

horizontan kavon, kreitan de la hodiaŭ ŝatata pure pragmatika vizio, kiu eĉ la subskribon de apenaŭ efika *Kyoto*-protokolo ne akceptas.

Ĉu d-ro Carlevaro povus esperantigi d-ron Zamenhofo?

En ŝerca frazo, rakontanta pri li mem en la tria persono, dTC skribas: “Kalkulu kiom da mono, da laboro kaj da tempo mia amiko Carlevaro elspezis por Esperanta propagando. En pli ol 40-jara aktivado, li sukcesis vere esperantigi nur *unu* homon” (p.13). Tamen, samkiel en aliaj “pruvoj” de la broŝuro, temas ne pri reala malsukceso de Esperanto, eĉ ne ĉiam pri la malkompetenteco de niaj “apostoloj”, sed pri la fakto, ke la lokaj, la nunaj, la spirite malpuraj kondiĉoj necesigas *samfortajn agadojn de kontraŭa senco*, aŭ probable pli fortajn ol la obstakloj. Aliloke mi plurfoje rakontis pri tute aliaj sukcesoj de Esperanto (kiujn denove la dogmismaj stalinistoj detruis).

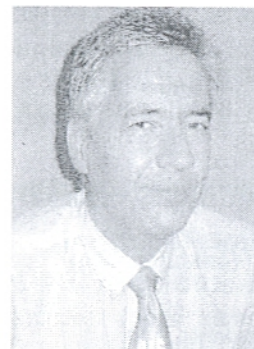
Oni ne scias, kiom da tempo kaj da trafaj argumentoj uzis dTC por dum 40-jara agado esperantigi unu solan homon. Mi ne kredas, ke li, kun siaj hodiaŭaj konceptoj, estis iam esperantisto – li estis nur bona esperantoparolanto. Do la koncernan esperantigitan homon oni devus trovi kaj konservi kiel “muzeaĵon”, ĉar, en la tuta historio, li estas la unua homo konvertita, fare de ateisto, al kredo je Dio. Tio estas grandega sukceso de dTC, ĉar, ekzemple, la nuntempa usona prezidanto *Bush* ne sukcesus konverti iun al islamismo. Mia opinio, d-ro *Carlevaro* ne vere povus esperantigi iun ajn homon, ankaŭ se tiu homo estus Zamenhofo mem.

Adreso de la aŭtoro

Prof. d-ro Ignat F. BOCIORT
Wustrower Strasse 1
DE – 13051 – Berlin / GERMANIO

Priaŭtora informo

La aŭtoro finstudis Filozofion kaj Filologion en la Universitato de *Bukarest*, Rumanio, kaj, je la sama universitato, ekfunkciis kiel asistanto kaj lektoro. Li doktoriĝis en la Universitato *Humboldt, Berlin (GDR)*, kie li gastdocentis (1965-1970) kaj esperantiĝis. Profesoro kaj estro de la fakto pri Beleta estetiko kaj Monda literaturo en la Universitato de *Timișoara*, Rumanio, ĝis la pensiigo. Krom siaj rumanlingvaj fakaj libroj, li aperigis multajn esperantlingvajn artikolojn pri strategio de nia movado kaj studojn pri estetiko de la beletro. Prof. d-ro Ignat Florian BOCIORT estas honora membro de UEA.



Gramatikaj aspektoj en komputila internaciigo

Reinhard FÖSSMEIER

1. Enkonduko

1.1 Kio estas internaciigo?

La tefniko de komputila internaciigo, alivorte kreado de plurlingve uzeblaj komputil-programoj, ekestis en la 1980-aj jaroj. La unua vaste konata sistemo tiucela estis NLS (*Native Language Support*) de firmao *Hewlett-Packard*; ĝi aperis en 1985 (vd. *Bale & Kellog* 1985).

Celo de internaciigo estas krei programojn, kies lingvo-dependaj ecoj estas izolitaj en aparta bloko, ŝanĝebla sen modifo de la programo mem. Tia lingvo-bloko esence estas listo de vortoj, frazeroj aŭ tutaj frazoj, kiujn la programo bezonas.

