArt News Agency. Ведутся работы по созданию универсального компьютерного языка. Prieiga per internetą: http://scripts.online.ru/it/news/98/11/27_083.htm
26. The Universal Networking Language (UNL). Specifications. Version 3 Edition 2. Prieiga per internetą: <http://www.unlc.undl.org/unlsys/unl/UNL%20Specifications.htm>
27. Technical translation company. Translation Memory. Prieiga per internetą:
http://www.neotech.ru/translation_memory/
28. Машинный перевод и Translation Memory для профессионалов. Prieiga per internetą:
<http://www.translationmemory.ru/>
29. Технология Translation Memory (TM). Prieiga per internetą:
<http://www.e-prompt.ru/catalog.asp?lang=ru&id=121&mode=full>
30. http://www.ci.ru/inform12_00/p05stf.htm
31. Диссертация А.Сокирко "Семантические словари в автоматической обработке текста (по материалам системы ДИАЛИНГ)". Prieiga per internetą:
<http://www.aot.ru/docs/sokirko/sokirko-candid-2.html#2-1>
32. Institute for Information Transmission Problems, Russian Academy of Sciences, Bol'shoi Karetnyj Pereulok 19, Moscow, 101447, Russia
33. Internetinis puslapis „Viskas apie vertimą“. Prieiga per internetą:
<http://www.translatorsworld.com/links1.htm>
34. Translation Memory Tools, Computer Aided Translation, Other Tools. Prieiga per internetą:
<http://www.translatum.gr/dics/translation-memory.htm>

8. CONCEPTUAL MODEL OF TRANSLATION-MEMORY-TECHNOLOGIE AND USE IN THE COMPUTERIZED SYSTEMS OF TRANSLATIONS.

Summary

„A translation memory is a multilingual text archive containing (segmented, aligned, parsed and classified) multilingual texts, allowing storage and retrieval of aligned multilingual text segments against various search conditions.”

TRANSMEM offers modern technologies for conducting and updatings of databases of the executed translations. Keeping work performed by the translator, *TRANSMEM* provides high efficiency and real profit in the future projects. The database of the executed translations releases the translator from repeated, taking away it is a lot of operating time, the translator keeps the full control over the maintenance and quality of the text. One of attractive features of system “Transmem” - the translator can work with a familiar word-processor – Open.Office.org. *TRANSMEM* compiles a linguistic database (TM) on the second plan while translators work in Open.Office.org. TM keeps the initial text and its corresponding translation. When there is a new segment (a phrase or the offer) the system immediately compares it with kept in TM segments. If identical or similar initial the segment is, it is displayed together with translation. Words and phrases which differ from the kept text, are highlighted by system. The translator makes a decision: to edit, reject or accept translation. Each change or new translation are kept in TM. As a result, the same offer should be never translated manually twice.

In the sixth part software’s “Transmem” system documentation and users manual is represented. A copy of software “Transmem” was written on the CD.

9. TERMINŲ IR SANTRUMPŲ ŽODYNAS

1. KONCEPTAS – nepriklausanti nuo konkrečios kalbos reikšmė, atitinkanti realią arba abstrakčią esmę, požymį, veiksmą arba kitą elementą, atspindintį ryšį tarp kitų reikšmių;
2. TERMINAS – tam tikros kalbos žodis arba žodžių junginys, reiškiantis ta kalba konkretų konceptą;
3. TERMINOLOGIJA – daugelis skirtingų kalbų terminų, reiškiančių tą patį konceptą;
4. SEGMENTAS – nenutrūkstamas teksto fragmentas, susidedantis iš vienos kalbos terminų, reiškiančių pagal tam tikrą kriterijų sujungtų konceptų grupę;
5. SEGMENTO VARIANTAS – segmentas, panašus į pirminį pagal tam tikrą kriterijų;
6. PIRMINĖ KALBA – kalba, iš kurios verčiama;
7. TIKSLINĖ KALBA – kalba, į kurią verčiama;
8. KALBŲ PORA – paeiliui sudėta segmentų pora, paskalbtų vertėju ekvivalentinės pagal esmę, pirma iš kurių turi terminus pirminėje kalboje, o antra – tikslinėje;
9. TM – Translation Memory (Vertimo atmintis);
10. DB – duomenų bazė;
11. IT – informacinės technologijos;
12. MS – Microsoft kompanija;
13. Nr. – numeris;
14. p. – puslapis;
15. pav. – paveikslėlis;
16. PĮ – programinė įranga.

