

KOMPIUTERINĖ LINGVISTIKA/ COMPUTATIONAL LINGUISTICS

Programinės įrangos testavimas internacionalizavimo požiūriu

Gintautas Grigas

Anotacija. Klaidos, matomos lokalizuotoje programoje, skirstomos į dvi grupes: lokalizavimo ir internacionalizavimo. Lokalizavimo klaidas daro ir gali ištaisyti lokalizuotojas. Internacionalizavimo klaidas daro ir gali ištaisyti programos autorius.

Straipsnyje nagrinėjama, kaip skirti šias dvi klaidų rūšis lokalizuotoje programoje arba kaip rasti internacionalizavimo klaidų originalioje (dar nelokalizuotoje) programoje, kad būtų galima apie tai informuoti originalo autorių ir jis spėtų laiku ištaisyti tokias klaidas. Taip atsirastų reali galimybė pasiekti lokalizuotos programos taisyklingos kalbos. Nagrinėjama kaip ir kada klaidų paieškai ir identifikavimui tikslinga panaudoti lokalizavimo išteklių analizę, pseudovertimą (pseudolokalizavimą), bandomąjį abejotinų programos vietų lokalizavimą, eksperimentavimą su veikiančia programa.

Remiamasi dešimties dažniausiai naudojamų lokalizuotų programų klaidų analize. Kartu netiesiogiai perteikiama ir internacionalizavimo klaidų paieškos patirtis.

Reikšminiai žodžiai: internacionalizavimas, lokalizavimas, testavimas, programinė įranga, lokalizavimo klaida, internacionalizavimo klaida.

Įvadas

Lokalizuotose programose pasitaiko nemažai klaidų: neišverstų žodžių arba frazių, neteisingai įvardytų programų komponentų arba veiksmų, neteisingų žodžių gramatinių formų, netelpančių frazių į joms skirtą vietą, neteisingų skaičių, datų, laiko ir kitų formatų, ribojimų vartoti kai kurias abėcėlės raides ir kt.

Klaidos menkina lokalizuotų programų kokybę. Dėl jų daugelis programų naudotojų, net gerai nemokėdami užsienio (konkrečiau – anglų) kalbos, kuria parašyta didžioji dalis programų, neskuba pereiti prie lokalizuotų programų.

Klaidos lokalizuotose programose skirstomos į dvi grupes: lokalizavimo klaidos ir internacionalizavimo klaidos. Šios ir su jomis susijusios kitos svarbiausios sąvokos tiksliai apibūdintos Tatjanos Jevsikovas disertacijoje (Jevsikova, 2009), kurioje išsamiai nagrinėjamas lokalizavimo ir internacionalizavimo santykis.

Internacionalizacija – programinės įrangos projektavimo proceso, pritaikant ją įvairioms kalboms ir kultūroms, rezultatas.

Internacionalizavimas – programinės įrangos ir jos duomenų struktūrų projektavimas taip, kad visa tai būtų galima lengvai adaptuoti įvairioms kalboms ir kultūroms.

Internacionalizavimo klaida – programinės įrangos projektavimo klaida, kai tam tikra jos savybė arba parametras nenumatomas įvairioms lokalėms pritaikyti.

Išteklių eilutė – programos tekstinių lokalizuojamųjų išteklių elementas – ženklų seka, skirta rodyti ekrane (pvz., meniu punkto, komandos, klaidos pranešimo tekstas) arba nu-

rodyti tam tikrą lokalizuojamos programos nuostatą (pvz., koduotę, šriftą).

Lokalizacija – programinės įrangos adaptavimo kultūrinei ir kalbinei aplinkai proceso rezultatas.

Lokalizavimas – programinės įrangos pritaikymas tam tikrai kalbinei ir kultūrinei aplinkai.

Lokalizavimo klaida – programinės įrangos kurios nors savybės neatitikimas lokalei, kai ta savybė yra internacionalizuota.

Lokalizuojamieji ištekliai – programos ištekliai, į kuriuos sudėti duomenys, priklausomi nuo kalbos ir kultūros;

Parametruotoji eilutė – išteklių eilutė, kurioje yra pavar-totas parametras – reikšmė, įterpiama į eilutę programą vyk-dant.

Lokalizavimo klaidas daro lokalizuotojas. Jas gali (ir pri-valo) ištaisyti lokalizuotojas. Jų taisymas nėra sudėtingas. Dažniausiai pakanka pataisyti lokalizuojamųjų išteklių eilutę, todėl darbas vyksta operatyviai.

Internacionalizavimo klaidos yra programos originale. Jas daro ir gali ištaisyti programos autorius. Jų taisymas sudėtingesnis ir ilgiau užtrunka, nes paprastai reikia pakore-guoti programos vykdomąją dalį, dažniausiai ją kuo nors papildyti.

Svarbu kiek galima anksčiau aptikti internacionalizavimo klaidas ir apie jas informuoti originalo autorių, kad jis turėtų daugiau laiko pastebėtoms klaidoms ištaisyti arba bent įvertinti, kaip nepataisytos klaidos atsilieps lokalizacijos kokybei. Jeigu yra rimtų klaidų ir jų laiku ištaisyti nesutinka originalo autorius, tai gal tos programos lokalizavimo nevertėtų pradėti.

Stebėdami veikiančią lokalizuotą programą matome visas joje esančias klaidas, bet dažnai negalime atskirti, kuri yra internacionalizavimo, kuri lokalizavimo klaida. Kartais jos visos vadinamos lokalizavimo klaidomis. Tokiu atveju internacionalizavimo klaidos gali būti laikomos lokalizavimo klaidų poaibių arba tik originalo klaidomis, esančios lokalizavimo klaidų priežastimi. Kadangi internacionalizacijos ir lokalizacijos klaidų taisymas iš esmės skiriasi, tai svarbu aiškiai ir griežtai skirti šias dvi klaidų grupes. Patarčiau racionaliau laikytis griežtesnės nuostatos, jog lokalizacijoje matomos dvi nesikertančios klaidų aibės, o ne aibė ir poaibis.

Lokalizavimo ir internacionalizavimo situacija

O'Sullivan (2001) pateikė šešis lokalizavimo dėsnius, kuriuos būtų tiksliau laikyti reikalavimais originalo internacionalizacijai:

1. Pirminis programos tekstas ir naudotojo sąsaja turi būti visiškai atskirti vienas nuo kito.
2. Naudotojo sąsajos eilučių valdymas programiniu būdu turi būti ribojamas.
3. Pirminis programos tekstas ir sąsaja turi būti kompiuojami atskirai.
4. Visi naudotojo sąsajos tekstai turi būti pateikiami su jų naudojimo kontekstu.
5. Eilučių vertimas ir tolesnis atnaujinimas turi būti atliekamas „ką matau tą gaunu“ aplinkoje.
6. Naudotojo grafinės sąsajos komponentų dydžių keitimas ir išdėstymas turi būti vykdomas automatiškai, programai veikiant.

Programuotojų visuomenė dar nėra įsisąmoninusi internacionalizavimo klausimų. Pagrindinių programavimo priemonių – programavimo kalbų ir transliatorių internacionalizavimo ir lokalizavimo lygis atsilieka nuo bendros paskirties programų lygio (Laucius, 2007). Jis turėtų būti ne žemesnis už pastarąjį, nes darbo priemonės kokybė persiduoda gaminio kokybei.

Fizinis lokalizuojamų išteklių atskyrimas nuo vykdomosios programos dalies ne visada būna pakankamas (Safar, 2010; Hogan, 2003). Iki visos kontekstinės informacijos pateikimo dar toli (Safar, 2010). Kol kas yra tik epizodinių darbų teorinis lygmuo (Jevsikova, 2009).

Pirmajame šio amžiaus dešimtmetyje didesnis dėmesys buvo skiriamas lokalizacijoms testuoti bei testavimui automatizuoti (Desikan, 2009). Tai natūralu, nes tik lokalizavus programą išryškėja visos klaidos ir atsiranda galimybė taisyti lokalizavimo klaidas.

Internationalizavimo klaidas taisyti lokalizuotoje programoje jau per vėlu, nes lokalizuotojas jų ištaisyti negali, o originalios programos autorius jas ištaisyti geriausia atveju tik naujoje programos laidoje. Tai mažina lokalizuotojo motyvaciją informuoti programos autorių apie klaidas. Dėl to tos pačios klaidos keliauja iš vienos programos laidos į kitą. Susidaro nuomonė, kad kompiuteris kitaip (t. y. kalbiškai teisingai!) ir negali veikti ir kalba turi taikytis prie kompiuterio. Iš tiesų, informacinė technika turi įtakos kal-

bai, bet kalbą taikyti prie internacionalizavimo klaidų nėra pagrindo.

Tos pačios internacionalizavimo klaidos atsiliepia daugeliui kalbų, į kurias verčiamos programos. Jeigu klaida bus aptikta ir ištaisyta prieš lokalizavimą, tai nebereikės jos ieškoti ir rūpintis jos taisymu kiekvienos naujos kalbos lokalizuotojui (Safar, 2010).

Internationalizavimo klaidų taisymo sąnaudos nėra didelės. Jos gali atsipirkti net ir vienos kalbos atveju, jeigu lokalizuojamos kelios tos pačios programos laidos (Dagienė, 2004).

Šiame darbe bus nagrinėjamas internacionalizavimo klaidų atskyrimas nuo lokalizavimo klaidų ir jų ankstyva paieška prieš pradedant lokalizavimą ar pačioje lokalizavimo pradžioje. Tai turėtų pagerinti lokalizuotų programų kokybę ir sumažinti iš svetur per kompiuterius ateinančią neigiamą kitų kalbų įtaką lietuvių kalbai.

Testavimas

Klaidos randamos testuojant programą. Testavimas automatizuojamas, tačiau automatinis testavimas dar ne visada duoda apčiuopiamą naudą, nes jam reikia paruošti daug duomenų. Sąnaudos jiems paruošti kartais gali prilygti rankinio testavimo sąnaudoms (Safar, 2010).

Dažniausiai automatizuojami atskiri rankinio testavimo darbai. Daugiausia naudos duoda išteklių eilučių pseudovertimas. Išteklių eilutės neverčiamos, o tik numeruojamos ir jų numeriai prirašomi jų pradžioje. Vykdamas programą ir stebint ekrane pasirodančius numeruotus užrašus galima pamatyti kurios eilutės nepateko į lokalizuojamuosius išteklius (jos neturės numerių), pastebėti ekrane matomas frazes, formuojamas iš kelių išteklių eilučių ir dėl to atsirandančias žodžių gramatinių formų nesuderinamumo ir kitas „užprogramuotas“ kalbos klaidas.

Toks testavimas gali būti laikomas pradiniu lokalizavimo darbu (Handerson, 2003). Kartais jį naudoja ir originalo testuotojai norėdami patikrinti, ar įvairių kalbų savitieji rašto ženklai bus teisingai rodomi ekrane. Originalios išteklių eilutės pakeičiamos eilutėmis, sudarytomis vien iš savitųjų ženklų. Turinys nesvarbu, todėl eilutės gali būti ir vienodos.

Testavime susiduriama su inžinerijos (koduotės, formatai, ir pan.) ir kalbos (vertimas, redagavimas) problemomis. Testuotojas turėtų būti plataus profilio specialistas. Palaipsniui šios dvi sritys atsiskiria (Safar, 2010). Svarbiausia, kad filologas susikalbėtų su informatiku. Be to, inžinerinė testavimo dalis daugiau pasiduoda automatizavimui. Siekiant palengvinti kalbinį testavimą pirmiausia naudinga atlikti inžinerinį testavimą.

Testavimo eksperimentai

Siekiant surinkti informaciją apie dažniau pasitaikančias klaidas buvo testuojamos plačiai naudojamos lokalizuotos programos (*Microsoft Office 2003*, *OpenOffice.org*, *Adobe Reader*, *Skype*, *Total Commander*, *Picasa*, *Opera*, *Mozilla Firefox*, *Mozilla Thunderbird*, *7-zip*). Buvo parengtas bendras klausimynas su 80 klausimų apie daugiau tikėtinas visas klaidas programose, t. y. neskirstant jų į lokalizavimo

ir internacionalizavimo. Klaidų buvo ieškoma vykdant programas lokalizuotų operacinių sistemų „Windows XP“ arba „Windows Vista“ aplinkoje kompiuteriuose su lietuviška standartinė klaviatūra (LST 1582).

Vėliau buvo analizuojamos klaidų priežastys ir pagal tai klaidos suskirstytos į internacionalizavimo ir lokalizavimo klaidas. Iš bendro klausimyno buvo atrinktos dažniau pasitaikančios klaidos ir iš jų sudarytas internacionalizavimo klaidų klausimynas, turintis 60 klausimų. Tokia didelė šioms klaidoms tinkanti klausimų dalis gali būti paaiškinta taip:

1. Kai kurios klaidos gali būti priskirtos abiem klaidų grupėms. Pavyzdžiui, lokalizuotoje programoje matomas užrašas „Lentelė N“ (o ne „N lentelė“) gali būti neteisingas originalios eilutės „Table %N“ (čia procento ženklų pažymėtas parametras – lentelės numeris) vertimas (lokalizavimo klaida) arba negalėjo būti teisingai išversta, jeigu lokalizuotiniuose ištekluose buvo tik eilutė „Table“, o numerio vieta įkoduota į programos vykdomąją dalį (internationalizavimo klaida).

2. Klausimyne nebuvo detalizuotos redagavimo, stiliaus ir kitos kalbos klaidos, vienareikšmiškai priskirtinos prie lokalizavimo klaidų.

Išanalizavus dešimtį programų nerasta nė vienos, kurioje nebūtų internacionalizavimo klaidų (ir dažnai – šiurkščių). Daugelis jų kartojasi keliose programos laidose. Tai galima paaiškinti tuo, kad internacionalizavimo klaidos nebuvo pastebėtos prieš lokalizavimą ar bent jo pradžioje. Apie vėliau pastebėtas klaidas lokalizuotojai ne visada praneša originalo autoriams, žinodami, kad originalo autoriai nebespės jų ištaisyti. Juolab, kad lokalizuotojas gali pasiteisinti, jog klaidos ne dėl jo kaltės. Dalis klaidų lieka išvis nepastebėtų arba nenustatytą jų priklausomybę internacionalizavimo klaidų grupei.

Internationalizavimo klaidų gali patekti į originalą ir iš kitų jame panaudotų programų (programų komponentų). Jas gali ištaisyti tik tų programų (komponentų) autoriai, t. y. trečiosios šalys.

Tiek originalo, tiek trečiųjų šalių autoriai dažnai nepastebi internacionalizavimo klaidų dėl to, kad dauguma jų tiesiogiai nėra matomos toje kalbos ir kultūros aplinkoje, kurioje kuriamas originalas, o jų autoriai dirba su jų pačių kuriamomis originaliomis (ne lokalizuotomis) programomis, neturi pakankamai informacijos apie kalbas, į kurias bus lokalizuojama programa.