Baza problemo de internaciigo estas, ke programisto ne povas antaŭvidi, en kiuj lingvoj lia programo iam estos uzata, kaj se li povus, li ne scius tiujn lingvojn kaj iliajn apartaĵojn. Tial li ne povas memstare prepari sian programon tiel, ke ĝi funkciu kun lingvo-blokoj el ĉiuj tiuj lingvoj. La tefniko de internaciigo solvu la problemon; ĝi faras tion ofertante programadajn kadrojn (kia NLS), kiuj kapablas trakti tiujn lingvajn apartaĵojn. Tia kadro kutime havas la formon de biblioteko da subprogramoj.

Ekzemplo estas, ke la maniero, kiel skribi daton, diferencas inter lingvoj (kaj kulturoj). Internaciiga kadro do proponu metodon, kiu skribas daton laŭ la lingvaj kaj kulturaj kutimoj de la uzanto; la programisto simple voku tiun metodon.

„Internaciigo“ nomiĝas ankaŭ la pretigo de unulingva programo al plurlingva uzo. Kombino el lingvo kaj kulturo en internaciigo nomiĝas „lok-medio“ (angle: *locale*). La konkreta adapto de programo al specifa lok-medio nomiĝas „lokecigo“.

1.2 Kiel funkcias internaciigo?

La principo de internaciigo estas sufiĉe simpla: Por pretigi plurlingve uzeblan („internaciigitan“) programon necesas tri specoj de scio:

- A) ĝenerala scio pri programado (tia scio kompreneble necesas ankaŭ por unulingva programo);
- B) scio pri la faka funkciado de la programo (ankaŭ tia scio necesas por unulingva programo);
- C) scio pri la lingvoj, en kiuj la programo funkcias.

Kiam temas pri unulingva programo, eblas kunigi tiujn sciojn en unu persono, sed pri internaciigita programo tio ne eblas. Internaciigo procedas tiel, ke ĉiu persono havu nur du el la tri scioj:

1. Programistoj havu, kiel tradicie, la unuajn du sciojn (A, B), sed ne bezonu scion pri (pluraj aŭ eĉ multaj) lingvoj.
2. Tradukistoj sciu lingvojn (C) kaj tiom multe pri la faka funkciado de la programo (B), ke ili enhave komprenas la uzatajn tekstojn.
3. Fakuloj pri internaciigo sciu (kolektive, en grupo) multajn lingvojn kaj komprenu iliajn strukturojn (C). Krome ili estu fakuloj pri programado (A). Per tiuj du scioj ili kapablas krei internaciigan program-kadron sen scii, kiaj programoj iam uzos tiun kadron.

NLS estis la unua tia kadro, kiu estis vaste uzata. Aliaj ekzemploj estas la internaciigaj kadroj de la program-interfacoj POSIX kaj Qt kaj de la programad-lingvo Ĝavo (*Java*), kiuj ne havas apartajn nomojn.

2. Solvitaj problemoj

La modernaj internaciigaj kadroj bone funkcias, kiam temas pri jenaj ofte uzataj aspektoj de internaciigo (en program-mesaĝoj, grafikaj surfacoj ktp.):

- Nevariaj vortoj, frazeroj aŭ frazoj;
- Prezento de datoj kaj tempoj;
- Prezento de nombroj (kun dekuma kaj grupigaj signoj), eventuale kun valutoj (prezoj);
- Alfabeto ordigo en alfabetaj lingvoj;
- Ŝanĝo inter majuskloj kaj minuskloj;
- Malsamaj skrib-direktoj, eĉ mikse en unu teksto.

Videblas, ke temas pri la kampo de pura leksiko kaj pri tre klare priskribeblaj taskoj. Kiam implikiĝas gramatikaj aspektoj (morfoloĝio kaj sintakso), la subteno estas malpli bona. Tion montru jenaj ekzemploj.

3. Gramatikaj problemoj

3.1 Pluraj variaj elementoj en frazo

Okazas, ke en frazo-ŝablono estas unu aŭ pluraj variaj elementoj, ekzemple nombroj aŭ nomoj. Tiun ĉi problemon bone priskribas *Burke & Lachler* (1999). Estus eraro dividi la frazon al pluraj partoj (antaŭ, inter kaj post la variaj elementoj) kaj internaciigi tiujn fragmentojn aparte, ĉar fragmentoj ofte ne estas senhave tradukeblaj. Oni jam en la 1980-aj jaroj solvis tiun problemon per substituaĵoj en la ŝablono, kiujn anstataŭas la variaj elementoj. La proceduroj de programlingvo C kutime uzis la procento-signon por marki substituaĵojn.

En internaciigo aperas la problemo, ke la vicordo de pluraj variaj elementoj povas varii en diversaj lingvoj. Pro tio necesas distingi la substituaĵojn, ekzemple per numero. La klaso *MessageFormat* en Ĝavo uzas kunigajn krampojn (vinkulojn): $\{0\}$ estas la unua substituaĵo, $\{1\}$ la dua ktp.

3.2 Nombrado de objektoj

Kiam varia elemento en frazo estas la nombro de preparolataj objektoj, diversaj lingvoj aplikas diversajn esprimajn rimedojn (ankaŭ tiun ĉi problemon tre ĉarme traktas *Burke & Lachler*, 1999):

- Unu nevaria formo de la substantivo.
- Du diversaj formoj de la substantivo, nomataj singularo kaj pluralo
 - kun singularo uzata por 0 objektoj aŭ
 - kun pluralo uzata por 0 objektoj
- Pliaj formoj de la substantivo por specialaj nombroj, ekz. dualo (aldone al singularo kaj pluralo).
- Pluraj formoj de la substantivo, kiuj estas uzataj por po pluraj nombroj (ekz. en slavaj lingvoj),
- specialaj nombrad-substantivoj metataj antaŭ la nombratan substantivon (ekz. en la japana, la ĉina kaj la ĉin-angla piĝino).

La unua formo estas teĥnike plej simpla. Tial ĝi foje estas imitata en lingvoj kun pluralo, per enkrampigo de la plurala formo („... pe-co(j)“). Necesas nenia subteno de internaciiga kadro. Sed ĉar multaj lingvoj – inter ili la multe uzata angla – havas apartajn singularon kaj pluralon, ekestis pli komfortaj solvoj.

3.2.1 Singularo kaj pluralo: Qt kaj KDE

De ĉirkaŭ 1997 la programad-kadro Qt (kp. la manlibron kaj *Tinkl* kaj al.) de firmao *Trolltech*, uzata ekzemple en la *Linuks*-a sistemo KDE, proponas en sia teksto-klaso *QString* metodojn kun la nomo „*translate*“ (angle: „traduku“); ili prenas tekstojn el teksto-listoj, kiuj ekzistas aparte por ĉiu uzata lingvo. „*translate*“ havas variaĵon kun apartaj tekstoj por singularo kaj pluralo. Se iu variabla enhavas la nombron, eblas do havi tekstoparojn laŭ jena ekzemplo:

- Singularo: „via esprimo estis trovita unu fojon“
- Pluralo: „via esprimo estis trovita %n fojojn“

Tiu metodo ne konsideras apartaĵojn de multaj lingvoj; tio ne estas tre surpriza, ĉar la dokumentoj de KDE parolas precipe pri „la angla“ kiel unua lingvo de programo; ĝi laŭeble estu uzata en la programo mem, kaj tradukoj el aliaj lingvoj ĉiam eliras de la angla. La lingva mondbildo de KDE do estas iom angla-centra.

3.2.2 Pluraj formoj por pluraj nombroj: Java (Ĝavo)

La internaciiga kadro de la programadlingvo Ĝavo (angle: *Java*) enhavas la klason *ChoiceFormat*, kiu permesas havi pli ol du formojn por nombrata substantivo. Tre grave estas, ke povas esti malsame da tiuj formoj en diversaj lingvoj; ilian nombron fiksas la tradukanto, ne la programanto.

Versio 1.1 de Ĝavo, aperinta en februaro 1997, enhavis tiun klason *ChoiceFormat*, sed ne klarigis ĝian uzon. Taŭga ekzemplo venis nur kun versio 1.2 en decembro 1998. Laŭ ĝi oni adaptas uzon de la klaso al nova lingvo per vico da tekstoj, unu por ĉiu nombro-dependa formo de la substantivo.