10. PRIEDAI

10.1 CD diskas

CD diske yra pateikta autoriaus sukurta PĮ „*Transmem*“ bei jai reikalingos paleidimo bibliotekos.

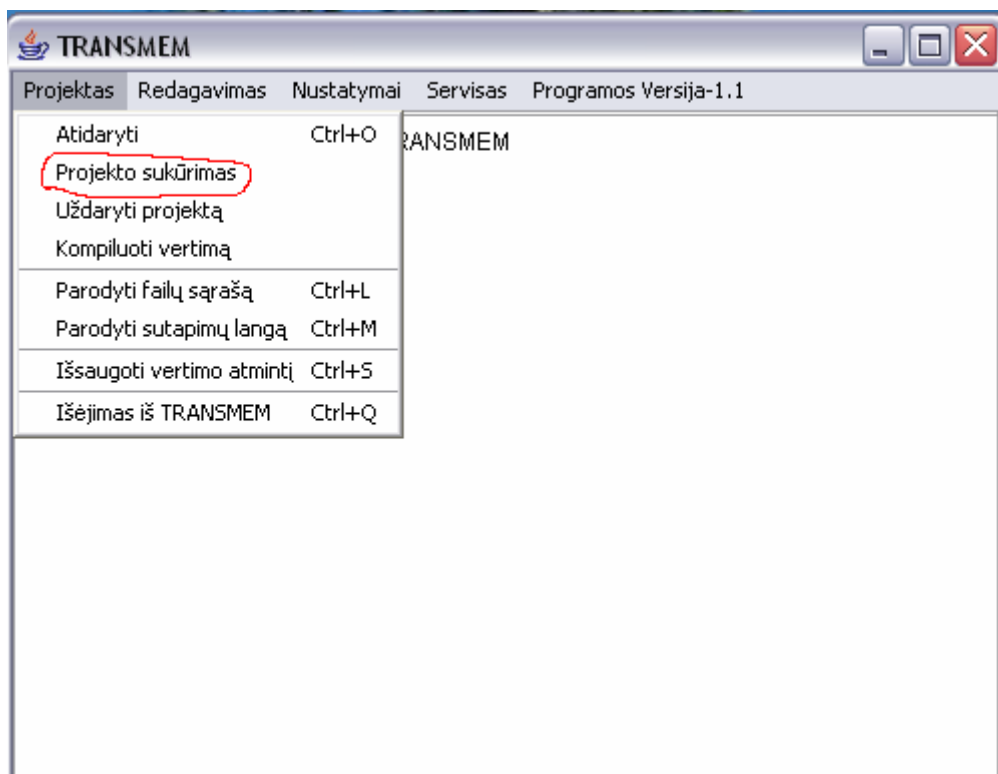
10.2 Sistemos vartotojo vadovas

Programos „*Transmem*“ darbui reikalinga Java Runtime Environment (Java išpildymo terpė).