Prie trečiųjų šalių programų priskirtinos ir tos, kurios sudaro lokalizuotos programos aplinką, pavyzdžiui, operacinė sistema (pati svarbiausia!), „Javos“ aplinka ir kt. Testuojant jas reikia naudoti lokalizuotas, su parinktomis lietuvių kalbai būdingomis nuostatomis, kad jų klaidos, patekusios į testuojamos programos langą, nebūtų palaikytos testuojamos programos klaidomis.

Nemažą internacionalizavimo klaidų dalį galima aptikti dar prieš programos lokalizavimą. Klaida atsiranda dėl to, kad kuris nors lokalizuotinas elementas (tekstas, iliustracija, koduotės pavadinimas ir pan.) nėra iškeltas į lokalizuojamuosius išteklis.

Tekstus, kurie matomi kompiuterio ekrane, bet neįtraukti į lokalizuojamuosius išteklis, galima aptikti automatizuotu būdu atliekant tų išteklių pseudovertimą ir po to vizualiai stebint ar ekrane matomos eilutės turi pseudovertimo metu pridėtas žymes (eilučių numerius).

Pagrindiniai lokalės elementai

Lokalė – tai kompiuteryje ir jo programinėje įrangoje naudojamų elementų, priklausančių nuo kalbos ir kultūros normų, visuma. Kiekviena kalba turi savo lokalę, o kartais ir kelias, jeigu ji vartojama keliose valstybėse, pvz., Prancūzijos prancūzų ir Kanados prancūzų. Jeigu tie elementai naudojami lokalizuotoje programoje, tai jie turi atitikti lokalę.

Dalis lokalės elementų yra tokie svarbūs, kad be jų programos naudojimas neturėtų prasmės. Tokie elementai įtraukiami ir į nelokalizuotas programas. Juos vadinsime pagrindiniais lokalės elementais. Tai kalbos abėcėlė, raidžių eilė abėcėlėje, kiti svarbiausi ženklai, koduotės, skaičių, datų, laiko formatai, matavimo vienetai. Jų aibė nėra apibrėžta ir didėja tobulėjant kompiuteriams bei plintant jų naudojimui įvairiose gyvenimo srityse.

Pagrindiniai lokalės elementai įtraukiami ir į nelokalizuotas programas, visų pirma į operacines sistemas, skirtas platinti įvairiose valstybėse. Kadangi operacinė sistema yra kiekviename kompiuteryje ir daugelyje kitų universalių skaitmeninių įrenginių (pvz., mobiliajame telefone), tai kitos juose veikiančios programos lokalių elementus paprastai pasiima iš operacinių sistemų. Jeigu testuojant originalią programą aptinkamas neatitikimas pagrindiniam lokalės elementui, tai jį reikia laikyti internacionalizavimo klaida.

Klausimynas

Siekiant naujų programų testuotojams suteikti galimybę rasti daugiau internacionalizavimo klaidų ir paspartinti jų paiešką, buvo parengtas klausimynas.

Toliau pateikiamos į tris grupes suskirstytos klaidos. Naudojantis klausimynu lengviau tikrinti internacionalizavimo klaidas. Klausimai – tai užuominos kur ir ką reikia tikrinti.

Rašto ženklai

Visavertis visų lietuviškų raidžių vartojimas visuose kompiuterio ruošiamuose arba tvarkomuose dokumentuose ir kompiuterio ekrane matomuose tekstuose yra esminis dalykas, todėl jį reikia išsamiai patikrinti.

Lietuvos standartais įteisintos dvi aštuonių bitų koduotės: ISO 8859-13 ir „Windows-1257“. Nors jos skirtingos, visų lietuvių kalbos abėcėlės raidžių kodai nesiskiria, todėl problemų jas vartojant nėra.

Lietuvių kalboje vartojamos kabutės („“) taip pat yra abiejose koduotėse, tačiau jų kodai skiriasi. Todėl ten, kur kartu su tekstu nurodoma ir jo koduotė (pvz., elektroniniuose laiškuose) turi būti numatytas automatinis perkodavimas.

ISO 8859-13 koduotėje nėra brūkšnio. Tekstą perkoduojant į šią koduotę brūkšnys paprastai keičiamas vienu brūkšneliu arba dviem brūkšneliais.

Mobiliuosiuose telefonuose tekstinės žinutės koduojamos unikodu, tačiau čia ženklų aibė ribojama. ETSI ES 202 130 standartas nustato, kad lietuviškose žinutėse privalomos visos 32 lietuvių kalbos abėcėlės raidės ir rekomenduojamos dar trys anglų kalbos abėcėlės raidės *q*, *w* ir *x*. Lietuviškos kabutės ir brūkšnys į privalomų ženklų aibę neįtrauktos.

Klaidos, susijusios su savitųjų lietuviškų raidžių (*qčėėjšųūž*) vartojimu dokumentuose (rašiniuose, tinklalapiuose, elektroniniuose laiškuose, pokalbių programų žinutėse), greitai pastebimos, ir kompiuterių programose jų retai pasitaiko, bet rečiau matomose kompiuterio ekrano vietose jos dar dažnos. Klausimyne joms skirta daugiau dėmesio.

Visų lietuvių kalbos abėcėlės raidžių vartojimą visur, išskyrus abonentą vardą, rašomą elektroninio pašto adrese prieš ženklą *@*, reglamentuoja Lietuvos ir tarptautiniai standartai. Jiems realizuoti nėra techninių kliūčių, todėl bet koks nesklandumas laikytinas internacionalizavimo klaida.

Toliau pateikiami klausimai testuotojams, susiję su rašto ženklais, taikytini programoms, turinčioms klausimuose minimas funkcijas:

1. Ar galima surinkti visas lietuvių kalbos abėcėlės raides, kabutes, brūkšnį, ir ar visi šie ženklai teisingai rodomi kompiuteriu rengiamame, skaitomame arba modifikuojamame rašinyje, tinklalapyje, tinklalapio pavadinime, elektroniniame laiške, žinutėje ir bet kuriame kitame dokumente ar jo dalyje?
2. Ar galima surinkti visas lietuvių kalbos abėcėlės raides, kabutes ir brūkšnį tinklalapio laukuose (pvz., komentare, anketoje), kuriuos pildo tinklalapio lankytojas, ar juose surinktas tekstas teisingai išsiunčiamas į tinklalapį?
3. Ar galima surinkti visas lietuvių kalbos abėcėlės raides, kabutes, brūkšnį ir ar šie ženklai teisingai rodomi tekstui skirtuose dialogo langų laukuose?
4. Ar teisingai persiunčiami laiškai, žinutės ir kiti tekstiniai pranešimai tarp pašto programų, mobiliųjų telefonų ir kitokių įrenginių?
5. Ar teisingai pavaizduojamos visos lietuvių kalbos abėcėlės raidės ir kabutės dokumente, konvertuotame į kitą formatą (iš kito formato)? (Pavyzdžiui, tarp DOC, HTML, TXT, XLS formatų).
6. Ar teisingai keičiamos mažosios raidės didžiosiomis (ir atvirkščiai)?
7. Ar teisingai rikiuojami tekstai pagal lietuvių kalbos abėcėlę: *aqbcčdeėėfghiųjklmnoprsštuiūvz*, ar raidės *aq*, *eėė*, *iųj*, *uiū* rikiuojant gali būti laikomos lygiavertėmis (bet ne *cč*, *sš*, *žž*)?
8. Ar paieškose netapatinamos raidės *q* su *a*, *č* su *c* ir pan.?
9. Ar gali būti vartojamos visos lietuvių kalbos abėcėlės raidės registracijos varduose (išskyrus abonentą vardą elektroninio pašto adrese), asmenvardžiuose ir kituose registracijos duomenyse?
10. Ar gali būti vartojamos visos lietuvių kalbos abėcėlės raidės slaptažodžiuose?
11. Ar visos lietuvių kalbos abėcėlės raidės vartojamos kontroliniuose koduose ir iš jų sudaromi lietuvių kalbai būdingi raidžių deriniai?
12. Ar galima atverti tinklalapius, kurių sričių varduose yra raidžių su diakritiniais ženklais (pvz., www.zalgiris.lt, www.ulevičius.lt) ir teisingai parodyti jų adresus?

13. Ar gali būti vartojamos visos lietuvių kalbos abėcėlės raidės failų, katalogų, aplankų varduose?

14. Ar galima į programą įkelti dokumentą, kurio failo varde vartojamos bet kurios lietuvių kalbos abėcėlės raidės?

15. Ar galima į programą įkelti dokumentą iš katalogo arba įrašyti į katalogą, kurio varde vartojamos bet kurios lietuvių kalbos abėcėlės raidės?

16. Ar pakavimo programa gali teisingai supakuoti ir išpakuoti katalogus ir failus, kurių varduose vartojamos bet kurios lietuvių kalbos abėcėlės raidės?

17. Ar programa (pokalbių, pašto ir kt.) gali persiųsti failą, kurio varde vartojamos bet kurios lietuvių kalbos abėcėlės raidės?

18. Ar galima atverti tinklalapius, kurių aplankų ir failų varduose yra raidžių su diakritiniais ženklais ir teisingai parodyti jų adresus?

19. Ar programa gali būti įdiegta į katalogą, kurio varde yra savitųjų lietuviškų raidžių?

20. Ar visuose elektroninio laiško laukuose (*Kas*, *Kam*, *Tema* ir kt.), išskyrus elektroninio pašto adresą, galima vartoti bet kurias lietuvių kalbos abėcėlės raides?

21. Ar elektroninio laiško tekstas koduojamas Lietuvos standartus atitinkančia koduote ISO 8859-13, „Windows-1257“, ISO 10646 (unikodu) ir jos pavadinimas (iso-8859-13, „Windows-1257“, UTF-8) įrašomas į elektroninio laiško antraštę?

22. Ar atsakymas į laišką koduojamas ta pačia koduote, kaip ir originalus laiškas?

23. Ar galima atverti ir teisingai rodyti bei spausdinti laišką, parašytą bet kuria Lietuvos standartus atitinkančia koduote, nurodyta laiško antraštėje?

24. Ar tekstai, įrašomi hiperteksto (HTML) formatu, koduojami Lietuvos standartus atitinkančia koduote ISO 8859-13, „Windows-1257“, ISO 10646 (unikodu) ir jos pavadinimas (iso-8859-13, „Windows-1257“, UTF-8, UTF-16) įrašomas į tinklalapio antraštę?

25. Ar galima atverti ir teisingai rodyti bei spausdinti tinklalapį, parašytą bet kuria Lietuvos standartus atitinkančia koduote?

26*. Ar raidės *Č*, *č*, *Š*, *š*, *Ž*, *ž* įrašomos į ISO 8859-13 arba „Windows-1257“ koduote koduojamą hiperteksto (HTML) dokumentą, koduojamos 8 bitų kodais, o ne skaitiniais kodais (&xxxx; pavidalu)?

27*. Ar išsiunčiamų laiškų antraščių laukai koduojami B arba Q kodu?

28*. Ar lokalizuojant programą galima keisti komandų klavišų raides?

29. Ar prieigos klavišai neblokuoja kurio nors trečiojo lygio ženklo, renkamo klaviatūra (t. y. ar galima surinkti visus trečiojo lygio ženklus)?

30. Ar komandų klavišams nenaudojami kitų ženklų klavišai, išskyrus raidžių ir skaitmenų klavišus (pvz., Vald+%)?

Pastaba. Žvaigždutėmis pažymėtoms klaidoms rasti reikia tam tikrų informatikos žinių.

Bandytose programose visos su ženklais susijusios klaidos buvo internacionalizavimo klaidos. Reikia tikėtis, kad šiai grupei priklausys ir su ženklais susijusios klaidos kitose programose. Tokių klaidų radus galima jas laikyti internacionalizavimo klaidomis.

Jeigu tikrinamas duomenų persiuntimas telekomunikacijos priemonėmis tarp skirtingų kompiuterių arba tarp pro-

gramų kompiuterio viduje, tai viename punkte – siuntimo arba gavimo, turi būti etaloninė programa (arba įrenginys), kuri tą funkciją atlieka teisingai, o kitame – testuojamoji. Antraip, aptikus klaidą, bus neaišku, kuri programa klysta.

Formatai

Dažniausiai vartojami formatai priskiriami pagrindiniams lokalės elementams. Tai – skaičių, datų, laiko matavimo vienetų formatai. Jie turėtų būti teisingi ir originalioje programoje, nustatomi pagal parinktą kalbą dažniausiai paimant jų reikšmes iš operacinėje sistemoje esančių lokalės duomenų.

Rečiau vartojami formatai (pvz., telefono numerio formatai), kurių nebūna lokalėse, turėtų būti lokalizuojamuose ištekliuose. Jeigu jie ne tokie, kokie turi būti lietuviškoje aplinkoje, dar nereikia, kad tai internacionalizacijos klaida. Reikia dar patikrinti – gal jie yra lokalizuojamuose ištekliuose arba juos galima parinkti programos nuostatose.

Visi klausimai formuluoti taip, lyg būtų testuota lokalizuota programa, ir nuspręsta, kad gavus neigiamą rezultatą, dar reikia patikrinti programos nuostatas ir lokalizuojamusius išteklius.

1. Ar skaičiai į dokumentą rašomi ir rodomi teisingu formatu: trupmenos skirtukas yra kablelis, skaitmenų grupių (tūkstančių) skirtukas – taškas arba tarpas?
2. Ar dialogo langų laukai priima teisingu formatu užrašytus skaičius, rodo ir patys juos generuoja teisingu formatu: trupmenos skirtukas yra kablelis, skaitmenų grupių (tūkstančių) skirtukas – taškas arba tarpas?
3. Ar į dokumentą automatiškai įterpiama data teisingu formatu (trumpuoju ir (arba) ilguoju)?
4. Ar dialogo langų laukai priima teisingu formatu užrašytas datas, rodo ir patys jas generuoja teisingu formatu (trumpuoju ir (arba) ilguoju)?
5. Ar į dokumentą automatiškai įterpiamas laikas teisingu 24 val. formatu?
6. Ar pirmąją savaitės dieną laikomas pirmadienis?
7. Ar dokumentų lapų formatai atitinka Lietuvoje galiojančias raštvedybos normas (A3, A4, A5 ir pan.)?
8. Ar puslapio, paraščių, įtraukų dydžiai ir kiti matmenys pateikiami metrinės matų sistemos vienetais (centimetrais, milimetrais)?
9. Ar numatytasis pastraipos pirmosios eilutės įtraukos dydis yra 0,7–0,8 cm?
10. Ar pašto adresų formos atitinka Lietuvos administracinį suskirstymą?
11. Ar pašto adresų komponentai atitinka Lietuvos administracinį suskirstymą?

Tekstai

Visi tekstai, išskyrus panaudotus formatuose (žr. skyrelį *Formatai*), originalioje programoje būna originalo kalba. Daugiausia problemų sukelia tekstai, formuojami sujungiant kelias išteklių eilutes.

Eilutės sujungiamos taip, kad jų eilės tvarka ir gramatinės žodžių formos atitiktų originalo kalbos, kuri dažniausiai būna anglų, taisyklės. Kitų kalbų taisyklės gali skirtis (taip dažniausiai ir būna). Pavyzdžiui, užrašas prie mygtuko, atšaukiančio paskutinį įterpimą, angliškoje programos ver-

sijoje būna „Undo Insert“. Lietuviškai derėtų versti „Atšaukti įterpimą“, bet ištekliuose yra tik atskiros eilutės „Undo“ ir „Insert“, kurios taip pat naudojamos ir kitose programos vietose atskirų komandų atskiriems mygtukams pažymėti. Jeigu išversime „Atšaukti“ ir „Įterpti“, tai atskiriems mygtukams tiks, bet vietoj „Undo Insert“ pasirodys užrašas „Atšaukti Įterpti“. O jei „Insert“ išversime kaip „įterpimą“, tai ir ant atskiros įterpimo komandos mygtuko (ir galbūt ne vieno) pasirodys užrašas „įterpimą“. Nei šis, nei tas. Todėl jeigu ekrane po pseudoveravimo pastebima suključiota eilutė (t. y. turinti du numerius, pvz., „25-Undo 123-Insert“), tai internacionalizavimo klaidos nebus tik tada, jeigu niekur kitur ekrane nebus eilučių, pažymėtų numeriu 123 (šiuo atveju eilutė Nr. 25 gali būti).

Panašių internacionalizavimo klaidų taisymas programos originale nėra sudėtingas – tereikia į lokalizuojamusius išteklius įtraukti atskiras eilutes su reikalingomis žodžių gramatinėmis formomis. Toks taisymas naudos gali duoti daug. Ištaisius klaidas lokalizuotoje suključytoje eilutėje ne tik neliks akivaizdžios kalbos klaidos, bet lietuviui lietuviškas užrašas bus suprantamesnis, negu anglui angliškas.

Esant sudėtingiems atvejams, naudinga išversti keletą įtartinų vietų, ir, paleidus programą, pažiūrėti, kaip tie vertimai atrodo kompiuterio ekrane.

Apskritai, lokalizavimo pradžioje naudinga išbandyti ir kitokias įtarimus sukeliančias išteklių eilutes (pvz., daugia-reikšmius angliškus terminus).

Toliau pateikiamas klausimyno fragmentas:

1. Ar visi ekrane matomi tekstai įtraukti į lokalizuojamusius išteklius?
2. Ar visi išsiunčiami tekstai (pvz., kalendoriaus programos laiškas apie būsimą įvykį) įtraukti į lokalizuojamusius išteklius?
3. Ar puslapių, lentelių, paveikslų numeriai rašomi pradžioje (prieš santrumpą), pavyzdžiui, 1 pav.?
4. Ar ekrane matomi elektroninio laiško antraštėje esančių laukų pavadinimai (*Subject, Re, Fwd* ir kt.) įtraukti į lokalizuojamusius išteklius?
5. Ar pranešimai apie išsiųstą laišką ir gautą laišką įtraukti į lokalizuojamusius išteklius?
6. Ar atsakymuose ir persiunčiamuose laiškuose pateikiamos informacijos tekstai įtraukti į lokalizuojamusius išteklius?
7. Ar pranešimai apie nerastą tinklalapį įtraukti į lokalizuojamusius išteklius?
8. Ar laukuose, sąrašuose ir kitose vietose asmens vardas eina prieš pavardę?
9. Ar yra linksniavimo galimybė ekrane matomų registracijos vardų ir asmenvardžių (pvz., *Sveiki, Jonai; Laiškas persiųstas Jonui*)?
10. Ar yra ekrane matomų būdvardžių ir dalyvių giminės suderinimo galimybė su asmens gimine?
11. Ar žinyne ir kitur pateikiamos iliustracijos, turinčios verstinių tekstų, yra įtrauktos į lokalizuojamusius išteklius?
12. Ar daiktavardžių linksniai derinami su skaičiais ir turi tris gramatinės formas (t. y. ar lokalizacijoje galima padaryti, kad būtų rodoma, pvz., *1 daiktas, 2 daiktai, 10 daiktų*)?

13. Ar tinklalapiai, į kuriuos eina saitai iš dialogo langų, turi lietuviškus variantus? Jei neturi, tai jie turėtų būti lokalizuojami kartu su programa?

14. Ar tinklalapiai, turintys informacijos, būtinai reikalingos programos naudotojui, į kuriuos eina saitai iš programos žinyno, turi lietuviškus variantus? Jei neturi, tai jie turėtų būti lokalizuojami kartu su programa.

15. Ar užrašai telpa į jiems skirtas fiksuoto ilgio vietas?

16. Ar teisinga matavimo vienetų skyryba: tarp skaičiaus ir vieneto pavadinimo ar jo trumpinio yra tarpas (pvz., 8 bitai, 5 cm, bet ne 8bitai, 5cm)?

17. Ar dimensijose vienetai, kurie neįtraukti į lokalizuojamuosius išteklis, skiriami dešiminiu brūkšniu (pvz., b/s, ne bps)?

18. Ar nėra nuostatų, kurių buvimas lokalizuotoje programoje klaidintų jos naudotoją (pvz., siūlymas savaitės dienų pavadinimus rašyti iš didžiųjų raidžių)?

19. Ar tekstų rengyklėse vartojami dokumentų (tarnybinių raštų, laiškų, pažymų, įgaliojimų ir kt.) šablonai įtraukti į lokalizuojamuosius išteklis?

Šis, paskutinis, klausimas labai bendras. Dokumentų šablonų yra daug ir jie sudaro didelę tekstų rengyklių lokalizuojamųjų išteklių dalį, dedamą į atskirus failus. Jų lokalizavimas, tiksliau dokumentų šablonų parengimas, yra didelis specifinis darbas, kurį reikėtų nagrinėti atskirai.

Išvados

Lokalizuotų programų testavimas ir jų lokalizuojamųjų išteklių analizė parodė, kad iš visų rastų klaidų galima išskirti internacionalizavimo klaidas. Panaudojant išteklių analizę, jų pseudovertimą ir bandomąjį abejotinų išteklių eilučių vertimą, daugelį internacionalizavimo klaidų galima rasti dar nelokalizuotoje originalioje programoje.

Gintautas Grigas

Software Testing According to Internationalization Bugs

Summary

Bugs, found in localized software, may be classified into two categories: localization bugs and internationalization bugs. The localization bugs are made by the localizer and may be fixed by the localizer. Internationalization bugs are made by the author of software source code and may be fixed by the author.

The paper discusses how the two kinds of bugs can be separated from each other in the localized program, or how to find internationalization bugs in original program before its localization. Early bug reports enable product's author to fix bugs before starting the localization process, or before the public release of the product. In the worst case, early identification of internationalization bugs enables the localizer to estimate the influence of not fixed bugs on localization quality. The tools and methods to identify the internationalization bugs are discussed: analysis of localization resources, pseudotranslation (pseudolocalization), local experimental localization, experimentation with the program during its run-time.

The investigation was based on testing of ten popular programs. Some experience on search for internalization bugs is presented as well. The questionnaire with the list of the most frequent internationalisation bugs is presented. The bugs are classified into three categories: alphabetic symbols, formats, and texts. The questionnaire may be considered as a guide for testers enabling to find bugs faster and in deeper layers of the program.

Straipsnis įteiktas 2010 02
Parengtas spaudai 2010 09

Apie autorių

Gintautas Grigas, informatikos mokslų daktaras, Vilniaus universiteto Matematikos ir informatikos instituto Informatikos metodologijos skyriaus emeritas. *Mokslinės veiklos sritys*: programavimo kalbos, programavimo mokymas, programinės įrangos lokalizavimas ir internacionalizavimas, kompiuterijos terminija.

Adresas: Vilniaus universitetas, Matematikos ir informatikos institutas, Akademijos g. 4, 08663 Vilnius.

El. paštas: grigas@kti.mii.lt

Ankstyvas internacionalizavimo klaidų radimas sudaro palankias sąlygas joms ištaisyti ir taip pasiekti geresnės lokalizacijos kokybės, sumažinti per kompiuterius plintanti neigiamą anglų kalbos poveikį lietuvių kalbai. Taip pat padeda išsklaidyti abejones, kad dar pasitaikančių kalbos nesklandumų kompiuterio įrangoje priežastis yra kompiuterio, o ne kalbos problema, išsprendžiama pašalinus internacionalizacijos klaidas.

Siekiant paspartinti klaidų paiešką parengtas klausimynas, į kurį įtrauktos daugiausia tikėtinos internacionalizavimo klaidos.

Internationalizavimo klaidos, sukeliančios lietuviškos lokalizacijos klaidas, analogišką neigiamą poveikį daro ir daugelio kitų kalbų lokalizacijoms.

Literatūra

1. Dagiene, V., Laucius, R., 2004. Internationalization of Open Source Software: Framework and Some Issues. In: T. Boyle, P. Oriogun, A. Pakstas, eds. *2nd International Conference Information Technology: Research and Education*. London, pp.204-207.
2. Desikan, S., Ramesh, G., 2009. *Software Testing: Principles and Practices*. Pearson Education.
3. Handerson, I., 2003. The Art of Multilingual Software Testing. *Software Business Magazine*, Nov.-Dec.
4. Hogan, J. M., Ho-Stuart, C., Pham, B., 2003. Current Issues in Software Internationalisation. *Australian Computer Science Conference*. Adelaide, May 2003.
5. Jevsikova, T., 2009. *Internetinės programinės įrangos lokalizavimas*. Daktaro disertacija. VDU ir MII.
6. Laucius, R., 2007. *Kompiliatorių internacionalizacija*. Daktaro disertacija. VGTU ir MII.
7. Safar, L., Machala, J., 2010. Best Practices in Localization Testing. *Multilingual*, 21(1), pp.1-5.
8. O'Sullivan, P., 2001. *User Interface and Translation*. Doctoral Dissertation. University of Limerick.

DOI: 10.5755/j01.sal.1.17.43378