La formojn dividas vertikala streko („masto“, „|“). Ĉiu teksto konsistas el la minimuma nombro, por kiu ĝi estas aplikenda, kaj la eligenda teksto; inter tiuj du partoj estas dieso („#“). Tiujn du disigilojn ne menciis eldono 1.1

Jen ekzemplo por la lingvo slovaka:

“0#nevidel som autá|1#videl som jedno auto|2#videl som {0} autá|5#videl som {0} aut”

Tiuj kvar partoj signifas, ke por la nombro 0 oni uzas la unuan parton („mi ne vidis aŭton“), por la nombro 1 la duan („mi vidis unu aŭton“), por 2, 3 kaj 4 la trian kaj ek de 5 la kvaran („mi vidis ... aŭtojn“), laŭ Tabelo 1.

Tabelo 1: *ChoiceFormat* por slovaka teksto

Tekstoparto	Validas por	Ekzemplo	Traduko
0#nevidel som autá	0	nevidel som autá	mi ne vidis aŭtojn
1#videl som jedno auto	1	videl som jedno auto	mi vidis unu aŭton
2#videl som {0} autá	2, 3, 4	videl som 3 autá	mi vidis 3 aŭtojn
5#videl som {0} aut	5 ktp.	videl som 12 aut	mi vidis 12 aŭtojn

Tiu teĥniko tre bone funkcias por la slovaka, sed ne por la pola, kiu uzas malsamajn formojn depende nur de la lastaj du ciferoj de la nombro. Se ne estas limo por la grandeco de la nombroj, necesas distingi nefinie da intervaloj, kio ne eblas per la teĥniko de klaso *ChoiceFormat*.

3.2.3 Pluraj formoj laŭ kompleksaj kriterioj: GNU

La programaj bibliotekoj de projekto GNU proponas funkcion nomatan *gettext* kaj similajn, kiuj helpas pri internaciigo de tekstoj. Komence ili ne serioze okupiĝis pri la problemo de nombrado, sed ek de versio 2.2 de biblioteko *GNU-libc*, kiu aperis en 2001, la subteno estas sufiĉe bona, ĉar ĉiu lingvo povas specifi, kiom da pluralaj formoj ĝi havas kaj por kiuj nombro-intervaloj ili estas aplikendaj.

Male al la Ĝavo-klaso *ChoiceFormat*, GNU permesas specifi la intervalojn per formuloj, tiel ke eblas en finia teksto specifi nefinie da intervaloj. La formulo de iu lingvo devas por ĉiu nombro kalkuli la numeron de la uzenda formo. La manlibro de *GNU-gettext* donas interalie la ekzemplojn de la Tabelo 2.

Tabelo 2: Formuloj por GNU-*gettext* por kelkaj lingvoj

Gramatikaj formoj	Formulo	Ekzemploj
Singularo kaj pluralo, „nul“ uzas pluralon	nplurals=2; plural=n != 1;	Angla, Esperanto
Singularo kaj pluralo, „nul“ uzas singularon	nplurals=2; plural=n > 1;	Franca
Singularo (laŭ la lasta cifero) kaj pluralo, „nul“ uzas apartan formon	nplurals=3; plural=n%10==1 && n%100!=11 ? 0 : n != 0 ? 1 : 2;	Latva
Singularo, dualo, pluralo	nplurals=3; plural=n==1 ? 0 : n==2 ? 1 : 2;	Irlanda Gaela
Tri formoj por nombroj finiĝantaj per 1 (ne 11), per 2-4 (ne 12-14) kaj aliaj.	nplurals=3; plural=n==1 ? 0 : n%10>=2 && n%10<=4 && (n%100<10 n%100>=20) ? 1 : 2;	Pola

La formuloj uzas la operaciilojn de programlingvo C (kaj parencaj). Ekzemple la pola formulo estas legebla jene:

- Se n egalas al 1, uzu 0 (la unuan indikitan formon).
- Alie, se la lasta cifero ($n\%10$) estas inter 2 kaj 4 (inkluzive), sed la lastaj du ciferoj ($n\%100$) ne estas inter 10 kaj 19, uzu 1 (la duan indikitan formon).
- Alie uzu 2 (la trian indikitan formon).

Per tiuj formuloj GNU povas ĝuste trakti la nombradon de substantivoj en multaj lingvoj.

La formuloj, kiuj decidas pri uzo de la substantivaj formoj, ne estas parto de la internaciiga kadro, do de la program-biblioteko; ili estas

en la lokecigitaj (tradukitaj) tekstoj de ĉiu programo. Tio havas la malavantaĝon, ke ĉiu tradukisto, kiu lokecigas programon, devas provizi la formulon (normale kopiante ĝin de alia programo). Aliflanke ĝi ebligas al tradukisto la uzon de lingvo, kiu ĝis nun neniam estis uzata; li do ne dependas de la provizantoj de la internaciiga kadro kaj de ia listo de subtenataj lingvoj.

3.2.4 Nombradaj helpvortoj

Kelkaj lingvoj uzas help-vortojn antaŭ nombrataj substantivoj. Tiuj help-vortoj povas variu laŭ la speco de nombrataj objektoj, sed (escepte de fonetika asimilado) ne laŭ la nombro; tial ili ne prezentas problemojn en internaciigo. Escepto estas, ke pro tiuj vortoj oni nepre ne uzu la saman tradukotan tekston en nombrata kaj nenombrata senco, ĉar tio metas tradukiston en dilemon: Ĉu traduki la vorton “libroj” al la japana “hon” (= libro(j)) aŭ “satsu hon” (= da libroj)? Sed la ebla ekzisto de pluraj nombrad-formoj en aliaj lingvoj ĉiukaze devigas uzi apartan tekston por nombrataj substantivoj.

3.3 Artikoloj

Alia gramatiko problemo dum internaciigo estas la malsama uzo de artikoloj en diversaj lingvoj. Tiu problemo estas ĝis nun malmulte aŭ tute ne konsiderata en internaciigaj program-kadroj.

Lingvoj diferencas pri artikoloj interalie jene:

- Kelkaj lingvoj havas difinajn kaj nedifinajn artikolojn, aliaj nur unu specon, aliaj tute ne havas.
- En iuj lingvoj la artikolo varias laŭ la nombro, la kazo kaj/aŭ la genro.
- En iuj lingvoj artikoloj estas antaŭ la substantivoj, en aliaj malantaŭ ili, aŭ mikse.

La plej simpla solvo ŝajnas esti neniam programe manipuli artikolojn, sed formuli ĉiujn tekstojn komplete kun la necesaj artikoloj. Sed ekestas problemo, kiam frazo enhavas variajn elementojn, kiel en la ĵusa sekcio 3.1. Ni prenu ekzemplan tekston pri komputilaj flank-aparatoj, kiuj havas specon (ekz. „presilo“, „modemo“) kaj nomon (ekz. „P 400“, „103 A“) kaj aperu en tekst-fragmento de la formo „stato de presilo P 400“.

La normala procedo por internaciigo estas meti du substituajojn en la fragmentojn, unu por la speco kaj unu por la nomo de la aparato.

Ĉar la vort-ordo varias inter la lingvoj, la substituajoj devas esti numeritaj. Por Esperanto la ŝablono de la fragmento do aspektas jene: „stato de {0} {1}“. El tiu ŝablono eblas konstrui ne nur la ĉi-supran ekzemplon pri presilo, sed ankaŭ „stato de ekrano TK 4014“ aŭ „stato de modemo 103A“.

Tradukante en la anglan, germanan aŭ rumanan lingvon oni normale uzas difinan artikolon por tiuj aparatoj. En la angla lingvo oni povus uzi la ŝablonon „state of the {1} {0}“ (la angla emas meti la nomon antaŭ la specon), ĉar en tiu lingvo estas nur unu difina artikolo, kies formo do ne dependas de la substantivo:

“state of the {1} {0}” + “printer” + “P 400” = “state of the P400 printer”

Sed en la germana la artikolo varias laŭ la genro kaj kazo, do ne eblas meti la artikolon en la ŝablonon. Anstataŭe oni povus meti ĝin al la substantivo (de la aparat-speco) kaj traduki kun ĝi:

“Zustand {0} {1}” + “des Druckers” + “P 400” = “Zustand des Druckers P400”

