



Vrije Universiteit Brussel

Rectoraat
R&D Department

afdeling Wetenschapscommunicatie

Sofie Van Den Bossche 02/629.22.24.

Fax 02/629.36.40

Email ssvdboss@vub.ac.be

Pleinlaan 2 B-1050 Brussel

Informatie aan de pers

Datum 02/03/2001

Betreft **ONDERZOEK NAAR TECHNIEKEN TER BESTRIJDING VAN SOFTWARE-EROSIE**

ALGORITME CONTROLEERT COMPUTERPROGRAMMA'S

Computerprogramma's zijn enorme complexe bouwsels. Denk maar aan de hetze rond de 'milleniumbug'. Of een meer recent voorbeeld: de conversie van software naar de Euro-munteenheid. In beide voorbeelden krijgen bedrijven het hard te verduren om hun software tijdig aan te passen. Niet enkel de aanpassingen eisen hierbij het werk op maar ook het nakijken van de honderdduizenden lijnen die een programma kan tellen. Software-architectuur biedt een overzichtelijke oplossing. VUB-onderzoek leidde tot controle van de software-architectuur via een algoritme.

De jonge wetenschapper Dr. Kim Mens onderzocht een algoritme om na te gaan of een programma nog steeds conform is aan haar architectuur. Indien niet geeft het algoritme informatie over waar er inconsistenties tussen het programma en de architectuur bestaan. Een belangrijke evolutie voor de software-producenten.

Naar analogie met de architectuur van gebouwen vat een software-architectuur de structuur van een groot programma samen in een simpel schema. Net zoals een plan van een gebouw uit verschillende zichtpunten kan bestaan (vooraanzicht, grondplan, elektrische bedrading) kan een computerprogramma verschillende architecturale zichtpunten hebben. De software-architectuur bestaat uit alternatieve zichtpunten die toelaten het programma te bekijken vanuit verschillende perspectieven en die een ideale documentatie van een programma vormen. Een programma met een duidelijke architectuur is veel eenvoudiger te begrijpen en te onderhouden dan een programma zonder weldoordachte architectuur. Software heeft echter de neiging snel te "eroderen". Onder tijdsdruk wijzigen programmeurs vaak alleen de programmacode, bijvoorbeeld om fouten te verbeteren, zonder dat ze de architectuur mee aanpassen. Het duurt dan meestal niet lang voor de architectuur volledig achterhaald is en nog maar weinig nut heeft als documentatie van de software. Dit probleem van software-erosie is de voornaamste redenen waarom programmeurs maar zeer sporadisch de architectuur van hun programma's documenteren. Waarom zouden ze ook: er kruipt veel tijd in, en binnen de kortste keren is de gedocumenteerde architectuur toch achterhaald. Daarom is er een nood aan technieken om dit probleem tegen te gaan. Het Laboratorium voor Programmeerkunde aan de Vrije Universiteit Brussel onderzoekt zo'n technieken.

Het voorgestelde algoritme voor conformiteitscontrole is zeer eenvoudig en intuïtief. Dit omdat het algoritme gebruik maakt van een aantal zeer logische en begrijpbare beschrijvingen. Toch zal de gewone computergebruiker niet meteen met het algoritme overweg kunnen. Software-ontwikkelaars daarentegen kunnen het algoritme enkel toejuichen. In de toekomst zou het algoritme ook een aantal hints moeten geven om eventuele inconsistenties op te lossen en de oplossingstrategie semi-automatisch laten uitvoeren. Het is een veelbelovende eerste stap om het niet-triviale probleem van software-erosie op te lossen.

Voor meer informatie over dit onderzoek kan U zich wenden tot:

Dr. Kim Mens (kimmens@vub.ac.be), tel.: 00 32 (0)2 629 35 81

Laboratorium voor Programmeerkunde, Vakgroep Informatica

Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussel