

Over het gewicht van een informatica-opleiding.

Bij (niet geheel los van elkaar staande) besprekingen zijn bij herhaling vijf vakken als "typische informaticavakken" naar voren gekomen. Kortheidshalve duid ik ze aan met

- 1) programmeren
- 2) informatiebeheer
- 3) compilerbouw
- 4) operating systems
- 5) vergelijkende taalstudie.

Het kost mij bijzonder veel moeite, deze als "kernvakken", als nucleus van een academische opleiding te zien, omdat ik aan zo'n nucleus een dubbele eis wil stellen: enerzijds wil je, dat de vakken praktisch relevant zijn, anderzijds wil je dat ze theoretisch op een zo solide basis staan, dat ze een behoorlijke mate van duurzaamheid hebben dan wel methodologisch een zo rijke inhoud hebben, dat ze door hun algemeenheid een gevarieerd (en daardoor naar je hoopt: blijvend) toepassingsgebied hebben. Waar staan in dit opzicht deze vakken?

- 1) Programmeren scoort wat praktische relevantie betreft hoog. Er is enige relevante theorie en in de vier jaar dat ik het gegeven heb is een methodologie begonnen zich af te tekenen. Dit vak is tot een zodanige graad van wasdom gekomen, dat ik het in een academisch curriculum niet misplaatst vind. Het is vooralsnog een "klein" vak.
- 2) Informatiebeheer scoort wat praktische relevantie betreft hoog. Als we ook beveiliging tegen informatieverlies er in betrekken, wordt het bovendien erg moeilijk. De solide theoretische basis en duidelijke methodologische elementen wachten, dacht ik, nog op hun ontwikkeling. (Met o.a. de hoop hierop is Lunbeck aangetrokken.)
- 3) Compilerbouw scoort qua relevantie veel minder hoog. Het feit dat alle programmeren in hogere programmeertalen gebeurt, geeft compilerbouw nog geen centrale positie: dit wel een centrale positie geven is nu dezelfde denkfout, die mensen tien jaar geleden gemaakt hebben toen ze electrotechniek een centrale plaats gaven. Qua theoretische fundering scoort compilerbouw hoog (zelfs hoger dan nodig) waar het "parsing" betreft, maar deze theoretische

fundering betreft een erg specifiek aspect. Verder is "compiling a data processing job like any other". (Voor taalelementen en overeenkomstige structuur van object code, evaluatie van instructiesets etc. verwijs ik naar 5.)

4) Operating systems scoort qua relevantie (op het ogenblik nog?) iets hoger omdat zij zo'n diepgaande invloed hebben op de macroscopische bruikbaarheidsaspecten van computersystemen. De theoretisch fundering is pover. Het is illustratief voor hoe mensen worstelen met een amorf probleem met teveel haken en ogen, dat wel. Verder is het "a programming job like any other".

5) Vergelijkende taalstudie zou een discussie moeten zijn van taalelementen -parameters, pointers, records, types etc.- en hun operationele consequenties -zowel van de kant van de gebruiker als van de kant van de implementator. Dit is wat de taalontwerper zou moeten weten: hoogst relevant maar als "body of ordered knowledge" helaas nog non-existent.

De moraal is, dat er op deze manier bezien, eigenlijk nog niet zo heel veel is. Welke consequenties verbinden wij hieraan voor een curriculum? Ik zie vier mogelijkheden.

A) Men laat de praktische relevantie als criterium vallen en vult het curriculum met wat ik collectief als "automatentheorie" aanduid. Dit geeft aanleiding tot een zuiver wiskundige opleiding, naar traditioneel academisch vooroordeel alleszins respectabel maar onverenigbaar met ons streven naar bruikbaarheid. (Mag is dit "de Utrechtse school" noemen?)

B) Men laat de gezonde theoretisch basis en methodologische inhoud, kortom de duurzaamheid als criterium varen. Dit geeft aanleiding tot een hogere beroepsopleiding die zijn duurzaamheid niet ontleent aan innerlijke kwaliteiten, maar aan de traagheid van de wereld om ons heen, die in de wurggreep van een te haastig in elkaar gestampt verleden volhardt in het gebruik van obsoleete technieken. (Mag ik dit "de compatibiliteitsscholing" noemen?)

C) Men kan niet kiezen tussen A en B en neemt daarom half-om-half. Dit in de onverantwoorde hoop dat, als we zelf de synthese niet kunnen bereiken, onze studenten dit wel zullen kunnen. (Mag ik dit "de tweeslachtige opleiding" noemen?)

D) We gaan in wezen door op de ingeslagen weg van "accentsverschuiving" in onze opleiding voor wiskundig ingenieur, maar waken ervoor daarmee significant verder te gaan dan de beperkte "state of the art" rechtvaardigt. (En mag ik dit "de bescheiden opleiding" noemen?)

Dat ik een zekere voorkeur voor mogelijkheid D koester, kan voor de lezer die mijn woordgebruik geobserveerd heeft geen verrassing meer zijn. De reden is, dat D mij treft als het enige eerlijke voorstel, omdat A, B en C eraan voorbij gaan, dat het vak eigenlijk (nog?) heel klein is: ze treffen me als de onwaarachtige lading, waarmee men een voortijdig fier gehesen vlag probeert te dekken, elk is op zijn manier daarom een soort van bedrog. De kans, dat we op A terecht komen, lijkt me, gezien de constellatie van de onderafdeling, heel klein. Het gevaar, dat we op B of C (vooral B) terecht komen acht ik echter levensgroot aanwezig, omdat B op de bekende circumstantiele gronden zo voor de hand liggend lijkt.

De redenering, waarom je, als je D wilt, B moet doen, loopt ongeveer als volgt.

- 1) Om D te kunnen doen heb je knappe studenten nodig.
- 2) Om knappe studenten te krijgen moet je ze kunnen selecteren uit een grotere groep.
- 3) Om een grotere groep aan te trekken, moet je de informatica een zo duidelijk gezicht geven, dat ze als vliegen op de honing afkomen.
- 4) Het merendeel van deze studenten moet je op de praktijk voorbereiden, dus geef die dan B.
- 5) Als die enkele knappe niet van de praktijk op de hoogte is, krijgt hij later in het bedrijfsleven geen poot aan de grond, dus geef die knappe in elk geval ook B.
- 6) Ach, laat D dan maar varen, want (naar keuze)
 - 6a) de goede student komt er toch wel
 - 6b) het bedrijfsleven weet toch geen raad met goede mensen.

En ergens langs de lijn ben je vergeten, dat opleiding B op de keper beschouwd bedrog was.