

## Generality Metrics of Software Translators

Ec. Robert ENYEDI

*Measuring the evolution of generality is of a great importance in the development process of software translators. This paper proposes a methodology for computing generality metrics for software translators. The importance of the evolution of generality in correlation with translator types is also discussed.*

**Keywords:** software translators, software metrics, generality of translators

### Introducere

Translatoroarele software sunt instrumente ce realizează transformarea automată a unui program scris într-un limbaj de programare denumit limbajul sursă într-un program echivalent exprimat într-un limbaj de programare diferit denumit limbajul țintă.

Generalitatea unui translator software se referă la procentul de acoperire a construcțiilor și bibliotecii standard ale limbajului sursă.

Construcțiile limbajului sursă sunt descrise de gramatica acestuia și includ: structuri de program, definiții de date, instrucțiuni

Structurile de program sunt utilizate pentru a grupa instrucțiuni și definiții de date în blocuri omogene cu un flux de apelare precis stabilit. Definițiile de date descriu structura zonelor de memorie utilizate de aplicație în timp ce instrucțiunile produc acțiuni de manipulare a datelor și de comunicare cu variabilele componente ale sistemului gazdă precum și ale altor sisteme interconectate cu sistemul gazdă.

### Metrici simple ale generalității translatoroalelor

Aprecierea gradului de generalitate pentru translatoroarele software ia în calcul în primul rând gradul de acoperire a construcțiilor și funcționalităților limbajului sursă. De exemplu, un generator de scheme logice este generic dacă întreaga funcționalitate a limbajului sursă este implementată.

Generalitatea translatorului se referă la gradul de acoperire a construcțiilor de limbaj sursă și o primă formulă de calcul este:

$$G_T = \frac{ConI}{ConT} \quad (1), \text{ unde:}$$

$G_T$  - generalitatea translatorului

$ConI$  - numărul de construcții implementate pentru limbajul sursă

$ConT$  - numărul total de construcții ale limbajului sursă

Conform formulei de calcul, generalitate translatorului ia valori în intervalul  $[0;1]$ .

Valoarea 1 semnifică generalitatea maximă realizată atunci când numărul construcțiilor implementate pentru limbajul sursă este egal de numărul total de construcții ale limbajului sursă. Valoarea 0 indică lipsa implementării oricărei construcții și în această situație nu putem vorbi despre un translator propriu zis.

Dincolo de implementarea/neimplementarea construcțiilor, aprecierea gradului de generalitate pentru un translator trebuie să țină seama și de importanța fiecărei construcții.

Formula modificată de calcul al generalității translatorului devine:

$$G_T = \frac{\sum CoImp_i}{ConT}, \quad CoImp_i \in [0;1] \quad (2), \text{ unde:}$$

$G_T$  - generalitatea translatorului

$CoImp_i$  - coeficientul de importanță pentru o construcție a limbajului sursă

$ConT$  - numărul total de construcții ale limbajului sursă

Coeficientul de importanță a unei construcții de limbaj se determină de către dezvoltatorul translatorului și ia în calcul în primul rând frecvența medie de utilizare a construcției în cadrul programelor. Programele ce sunt luate în calcul sunt proiectele de tradus, în cazul unui translator ca proiect, sau programe reprezentative pentru limbajul respectiv de la exemple, aplicații interne sau externe cu codul sursă disponibil etc.

### Metrici complexe

În cazul unor translatoroale complexe se ia în calcul și coeficientul de implementare pentru caracteristicile limbajului sursă. Pentru un limbaj sursă complex, apare situația în care unele construcții sunt parțial suportate.

Ținând cont și de coeficientul de implementare, formula generalității translatorilor devine:

$$G_T = \frac{\sum CoImp_i \cdot CoImpl_i}{ConT}, CoImp_i \in [0; 1], CoImpl_i \in [0; 1] \quad (3)$$

unde:

$G_T$  - generalitatea translatorului

$CoImp_i$  - coeficientul de importanță pentru o construcție implementată a limbajului sursă

$CoImpl_i$  - coeficientul de implementare pentru translatarea unei construcții a limbajului sursă

$ConT$  - numărul total de construcții ale limbajului sursă

Determinarea coeficientului de implementare se face de către dezvoltatorul translatorului. În stabilirea cuantumului coeficientului se ia în calcul efortul depus, de exemplu în omzile sau cost, pentru implementarea în stadiul actual și efortul estimat pentru o implementare completă.