10.2.1. Pradedant darbą – projekto pavyzdys

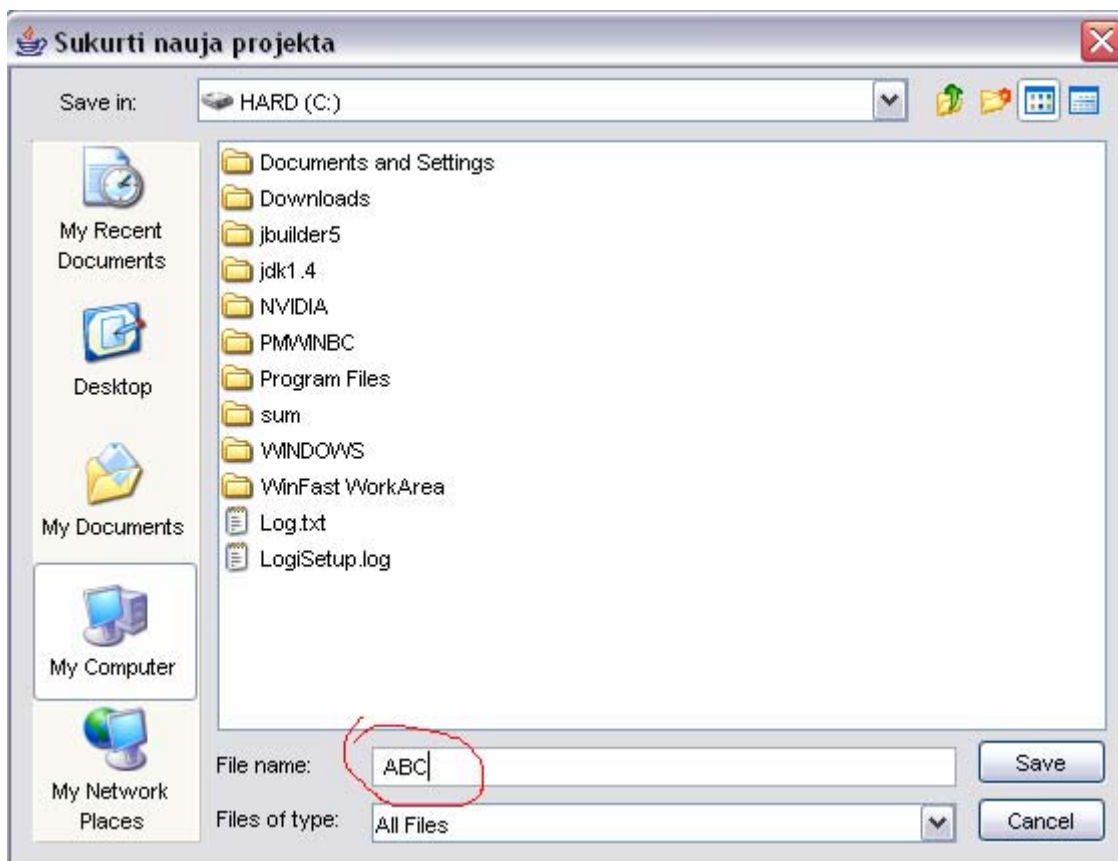
Nurodyta žemiau procedūra tarnauja kaip paaiškinimas, nebūtina griežtai laikytis jos praktikoje.

1. Paleiskite programą „*Transmem*“.
2. Sukurkite naują projektą, pasirinkę meniu *Projektas > Projekto sukūrimas*.



3. Atsiras navigacinis dialogas. Tame dialoge išsirinkite tinkamą direktoriją projektui patalpinti.
4. Lauke Failo vardas, įveskite projekto vardą (pavyzdžiui, ABC) ir patvirtinkite.

5. Iškviečiamas kitas dialogas, bet šį kartą su projekto direktorių perskaičiavimu. Tame dialoge programa „Transmem“ siūlo sukurti naują direktorią su vardu, išrinktu vartotojo (pavyzdžiui, ABC), ir penkias pagalbines direktorijas joje. Tų direktorių buvimo vieta, nurodyta pagal nutylėjimą, bet kaip pavyzdį, patvirtinkite nutylėjimą. (Smulki informacija apie failus ir projekto direktorijas – *Projekto failai ir direktorijos*.)



6. Įveskite kalbos kodus ir išeinamosios bei tikslinės kalbos lokales, pavyzdžiui, DE-CH vokiečių kalbai (Šveicarija), EN-UK anglų kalbai (Didžioji Britanija). Kodai neturi ypatingos reikšmės, jei tik numatomas pasikeitimas vertimų atminties failais tarp klientų arba kolegų, dėl ko reikia kruopščiai vadovautis reikalaujamu standartu (dažniausiai ISO).

Sukurti naują projektą

Jei norite, čia Jus galite nurodyti nestandartinius katalogus projekto failams.

Katalogas su pirminiais failais:

C:\Documents and Settings\Gocu\My Documents\Magistrovskaja_New\Proektas1\Pirmas\source\

Išverstų failų katalogas:

C:\Documents and Settings\Gocu\My Documents\Magistrovskaja_New\Proektas1\Pirmas\target\

Glosarijų katalogas:

C:\Documents and Settings\Gocu\My Documents\Magistrovskaja_New\Proektas1\Pirmas\glossary\

Vertimų atminties failų katalogas (TMX failų):

C:\Documents and Settings\Gocu\My Documents\Magistrovskaja_New\Proektas1\Pirmas\tm\

