

Toekomstverwachting Fundamentele Programmering.

Onder handen is een onderzoekproject naar de mogelijkheden om ons programmerend vermogen een orde van grootte te verhogen. De motivering voor dit onderzoek is gelegen in de observatie, dat de huidige State of the Art ten duidelijkste een gezonde methodologie ontbeert, met alle nadelige gevolgen van dien: "kleine" programma's worden met vallen en opstaan gemaakt, "grote" programma's ontaarden maar al te vaak in debacles. De reden, dat ik dit onderzoek ter hand heb genomen is omdat ik dacht in dezen mijn steentje te kunnen bijdragen, een hoop die op grond van aard, bezorgdheid, werkwijze, ervaring en in het verleden bereikte resultaten niet helemaal ongerechtvaardigd lijkt.

Ik ben nu zo ongeveer een jaar bezig. Het eerste half jaar is gaan zitten in een analyse van waarom programmeren zo vaak zo onmenselijk moeilijk wordt ("een catalogus van ons onvermogen") en een uitzoeken van wat we nog betrouwbaar kunnen zonder ons voorstellings- of overzichtsvermogen te zeer te overbelasten ("een catalogus van onze relevante vermogens"). Dit alles om een gevoel te krijgen, tot wat voor programmastructuren we ons in naam van de concipieerbaarheid zouden moeten proberen te beperken.

In het tweede half jaar heb ik allerlei programmastructuren geprobeerd, aanvankelijk toch nog erg tastend in het donker. (Het grote probleem bij deze experimenten was natuurlijk om ze zo goedkoop mogelijk uit te voeren, om "speelmateriaal" van net voldoende complexiteit te vinden, dat er iets van te leren viel!) Duidelijke aanwijzingen ter zake van een nuttige structuur hebben zich pas aangediend toen ik serieus recht liet wedervaren aan de eis van "aanpasbaarheid", dwz. dat elk groot programma (althans potentieel) in honderden aanverwante versies moet kunnen bestaan, zonder dat de hiervoor benodigde intellectuele investering evenredig met dit aantal stijgen mag. Sindsdien zijn mijn ervaringen uiterst bemoedigend: de eis van bewijsbaarheid, dat het programma correct is, evenals de eis, dat het programma aanpasbaar zij, blijken, wanneer ter harte genomen, de taak van de programmeur aanzienlijk te verlichten, omdat ze hem voor allerlei verwarrende kronkels behoeden.

De meest significante uitslag van de experimenten tot op heden is wellicht dat duidelijk aan het licht is gekomen dat programmeertalen, zoals die op het ogenblik en vogue zijn (inclusief al hun varianten) als programmeervehikel inadequaet zijn, wanneer we de eis stellen, dat we het programma zo willen formuleren als we het begrijpen en concipiëren kunnen. Er kleeft aan hen nog hetzelfde bezwaar dat ook kleefde aan de machinetalen, nl. dat het programma in wezen geschreven en begrepen moet worden op eenzelfde semantisch niveau. Als ik nu een ALGOL-programma bedenken, merk ik, dat ik het op de mij inmiddels aangeleerde "getrapte" manier concipieer en opschrijf -met grote winst aan tijd en betrouwbaarheid; maar wat ik dan heb opgeschreven moet ik "met de hand" zo goed en kwaad als het gaat in ALGOL vertalen. Aan den lijve heb ik ervaren, hoe riskant dit proces is; nu weet ik, hoe "onleesbaar" bv. ALGOL-programma's zijn en begin ik in te zien, waarom.

Wat begonnen is als het zoeken naar een geestelijke discipline "Hoe programmeren we voor een gegeven machine, resp. in een gegeven programmeertaal?" groeit naar de ontwikkeling van een nieuw "programmeergereedschap". Nu speel ik nog, maar er komt (hoop ik!) een moment, dat dit gereedschap er om vraagt om tot een voldoende compleet systeem voltooid te worden en tot een werkend systeem geïmplementeerd te worden. (En met "werkend" bedoel ik: met een realistische graad van performance: "the proof of the pudding is the eating", maar dat werkt echt niet, als je eens per maand een muizenhapje nemen mag!)

Op de mogelijkheden hiervoor kan ik op tweeneenhalve manier aandringen. (Naarmate het project volgroeit met meer klem en wellicht met drie.)

- 1) Het project streeft een uiterst praktisch doel na en zij realistisch. Zonder de realisatiemogelijkheden wordt ik gedoemd tot de rol van kamergeleerde (resp. kamerdromer), een bespiegelende functie, die ik -dacht ik- nog niet verdient. (Terzijde: dit is een mieters dubbelzinnige formulering!)
- 2) Als in mijn hoofd het gereedschap tot een voldoende graad van volledigheid ontwikkeld is, dan heb ik het nodig om studenten metterdaad te leren programmeren. Studenten leren om zich te behelpen met een stuk gereedschap, waarvan je de gebreken inmiddels kent is erg en deprimerend genoeg; op het moment, dat je weet, hoe je deze gebreken ondervangen kunt, wordt zulks de academie onwaardig.

3) (Dit is voorshands het halve argument.) Als het gereedschap conceptueel uit de verf is gekomen, dan is deszelfs implementering nog een aanzienlijk software project, dat wij zelf voor onze rekening zullen moeten nemen. De vraag is, in hoeverre via geraffineerde "bootstrapping" het stuk gereedschap in ontwikkeling zinnig gebruikt kan worden voor zijn eigen opbouw. Het antwoord hierop weet ik nog niet. Als het echter bevestigend luidt, ligt hier een ideaal werkterrein voor afstudeerders: beter kunnen ze het dan niet hebben!

Als de beloften, die dit project lijkt in te houden, vervuld worden, is het een project voor tenminste twee vaste medewerkers. We moeten proberen het zó te organiseren, dat meer mensen (vaste medewerkers, afstudeerders of misschien zelfs stageaires) hieraan kunnen meewerken, opdat zij ervan leren. Uit educatieve overwegingen is dit gewenst, over het bereikbare aantal is in dit stadium -nl. voordat een solide stuk ontwerpwerk gedaan is- nog geen verantwoorde uitspraak te doen. (Mijn persoonlijke neiging is om met kleine groepen te werken en, eventueel ten koste van de doorlooptijd, de hoeveelheid werk te minimaliseren. Weet ik enerzijds dat premature parallellisering tot debacles leidt, anderzijds weet ik ook (van dichtbij) dat het streven om parallele ontwikkeling mogelijk te maken, een ontwerp zeer ten goede kan komen. Ik zal mijn best doen.)

Als het project op gang komt en wil blijven, zullen de vaste medewerkers zich hier volledig aan moeten kunnen wijden en tegen de tijd, dat we een machine in onze werkzaamheden kunnen inschakelen, ongelimiteerd acces tot deze machine hebben.