Publicatie : New Scientist special
Datum : 28 feb 2021 Regio : Nederland Pagina : 39 Bereik : -1
cm2 : 319 Frequentie : incidenteel AVE : € 1650,00 Oplage : 15.000

column

Software als taal

Veel succesvolle softwareproducten gaan jarenlang mee. Tijdens hun leven worden ze gaandeweg aangepast en uitgebreid. Daardoor groeien deze stukken software (van slimme software van een kleine start-up, tot aan producten van online banken en zelfs de Belastingdienst) vaak uit tot ondoorgrondelijke en onbetaalbare draken. Hoe voorkomen we dat? En hoe repareren we het? Dit zijn de vragen die het onderzoeksgebied *software engineering* kenmerken: het wat, waarom en hoe van software.

Bij het CWI is het perspectief 'software is taal' van oudsher leidend. Met andere woorden: software is een communicatiemiddel tussen de ontwerper en de computer, maar ook tussen ontwerpers onderling. Dit perspectief heeft twee kanten. Betere talen kunnen tot betere communicatie (dus betere software) leiden. Daarnaast kunnen we bestaande software bestuderen als een soort tekst.

De ene kant betekent dat je het ontwerp van een programmeertaal kunt zien als de software-engineering-variant van het credo *form follows function* uit de industriële vormgeving. Domeinspecifieke programmeertalen – kleine talen waar je niet alles mee kan, maar wát je er mee kan, kan heel efficiënt – zijn de logische invulling van deze gedachtegang. Het CWI heeft vanaf het prille begin bijgedragen aan het wat, hoe en waarom van deze 'kleine talen.' Zo heeft het de programmeertaal Rascal ontworpen: een meta-programmeertaal die dit soort talen voor de gebruiker kan lezen, doorgronden en aanpassen. Wie de 'basistaal' Rascal machtig is, kan effectief en goedkoop gereedschappen voor legio kleine programmeertalen ontwerpen.

De andere kant betreft de miljarden regels broncode die intussen de wereld draaiend houden. De realiteit is dat er heel veel software is, hoe houden we die beheersbaar? Terwijl de wetenschap het antwoord op die vraag zoekt, moeten we ook andere prangende vragen beantwoorden. Bijvoorbeeld: hoe goed is mijn privacy gewaarborgd door bestaande softwaresystemen? Hoe kunnen we een onderdeel van een groots programma vervangen, zonder dat het systeem omvalt? Zulke vragen beantwoorden vraagt om geautomatiseerde gereedschappen. Software is vaak te groot en complex om het met de hand te doen. Door software als taal te zien, kan broncode automatisch ontleed, uitgeplozen en bevraagd worden. Ook met dit doel hebben we Rascal ontworpen. De computer kan ons zo helpen software beter te begrijpen en te onderhouden.

Software is onmisbaar in de huidige samenleving. Daarmee is software engineering ook een cruciaal vakgebied geworden. De grote moeilijkheid is dat in dit vakgebied korte- en langetermijnuitdagingen vechten om voorrang: de kwantiteit van ongebreidelde groei versus de bezinning op kwaliteit. Om uitdagingen het hoofd te bieden, én om te kiezen welke uitdagingen de hoogste prioriteit hebben, moeten verschillende disciplines samenwerken. Software is taal, en meer dan ooit moeten wetenschappers, industrie en beleidsmakers elkaars taal leren spreken om de kwaliteit van software te waarborgen.

Jurgen Vinju en Tijs van der Storm