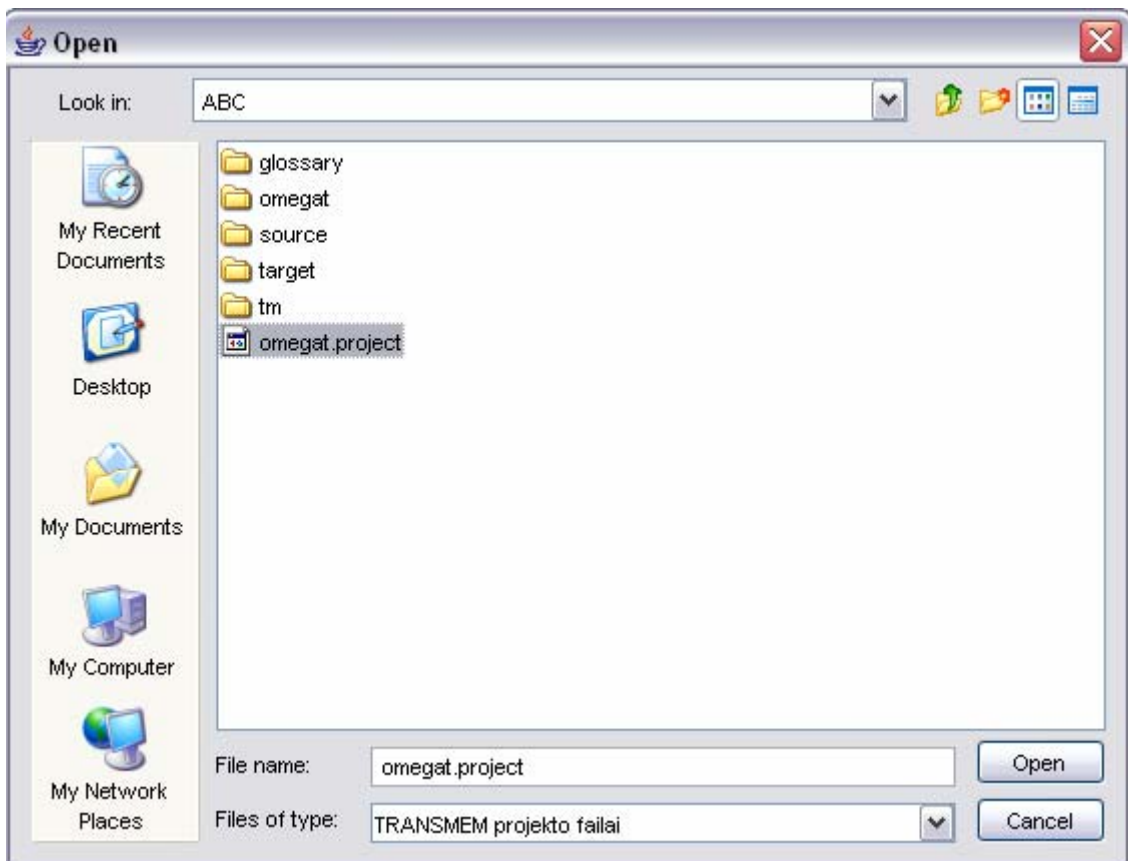
Pirminė kalba (pvz. EN-US, EN, LT, DE, ir t.t.)

LT

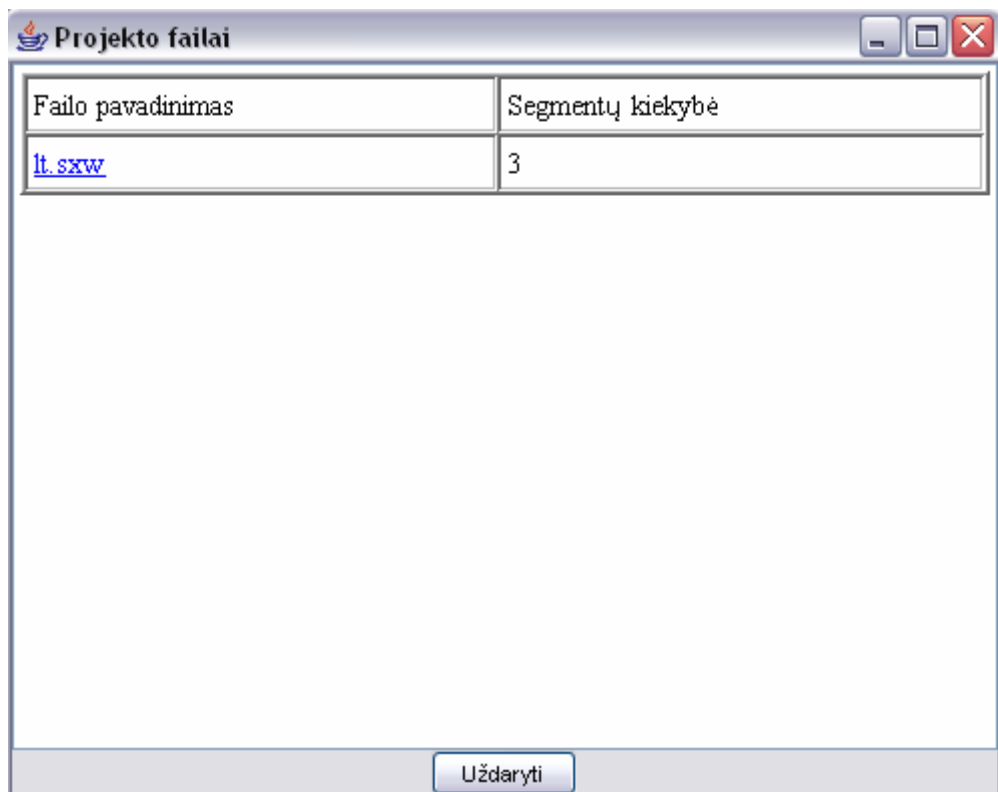
Tikslinė kalba (pvz. LT-LT, LT, RU ir t.t.)

EN-UK

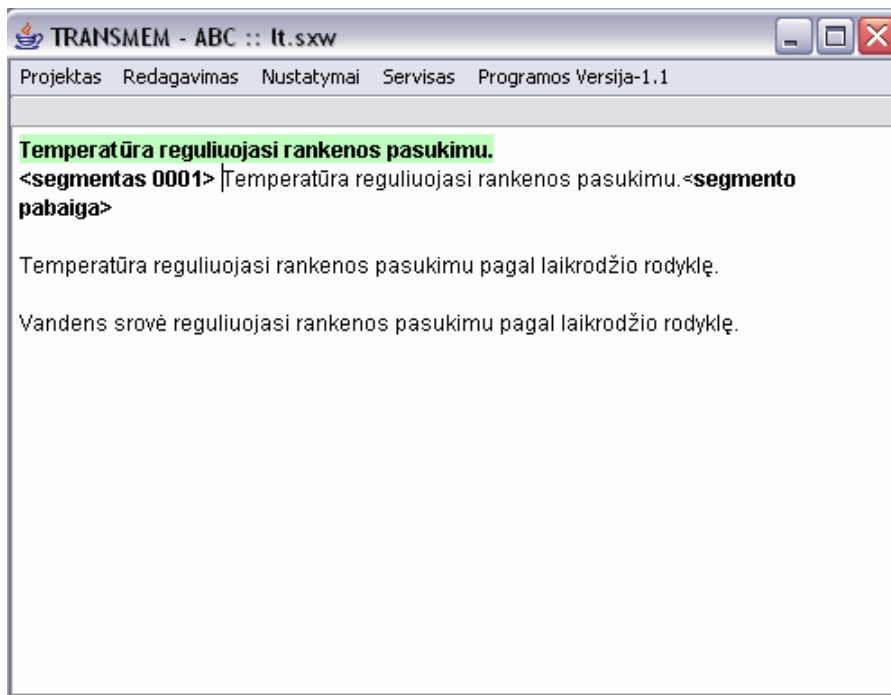
7. Programa „Transmem“ dabar sukurs pagrindinę direktoriją (taip pat vadinamą „pagrindinę projekto direktoriją“) su anksčiau pasirinktu vardu (pavyzdžiui, ABC). Projekto direktorijoje yra penkios antrinės direktorijos: /glossary, /source, /transmem, /target ir /tm.
8. Raskite failą vertimui. Išsaugokite jį kaip tekstinį failą su praplatinimu *.txt ir patalpinkite į /source.
9. Atidarykite sukurtą projektą per *Projektas* > *Atidaryti*. Programa „Transmem“ automatiškai kreipiasi į direktoriją, atidarytą jos praeitą kartą. Pagrindinė projekto direktorija atsispindi su *Transmem* ženkliuku, nurodančiu, kad tai – projektas.
10. Atidarykite projektą, iš pradžių išsirinę projekto direktoriją, paženklinatą ženkliuku (dvigubu pelės paspaudimu arba išskyre ir paspaudę „atidaryti“), po to failą *omegat.project*. Nebandykite atidaryti kito failo, pavyzdžiui, išeinamą teksto failą – nepavyks!



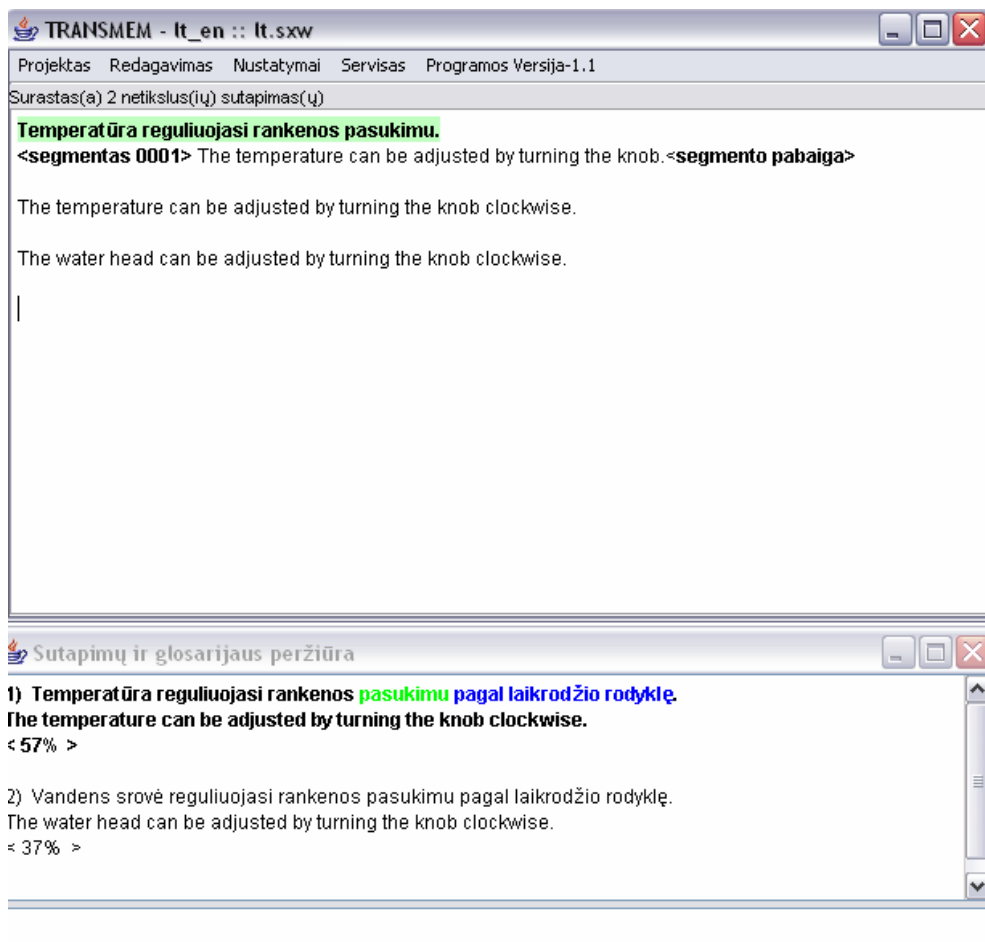
11. Programa „Transmem“ atidarys projektą ir parodys failo tekstą pagrindiniame programos „Transmem“ lauke.



12. Patalpinkite kursorą pirmo segmento tiksliniame lauke, tarp <segmentas0001> ir <segmento pabaiga>. Įveskite vertimą bei ištrinkite originalų tekstą. Paspauskite *Enter* patvirtinimui.



13. Pakartokite šią procedūrą kiekvienam segmentui, kol nebus išverstas visas tekstas.
14. Dabar išskirkite *Projektas > Išsaugoti vertimo atminti*, po to *Projektas > Kompiliuoti vertimą*. Po to galima uždaryti programą „Transmem“ (*Projektas > Išėjimas iš TRANSMEM*).
15. Failas su vertimu yra direktorijoje /target



Projekto sukūrimo langas:

Sukurti naują projektą

Jei norite, čia Jus galite nurodyti nestandartinius katalogus projekto failams.

Katalogas su pirminiais failais:

C:\Documents and Settings\Gocu\My Documents\Magistrovskaja_New\Proektas1\lt_ru\source\

Išverstų failų katalogas:

C:\Documents and Settings\Gocu\My Documents\Magistrovskaja_New\Proektas1\lt_ru\target\

Glosarijų katalogas:

C:\Documents and Settings\Gocu\My Documents\Magistrovskaja_New\Proektas1\lt_ru\glossary\

Vertimų atminties failų katalogas (TMX failų):

C:\Documents and Settings\Gocu\My Documents\Magistrovskaja_New\Proektas1\lt_ru\tml

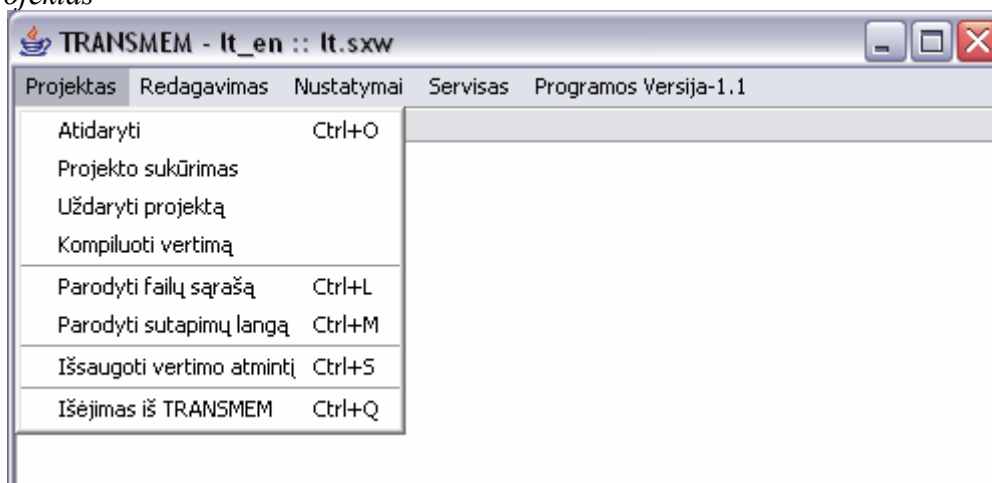
Pirminė kalba (pvz. EN-US, EN, LT, DE, ir t.t.)

LT

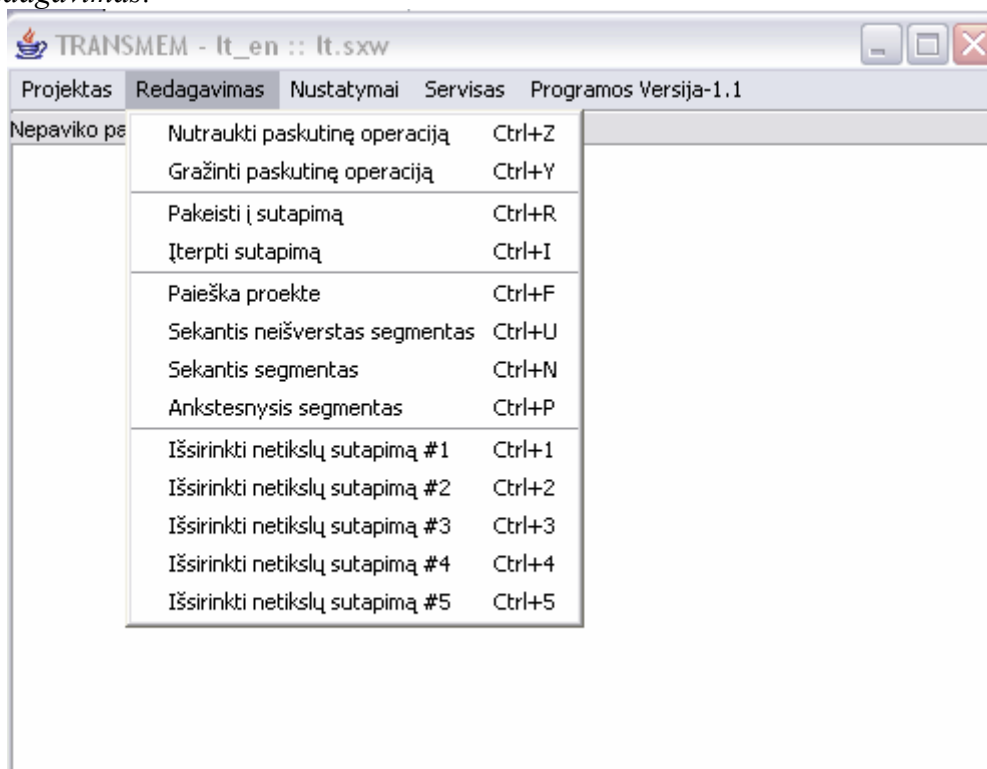
Tikslinė kalba (pvz. LT-LT, LT, RU ir t.t.)

