

Formele Methoden voor uw eigen Applicatietaal

De automatiseringswereld kent een overvloed aan talen. Activiteiten als programmeren (COBOL, C, Pascal), specificeren (VDM, COLA, Z), systeembeheer (DOS of Unix shell, Perl) worden vaak gekenmerkt door de taal die gebruikt wordt.

Hier komt nog bij dat veel bedrijven hun eigen talen definiëren. Wanneer bestaande hulpmiddelen toereikend zijn zal dit niet nodig zijn. In veel situaties echter, rechtvaardigen de specifieke eisen van een bepaalde applicatie de introductie van een nieuwe taal. Zo hebben, om enkele voorbeelden te noemen, de afgelopen jaren de PTT, Nederlandse Spoorwegen, CAP Volmac, Philips en diverse kleinere bedrijven (mee)gewerkt aan ontwerp en implementatie van op hun eigen behoeften toegespitste applicatietalen (zie ook Tabel 1).

Renteprodukten

Een aardig voorbeeld van de ontwikkeling van een nieuwe taal speelde zich af bij de bank MeeSPierson. Het management van deze bank realiseerde zich dat het invoeren van nieuwe producten (op het terrein van rente en derivaten) en het beheersen van de daaraan verbonden risico's, bemoeilijkt werd door de lange ontwikkeltijd van de bijbehorende systemen.

In samenwerking met Cap Volmac (divisie Finance) ontstond daarom het idee om de essentie van dergelijke producten (d.w.z. hoe zij kasstromen genereren) vast te leggen in een *produktbeschrijvingstaal*. Het gebruik van de taal verbeterde de communicatie tussen eindgebruikers en systeembouwers. Bovendien kon een produktbeschrijving dienen om de voor een produkt benodigde software (zoals data-opslag procedures of rente-risico routines) volledig automatisch te genereren. Dit verkorte de ontwikkelingstijd aanzienlijk.

Taalontwerp

Hoe kan de introductie van een nieuwe taal in goede banen geleid worden?

Taalontwerp begint met een document waarin de noodzaak voor een nieuwe taal wordt uitgelegd. Een probleemstelling wordt geformuleerd, en er wordt uiteengezet hoe de nieuwe taal hiervoor een oplossing biedt. Belangrijk is verder een korte beschrijving van enkele kenmerkende taalconstructies. Daarnaast is het raadzaam voorbeeldprogramma's of -specificaties te geven, om de uitdrukingskracht van de taal aan de praktijk te kunnen toetsen.

Op grond van een dergelijk document wordt besloten of een nieuwe taal de te volgen weg is. Is dit het geval, dan zijn een *taaldefinitie* en taal-specifieke *tools* nodig. De taaldefinitie vertelt toekomstige gebruikers hoe zij een programma, specificatie, beschrijving, of query m.b.v. de taal kunnen opschrijven. De tools zorgen voor compilatie, interpretatie, of codegeneratie, en maken daadwerkelijk gebruik van de taal mogelijk. Bij het bouwen van deze tools dient de taaldefinitie als uitgangspunt.

De beste manier om een taaldefinitie te schrijven en de benodigde tools te verkrijgen is door gebruik te maken van *formele methoden*, niet in het minst omdat dit kan leiden tot een aanzienlijke tijdbesparing.

Formele Taaldefinities

Een formele methode kenmerkt zich door het gebruik van een wiskundig streng gedefinieerd formalisme. Een *formele taaldefinitie* gebruikt zo'n formalisme om de belangrijkste eigenschappen van de taal zeer precies te beschrijven.

In vergelijking met taaldefinities in natuurlijke taal (Nederlands, Engels), zijn de extra *nauwkeurigheid* en *ondubbelzinnigheid* veelgenoemde voordelen. Nog belangrijker is *volledigheid*: Een formele definitie maakt het onmo-

gelijk te ontsnappen naar mistige volzinnen die moeten verhullen dat een bepaald concept nog niet voldoende begrepen is.

De afgelopen decennia is door diverse universiteiten in binnen- en buitenland intensief onderzoek gedaan naar toepassingen van formele methoden op het gebied van taaldefinities. Diverse smaken en voorkeuren zijn mogelijk, zoals aangegeven in Tabel 2. De *leesbaarheid* van dergelijke formele definities moet vaak vooral komen van de begeleidende en verklarende tekst (documentatie, commentaar). Een uitzondering hierop vormt “aktie semantiek”, een methode die speciaal ontworpen is met leesbaarheid als doel.

Tools genereren

Het schrijven van een formele taaldefinitie vergt een aanzienlijke inspanning. In veel gevallen echter, kan deze investering terugverdiend worden. Wanneer de formele taaldefinitie voldoende gedetailleerd is, kan die gebruikt worden om de benodigde tools automatisch te genereren.

Het soort tools dat afgeleid kan worden hangt af van de eigenschappen die formeel gedefinieerd zijn. Uit een syntaxdefinitie bijvoorbeeld, valt eenvoudig een parser of syntax-gestuurde editor af te leiden. Heeft men gedefinieerd welke COBOL code correspondeert met een bepaalde produktbeschrijving, dan kan hieruit een COBOL generator afgeleid worden. Is een zogenaamde “operationele semantiek” gedefinieerd, dan kan die als basis dienen voor een interpreter van de gedefinieerde taal.

Het genereren van taalspecifiek gereedschap is de laatste jaren uitgebreid aan bod gekomen binnen het wetenschappelijk informatica-onderzoek. Groepen in Frankrijk, Groot-Brittannië, Denemarken, maar ook in Nederland hebben technieken uitgewerkt om de automatische generatie mogelijk te maken.

Aan de Universiteit van Amsterdam en het Centrum voor Wiskunde en Informatica is bijvoorbeeld een interactieve toolgenerator ontwikkeld, de “ASF+SDF Meta-omgeving”. Een gebruiker voert een taaldefinitie in, daarbij geholpen en gecorrigeerd door het systeem. De opbouw van de generator is dusdanig dat direct met de afgeleide tools geëxperimenteerd kan worden.

De ASF+SDF Meta-omgeving is succesvol ingezet tijdens het ontwerp van de rente-produkttaal zoals gebruikt bij MeesPierson. Er werd een formele specificatie gegeven van de syntax, onderliggende data-types, typeringseisen, en zelfs de generatie van de juiste COBOL routines.

Formele methoden zijn goed toepasbaar tijdens het ontwerp van applicatietalen. De taaldefinitie wint hierdoor aan nauwkeurigheid, en de tools kunnen automatisch afgeleid worden. Bijkomend voordeel is bovendien dat die tools snel beschikbaar zijn. Hierdoor kan de taal in een vroeg stadium daadwerkelijk gebruikt worden, en kunnen gebruikers feedback geven aan de taalontwerpers.

Arie van Deursen

dr. A. van Deursen is werkzaam bij de sectie Formele Methoden aan de Faculteit Wiskunde en Informatica van de Technische Universiteit Eindhoven. In september 1994 promoveerde hij onder begeleiding van prof. dr. P. Klint aan de Universiteit van Amsterdam op het proefschrift getiteld “Executable Language Definitions – Case Studies and Origin Tracking Techniques”. Zijn proefschrift behandelt een drietal voorbeeld taaldefinities, en bespreekt een nieuwe techniek om geavanceerde tools uit taaldefinities af te leiden.

Taal	Instelling	Doel
ISL	Nederlandse Spoorwegen	<i>Interlocking Specification Language</i> . Gebruikt door NS om veiligheidsinstallaties te beschrijven.
CHILL	PTT	CCITT High Level Language. Gebruikt voor het beschrijven van telecommunicatiestandaards.
MSL	Micro Technology Nederland	Op Nederlands lijkende scripttaal om gegevens via een modem te versturen.
COLD	Philips	Specificatietaal voor het beschrijven van software die complexe datastructuren manipuleert.
Risla	MeesPierson & Cap Volmac	Beschrijvingstaal voor renteprodukten en de kasstromen die zij genereren.

Figuur 1: Enkele in Nederland ontworpen computertalen

Methode	Omschrijving
(E)BNF	(Extended) Backus Naur Form. Dient uitsluitend om syntax (structuur) te beschrijven. Veel gebruikt.
Attribuut Grammatica's	BNF grammatica's, uitgebreid met attributen om extra informatie op te slaan. Beschrijving van syntax, statische constraints (type checking), of vertaling van ene taal naar andere.
Denotationele Semantiek	Geef een vertaling naar een wiskundige structuur. Van groot theoretisch belang, maar ook toegepast om bijv. Ada formeel te definiëren (gebruik makend van VDM als formalisme).
Operationele Semantiek	Beschrijf een wiskundige "machine", en hoe deze de taal kan interpreteren.
Aktie Semantiek	Geef aan hoe de taal correspondeert met bepaalde standaard "akties". Combineert leesbaarheid met de mogelijkheid om realistische, complexe talen te beschrijven.

Figuur 2: Enkele veelgebruikte methoden voor formele taaldefinities