Oricare ar fi limbajul în care se dezvoltă o aplicație software, programatorul utilizează, dincolo de construcțiile de limbaj, o bibliotecă standard de programe precum și un set de biblioteci de la terți. Translatarea unei aplicații către un nou limbaj presupune și translatarea bibliotecilor utilizate de programele scrise în limbajul sursă.

Spre deosebire de aplicațiile de translatat care au disponibile codul sursă, pentru biblio-

$$G_{TB} = \frac{\sum CoImpSub_i \cdot Subr_i}{SubrT}, Subr_i \in \{0, 1\}, CoImpSub_i \in [0; 1] \quad (4), \text{ unde:}$$

$CoImpSub_i$  - coeficientul de importanță a subrutinei

$Subr_i$  - indicator de subrutină cu comportamentul translatat

$SubrT$  - numărul total de subrutine în biblioteca standard

Determinarea coeficientului de importanță pentru o subrutină se calculează pe un set de aplicații de translatat. Procesul de calcul se

$$G_T = C_L \cdot \frac{\sum CoImp_i \cdot CoImpl_i}{ConT} + (1 - C_L) \cdot G_{TB_{st}}, CoImp_i \in [0; 1], CoImpl_i \in [0; 1] \quad (5), \text{ unde:}$$

$G_T$  - generalitatea translatorului

$CoImp_i$  - coeficientul de importanță pentru o construcție implementată a limbajului sursă

$CoImpl_i$  - coeficientul de implementare pentru translatarea unei construcții a limbajului sursă

$ConT$  - numărul total de construcții ale lim-

tecile utilizate există posibilitatea indisponibilității codului sursă. Mai mult, pentru multe limbaje nu este disponibil nici măcar codul sursă pentru biblioteca standard.

În calcularea gradului de translatare a unei biblioteci  $G_{TB}$  există două abordări. În situația în care codul sursă al bibliotecii este disponibil avem relația  $G_{TB} = 0$ , adică gradul de translatare a bibliotecii nu influențează generalitatea translatorului. Egalitatea se realizează deoarece pentru a realiza translatarea automată a bibliotecii se utilizează același translator ca pentru aplicație.

Atunci când codul sursă pentru o bibliotecă nu este disponibil, nu se mai realizează o translatare automată. Translatarea comportamentului unei biblioteci a limbajului sursă într-un comportament echivalent în limbajul țintă devine o operație manuală. Descrierea comportamentului unei biblioteci este suma comportamentelor individuale ale fiecărei subrutine. Aceste comportamente sunt descrise în documentația bibliotecii. Astfel, gradul de translatare a bibliotecii  $G_{TB}$  în această situație se calculează după formula:

bazează pe numărul de utilizări ale fiecărei subrutine și se determină ponderea relativă a utilizării fiecărei subrutine în numărul total de apeluri de subrutine.

Deoarece biblioteca standard este utilizată de orice aplicație în limbajul sursă, este foarte importantă extinderea indicatorului de generalitate astfel încât să includă și gradul de translatare a bibliotecii standard.

bajului sursă

$C_L$  - coeficientul de importanță a limbajului

$G_{TB_{st}}$  - gradul de translatare a bibliotecii standard

Coeficientul de importanță a limbajului  $C_L$  reprezintă ponderea generalității de translatare a limbajului în raport cu generalitatea de

translatore a bibliotecii standard. Determinarea coeficientului de importanță a limbajului depinde de limbajul sursă și se determină în funcție de numărul de construcții ale limbajului ConT și numărul total de subrutine ale bibliotecii standard SubrT. În cazul limbajelor orientate obiect, metodele unei clase se consideră subrutine.

În situația în care codul sursă este disponibil în totalitate pentru limbajul sursă,  $C_L$  are valoarea maximă (1) deoarece generalitatea translatorului este în totalitate condiționată de gradul de acoperire a construcțiilor limbajului. Deși este situația ideală, în general ea nu se realizează deoarece bibliotecile standard au o implementare proprietară. Disponibilitatea se referă și la permisiunea legală de a transla codul sursă disponibil pentru biblioteca standard.

Această ultimă variantă de calcul a generalității este deosebit de importantă pentru a evidenția dinamica dezvoltării iterative a translatorilor ca produs și în special pentru translatorii ca proiect.

#### **Importanța comensurării generalității**

Importanța generalității translatorilor software este în funcție de poziționarea pe piață a translatorului: translator ca produs sau translator ca proiect.

Generalitatea este crucială pentru un translator care este un produs, deoarece trebuie să translateze corect orice aplicație în limbajul sursă suportat. Aceasta înseamnă că toate construcțiile limbajului trebuie să fie tratate și să se ofere o translatore a funcționalității bibliotecii standard. Un translator dezvoltat ca proiect în general are cerințe mai reduse în sensul necesității de a suporta doar o parte din construcțiile limbajului sursă și a bibliotecii standard.

Disponibilitatea codului sursă pentru biblioteca standard influențează în mare măsură efortul necesar dezvoltării unui translator software. Situația ideală este cea în care codul sursă al bibliotecii standard există și este utilizabil legal pentru a aplica o translatore asupra ei către noul limbaj țintă. De cele mai multe ori însă codul sursă al bibliotecii standard nu este disponibil, acesta fiind și unul dintre motivele pentru care a apărut nevoia unui translator pentru limbajul respectiv:

tehnologiile proprietare.

Tendința actuală în industria software este către standarde deschise. Astfel prezintă importanță tehnologiile care sunt transparente deoarece există concurență între implementările alternative și oricând este posibilă realizarea unei implementări compatibile, cu un efort calculabil. De asemenea, translatorele către tehnologii moderne va deveni mult mai accesibilă atunci când evoluțiile tehnologice o vor cere. În prezent în majoritatea cazurilor limbajul și tehnologiile sursă pentru un translator software sunt cele proprietare și în general învechite și nesuportate comercial iar limbajul și tehnologiile țintă sunt unele standardizate.

#### **Concluzii**

Translatorii software sunt instrumente de transformare automată a programelor de la un limbaj țintă la un limbaj destinație. Caracteristicile software generale se aplică și translatorilor, însă apar cerințe suplimentare determinate de particularitățile translatorilor.

Caracteristica de generalitate a translatorilor software este comensurabilă și este direct proporțională cu rata de implementare a construcțiilor limbajului sursă. De asemenea, pentru limbaje sursă ce folosesc biblioteci standard cu codul sursă nedisponibil, generalitatea este în dependență directă cu rata de implementare a subrutinelor bibliotecii standard. Există două abordări distincte referitor la construirea unui translator software. Dezvoltarea unui translator ca produs presupune atingerea unui nivel ridicat de generalitate la lansarea primei versiuni publice a translatorului, pe când un translator ca proiect se dezvoltă iterativ pe măsura realizării de translații de aplicații cu ajutorul acesteia. Finalitatea ambelor abordări o reprezintă o generalitate maximă a translatorului, egală sau foarte apropiată de 1.

Una din direcțiile în studiul viitor al generalității translatorilor se va concretiza în cercetarea dependențelor dintre generalitatea și complexitatea translatorilor corelate cu costurile marginale.

#### **Bibliografie**

[AHOS86] Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman – Compilers: Principles,

Techniques, and Tools – Addison-Wesley, 1986

[ENY203] Robert Enyedi – Translatoare către limbaje grafice – *Informatica Economică*, Vol. 7, Nr. 1, 2003

[ENYE05] Robert Enyedi – Generalitatea - caracteristică de calitate a translatoarelor software – referatul al 2-lea din planul doctoral, ASE București, 2005

[IVPA97] Ion Ivan, Panagiotis Siniros, Mihai Popescu, Felix Simion – *Metrici software* – Infotec, București 1997

[WATT93] David A. Watt – *Programming Language Processors: Compilers and Interpreters* – Prentice Hall International, 1993