EN-UK

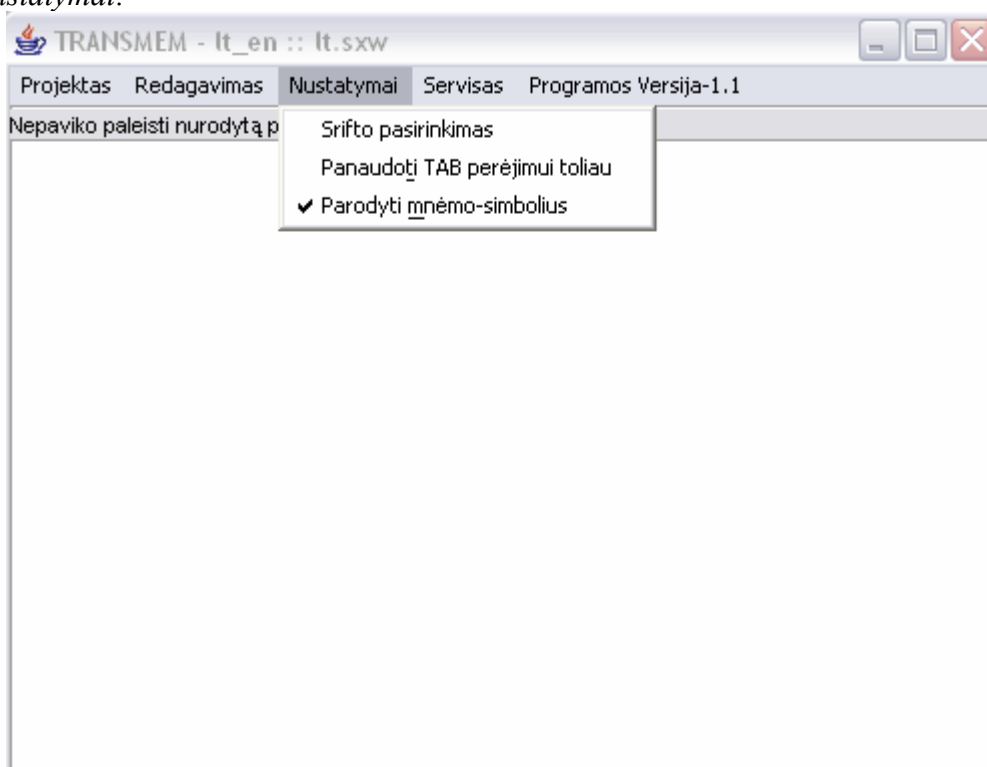
Meniu Projektas“



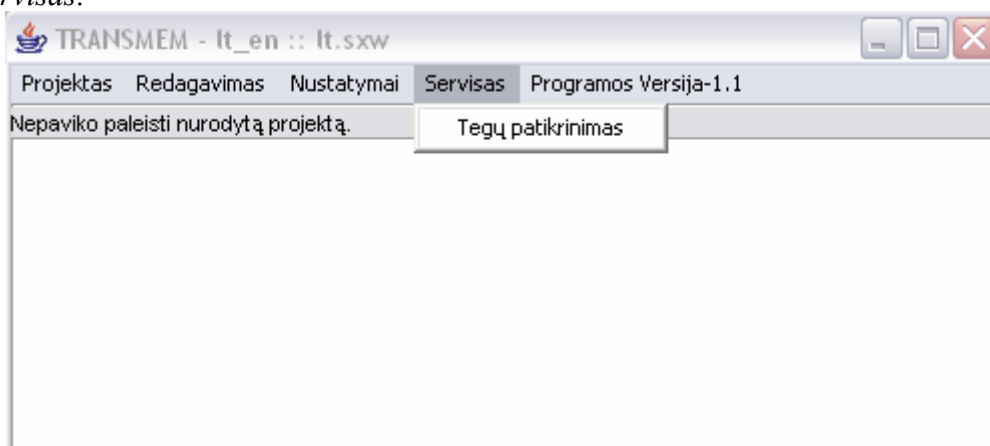
Meniu Redagavimas:



Meniu Nustatymai:



Meniu Servisas:



Meniu Programos versija:

