

Kom
verder



Sense and nonsense of sensing

Een terugblik

Uitgave ter gelegenheid van het afscheid van
drs. Henk van Leeuwen, lector Ambient Intelligence

saxion.nl/designtechnologie

Inhoudsopgave

1. Intro	4
2. Wat doet een informaticus?	6
Oorsprong van het vakgebied	6
Waar houdt een informaticus zich mee bezig?	7
3. Karakterisering van systemen	8
Tweedimensionaal schema	8
Interactie en persoonlijk gebruik worden dominant	9
Interactiviteit tussen apparaten	14
Sensing	14
Internet of Things	19
Reflectie op de assen van vrijheid en interactie	22
4. Programmeren	23
Stelling	23
Belang van software in onze maatschappij	23
Essentie van softwarebouw	24
Visualisatie van de problematiek	24
Software is inherent complex	25
Agile softwarebouw	26
Bibliotheken	27
Programmeren en digitale overvloed	27
5. Het opleiden van informatici in het hbo	28
De ontwikkeling van het hbo-informaticaonderwijs	28
De infrastructuur	30
Onderzoek om onderwijs te vernieuwen	31
Het hbo verlaat het vo: eigen leerplannen per hogeschool	31
RTES	32
Mobile Development	32
Programmeren	32
Vergelijking van leerplannen	33
Nieuwe thema's	37
Samenwerking of isolatie	38
Landelijke congressen over informaticaonderwijs	39
6. Onderzoek in het hbo	40
Onderzoek binnen de informatica	40
SERTES	40
DevLab	41
Ambient Intelligence	42
Kenniscentrum Design en Technologie	43
IIP Sensornetwerken HBO	43
Tech For Future	44
Topsector HTSM en ons lectoraat	44
Verstevinging onderzoek	44
7. De rol van onderzoek in het onderwijs	47
8. Terugblik en conclusies	52
9. Met dank aan	54
Appendix 1	56
Lestabellen 1971 – 2013	56
Referenties	64

1. Intro

Ongeveer 45 jaar geleden begon ik me in de automatisering te verdiepen door het lezen van het boek 'De nieuwe wereld der Automatie' (Polak, 1966).



Dit boek beschrijft de maatschappelijke gevolgen van de automatisering. Het duurde nog vier jaar voor ik zelf leerde programmeren. In de taal Algol60 moest ik een probleem uit de numerieke wiskunde oplossen. Mijn serieuze belangstelling voor informatica als zelfstandig vakgebied naast de wiskunde werd pas later gewekt, in 1981. Maar vanaf die tijd is de informatica me

blijven boeien. Vanaf 1984 heb ik aan die interesse vorm kunnen geven binnen de opleiding Hogere Informatica (HIO) aan 'De Maere', een instituut voor hbo in Enschede. De opleiding is, na een reeks fusies, onderdeel van Saxion en doet het daar nog steeds goed. De laatste tien jaren is onderzoek een belangrijk aandachtsgebied binnen Saxion geworden. Binnen het lectoraat Software Engineering voor Realtime en Embedded Systemen, later omgedoopt tot Ambient Intelligence, heb ik mijn belangstelling voor de informatica en haar toepassingen kunnen verdiepen. Dit niet zozeer om mijn eigen nieuwsgierigheid te kunnen bevredigen, maar vooral om te kunnen bijdragen aan de ontwikkeling van de opleidingen Informatica en Technische Informatica en evenzeer aan bedrijven en organisaties die belang hebben bij nieuwe toepassing van ICT.

In deze uitgave kijk ik niet alleen terug op interessante zaken uit het verleden, maar probeer ik ook te onderkennen welke drijvende krachten invloed hadden. Hierbij heb ik niet gestreefd naar volledigheid. Het is een persoonlijke kijk, die berust op eigen ervaringen en die tot discussie kan prikkelen. Naast de vakinhoudelijke observaties neem

ik ook het hbo-informaticaonderwijs en het ICT-onderzoek onder de loep.

Het begrip 'sensing' in de titel 'Sense and nonsense of sensing' van deze uitgave heeft dan ook niet primair een technische betekenis. Sensing is in dit verband een manier van observeren, van snuffelen. In ons vak gebruiken we daar tal van sensoren voor. Over sensoren gaat het zeker, maar nog meer over mijn persoonlijk observeren, interpreteren van wat ik opmerk en mijn reflectie daarop. Dat leidt tot uitspraken over zin en onzin, sense en nonsense, van wat ik waarneem, nu en in het verleden. Ik neem u mee in de ontwikkelingen van informatica zoals ik die heb gezien en breng daarvan verslag uit. De vraag is of we uit de lijnen die we zien in het verleden, ontwikkelingen voor de toekomst kunnen afleiden.

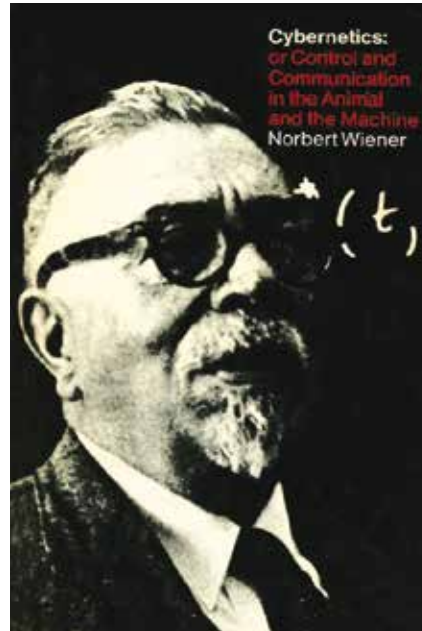
Henk van Leeuwen

2. Wat doet een informaticus?

Oorsprong van het vakgebied

Oorspronkelijk sprak men niet over informatica, maar over computer science. In het begin van het computertijdperk ging het erom digitale apparaten te bouwen die berekeningen konden uitvoeren. Rekenen was de kernactiviteit. Denk aan cryptografie en ballistiek in oorlogstijd. Iemand die de berekeningen zo kon opstellen dat de computer ermee aan de slag kon heette een wetenschappelijk rekenaar. Die term is niet meer in gebruik. Naast het rekenen was verwerking van data een taak die men door computers liet doen. Die berekeningen en dataverwerking waren van groot belang bij het aansturen van apparaten (Wiener, 1948). Daarbij werden complexe analoge meters, sensoren, gebruikt als input devices. Later werden rekenen en data verwerken gebruikt voor het aansturen van bedrijfsprocessen, zowel industrieel als administratief.

Wat wij nu heel gewoon vinden, moest grotendeels nog bedacht worden. De problemen in de begintijd lagen zowel op het gebied van de harde technologie van de elektronica, de apparatuur, als op zaken als de invoer van gegevens in de computer, de opslag van data, de programmeertalen, compilers, algoritmen en datastructuren, het zichtbaar



Norbert Wiener

maken van resultaten, de uitvoer. Enthousiaste programmeurs hebben daarvoor oplossingen gevonden. Eén van de steunpilaren in de wetenschappelijke en praktische ontwikkeling was, en is nog steeds, de wiskunde. Iemand die wij nu informaticus noemen, beschouwde men als programmeur. Zelfs de beroemde hoogleraar E. Dijkstra noemde zichzelf een 'humble programmer' (Dijkstra, 1972).

Waar houdt een informaticus zich mee bezig?

Alvorens me aan een terugblik te wagen, geef ik antwoord op vragen als 'Wat is een informatica?' en 'Wat doet een informaticus?'. Informatica kunnen we omschrijven als de wetenschappelijke en professionele discipline die zich bezighoudt met de opslag, verwerking en verspreiding van informatie. Informatici bedenken en realiseren systemen waarin slimme en veilige software een centrale rol vervult als oplossing voor problemen bij het omgaan met informatie. Het gaat dus zowel om de wetenschap als om de beroepspraktijk. Vandaar dat in het hbo informatica-opleidingen zijn

ontstaan, zelfs nog voordat de universiteiten in Nederland een formele informaticastudierichting kregen.

Het omgaan met informatie, is een discipline op zich aan het worden. Het ontstijgt elke andere discipline. Zo kunnen informatieproblemen uit de astronomie conceptueel met vergelijkbare methoden worden aangepakt, zoals in de biologie of psychologie. Informatica heeft twee gezichten; engineering en science.

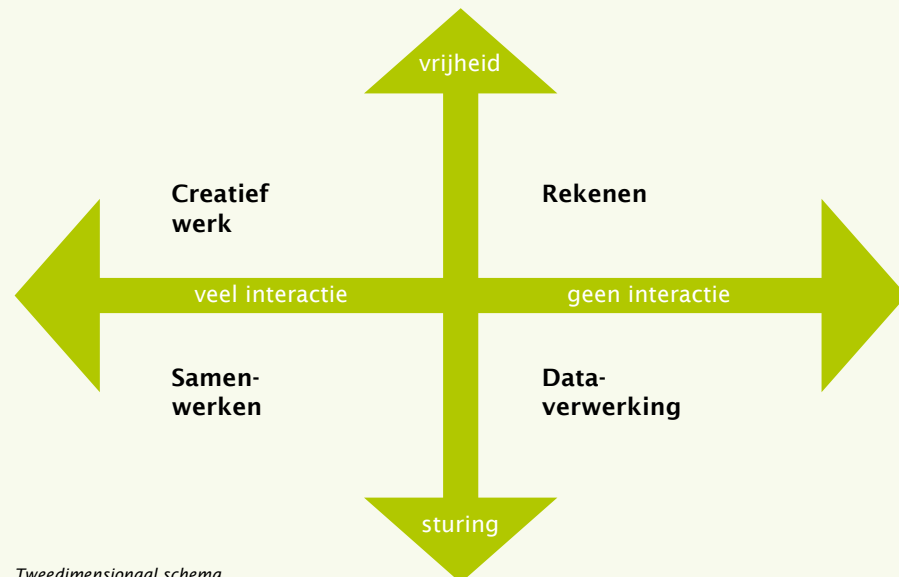
Na de terugblik op de ontwikkelingen van de afgelopen decennia kijk ik terug op de vraag of deze definitie van informatica die sterk gefocussed is op informatie, nog valide is.

3. Karakterisering van systemen

We kunnen het gebruik van computersystemen karakteriseren over twee assen. De ene as typeert de mate van vrijheid versus sturing en de andere as de mate van interactie, oftewel wederzijdse communicatie. Die twee assen zijn uitgewerkt door de oud-voorzitter van de gecommiteerden van de HIO A. Zijlker (Zijlker 1988, en Zijlker, 1989). Ik geef er een eigen interpretatie aan laat ook zien welke assen in de loop van de tijd aanvullend een plaats in het schema verdienen.

Tweedimensionaal schema

Rekenen staat in dit schema voor de vrije, wetenschappelijke ontwikkeling van algoritmen voor berekeningen. Denk aan het oplossen van differentiaalvergelijkingen, matrixberekeningen, statistische analyses en dergelijke. Bij dataverwerking staat het bedrijfsbelang centraal, waarbij het bedrijfsbelang sturing geeft aan de gewenste dataverwerking. In haar uiterste vorm is hierbij de vrijheid om het eens anders te doen verdwenen. Bij rekenen



Tweedimensionaal schema

en dataverwerking was interactie geen issue. Daarmee is de verticale as gekarakteriseerd door de mate van vrijheid versus sturing

Al snel ontstond de behoefte om interactiefwijzigingen in dataverzamelingen te kunnen aanbrengen. Eind jaren vijftig sprak men over 'interactive systems'. Die kregen een rol in klassieke verkoopsystemen, in eerste instantie strak geregeld en aangestuurd. Later is het gemeengoed geworden, in allerlei vormen, zoals ordersystemen en webshops. De tweede as, de horizontale, geeft de mate van interactie, tweezijdige communicatie weer. Aan de verspreiding van informatie van de ene computer naar een andere ontstond een grote behoefte en leidde tot datacommunicatie en computernetwerken.



Een ontwikkeling die in de jaren '80 de persoonlijke vrijheid in het computergebruik teruggebracht heeft, is de opkomst van de PC. Niet alleen klassiek computergebruik, zoals berekeningen maken (spreadsheet) en dataverwerking (databases), kwamen binnen handbereik van velen, maar ook persoonlijke interactie met de computer (de muis en een grafisch scherm).

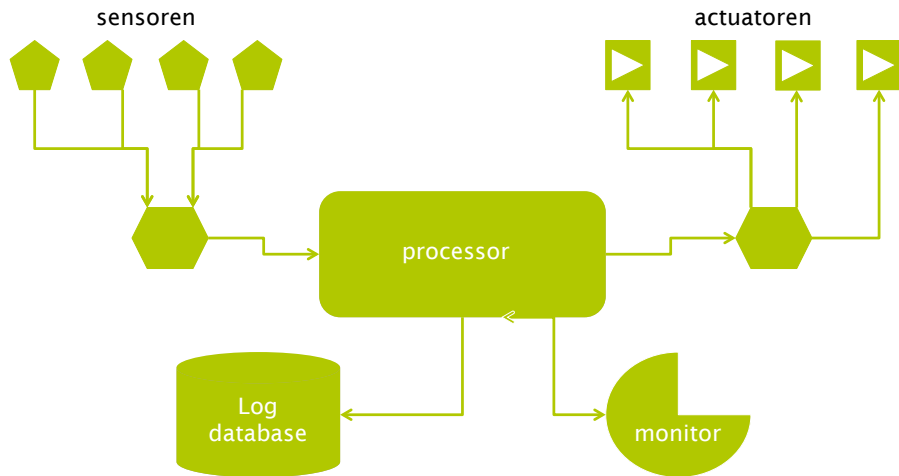
Bovendien biedt de PC de mogelijkheid eigen creativiteit vorm te geven in tekst, presentaties, grafisch werk, geluid en interactie met de omgeving, zodra sensoren aangesloten kunnen worden. De conclusie is dat met de PC de aanzet is gegeven tot een integratie van vrijheid en interactie.



Interactie en persoonlijk gebruik worden dominant

In de afgelopen twintig jaren hebben enkele ontwikkelingen sterk de aandacht getrokken. Deze hebben geleid tot wat we het Internet of Things (IoT) noemen.

Laten we beginnen bij embedded systemen. In de procesindustrie worden al sinds de jaren vijftig computers gebruikt voor de aansturing van industriële processen. De apparaten die dit verzorgen hebben als kern een embedded computer. De computer is daarbij geen op zichzelf staand apparaat, maar maakt onderdeel uit van een groter systeem.



Embedded systeem

Om de aansturing van het systeem te kunnen realiseren zijn in het systeem vier elementen onontbeerlijk:

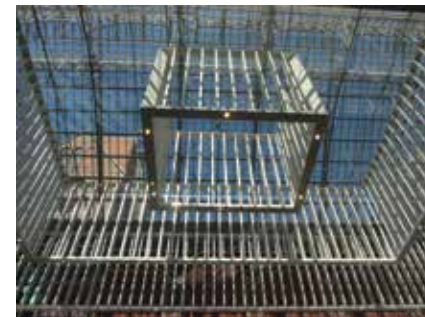
- Sensoren die de input leveren aan de embedded computer. De sensoren zijn de zintuigen die meten wat de toestand van een deel van het systeem is.
- Actuatoren voeren besturingsopdrachten uit. Deze sturen bijvoorbeeld kleppen aan, waarmee de flow van een gas of vloeistof wordt geregeld, ze regelen de spanning in een elektrisch circuit, ze sturen motoren aan, ze regelen de druk in een vat.
- Embedded software is een operationele beschrijving van het systeem en van de regels volgens welke het systeem bestuurd wordt.
- Processing of control unit is de specifieke embedded computer. Daar komen alle data vanuit de sensoren binnen. Actuatoren krijgen opdrachten op basis van de resul-

taten van de verwerking van de sensordata door de embedded software. De embedded software zal niet alleen rekening moeten houden met sensordata en systeemgedrag zoals het zou moeten zijn, maar ook met falend gedrag, zoals een motor die niet reageert. In ons onderzoek bij Océ naar software engineering voor copiers als embedded systeem kwam duidelijk naar voren dat engineers expliciet oog moeten hebben voor de error flow in een systeem. Het probleem van ongewenst gedrag heeft te maken met de samenhang tussen verschillende onderdelen en wordt versterkt door de strikte tijdseisen in real time gedrag. Deze factoren maken embedded software complex.

- Om achteraf een beeld te hebben hoe het systeem heeft gefunctioneerd, wordt het gedrag van het systeem vastgelegd in logbestanden.



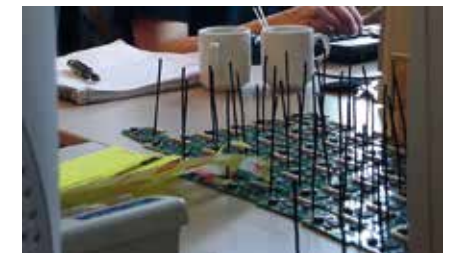
Een klassiek embedded systeem is gemakkelijk te plaatsen in het schema van interactie en vrijheid. Bij de uitdieping van het systeem kwam de inherente samenhang in het systeem en de invloed hiervan op het gedrag van het systeem sterk naar voren. Het gaat bij het ontwerpen van een systeem (ook niet embedded) niet alleen om een afzonderlijk element, maar



om de samenhang en daarmee om de architectuur. Het gaat ook niet alleen om gewenst gedrag, maar ook om het voorkomen van ongewenst gedrag. Informatici zijn, als het goed is, systeemdenkers. Zijn ze dat niet dan leveren ze slechts een beperkte bijdrage. Een ander aspect is dat complexiteit beheerst moet worden. Het reduceren van complexiteit vraagt zowel om het

vermogen tot abstractie, het weglaten van details, het oog hebben voor hoofdzaken, als om het doorzien van keuzes op detailniveau. Het helpt om problemen vanuit verschillende standpunten te beschouwen, zoals beide foto's van dezelfde dakconstructie laten zien. Welke problemen en kansen hier optreden behandel ik in het vervolg, als ik dieper inga op programmatuur.

Vanuit de beschreven embedded systemen maak ik een stap naar kleine onderling communicerend embedded systemen. Sensorsystemen zijn onderling gaan communiceren. Sensornetwerken zijn van groot belang voor het netcentrisch werken, waarbij mensen op verschillende locaties een gemeenschappelijk beeld moeten hebben van de actuele situatie. Bij de bestrijding van



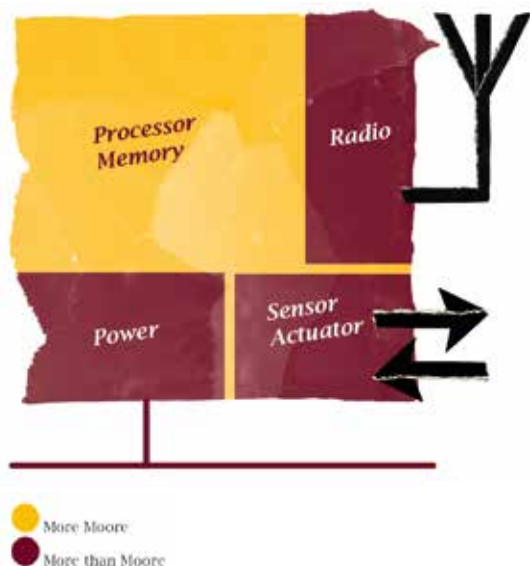
een ramp is dit van essentieel belang. Sensorsystemen zijn de laatste jaren steeds kleiner geworden, met afmetingen van enkele millimeters, bij voorkeur zelfs kleiner dan enkele micrometers. Samen vormen dergelijke systeemjes een (wireless) sensornetwerk. Elk systeemje is een node in het netwerk. Dat betekent dat elke node beschikt over een radio. Conceptueel is de radio zowel een sensor (ontvanger), als een actuator (zender).

De afmetingen zorgen voor schalingsproblemen in de zin van beperkte resources en kwetsbaarheid. Een groter probleem dat opgelost moet worden is het energievraagstuk. Veelal worden dit soort systemen niet gebruikt in een omgeving waarin elke node aan het stroomnet kan worden gehangen. De energie die een batterij levert is beperkt. Batterijen vervangen is meestal geen optie. Een ander vraagstuk is beveiliging. Dit soort systemen vraagt door hun kwetsbaarheid en hun beperkt geheugen en rekenkracht extra aandacht voor het inrichten van de security.

Communicerende sensorsystemen zijn interessant, maar nog boeiender wordt het als sensoren mobiel worden en het netwerk zich dynamisch configureert en als sensornetwerken bovendien gekoppeld worden aan andere netwerken.

De laatste tijd zien we dat niet alleen specifieke sensornodes een netwerk vormen. Ook smartphones kunnen als node optreden, zowel in een dedicated sensornetwerk als in een groot mobiel netwerk. Daarmee wordt sensing persoonlijk. De mobile smartphone kan zowel persoonlijke factoren (beweging,

positie, gedrag, spraak, foto's), als omgevingsfactoren (ondere andere geluid, licht, vochtigheid, radiosignalen) meten en doorgeven. Nieuw zijn de grote mogelijkheden die mensen krijgen doordat deze devices onderling communiceren en door het gebruik van sensoren. Door het persoonlijk gebruik vormen ze een grote databron voor anderen. Denk hierbij ook aan de bijna oeverloze communicatie via sociale digitale netwerken als Twitter en Facebook.



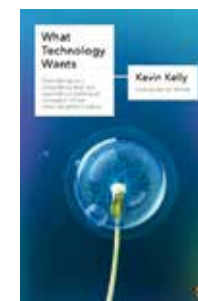
Intelligent systems need a Brain, but they also need Ears, Eyes, Arms and Legs.

Je kunt je afvragen welke technologische ontwikkelingen hebben geleid tot deze vergaande vorm van interactiviteit en persoonlijk gebruik. Verschillende technologieën als sensortechnologie, communicatietechnologie hebben hieraan sterk bijgedragen en vooral ook de

integratie van functionaliteiten op een chip (ENIAC, 2007). Zeer kleine processoren presteren voldoende om in allerlei alledaagse objecten te worden gebruikt en om van gewone dingen smart objects te maken. De processorkracht en geheugencapaciteit namen de afgelopen decennia exponentieel toe. Om smart te zijn moeten objecten kunnen reageren op veranderingen in hun omgevingen. Er is nog steeds een enorme ontwikkeling gaande om sensoren kleiner en betrouwbaarder te maken. Om interactiviteit mogelijk te maken is communicatie vereist. Datacommunicatie wordt, net als processing, steeds sneller. De integratie van technologieën leidt tot systems on a chip, SOCS. Nanotechnologie speelt op de achtergrond een grote rol. Daarnaast spelen sociale factoren een rol.

Al met al belanden we in een tijd van digitale overvloed. De schaarste aan computers, communicatiefaciliteiten en sensoren en actuators lijkt voorbij te zijn. De impact hiervan kunnen we nog niet overzien. De transitie van digitale schaarste naar digitale overvloed zal niet zonder rimpelingen gaan, maar de trend is duidelijk te herkennen. Een eerste effect is te zien in het Internet of Things. Ik ga ook in op enkele andere. De vraag rijst of er factoren zijn die schaars blijven? Interessant is dat de overvloed niet betekent dat systemen perfect zijn. Goedkope sensoren en publieke netwerken zijn bijna per definitie onvolmaakt. Hoe gaan we om met deze onvolkomenheden? Daarover later meer.

Welke sociale aspecten hebben een rol gespeeld bij de acceptatie van de nieuwe technologieën en nieuwe apparaten als smartphones? Zonder volledigheid na te streven en met de erkenning dat ik op dit gebied een leek ben, onderken ik een aantal factoren: comfort, veiligheid, de menselijke behoefte aan onderling communiceren, zakelijke belangen, economische groei en politieke factoren (Intel, 2009) en ENIAC, 2007). Of is technologie een drijvende kracht die min of meer autonoom is? Dit is een moeilijke vraag. Ik zie drijvende krachten zoals hierboven genoemd. De vraag is of die niet vooral van invloed zijn op het gebruik van technologie en minder op het ontwikkelen van conceptueel disruptieve, nieuwe technologie. Kelly gaat ervan uit dat technologie een eigen dynamiek heeft, die niet gestuurd wordt door mensen. In zijn boek 'What technology wants' (Kelly, 2010) licht hij



dit met overtuiging toe. Zijn motto is "Technology is a living force that can expand our individual potential - if we want to listen to what it wants". Of de vraag van sturing ooit objectief en bevredigend beantwoord zal worden is nog onbeslist.

Interactiviteit tussen apparaten

Het schema met de assen vrijheid en interactie is vooral gericht op de omgang van mensen met computer- en informatiesystemen. Je kunt het schema ook bekijken voor de interactie tussen systemen zonder directe menselijke tussenkomst. Het is duidelijk dat we dan in de linker kwadranten van het schema zitten. De vraag rijst of er naast directe sturing vanuit van tevoren opgelegde regels ook een bepaalde mate van autonomie een rol speelt.

Communicatie tussen machines (M2M) komt voort uit de behoefte aan remote monitoring, remote diagnostics, remote control en remote repair. De menselijke actieve betrokkenheid neemt daarin af. De ruimtevaart en het gebruik van satellieten hebben een sterke stimulans gegeven. Satellieten zijn niet alleen voor communicatie interessant, maar in de regel zijn het dragers van sensoren. De sensoren worden op afstand uitgelezen, continu, periodiek of event driven. Devices met hun sensoren en actuatoren hebben steeds meer een eigen IP-adres en zijn daarmee aangesloten op het internet. Ook bij sensornetwerken kan een speciale node als gateway het netwerk op het internet zichtbaar maken. Wat dit betekent zal ik onder het thema Internet of Things nader beschouwen.

Wat betreft het twee-assen schema stel ik vast dat de accentverschuiving naar de linkerkant duidelijk weer te geven is; de interactie is explosief toegenomen. Dat neemt niet weg dat rekenen

en dataverwerking belangrijk blijven, maar vaak onzichtbare elementen zijn van nieuwe geïntegreerde systemen. Bovendien betekent de enorme toename van data dat er een enorme behoefte ontstaat deze enorme berg van Big Data te interpreteren en te verwerken. Of dit beschouwd moet worden als dataverwerking in de klassieke zin is nog de vraag.

Niet in het schema weer te geven is de enorme groei van devices en de diversiteit van systemen, zowel qua functie, als qua inbedding in de omgeving (pervasive en ubiquitous systems). Daarnaast is de aard van de interactie dermate gevarieerd geworden dat daarvoor een andere as nodig is.

In de volgende paragrafen neem ik enkele thema's nader onder de loep. Als eerste komt Sensing aan de orde, gevolgd door de opkomst van het Internet of Things, kortweg IoT. Het zal blijken dat voor Sensing en het IoT software een essentiële rol speelt. Programmatuur is niet alleen de lijm die de samenhang geeft, maar bepaalt de interpretatie van sensordata door deze aan andere databronnen te correleren en daar in samenhang analyses op uit te voeren. Het ligt dan ook voor de hand dat ik nader inga op de rol en bouw van programmatuur. Dat doe ik verderop in deze uitgave.

Sensing

Het waarnemen van dingen in onze omgeving doen we met onze zintuigen. Sensoren werken conceptueel als onze zintuigen. Neem het oog. Licht valt via de lens op het netvlies. De kegeltjes

en staafjes nemen veranderingen waar en geven prikkels door aan de oogzenuw. Om te kunnen zien hebben we meer nodig. De zenuw geeft de prikkels door aan het gezichtscentrum in de hersenen. Daar vindt uiteindelijk het zien plaats en worden we ons ervan bewust dat we iets zien. De afbeeldingen van Paul Klee en Van der Leek hieronder maken duidelijk dat we onvolledige beelden met onze hersenen interpreteren zodat we een mens of een gezicht zien. Met de woorden van Johan Cruijff: "Je gaat het pas zien, als je het door hebt".



Verder vindt er in de hersenen interpretatie plaats van wat we hebben gezien. Daarna vinden er nog andere processen plaats, waardoor we reageren op wat we waarnemen met onze ogen. Onze reacties worden niet alleen door ons gezicht bepaald, maar ook door wat we waarnemen met onze andere zintuigen en door wat we al dan niet bewust weten over onze omgeving en onze situatie en hoe die zich kan ontwikkelen. Al met al kunnen we concluderen dat onze zintuigen slechts een element zijn in de keten van waarnemen. We kunnen de zintuigen niet missen, maar als de rest van de keten

niet functioneert, nemen we nog niets waar. In die keten kan van alles mis gaan. Er kunnen foute waarnemingen zijn door fouten in het zintuig. Als de lens van het oog niet goed werkt, belemmert dat ons zien en kunnen we bijvoorbeeld niet lezen zonder bril. Als het netvlies beschadigd is, zien we niet of slechts een deel. Als we iets onbekends zien, hebben we moeite met het interpreteren van wat we zien. Ook treedt gemakkelijk gezichtsbedrog op. We denken iets te zien dat er niet is.

Deze beschouwing over onze zintuigen leert ons het nodige over het gebruik van sensoren:

- Sensoren maken deel uit van een keten in een sensing systeem.
- Sensoren worden steeds beter, maar zijn meestal niet volmaakt.
- Sensoren leveren data op, maar om te komen tot informatie is analyse en interpretatie nodig.
- Kennis ontstaat pas als we ook betekenis kunnen geven aan onze waarneming (semantische analyse). Ook willen we graag een inschatting kunnen maken wat de waarneming betekent voor ons handelen (projectie). Dat betekent dat we een inschatting maken van hoe de situatie zich ontwikkelt. Pas dan spreken we van situational awareness.

Sensing is zoals we gezien hebben essentieel voor embedded systemen om de interactie met de omgeving te realiseren. Ook in de begintijd van de computersystemen speelde sensing een rol om feedback in regelsystemen vorm te geven (Wiener, 1948). Daar kwam nog veel rekenwerk aan te pas.

Rekenwerk op computers die niet met reële getallen werken, maar met benaderingen, hebben een nieuwe tak in de wiskunde doen ontstaan: de numerieke wiskunde.

Het probleem van inzicht krijgen in een situatie door sensing is verwant aan de vraag hoe kennis over onze wereld ontstaat. De discussie ging in de zeventiende en achttiende eeuw over de vraag of kennis ontstaat uit waarneming door de zintuigen of vanuit de ratio; het redeneren. Uiteindelijk is dit probleem door Kant opgelost, door de argumentatie dat zonder ratio een waarneming niet te duiden is (Gude, 2010). Op de één of andere manier geven modellen over de werkelijkheid betekenis aan wat we waarnemen. Vandaar dat in het veelgebruikte sense-think-act paradigma van ambient intelligence de sensor-data door redeneren vanuit modellen en bestaande kennis geduid kunnen worden en via redeneren over de doelstelling en context tot een actie leidt.

Sensoren zijn gemeengoed geworden. Elke smartphone beschikt over minstens tien sensoren. In een formele definitie (IEEE 1451.2) bevat een sensor één of meer sensorelementen. Een sensorelement zet één vorm van energie om in een andere vorm. Een sensor wordt een smart sensor genoemd als er (applicatie) processing-, dataopslag- en communicatiefaciliteiten aan zijn toegevoegd.

De trend is dat sensoren steeds goedkoper worden, lichter en kleiner, terwijl de kosten dalen, de betrouwbaar-

heid, nauwkeurigheid en gevoeligheid toenemen en ingebouwd worden in mobiele devices en smart objects (Yurish, 2011). De energie-efficiëntie neemt toe, maar het energieverbruik is nog een grote uitdaging. Ik zie tegelijk ook de trend dat we veel goedkope, kleine en onvolmaakte sensoren in systemen accepteren. Het gevaar van gezichtsbedrog' ligt op de loer. Door analyses zullen we 'outliers', sensoren die waarden opgeven die niet correct zijn, kunnen detecteren.

Deze door Yurish beschreven trend is al een tijd zichtbaar en gaat door. Daardoor is het mogelijk goedkope sensorsystemen voor tal van nieuwe toepassingen te ontwikkelen. Denk onder andere aan monitoring van vitale lichaamsfuncties in gezondheidszorg, veiligheidstoepassingen en sport, aan alarmering in kritieke situaties en aan regelsystemen voor verkeer en landbouw.

Sensorsystemen bevatten meerdere sensoren. We zien toepassingen als

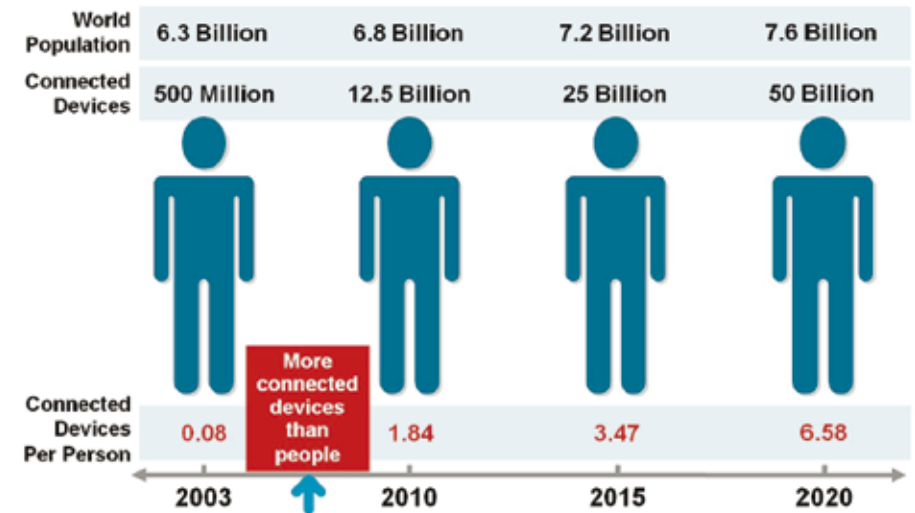


Sensorshirt

een Body Area Network, waarbij sensoren functies van het menselijk lichaam meten zoals hartslag, temperatuur, bloeddruk, ademhaling of andere functies die van vitaal belang zijn. Denk ook aan Personal Area Networks, waarin persoonlijke devices en sensoren zijn opgenomen. Maar er is ook sprake van sensornetwerken die een heel gebied, gebouw of een wegnnet monitoren. Denk aan de ijdijk in Groningen of aan sensornetwerken in een vliegtuig of auto. Sensoren en sensorsystemen maken steeds vaker deel uit van het internet. Een smart sensor met een eigen IP-adres kan via het web benaderd worden. Een lokaal sensornetwerk kan via een gateway (met eigen IP-adres) opgenomen worden in het internet. Zo is het sensorweb ontstaan, dat uitloopt op een Internet of Things. Een bijzonder invloedrijke ontwikkeling is de opkomst van smartphones. Voor veel mensen vervult de smartphone de rol van computer en communicatiedevice bij uitstek. Maar met een smartphone kan veel meer. De smartphone beschikt over een rijke set van sensoren, zoals bewegings-sensoren, camera, microfoon, kompas, GPS. De sensoren maken samen met de communicatie-, processing- en opslagfaciliteiten applicaties mogelijk op gebieden als veiligheid, gezondheid-zorg, milieu en mobiliteit. Deze applicaties zijn in principe gedistribueerd en ondersteunen niet alleen persoonlijke, maar ook groepsactiviteiten. Het aantal smartphones dat in 2013 wereldwijd verkocht wordt, benadert de miljard. In 2017 verwacht men wereldwijd 1,5 miljard smartphones

te verkopen. Deze aantallen onderstrepen de reeds vermelde overvloed aan processing- en sensingsystemen.

Nu er zoveel wordt gemeten met communicerende sensoren en voor de interpretatie gebruik gemaakt wordt van allerlei typen databronnen ontstaat de behoefte aan een architectuur voor de combinatie van sensing, analyse, interpretatie, communicatie en opslag. Waar gegevens worden verwerkt, lokaal bij de sensor, of centraal op een server, of gedistribueerd in het netwerk, wordt bepaald door de architectuur. Lokale verwerking is de enige optie als er geen communicatie mogelijk is. Maar ook als er wel communicatiefaciliteiten zijn, kan lokale verwerking zinvol zijn, om uit alle data alleen interessante data of afgeleide informatie te versturen. Zowel bij centrale als bij gedistribueerde verwerking worden gegevens van mogelijk diverse typen verzameld om daarmee meer nauwkeurige en meer volledige informatie te genereren. Gedistribueerde verwerking biedt voordelen die centrale verwerking niet biedt. Denk aan schaalbaarheid, grotere fouttolerantie, efficiënt gebruik van communicatie en het kunnen omgaan met dynamische veranderingen in het netwerk (Khan, 2004). Het combineren en correleren van heterogene en multimodale sensorinformatie is nog sterk in ontwikkeling. Om alle gedistribueerde elementen smart sensors en andere devices met elkaar een geïntegreerd geheel te laten vormen, zodat ze elkaars informatie kunnen ontvangen en interpreteren, is



Source: Cisco, IBSG, april 2011

middleware als een infrastructuurlaag nodig in het netwerk. Die laag biedt de mogelijkheid om dynamisch veranderingen aan te brengen in het netwerk, zoals nieuwe sensoren: het 'plug-and-play'- model (Cook e.a., 2009).

Internet of Things

Eind 2013 zijn circa 8,5 miljard devices aangesloten op het internet (Evans, 2011). Het merendeel van deze apparaten, 7,5 miljard, bestaat uit mobiele telefoons met een eigen IP-adres. De overige zijn machines en andere sensorapparaten. Een optimistische verwachting is dat in 2020 het aantal devices dat is verbonden met het internet rond de 50 miljard zal bedragen (Evans, 2011 en Feki, 2013). Zo verandert het karakter van het internet en het World Wide Web.

Waren de gebruikers van het internet eerst vooral mensen die consument-informatie op het internet zochten en lazen, later werden gebruikers ook producenten. Zij plaatsten zelf informatie op het net, in plaats van alleen maar iets op het net te lezen. Interactiviteit is verder toegenomen door sites als Facebook, Twitter en WhatsApp. Nu apparaten zowel hun data op het internet publiceren, als data van andere apparaten via het internet ontvangen en de verwerking ergens op het net centraal of gedistribueerd plaatsvindt, spreken we van het Internet of Things. Apparaten worden smart objects die reageren op hun omgeving. Die omgeving kan fysiek aanwezig of virtueel op het net zichtbaar zijn en zich elders bevinden. Informatie uit sensorsystemen kan wereldwijd verzameld, verwerkt en gepresenteerd worden. Fysieke grenzen zijn daarbij niet meer relevant.

Nu dit Internet of Things razendsnel vorm krijgt, zie ik niet alleen veel nieuwe toepassingen, maar treden er ook conceptuele veranderingen op. In (Kortuem, 2013) worden er enkele genoemd, zoals:

- het samenvloeien van het fysieke en het digitale domein;
- fysieke objecten worden first-class entiteiten op het internet;
- de continue toename van data en de waarde die eraan wordt gehecht. De verwevenheid van fysiek en digitaal op een schaal die met het Internet of Things mogelijk is, zorgt voor een nieuwe manier van omgaan met informatie en interactie. Sommigen beschrijven het als een inherent democratisch verschijnsel, waarin iedereen kleine simpele devices kan inbrengen die onderling communiceren en waarin elk device bijdraagt aan een groter geheel (Kortuem, 2013). Omgekeerd krijgt ook iedereen data van anderen ter beschikking. Waartoe deze ontwikkeling leidt laat zich nog maar nauwelijks beschrijven.

Er is discussie over de betekenis van wat een first-class object is. De term wordt vooral gebruikt in de context van programmeren. In (Abelson, 1996) krijgt een element een first-class status als deze als een variabele benoemd kan worden, als een argument kan worden overgedragen bij de aanroep van een procedure of als een resultaat van een functie opgeleverd kan worden (return value) en als deze opgenomen kan worden in een datastructuur. Als een element aan al deze

voorwaarden voldoet is het first-class en heeft het de minste beperkingen in het gebruik. Als een sensor een first-class object is betekent dit dat we sensoren als argumenten kunnen meegeven bij de aanroep van een functie die een nieuwe sensor als terugkeerwaarde oplevert. Zo kunnen operaties op sensoren worden gedefinieerd die het mogelijk maken eenvoudiger sensorwaarden te manipuleren. Denk aan virtuele sensoren die een aggregatie van sensoren representeert, of aan virtuele sensoren die staan voor het maximum of gemiddelde van een aantal sensoren. Een aanzet hiertoe is gegeven door het Open Geospatial Consortium in de Sensor Model Language (OGC, 2007).

Er is in publicaties over het Internet of Things veel aandacht voor de Things, de devices. Sensoren en smart objects zijn op dit moment voor velen beeldbepalend. Uiteindelijk zal evenwel het belang van data belangrijker zijn dan de devices. Big data is de term die in dit verband gebruikt wordt. Voor de wetenschap is de gigantische hoeveelheid data die op het Internet of Things verzameld wordt, enorm interessant om onderzoek over tal van onderwerpen uit te voeren, op een schaal die ongekend is. Ook het bedrijfsleven en de overheid hechten veel waarde aan de gigantische berg data. Het Internet of Things zal de interesse van overheden voor wat er gebeurt op het internet versterken.

Big data kent een reeks boeiende uitdagingen op de het gebied van

data-analyse, datamining, data-interpretatie, het afleiden van de betekenis, patronen en trends. Kunstmatige intelligentie zal ter ondersteuning van de analyses een belangrijke rol gaan spelen. Welke balans gevonden zal worden tussen vertrouwelijkheid, privacy en andere belangen is nog een open vraag. De één denkt aan een uitdaging, voor de ander vormt het een schrikbeeld.

De onderlinge communicatie tussen apparaten die geografisch sterk verspreid zijn, is ook voor de gewone burger interessant. Realtime gegevens, bijvoorbeeld over de locatie waar je huis, auto, over de gezondheid van familieleden, zijn in bepaalde situaties relevant of prettig om te weten. Al deze gegevens kunnen komen uit diverse databronnen, waaronder apparaten die via het internet communiceren.

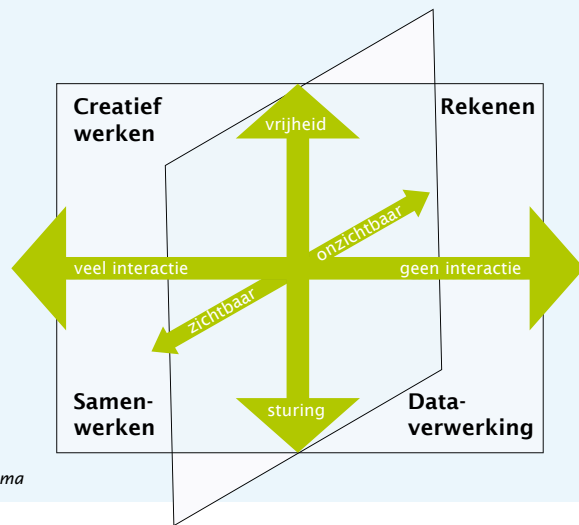
Naast de informatie en interactie speelt de beleving een belangrijke rol. Applicaties zullen gebruik gaan maken van visualisatie, geluid, spelvormen en ook ander zintuigen dan oog en oor aanspreken. Denk aan geur, beweging en haptiek. Het Internet of Things is een belangrijke enabler van ambient intelligence. Omgevingen worden zich bewust van de aanwezigheid van personen, hun gedrag of zelfs van intenties, en daarmee van bepaalde situaties, en kunnen daarop reageren. Tegelijk zijn de ICT-elementen onzichtbaar verweven met de dingen van alledag. De stelling van Mark Weiser wordt waargemaakt. De computertechnologie is zo fundamenteel

dat ze verdwijnt door op te gaan in de omgeving (Weiser, 1991). Daarom is het IoT voor ons lectoraat Ambient Intelligence van groot belang.

Veel bedrijven zien een nieuwe markt in het ontwikkelen van kleine programmeerbare apparaatjes als een vorm van halffabricaat. Mensen kunnen daarmee zelf smart devices voor het IoT bouwen. Naast het samenstellen van sensoren, actuatoren bij zo'n halffabricaat, zal er programmatuur geschreven moeten worden. Daarbij wordt er in de regel een programmeeromgeving meegeleverd. Net als in de begintijd van de computers zal een aantal zaken zelf bedacht moeten worden, maar de faciliteiten die nu geboden worden zijn veel rijker dan 60 jaar geleden. Deze kleine apparaten zelf verder uitbouwen en programmeren is een zeer leerzame activiteit. Vandaar dat er in veel opleidingen, niet alleen bij Informatica, mee geëxperimenteerd wordt. Of bij dat experimenteren door hobbyisten en niet-professionals de kwaliteit van de programmatuur op niveau blijft is een zorgpunt. Denk aan een mogelijke inbedding in een groter systeem, onderhoudsmogelijkheden en aanpasbaarheid. Voor de studenten die we opleiden ligt hier een mooie taak om goed ontworpen en uitbreidbare programmatuur te ontwikkelen voor toepassingen op het IoT.

Onderstaande lijst vermeldt een aantal programmeerbare devices die nu aangeboden worden:

- Arduino (www.arduino.cc) met shields (www.shieldlist.org).



Driedimensionaal schema

- Jennic (www.jennic.com).
- Mbed (www.mbed.org) met online compiler.
- Linux devices, onder andere Raspberry (www.raspberrypi.org) en BeagleBone (www.beagleboard.org).
- Microsoft .Net Gadgeteer (www.netmf.com/gadgeteer/).
- InBridge (www.inbridge.com).
- Electric Imp (www.electrimimp.com).



Hoe interessant deze ontwikkeling ook is vanuit het oogpunt van techniek, ik zie de aandacht terecht verschuiven; van de snelle technologische ontwikkeling naar nog te ontwikkelen kennis van en ervaring met de manier waarop en de mate waarin mensen kunnen omgaan met deze technologie. Het gaat daarbij om de interactie met smart things en om het menselijk inzicht in die interactie. Deze kennis helpt ontwikkelaars

applicaties en smart objects zo te ontwikkelen dat mensen ze effectief, veilig en prettig kunnen gebruiken.

Reflectie op de assen van vrijheid en interactie

Terugkijkend op het tweedimensionale schema zien we dat interactie een andere dimensie heeft gekregen. Interactie is niet meer alleen interactie van een persoon met een systeem en duidelijk zichtbaar via een user interface. Veel interactie in het Internet of Things vindt plaats tussen apparaten, is in eerste instantie aan het oog onttrokken en draagt bij aan informatie die uiteindelijk een rol speelt in de interactie met personen. De zichtbaarheid van de interactie is daarmee voor een groot deel onder motorkap van het internet en de servers in de cloud verdwenen. Interactie tussen mens en smart objects kan ontbreken, passief ondergaan worden of vorm krijgen in actieve deelname. Dat pleit voor het toevoegen van een derde dimensie aan het tweedimensionale schema van Zijlker: het al of niet zichtbaar zijn van interactie.

4. Programmeren

In het voorgaande heb ik ontwikkelingen gevolgd van grote, maar zeer beperkte computers in de jaren '40 van de vorige eeuw tot de kleine smart objects van vandaag. Dataverwerking was eerst een vervanging van klerkenwerk, nu spreken we van Big Data die geanalyseerd wordt; iets wat zonder computers onmogelijk is. Was interactie eerst iets wat op kleine schaal gerealiseerd kon worden door een persoon interactief met een computer te laten werken, nu is interactie multimodaal en intuïtief. We spreken nu over slimme omgevingen en pervasive systems. Hardware is op een enorme schaal geëvolueerd. Dat zien we ook bij sensoren. Er is een enorme diversiteit van sensoren ontwikkeld, die tegen een acceptabele prijs te koop zijn. In smartphones zitten er al tientallen. Maar wat geeft nu een meerwaarde aan sensoren? Aan sensorwaarden alleen hebben we niet zoveel. We zien dan hooguit een rij sensorwaarden, die ons weinig zegt. Software maakt het ons mogelijk die rij getallen te analyseren, ze te interpreteren in een context en er betekenis aan te geven, zodat we er iets mee kunnen (laten) doen. Zelfs onvolmaaktheden van sensoren kunnen worden gecompenseerd door software. In software worden modellen voor analyse en interpretatie gerepresenteerd.

Stelling

Ik poneer graag de stelling: Sensing zonder software voor de analyse en duiding leidt tot nonsense. Het is net zo zinloos als foto's maken en er totaal niets mee doen; niet delen, niet verzamelen, niet in een album opnemen, helemaal niks.

Belang van software in onze maatschappij

We zeggen dat de wereld digitaal wordt. Daarmee zeggen we ook dat software verweven wordt met onze omgeving. Zonder software staat de trein stil, rijden veel auto's niet, kunnen we niet internetten, werkt de beveiliging niet en zitten mensen op kantoor met de armen over elkaar zich te ergeren aan de computer die 'down' is. Zelfs onze vrije tijd vullen we met activiteiten waarin software een rol speelt. Dit geldt ook voor de manier waarop we met vrienden omgaan. Kortom, we leven bijna voortdurend in een omgeving waarin software-systemen aanwezig zijn. Soms duidelijk merkbaar, vaker nog onzichtbaar en onopgemerkt. Maar ze hebben wel een enorme impact, zowel op onze persoonlijke en maatschappelijke levenssfeer, als op mondiale politieke en economische schaal.

Nieuwetechnologische ontwikkelingen worden ondersteund door software en worden omgezet in producten waarin software verweven zit. Zonder ICT - dus software - geen innovatie.

Essentie van softwarebouw

De vraag die opkomt is hoe goed en betrouwbaar die software is waarvan we zo afhankelijk zijn. De statistieken stemmen niet optimistisch: in elke 1000 regels programmatuur zitten enkele fouten. En toch zijn we niet wanhopig, want we zijn eraan gewend geraakt. Maar hoe zit het dan met het schrijven van al die software? Kan dat niet beter?

Er is een bijzonder leerzaam artikel geschreven door Fred Brooks onder de titel 'No Silver Bullet' (Brooks 1987). Van de strekking van zijn betoog moet elke software engineer diep doordrongen zijn. Zijn verhaal is nog steeds actueel.



Fred Brooks

Brooks vraagt zich af: "Wat is de essentie van software engineering?". Hij onderscheidt essentiële taken en bijkomstige (accidental) taken. Onder essentiële taken verstaat hij de modelering van de complexe conceptuele structuren waaruit de abstracte software entiteit bestaat. Deze entiteit geeft aan hoe de software is opgebouwd en bestaat uit de dataverzamelingen, de onderlinge relaties tussen de verschillende data, de algoritmen en de aanroep van functies. De bijkomstige taken bestaan uit het weergeven van deze abstracte entiteiten in een programmeertaal en de afbeelding daarvan op efficiënte machinecode. Dezelfde abstracte entiteit kan op verschillende manieren worden geprogrammeerd en in diverse programmeertalen, omdat ze onafhankelijk is van de representatie. De representatie moet een correcte weergave zijn van de entiteit. Dit kunnen we weergeven in een schema met verschillende lagen.

Visualisatie van de problematiek

Op de bovenste laag (laag 1) gaat het om de zaak waarom het draait, het probleem dat we met programmatuur willen oplossen. Laag 2 daaronder staat voor de conceptuele structuren, die in de diepere laag 3 in een programmeertaal worden weergegeven. Een compiler beeldt de programmatuur van laag 3 af op assembler en machinecode. In die laatste slag worden optimalisaties uitgevoerd ten behoeve van snelle executie, met weinig beslag op geheugenruimte.



Programmeerlagen

Wat Brooks beweert is dat de eerst slag van laag 1 naar laag 2 de essentiële taak is. De volgende slagen zijn een representatie van laag 2. Deze vervolgslagen zijn niet onbelangrijk, maar zijn bijkomstig in de zin dat we hier te maken hebben met tijdgebonden technologieën, zoals programmeertalen, programmeergereedschappen, ontwikkelomgevingen en dergelijke.

De grote moeilijkheid van software engineering is dan ook niet het hanteren van een programmeertaal, maar zit in het begrijpen van het probleem dat in laag 1 wordt aangeduid, de specificatie daarvan, het ontwerp van de software entiteit en het testen van de betrouwbaarheid van de representatie. De grote fouten zijn niet de syntaxfouten, maar de fouten in specificatie, ontwerp en testen.

Software is inherent complex

Brooks heeft zich ook afgevraagd waardoor het komt dat software-ontwikkeling zo moeilijk is en zo snel leidt tot fouten in programmatuur. Hij concludeert dat complexiteit inherent is aan software. Deze complexiteit is essentieel en kan niet worden geëlimineerd door abstracties, hoe behulpzaam die ook zijn. Bovendien groeit de complexiteit als de programmatuur opschaaft. De hoeveelheid onderlinge relatie tussen structuren groeit en de interactie tussen elementen neemt toe. Deze groei is sterker dan lineair. De toename van complexiteit is niet alleen technologisch probleem, maar ook een management-probleem. De toename maakt het moeilijk het overzicht te houden en bedreigt daarmee de integriteit en security van de software.

Drie bijkomende problemen versterken de complexiteit:

1. Software moet goed aansluiten bij interfaces die al door anderen zijn bedacht.
2. Programmatuur moet voortdurend worden aangepast aan nieuwe eisen.
3. De onzichtbaarheid van software maakt het moeilijk en het visualiseren van software, hoe nuttig ook, biedt nooit voldoende inzicht in de complexiteit.

Hoewel Brooks de complexiteit als een essentieel probleem onderkent bij softwarebouw, zag hij hoopgevende mogelijkheden, waarvan enkele in de afgelopen jaren vruchten hebben afgeworpen. Enkele van zijn tips:

- Maak niet zelf software als je goede software kunt kopen.
- Omdat niemand vooraf precies en compleet de requirements van een modern softwareproduct kan specificeren, voordat er al enkele versies

zijn gebouwd en uitgeprobeerd, is het essentieel iteratief te ontwikkelen en prototyping te gebruiken. Hierdoor ontstaat een beter zicht op de requirements en dat draagt bij aan de essentiële kwaliteit van software.

- Ontwikkel incrementeel. Laat programmatuur groeien.
- Mensen zijn essentieel. Niet iedereen is een geweldig ontwerper. Identificeer de 'great designers' in je organisatie.

Agile softwarebouw

Terugkijkend op de trends in software engineering sinds Brooks artikel, zie ik dat incrementeel en iteratief ontwikkelen sterk is aangeslagen. Barry Boehm heeft een spiraalmodel beschreven waarin het conceptueel model van de software in een aantal iteraties groeit (Boehm, 1988). Ook met agile werkwijzen als Scrum (Kniberg, 2007) wordt gepoogd recht te doen aan de tweede

en derde tip. Ook is duidelijk dat mensen het verschil maken. Daarmee zijn betere ontwikkel- en testomgevingen niet zinloos, want de integriteit van de software kan daarmee effectief onderzocht worden.

Bibliotheken

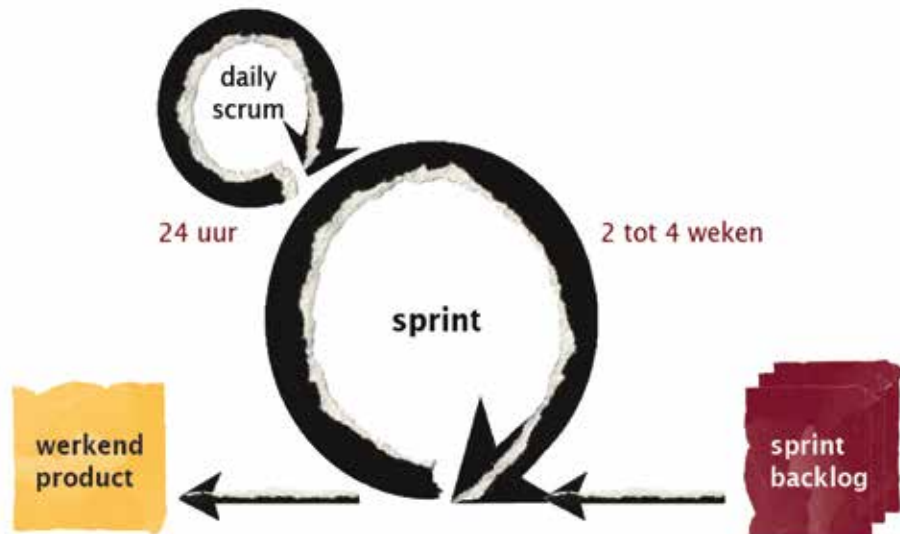
Programmeren heeft zich ontwikkeld doordat programmeertalen en programmeeromgevingen zijn veranderd. Er zijn nu veel bibliotheken met programmatuur beschikbaar. Die waren er dertig jaar geleden nog maar mondjesmaat. Onder UNIX konden system calls gebruikt worden. Bij Pascal was er nauwelijks iets als een bibliotheek. Nu zal een programmeur bibliotheken moeten gebruiken om productief te zijn. Dat vereist de vaardigheid en kennis om bibliotheken te vinden en te beoordelen, om een goede keus te maken die geschikt is voor de te schrijven programmatuur.

Programmeren en digitale overvloed

Een uitdaging die we nu, maar in de toekomst nog sterker, moeten aangaan is het omgaan met de overvloed van processing power. We hoeven niet meer zuinig te zijn. Dat kan betekenen dat er programmeertalen en programmeerconcepten gebruikt kunnen worden die nu nog als lastig of inefficiënt worden beschouwd. Denk hierbij aan functionele talen en parallelisme. Door gedistribueerde verwerking kunnen taken verspreid worden uitgevoerd. Dit komt zelfs binnen een computer voor als we kijken naar

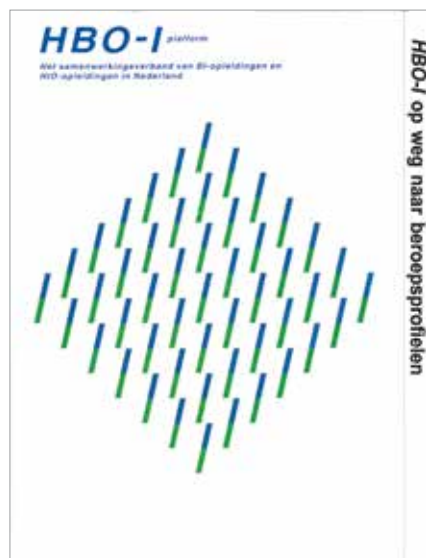
multicore processoren. Het is zaak de rekenkracht van de verschillende cores optimaal te benutten. Voor zover expressies onderling onafhankelijk zijn kunnen die op verschillende cores worden berekend. Voor een functionele taal liggen hier tal van mogelijkheden. De vraag is of een compiler dit geheel oplost, of dat een programmeur dit zelf moet aangeven. Dat laatste zou door een compiler overbodig gemaakt moeten worden.

Functionele en logische talen worden nog weinig gebruikt. Gelukkig zien we dat er in verschillende programmeertalen functionele concepten worden opgenomen. Talen krijgen een multiparadigma karakter. Een interessante vraag is welke programmeertalen we over twintig jaar zullen gebruiken. Worden multi-paradigma talen voor de standaard of zal er een nieuwe dominante taal opkomen?



verschillende randapparaten en de computers bouwden werden in de regel opgeleid in de opleiding Elektrotechniek. Een uitzondering was de opleiding Computertechniek, die in 1973 als enige opleiding in Nederland van start ging aan de Enschedese HTS.

In de periode 1987 tot 1990 breidde het aantal informaticaopleidingen in het land zich snel uit. Deze uitbreiding liep parallel met het ontstaan van grote hbo-instellingen. De wet op het hoger beroepsonderwijs, die in 1986 werd ingevoerd, heeft veel invloed gehad. Doordat een landelijk kader ontbrak voor het overleg tussen de informaticaopleidingen, is vanuit de opleidingen zelf het initiatief genomen tot samenwerking. Het project Kwaliteitsverbetering Informaticaonderwijs (KIO) is in 1992 uitgevoerd en heeft de stoot gegeven tot een permanente overlegstructuur die na enkele jaren



vorm heeft gekregen in de stichting HBO-I. De samenwerking heeft met een reeks van publicaties geleid tot een gemeenschappelijk kader voor de competenties die studenten verwerven tijdens hun studie Informatica in het hbo.

De infrastructuur

In de beginjaren had de HIO Enschede nog geen eigen computersysteem. Programma's werden via een datalink geupload naar een mainframe in Groningen. De output kwam via dezelfde lijn terug en werd in de HIO afgedrukt. Er is veel discussie geweest over het besturingssysteem dat op de toekomstige eigen computers van de opleiding moest draaien. Uiteindelijk is de knoop doorgehakt en werd het UNIX op een PDP11 van Digital Equipment. Daarmee wist de opleiding zich niet alleen te onderscheiden van andere opleidingen, maar belangrijker was, dat de studenten inzicht kregen in de functionaliteit en structuur van een modern interactief besturingssysteem en hands on daarmee leerden werken. Het duurde nog tot 1985 tot de opleiding voor alle leerjaren twee Vax 11-750 met UNIX als operating systeem kreeg. Daarmee werd de



datalink met Groningen overbodig. Enkel jaren later werden SUN-workstations in gebruik genomen met een compleet lokaal netwerk. Maar daarvoor was al veel ervaring opgedaan met lokale netwerksoftware.



Onderzoek om onderwijs te vernieuwen

In de jaren '80 kwam er veel meer aandacht voor computernetwerken. Lokale netwerken waren in opkomst. De HIO Enschede heeft daarmee een bijzondere plaats verworven, door uitvoering van het MANUS-project (Florijn, 1986). MANUS stond voor het Maere Netwerk van Unix Systemen. Daarin ging het erom de software voor een eigen lokaal netwerk van de basiselementen af aan te ontwerpen en te bouwen. Daarvoor was subsidie binnengehaald en met dat geld kon een groep docenten en oud-studenten de kennis op dit gebied verkennen, uitbreiden en in de opleiding brengen. Met MANUS ontstond in het onderwijs een experimenteeromgeving voor studenten, die landelijke bekendheid kreeg (De Ridder, 1984). In die tijd werkte ik aan SPIDER, een onderzoek naar gedistribueerde beheerssystemen.

Doordat SPIDER in 1983 MANUS tegen het lijf liep, ontstond het contact tussen de HIO en mij.

Dit MANUS-project was onderzoek avant la lettre in het HBO. Dit project heeft veel spin-off gehad en leidde ertoe dat ook op andere gebieden onderzoek gedaan werd door docenten. Denk aan de methode van informatieanalyse NIAM in overleg met G. Nijssen evenals de tooling voor NIAM in het onderwijs, het Epilog-systeem. Denk ook aan de invoering van object georiënteerd programmeren en de contacten daarover met Bernard Meijer. Ook was er contact met de groep van Andrew Tanenbaum aan de Vrije Universiteit over Amoeba, een gedistribueerd operating systeem.

Het hbo verlaat het vo: eigen leerplannen per hogeschool

In de tweede helft van de jaren tachtig werd de Wet Hoger Onderwijs van kracht. De hbo-opleidingen werden losgekoppeld van het voortgezet onderwijs (Van Bommel, 2006) en kregen met de invoering van de WHO de vrijheid hun eigen leerplan voor de opleidingen te formuleren. In die tijd ontstonden er veel initiatieven voor nieuwe informaticaopleidingen. Door de vrijheid de studieprogramma's zelf in te richten traden er veel veranderingen op. Zo kwam er meer aandacht voor software engineering. In de HIO Enschede beperkte zich dat tot analyse, ontwerp als aanloop naar het programmeren, en daarnaast tot het verkennen van het paradigma objectoriëntatie. De rol van software engineering werd versterkt in het begin van de jaren '90.

Toen werd vlak voor de stage een software engineering project ingevoerd waarin veel activiteiten gerelateerd aan het ontwikkelproces aandacht kregen. Denk aan gesprekken met opdrachtgevers, informatieanalyse, documentatie, testen, iteratief en projectmatig samenwerken en juridische aspecten. De integratie van leeractiviteiten in een project van ongeveer 10 weken werd een goede voorbereiding op de stage. Vanaf eind jaren negentig werd Software Engineering niet alleen een vak, maar ook een lijn in het onderwijs. De aandacht voor de processen en activiteiten die tot doel hebben valide en betrouwbare programmatuur en bijbehorende documentatie op een efficiënte manier te realiseren, nam een steeds een belangrijker plaats in.

RTES

In diezelfde tijd kwam in de informaticaopleiding het thema Realtime en Embedded Systemen sterk naar voren. Samen met Thales Nederland te Hengelo werd in de jaren 1999 tot 2001 een vernieuwingsproject uitgevoerd op dit gebied. Dat project werd gesubsidieerd door de Stichting Kennisontwikkeling, SKO.

Door de uitvoering van dit project werd het mogelijk embedded software engineering te onderwijzen aan eigen studenten, maar ook aan (nieuwe) medewerkers van bedrijven. Aan deze mensen was in die jaren grote behoefte bij bedrijven als Ericsson, ASML, Yacht, Mbedded, Thales, Atos Origin en Philips.

Toen het vernieuwingstraject afgesloten was, leefde er een sterke wens het onderzoek op dit gebied voort te zetten. Dat was de motivatie om in 2001 een lectoraat aan te vragen. Over dit lectoraat later nog meer.

Mobile Development

Rond 2010 werd opnieuw een belangrijke keuze gemaakt voor de actualisering van het programma: Mobile Development. Deze keuze werd ingegeven door de opkomst en populariteit van de smartphone en het idee dat de smartphone een aantrekkelijk en geschikt platform is om software engineering te leren. De opleiding beoogde ook met Mobile Development meer nieuwe studenten aan te trekken. Ook werd voor game programming een nieuwe studieroute ingesteld. De ervaring met Mobile Development in het onderwijs is positief. Door ook naar de systemen achter de smartphones te kijken, krijgen studenten inzicht in de netwerken en servers die mobile apps faciliteren. Hiermee heeft de opleiding de trend gevolgd die in de ICT als geheel te zien is: meer interactie en communicatie.

Programmeren

Een vraagstuk dat er nog steeds toe doet, is het onderscheid dat Fred Brooks maakte tussen de essentiële taken in softwarebouw en de bijkomstige zaken (Brooks, 1987). Lukt het de opleiding software engineers op te leiden die niet alleen goed de bijkomstige zaken als tools, programmeertalen goed beheersen, maar die ook en juist goed zijn in het opstellen van de concep-

tuele softwarekarakteristieken, zoals datastructuren, algoritmen in hun onderlinge samenhang en testbaarheid? Deze vraag is nog steeds actueel. Dat heeft alles te maken met de vraag of de goede software gebouwd wordt. En dat is meer dan de vraag of software goed gebouwd wordt. We zien hier het verschil tussen validiteit, 'de goede software', en betrouwbaarheid, 'de goed gebouwde software'. Bertran Meyer onderscheidt de externe en interne kwaliteitskenmerken (Meyer, 1988). De externe factoren betreffen het voldoen aan de requirements. De interne factoren hebben te maken met de manier waarop de software ontworpen en gerealiseerd is. Over de relatie daartussen merkt hij op "... internal factors are the key to ensure that external factors are satisfied". Aandacht voor goed programmeren blijft essentieel. Het is in dit verband interessant dat de opleiding veel ervaring heeft opgedaan met agile softwarebouw. Hierbij wordt software in een aantal iteratieve stappen incrementeel gemaakt. Per stap (sprint) van enkele weken worden resultaten getest en aan de klant voorgelegd. Dit voorkomt dat het product gaat afwijken van wat beoogd wordt. Deze werkwijze is gebaseerd op SCRUM (Kniberg, 2007). Bij de invoering ervan heeft het bedrijf Alten PTS ondersteuning geboden. In 2008 is er een onderzoek verricht onder onze studenten naar deze manier van software-ontwikkeling (Blom, 2009). Daaruit bleek dat studenten enthousiast waren over deze manier van ontwikkelen, maar ook dat zij moeite hebben user stories te formuleren en de werk-

last van een story goed in te schatten.

Vergelijking van leerplannen

Al jaren was ik er nieuwsgierig naar of de accenten in de leerplannen van de informatica-opleiding in Enschede de afgelopen veertig jaren zijn verschoven. Daarom heb ik een aantal leerplannen bekeken en van daaruit een inventarisatie gemaakt van de zwaarte van verschillende vakgebieden. Als directeur van de HIO, en later in andere rollen, ben ik jaren nauw bij de ontwikkelingen betrokken geweest. Nu, tegen het einde van mijn loopbaan, wil ik het overzicht over de afgelopen veertig jaar graag compleet hebben.

In de volgende tabellen geef ik een snapshot van de ontwikkeling van het leerplan door de jaren. Het was ondoenlijk alle leerjaren te bekijken. Uit wat ik beschikbaar had, heb ik gekozen voor een reeks van jaren met een onderling tijdsverschil van ongeveer tien jaren. Daarbij heb ik ook rekening gehouden met organisatorische veranderingen. Tot 1995 had de opleiding een grote mate van zelfstandigheid op het gebied van leerplanontwikkeling. Met de vorming van een instituut ICT, waarin alle drie informaticaopleidingen bij elkaar gebracht werden, is er geprobeerd enige samenhang in de leerplannen te bewerken. Dat leverde tegen de verwachting in niet veel meerwaarde op en heeft zich na verloop van enkele jaren vooral beperkt tot een paar gemeenschappelijke vakken met de opleiding Technische Informatica. Sinds 2011 is het leerplan met de keuze voor Mobile Develop-

ment nog sterker opleidingsspecifiek geworden.

Een andere factor was de maat die gehanteerd werd in de leerplannen. Tot eind jaren 1990 werd het aantal uren voor theorie en praktijk als eenheid gebruikt. Vervolgens werden het studiepunten van elk 40 uur studiebelasting. Enkele jaren later stond een studiepunt voor 28 uur studiebelasting. Om toch te kunnen vergelijken heb ik vanuit de verschillende eenheden het relatieve aandeel in het leerplan uitgedrukt in percentages per leerjaar. Zo zijn onderstaande tabellen ontstaan. Uitgebreidere tabellen met daarin zichtbaar de verschillende eenheden zijn te vinden in Appendix 1

Leerjaar1

	1973	1982	1994	2003	2013
Informatica	24,2	32,3	66	78	85
Wiskunde	30,3	35,5	26	17,1	5
Economie en Bedrijfskunde	27,3	19,4	8,1	0	0
Algemene vakken	18,2	12,9	0	4,9	10

Leerjaar2

	1973	1982	1994	2003	2013
Informatica	31,4	58,1	75,1	87,8	85
Wiskunde	25,7	19,4	11,6	0	0
Economie en Bedrijfskunde	25,7	16,1	5,2	0	0
Algemene vakken	17,1	6,5	8	12,2	15

Leerjaar3

	1973	1982	1994	2003	2013
Stage	100	100	100	50	50
Specialisatie (informaticadeel) en keuzevakken	0	0	0	42,9	35
Specialisatie (wiskundedeel)					5
Specialisatie (professionele. vaardigheden)	0	0	0	7,1	10

Leerjaar4

	1973	1982	1994	2003	2013
Informatica	64,7	58,1	46,7	42,9	25
Wiskunde	0	9,7	6,7	0	15
Economie en Bedrijfskunde	11,8	12,9	6,7	0	0
Algemene vakken	11,8	0	6,7	7,1	10,0
Afstudeerproject	11,8	19,4	33,3	50,0	50,0

De tabellen maken een aantal observaties mogelijk:

- Het aandeel van informatica is flink gestegen, vooral in de eerste twee leerjaren en juist niet in het vierde leerjaar.
- De aandacht voor wiskunde is flink gedaald.
- De economische vakken zijn geheel verdwenen.
- De stage is teruggebracht van een heel jaar naar een half jaar.
- Het afstudeerfase is uitgebreid tot een half jaar.
- De algemene vakken zijn eerst gedaald en later weer gestegen tot het niveau van 1973. Wie de tabel in de appendix leest zal zien dat het karakter van de algemene vakken veranderd is. Ze zijn nu meer gericht op de beroepspraktijk.

Het is de moeite waard na te gaan waarom de aandacht voor wiskunde en voor economische en bedrijfskundige vakken zo verminderd is. Over de wiskunde heeft Matthijs Langhout in 1996 een interessante analyse gemaakt (Langhout, 1997). Bovendien

is het interessant te onderzoeken wat het effect van de reductie van wiskunde in de opleiding is op de analytische vaardigheden van de afgestudeerden en of een brede blik op de context waarin een afgestudeerde werkt gemist wordt. Of heeft meer aandacht voor software engineering, projectmatig werken en modellering van software één en ander gecompenseerd?

Wat uit de tabellen niet naar voren komt is de verandering in onderwijs-werkvormen. Zo is het aantal projecten en projectmatig werken enorm toegenomen. Eerst alleen in het semester 4-project (jaren '90), later in de specialisatiethema's en minoren. Maar ook in de eerste leerjaren heeft, sinds pakweg 2003, het werken in projecten in elk semester een vaste plaats gekregen. Dit loopt van 25% in het eerste leerjaar op tot 40% in het tweede en zelfs 75% in de laatste leerjaren.

Nieuwe thema's

De opleiding Informatica zoekt met Mobile Development aansluiting bij de wereldwijde ontwikkeling van mobiele

ICT-platformen. Om in het derde leerjaarstudentendemogelijkheidtebieden zich te specialiseren heeft de opleiding gezocht naar representatieve thema's. Deze zoektocht heeft geleid tot vijf thema's:

- a. Big Data
- b. Internet of Things
- c. Enterprise Systems Architecture
- d. User experience software
- e. Programming tools en frameworks.

Hiermee laat de opleiding zien geavanceerde thema's onder de aandacht van de studenten te brengen en up-to-date kennis en vaardigheden in het leerplan te verankeren. Het is niet moeilijk een sterke onderlinge samenhang te zien en de aansluiting met thema's die de komende jaren veel invloed zullen hebben. Zo wil de opleiding haar vooraanstaande plaats in het informaticaonderwijs bevestigen

Samenwerking of isolatie

Binnen de Hogeschool Enschede kenden we begin jaren '90 drie informaticaopleidingen, twee in de sector Techniek (HIO en Computertechniek) en een in de sector Economie (bedrijfskundige Informatica).

Met het verschijnen van Een Nieuwe Agenda voor het InformaticaOnderwijs (Penning, 1990) kwam er een beweging op gang



om de drie opleidingen bij elkaar te brengen. Dat had veel voeten in de aarde, maar uiteindelijk werden de informaticaopleidingen in een nieuwe instituut ICT gebundeld.

Een stimulans voor versterking van de samenwerking tussen de informaticaopleidingen in het hele land zijn de visitaties geweest die sinds 1993 plaatsvinden voor de accreditatie van de opleidingen.



Door de gestructureerde samenwerking tussen de alle informaticaopleidingen in het hele land in het hbo-I is een beeld ontstaan van de beroepsprofielen waarvoor opgeleid wordt. In de periode 1992 tot 2000 heb ik daarin met veel plezier een rol kunnen spelen. Het hbo-I heeft in een reeks publicaties een kader voor de eindkwalificaties (Bordewijk, 2009) opgesteld, waarbinnen elke opleiding zijn eigen opleidingscompetenties heeft geformuleerd. Deze competenties zijn leidend bij de toetsing en examinering.



Landelijke congressen over informaticaonderwijs



Om bij te dragen aan de kwaliteit van het informaticaonderwijs draagt de stichting NIOOC er sinds 1990 zorg voor dat elke twee jaar een groot landelijk congres wordt gehouden.

Een sterk punt is dat op dit congres bijdragen vanuit het onderwijsveld worden gepresenteerd aan docenten en beleidsmakers in het informaticaonderwijs uit zowel het vo, hbo en wo. Ook vanuit het bedrijfsleven en bedrijfsopleidingen wordt geparticipeerd. De congressen worden als stimulerend ervaren. Er worden innovatieve ideeën gedeeld, die door anderen worden opgepakt voor de eigen opleiding. In 1999 en 2002 was de Hogeschool

Enschede, nu opgegaan in Saxion, de gastheer voor twee congressen. Als bestuurslid ben ik betrokken bij de stichting NIOOC. Dat levert een nuttig beeld op van wat er speelt in de opleidingen. Daarnaast ontstaan zinvolle contacten.

Er is nog een interessante ontwikkeling die zich de laatste jaren heeft aangediend, Het gaat om de plaats van onderzoeksvaardigheden in de opleiding. Voordat dit aan de orde kan komen moeten we eerst een andere belangrijke verandering behandelen.

6. Onderzoek in het hbo

Hogescholen hebben een steeds bredere opdracht gekregen. Niet alleen wordt beroeps onderwijs verzorgd binnen het hbo, maar ook onderzoek krijgt aandacht. Met het oog op die taak zijn niet alleen lectoraten in het leven geroepen, maar moeten ook studenten in de opleidingen vaardigheden verwerven om onderzoek te kunnen uitvoeren. Het gaat daarbij om praktijkgericht onderzoek. Dat betekent dat de problemen waarop het onderzoek zich richt uit het werkveld voortkomen. De toepassing in de beroepspraktijk staat centraal. Hiermee wordt duidelijk dat het niet om wetenschappelijk onderzoek gaat, maar om praktijkgerichte research. Denk aan onderzoek om innovaties te realiseren, nieuwe producten te ontwerpen, processen te herstructureren en werkwijzen te verbeteren. Dit betekent dat studenten leren om keuzes in hun beroepspraktijk te verantwoorden en de argumenten te onderbouwen.

Onderzoek binnen de informatica

Onderzoek in het hbo is niet iets nieuws, maar er was weinig aandacht voor. Binnen de opleiding Informatica werd er eind jaren '80 al voor gepleit enkele medewerkers ruimte te geven om onderzoek ten behoeve van onderwijs-

vernieuwing te doen. Onderzoek was toen geen vanzelfsprekende activiteit. Mooie voorbeelden zijn het MANUS-project (1983-1985), de voorbereiding van het project Object Oriented Design en Programmeren (1988-1991) en het vernieuwingsproject Realtime Embedded Systemen (1998-2001).

SERTES

Voor de informaticaopleidingen van Saxion lag het in de lijn om na het vernieuwingsproject een lectoraat aan te vragen, om daarmee continuïteit te bieden aan de problematiek van embedded software. Dit lectoraat, onder de naam Software Engineering voor Realtime en Embedded Systemen, SERTES, begon in 2002, onder de leiding van Theo de Ridder als nieuwe lector. De lijn die in de jaren negentig met veel Europese Esprit-projecten



gericht op geïntegreerde systemen en embedded software ingezet was, werd nu ook in het hbo zichtbaar. Vanaf het begin participeert Thales in het lectoraat door het beschikbaar stellen van een bedrijfslector.

Naast allerlei interne projecten werd op verzoek van Océ te Venlo onderzoek uitgevoerd naar de praktijk van software engineering in dit bedrijf. De vraag daarbij was hoe alle engineers



hun activiteiten beter konden uitvoeren, in lijn met de wens om betere kwaliteit te bereiken, onder andere door het toepassen van uitgebalanceerde oplossingsstrategieën en hergebruik van uitgeteste software voor deelproblemen. Dit was een prachtige kans om de industriële praktijk van embedded software van binnen uit te leren kennen. Het resultaat van het onderzoek was een cursus die in de periode 2003 tot 2007 vanuit Saxion aan alle software engineers van Océ gegeven is.

DevLab

Het bleef niet bij Océ. Toen binnen de FHI, de federatie van technische brancheorganisaties, het idee van een R&D-lab, het DevLab, voor mkb'ers



ontstond waren twee lectoraten, waaronder het SERTES-lectoraat van Saxion, hierbij nauw betrokken. In dit DevLab bedreven hbo-studenten samen met ondernemers, universitaire studenten en promovendi op een praktische manier onderzoek naar innovatieve technologieën. Daarbij werd niet alleen onderzocht of een concept gerealiseerd kon worden, maar ook of de gewenste doelen met het product bereikt konden worden. Dit product moest de ondernemers kennis opleveren die bruikbaar was in hun eigen productontwikkeling. Het doel was de marktkansen van de ondernemers te verbeteren en daarmee een bijdrage te leveren aan de economie van ons land. Er dreigde namelijk veel innovatief werk verplaatst te worden naar lagelonenlanden. Om die verplaatsing van werk en kennis te reduceren en een hele keten van kennis en innovatie in ons land te behouden en te verstevigen, is het DevLab opgericht.

In de periode 2005 tot 2007 heeft het lectoraat SERTES een belangrijke rol gespeeld in het DevLab, samen met het lectoraat Architectuur van Embedded Systemen van Fontys en enkele universiteiten. De goede ervaring die opgedaan was in het DevLab stimuleerde de

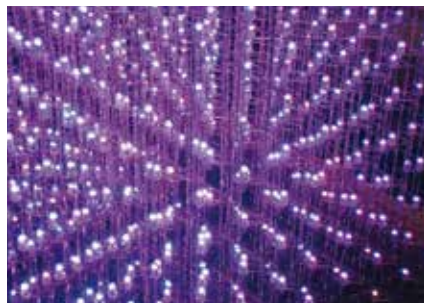


ambitie om intensief samen te werken met regionale bedrijven in Twente om te komen tot een vergelijkbaar lab, onder de noemer SaxLab of TwenteLab.

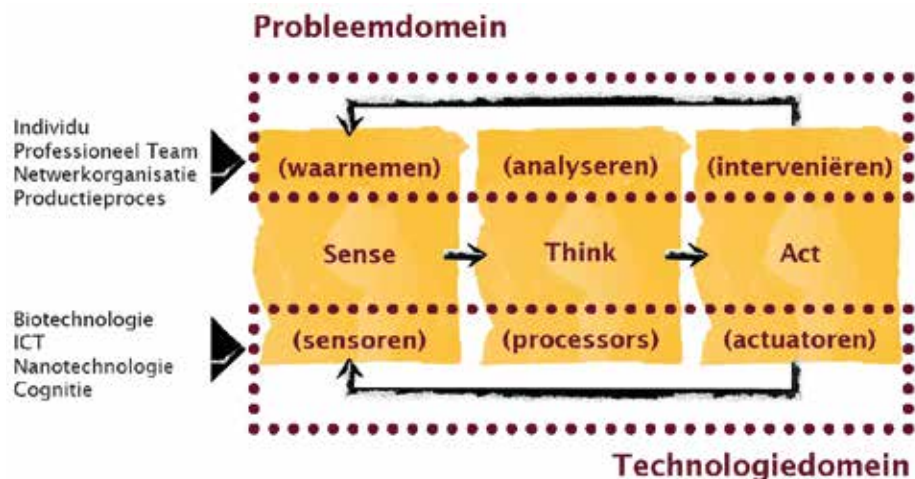
Ambient Intelligence

Ondertussen deden zich internationaal, en zeker in Europees onderzoek belangrijke nieuwe ontwikkelingen voor. Embedded systemen bleven niet beperkt tot de industrie en gingen deel uitmaken van de persoonlijke levenssfeer van mensen. In de leef- en werkomgeving drongen steeds meer kleine apparaten door, waarin sensoren en actuators een rol speelden. Het ging niet om losstaande elementen, maar om communicerende en samenwerkende apparaten. De verwezenlijking van de visie van ambient intelligence begon tot de mogelijkheden te behoren. Daarbij gaat het om meer dan techniek. De mens staat centraal. Het paradigma van Sense-Think-Act

kreeg invulling om het leven en werken van mensen veiliger, comfortabeler en productiever te maken. De technische invulling daarvan gebruikt vaste en mobiele sensoren, sensornetwerken en het internet als web van sensoren om te monitoren en waar te nemen. De intelligentie zorgt voor interpretatie, het geven van betekenis aan de data die sensoren opleveren en het scheppen van een beeld van de

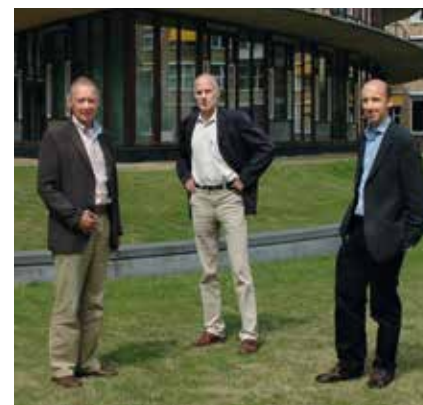


situatie (situational awareness). De intelligentie zit ook in de ondersteuning van het besluitvormingsproces, waar-



bij het erom gaat hoe de systemen of mensen het best kunnen reageren op de situatie. Daarvoor is een groot scala van interactietechnologieën beschikbaar. Denk aan gesproken berichten, beelden op schermen, SMS, het aanspreken van allerlei zintuigen, aanpassing van de omgeving en dergelijke. Mensen kunnen daardoor al dan niet bewust merkbaar geholpen worden. Gedragsbeïnvloeding kan daarbij een rol spelen. Hierbij dienen zich ethische vraagstukken aan.

Voor het lectoraat was het duidelijk dat de keus voor ambient intelligence ook niet-technische studenten de mogelijkheid zou geven in het onderzoek te participeren. Dat zien we de laatste jaren dan ook gebeuren. De kern blijft de technologie van ICT, maar de wijze van toepassen vraagt bredere inzichten. Daarover is meer te lezen in het boek dat ter gelegenheid van de lectorale redes van Wouter Teeuw, Piet Griffioen en mijzelf is geschreven onder de titel 'De computer verdwijnt, leve de computer!' (Van Leeuwen, Teeuw en Griffioen, 2010).



De heren Van Leeuwen, Teeuw en Griffioen.

Kenniscentrum Design en Technologie

Het lectoraat Ambient Intelligence werkt sinds 2006 samen met enkele andere technische lectoraten van Saxion in het Kenniscentrum Design en Technologie. Eerst vooral met de lectoraten Smart Functional Materials, Fashion Materials Design en Industrial Design. Later zijn daar Mechatronica en twee lectoraten rond Nanotechnologie bijgekomen. En zeer recent nog het lectoraat Lichtgewicht Construeren.



IIP Sensornetwerken hbo

Vanuit het lectoraat is op initiatief van het ICT Innovatie Platform Sensornetwerken, IIP SN, een landelijke overleggroep van lectoren en docenten in het hbo, in het leven geroepen om toepassingen van sensornetwerken in het hbo te stimuleren. Dit heeft geleid tot de publicatie 'Intelligente Sensornetwerken' (Van Leeuwen, Maas, Land, 2011), waarin wordt beschreven op welke manier lectoraten en opleidingen gebruik maken van deze nieuwe technologie. Vanuit deze groep worden vooral regionale initiatieven ondersteund. Ons lectoraat heeft al jaren samenwerking met Fontys en richt zich vanuit IIP SN nu sterk op het HIT (Hanze, Assen), de NHL (Leeuwarden) en Windesheim (Zwolle).

Tech For Future

Saxion heeft als focus voor haar onderzoek en onderwijs gekozen voor Living Technology. De sterke aandacht voor High Tech Systemen en Materialen, HTSM, in de regio heeft bijgedragen aan deze keus. Daarnaast vormt Saxion samen met de hogeschool Windesheim de twee Overijsselse Centres of Expertise: TechYourFuture, het Centre of Expertise TechniekOnderwijs, en TechForFuture, het Centre of Expertise High Tech Systemen en Materialen (HTSM) Oost. Voor ons lectoraat Ambient Intelligence biedt het onderzoeksprogramma in TechForFuture uitgelezen kansen om samen met andere lectoraten en opleidingen projecten uit te voeren die op verschillende toepassingen gericht zijn.

Topsector HTSM en ons lectoraat

Voor de verschillende topsectoren zijn roadmaps uitgebracht. Hoewel ICT niet een topsector op zich is, is er een wel roadmap gewijd aan ICT. Binnen alle topsectoren speelt ICT een belangrijke enabling rol waarzonder een topsector zich niet goed kan ontwikkelen. Dit geldt zeker voor de topsector HTSM. Deze sector is de hoeder van ICT. Vandaar dat vanuit HTSM de "Roadmap ICT for the top sectors" is geschreven (Lundqvist, 2012). Het belang van ICT wordt daarin als volgt samengevat.

"ICT delivers and will continue to deliver. It is the driver of innovation, disruption and productivity. In an OECD study, almost 60% of the economic growth

in 1985 - 2000 is deduced from the application of ICT. And, also in the US, it has been demonstrated time and again that investing in ICT improves productivity, more than anything else and increasingly more than previously expected."

Een tweede roadmap die binnen HTSM is verschenen betreft Embedded Systemen. Hier mengt ICT zich met andere apparatuur. Embedded software is daarin de kern om functionaliteit, flexibiliteit en bruikbaarheid te realiseren. Duidelijk is dat ons lectoraat een kernrol speelt vanuit HTSM perspectief.

Versteving onderzoek

Om een beeld te schetsen voor welke problematieken we ambient intelligence inzetten geef ik hieronder een kort overzicht van onze projecten. In al deze projecten wordt samengewerkt met een consortium van bedrijven en kennisinstellingen. Meer informatie over de projecten is te vinden op de website van het lectoraat www.saxion.nl/ami.

Er is een stevige samenwerking van ons lectoraat met de Universiteit Twente. Met de leerstoel Pervasive Systems van professor Havinga voeren we een aantal van onderstaande projecten uit. Een onderzoeker voert binnen het project GoGreen een promotieonderzoek uit. Binnen Veiligheid op de Werkvloer werkt een onderzoeker aan zijn promotie bij de leerstoel Human Media Interactie van professor Heijlen.



Veiligheid op de Werkvloer

De kernvraag is: Hoe kunnen we de veiligheid op de werkvloer met behulp van ambient technologie bevorderen? Hierin werkt ons lectoraat samen met de lectoraten Smart Functional Materials en Industrial Design. Twee kernthema's worden onderscheiden: enerzijds gedragsbeïnvloeding en anderzijds monitoring van de conditie van werkers en hun omgeving.

Future Store

De kernvraag is: Wat betekent een veranderend wereldbeeld richting sociale media voor het (koop)gedrag van de consument, en daarmee voor het concept en de inrichting van de retailer in relatie tot visuele en virtuele representatie?

Virtuele Verwachting, Fysieke beleving

Aansluitend op Future Store is het doel van dit project:

Het ontwikkelen en toepassen van media- en ICT-innovaties voor het beïnvloeden van winkelbeleving en koopgedrag van shoppers in de modesector, zowel fysiek als online. De online beleving van een webshop schept verwachtingen, die worden waargemaakt als je fysiek het product in handen hebt.

GoGreen

In dit project wordt een lerend systeem ontwikkeld dat zich bewust is van de energiebehoefte en energiegebruik van mensen in een omgeving, en dat daarbij mensen ondersteunt zo efficiënt mogelijk met energie om te gaan met behoud van comfort.

Wireless Sensortechnologie bij Calamiteiten

De kernvraag is: Hoe kan wireless sensortechnologie bijdragen aan de detectie en bestrijding van calamiteiten? Denk daarbij vooral aan branden.

Mobile Sensing for Safety

Hier gaat het om de vraag: Hoe kan mobiele technologie situational awareness vergroten zodat tijdig een betrouwbaar beeld ontstaat voor hulpverleners en individuele personen op basis waarvan zij een adequaat besluit kunnen nemen om de veiligheid te verhogen?

Het platform dat in dit project gebruikt wordt is de smartphone.

Windesheim is hogeschoolpartner.

TEC4SE

Binnen dit regionale veiligheidsprogramma richt Saxion zich op Object-informatie. Het gaat daarbij om statische en dynamische informatie in de stadscampus van Saxion. Met behulp van diverse sensoren en communicatiesystemen wordt informatie verzameld over de stadscampus en de aanwezigheid van mensen. Deze informatie is anoniem met het oog op de privacy.

Firebee

Dit project voert ons lectoraat uit samen de hogeschoolen Fontys en Avans. Ons aandeel betreft indoor lokalisatie met behulp van inertial navigation. Bewegingssensoren en ontwikkeling van algoritmen voor data-analyse en plaatsbepaling spelen een kernrol.

Innovatieve Technologie voor Prestatieverbetering in de Topsport

In dit project gaat het om de vraag: Hoe kan innovatieve technologie worden ingezet om sporters tot betere prestaties te brengen? Het observeren van bewegingen en meten aan het lichaam door middel van slimme technologie biedt hiertoe nieuwe mogelijkheden.

Sax Shirt

In een aantal projecten wordt een demonstrator ontwikkeld van een shirt waarin geavanceerde textielen samen met sensoren worden toegepast om een comfortabel sensor-shirt te ontwikkelen. Per project wordt aan verschillende aspecten gewerkt. Denk aan de materialen en sensorinfrastructuur, de analyse van de sensordata en de toepassingsmogelijkheden. Vanuit andere lectoraten is ons lectoraat gevraagd een inhoudelijke bijdrage te leveren aan projecten. Ik noem hierbij een drietal projecten:

- Medical Robots, geïnitieerd door het lectoraat Mechatronica.
- Duurzame Intelligente Gebouwen, geïnitieerd door het lectoraat Duurzame Energievoorziening.
- De Slimme Vloer, project-initiator is het lectoraat Smart Functional Materials.

7. De rol van onderzoek in het onderwijs

In het hbo is een trend waar te nemen van beroepsgericht onderwijs naar een combinatie van onderwijs en onderzoek. Nog steeds staan beroepsvaardigheden centraal. Als element daarbinnen komt er meer aandacht voor informatie- en onderzoeksvaardigheden. Bedrijven willen niet alleen een softwaresysteem: er moet een probleem opgelost worden. Dat daarvoor software nodig is, is vaak wel duidelijk, maar waar moet de software aan voldoen en wat is er al beschikbaar? Vaak zal reeds bestaande software kunnen worden gebruikt en aangepast. Met de komst van lectoraten is er een instrument beschikbaar om praktijkgericht onderzoek in het hbo te verankeren. Ook in de beroepspraktijk van afgestudeerden worden onderzoeksvaardigheden belangrijk. Denk aan advieswerk en aan situaties waarin complexe problemen moeten worden opgelost. Op universiteiten komt veel onderzoek voort uit vakinhoudelijke nieuwsgierigheid. Dit wetenschappelijk onderzoek is gericht op grensverleggende kenniscreatie. Bij technische studies kan onderzoek gericht zijn op het ontwerpen en valideren van een systeem. Dan komt de toepassing in beeld en spreken we van toegepast onderzoek. Voor het hbo komt onderzoek primair in beeld vanuit een vraagstelling in de praktijk. Er moet

in die praktijk bijvoorbeeld een probleem opgelost worden, een systeem ingericht, of een besluit voorbereid worden. We spreken dan van praktijkgericht onderzoek. Niet altijd zal het onderscheid tussen verschillende types onderzoek zo scherp gemaakt kunnen worden. Er is een glijdende schaal van wetenschappelijk naar toepassingsgericht onderzoek. Voor de discussie is het onderscheid evenwel bruikbaar.

Binnen het domein van de ICT zijn verschillende researchmethoden gangbaar. Johnson onderscheidt in zijn artikel 'Basic Research Skills' in Computing Science, (Johnson, z.j.) de volgende methoden:

Implementatiegedreven onderzoek

Bij deze aanpak wordt een systeem gebouwd om eraan te meten of om te bewijzen dat een hypothese geldig is. Denk aan de haalbaarheid van de realisatie van een systeem met een nieuwe technologie. Dit komt veel voor in de informatica. Deze werkwijze kent beperkingen en risico's die kunnen voortkomen uit de kwaliteit van het gebouwde systeem of het gevolgde proces tijdens de bouw. In veel afstudeerprojecten komt deze aanpak voor. Hij ligt vlak bij een gezonde engineeringwerkwijze.

Wiskundige bewijstechnieken

Dit is een formele aanpak, die voor het hbo maar in beperkte mate bruikbaar is.

Empirisch onderzoek

Vanuit een hypothese wordt een experiment opgezet, gegevens worden verzameld en conclusies worden getrokken. Het cijfermateriaal wordt verwerkt met statistische methoden. De beperking zit in de mate waarin het experiment goed gecontroleerd kan worden uitgevoerd.

Observatie

Het gebruik van een systeem wordt geobserveerd. Door de observaties onder verschillende omstandigheden uit te voeren kunnen verschillende resultaten worden verkregen. Daarmee wordt dit soort onderzoek vaak erg ongericht. Het is van belang vooraf goed te bepalen onder welke condities de observaties worden gedaan.

Implementatiegedreven onderzoek ligt voor de hand als veelgebruikte methode in de informatica. Als onderdeel daarvan komt de technologiekeuze aan de orde. De keuze moet goed onderbouwd kunnen worden. Dat vereist goed inzicht in de state of the art. Als vervolg op het bouwen en valideren van een systeem komen na dit soort onderzoek vaak nieuwe onderzoeksvragen op. Daarvoor is het nodig experimenten met dit systeem of observaties van het gebruik ervan uit te voeren.

Welke competenties voor een hbo-student informatica of techniek kunnen we formuleren? De drie TU's hebben voor het technisch WO competenties geformuleerd (Meijers, 2005). Dat kan ons inspireren ook voor het technisch hbo een aantal vaardigheden en houdingen te formuleren:

De student:

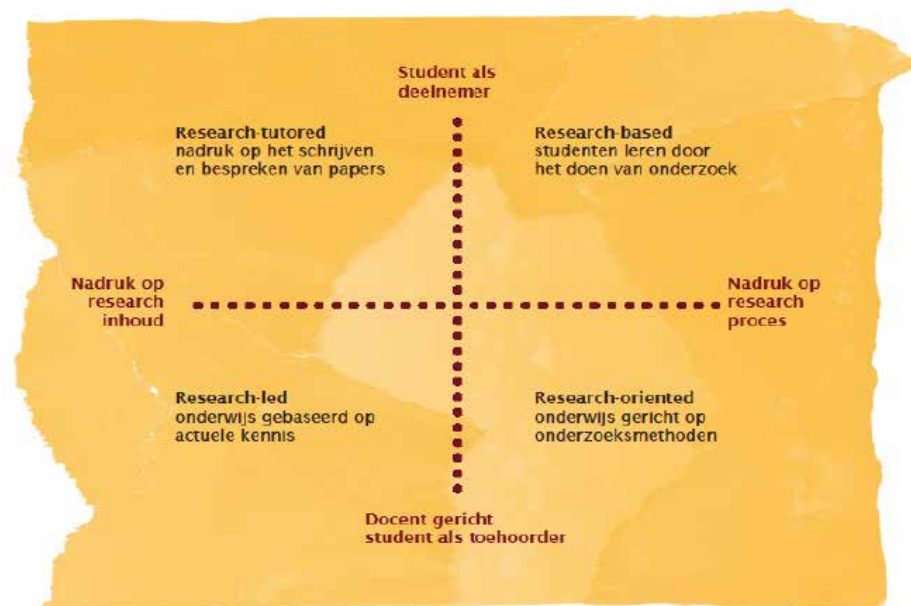
- is in staat om een slecht gestructureerd probleem (praktijkvraag of onderzoeksvraag) te herformuleren; kan daarbij het bereik (in of out of scope) afbakenen en kan deze nieuwe interpretatie verdedigen;
- is opmerkzaam en heeft de creativiteit om eigen en andermans werk vanuit verschillende standpunten en op verschillende niveaus van abstractie te beschouwen;
- kan een onderzoek ontwerpen en dit uitvoeren (onder begeleiding);
- kan een transformatie maken van een concreet probleem naar een generiek probleem en omgekeerd een vertaling maken van een generieke oplossing naar een specifieke situatie;
- is flexibel en kan de scope van een probleem aanpassen; is zich bewust van de beïnvloeding van onderzoeksresultaten door de vraagstelling, de inrichting en ontwerp van het onderzoek;
- is in staat de bruikbaarheid, de wenselijkheid, de uitvoerbaarheid en de kosten van een onderzoek in te schatten;
- is in staat te reflecteren op eigen werkwijze en behaalde resultaten;
- kan rapporteren (mondeling en

schriftelijk) over het onderzoek;

- kan informatievaardigheden toepassen;
- kan indien nodig statistische methoden hanteren.

We kiezen ervoor studenten hun competenties te laten ontwikkelen door te participeren in lopend onderzoek in het kenniscentrum, bij één van de lectoraten. Natuurlijk is inleidend onderwijs nodig in de aard van onderzoek, onderzoeksmethoden en het kunnen presenteren van werkwijzen en resultaten, maar het doen van onderzoek komt primair. Dat het daarvoor noodzakelijk is dat het onderwijs zelf actueel is en put uit state of the art kennis, is een vanzelfsprekendheid. In het onderstaand kwadrant (Borgdoff, 2007) ligt ons accent op het kwadrant rechtsboven.

Binnen het lectoraat Ambient Intelligence worden veel langlopende projecten uitgevoerd. Informaticastudenten kunnen daarin participeren door een deelproject uit te voeren. Dat geldt overigens ook voor studenten van andere opleidingen. Onderzoeksvaardigheden die in de opleiding worden onderwezen, worden al doende steeds meer aangescherpt. Het doel is niet het onderzoek op zich, maar te komen tot betere (software)producten en een beter inzicht in het zowel het doel van het product als de bruikbaarheid van nieuwe (software)technologie. Onderzoek scherpt het denken. Mijn hoop is dat de reductie van aandacht voor analytische vaardigheden die onder andere met wiskunde beoogd werd, enigszins gecompenseerd kan worden door te leren onderzoeken. Of het ook zo uitpakt, zal in de praktijk blijken.

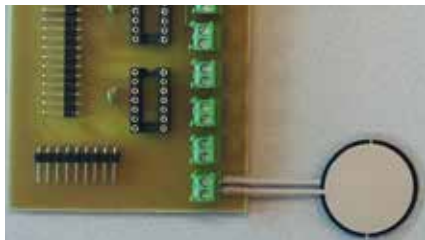


Studenten zijn zowel betrokken geweest bij projecten in het lectoraat zelf, als binnen het DevLab in Eindhoven. Ik noem er hier slechts enkele:



Protami

Meten van druk op de huid in een prothese tijdens het lopen.



Gossip netwerken met ambient nodes

Het Gossip-protocol implementeren op een wireless sensornode.



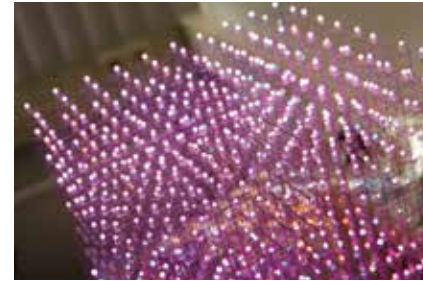
Branddetectie

Het detecteren van het begin van een brand zowel indoor als outdoor.



3D LED Kubus

Het weergeven van letters en patronen in een kubus van 1000 multicolor LEDs.



Mobile Sensing

Het verkennen van de sensingmogelijkheden van een smartphone. Casus: een val detecteren.

Sensor Shirt

Opstellen van eisen aan een shirt waarin sensoren en slimme textiel worden gecombineerd. Tevens realiseren van een sensorshirt.



Inertial Navigation

Het traceren van mensen in een brandend gebouw.

Track & Trace

Het via GPS volgen van personen die lopen en af en toe stilstaan.



Veilige ontruiming

Het onderzoeken wat voor type signalering het meest effect heeft bij het ontruimen van een brandend gebouw waarin al veel rook het zicht belemmert.



8. Terugblik en conclusies

In het tweedimensionale schema van vrijheid en interactie hebben we interactie een steeds prominentere rol zien spelen, zowel zichtbaar als verborgen in de communicerende systemen. De prominente rol die onzichtbare systemen spelen, is de aanleiding het schema met een derde as uit te breiden: zichtbaarheid. Aansluitend is het een open vraag of het vakgebied informatica een wetenschap is die zich alleen richt op informatie, of dat interactie een gelijkwaardige plaats daarin heeft.

De evolutie van systemen is niet als een lineair proces verlopen. Het ging niet alleen om steeds kleinere componenten en devices, steeds krachtiger processoren, grotere geheugens op kleine chips maar ook om snellere communicatie, nieuwe interactie technologieën, nieuwe toepassingen van sensoren en actuatoren. Verrassingen kwamen door integratie met nieuwe elementen, zoals sensoren en actuatoren op chips. Een voorbeeld van een niet voorziene stap met een enorme impact was het world wide web, met toepassingen die ongekend waren. Hier kwam de doorbraak dankzij een nieuwe taal (HTML) en de integratie van bestaande netwerken. Ontwikkelingen zijn moeilijk te voorspellen. ICT maakt gebruik van nieuwe

materialen en wordt geïnspireerd door verrassingen in andere vakgebieden, zoals in de biologie, taal, logica, nanotechnologie, brain science en quantumfysica. Veel ontwikkelingen zijn disruptief. Het proces is zeker niet lineair. Er zijn versnellers of vertragers in andere domeinen dan ICT die het proces dominant beïnvloeden.

Er is een enorme overvloed van devices. De meeste zijn klein en krachtig. Er is ook een explosieve groei van rekenkracht. Op termijn kunnen we het ons veroorloven niet meer zuinig te zijn met processorpower en geheugens. Dat betekent dat er ruimte is om minder efficiënte maar krachtiger of meer abstracte programmeerconcepten te gebruiken en paralleliteit uit te baten.

Software speelt een essentiële rol in innovatieve systemen, ook en niet in het minst in niet ICT-systemen. Als software als enabler ontbreekt wordt innovatie vertraagd of belemmerd.

Software is de drager van de interpretatie van sensordata. Het zal niet zo zeer om de sensoren op zich gaan, maar om de vraag wat je met de data kunt doen. Zo kan zin (sense) worden gegeven aan sensing en kan nonsense van sensing worden voorkomen.

Software is inherent complex en daardoor is het een moeilijke taak om software engineers op te leiden die kunnen uitgroeien tot 'great designers'. Hoeveel vermogen tot abstractie en analyse kunnen studenten verwerven in combinatie met het vermogen creatieve oplossingen te realiseren? Welke elementen in een curriculum dragen daartoe het sterkst bij?

De opleiding Informatica heeft een eigen karakter weten te behouden en heeft daarbij niet het isolement gezocht. Er zijn veel contacten met bedrijven en andere opleidingen.

Onderzoek in het hbo heeft na tien jaar een vaste plaats gekregen. De positie is per hogeschool verschillend. Soms zitten lectoraten binnen de opleiding. Soms zijn ze gebundeld tot kenniscentra verwant met maar geplaatst naast de opleiding. Het blijft zaak een sterke synergie met de opleidingen te behouden, zonder het onderzoek zelf en de relatie met bedrijven en kennisinstellingen te verwaarlozen. Binnen Saxion en vooral binnen het Kenniscentrum Design en Technologie lukt dit verrassend goed. Het is daarmee een kostbaar goed dat gekoesterd moet worden.





9. Met dank aan

De afgelopen decennia zijn bijzonder geweest voor de informatica en haar toepassingen. Het vak is volwassen geworden. Het was bijzonder plezierig de vele innovaties te kunnen meemaken en te zien wat daarvan in onderwijs en hbo-onderzoek een relevante plek kon vinden.

Bij het omgaan met studenten zijn er leerzame en inspirerende momenten. Als docent leer je zelf en het is boeiend te zien hoe studenten zich ontwikkelen. Ik ben dankbaar voor de kansen die ik heb gehad om met studenten in hun leerproces op te trekken. Aan mijn collega's bewaar ik bijzondere herinneringen. Soms waren we elkaars slijpsteen, dat is nuttig. Vaak

heb ik de onderlinge gesprekken als bijzonder stimulerend ervaren. Dat kon zowel gaan over het informatica als vak, als over het onderwijs, de rol van afgestudeerden in de beroepspraktijk, maar ook over heel andere, persoonlijke onderwerpen als muziek, literatuur, onze families en dingen die je drijfveer zijn en richting geven. Naast het onderwijs verzorgen en ontwikkelen is een aantal jaren veel energie gaan zitten in het werven en uitvoeren van externe onderwijsopdrachten. De samenwerking met collega's daarin heeft tot mooie trajecten geleid. Het vormgeven aan praktijkgericht onderzoek is een zoektocht geweest. Samen met collega-lectoren is een lectoraat opgebouwd dat er zijn

mag. Tegen alle collega's met wie ik door de jaren heen in verschillende situaties heb samengewerkt heb, zeg ik: collega, bedankt!

In de Saxion organisatie heb ik met veel mensen korter of langer opgetrokken. Veel ondersteuning heb ik gekregen vanuit de diensten, vooral vanuit Financiën, in verband met de vele begrotingen en financiële rapportages van diverse activiteiten. Ook afdelingsmedewerkers en de ondersteuners van het bedrijfsbureau van het kenniscentrum hebben me geholpen dingen te doen die ik alleen niet kan. Mijn dank daarvoor wil ik bij deze uitspreken. Mijn leidinggevendenden waren zeer divers. Al met al heb ik van hen

veel ruimte gekregen me te ontwikkelen en nieuwe uitdagingen aan te gaan. Dat was bijzonder aangenaam. Bedankt!

In de rollen die ik mocht vervullen speelden externe contacten met andere hogescholen, universiteiten, bedrijven en samenwerkingsverbanden geregeld een rol. De blik naar buiten heb ik naast alle interne activiteiten als verrijkend ervaren. Ik heb daarvan mogen genieten. Zeker van de contacten die over wat langere tijd speelden. Beste mensen, bedankt!

Wat is een mens zonder een stabiel en warm thuisfront? Erna en dochters, ik ben vooral jullie zeer erkentelijk.

Appendix 1

Lestabellen 1971 – 2013

		1973		1982			1994		2003		2012	
Eerste jaar		eenheid: lesuren per week	percentage	eenheid: lesuren per week	percentage		eenheid: lesuren per week	percentage	eenheid: studiepunten; 1 stp = 40 uur studielast		eenheid: studie- punten; 1 stp= 28 uur studie- last	
informatica	programmeren	3		4			4		4		10	
	software engi- neering incl inf. analyse						2,7		12		16	
	software systemen						2		2		13	
	apparatuur	2		3			2,7		0		2	
	practica	3		3			10,7		14		10	
	totaal		24,2		32,3			66,0		78,0		85
wiskunde	analyse	2		3			2					
	discrete wis- kunde & logica	3		3			2		4		3	
	statistiek & kansrekening	2		2			2		1			
	practica	3		3			2,7		2			
	totaal		30,3		35,5			26,0		17,1		5
economie & bedrijf- skunde	theorie	7		6			1,7		0			
	practica	2					1		0			
	totaal		27,3		19,4			8,1		0,0		
algemene vakken	sport	2		2			0		0			
	talen	4		2					0			
	recht						0	0,0	0			
	prof. vaardig- heden								2			
	onderzoek										6	
	totaal		18,2		12,9					4,9		10
Totaal		33	100	31	100		33,5	100	41	100	60	100

		1973		1982			1994		2003		2012	
Tweede jaar		eenheid: lesuren per week	percentage	eenheid: lesuren per week			eenheid: lesuren per week		eenheid: studiepunten; 1 stp=40 uur studielast		eenheid: studiepunten; 1 stp= 28 uur studielast	
informatica	programmeren & algoritmen	4		4			4,7		6		3	
	software engineering incl. inf. analyse								4		31	
	software systemen	2		6			4		8		12	
	apparatuur			2							3	
	practica	5		6			10		18		2	
	totaal		31,4		58,1			75,1		87,8		85,0
wiskunde	analyse	2		2								
	discrete wiskunde	3										
	statistiek	2		2			2					
	practica	2		2			0,9					
	totaal		25,7		19,4			11,6		0,0		0
economie & bedrijfskunde	theorie	7		5			1					
	practica	2					0,3					
	totaal		25,7		16,1			5,2		0,0		0
algemene vakken	sport	2					0					
	talen & communicatie	4					1,3		1			
	recht			2			0,7		1			
	prof vaardigheden								3		9	
	totaal		17,1		6,5			8,0		12,2		15
Totaal		35	100	31	100		24,9	100	41	100	60	100

		1973		1982			1994		2003		2012	
Derde jaar		eenheid: uren per week	percentage	eenheid: uren per week	percentage		eenheid: uren per week	percentage	eenheid: studiepunten; 1 stp=40 uur studielast	percentage	eenheid: studiepunten; 1 stp= 28 uur studielast	percentage
stage		40		40			40	100	21		30	
	totaal		100		100			100		50		50
Specialisatiethema en keuzevakken												
informatica	specialisatie- thema								14		21	
	keuzevakken								4			
	totaal									42,9		35
wiskunde	in specialisatie- thema:										3	
	totaal											5
algemene vakken	specialisatie- thema								3		6	
	totaal									7,1		10
Totaal									41	100,0	60	100

		1973		1982			1994		2003		2012	
Vierde jaar		eenheid: uren per week	percentage	eenheid: uren per week	percentage		eenheid: uren per week	percentage	eenheid: studiepunten; 1stp=40 uur studielast	percentage	eenheid: studiepunten; 1 stp= 28 uur studielast	percentage
informatica	programmeren	5									4	
	software-sys- temen als OS, compiler, dbms,	10		8			6					
	software engineering			2			2					
	practicum	7		8			6					
	keuzevakken								4			
	specialisatie- thema/minor								14		11	
	totaal		64,7		58,1			46,7		42,9		25,0
wiskunde	besliskunde	0		2								
	discrete wiskunde/lin. algebra										9	
	prestatie analyse						2					
	practicum			1								
	totaal		0,0		9,7			6,7		0,0		15,0
economie en bedrijfskunde	adm. organisatie	0										
	sociale beheer- saspecten	4		4								
	best. Inf. systemeen						2					
	totaal		11,8		12,9			6,7		0,0		0,0
algemene vakken												
	sport, Nederlands	4										
	recht						2					
	specialisatie- thema/minor								3		6	
	totaal		11,8					6,7		7,1		10,0
afstudeeropdracht		4		6			10		21		30	
	totaal		11,8		19,4			33,3		50,0		50,0
Totaal		34	100	31	100		30	100	42	100	60	100

Referenties

Abelson, H., Sussman, G.J., Sussman, J. (1996). Structure and interpretation of computer programs, 4-de druk, MIT Pres, MIT

Bemmel, A. van, (2006). Hogescholen en hbo in historisch perspectief, HBO-Raad

Boehm, B.W. (1988). A Spiral Model of Software Development and Enhancement, IEEE Computer, May 1988

Blom, G, Leeuwen, H., van, (2009). SCRUM en XP frisse wind in onderwijs, Bits & Chips, 26 maart 2009

Bordewijk, J. (red) (2009). Bachelor of ICT, domeinbeschrijving, HBO-I Stichting, Amsterdam

Borgdorff, H., Staa, A.L. van, en Vos, J. van der (2007). Kennis in context. Onderzoek aan hogescholen, Th&ma, Tijdschrift voor hoger onderwijs en management, jrg. 14, nr. 5, pp. 10-17.

Brooks, F.P. (1987). No Silver Bullet- Essence and Accidents of Software Engineering, IEEE Computer, Volume 20, Number 4, pag. 10 – 19

Cook, Augusto, Jakkula (2009). Ambient Intelligence: Technologies, applications and opportunities, Pervasive and Mobile Computing Volume 5 (2009) pag. 277 – 298

Dijkstra, E.W.(1972). The humble programmer, Communications of the ACM, Volume 15 Issue 10, pag. 859 – 866

EINIAC, European Nanoelectronics Initiative Advisory Council (2007). Strategic Research Agenda, European Technology Platform Nanoelectronics

Evans, D. (2011). The Internet of Things, How the Next Evolution of the Internet Is Changing Everything, White Paper, Cisco IBSG

Feki, M.A., Kawsar, F., Boussard, M. Trappeniers, L.. (2013). The Internet of Things: The next technological revolution, IEEE Computer, February, 2013

Florijn, G. (1986). MANUS – An experimental educational UNIX Network, proceedings EEUG, Autumn 1986

Gude, R.,(2010). Immanuel Kant, Geschiedenis van de filosofie, , CD4, Luisterwijs

Intel (2009). Rise of the Embedded Internet, White Paper Intel Embedded Processors, Intel Corporation

Jenkins, A. & Healey, M. (2005) .Institutional strategies to link teaching and research. http://www.heacademy.ac.uk/assets/York/documents/ourwork/research/Institutional_strategies.pdf, 14-1-2010

Johnson (z.j.). Basic Research Skills in Computing Science, www.dcs.gla.ac.uk/~johnson/teaching/research_skills/basics.html , 4 april 2008

Kelly, K. (2010). What technology wants, Penguin Books, New York

Khan, W.Z., Xiang, Y. (2013). Mobile Phone Sensing Systems: A Survey, IEEE Communications Surveys & Tutorials, Vol. 15, No1, First Quarter 2013

Kniberg, H. (2007). Scrum and XP from the trenches, InfoQ, <http://www.infoq.com/minibooks/scrum-xp-from-the-trenches>, augustus 2013

Kortuem, G., Bandara, A.K. e.a. (2013) Educating the Internet-of-Things Generation, IEEE Computer, February 2013

Langhout, M.J. (1997). Het belang van formele methoden voor de opleidingen HI en BI, 25 jaar HBO Informaticaonderwijs, Uitgeverij ten Hagen & Stam, pp. 42 - 51

Leeuwen, H., van, Maas, J., Land, Th. (2011). Intelligente Sensornetwerken, Thematische Impulsen, SIA, Den Haag.

Leeuwen, H., van, Paulissen, A. (1997). 25 jaar HBO Informatica-onderwijs, Ten Hagen & Stam Uitgevers, Den Haag

Leeuwen, H., van, Teeuw, W.B., Griffioen, P.S. (2010). De computer verdwijnt, leve de computer, Saxion, Enschede

Lundqvist,A. Apers, P., Smeulders, A., Huizer, E.. Mandersloot, P. (2012).

Roadmap ICT for the top sectors.

Meyer, B. (1988). Object-oriented Software Construction, Hemel Hempstead, Prentice Hall

Meijers, A.W.M., Overveld, C.W.A.M., van, Perrenet, J.C. (2005). Criteria voor Academische Bachelor en Master Curricula, 3TU, Delft,, Enschede, Eindhoven

OCC, Open Geospatial Consortium (2007). Open GIS Sensor Model Language (sensor ML) Implementation Specification, www.opengeospatial.org, mei 2013

Penning, A. (1990). Een nieuwe agenda voor het informatica-onderwijs, HBO-Raad, Den Haag.

Polak, F.L, (1966). De nieuwe wereld der automatie, een industriële en sociaal-culturele revolutie, Uitgeversmaatschappij W. de Haan, Hilversum

Ridder, Th., de. (1984). Een lokaal computernetwerk als manus-je van alles, Computable, 2 november 1984

Verrijn Stuart, A.A. (1979). Hoger Informatica Onderwijs, HIO Stuurgroep, Den Haag

Weiser, M. (1991). The computer for the 21st century, Scientific American Vol. 265, No 3, pp. 94 - 104

Wiener, N. (1948). Cybernetics or control and communication in the animal and the machine, The M.I.T. Press, Cambridge, Massachusetts

WIHBO (1973). Leerplan voor het hoger informatica onderwijs

Yurish, S.Y. (2011). Smart Sensor Systems Integartion: New Challenges, The 6th International Conference on Systems (ICONS' 2011), St. Maarten, The Netherlands Antilles, 24 January 2011

Zijlker, A.W. (1989). Informatietechnologie en 'competitive edge': Na de storm, Delden

Zijlker, A. (1988). "Setting the Scene for IRM," Information and Management, Volume 15, pp. 79-84.

ISBN/EAN: 978-94-6213-007-4
Titel: Sense and nonsense of sensing
Auteur: Henk van Leeuwen
Uitgever: Saxion Kenniscentrum Design en Technologie
Datum: 20 november 2013
Plaats: Enschede