

Het curriculum informatica aan de THE.

"Informatics must become a field of exceptional character at a new abstract level; the graduate of this curriculum must be a scientist of a class which has not existed until now: he must be an engineer by education and by mentality, but he must construct abstract objects which so far had been studied by mathematicians only. Mathematicians, however, study abstract objects for quite different purposes and in a different frame of mind."

(Professor Dr.Heinz Zemanek, "What is informatics", Management Informatics, Vol.1 [1972], No.2, page 43 - 52.)

Uitgangspunten.

1. Ongeacht het in de wet genoemde oogmerk waarmee een academische studie wordt gevolgd, dient de opleiding qua kennis, vaardigheden en habitus een gezonde basis the kunnen leggen voor een rijk en werkzaam verder intellectueel leven. Dit lange-termijn-aspect bestempelt o.a. snel verouderende kennis tot een suspect ingrediënt voor academische curricula.

2. Wetenschappelijke disciplines representeren hoe de kennis in de wereld over mensen is verkaveld: zij representeren, om een term uit de informatica te gebruiken, de "modularisering" van menselijke kennis. Opdat deze modularisering effectief zij, dwz. opdat een wetenschappelijke discipline levensvatbaar zij, moet tenminste aan de volgende vier voorwaarden zijn voldaan.

2.1 De voor de discipline onmisbare kennis mag qua omvang de capaciteit van een redelijk begaafd mensenhoofd niet te boven gaan.

2.2 De voor de discipline onmisbare vaardigheden mogen niet te veel, noch te moeilijk zijn: een redelijk begaafd mens moet ze zich eigen kunnen maken en ze op een voldoende hoog peil kunnen handhaven.

2.3 De discipline dient een grote mate van innerlijke coherentie te vertonen, zodat de kennis en de vaardigheden kunnen bijdragen tot wederzijdse ondersteuning en perfectionering.

2.4 De discipline dient zo "self-supporting" te zijn, dat zijn kennis bestudeerd en ontwikkeld kan worden, zijn vaardigheden aangekweekt en uitgeoefend kunnen worden, desgewenst in isolatie, in (tijdelijke) afzondering van de geestelijke rijkdom (en noden!) van andere gebieden.

3. Bij de vraag of een academisch curriculum in de informatica ingesteld moet worden, dient met het oog op het in 1. gestelde de existentie van een wetenschappelijke discipline, die aan het in 2. gestelde voldoet, als noodzakelijke en centrale voorwaarde beschouwd te worden.

3.1. Het kompas dat zich richt op polen zoals "huidige werkkringen in de automatisering" is voor richtingsbepaling bij het opstellen van een curriculum in de informatica onvoldoende betrouwbaar: nu juist deze polen zelf vrij drastisch lijken te gaan verschuiven, is dat kompas zelfs voor kortetermijn-planning nauwelijks bruikbaar, laat staan voor de lange-termijn-planning die met elk weloverwogen curriculum voorstel dient samen te gaan. (Dit alles nog afgezien of de expliciete afstemming op "werkkringen" bij het academisch onderwijs --in tegenstelling tot het hoger-beroepsonderwijs-- wel zo op zijn plaats is.)

4. Als een kernopgave van de informaticus beschouwen wij het beheerst ontwerpen en ontwikkelen van (zonodig uiterst geraffineerde) systemen, die een welgekozen doel op aantoonbaar betrouwbare wijze zullen realiseren. De wetenschappelijke discipline "informatica" bestrijkt de voor deze opgave onmisbare kennis, inzichten, theorieën, methodologieën en vaardigheden.

4.1 Een analyse van deze taakstelling impliceert dat de informatica een discipline van uitgesproken wiskundige signatuur moet zijn: zonder als wiskundige te werk te gaan, zal de informaticus nimmer de "aantoonbare betrouwbaarheid" kunnen bereiken. Elk informatica curriculum dat (om wat voor reden dan ook) dit wiskundig aspect verdoezelt, is dan ook tot falen gedoemd.

4.2 Een analyse van deze taakstelling impliceert dat informatica een discipline is met uitgesproken ingenieursaspecten, al was het alleen maar vanwege het type contractuele verplichting die de informaticus aangaat en vanwege de wijze waarop hij door de fysische realiteit gestelde grenzen dient te respecteren.

\*            \*            \*

Met het oog op het bovenstaande is de THE --uiteraard ook onder invloed van locale omstandigheden-- gekomen tot een curriculum voorstel voor een afstudeerrichting "informatica" voor wiskundig ingenieurs, een curriculum dat thans grotendeels is gerealiseerd of zich in staat van actieve voorbereiding bevindt. De beslissing is genomen op grond van ons inzicht dat de informatica zich inmiddels als voldoende rijke, coherente en isoleerbare discipline begint af te tekenen om zulk een curriculum te kunnen schragen. Het spreekt vanzelf, dat hierbij ruimte gelaten wordt voor accentsverschuivingen, die op grond

van didactische ervaringen of verdere ontwikkelingen van de discipline noodzakelijk kunnen blijken. (Aan de onderstaande nummering dient geen betekenis toegekend te worden.)

1. Inleiding tot de kunst van het programmeren. (2 \* 2 semesteruur).

De bedoeling is op basis van een intuïtief geïntroduceerde semantiek de student te leren heldere en effectieve programma's te concipiëren; een college met sterk methodologische inslag.

2. Toegepaste logica. (2 \* 2 semesteruur).

De bedoeling is de student vertrouwd te maken met een aantal formele technieken van standing, welke onmisbaar zijn als men het vak op solidere basis wil bedrijven; een fundamenteel college.

3. Inbeddingsproblematiek. (1 \* 2 semesteruur).

De bedoeling is de student een gedeelte van de kennis en inzichten bij te brengen om hardware --in het bijzonder adressering in de ruimste zin des woords-- op adequaatheid te kunnen taxeren en de taak van z.g. "operating systems" gestalte te geven; een meer encyclopedisch college.

4. Representatietechnieken. (1 \* 2 semesteruur).

De bedoeling is de student bekend te maken met de representatietechnieken voor lijsten, bomen, grafen, ruimtelijke objecten etc., in het licht van de wijze waarop zij moeten worden gewijzigd en geraadpleegd; een methodologisch/encyclopedisch college.

5. Syntactische technieken. (1 \* 2 semesteruur).

De bedoeling van dit college is de student te leren zijn talen zo te ontwerpen dat mechanische tekstverwerking --"parsing", "compile time checking", "error recovery" etc.-- geenodeloze complicaties oproept; een disciplinerend college.

6. Axiomatisering van de semantiek. (1 \* 2 semesteruur).

De bedoeling is de student vertrouwd te maken met axiomatisch gedefinieerde semantiek als basis voor formele correctheidsbewijzen en een calculus voor de formele afleiding van programma's; een fundamenteel/practisch college.

7. Informatiebeheer. (1 \* 2 semesteruur).

De bedoeling is de student vertrouwd te maken met de (zowel logische als technische) problematiek die optreedt bij het bespelen van grote gegevensbestanden; een meer encyclopedisch college.

8. Parallellisme en communicatie. (1 \* 2 semesteruur).

De bedoeling is de student vertrouwd te maken met de moeilijkheden en mogelijkheden van de organisatie van de samenwerking tussen sequentiele processen; een fundamenteel college.

9. Computer architecture. (1 \* 2 semesteruur).

Nog niet nader gespecificeerd.

Bovenstaand curriculum is ontworpen omdat dit de enige weg was die wij zagen om als informatici onze wetenschappelijke verplichtingen jegens de maatschappij na te komen. Ter onderstreping van de urgentie en aard dezer verplichtingen eindigen wij met een tweede citaat, ditmaal van professor Andrei P. Ershov uit Novosibirsk:

"In my opinion programming is, for at least the following three reasons, the most humanly difficult of all professions involving numbers of men:

1. Programmers constitute the first large group of men whose work brings them to those limits of human knowledge which are marked by algorithmically unsolvable problems and which touch upon deeply secret aspects of the human brain.

2. A programmer's personal push-down stack must exceed the depth of 5-6 positions, which psychologists have discovered to characterize the average man; his stack must be as deep as is needed for the problem which faces him, plus at least 2-3 positions deeper.

3. In his work, the programmer is challenged to combine, with the ability of a first-class mathematician to deal in logical abstractions, a more practical, a more Edisonian talent, enabling him to build useful engines out of zeros and ones, alone. He must join the accuracy of a bank clerk with the acumen of a scout, and to these add the powers of fantasy of an author of detective stories and the sober practicality of a businessman. To top all this off, he must have a taste for collective work and a feeling for the corporate interests of his employer."

(A.P. Ershov, "Aesthetics and the Human Factor in Programming." Comm.A.E.M. Vol. 15, No. 7 (July 1972), page 501 - 505.)