Tiu aliro funkcias ankaŭ por la rumana, en kiu la difina artikolo estas finaĵo de la substantivo kaj same varias laŭ genro kaj kazo:

“starea {0} {1}” + “imprimantei” + “P 400” = “starea imprimantei P400”

Tabelo 2: Du-faza substituado en ŝablonoj			
Lingvo	Frazo-ŝablono	Speco-ŝablono	Rezulto
Eo	Stato de {0}	Presilo {0}	Stato de presilo P 400
De	Zustand {0}	des Druckers {0}	Zustand des Druckers P400
Ro	Starea {0}	imprimantei {0}	Starea imprimantei P 400
En	State of	the {0} printer	State of the P400 printer

Sed ĝi ne funkcias en lingvo, kiu kombinas la variecon de la artikolo kun la antaŭmeto de la nomo (kiel ekzemple la angla kun la nedifina artikolo, kiu havas du formojn): Tie la „nomo“ staras inter la „speco“ kaj ĝia artikolo, tiel ke ne simple eblas kodi tiujn du en unu teksto. Tamen eblas solvo per du-faza substituado de substituajoj: la lingvo-dependaj tekstoj por la „specoj“ enhavu substituajojn por la „nomo“. Tiam unue la „nomo“ estas enmetita en la speco-ŝablonon kaj poste tiu ĉi en la ŝablonon de la fraz-fragmento, laŭ tabelo 2.

Ni enkondukis malfacilaĵojn unu post alia (varieco de artikoloj, pozicio de artikoloj, vort-ordo, kombinoj el tiuj ĉi) kaj ĉiam trovis solvon, sed nia fina solvo havas du malavantaĝojn:

- Ne estas klare, ĉu ĝi funkcias por aliaj lingvoj kun eble aliaj apartaj ecoj;
- Ni trovis ĝin nur post konsidero de tradukoj en diversajn lingvojn; tio estas kontraŭ la principo de internaciigo, ke programisto ne bezonu lingvan scion.

Nunaj internaciigaj kadroj ne donas subtenon pri artikoloj, sed nia ekzemplo montris, ke ilia uzo ne estas senproblema. Konkludeblas, ke la problemo de variaj artikoloj en internaciigo ankoraŭ ne estas solvita.

4. Konkludo

Ni vidis, ke internaciigo sukcese solvis la problemojn de pluraj variaj teksto-partoj kaj de nombrado, sed bezonis por ĉi-lastata konsiderindan tempon; la solvo de GNU en 2001 estis la unua kontentiga. La problemon de variaj artikoloj kaj variaj vort-ordoj internaciigo ankoraŭ ne ĝenerale solvis, kvankam temas pri sufiĉe elementa gramatik-aĵo. Pri tiu kaj similaj gramatikaj problemoj la programistoj do ankoraŭ dependas de la propraj lingvaj scio kaj intuicio.

La ekzemplo de nombrado bone ilustras, ke la ĝisnuna evoluo de internaciigo konsistis en solvado de unuopaj problemoj, kiuj ekestis dum lokecigo; la praktiko de tradukado pelis la evoluon. Ŝajnas, ke la plej urĝaj tiaj problemoj nun estas solvitaj; la resto ne estas forta stimulo por pluevoluigo. Ne atendindas, ke laŭ tiu vojo okazos signifa plua progreso; por tio necesas formala aliro, kiu povas veni nur de la teoria flanko, de lingvistiko.

Por formale trakti gramatikajn fenomenojn de internaciigo necesas klasi ilin en formo, kiu ne dependas de iu(j) lingvo(j). Eble por tio povas utili projektoj pri ponto-lingva maŝina tradukado, ekzemple DLT (vd. *Schubert* 1988) kaj UNL (vd. *Hong & Streiter* 1999 kaj *Delbecq* 1998); ili jam faris tian klasadon en komputile uzebla formo. Ambaŭ ĝis nun ne estas praktike aplikataj por tradukado; eble ili povus utili sur la kampo de internaciigo.

5. Referencoj

Jonathan E. BALE, Harry E. KELLOG: Native Language Support for Computer Systems. En: Hewlett-Packard Journal, junio 1985, 27-32.

Sean M. BURKE, Jordan LACHLER: Localization and Perl: gettext breaks, Maketext fixes. En: The Perl Journal (13), 1999.

Denis DELBECQ: UNL, un espéranto pour le Web. Le Monde Interactif (1998-11-07, daŭrigo 1998-12-06).

GNU gettext utilities (programista manlibro), 2002, <http://www.gnu.org/software/gettext/manual/>

Munpyo HONG, Oliver STREITER: Overcoming the Language Barriers in the Web: The UNL-Approach. En: Multilinguale Corpora: Codierung, Strukturierung, Analyse. 11. Jahrestagung der Gesellschaft für Linguistische DatenVerarbeitung. Jost GIPPERT kaj Peter OLIVIER, Enigma Corporation 1999, Prago, Ĉeĥio.

Guide to the Qt Translation Tools. Trolltech, 2003. <http://doc.trolltech.com/3.1/linguist-manual.html>

Klaus SCHUBERT: The architecture of DLT – interlingua or double direct? En: *Dan MAXWELL, Klaus SCHUBERT, Toon WITKAM* (kompilintoj), New Directions in Machine Translation. Foris Publication, 1988, Dordrecht, Nederlando.

Lukáš TINKL, Matthias KIEFER, Gary CRAMBLITT: KDE Programmer's i18n howto. Versio 3.3, 2006. <http://developer.kde.org/documentation/library/kdeqt/kde3arch/kde-i18n-howto.html>

Adreso de la aŭtoro

OProf. Dr. Reinhard FÖSSMEIER
Rahel-Straus-Weg 19
DE – 81673 München / GERMANIO
<reinhard@foessmeier.name>

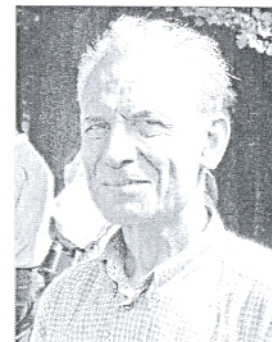
Priaŭtoro informo

La aŭtoro estas komputilisto ĉe firmao *Sofilab GmbH* en Munkeno kaj instruas informadikon kiel honora profesoro de Universitato *Lucian Blaga* en *Sibiu-Hermannstadt* kaj ĉe Akademio Internacia de la Sciencoj (AIS) San Marino. Krome li estas senata sekretario de AIS San Marino.

Lingvopolitikaj scenaroj en Eŭropa Unio

– Komento al *François GRIN* –

Johan H. DERKS



La angla lingvo en la natursciencoj

Multaj aŭtoroj jam atentigis pri la giganta progreso de la angla lingvo en la lastaj okdek jaroj. En la pasinta jaro *Vergara* (2006a) kolektis specifan materialon pri la penetrado de la angla lingvo en la kampon de la sciencoj.

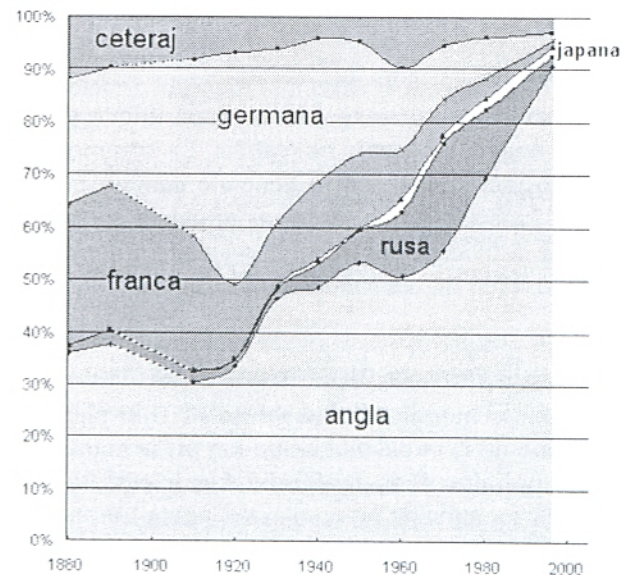


Fig. 1: Evoluo de la uzado de grandaj lingvoj en la natursciencoj laŭ *Ammon & Dittmar* (2006)