



# I/O VIVAT

JAARGANG 25  
NUMMER 2

## Android tutorial

Deel 1: Bouw je eigen  
Twitter applicatie

## 3D Televisie

Komen de beelden nu  
écht tot leven?

## Electronic hacking

Word jij niet gecontroleerd?

## Parallele berekeningen

### door de GPU

Gebruik de GPU voor meer dan  
alleen het renderen van beelden

## En verder...

Conversational Agents  
Aspect-oriented programming  
Data-integratie onzekerheden  
Sprakherkenning



Inter-Actief

*Advertentie*  
*TNO*



Jaargang 25, nummer 2,  
februari 2010  
ISSN: 1389-0468

I/O Vivat is het populair-wetenschappelijke tijdschrift van I.C.T.S.V. Inter-Actief, de studievereniging voor Technische Informatica, Bedrijfsinformatietechnologie en Telematica van de Universiteit Twente. I/O Vivat verschijnt vier maal per jaar en heeft een oplage van 1800 exemplaren.

Hoofdredactie:

Bas Stottelaar

Redactie:

Michel Brinkhuis, Rick van Galen,  
Marije de Heus, Jelte Zeilstra

Vormgeving:

Niels Witte

Gastschrijvers:

Rom Langerak, Francis Henninger,  
Dong Nguyen, Laurens Satink,  
Oliver Jundt

Voor vragen, suggesties en tips is I/O Vivat bereikbaar via e-mail op [vivat@inter-actief.net](mailto:vivat@inter-actief.net), telefonisch op 053-489 3756 of per post:  
Studievereniging Inter-Actief  
Postbus 217  
7500AE Enschede

De studievereniging wil de adverterende bedrijven bedanken voor de samenwerking.

Drukwerk:

Drukkerij van den Bosch & Fikkert  
© 2010 I.C.T.S.V. Inter-Actief



I/O VIVAT

## Redactioneel

Studiejaar 2009-2010 zijn we goed begonnen met een super toffe I/O Vivat editie, namelijk jaargang 25, editie 1. Die extra goed gevulde editie werd warm onthaald door alle trouwe lezers, dat was te merken aan de vele bedankjes achteraf. Met ook ruim 100 mensen op de borrel, voorzien van champagne, illusionist en aluminiumfolie, was ook de presentatie zeer geslaagd. Toch was 25.1 ook een moment waar een aantal leden hebben aangegeven te willen stoppen. Zo werd het stokje overgedragen van de oude garde naar een nieuwe fris en jonge commissie!

Hoewel ik pas sinds 24.3 betrokken ben bij de I/O Vivat, heb ik nu al de eer gehad om het stokje van Eelco Eerenberg als hoofdredacteur over te nemen. Ook voor mij is het even wennen dat ik nu de verantwoording heb over een blad, wat met 1800 exemplaren per editie, toch best groot te noemen is. Gelukkig sta ik er niet alleen voor en krijg ik de hulp van vijf andere commissieleden. Ik mag misschien dan ook niet alleen namens mijzelf spreken.

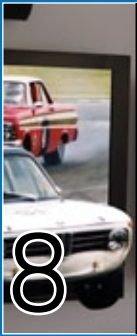
Ook voor deze uitgave zijn de nodige uren vrijgemaakt. We blijven in de stijl van 25.1, maar dan met een paar veranderingen. Zo proberen we elke uitgave een tutorial te gaan schrijven die iets meer is dan slechts 'Hello World'. Jelte heeft hiervoor de aftrap gedaan met zijn tutorial over Android.

Namens de nieuwe I/O Vivat commissie mag ik u veel plezier wensen met ook weer een interessante 25.2!

Bas Stottelaar

Hoofdredacteur I/O Vivat

## Artikelen



### 3D Televisie

*Door Michel Brinkhuis*

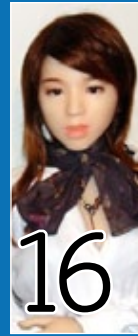
3D, TELEVISIE, STEREOSCOPIE, AUTOSTEOSCOPIE, HEADTRACKING



### GPU: Parallele berekeningen

*Door Rick van Galen*

REKENKRACHT, GRAFISCHE PROCESSORS, SUPERCOMPUTERS, PARALLELISME, PARADIGMA'S



### Conversational Agents

*Door Marije de Heus*

INTELLIGENT AGENTS, CONVERSATIONAL AGENTS, INSTRUCTIEPROGRAMMA'S, SOFT & HARD COMMUNICATION SKILLS



### ENIAC: Master in de VS

*Door Dong Nguyen*

MASTER, AMERIKA, LANGUAGE TECHNOLOGIES, PITTSBURGH, CMU



### ENIAC: Spraakherkenning

*Door Laurens Satink*

TEKSTEN, CONTEXT-AWARENESS, SPRAAKTECHNOLOGIE, STATISTISCHE ANALYSE



### Tutorial: Android deel 1/2

*Door Jelte Zeilstra*

ANDROID, TWITTER, JAVA, ECLIPSE, SDK, TUTORIAL

## Nieuws



### '2009 Goed jaar voor BREIN'



### Volgsoftware voor kind



### CES: Een kijkje in de toekomst

topicus 



ASML

Overboek  
Adresge  
18

*Aspect-oriented programming*  
Door Jelte Zeilstra

---

CODE ORDENEN, BASISPROGRAMMA, UITBREIDEN, ADVISIES, JOIN POINTS, ASPECTJ

22

Word jij niet gecontroleerd?  
Door Bas Stottelaar

---

HACKING, ELEKTROMAGNETISCH, ONTCIJFEREN, SLEUTEL, GEHEUGEN

26

Data-integratie onzekerheden  
Door Oliver Jundt

---

DATABASES, INTEGRATIE, VERZAMELING, FEEDBACK

## Columns

25

Frank Gehry en het hoorcollege  
Door Rom Langerak

28

Van de voorzitter  
Door Rick van Galen

29

ENIAC: Van de penningmeester  
Door Francis Henninger

## Verder

15

Top vijf 'IT-fails' in 2009

38

Op bezoek bij: Topicus  
Door Rick van Galen

---

TOPICUS, ICT, BEZOEK, CARE, PDA

39

Volgende keer in de Vivat



*Belastingdienst*  
**Centrum voor ICT**



## Nieuws

Redactie: Bas Stottelaar  
& Michel Brinkhuis

BREIN, FTD, NZB, VOLGSOFTWARE,  
PROTECTOR, CES, TOEKOMST

# Nieuws

## Vacatures:

### I/O Vivat

Ben jij een schrijftalent, of is Adobe Illustrator voor jou gesneden koek? Kom dan bij de I/O Vivat, en werk mee aan het tijdschrift waarin je nu aan het lezen bent.

### Lustrumcommissie

Inter-Actief is op zoek naar enthousiaste mensen die in 2011 een spetterend 6e lustrum willen organiseren. Wil jij contacten leggen met het bedrijfsleven, een grote commissie doen die goed staat op je cv en een kans benutten die zich maar ééns in de vijf jaar voordoet? Vraag ernaar bij het bestuur.

### Noizia

Wil je graag wat schrijven, maar liever niet al te serieus? Neem dan eens contact op met de noizia. Deze gezellige club is altijd op zoek naar versterking.

### BoCie

Ah ja, bier! De BoCie verzorgt dit voor alle kleinere en grotere borrels van Inter-Actief. Bij de BoCie tap je niet alleen, maar organiseer je ook borrels. Naast het drinken zorgt de BoCie zo af en toe ook voor snacks.

### Hyper-Actief

Ben jij graag actief bezig en heb je wat met sporten? Bij de Hyper-Actief zoekt men mensen die het leuk vinden om sportevenementen te organiseren. Ook kan je helpen met o.a. het organiseren van het StaF-toernooi en de Batavierenrace.

### FeCie

De FeCie zorgt er voor dat er voldoende gefeest wordt bij Inter-Actief. In het drukke leven van een ICT-student mag dit natuurlijk niet vergeten worden. Wil jij meehelpen met het organiseren van feesten voor wel 200 personen? Dan is de FeCie op zoek naar jou!

### Neem contact op

Spreekt een van deze commissies je aan? Informeer ernaar bij het bestuur, en ze vertellen je er graag meer over.

## '2009 goed jaar voor Stichting BREIN'

Stichting BREIN (Bescherming Rechten Entertainment Industrie Nederland) heeft laten weten dat ze in het jaar 2009 totaal 615 sites heeft afgesloten, waaronder 393 torrentsites, 14 usenet portals, 35 eDonkey servers en 38 streaming video sites. Ten opzichte van 2008 is dit een groei van 50% - in dat jaar sloten ze namelijk 398 sites af. Het jaar 2009 was ook het jaar dat de torrentsite Mininova illegale bestanden actief ging weren. Ook The Piratebay kreeg het zwaar te voorduren - de Zweedse aanbieder van torrents werd door de Nederlandse rechter gedwongen om voor 1 maart 2010 al het aanbod naar illegale materialen te verwijderen. Daarnaast moet ze Nederlandse bezoekers weren.

In het jaar 2010 zal de anti-piracy organisatie zich voornamelijk richten op usenet portals, waaronder ook de Nederlandse community FTD. De community klaagde BREIN zelf ook

al aan in mei 2009 omdat ze een rectificatie van BREIN eist. Die noemde de activiteiten van FTD strafbaar, zelfs nadat FTD alle links naar content verwijderde - waaronder de NZB-knop.

Verwacht wordt dat ook in het jaar 2010 de downloader niet zal worden afgeschrikt. Een downloadverbod zal er voorlopig nog niet doorkomen, maar het kabinet heeft in oktober 2009 laten weten dat zij over een aantal jaren het downloaden van auteursrechtelijk beschermd materiaal wil verbieden. Dit alleen als blijkt dat materiaal duidelijk uit een illegale bron komt. Een voorwaarde voor deze wet is wel dat er voldoende alternatieve en legale mogelijkheden beschikbaar komen vanuit de entertainmentindustrie voor de downloader.

## 'Bedrijf komt met volgsoftware om kind te controleren'

Het Amerikaanse bedrijf TASER (bekend van de gelijknamige stun guns) heeft een parental control programma voor mobiele telefoons gepresenteerd om kinderen op afstand te kunnen controleren. PROTECTOR, zoals het programma genaamd is, is een applicatie die op de telefoon van het kind en op de computer of telefoon van de ouder geïnstalleerd moet worden. Op deze manier is de ouder in staat om sms-berichten te lezen, oproepen te controleren, nummers te blokkeren en op basis van GPS het kind te volgen. Het programma is ook in staat om gesprekken te kunnen afluisteren en het gebruik van de telefoon te beperken,

bijvoorbeeld wanneer het kind achter het stuur zit. De ouder kan zelf bepalen waarop het kind gecontroleerd dient te worden en wanneer één van de bovengestane gebeurtenissen plaats vindt, zal de ouder automatisch een bericht krijgen.

Hoewel zoets als parental control niet nieuw is, richt TASER zich tot de mobiele gebruikers om deze te beschermen. Jongeren worden door het gebruik van een mobiele telefoon steeds kwetsbaarder. Virtueel pesten, seksueel getinte smsjes, loverboys en auto-ongelukken door het gebruik van een mobieltje achter het stuur komen



steeds vaker voor. Maar ook kinderen die zoekgeraakt zijn, kunnen door gebruik van GPS makkelijker gevonden worden.

Wanneer de applicatie beschikbaar is en of deze ook in Nederland verkrijgbaar zal zijn, is niet bekend. De site

van TASER suggereert dat de applicatie zowel voor de iPhone als BlackBerry beschikbaar zal komen. Op dit moment kunnen alleen beta-testers op de site inloggen, wat er waarschijnlijk op duidt dat het programma nog volop getest wordt.

## CES 2010: Een kijkje in de toekomst

Honderden stands, tientallen nieuwe producten en eigenlijk een klein blikje in de toekomst. Van 7 tot en met 10 januari werd in Las Vegas de Consumer Electronics Show gehouden. Zowel grote fabrikanten als nieuwe spelers op de markt presenteerden daar het nieuwste van het nieuwste op het gebied van consumentenelectronica.

Heel wat product-aankondigingen stonden in het teken van kleiner en dunner. Zo presenteerde Asus een wel heel kleine PC, het EEE Keyboard. Een volledige pc verwerkt in een toetsenbord. Op de plaats waar bij veel toetsenborden het numerieke deel te vinden is heeft Asus een 5,5"-touchscreen geplaatst. De accu van dit hele systeem, waarmee zelfs FullHD-beelden afgespeeld kunnen worden, gaat zo'n vier uur mee. Dat ook netbooks nog kleiner kunnen dan ze al zijn bewijst het Koreaanse bedrijf Viliv. Zij toonden een Windows 7 netbook met een schermdiagonaal van 12,7 centimeter. De netbook met een dikte van 2,5 cm weegt net geen 400 gram.

Fabrikanten van TV-schermen gingen niet voor 'kleiner', maar zeker wel voor 'dunner'. LG presenteerde onder meer een 60"-scherm met een dikte van maar 5 centimeter. Panasonic komt binnenkort met een TV op de markt mét 3D-ondersteuning. Daarmee is het bedrijf niet de enige, ware het



niet dat het Japanse bedrijf de beeld-diagonaal heeft weten op te rekken tot maar liefst 152".

'Consumer Electronics' gaat heel wat verder dan enkel PC's, laptops en TV's, zo bleek wel uit een aantal opvallende producten. Zo komt het bedrijf Parrot met de 'AR.Drone'. Dat is een mini-helikopter die bestuurd kan worden via een iPhone of iPod Touch. Het apparaat beschikt over twee camera's. Met één camera worden direct beelden naar de iPhone of iPod verzonden. De andere camera wordt gebruikt voor de automatische piloot.

Voor sportievelingen komt het Amerikaanse bedrijf Liquid Image de komende tijd met een aantal interessante producten. Op de CES toonde men een zwembril en skibril met digitale videocamera. Beide camera's worden bediend middels knoppen aan de zijkant, en kunnen naast videobeelden ook foto's vastleggen. Deze worden opgeslagen op het interne geheugen, welke uit te breiden is tot maximaal 16 gigabyte.

Videobrillen-fabrikant Vuzix toonde op de CES voor het eerst haar nieuwe top-

model, de WRAP 920AR. Dat is een videobrill die in staat is om 3D-beelden te genereren, waardoor augmented reality een stapje dichterbij moet komen. Bij augmented reality vervaagt de grens tussen werkelijkheid en fictie, doordat de beelden van de bril inspelen op wat er werkelijk in je omgeving te zien is.

De ontwikkelingen op het gebied van elektronisch papier staan ook zeker niet stil, gezien het grote aantal presentaties van e-readers op de CES. Samsung presenteerde een tweetal op E-ink-technologie gebaseerde schermen (6" en 10") waarop met een speciale stylus door de gebruiker kan worden geschreven. Bij gemiddeld gebruik zouden de readers zo'n twee weken moeten meegaan. Als onderdeel van een aantal conceptmodellen toonde Asus de 'Waveface Ultra'. Dat is een buigbaar OLED-scherm welke als armband kan worden gedragen. Eenmaal uitgerold ontvouwt zich een platte mini-tablet. Een e-reader met flexibel scherm welke we eerder in de winkels kunnen verwachten is de Skiff Reader. Deze e-reader maakt gebruik van een buigbaar LG-touchscreen. Het apparaat zelf is echter niet buigbaar, en het buigbare karakter moet de Skiff Reader dan ook voornamelijk tegen breken beschermen.

# 3D Televisie



Michel  
Brinkhuis  
Redacteur I/O Vivat

3D, TELEVISIE, STEREOSCOPIE, AUTOSTEREOSCOPIE, HEADTRACKING

## Komen de beelden nu écht tot leven?

In steeds meer huiskamers zijn FullHD-televisies te vinden, en de ouderwetse beeldbuis verdwijnt in hoog tempo. De techniek staat echter niet stil, en er wordt door diverse bedrijven en universiteiten hard aan gewerkt om onze tv-ervaring

er diverse onderzoeksprojecten gaande naar het gebruik van RGB-lasers om 3D-beelden mee te projecteren.

3D-schermen vandaag de dag

Steeds meer fabrikanten van LCD- en

onderzoeken tijdelijk te stoppen. Philips ziet de komende jaren nog geen toekomst in 3D-televisie voor de consument. Concurrent Sony heeft daarentegen hoge verwachtingen van 3D-televisie in de nabije toekomst. Nog dit jaar wil het bedrijf een 3D-televisie op de markt brengen, en zullen er games voor de PlayStation 3 in 3D worden uitgebracht. Deze Sony televisies zullen wel een speciale bril vereisen om 3D-beelden te kunnen zien.

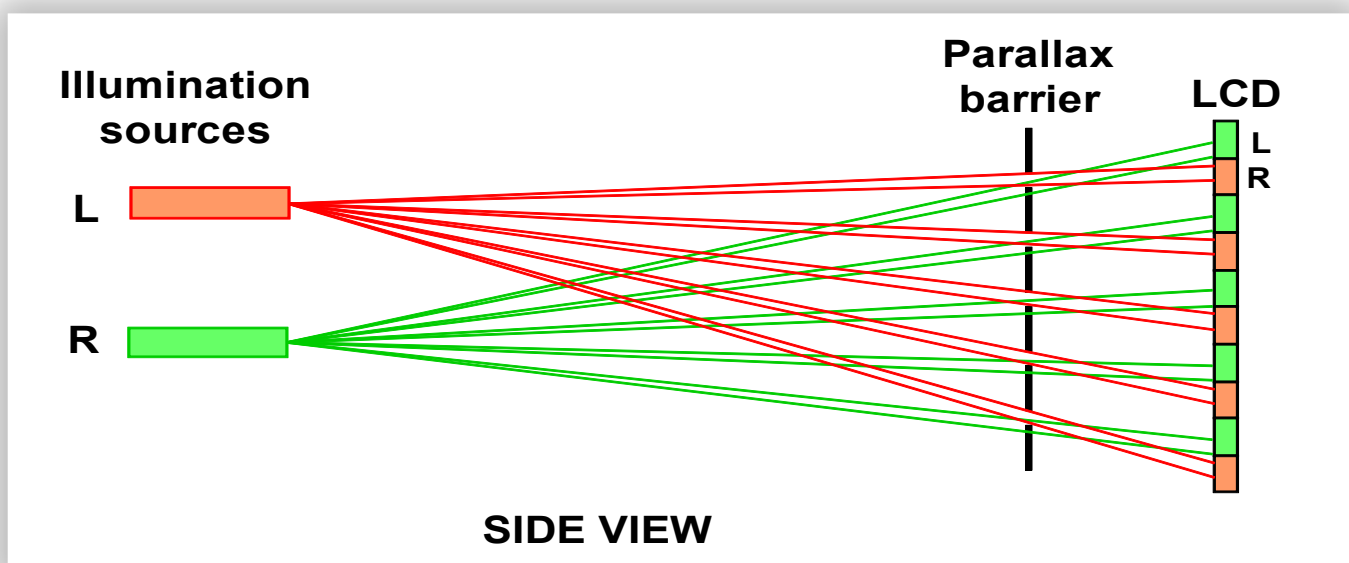
## Hoge verwachtingen 3D-televisie in nabije toekoms

nog verder te vergroten. De volgende stap zal waarschijnlijk het overgaan van 2D- naar 3D-beelden zijn. Hoe? Daar is de industrie zelf ook nog niet helemaal uit. Er wordt gewerkt aan diverse technologieën. Allereerst zijn er de LCD- en plasmaschermen, waarbij men inmiddels 3D-beelden kan weergeven mét en zonder benodigde bril. Daarnaast zijn

plasmaschermen bieden schermen met 3D-technologie aan. Philips bracht twee jaar geleden het eerste consumentenmodel op de markt, en ook Panasonic heeft onlangs een 3D-ready televisie op de markt gebracht. Philips was dan wel de eerste, het bedrijf trok zich al snel weer terug uit de markt. Begin 2009 maakte men bekend de productie en

Stereoscopie

Stereoscopie is een technologie waarbij het mogelijk is om 3D-beelden te zien vanaf een plat beeld. Deze technologie wordt toegepast in de 3D-schermen die momenteel op de markt zijn. Hierbij worden twee beelden gebruikt die vanuit een iets verschillend standpunt zijn genomen. Door ieder oog een van de



Figuur 1: De werking van de parallax barrière.





twee verschillende beelden te laten zien ontstaat in je hersenen het idee dat je kijkt naar een 3D-beeld.

Een welbekend voorbeeld van stereoscopie zijn de 3D-films waarbij je een rood/groene bril op moet zetten. De twee verschillende beelden worden hierbij genomen vanuit twee standpunten met een onderling verschil van 65 millimeter. Ieder oog krijgt dankzij de bril (een kleurenfilter) een ander beeld te zien.

#### Autostereoscopie

In tegenstelling tot de normale stereoscopie methoden is men bij autostereoscopie geen speciale bril of ander hulpmiddel nodig. Schermen die gebruik maken van deze techniek zijn voorzien van zogenaamde lenticulaire lenzen. Dit type lenzen zorgt ervoor dat er vanuit een veranderende inblikhoek een ander beeld ontstaat. Zo krijgt ieder oog, dankzij de breking van de lens, een ander beeld te zien.

Een tweede techniek die vandaag de dag kan worden toegepast is de parallax barrière. Dit is een tweede scherm dat voor een LCD-scherm geplaatst wordt, welke ervoor zorgt dat ieder oog een andere verzameling pixels van het LCD-scherm te zien krijgt.

Deze laatste techniek wordt ook gebruikt in de nieuwste navigatiesystemen van Range Rover. Op deze systemen kan de bestuurder van de auto de route zien, en kan de passagier op hetzelfde scherm een film kijken.

Het voordeel van autostereoscopische

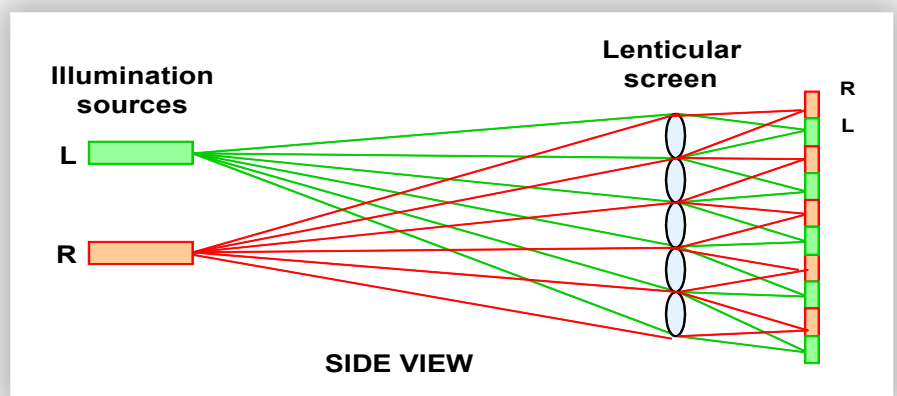
schermen is dat ze de dezelfde afmetingen hebben als normale platte schermen. Daarnaast kunnen ze ook gewoon overweg met 2D-beelden. Het grote nadeel van een 3D-ervaring vanaf platte schermen is dat je niet naar een echt 3D-beeld kijkt. In je hersenen wordt de suggestie gewekt dat het gaat om 3D-beelden. Sommige mensen krijgen na een tijdje hoofdpijn van het kijken naar zo'n scherm.

#### 'True 3D' dankzij laser plasma

In Japan werkt het National Institute Of

waargenomen met het menselijk oog. Een laser kan heel snel bewegen en zo doende kunnen met enkele laserstralen meerdere bolletjes in de lucht worden geprojecteerd, en kan een beeld worden gevormd.

Het systeem werkt op een enkele RGB-laser. Deze zendt infrarode pulsen uit naar een 'X-Y-Z'-scanner. Dat is een apparaat voorzien van een motor en een aantal lenzen die de laserstralen afbuigen naar bepaalde posities in de lucht. Op punten waar de laserstralen elkaar kruisen kan dan convergentie optreden.



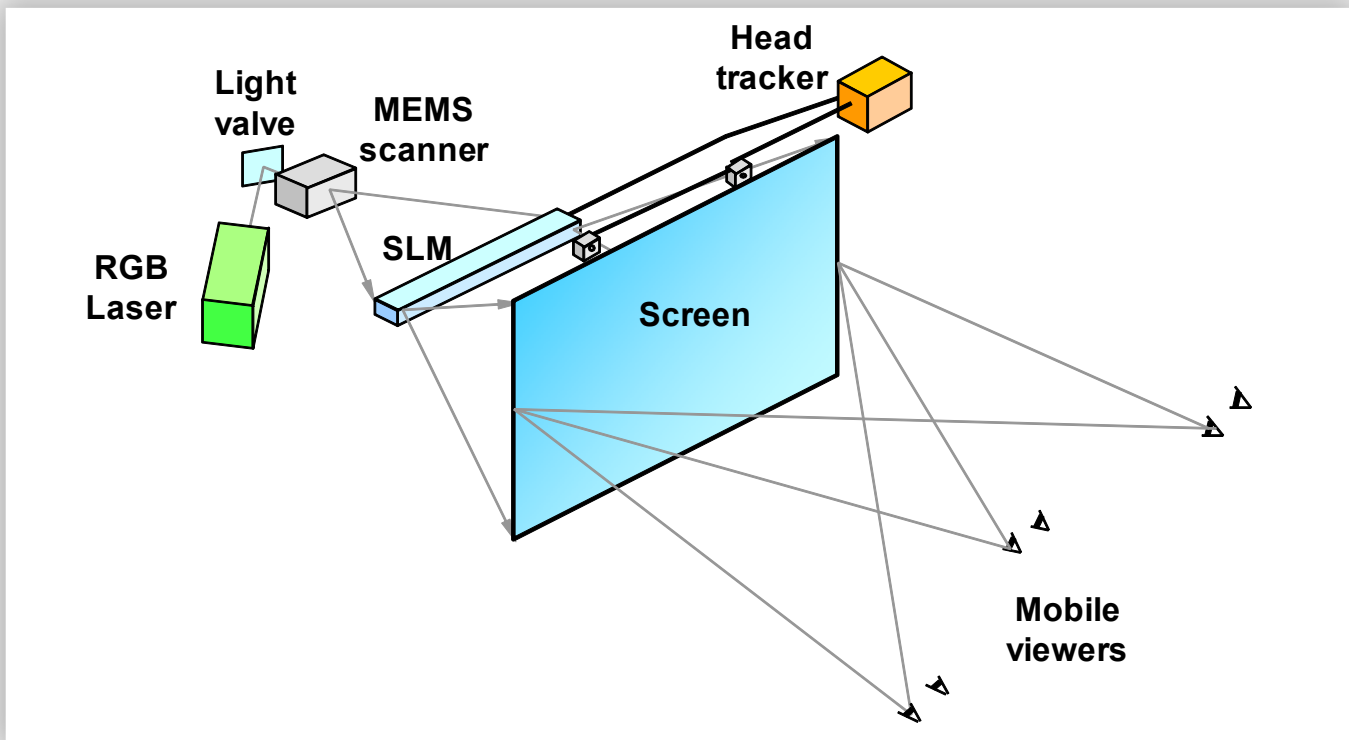
Figuur 2: De werking van een lenticulaire lens.

Advanced Industrial Science And Technology (AIST) samen met de bedrijven Burton Inc. en Hamamatsu Photonics aan de ontwikkeling van een 3D-techniek waarbij gebruik wordt gemaakt van laserstralen.

Door deze stralen te laten samenvallen in vastgestelde punten treedt er convergentie op. Op deze convergentiepunten vormt zich vervolgens een kleine hoeveelheid plasma in de vorm van een bolletje. Dit bolletje kan worden

Het geheel van de laser en de scanner kan worden gekoppeld aan een computer met speciale software. Met deze software wordt het mogelijk om een bewegend 3D-beeld in de lucht te kunnen weergeven, door de laser opeenvolgend verschillende bolletjes te laten projecteren.

Deze techniek staat echter nog in de kinderschoenen. De beelden bestaan nog maar uit één kleur, en ook de helderheid is ook nog niet optimaal. Door-



Figuur 3: Schematische weergave van het Helium3D-project

dat de bolletjes in de lucht relatief groot zijn kunnen er nog geen gedetailleerde beelden worden gecreëerd. Wel is het met één laser mogelijk om een 3D-object met een omvang van enkele meters te creëren. Echter, naarmate de afmetingen van de projectie toenemen, neemt de kwaliteit wel af.

#### Helium3D - Europees onderzoeksproject

Niet alleen de onderzoekers van AIST geloven in een 3D-technologie die gebruik maakt van lasers. In Engeland werken onderzoekers van de De Montfort University Leicester (DMU) aan het HELIUM3D-project (High Efficiency Laser-Based Multi-User Multi-Modal 3D Display). Dit is een grotendeels door de Europese Unie gefinancierd onderzoeksproject, dat wordt uitgevoerd in samenwerking met een zestal partners.

Er wordt gewerkt aan een technologie waarbij meerdere mensen naar eenzelfde scherm kunnen kijken en toch verschillende 3D-beelden zien. Dit kan handig zijn in verschillende situaties: als je liever niet hebt dat iemand 'meegluurt'; of als je met meerdere mensen naar verschillende tv-programma's wil kijken.

Het systeem werkt aan de hand van een RGB-laser, die een laserstraal uitzendt. Deze straal passeert allereerst een light valve, een optische schakelaar waarmee

de hoeveelheid licht die mag passeren kan worden bepaald. Vervolgens gaat de straal door een MEMS-scanner, een microelectromechanical system. Een MEMS-scanner is een spiegel opgehangen in een elektronisch circuit. Deze scanner kan de lichtbundels van de laser 'sturen', zodat er van links naar rechts en boven naar beneden een beeld kan worden opgebouwd. De lichtbundels gaan vervolgens door een SLM (spatial light modulator), een apparaat waarmee de fase en/of de intensiteit van het licht kunnen worden veranderd. Deze SLM wordt aangestuurd door de head tracker boven het scherm. Deze tracker bepaalt de positie van de kijkers, en is dusdanig nauwkeurig dat het de locatie van iedere afzonderlijke pupil van de kijkers kan bepalen.

#### 3D-televisie in de toekomst

Er is nog geen industriële standaard vastgesteld, er is weinig content en er is nog geen feilloos werkend exemplaar van een echte 3D-televisie te koop. De verschillende onderzoeksprojecten en de plannen van fabrikanten scheppen daarentegen hoge verwachtingen. Films en series kijken in 3D, oog in oog staan met levensechte tegenstanders bij het gamen en nooit meer ruziën om de afstandsbediening. Zomaar een aantal mogelijkheden die 3D-televisie ons in de toekomst kan bieden.

In hoeverre 'true 3D'-beelden zullen

doorbreken in de huiskamer is nog maar de vraag. Deze apparaten vereisen heel wat meer ruimte, omdat er niet alleen in de lengte maar ook in de diepte ruimte benodigd is.

Interactieve 3D-televisie voor meerdere personen zal in de medische wereld, maar ook in een groot aantal andere industrieën, nieuwe mogelijkheden bieden. Chirurgen kunnen bijvoorbeeld samen een operatie voorbereiden, en ook de resultaten van een CT- en MRI-scans kunnen op een nog interactievere manier worden bekeken dan nu al mogelijk is.

Hoe hard er ook gewerkt wordt aan alle nieuwe 3D-technologieën, het zal nog wel enkele jaren duren voordat 3D-televisie mainstream is.

#### Bronnen

3D Display: Current and future technologies in Europe (2007)  
Surman, P. & Kai Lee, W.  
<http://bit.ly/3dtelevisie>

Three Dimensional in the Air (2006)  
AIST Press Release  
<http://bit.ly/true3d>

*Advertentie*

*ASML*

# GPU: Parallele berekeningen



Rick van Galen  
Redacteur I/O Vivat

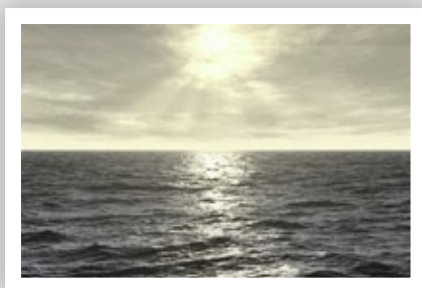
REKENKRACHT, GRAFISCHE PROCESSORS, SUPERCOMPUTERS, PARALLELISME, PROGRAMMEERPARADIGMA'S

## GPU bruikbaar voor meer dan beelden alleen

De softwarewereld kan steeds beter omgaan met de revolutie veroorzaakt door multicore processors aan het begin van deze eeuw. Er lijkt zich de komende jaren echter alweer een nieuwe revolutie aan te dienen, en dat uit onverwachte hoek. Het lijkt alsof de grafische processor, jarenlang niet meer geweest dan het extravagante speeltje van gamers, een enorm nieuw potentieel biedt voor grote hoeveelheden nieuwe rekenkracht.

### Specialistische hardware

In de laatste vijftien jaar heeft zich naast de centrale processor (CPU) een tweede processor in de computer verweven; de grafische processor (GPU). Al met de introductie van de IBM PC in de jaren '80 kwamen er vele 'grafische kaarten' in omloop om beeldschermen aan te kunnen sluiten, al dan niet met hardwarematige versnelling van sommige grafische processen. Jarenlang was de grafische kaart niets meer dan een hulpmiddel voor de rest van de computer om grafische taken snel uit te kunnen voeren; in de eerste plaats alleen met het tekenen op een tweedimensionaal vlak, maar later ook in de driedimensionale



Figuur 1: Voorbeeld van het werk van shaders met golven en zonlicht

ruimte. Op dat moment ontwikkelde de grafische kaart zich tot een plankje in de computer met een GPU erop, een specialistische extra processor die het tekenen van met name videogamewerelden en industriële toepassingen in de vorm-

is dergelijke hardware gemeengoed in computers.

### Generieke berekeningen

Een bijkomend effect van de introduc-

## Op topmodel GPU van AMD zitten 1600 shaders

geving en modellering aanmerkelijk versnelde. GPU's werden steeds sneller en geavanceerder, met extra toevoegingen om steeds meer grafische effecten te introduceren.

Met de introductie van Windows Vista in 2006 werd ook een nieuwe versie van de grafische bibliotheek Direct3D uitgebracht. Deze bibliotheek introduceerde het concept van unified shaders. De zogenaamde shaders waren daarvoor al een aantal jaar aanwezig in grafische hardware, en zorgden voor grafische zaken als lichteffecten en -vorming, schaduwen en golven in wateroppervlakken. In oude grafische hardware zijn deze effecten verdeeld over specialistische shaders die allen verschillende taken konden uitvoeren. Elk type shader had een verschillende instructieset en moest dus op een andere manier worden geprogrammeerd. Het concept unified shader gaat er juist om dat een dergelijke shader alle effecten en instructies uit kan voeren. Dit maakte het makkelijker om GPU's sneller te maken, en was voor ontwikkelaars makkelijker om te programmeren. Al snel volgde ook de eerste hardware van fabrikanten die hiermee om kon gaan. Inmiddels

tie van deze nieuwe vorm van shaders was het feit dat shaders in staat waren veel generiekere operaties uit te voeren op data. Net als op een CPU was het nu mogelijk om efficiënt 'gewone' berekeningen als "85.0 gedeeld door 3.0" en "286 min 412" uit te voeren. Grafische hardware biedt echter het voordeel dat er heel veel shaders op een GPU zitten: op het huidige topmodel GPU van fabrikant AMD zitten 1600 shaders, die allemaal in parallel berekeningen kunnen uitvoeren. Hoewel een enkele shader niet in de buurt komt van een CPU qua berekeningssnelheid, kan een hele GPU met al zijn parallel werkende shaders vele malen sneller ingewikkelde parallele berekeningen uitvoeren.

Deze toepassingen werden al snel ingezien door ontwikkelaars. Distributed computing projects, programma's die vrijwillig op computers worden gedraaid om mee te rekenen aan wetenschappelijke projecten, maakten al snel gebruik van deze nieuwe kracht. Al snel werd de berekeningskracht van het Folding@Home-netwerk, dat het uitvouwen van eiwitten voor medische toepassingen berekent, vertienvoudigd toen de nieuwe versie van het program-



ma gebruik kon maken van de nieuwe generatie grafische chips. Het concept werd tot GPGPU gebombardeerd: “general purpose graphics processing unit”. De unified shaders waren verworpen tot compute shaders, een naam die weergeeft waar de echte kracht van de nieuwe shaders lag – pure rekenkracht.

Het blijkt dat deze mogelijkheid tot snelle parallele berekeningen vooral voordelen biedt aan complexe simulaties en grote natuurkundige berekeningen. Dit had als gevolg dat GPGPU's werden geïntroduceerd in serverhardware; waar servers vaak nooit een grafische component nodig hadden, zijn er nu verschillende series hardware als de NVIDIA Tesla en de ATI FireStream

huidige grafische kaarten is er capaciteit over om complexe natuurkundige berekeningen als wapperende vlaggen en vloeistofmechanica uit te voeren op de grafische kaart. Op dit moment wordt er gewerkt aan het decoderen van video over de shaders van een GPU, in plaats van via de ingebouwde decodeerchip van de grafische hardware of de CPU. In de toekomst wil Microsoft een versie van Internet Explorer uitbrengen die de GPU gebruikt om snel internetpagina's te renderen. Ook is het zo dat bepaalde compressiealgoritmen gebruik kunnen maken van de verwerkingskracht van de GPU.

De verwachting is dat de GPGPU's weer een nieuw paradigma voor program-

## GPGPU's een nieuw paradigma voor programmeurs

die bijvoorbeeld niet alleen bij het renderen van 3D-scenes moeten helpen, maar ook het verwerken van complexe natuurkundige data aanmerkelijk moeten versnellen. Inmiddels verkopen een aantal fabrikanten kleine serverkastjes op basis van een aantal GPU's die de pure rekenkracht van supercomputers kunnen benaderen. Deze kastjes kunnen gewoon in kantoren worden geplaatst en zijn daarom benoemd tot deskside supercomputers.

### [Titanenstrijden en de toekomst](#)

Maar hoe merkt de 'gewone' thuisgebruiker deze nieuwe revolutie? De huidige toepassingen liggen bij games. Op

meurs gaan introduceren; nadat een aantal jaar geleden multicore processoren werden geïntroduceerd en programmeurs moesten denken in het opdelen van hun taken, moeten programmeurs in de toekomst wellicht denken in het opsplitsen van hun taken in rekenintensieve taken voor de GPGPU en in niet-rekenintensieve taken voor de CPU.

De adoptie van de GPGPU voor gewone programma's voor huis- en kantoorgebruik wordt echter nog bemoeilijkt door het ontbreken van een makkelijke manier om dit alles te programmeren. NVIDIA was lange tijd marktleider in het GPGPU-gebied met hun bibliotheek CUDA. Met deze methode is men

## Hoe werkt OpenCL?

OpenCL is de dominante API om de grafische kaart aan te spreken om berekeningen uit te voeren. OpenCL werkt, zoals andere huidige API's, met het opstellen van berekeningen en die vervolgens door de GPU te halen. Een opgestelde berekening wordt in OpenCL-jargon een kernel genoemd. Kernels zijn in een variant van C geschreven die gecompileerd kan worden naar de shaderinstructieset van GPU's.

Allereerst wordt er een context gedefinieerd waaruit OpenCL-kernels uit moeten voeren. Een context geeft bijvoorbeeld aan over welke apparaten een berekening uitgevoerd moet worden: bij gebrek aan een GPU kan OpenCL ook de berekeningen op de CPU doen, of een deel van het werk afstaan aan de CPU. Daarna moeten de apparaten worden aangesloten op de context, en vervolgens kan een kernel worden gecompileerd, geladen en uitgevoerd. Omdat dit veel 'bureaucratisch werk' is voor een berekening, kan hier niet een simpel voorbeeld worden gegeven. “Hello World” geschreven in OpenCL is bijvoorbeeld al meer dan 200 regels lang. Zie de referenties voor de voorbeeldcode hiervan.

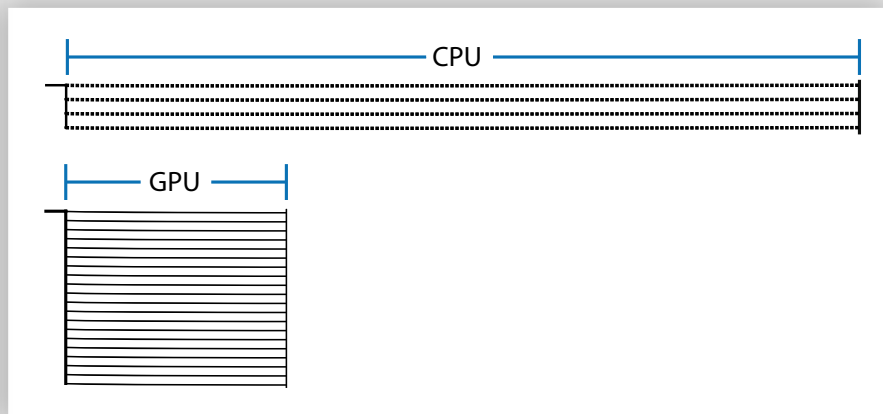
## Waarom is de GPU zo snel?

De reden dat een GPU veel sneller kan zijn dan een normale processor zit hem in het feit dat deze heel veel kleine parallele rekeneenheden naast elkaar heeft. Een normale CPU-kern heeft niet alleen rekenfunctionaliteit, maar bevat ook logica-, geheugenbewerkings- en beslissingsinstructies. Daardoor is het een veel complexere rekenkern dan een simpele shader. Een enkele shader is veel simpeler en trager dan een CPU-kern, maar heeft dus het voordeel dat het door zijn eenvoud er heel veel naast elkaar kunnen worden geplaatst.

Een voorbeeld van een berekening. Een complexe natuurkundige berekening die op te splitsen is in 10.000 kleinere berekeningen die individueel kunnen worden uitgevoerd. We nemen een quadcore-CPU die vier berekeningen in parallel kan uitvoeren. Over elke berekening doet een kern 3ns. Daarnaast zit een grafische processor met 200 shaderkernen. Elke shaderkern doet over een berekening 12ns. Een makkelijk reken-sommetje laat zien dat de CPU over deze taak 7500ns doet, en de GPU 600ns, een factor 12.5 verschil.

Hoewel dit voorbeeld versimpeld is, en niet meeneemt dat elke berekening niet even lang duurt en er onderlinge afhankelijkheid kan zijn tussen berekeningen, illustreert het wel de brute kracht die er in kleine shaders zit.

van zowel NVIDIA-hardware als -softwareondersteuning afhankelijk, en dit is dus geen bijster groot succes. Microsoft introduceerde met Windows 7 een bibliotheek DirectCompute als onderdeel van het besturingssysteem, maar deze bibliotheek is alleen beschikbaar voor Windows Vista en Windows 7. Men verwacht dat OpenCL, de bibliotheek die standaard meegeleverd gaat worden met Mac OS X en de meeste Linux-distributies en bovendien ook ondersteund wordt onder Windows, de standaard wordt. Echter zijn al deze bibliotheken geen pretje om in te programmeren, omdat de berekeningen via indirecte aanroepen aan de hardware



Figuur 2: Boven 4 CPU threads en onder vele parallele GPU threads.

worden gedaan.

Ook Intel moet zich in de strijd. Waar de hardware van concurrenten AMD en NVIDIA eigen typen shaders gebruiken, wil Intel een GPGPU introduceren die veel parallele rekeneenheden heeft op basis van de x86-architectuur. De shaders zijn gebaseerd op het oude simpele ontwerp van van de Pentium-processor uit 1996, en daardoor goed te clusteren in grote parallele groepen. Het bijkomend voordeel van dit concept is dat gewone x86-code, al jarenlang veruit meest dominante instructiecode, kan worden hergebruikt op deze nieuwe GPGPU. Of deze 'Larrabee'-chip, zoals Intel hem noemt, een revolutie gaat veroorzaken moet nog worden gezien. Intel heeft grote problemen met de markt-introductie van de Larrabee, omdat het maken van de chip tegen veel technologische grenzen aanloopt. Waarschijnlijk verschijnt deze in 2011.

AMD gaat als niet alleen directe concurrent van NVIDIA op GPU-gebied maar ook van Intel op CPU-gebied nog een stap verder. Recentelijk onthulde AMD wat details over hun Fusion-platform, de opvolger van hun huidige serie CPU's. In het Fusion-platform wordt de GPU onderdeel van de processor; binnen de Fusion-CPU, die in 2011 uit moet komen, zit een kleine GPU die 480 shaders bevat, veel meer dan de gemiddelde huis-, tuin- en keuken-GPU. Deze interne GPGPU kan bovendien zonder noemenswaardige restricties communiceren met de rest van de CPU, waar huidige GPU's afhankelijk zijn van de snelheid van het moederbord om te communiceren. Hoewel het concept van een GPU op een CPU niet nieuw is – dit is al gebruikelijk in mobiele processoren en een aantal toekomstige chips van Intel – is dit de eerste keer dat het

doel van deze grafische processor ook is om de hoofd-CPU te ondersteunen bij het uitvoeren van parallele berekeningen.

### Wachten op de ontwikkelaars

Het is nog afwachten wat de revoluties op softwaregebied gaan zijn die gebruik gaan maken van deze nieuwe bron van rekenkracht in computers. Het is de verwachting dat, net zoals bij multicore processors, deze revolutie langzaam zal gaan. De hardware is er echter al ruim klaar voor, en wordt in rap tempo volwassen.

## Bronnen

**MSDN Common Shader core**  
<http://msdn.microsoft.com/en-us/library/ee418282%28VS.85%29.aspx>

**NVIDIA Tesla Personal Supercomputer datasheet**  
<http://www.siliconmechanics.com/files/TeslaPSCDatasheet.pdf>

**Anandtech over Intel Larrabee**  
<http://anandtech.com/cpuchipsets/intel/showdoc.aspx?i=3367>

**AMD Introduction to OpenCL**  
<http://developer.amd.com/gpu/ATIS-treamSDK/pages/TutorialOpenCL.aspx>

**Apple's Hello World in OpenCL voorbeeld**  
[http://developer.apple.com/mac/library/samplecode/OpenCL\\_HelloWorld\\_Example/listing1.html](http://developer.apple.com/mac/library/samplecode/OpenCL_HelloWorld_Example/listing1.html)

# Top vijf 'IT-fails' in 2009

\*&^%\$#@! FAIL!

**H**oewel de technologie zich in 2009 op vele fronten in hoog tempo verder heeft ontwikkeld is het natuurlijk ook wel eens interessant om te kijken: wat ging er mis in 2009? Daarom: de vijf grootste 'IT-fails' van het voorbije jaar.

## 1. Twitter

Twitter werd diverse keren gehackt, waarbij ook de groeiplannen van het bedrijf werden buitgemaakt. Daarnaast heeft men nog geen verdienmodel gelanceerd.

## 2. Google Wave

Deze 'revolutie' werd door velen met open armen ontvangen, maar het werkelijke van nut van de toepassing is velen niet duidelijk.

## 3. Oracle neemt MySQL over

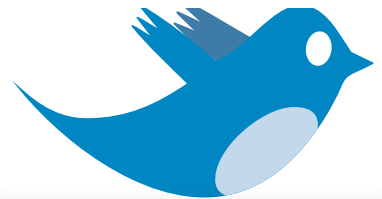
's Werelds grootste Open Source database MySQL werd overgenomen door 's werelds grootste betaalde databaseleverancier Oracle. Naar verwachting daalt het gebruik van MySQL de komende jaren met zo'n 10 procent.

## 4. China's grootste firewall

Sinds 1998 controleert de Chinese overheid al het internetverkeer middels een enorme firewall. In 2009 maakte China bekend ook censuursoftware te gaan installeren op iedere verkochte pc in het land.

## 5. fail!

2009 was het jaar van de Fail!-roepers. Iedere gelegenheid werd aangegrepen luid Fail! Te roepen. "Als iemand een pen laat vallen hoor je iedereen al – met hoge en harde stem..."



## Bronnen

CNN: 2009 Year in Review  
<http://bit.ly/O9fail>

Top 10 Failures of 2009  
<http://bit.ly/O9fail2>

# Conversational Agents



Marije  
de Heus  
Redacteur I/O Vivat

INTELLIGENT AGENTS, CONVERSATIONAL AGENTS, INSTRUCTIEPROGRAMMA'S, SOFT & HARD COMMUNICATION SKILLS

## Non-verbale en verbale communicatie van intelligentagents

Nederland doet het goed in de instructieprogramma-markt, zo was een tijdje terug op het NOS Journaal te horen. Een instructieprogramma is een programma waarin iets door een virtueel karakter wordt uitgelegd aan een mens. Op dit moment wordt bij de toepassing van dit soort programma's vooral gedacht aan het helpen van mensen die net een nieuwe baan hebben. Zo zou er een programma gemaakt kunnen worden voor iemand die bakker wil worden, waarin het virtueel karakter in een virtuele bakkerij laat zien hoe je de verschillende broden, koekjes, taarten et cetera maakt en bakt. Hierdoor kunnen alle studenten een soort privéles krijgen, zonder dat dit meteen onbetaalbaar is.

kunnen zelf nieuwe dingen leren, hun hele omgeving in de gaten houden en samenwerken met andere agents, terwijl simpele agents vaak maar één eigenschap van hun omgeving kunnen waarnemen en in hun eentje werken. Een conversational agent is een soort intelligent agent die zich bezighoudt met conversaties, het spreken en luisteren naar een of meer mensen.

De agents die worden gebruikt voor instructieprogramma's moeten waar kunnen nemen wat de persoon aan het doen is, zodat de agent kan reageren als het fout gaat. Daarnaast zal de agent waarschijnlijk meer informatie nodig hebben, bijvoorbeeld van de oven hoe warm die is ingesteld, en de tijd moet gecontroleerd worden. Bovendien moet

Als mensen een gesprek voeren vindt er niet alleen verbale, maar ook non-verbale communicatie plaats. Deze vorm van communicatie levert voor de deelnemers aan het gesprek extra informatie op. Aangezien het belangrijk is dat de communicatie tussen mens en agent zo natuurlijk mogelijk verloopt, moet de agent zo goed mogelijk de non-verbale communicatie, ook wel de soft communication skills genoemd, nadoen.

Het ligt voor de hand dat mensen als ze spreken naast gewone taal ook lichaamstaal gebruiken, maar ook de luisteraars zijn continu bezig met het uitzenden van non-verbale signalen. Deze signalen leveren weer informatie op voor de spreker: een luisteraar kan bijvoorbeeld door te knikken aangeven dat hij begrijpt wat de spreker vertelt, met gezichtsuitdrukkingen laten zien wat hij ergens van vindt, of met subtiele bewegingen aangeven dat hij graag het woord wil. Hoe deze bewegingen er uit zien verschilt van persoon tot persoon. Dit maakt het analyseren van non-verbale communicatie extra moeilijk.

Toch wordt er veel onderzoek gedaan naar het non-verbale gedrag van luisteraars. De vakgroep HMI doet bijvoorbeeld mee aan het internationale project SEMAINE, waarvoor ze onder andere vele uren aan opgenomen vergaderingen hebben geanalyseerd met behulp van een annotatieprogramma zoals ELAN (zie figuur 1). In zo'n programma kan je heel nauwkeurig aangeven wanneer iemand iets doet.

Er werd op verschillende soorten non-verbaal gedrag gelet: hoofdbewegingen, oogleden, richting van het hoofd, rich-

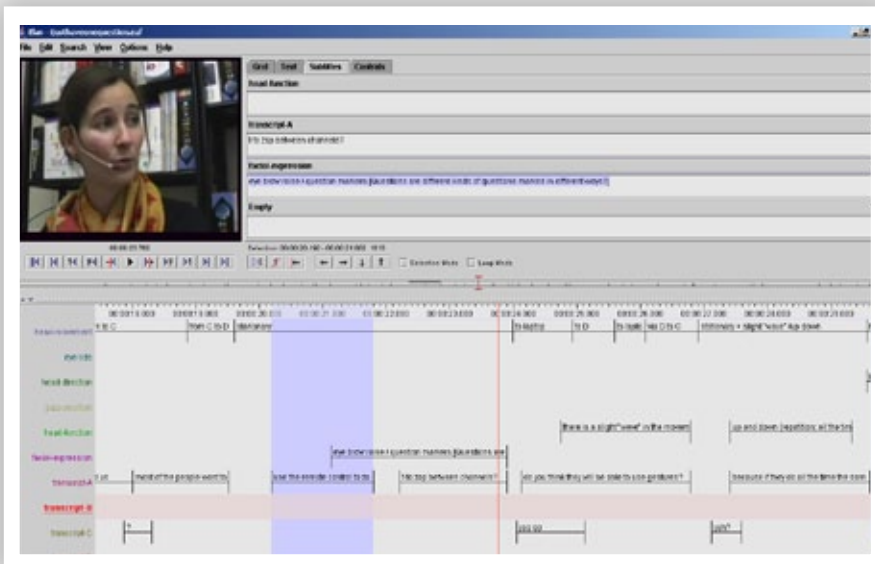
## Complexe agents kunnen zelf nieuwe dingen leren

Het virtuele karakter dat gebruikt wordt in een dergelijk programma wordt ook wel conversational agent genoemd. Een conversational agent is een soort intelligent agent. Een intelligent agent is een autonoom geheel, dat zijn omgeving waar kan nemen en aan de hand daarvan handelt richting een bepaald doel. In het geval van de bakkerij zal de intelligent agent in de gaten moeten kunnen houden wat de persoon in de echte bakkerij aan het doen is, bijvoorbeeld of hij een brood aan het kneden of aan het bakken is, en reageren als het fout gaat. Intelligent agents kunnen heel simpel zijn of heel complex, afhankelijk van hun taak. De meest complexe agents

de agent kunnen begrijpen wat de persoon zegt. Dit is de meest ingewikkelde taak, omdat er ontzettend veel woorden bestaan die op heel veel manieren gecombineerd kunnen worden, en er ook non-verbaal informatie over wordt gedragen. Het is praktisch niet mogelijk om met de hand elke mogelijke zin in het systeem te zetten en een goede reactie hierop te bedenken, dus daar moet wat anders voor worden bedacht. Hierover verderop in het artikel meer. Eerst volgt meer informatie over de non-verbale communicatie.

[Non-verbale communicatie](#)





ting van de blik en gezichtsuitdrukking. Zoals te zien is in de figuur kunnen er meerdere dingen tegelijk gebeuren en ook worden geannoteerd in het programma. Daarnaast is er ruimte voor de tekst die door de spreker wordt uitgesproken op dat moment zodat deze vergeleken kan worden met het gedrag van de luisteraar.

technieken gebruik moeten maken.

### Gelach

Non-verbale communicatie lijkt goed te overzien, maar verbale communicatie, oftewel hard communication skills, is een stuk ingewikkelder en er komt veel meer bij kijken om het goed te im-

## Om agents te verbeteren wordt het gedrag van mensen bestudeerd

Bij het analyseren van videomateriaal kan gekeken worden naar wat voor indruk een luisteraar maakt op de spreker. Als dit vervolgens gekoppeld wordt aan het gedrag van de luisteraar, kan er een redelijke voorspelling gemaakt worden over hoe een luisteraar overkomt. Dit kan dan gebruikt worden voor de agent: er kan zo bijvoorbeeld een vriendelijke agent worden gecreëerd, of een strenge, of een behulpzame. Een aantal opvallende conclusies tot nu toe zijn bijvoorbeeld dat mensen die kleinere bewegingen maken onderdanig overkomen, maar ook aantrekkelijk en dat mensen die grote bewegingen maken onvriendelijk overkomen.

Behalve de indruk die een luisteraar op iemand anders maakt, is er nog veel meer uit het videomateriaal op te maken. Spiegelen is het nadoen van elkaars lichaamstaal. Dit blijkt vaak te gebeuren, bijvoorbeeld bij de houding die mensen aannemen, maar ook is uit onderzoek naar voren gekomen dat luisteraars meestal kleine hoofdbewegingen maken op het ritme van de spreker. Een goede intelligent agent zou van dit soort

plementeren in een agent. Bij de communicatie tussen een mens en agent is het niet alleen belangrijk dat de agent de spraak herkent en de verschillende woorden kan onderscheiden, en vervolgens een goed verstaanbaar antwoord geeft, maar ook dat hij 'begrijpt' wat de mens zegt en een antwoord geeft dat daarbij past. In simpele gesprekjes met een beperkt aantal mogelijkheden is dit goed te doen, maar in de praktijk moet een agent over een enorm grote woordenschat beschikken. Er wordt geëxperimenteerd met het gebruiken van zogenaamde abstracte automaten om taal mee weer te geven, maar tot nu toe is dit slechts theorie en wordt het nog weinig toegepast.

Waar wel veel over bekend is, is het onderscheid tussen het geluid van gelachen en van gepraat. Het herkennen van gelachen is belangrijk, want door te lachen geeft iemand aan hoe hij zich voelt. Lachen is volgens veel onderzoekers de meest voorkomende soort non-verbaal gedrag.

De succesvolste manier om de karakte-

ristieken van lachen te onderzoeken, is om zowel de audio als de video tegelijk te analyseren. Dan wordt duidelijk dat lachen zich niet alleen onderscheidt van spraak door het geluid, maar ook door bewegingen, zoals de gezichtsuitdrukking. Prosody is de verzamelnaam voor het ritme, de lengte en de grootte van gebaren die worden gemaakt tijdens het lachen en spreken. Een combinatie van prosody, hoofdbewegingen en gezichtsuitdrukking blijkt een goede voorspeller te zijn van of een persoon lacht of praat.

### Toekomst

Op dit moment wordt er nog volop onderzoek gedaan naar virtuele karakters. Het is wellicht onmogelijk om een 'perfecte' agent te maken, dus het onderzoek zal voorlopig in ieder geval nog wel even doorgaan. Er zijn al intelligent agents in gebruik, veelal in instructieprogramma's. Bij de bronnen staat een link naar een site waar je je eigen intelligent agent kan maken door bepaalde karakteristieken van een persoon te kiezen, of zelfs je eigen foto kan gebruiken om een agent te maken. In de toekomst zal er steeds meer gebruik worden gemaakt van conversational agents in instructieprogramma's, maar er zijn nog veel meer toepassingen te bedenken. Misschien zijn er over tientallen jaren wel robots die bij je thuis schoonmaken en met jou kunnen praten gebruik makend van dezelfde technieken als conversational agents, of kan je oma bridgen met een conversational agent!

### Bronnen

**Multimodal Backchannel Generation for Conversational Agents**  
D.K.J. Heylen

**'Audiovisual Discrimination between Laughter and Speech'**  
Petridis and M. Pantic

**'Conversational Agents', The Practical Handbook of Internet Computing,**  
Chapman & Hall (2004)  
Lester, J.; Branting, K.; Mott, B.

ELAN  
<http://www.lat-mpi.eu/tools/elan>

maak je eigen intelligent agent  
<http://www.sitepal.com/>

# Aspect-oriented programming



Jelte  
Zeilstra  
Redacteur I/O Vivat

CODE ORDENEN, BASISPROGRAMMA, UITBREIDEN, ADVICES, JOIN POINTS, ASPECTJ

## Je programma in logische eenheden opdelen

Iedereen die wel eens gewerkt heeft met een objectgeoriënteerde programmeertaal weet dat het belangrijk is om je programma goed op te delen in logische objecten met een heldere structuur, om te zorgen dat je een beetje overzicht houdt. In een administratiesysteem voor banken zullen een rekening en een rekeninghouder twee verschillende klassen zijn, omdat het twee verschillende onderdelen zijn.

Als er echter een bedrag van de ene rekening naar de andere moet worden overgeschreven, volstaat het niet om alleen het bedrag bij de ene rekening af te

beveiliging en loggen vaak verspreid is door het hele programma. Hierdoor raak je het overzicht kwijt over deze onderdelen. Aspectgeoriënteerd programmeren (AOP) is een mogelijke oplossing voor dit probleem.

### Wat is het

Aspectgeoriënteerd programmeren probeert dit probleem op te lossen door deze onderdelen los te koppelen van het basisprogramma. Het basisprogramma hoeft zich helemaal niet bezig te houden met de aspecten, zoals de beveiliging. In plaats daarvan schrijf je het basispro-

of eigenschappen aan klassen, of de uitvoer van bestaande methoden te beïnvloeden. Bij het implementeren van de beveiliging geeft je vervolgens aan dat voor het uitvoeren van de te beveiligen methoden een ander stukje code moet worden uitgevoerd. Dat extra stukje code wordt een advice genoemd. In zo'n advice kun je bepaalde voorwaarden testen. Bijvoorbeeld of iemand wel genoeg rechten heeft. Vervolgens kun je bepalen of de oorspronkelijke methode wel of niet moet worden uitgevoerd. Maar er is nog meer mogelijk bij aspectgeoriënteerd programmeren. Je kunt ook de teruggegeven waarde van een methode veranderen, of een compleet andere methode uitvoeren. Je kunt hiermee bijvoorbeeld caching-functionaliteit toevoegen.

### Techniek

Aspectgeoriënteerd programmeren is voor veel gangbare programmeertalen mogelijk, zoals Java, C/C++, Python en C#. Hiervoor zijn extra modules of tools ontwikkeld, ook wel aspectoriented languages genoemd. Voor Java is onder andere AspectJ ontwikkeld. Een aspect-oriented language definieert een aantal join points, plaatsen in een programma waar een stukje extra code, een advice, kan worden ingevoegd. Voorbeelden van join points zijn het begin en einde van een methode of het creëren van een nieuw object. Afhankelijk van de gebruikte aspect-oriented language zijn bepaalde join points wel of niet beschikbaar. Door middel van zogenaamde pointcuts worden advices aan join points gekoppeld. Dit gebeurt onder bepaalde voorwaarden, zoals dat het join point een methodeaan-

## Beveiliging niet meer verspreid over allerlei klassen

halen en de andere op te tellen. Je moet ook controleren of de gebruiker wel dat bedrag mag overschrijven, en je wilt ook een melding maken in het logboek.

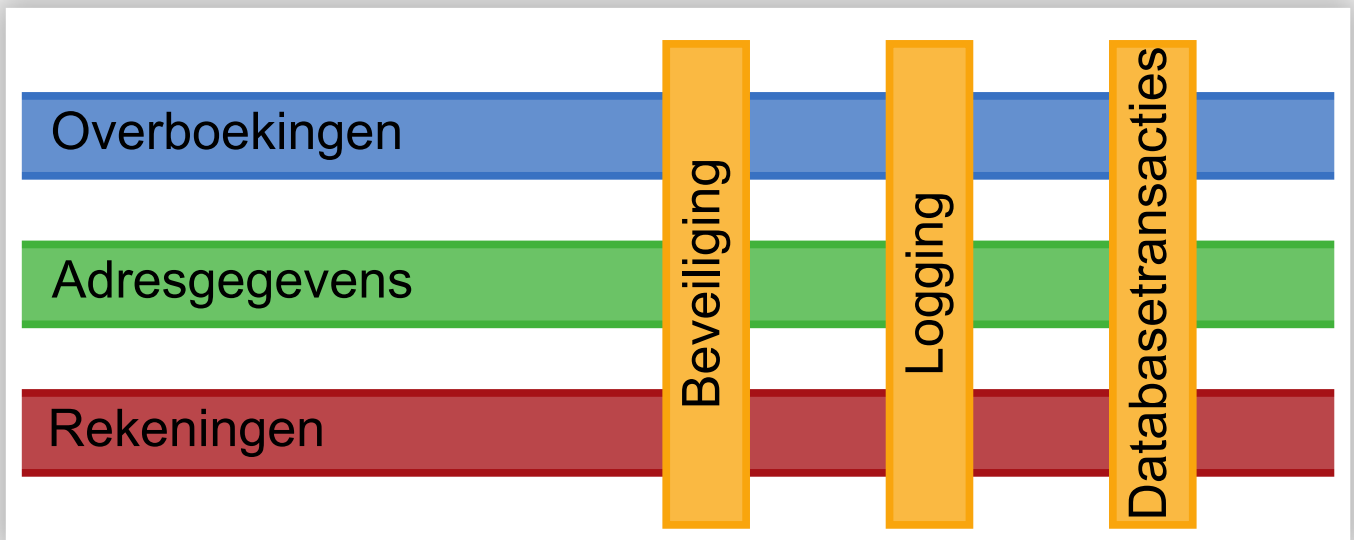
Je kunt dit alles doen in een methode van een rekeninghouder, maar dan komt er in die rekeninghouder-klasse ook beveiligings- en logboekcode te staan, terwijl dit toch een heel ander onderdeel is. Je zult vaak merken dat

programma rechttoe rechtaan. Hierbij laat je dus de beveiliging en het loggen volledig weg. Het basisprogramma is simpel en to the point. Het programmeren van de beveiliging van je programma doe je vervolgens centraal. Bij aspectgeoriënteerd programmeren pas je eigenlijk je basisprogramma aan. Je voegt daarbij op allerlei plaatsen functionaliteit toe aan je programma. Dit kan bijvoorbeeld door het toevoegen van extra methoden

```
void transfer(Account fromAccount, Account toAccount,
int amount) {
    if (fromAccount.getBalance() < amount)
        throw new InsufficientFundsException();

    fromAccount.withdraw(amount);
    toAccount.deposit(amount);
}
```

*De simpele code zoals je het graag ziet.*



Figuur 1: De aspecten (geel) overlappen de verschillende delen van het banksysteem.

roep moet zijn van een object van een bepaalde klasse. Een ander voorwaarde kan zijn “alle aanroepen van methodes waarvan de naam begint met ‘set’ van een object uit het com.company package”. Voorwaarden kunnen met AND, OR, en NOT met elkaar gecombineerd worden.

Uiteindelijk zullen de advices samengevoegd worden met het basisprogramma. Dit wordt weven genoemd. Dit kan op vijf verschillende plaasten gebeuren: in de broncode (dus voor het compileren), tijdens het compileren met een speciale

taal en aspect-oriented language welke van deze methoden gebruikt wordt.

#### Gebruik

Deze technieken kunnen voor veel verschillende onderdelen van een programma gebruikt worden. Hiervoor noemde ik al beveiliging, logging en caching.

Bij de beveiliging wordt gebruik gemaakt van de mogelijkheid om een methodeaanroep te beëindigen voordat de methode echt is uitgevoerd. Vaak wordt

worden gemaakt van de teruggegeven waarde van de basismethode om te beslissen of en welke gegevens naar het logboek geschreven worden. Bij caching wordt een basismethode alleen uitgevoerd als er nog geen uitkomst bekend is. Na het uitvoeren van de methode wordt de uitkomst opgeslagen en later weer gebruikt. Het is ook erg makkelijk om aan te geven welke andere methodeaanroepen er voor zorgen dat de cache gewist wordt.

Een hele andere toepassing is het updaten van de user interface. Als bijvoorbeeld de inhoud van een tekstvak gewijzigd wordt, kan door middel van aspecten het woordenaantal opnieuw berekend worden en de spellingscontrole opnieuw uitgevoerd worden. Normaal gesproken zou je dit soort wijzigingen in een update-event moeten aangeven. Doordat dit niet meer hoeft neemt de overzichtelijkheid sterk toe.

#### Voordelen

Het belangrijkste voordeel van dit aspectgeoriënteerd programmeren is dat je op deze manier echt alle onderdelen van je programma logisch kunt opsplitsen en indelen. Dit in tegenstelling tot gewoon objectgeoriënteerd programmeren, waarbij je gedwongen was je beveiliging in alle te beveiligen methodes te implementeren. Doordat deze onderdelen los worden toegevoegd, hoeft je er bij het programmeren van je basisfunctionaliteit van je programma geen rekening mee te houden. Ook hoeft je er bij later al je klassen aan te passen om toch de caching en logging in te voegen. Hierdoor neemt het overzicht in je code aanzienlijk toe.

```
void transfer(Account fromAccount, Account toAccount,
int amount)
    throws Exception {
    if (!getCurrentUser().canPerform(OP_TRANSFER))
        throw new SecurityException();
    if (amount < 0)
        throw new NegativeTransferException();
    Transaction tx = database.newTransaction();
    try {
        if (fromAccount.getBalance() < amount)
            throw new InsufficientFundsException();
        fromAccount.withdraw(amount);
        toAccount.deposit(amount);
        tx.commit();
        systemLog.logOperation(OP_TRANSFER, fromAccount,
            toAccount, amount);
    } catch(Exception e) {
        tx.rollback();
        throw e;
    }
}
```

*De uitgebreid zoals het vaak helaas wordt. Het vetgedrukt is het oorspronkelijke deel.*

compiler, in de bytecode (dus na het compileren), en tijdens of vlak voor het uitvoeren van het programma. Het is sterk afhankelijk van de gebruikte basis-

er dan ook een foutmelding weergeven.

Het loggen gebeurt vaak aan het einde van de methode. Hierbij kan gebruik

Behalve de overzichtelijkheid in je basisprogramma, wordt ook het overzicht over de aspecten erg verbeterd. Door het aspectgeoriënteerd programmeren komt alle gerelateerde code bij elkaar te staan. Je kunt nu in een opslag zien hoe de beveiliging in je programma geregeld is.

Als er dan ook nog meerdere advices aanspraak maken op het zelfde join point, kan dit tot grote onvoorspelbaarheid leiden. Bij dit soort situaties moet je er goed op letten dat er 'voorrangsregels' opgesteld worden.

Een ander probleem is de snelheid

```
pointcut set() : execution(* set*(*) ) && this(Point)
&& within(com.company.*);
```

Definieert *pointcut* *x* bij het uitvoeren van een *set\**-methode van een *Point*-object in het *com.company*-package.

```
after() : set() {
    Display.update();
}
```

Definieert het *advise*: voer na *pointcut set()* de methode *Display.update()* uit.

Het grootste voordeel is te behalen als bijvoorbeeld het beveiligingssysteem later moet worden gewijzigd. Omdat je beveiliging niet meer verspreid is over allerlei klassen, hoef je niet meer de hele code door te spitten. Het is op deze manier ook mogelijk om de complete beveiliging uit te schakelen tijdens het testen, of juist niet uitgebreid te loggen in de Productieomgeving.

### Nadelen

Geen enkel systeem heeft alleen maar voordelen, en dus heeft aspectgeori-

waarmee het programma uitgevoerd wordt. Zeker als de aspecten dynamisch (tijdens het uitvoeren) geweven worden, gaat dit ten koste van de snelheid. Dit kan worden voorkomen door de broncode of de bytecode al te verweven.

### Toepassingen en onderzoek

Aspectgeoriënteerd programmeren wordt nog lang niet zoveel toegepast als het bekende objectgeoriënteerd programmeren, bekend van onder andere Java. Er zijn verschillende frameworks die gebruik maken van aspectgeoriën-

Door gebruik te maken van aspectgeoriënteerd programmeren kan je ervoor zorgen dat alle aspecten centraal beheerd kunnen worden en niet verspreid zijn over allerlei klassen binnen je project. Dit kan ervoor zorgen dat de code helder is en het vergemakkelijkt het onderhoud aan de aspecten.

Tegelijk kan het er ook voor zorgen dat het verloop van het programma juist onduidelijk of zelfs onvoorspelbaar wordt. Ook laat de snelheid soms te wensen over.

Bij veel programmeertalen is het al mogelijk om aspectgeoriënteerd te programmeren, maar desondanks wordt het nog niet veel gebruikt. AOP wordt echter wel steeds populairder. Misschien weten we over tien jaar niet anders meer.

## Heldere code en vergemakkelijkt onderhoud

enteerd programmeren ook een aantal nadelen. Een deel van de problemen wordt veroorzaakt door het feit dat ondanks de groeiende populariteit, AOP nog niet echt wijdverspreid is. Zo zijn er nog niet heel veel tools op de markt ter ondersteuning van het aspectgeoriënteerd programmeren en is er ook nog vrij weinig onderwijs, zoals cursussen en boeken, beschikbaar. Doordat delen van het programma verplaatst zijn buiten het basisprogramma en het basisprogramma geen weet heeft van de toegevoegde aspecten, is aan de code ervan niet precies te zien wat er gebeurt. Een methodeaanroep kan zomaar geblokkeerd worden door een advice, terwijl je daar niets van ziet in je basiscode. Hierdoor is niet direct duidelijk hoe het programma eigenlijk verloopt en wordt het debuggen dus bemoeilijkt.

teerde technieken, zoals Spring (voor Java).

Ook gebruikt Siemens deze technieken in Soarian, een systeem voor het beheren van medische gegevens.

Een voorbeeld van onderzoek bij de Universiteit Twente, is het onderzoek naar het combineren van verschillende domeinspecifieke aspectgeoriënteerde talen. Dit zijn aspectgeoriënteerde programmeertalen die zich op één onderdeel richten, zoals beveiliging. Het combineren is lastig, vanwege de verschillen en overlap tussen de talen, wat voor conflicten of onvoorspelbaar gedrag kan zorgen.

### Slot

### Bronnen

Prototyping and Composing Aspect Languages: using an Aspect Interpreter Framework. In: Proceedings of 22nd European Conference on Object-Oriented Programming  
Havinga, W.K. and Bergmans, L.M.J. and Ak-it, M. (2008)

Interview met W.K. Havinga op 27 november 2008.

"Aspect-oriented programming", Wikipedia, The Free Encyclopedia,  
[http://en.wikipedia.org/w/index.php?title=Aspectoriented\\_programming&oldid=256503375](http://en.wikipedia.org/w/index.php?title=Aspectoriented_programming&oldid=256503375)

"Aspect-oriented software development", Wikipedia, The Free Encyclopedia  
[http://en.wikipedia.org/w/index.php?title=Aspectoriented\\_software\\_development&oldid=251187197](http://en.wikipedia.org/w/index.php?title=Aspectoriented_software_development&oldid=251187197)

*Advertentie*  
*ING*

# Word jij niet gecontroleerd?



Bas  
Stottelaar  
Redacteur I/O Vivat

HACKING, ELEKTROMAGNETISCH,  
ONTCIJFEREN, SLEUTEL, GEHEUGEN

## Onzichtbare controle door middel van hardware

Tegenwoordig kun je niet meer het internet op zonder bloot te worden gesteld aan allerlei spam, malware, trojan horses, hackers en virussen. Met behulp van softwarematige oplossingen, zoals firewalls en

er dat niemand ons afluistert?

Denk bijvoorbeeld aan een packet sniffer of keylogger. Deze voorbeelden hoeven, per definitie, niet uitgevoerd te worden als een stukje software dat op het doelwit draait, maar kunnen ook

## Zonder dat we het doorhebben kan iemand ons dus controleren.

virusscanners, proberen we de effecten van deze kwaadaardige 'dingen' zo goed mogelijk tegen te gaan. Dat het woord softwarematig in de vorige zin gebruikt wordt, doet misschien al wel iets vermoeden. Want hoe vol kunnen we een computer stoppen met software om deze te beschermen, en wie garandeert

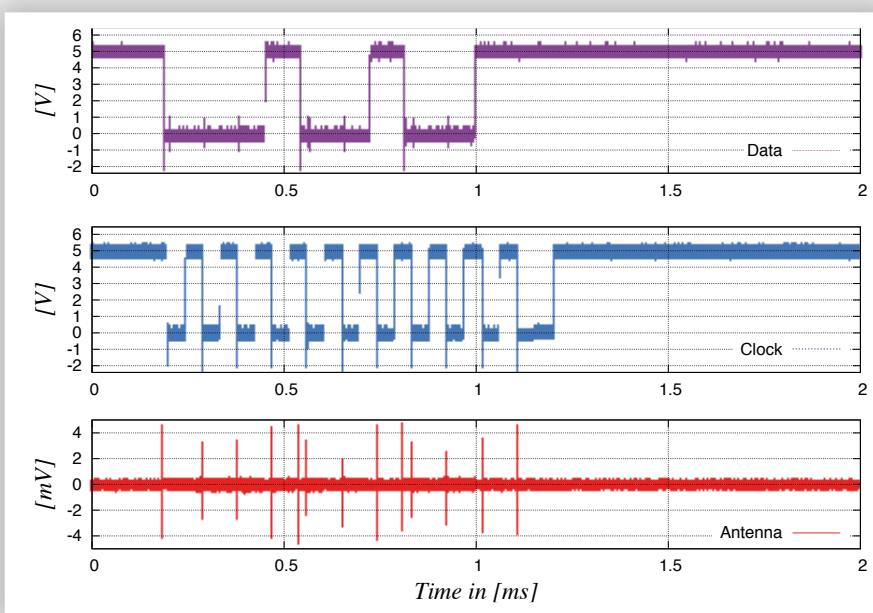
uitgevoerd worden in een stukje hardware. Met dank aan de hedendaagse techniek hoeft een keylogger tegenwoordig niet veel groter te zijn dan een penlite batterij. Deze kan dus onopvallend geïnstalleerd worden. Zonder dat we het doorhebben kan iemand ons dus controleren.

Dit artikel beschrijft drie studies die het abstract gezien mogelijk maken om ons te controleren. Het eerste onderzoek gaat over het uitlezen van toetsenbordsaanslagen op afstand, het tweede onderzoek over het uitlezen van geheugen nadat een computer is afgesloten. Het laatste onderzoek kan ons wel erg letterlijk controleren. Hier gaat het over het gebruik van draadloze signalen om door muren heen te kijken. Er wordt niet ingegaan op de complexe algoritmes en diepgaande details.

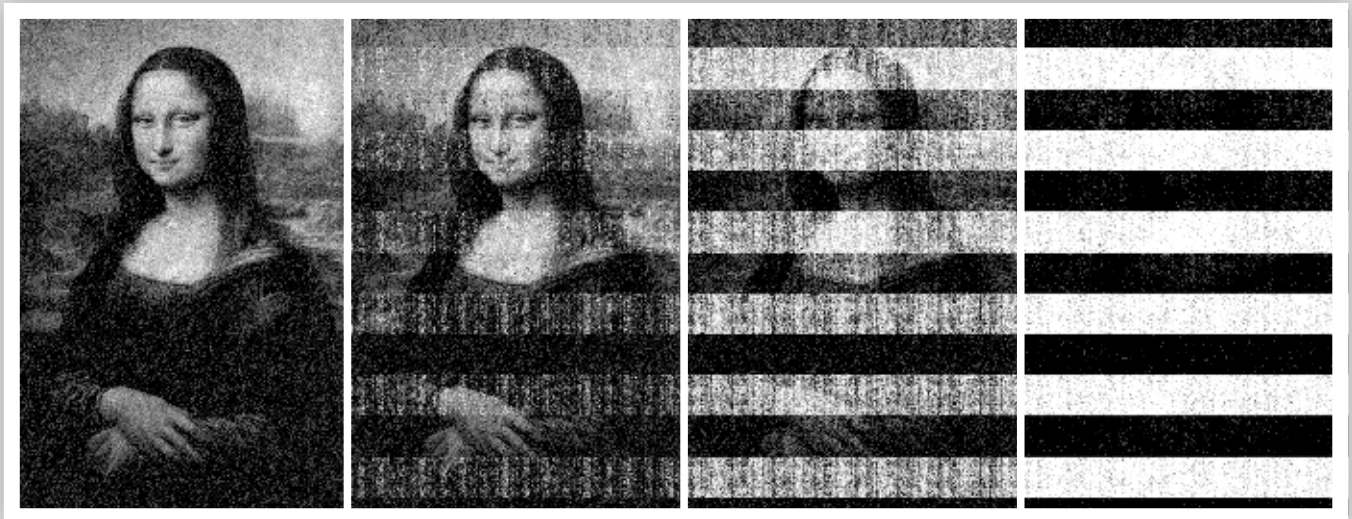
### Electromagnetic Emanations

Een elektromagnetisch effect treedt op bij alles waar stroom door vloeit. In sommige gevallen kan dit storen op andere hardware, denk bijvoorbeeld aan een GSM die kan storen op de radio. Twee studenten, Martin Vuagnoux en Sylvain Pasini, hebben in 2008 een onderzoek gepubliceerd, waarin ze op afstand konden registreren wat er op een toetsenbord getypt werd. Een soortgelijk iets is ook al eens tijdens de Tweede Wereldoorlog onderzocht. De Amerikanen bezaten een typemachine die tekst versleutelde, maar als bijeffect de toetsaanslagen ook de ether in stuurde. Per toeval werd ontdekt dat een oscilloscoop in een nabij laboratorium, steeds spikes weergaf als een bericht getypt werd. Uit nadere inspectie bleek het mogelijk te zijn om het onversleutelde bericht uit die spikes af te leiden.

Bij elke toetsaanslag op een toetsenbord, wordt een digitaal signaal, bestaande uit ééntjes en nulletjes, naar de computer verzonden. Dit signaal wordt gemaakt door een transistor die het signaal hoog of laag maakt. Dit heeft als bijeffect dat



Figuur 1: Drie diagrammen van de datalijn, de kloklijn en de antenne. De toetsaanslag wordt via het PS/2 protocol doorgegeven. De data die verzonden wordt is 0 00100100 1 1 (scancode 0x24).



Figuur 2: Het meest linker plaatje wordt op in het geheugen gestopt. Na het onderbreken van de spanning is het tweede figuur waargenomen op 30 sec, het derde figuur op 60 sec en het laatste figuur op 5 min.

er een elektromagnetisch veld opgewekt wordt. Met een hele gevoelige antenne, die een breed spectrum opvangt, kan dit signaal (met een amplitude in de orde van enkele millivolts) geregistreerd worden.

In de resultaten van het onderzoek worden vier methoden gepresenteerd, waarvan de eerste besproken wordt. Deze maakt het mogelijk om toetsaanslagen terug te halen met één bit aan onzekerheid (andere doen het zelfs nog beter). Om bovenstaande figuur te kunnen ontcijferen is het essentieel om te weten hoe PS/2 werkt. Zodra de klok laag is, wordt een databit gelezen door de computer. Er worden 11 bits verstuurd, waarbij de eerste bit een startbit is (0), vervolgens 8 databits (de scancode, uniek voor elke toets), een oneven pariteitsbit over de databits en een stopbit (1). In figuur 1 is bij elke neergaande flank van de klok en de data een piek in het onderste diagram te zien, en zelfs een dubbele als zowel de klok als de data een neergaande flank vertonen. Een dubbele piek in diagram drie geeft aan dat dit de neergaande flank is van één of meerdere 1-en. Hierdoor kan er niet met zekerheid nagegaan worden of 0x24 (00100100) of 0x34 (00110100) verzonden is, omdat er twee opeenvolgende 1-en verstuurd worden. Toch kan het aantal pogingen om achter een wachtwoord te komen flink worden teruggebracht. Stel dat scancode 0x24 lijkt op nog drie andere scancodes, dan zijn er voor een wachtwoord van twee tekens 16 combinaties mogelijk ( $4^2$ ). Dat is heel iets anders dan  $36^2$  (cijfers + letters, geen hoofdletters).

Vuagnoux en Pasini presenteren vier methodes om de signalen te ontcijferen. In hun onderzoek hebben ze 12 verschillende toetsborden gebruikt, waaronder USB-, draadloze- en laptop-toetsborden. Het is gelukt om verschillende toetsborden van elkaar te onderscheiden op basis van afwijkingen in de bit-tijden. Het effect van de elektromagnetische straling wordt beter waarneembaar wanneer de computer of laptop aan het lichtnet hangt. De voe-

bitwaarde) niet kwijt te raken. Gebeurt dit verversen niet, dan loopt de condensator langzaam leeg, en als deze voorbij het omslagpunt (de grens of het nu een één of nul is) komt, kun je niet met zekerheid meer zeggen of de bitwaarde nu een één of nul was. Het kan enkele seconden tot minuten duren voordat het geheugen de bitwaarde kwijt is, afhankelijk van het soort geheugen en systeem.

## Zelfs na tien minuten de inhoud van het geheugen vrijwel compleet terug te halen.

dingskabel en het lichtnet vormen dan een langere antenne. Het onderzoek is uitgevoerd onder ideale omstandigheden (weinig achtergrondruis), maar ook op een normale werkvloer, waar meer achtergrondstraling is van apparatuur. Het verwerken van de data kost op dit moment nog erg veel rekenkracht en is daarom niet realtime.

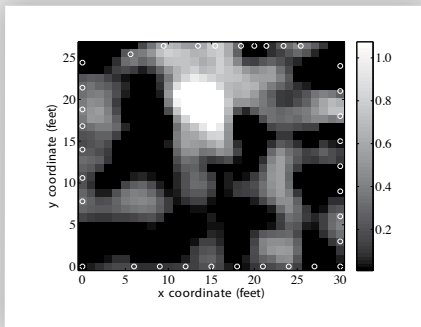
### Koud kunstje

Tijdelijke gegevens worden (meestal) in het RAM-geheugen bewaard, totdat de computer uitgezet wordt of wordt herstart. Daarna zijn alle gegevens kwijt, althans, dat verwacht je...

Helaas is dit niet direct zo. Een enkele bit in het dynamische RAM-geheugen bestaat uit een kleine condensator die een aantal keren per seconden ververst moet worden om de lading (de

Dit concept hebben een aantal onderzoekers van de Princeton University (New Jersey, USA) verder onderzocht op risico's. Allereerst koelen ze het RAM-geheugen om de zelfontlading van de condensatoren zo veel mogelijk te beperken. Hierdoor is het mogelijk om zelfs na tien minuten de inhoud van het geheugen vrijwel compleet terug te halen.

Het is vaak niet eens belangrijk om alles uit het geheugen terug te halen. Bijvoorbeeld, wanneer je enkel een geheime sleutel nodig hebt om versleutelde gegevens te ontcijferen. Het onderzoek richtte zich voornamelijk op de sleutels die gebruikt worden om de inhoud van de harde schijf te versleutelen. De onderzoekers waren in staat om Bitlocker van Windows Vista te kraken, maar ook andere vergelijkbare toepassingen, zoals FileVault van Apple en dm-crypt van



*Figuur 3: De verschillende signaalsterktes gemeten door de sensoren die een locatie omringen. Als een object beweegt, dan wijzigen de signaalsterktes.*

Linux zouden hiervoor gevoelig zijn.

Bovenstaande toepassingen vereisen dat de gebruiker een geheime sleutel moet invoeren (een wachtwoord) om bij de bestanden te komen. Dit wachtwoord wordt in het RAM-geheugen opgeslagen gedurende de sessie. Om achter de geheime sleutels te komen, moet de inhoud van het geheugen, direct na een harde herstart, weggeschreven worden naar bijvoorbeeld een externe schijf. Vervolgens worden een aantal programma's op de geheugendump losgelaten om informatie terug te halen. In het geval van Windows Bitlocker dient er gezocht te worden naar AES-sleutels in de geheugendump. Door middel van een slim algoritme, dat gebruikt maakt van de procedure om AES-sleutels te genereren en de Hamming Distance (het aantal verschillende bits tussen twee woorden van dezelfde lengte) tussen twee sleutels, is het mogelijk om zelfs beschadigde sleutels terug te halen. Vervolgens kunnen de kandidaat-sleutels gebruikt worden om de schijf te ontcijferen, omdat elke sector dezelfde sleutel gebruikt. Dit proces hebben de onderzoekers geautomatiseerd en op deze manier kunnen ze binnen enkele minuten een schijf 'kraken'.

De kans van slagen is afhankelijk van de staat van het geheugen. Het meeste succes heeft iemand die het geheugen bevriest voordat hij de computer herstart. Daarnaast is het ook afhankelijk of het BIOS van het doelwit bij het booten het geheugen (gedeeltelijk) overschrijft. Mocht dit het geval zijn, dan kan geprobeerd worden om het geheugen in een andere machine te plaatsen. In al deze gevallen dient de aanvaller fysieke toegang te hebben tot het doelwit terwijl deze aanstaat (anders is de kans erg klein dat er waardevolle data in het geheugen staat). Er is op dit moment nog

geen oplossing voor deze aanval, behalve het compleet overschrijven van je geheugen na elke computersessie. Zelfs al zou een geheime sleutel maar heel even in het geheugen staan, een geheugendump van dat moment is al voldoende. Misschien wordt het dan toch tijd voor wegwerpgeheugen dat ook maar één keer te gebruiken is?

### Draadloze signalen als radar

Het is misschien altijd wel een droom geweest om door muren heen te kunnen kijken. Daarvoor zijn dure apparaten beschikbaar, maar een recente publicatie van twee onderzoekers toont aan dat dit simpeler kan.

In het onderzoek van Joey Wilson en Neal Patwari wordt een nieuwe methode gepresenteerd die het mogelijk maakt om realtime objecten te volgen aan de andere kant van de muur, zonder dat dit object voorzien is van een systeem om deze te volgen. Ze maken hierbij gebruik van draadloze signalen en een sensornetwerk. De signaalsterkte op een punt binnen het netwerk (op de locatie) is de som van alle paden die een radiosignaal kan nemen om bij een andere sensor te komen. De sterkte van een signaal wordt beïnvloed door objecten op z'n pad. De meetresultaten van alle sensoren worden verzameld van alle sensoren. Hiervan wordt een tomo-gram gemaakt, een kaart die weergeeft waar signalen sterk en zwak zijn. De techniek wordt variance-based radio tomographic imaging genoemd omdat het gebruikmaakt van de signaalsterktes die variëren door bewegende objecten.

Volgens Wilson en Patwari zou het gebruik van draadloze signalen om door muren heen te kijken een goede toepassing kennen. Ze beschrijven een situatie waarbij reddingswerkers, het leger of de politie op een locatie aankomen die te gevaarlijk is om te betreden. Zodra een aantal sensoren rond de locatie verspreid worden, lokaliseren de sensoren zich vanzelf (met bijvoorbeeld GPS) en vormen ze een netwerk. Zo weet elke sensor waar deze zich bevindt. Vrijwel meteen meten de sensoren signaalsterktes van links die de locatie overbruggen. Deze gegevens worden uitgewisseld met een centrale node die vervolgens realtime een analyse maakt van bewegende objecten, op basis van verschuivingen van signaalsterktes.

Bovenstaande toepassing klinkt prachtig, maar brengt in de praktijk wel een risico mee. Stel dat de sensoren nog kleiner en zuiniger kunnen worden, dan kun je een bepaalde locatie gemakkelijk volgen. Privacy is dan niet meer gegarandeerd. Je zou dan in een Kooi van Faraday (dat is een signaalfrije ruimte) moeten wonen om signalen te blokkeren (of zo veel mogelijk te dwarsbomen).

### Conclusie

Gelukkig zijn bovenstaande experimenten meer een proof of concept dan werkelijkheid. Toch mogen we het risico niet onderschatten. Het af luisteren van toetsenborden werkt op beperkte afstand en is niet realtime, toch kun je 'afgeluisterd' worden zonder dat je het door hebt. Maar als de techniek het toelaat om snellere en gevoeliger hardware te produceren, die ook nog eens kleiner is, dan zou het wél realtime en op grotere afstanden kunnen. Het uitlezen van computergeheugen vereist fysieke toegang tot het apparaat terwijl deze aanstaat. Het gebruik van draadloze signalen als radar heeft een praktische toepassing in de toekomst, maar als de sensoren steeds gevoeliger worden, dan wordt het makkelijker om een locatie in de gaten te houden, omdat er minder sensoren nodig zijn.

Hoe dan ook, voor elk probleem is er wel weer een oplossing. Beter afgeschermd toetsenborden, goedkoper geheugen dat niet op basis van condensatoren werkt en stoorzenders om de radarsensoren te storen. Hoe de beschreven technieken in de toekomst gebruikt zullen worden, en hoe misbruik tegen gegaan kan worden, zal de toekomst ons leren.

### Bronnen

**Compromising Electromagnetic Emanations of Wired and Wireless Keyboards**

Vuagnoux, M. & Pasini, S.  
<http://bit.ly/6Ye6kY>

**Cold Boot Attacks on Encryption Keys**

Halderman, Schoen, Heninger, Clarkson, Paul, Calandrino, Feldman, Appelbaum & Felten  
<http://bit.ly/1WYzh7>

**Through-Wall Motion Tracking Using Variance-Based Radio Tomography Networks**

Wilson, J. & Patwari, N.  
<http://bit.ly/7NEhlp>



# Frank Gehry en het hoorcollege

Ik heb laatst eens een lijstje gemaakt van mijn favoriete activiteiten. Op de eerste plaats staat vaderschap, nummer twee is lezen. Op een gedeelde derde, vierde en vijfde plaats staan poolbiljart, Taekwondo, en iets wat je zelf kunt bedenken. En op nummer zes: hoorcollege geven! Wat is hoorcollege geven heerlijk. Je moet je goed voorbereiden, maar dan krijg je ook bijna twee uur aandacht, mensen lachen om je grappen, en als je (terecht of onterecht) het idee hebt dat je overkomt geeft dat een geweldige kick. Leve het hoorcollege!

Onlangs heb ik als verse OLD uit belangstelling een aantal hoorcolleges bezocht, en daar werd me weer eens duidelijk dat “geven is leuker dan nemen” zeker ook voor hoorcolleges geldt. De docenten waren uitstekend, de slides waren puik, en de onderwerpen boeiend, en toch: wat een opgaaf om twee keer 45 minuten passief te zijn! Hoe moet je na een dag hoorcolleges volgen nog de energie opbrengen voor zelfstudie? Is dat door mij zo geliefde hoorcollege wel zo kosher voor de student?

Een citaat uit een onderwijskundig handboek <sup>1</sup>: “Het hoger onderwijs in Nederland heeft in de laatste vijftig jaar een aantal belangrijke veranderingen doorgemaakt. Toch is het opmerkelijk, dat evengoed de stelling verdedigd kan worden, dat er eigenlijk bijzonder weinig veranderd is. Om bij het laatste te beginnen. Het didactisch model is zeker bij het wetenschappelijk onderwijs nauwelijks gewijzigd. Hoorcolleges spelen nog steeds een centrale rol.” Is het misschien tijd voor verandering?

Dat er iets moet veranderen in het onderwijs vindt in elk geval onze rector Ed Brinksma. In het UTNieuws van 26 november j.l. beargumenteert hij dat er een kwaliteitsimpuls nodig is voor het UT onderwijs. Ik vind zijn verhaal erg overtuigend, en ik vind het fantastisch

dat hij op deze manier luid aan de bel trekt. Maar dat betekent wel dat er nu echt iets moet gebeuren. We komen er niet met een beroep op ieders goede wil, met het formuleren van wat uit de lucht geplukte ambities, of met het ontwikkelen van verfijnde instrumenten om precies te kunnen meten hoe beroerd het er nu eigenlijk voor staat. De huidige situatie vraagt om radicale ingrepen, en niet om gepruts in de marge.

Kijk eens wat er gebeurd is in het Baskische stadje Bilbao. Een slaperig provinciestadje, tot Frank Gehry er langs kwam en er het gedurfde Guggenheim museum neerzette. En sindsdien staat Bilbao, zoals dat heet, op de kaart!

Onze rector lijkt me iemand met durf. Vandaar dat ik hem het volgende stappenplan voorstel:

- A) Schaf het hoorcollege af.
- B) Zoek een onderwijsarchitect van het kaliber van Frank Gehry, en laat die er iets verbluffends voor in de plaats zetten.
- C) Schrijf geschiedenis.

Ed, bedankt! En wat zal ik het missen, dat hoorcollege...

<sup>1</sup>. Vernieuwing in het hoger onderwijs. Onderwijskundig handboek. Hans van Hout, Geert ten Dam, Marcel Mirande, Cees Terlouw, Jos Willems (redactie). Uitgeverij van Gorcum, 2006. Citaat op pagina 1.



Rom Langerak

Opleidingsdirecteur  
Informatica

Sinds april 1992 is dr. ir. Rom Langerak universitair docent bij de Formal Methods and Tools groep van de faculteit EWI. Romanus (Rom) werd op 1 februari geboren in Dordrecht en ging naar het Christelijk Lyceum aldaar. Hij haalde op de Universiteit Twente met lof zijn studie Toegepaste Wiskunde, waar hij afstudeerde op een onderwerp over Databases. Het is dan ook niet vreemd dat hij na zijn afstuderen ging promoveren bij de toenmalige faculteit Informatica. Na zijn promoveren in 1992 bleef hij bij de faculteit werkzaam.

Rom houdt van literatuur, filosofie, gitaar spelen, biljarten en Taekwondo.

Sinds september 2009 is hij de nieuwe opleidingsdirecteur Informatica, een taak die hij met liefde zal gaan uitvoeren om zo het onderwijs voor zowel studenten als docenten nog verder te verbeteren.

# Data-integratie onzekerheden



Oliver  
Jundt  
AV11

DATABASES, INTEGRATIE, VERZAMELING, FEEDBACK

## Hoe gebruikers bij kunnen dragen

Heb je je ooit af gevraagd hoe jouw media-player je verzameling mp3 albums zo mooi met covers en informatie uitbreidt? Heb je enig idee hoe Google News weet dat twee artikelen uit verschillende bronnen hetzelfde onderwerp hebben ondanks verschillende titels? Weet je hoe jouw PC bij het synchroniseren met je mobieltje bepaalt dat de net verhuisde Peter dezelfde Peter is als de vorige keer?

### Data samenvoegen

Dit heeft allemaal te maken met data-integratie. Data-integratie is het verbinden van twee of meer databanken of gegevensbronnen tot één nieuwe databank. Meestal is het doel een kwalitatief betere databank te bouwen. Neem als voorbeeld Google News, waar je altijd een heel compleet nieuwsoverzicht van honderden nieuwsbronnen kunt vinden. Jouw mp3 verzameling, de contactdatabanken op de PC en mobiele telefoon worden ook kwalitatief verbeterd door er nieuwe informatie aan toe te voegen.

### Overlapping

"Mensenrechten mogen geen buit  
Gazet van Antwerpen - 54 minuten geleed  
Uitgerekend op de zestigste verjaardag van d  
minister van Buitenlandse Zaken Bernard Ko  
opmerkelijke uitspraken ontvallen. "In het bui  
mogen de mensenrechten geen buitensporig  
Britten willen 'Eén wet voor iedereen' Trouw  
Rechten van de mens NRC Handelsblad  
Algemeen Dagblad - Volkskrant - De Telegra  
alle 106 soortgelijke »

Figuur 1: 106 soortgelijke artikelen samengevat onder één link.

Het probleem bij data-integratie ontstaat als je bijvoorbeeld twee soortgelijke artikelen of twee keer dezelfde persoon op jouw PC of telefoon hebt. Dan

Soms kijken mensen ook met een verschillend persoonlijk perspectief naar iets. Dit gebeurt heel vaak als gegevens met gevoelens te doen hebben. Je buur-

## Mensen of sensoren kunnen fouten maken

is het wenselijk dat de nieuwe databank niet twee keer hetzelfde object opslaat, maar deze samenvat tot één. Het zou erg vervelend zijn als elke naam plotseling twee keer in je mobieltje staat.

Dit is echter makkelijker gezegd dan gedaan. Wat doe je als de naam of titel verschilt, maar ze toch allebei over dezelfde databank record gaan? "Boehse Onkelz", "Boehse Onkels", "Böhse Onkelz". Allemaal mogelijke namen voor de Duitse rockband "Böhse Onkelz". Sterker nog, het probleem ligt niet altijd bij het herkennen van gelijke objecten, maar ook bij het uit elkaar houden van verschillende objecten. Stel je hebt een album van "Duffy" in jouw verzameling. Als je op last.fm kijkt, zie je dat er vier verschillende artiesten zijn met deze naam. Hoe weet jouw mediaplayer nu welke Duffy bedoeld wordt?

### Oorzaken

Deze problemen heb je bijna altijd bij data-integratie. De oorzaak ligt vooral in verschillen van de kwaliteit en het formaat van gegevens. De gegevens komen vaak van mensen of sensoren, maar die kunnen fouten maken. Fouten zoals verschillende schrijfwijzen en onduidelijke of incomplete gegevens.

man zal de hemel van vandaag waarschijnlijk niet precies zo blauw vinden als jij en ook bij eten heeft iedereen een andere smaak.

### Wat kan je computer doen?

Hoe kan je databanken dan toch integreren? Bij heel kleine databanken kan je dat nog wel handmatig doen, maar voor een heel grote mp3 verzameling of Google News is dat geen optie. Daar moeten duizenden gegevensbestanden automatisch geïntegreerd worden en het liefst elke vijf minuten opnieuw.

Ten eerste moet je aan de computer vertellen hoe hij kan bepalen wanneer twee objecten gelijk zijn en wanneer niet, zoals: gelijke naam betekent dat de objecten gelijk zijn. Je geeft hem dus regels, ook "knowledge rules" genoemd. Vaak zijn deze regels een beetje tolerant, bijvoorbeeld op het gebied van schrijfwijzen. Op deze manier kan je al een grote aantal probleemgevallen oplossen.

Je kunt een heleboel tijd investeren in het aanpassen van deze regels en grenswaarden, zodat je algoritme zo min mogelijk fouten maakt bij het beslissen. Maar er zullen altijd kritische gevallen zijn waar een computer nooit zonder



problemen een keuze kan maken. Zelfs mensen hebben er soms moeite mee. Kan je met zekerheid zeggen dat “Tamal Roy Choudhury” dezelfde persoon is als “Tamal Sengupta”, als je enkel weet dat beide namen voor dezelfde filmrol op twee verschillende filmdatabank-websites zijn genoemd?

De knowledge rules mogen ook niet willekeurig tolerant zijn, want vaak leidt een verbetering bij het herkennen van bepaalde gevallen aan de ene kant tot een verslechtering aan de andere kant. Stel jouw knowledge rules zeggen dat een s een z mag zijn of dat ö ook als oe mag worden geschreven. Dan zal het nu een stuk beter werken voor “Böhse Onkelz”, maar andere namen die eerst nog te onderscheiden waren, worden nu ook (foutief!) met elkaar in verbinding gebracht.

*Als de computer onzeker is...*

Wat doe je nu met de resterende gevallen die dichtbij de tolerantiegrenzen zitten en waarvan je computer dus niet zeker kan zeggen of twee objecten gelijk zijn of niet (bijvoorbeeld 45% gelijk)?

Je kunt dit op twee manieren doen: je kunt de onzekerheid verbergen en gewoon een keuze maken, ook al verschillen de waarschijnlijkheden erg weinig; of je kunt deze onzekerheden met hun waarschijnlijkheidswaarde erbij in de gecombineerde databank opslaan.

De eerste manier leidt tot een “non-probabilistische” databank. Een voordeel van dit soort databanken is dat ze niet veel groter zijn dan de oorspronkelijke databanken die geïntegreerd worden en dat er helemaal geen gebruikerinteractie nodig is bij queries. Maar zo’n databank heeft heel waarschijnlijk ook fouten in zich, dus wees niet verrast als

jouw nieuwe album een verkeerde cover en release year krijgt.

De tweede en veel intuïtievare manier leidt tot een grotere databank, want bij kritische gevallen moet elke mogelijkheid opgeslagen worden. Nu kan je de onzekerheid aan de gebruiker doorgeven, met de hoop dat die zal weten wat wel en wat niet goed is. Jouw media-player kan bijvoorbeeld voor het album van Duffy alle vier de artiesten aangeven, gesorteerd naar waarschijnlijkheid. Op deze manier kan door de gebruiker ook eenvoudig feedback worden gegeven over het query resultaat.

*... kan de mens helpen*

User feedback is een erg nuttige methode om onzekerheden bij data-integratie op te lossen. Als een gebruiker heel zeker is dat een bepaald object gelijk is aan en ander, dan kunnen alle andere mogelijkheden worden verwijderd. Behalve positieve feedback kan hij ook negatieve feedback geven, dus zeggen dat twee objecten nooit hetzelfde kunnen zijn. In dit geval kun je deze ene mogelijkheid verwijderen. Hieruit blijkt dat positieve feedback een stuk efficiënter is dan negatieve feedback, maar allebei helpen ze om stap voor stap onzekerheden uit de databank te verwijderen. Op deze manier benadert een probabilistisch databank na een tijdje een non-probabilistisch databank en wordt gebruikersinteractie minder nodig.

Feedback heeft nog meer voordelen. Je hoeft niet meer zo veel tijd in het tunen van grenswaarden te steken. Men kan laten zien dat zelfs simpele regels volstaan om een grote deel automatisch op te lossen en het dan efficiënter is om de resterende onzekerheden aan de gebruiker te geven. Met simpele knowledge rules en user feedback kom je sneller tot

goede data-integratie dan zonder user feedback en met geavanceerde regels.

Feedback heeft natuurlijk niet alleen maar voordelen. Als je elke keer door jouw media-player gevraagd wordt welk album nu die juiste is, omdat de knowledge rules veel te stom zijn om zelf simpele feiten te onderscheiden, zal het snel vervelend worden. Ook mogen de lijsten niet te lang zijn en moeten de waarschijnlijkheden goed berekend zijn. Niemand heeft er zin in om honderd mogelijkheden door te kijken en de juiste gegevens pas aan het eind te vinden.

Hier gaan wij er ook nog eens van uit dat de gebruiker altijd weet wat het goede antwoord is, maar dat is niet altijd het geval.

*In toekomst*

In de afgelopen jaren is veel onderzoek in deze richting gedaan om data-integratie nog efficiënter te maken, en er zal nog meer onderzoek plaatsvinden. Ook in de praktijk krijgt data-integratie steeds meer aandacht, vooral bij zogenoemde Mash-Ups (zie I/O Vivat 23-3) zoals Google News. Data-integratie geeft ons interessante nieuwe mogelijkheden om met al bestaande informatie te werken. Je zult het in toekomst steeds vaker tegenkomen.

## Bronnen

Qualitative Effects of Knowledge Rules in Probabilistic Data Integration (2008)

*Maurice van Keulen, Ander de Keijzer*

User Feedback in Probabilistic XML (2007)

*Maurice van Keulen, Ander de Keijzer*



Rick  
van Galen  
Voorzitter Inter-Actief

Rick van Galen werd op 3 augustus 1989 geboren in Nijmegen, maar bracht zijn jeugd door in Huissen, vlakbij Arnhem. Toen hij zijn tweetalig vwo met N&T-profiel in het alom bekende Bemmeler af ronderde, stevende hij af op een studie aan Rijksuniversiteit Nijmegen. Toen hij gegrepen werd bij een kennismaking met de campus van de UT, heeft hij echter het avontuur gekozen en is sindsdien betrokken geweest bij Inter-Actief. Bij Inter-Actief heeft hij zich onder andere ingezet voor het symposium ecomputing. Sinds oktober 2009 is hij voorzitter van Inter-Actief.

# Van de voorzitter

## De lijn moet ergens beginnen

Het is, hopelijk, een jaarlijks fenomeen dat het zittende bestuur genoeg heeft van de bende die een jaar lang is opgebouwd in de Inter-Actief-kamer en een dag in hun agenda markeert met een grote X. Die dag gaat het voltallige bestuur aan de slag met het schoonmaken van de kamer.

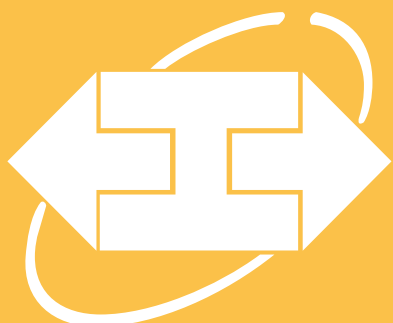
Dag X is ons net achter de rug. Natuurlijk zijn we tevreden over de netheid en uitstraling van de Inter-Actief-kamer en de gezelligheidshoek op het balkon. We hadden groot medelijden met het universiteitsschoonmaak bij wie we een schrobbeurt van de kamer hadden aangevraagd – als je een stel bedrijvige studenten een hok geeft, dan weet je niet wat je elk jaar aantreft.

Wat je als bestuur ook niet weet is de historie die in de Inter-Actief-kamer is opgenomen. Bij onze opruimings- en archiveringsdrift stuitte we op reisverslagen uit de vorige eeuw; hoe vroeger een avontuur werd ondernomen naar Scandinavië als een van de eerste studiereizen van onze verenigingen. Lachend keken we er met ons zessen naar het geringde hoopje A4'tjes, waarin met primitieve technieken als een typemachine en een fotokopieerapparaat een reisverslag in elkaar was gezet met knippen en plakken in de oude stijl. Tegenwoordig wordt er gelachen als er bij een andere vereniging een dergelijke reis wordt georganiseerd – het is 'niet eens' buiten Europa – en een dergelijk verslag wordt aangeleverd. Toch was dit indertijd voor Inter-Actief een hele prestatie en legde het het fundament voor ondermeer de studiereis Pixel die deze herfst naar Amerika zal vertrekken.

Een ander stukje historie kwam tevoorschijn uit het ViZiON-archief. Het

herinnerde aan de tijd dat informatica, telematica en bedrijfsinformatietechnologie nog hun hoogtijdagen beleefden, ten tijde van de dotcom-hype. Het was de tijd dat breezah-taal nog hip was (de titel van deze voorloper van de noizia getuigt daarvan), iedereen naar Britney Spears luisterde en terrorisme niet aan de orde van de dag was. Toch laat de ViZiON een beeld zien van Inter-Actief dat, afgezien van wat opmerkelijke kledingkeuzes, veel meer lijkt op wat er nu binnen onze vereniging gebeurt. Hoewel Inter-Actief qua leden niet meer zo'n grote vereniging als toen is (de instroom is helaas gedaald van zo'n 200 per jaar naar zo'n 80 per jaar), is indertijd de basis gelegd van veel zaken die we nu voor lief nemen binnen onze vereniging. Een jaarlijks dagvullend symposium gecombineerd met een jaarlijkse studiereis zijn niet altijd gemeengoed geweest.

Dit alles brengt mij terug bij het verhaal van de stijgende lijn, dat ik in mijn vorige column als kandidaat-voorzitter uiteen mocht zetten. Elk jaar bij Inter-Actief doe je je best om het een beetje mooier te maken als dat het voor jou was. Op welk moment je dit op de stijgende lijn mee mag maken, is eigenlijk irrelevant, want het is altijd spannend. Inmiddels zijn we bijna aan ons zesde lustrum toe – in 2011 bestaat Inter-Actief 30 jaar – en worden we misschien wel volwassen. Laten we hopen dat er in de toekomst bij een opruiming nog stukken uit ons jaar gevonden kunnen worden, en dat er dan weer iemand zegt "Kijk, zij hebben het voor ons opgezet". Het is een goed voornemen om dat dit jaar te bewerkstelligen. Tijd maar weer om de bezem weer in de kast te zetten en aan de slag te gaan.



Inter-Actief

# ENIAC: Van de penningmeester



Francis  
Henninger

*Penningmeester en  
secretaris alumnivereni-  
ging ENIAC*

Afgestudeerd 30 november 2005,  
Software Engineering.

Voor slechts € 5,- per jaar kan je al lid worden van ENIAC. Je krijgt dan in ieder geval de Vivats die jaarlijks verschijnen (meestal zo'n 4 stuks, maar niet helemaal per kwartaal) en uitnodigingen voor de activiteiten die we organiseren (meestal per mail). Daar mag je dan vervolgens (veelal gratis!) aan deelnemen. En al doe je maar eens in de paar jaar ergens aan mee, die € 5,- kan toch bijna iedere informatica-alumnus wel missen? Zo houd je toch nog wat binding met je wetenschappelijke roots en af en toe contact met vrienden uit je studietijd.

Dit keer geen schrijven van de voorzitter, maar eentje van de penningmeester of van de secretaris. Welke het is hangt af van welke pet ik op heb, hoe dat zit lees je wat verderop. Dit is de eerste keer dat ik een artikel voor de IO-Vivat schrijf, en hopelijk niet de laatste keer. Het stuk zal gaan over het reilen en zeilen achter de schermen van ENIAC, ofwel het gene wat ik doe voor de vereniging.

Voordat ik daarmee begin, begin ik eerst met hoe ik met ENIAC in aanraking kwam en hoe ik in het bestuur ben gekomen. Mijn eerste aanraking was via de aanmeldingsbrief die ik de afstudeertas van IA vond. Eén van de belangrijkste redenen om lid te worden was voor mij om een binding met de universiteit en faculteit te onderhouden. Het alumnibeleid van de UT was opdat moment nog niet zo actief en herkenbaar zoals het nu is. Ik zou nu nog steeds lid van ENIAC zijn geworden, omdat het alumnibeleid van UT gericht is op de hele universiteit en niet specifiek een faculteit of opleiding.

Na ongeveer een jaar werd gevraagd of ik interesse had in een bestuursfunctie binnen ENIAC. Op dat moment had ik niet direct een antwoord, aangezien ik vind dat een dergelijke vraag weloverwogen beantwoord dient te worden. Na een kleine bedenktijd had ik besloten om het aanbod te aanvaarden. Na een jaar in het bestuur te hebben gezeten was het bestuur gehalveerd: twee van de vier bestuursleden konden door omstandigheden onvoldoende tijd vrijmaken voor het vervullen van de functies. In dat geval kan je als resterend bestuur twee dingen doen: de vereniging opheffen of roeien met de riemen die je hebt. Gelukkig is het de laatste optie geworden, anders zou er geen jaarboek of scriptieprijsuitreiking zijn geweest in de afgelopen jaren.

Terug naar de realiteit van vandaag de dag: wat doe ik als penningmeester dan wel secretaris binnen de vereniging. In feite komt het erop neer dat ik verantwoordelijk ben voor de ledenadministratie en de boekhouding, waarbij ik moet opmerken dat een groot deel van de ledenadministratie is uitbesteed aan het AlumniBureau. Aan beide taken ben ik niet al te veel tijd kwijt. De meeste tijd gaat zitten in het versturen van de uitnodigingen van de algemene ledenvergadering en het bijhouden van de geïnde contributie voor ruim 800 leden.

Het andere deel van het bestuurslid-zijn is de vertegenwoordiging van de vereniging. Dit bestaat vooral uit het aanwezig zijn op, de momenteel spaarzame, activiteiten van ENIAC. Daarmee kom ik ook op een punt dat niet goed belicht is. ENIAC is voor en door leden, wat wilt zeggen dat je als bestuur niet alles hoeft en zou moeten doen. Inbreng van de leden is van zeer groot belang, zoals opbouwende kritiek of suggesties. De voorkeur gaat echter uit naar daadwerkelijke inzet zoals het organiseren van een activiteit, jaarboek, of het vervullen van een bestuursfunctie.

Uit eindelijk blijft het vrijwilligerswerk en kan er niemand gedwongen worden om iets te doen. Voorlopig blijf ik me inzetten voor ENIAC, omdat ik het leuk vind om te doen en omdat er, gelukkig, nog steeds leden zijn die het waarderen. Dit laatste kwam bij de afgelopen scriptieprijsuitreiking duidelijk naar voren, en dan krijg ik toch wel een warm gevoel van binnen.

# Master in de VS



MASTER, AMERIKA, LANGUAGE TECHNOLOGIES, PITTSBURGH, CMU

## Hard werken maar een hele leuke ervaring

**A**l tijdens mijn studie in Enschede liep ik met het idee rond om een periode in het buitenland te studeren. Amerika trok mij aan, omdat het land erg invloedrijk is in de rest van de wereld en ook goede universiteiten op het gebied van informatica heeft. Uiteindelijk besloot ik na mijn bacheloropleiding de gok te wagen. Een jaar voordat ik er daadwerkelijk heen ging ben ik begonnen met

uitslag bleek dat ik overal terecht kon. Vanwege mijn interesse in taaltechnologie heb ik voor Carnegie Mellon University (CMU) gekozen en dat bevalt erg goed!

Nadat ik was toegelaten, is het nog lange tijd onzeker gebleven of ik daadwerkelijk naar Amerika kon. Ondanks dat ik een beurs zou ontvangen, kwam ik namelijk nog veel geld te kort. Mijn programma kost per jaar alleen aan col-

mijn verblijf kan bekostigen.

Aan CMU volg ik de masteropleiding 'language technologies', wat binnen de faculteit Informatica valt. De opleiding past informatica toe om menselijke taal te analyseren. Je kunt hierbij denken aan information retrieval, vertalen van taal, analyseren van informatie op het web om trends te ontdekken en nog veel meer. Afgelopen semester had ik de vakken 'machine learning' en 'algorithms in natural language processing' en volgend semester ga ik de vakken 'information retrieval' en 'analysis of social media' volgen. Erg interessant dus! Mijn programma bestaat voor de helft uit vakken en voor de andere helft uit het doen van onderzoek. Voor mijn onderzoek werk ik met statistische modellen voor het modelleren van tekst. Ik werk aan twee opdrachten. De ene draait om het visualiseren van informatie; de andere om het modelleren van personen/groepen in een documentencollectie (bijvoorbeeld personen in e-maildata of groepen gebruikers in forums).

## Bedrijven als Microsoft, Google en Yahoo komen regelmatig langs

alle voorbereidingen. Zo moet je allerlei toetsen afleggen (waar je ook nog echt wel voor moet studeren), een motivatiebrief schrijven en aanbevelingsbrieven regelen. In december 2008 was het hele pakket net op tijd klaar. Om mijn kansen te spreiden heb ik mij voor meerdere universiteiten aangemeld. Na enkele maanden in spanning wachten op de

legegeld al 35.000 dollar. Gelukkig is het gelukt om bij CMU een Research Assistantship te bemachtigen. Dit houdt in dat ik in de tijd dat ik onderzoek moet doen voor mijn studieprogramma, ik dit doe in het kader van een gefinancierd project. Ik hoef hierdoor geen collegegeld te betalen, en ik ontvang zelfs nog een maandelijks bedrag, zodat ik

De studie is ontzettend interessant, maar ook ontzettend hard werken! De vakken zijn hier veel zwaarder dan in Nederland, maar gelukkig wel erg interessant en ik leer heel veel. Omdat er een strenge selectie aan vooraf gaat, zijn de studenten goed en gemotiveerd. Je cijfer hangt af van hoe de andere studenten het doen, en dat maakt de prestatiedruk hoog. Verder is het best schools. Ik heb bijvoorbeeld regelmatig huiswerk. Dat dwingt om constant bij te blijven, maar ik ben ook intensiever met de stof bezig dan in Nederland. Ook is het contact met de professoren informeel, ze zijn goed benaderbaar.





Als student heb je hier een office (werk-kamer), en aangezien de universiteit dag en nacht open is, zitten velen tot 's avonds laat op de universiteit. Ik kwam er al snel achter dat de bank in mijn office vooral als slaapplek wordt gebruikt: mijn office-genoten blijven hier regel-

wel ik er van te voren niet veel van wist, blijkt het een boeiende stad te zijn. Ik woon in een studentenbuurt tussen twee grote parken. Pittsburghers zijn enorme sportfans en om wat van het Amerikaanse leven te ervaren, ben ik naar een baseball- en een football-wed-

## De vakken zijn hier veel zwaarder dan in Nederland

matig een nachtje slapen, nadat ze tot laat hebben doorgewerkt!

Een groot voordeel van het studeren aan een goede universiteit is het feit dat er elke dag interessante praatjes zijn. Zo is er vrijwel elke dag wel een lunchlezing op het gebied van informatica. Er komen regelmatig wetenschappers van over de hele wereld op bezoek en ook grote bedrijven zoals Google, Microsoft en Yahoo komen regelmatig langs.

Ik studeer nu samen met mensen met allerlei verschillende achtergronden. Van de nog geen twintig eerstejaars van mijn opleiding zijn er drie Amerikanen, de rest komt uit India, China, Japan, maar ook Canada, Portugal en andere landen. Het grote percentage buitenlandse studenten maakt CMU een hele bijzondere, dynamische omgeving.

CMU is gevestigd in Pittsburgh. Hoe-

strijd geweest. Met die sporten heb ik niets, maar het was leuk om te midden van allemaal enthousiaste Pittsburghers te zitten. Verder maak ik de Amerikaanse feesten mee. Halloween wordt hier groots gevierd, maar het bijzonderste tot nu toe vond ik Thanksgiving. Ik heb dat in Washington gevierd met echte Amerikanen. We hadden een traditioneel diner, inclusief kalkoen, pumpkin pie en andere lekkernijen.

Kortom, het is een hele leerzame en bijzondere ervaring om in Amerika te studeren. Ik raad het iedereen aan om tijdens je studie een periode in het buitenland door te brengen. Mocht je interesse hebben om te studeren in Amerika, neem dan gerust contact met me op.

# Spraakherkenning in praktijk



Laurens  
Satink  
ENIAC

TEKSTEN, CONTEXT-AWARENESS,  
SPRAAKTECHNOLOGIE, STATISTI-  
SCHE ANALYSE

## Academische waarden in de praktijk

In 1999 ben ik begonnen met Technische Informatica te studeren in Enschede. Het begin van mijn studie voorliep erg voorspoedig, en ik studeerde met veel plezier. Het niveau van het onderwijs sprak mij erg aan, en door een wiskundeleeraar op de middelbare school met zijn

ten tussen kennis hebben en het kunnen toepassen. Het identificeren van het eigenlijke probleem, en het zelf verkrijgen van de noodzakelijke randvoorwaarden is een competentie die niet direct uit de boeken tevoorschijn kwam. Dit in een praktijkgerichte situatie leren heb ik heel waardevol gevonden.

Spraaktechnologie wordt in verschillende gebieden ingezet, waaronder het telefoniedomein. Hierbij valt te denken aan klantidentificatie (“spreek na de piep uw postcode in”) maar ook aan open vraag classificatie: “Hallo met ... waarmee kunnen we u van dienst zijn? Spreek na de piep uw vraag kort en duidelijk in”.

## Spreek na de piep uw vraag kort en duidelijk in...

eigen opvattingen over wat er benodigd was, had ik een kleine voorsprong ten opzichte van medestudenten op dat gebied. Het leren van en over de aspecten van informatica die in de eerste 2 jaar behandeld werd ging me goed af, maar ik had het gevoel dat ik meer wilde.

De gelegenheid deed zich voor om een bijbaantje op mijn vakgebied te krijgen bij een klein software bedrijfje. Dit was Dijkoraad IT (tegenwoordig SG Facilitator), met destijds ongeveer 10 medewerkers. Het is een bedrijf dat zich richtte op het ontwikkelen van facilitaire software voor grote organisaties en bedrijven, en daarnaast projectgebaseerd een diversiteit aan opdrachten had. In eerste instantie ben ik aangenomen voor wat bugfixes en kleine klusjes waar de rest van de mensen niet aan toe kwam. De eerste opdracht die ik kreeg vond ik destijds een behoorlijk moeilijke opdracht. Een hoofd met twee jaar aan academische kennis, en een praktische opdracht waar een oplossing voor moest komen, hoe moeilijk kon dat zijn. Dat was het moment waarop ik voor het eerst merkte dat er een verschil kan zit-

ten tussen kennis hebben en het kunnen toepassen. Door de jaren erna groeide mijn werk bij Dijkoraad IT uit tot een situatie waar ik, weliswaar maar voor één dag in de week, mee draaide in een aantal projecten en mee ontwikkelde heb aan nieuwe ontwikkelingsgebieden. Hiervoor was ook meer inhoudelijke kennis nodig dan alleen software engineering technieken, en het toepassen van de opgedane kennis heb ik als ontzettend leuk en uitdagend ervaren.

Al voor mijn afstuderen ben ik begonnen met werken bij Telecats; daar zit ik nu iets meer dan een jaar. Het is een relatief klein bedrijf met iets minder dan 30 werknemers. Eén van de toepassingsgebieden van Telecats is spraaktechnologie; ik ben dan ook als in dienst als software engineer en spraak- en taaltechnoloog. Het is een ontzettend leuke en uitdagende baan, omdat de praktijk en theorie elkaar raken. Het is meer dan alleen onderzoek doen naar de kwaliteit van spraakherkenning, en hoe die verbeterd kan worden, en meer dan alleen software ontwerpen (en deels implementeren).

De herkenning van een postcode is het herkennen binnen een eindig aantal alternatieven. Een postcode is opgebouwd uit vier cijfers en twee letters, en de mogelijke uitspraken en combinaties van uitspraken zijn te benoemen. Het herkennen van een duidelijke ingesproken postcode gaat in meer dan 95% van de gevallen goed. De toepassingsgebieden hiervan zijn met name in het telefonie domein erg handig, waneer er een script gevolgd wordt om te achterhalen waar in de organisatie een klant het best geholpen kan worden.

Het herkennen van (gesproken) vrije tekst is een stuk lastiger. De mogelijke alternatieven zijn hier te talrijk om te benoemen; er worden taalmodellen gebruikt. Deze taalmodellen zijn statistisch van aard; ze beschrijven de waarschijnlijkheid dat bepaalde volgordes van woorden voorkomen.

De herkenningresultaten van conversatiespraak zijn anno 2010 nog steeds vrijwel nooit goed genoeg om zonder meer mee te lezen. Om dat te kunnen moet eigenlijk rond de 95% van de woorden goed herkend worden; en zeker in een ongestructureerd gesprek via een telefoon is het bijzonder zeldzaam dat dit soort scores gehaald worden. Vaak is het echter ook niet nodig om de tekst woordelijk mee te kunnen lezen.





De herkenningresultaten zijn wel goed genoeg om als invoer voor andere technologie te dienen, zoals het uitvoeren van een classificatie of het doerzoekbaar maken van de audio.

Kernwoorden worden relatief beter herkend dan vullingswoorden, wat het geschikt maakt om als query te beschouwen tegen een verzameling documenten. Een toepassing hiervan is onder andere het classificeren van een open vraag in een call centre; waarbij het classificatie resultaat verder gefinetuned kan worden met business-logica als “deze klant heeft product X”.

Ook zijn de herkenningresultaten geschikt om een index op te bouwen, die met tekst doorzoekbaar gemaakt kan worden. Een koppeling met de tijden in de audio zorgt ervoor dat het juist stuk afgeluisterd kan worden.

Telecats is bij een aantal projecten van STEVIN betrokken geweest; ikzelf bij twee ervan. Het doel van STEVIN is de kennis vanuit het academische leven en de klantmarkten vanuit het bedrijfsleven naar elkaar toe te laten komen. Het merendeel van de projecten wordt uitgevoerd door consortia van bedrijven en universiteiten.

Het eerste STEVIN project waar ik bij betrokken raakte is het toepassen van spraaktechnologie in rechtszalen. Zittingen kunnen worden opgenomen, waarna er (offline) spraakherkenning op de individuele opnames van personen uitgevoerd kan worden. De griffier annotteert wie wanneer aan het woord is, waarmee het gebruik van sprekerspecifieke modellen mogelijk is. De zit-

tingen worden dan via de herkenningresultaten automatisch doorzoekbaar gemaakt, waarna de relevante stukken afgespeeld kunnen worden. Er is een demonstratiefilmpje op Youtube aanwezig voor de geïnteresseerden, met de zoekwoorden “Rechtbank Spraakherkenning” is het de 1e hit.

Het tweede STEVIN project waar ik onderdeel van uitmaakte is Nederlandse ONdertiteling. De Vlaamse en Nederlandse omroepen moeten de meerderheid van hun Nederlandstalige uitzendingen voorzien van ondertitels, iets dat een ontzettend arbeidsintensief proces is. Binnen dit project werd gekeken in hoeverre taal en spraaktechnologie ingezet kan worden om dit proces te bespoedigen. Er werd een consortium opgericht waarin de universiteiten van Gent (signaalanalyse voor detectie van spraak en sprekerclustering), Leuven (gebruik van de spraakherkenner SPRAAK voor het oplijnen; voortgekomen uit een eerder STEVIN project) en Antwerpen (indikken van de tekst) deel van uitmaakten. De eerste ervaringen die met het koppelen van de technologieën opgedaan zijn, zijn erg veelbelovend wanneer de ondertitels van te voren beschikbaar zijn.

Het uitdagende van bovenstaande projecten is dat er enerzijds een degelijke hoeveelheid academische kennis en competenties benodigd is om een project inhoudelijk vorm te kunnen geven, maar dat anderzijds ook het vertalen van die kennis naar een product (of prototype) nodig is. Het levert een unieke en erg leuke project samenstelling op, met deelnemers vanuit het bedrijfsleven die erop gericht zijn om hun processen

te verbeteren, en deelnemers die hun kennis en technologie willen delen en toetsen aan de praktijk.

Naast bovenstaande projecten is bij vrijwel elk project op spraakgebied expertise vereist van spraaktechnologie, alleen de mate waarin wisselt. Er is een serie projecten waarbij spraakherkenning, met een serie tools om modellen te construeren, “off-the-shelf” gebruikt wordt, en er zijn projecten waarbij meer komt kijken.

Het werk bij Telecats is precies wat ik gedurende mijn studie voor ogen had: het is een relatief klein bedrijf waarbinnen je makkelijk(er) je eigen invulling kunt geven aan projecten. De vrijheid om, uiteraard binnen redelijke grenzen, zelfstandig te werken vind ik ongelooflijk waardevol. Daarnaast vind ik het erg leuk om continu uitgedaagd te worden op inhoudelijk vlak, iets dat regelmatig gebeurt.

Terugkijkend op mijn studietijd realiseer ik me pas in hoeverre de competenties die je opdoet van belang zijn. De feitelijke kennis is zeker een belangrijk hulpstuk, en zorgt dat je de taal van informatica kunt praten en begrijpen. Maar het zijn de werkwijzes, manieren van problemen analyseren en kritische houding ten opzichte van de uitkomsten en methodes zorgen er voor dat je in staat bent complexe problemen die je in het dagelijks bedrijfsleven tegenkomt het hoofd kunt bieden.

# Tutorial: Android deel 1



Jelte  
Zeilstra  
Redacteur I/O Vivat

ANDROID, TWITTER, JAVA, ECLIPSE,  
SDK, TUTORIAL

## Bouw je eigen Twitter-applicatie in twee delen

Android is het besturingssysteem voor mobile telefoons dat door Google en de Open Handset Alliance ontwikkeld is. Het wordt gebruikt door telefoons van verschillende merken, onder andere HTC (Magic en Hero) en de Motorola Droid. Het besturingssysteem is gebaseerd op de Linux-kernel en de meeste applicaties worden ontwikkeld in Java. De broncode voor Android is vrijgegeven onder de Apache 2.0-licentie, en is dus beschikbaar voor iedereen.

Zoals gezegd gaat het ontwikkelen van applicaties met Java. Hoewel de taal hetzelfde is als voor normale computers, zijn niet alle libraries van Java SE of ME

grote aantal al bestaande applicaties voor Twitter.

De twee belangrijkste functies van onze applicatie worden het bekijken van de tweets van een gebruiker, en het posten van een tweet. Dat laatste impliceert ook dat we een inlogscherf moeten maken omdat je een gebruikersnaam en wachtwoord nodig hebt om te twitteren. Aan het einde van deze tutorial heb je dan een applicatie die je op je Android-telefoon kunt gebruiken. Ook kun je dan extra functies toevoegen, zoals het zien van de tweets van de mensen die je volgt en het sturen van directe berichten.

waar je deze SDK uitpakt.

Hoewel het nu al mogelijk is om applicaties te ontwerpen, is het makkelijker om Eclipse te gebruiken, omdat dan alle compileerstappen automatisch gaan. Eclipse is te downloaden van <http://eclipse.org/>. (Ik heb Eclipse 3.5 IDE for Java EE gebruikt.) Voor Eclipse is een plug-in ontwikkeld, de Android Development Tools (ADT). Deze plug-in voegt de Android-ondersteuning toe aan Eclipse. De installatie-instructies staan op <http://developer.android.com/sdk/eclipse-adt.html>. Er is ook een plug-in geschreven voor NetBeans, deze is te vinden op <http://kenai.com/projects/nbandroid>.

## Twitter heeft namelijk een uitgebreide API

beschikbaar. Daarentegen heeft Google een eigen library geschreven voor Android. Een voordeel hiervan is dat via deze API de specifieke functies van de mobiele telefoon zijn aan te spreken, zoals de GPS-ontvanger. Ook maakt Android geen gebruik van AWT en Swing, de standaard grafische functies van Java SE. Desondanks is het vrij eenvoudig om je eigen applicatie voor dit besturingssysteem te ontwikkelen.


In deze tutorial gaan we in twee delen een applicatie maken die tweets (berichten) leest van Twitter en ook tweets kan posten. Twitter heeft namelijk een uitgebreide API, waarmee je als softwareontwikkelaar in staat bent een programma te maken dat communiceert met de site. Dit verklaart ook deels het

### Stap 1: Voorbereiding

Voor dat we kunnen programmeren voor de Android moeten we nog een aantal tools installeren. Allereerst heb je de JRE (Java Runtime Environment, meestal wel geïnstalleerd) en de JDK (Java Development Kit), beide versie 5 of 6, nodig. Die zijn te downloaden van <http://java.com> en <http://java.sun.com/javase/downloads/>.

Het belangrijkste is de Android SDK (Software Development Kit), die is te downloaden van <http://developer.android.com/sdk/>. Deze SDK is beschikbaar voor Windows, Mac OS X en Linux en bevat ook een emulator, zodat je zelfs applicaties kunt testen zonder een telefoon te hebben. Onthoudt de locatie

Nu moet de ADT nog weten waar die de Android SDK kan vinden. Dit kan worden ingesteld via het menu Window > Preferences op de pagina Android. Kies bij SDK Location de installatiemap van de SDK.

Ga vervolgens naar Window > Android SDK and AVD Manager of klik op het -icoon. Installeer bij Available Packages op z'n minst een van de SDK Platforms, maar het liefst allemaal zodat je de applicatie met meerdere versies kan testen. Vervolgens kan je op de pagina Virtual Devices een virtuele Android-telefoon (AVD, Android Virtual Device) aanmaken. Bedenk een naam en kies bij Target een van de versies. Kies als SD Card size bijvoorbeeld 1024 MB. Klik vervolgens op Create AVD.

Klik vervolgens op Start en daarna op Launch om de AVD te starten. Het opstarten kan enige tijd duren. Je AVD is nu gereed om te gebruiken en je kunt al



Figuur 1: Het eindresultaat

de standaardapplicaties gebruiken.

Als laatste moeten we nog een nieuw project aanmaken. Ga hiervoor naar File > New > Android Project. Staat die er niet bij, kies dan Other en selecteer dan Android > Android Project. Kies een projectnaam en een applicatienaam. Neem als Build Target de laagste versie, zodat je applicatie op zoveel mogelijk telefoons draait. Telefoons met een nieuwere versie van Android kunnen namelijk wel applicaties draaien die geschreven zijn voor oudere versies, maar niet andersom. Neem als package bijvoorbeeld ia.myapp en schakel het vinkje bij Create Activity uit. Klik dan op Finish.

Je bent nu helemaal klaar om de eerste applicatie te maken!

### Stap 2 Het begin

Het project dat is aangemaakt bevat een aantal mappen. In de map src komen alle zelfgemaakte Java-bestanden te staan. De map assets is voor binaire bestanden en gebruiken we niet. In res komen de resources te staan, dat zijn bijvoorbeeld iconen, interfacelay-outs en het bestand met de gebruikte teksten in de applicatie (values/strings.xml). In gen staat een automatisch gegenereerd bestand, R.java, dat constanten

bevat om te verwijzen naar die resources, bijvoorbeeld R.string.app\_name, verwijst naar de string app\_name in values/strings.xml.

Een Android-applicatie bestaat uit verschillende componenten: Activities, Services, Broadcast receivers en Content providers. Wij richten ons in deze tutorial alleen op Activities, een Activity is een individueel venster op het beeldscherm van de telefoon.

Klap de src-map uit en klik met rechts op de nu nog lege package. Kies New > Class. Het dialoogvenster “New Java Class” is nu geopend. Kies als naam Main en als superclass android.app.Activity, en klik op Finish. Ga in het menu vervolgens naar Source > Override/Implement Methods, selecteer de methode onCreate(Bundle)

onder Activity en klik op OK. De methode onCreate(Bundle) wordt aangeroepen zodra de Activity wordt aangemaakt, tijdens het starten van de applicatie. Laat de aanroep naar de super-methode staan en typ eronder:

```
setContentView(R.layout.  
main);
```

Hiermee koppel je de lay-outdefinitie in res/layout/main.xml aan deze activity. Deze toont bij het uitvoeren de tekst “Hello World!” op het scherm.

```
Button showTweets = (Button) findViewById(R.id.main_  
showTweets);  
showTweets.setOnClickListener(new View.  
OnClickListener() {  
    public void onClick(View arg0) {  
        // TODO Komt nog  
    }  
});
```

Nu moet alleen de Activity nog toegevoegd worden aan het bestand AndroidManifest.xml, zodat Android weet welke klasse hij moet opstarten. Open het bestand AndroidManifest.xml en ga (onderin) naar het tabblad Application. Klik op Add en kies Activity. Selecteer bij Name de klasse Main en kies bij label de string app\_name. Voeg vervolgens op dezelfde manier een Intent Filter toe

en voeg daaraan de Action android.intent.action.MAIN en de Category android.intent.category.LAUNCHER toe. Dit geeft aan dat deze Activity gestart moet worden bij het starten van de applicatie. Op het tabblad AndroidManifest.xml kun je zien dat het xml-bestand is aangevuld.

Sla de bestanden op en klik met de rechtermuisknop op de projectmap en kies Run As > Android Application. De virtuele telefoon wordt nu gestart met onze pas geschreven applicatie.

### Stap 3 De eerste knop

Bij deze stap gaan we een knop maken, die een nieuw venster opent waarin we de tweets van een bepaalde gebruiker kunnen bekijken.

We gaan nu een knop toevoegen aan res/layout/main.xml die straks een nieuwe activity start. Open het bestand, klik met de rechtermuisknop op de tekst “Hello World!” en kies Remove. Het lijkt alsof er niets meer over is, maar in de outline (rechts) zie je nog de LinearLayout. Dit is de Android-equivalent van de FlowLayout. Selecteer nu onder Views de Button en sleep die in het zwarte vlak. Verander de id in het Properties-vak in “@+id/main\_showTweets”. Hierdoor wordt automatisch de constante R.id.main\_showTweets aangemaakt om naar de button te verwijzen. Verander Text naar “Show tweets”.

Ga nu naar Main.java en voeg aan de methode onCreate() de onderstaande code toe:

Met de sneltoets Ctrl+Shift+O worden de nodige import-statements toegevoegd. Op de plaats van de TODO komt later de code om een nieuwe activity (venster) te starten. Deze code wordt uitgevoerd wanneer er op de knop met id main\_showTweets geklikt wordt.

Voor het nieuwe venster gaan we eerste een nieuw lay-outbestand maken. We maken in de map res/layout een nieuw

bestand van het type Android XML file aan. Noem dit bestand showtweets.xml van het type Layout. We laten deze lay-out eerst leeg. Daarna gaan we een nieuwe klasse toevoegen, genaamd ShowTweets met als superclass android.app.Activity. Voeg hier ook weer de methode onCreate(Bundle) toe. Koppel de nieuwe lay-out met de nieuwe klasse met de methode setContentView(R.layout.showtweets);

Ook deze activity moet nog worden toegevoegd aan AndroidManifest.xml, maar nu zonder IntentFilter etc., alleen de Name is belangrijk.

In de onClick()-methode in Main.java kan nu de code komen om de nieuwe activity te starten. Dit gebeurt met de volgende code:

```
Intent i = new
Intent(Main.this, ShowT-
weets.class);
startActivity(i);
```

Eerst maken we een nieuw Intent-object aan met daarin de context (huidige activity) en de activity om te starten. Vervolgens starten we die activity ook daadwerkelijk met de methode startActivity().

Als je nu de code test moet er een zwart scherm verschijnen na het indrukken van de knop. Door op de terugknop te drukken keer je terug naar het eerste venster.

#### Stap 4 Berichten tonen

Als eerste gaan we de lay-out res/layout/showtweets.xml verder invullen. Bovenin komt een invulveld met een Go-knop om een gebruikersnaam in te voeren en daaronder komt de lijst met tweets van de gebruiker te staan. Om te zorgen dat het invulveld en de knop op een regel staan, moeten we een extra LinearLayout toevoegen in de bestaande LinearLayout. De buitenste lay-out wordt vertikaal, en de binnen horizontaal.

Zet eerst de eigenschap Orientation van de al bestaande LinearLayout op vertical. Voeg daarna de nieuwe LinearLayout toe. Daarna moet aan de binnenste lay-out een EditText-object toegevoegd worden. Dit wil niet door te slepen, maar kan door met de rechtermuis-knop op de binnenste LinearLayout in de outline te klikken en dan Add te

kieszen. Voeg vervolgens ook aan de binnenste lay-out een Button toe, en aan de

```
go.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        String uname = username.getText().toString();
        if (uname.length() != 0) { loadTweets(uname); }
    }
});
```

buitenste lay-out een ListView. Noem de objecten showtweets\_username, showtweets\_go en showtweets\_list (met @+id/ ervoor). De Text van de EditText kun je leegmaken, en de Text van de Button op "Go" zetten.

Door de eigenschap Layout width van de binnenste LinearLayout op fill\_parent te zetten, en de Layout weight van de EditText op 1, zorg je ervoor dat de breedte wordt opgevuld.

Dan nu de code. Maak in ShowTweets.java drie private variabelen aan:

```
private EditText username;
private Button go;
private ListView list;
```

In de methode onCreate() slaan we dan de object in de variabelen op:

```
username = (EditText) findViewById(R.id.showtweets_
username);
go = (Button) findViewById(R.id.showtweets_go);
list = (ListView) findViewById (R.
id.showtweets list);
```

Is er op de Go-knop gedrukt wordt moet de (nog aan te maken) methode loadTweets() aangeroepen worden, maar alleen als er een gebruikersnaam

```
public class HTTPHelper {
    public static HttpResponse doGet(String url) throws
        IOException {
        HttpClient httpclient = new DefaultHttpClient();
        HttpGet httpget = new HttpGet(url);
        System.out.println("HTTP GET: " + url);
        return httpclient.execute(httpget);
    }
    public static String getResponseContent(HttpResponse
        response) throws IOException {
        InputStream content = response.getEntity().
            getContent();
        StringBuilder res = new StringBuilder();
        BufferedReader in = new BufferedReader(new
            InputStreamReader(content));
        String buf;
        while ((buf = in.readLine()) != null) {
            res.append(buf);
            res.append("\n");
        }
        return res.toString();
    }
}
```

is ingevoerd. De code voor deze controle is als volgt:

Voordat we de methode loadTweets() maken, komt eerst nog een helperklasse. Deze klasse bevat twee methoden om het uitvoeren van HTTP-requests, oftewel communiceren met de Twitter-API, gemakkelijker te maken. Deze HTTP-request code staat onderaan deze pagina.

De methode doGet() voert een HTTP GET-request uit, en getResponseContent() leest de teruggekegen content uit. Deze methodes worden gebruikt in loadTweets().

De methode loadTweets() staat op de volgende pagina.

Omdat we nu ook verbinding met het internet maken, heeft onze applicatie de INTERNET-permissie nodig. In An-

droidManifest.xml kan bij het tabblad Permissions de permissie toegevoegd worden door Add > Uses Permission te kiezen en dan android.permission.IN-

```

private void loadTweets(String username) {
// Datastructuur gebruikt door de ListAdapter verderop
// Voor iedere tweet wordt een map aangemaakt met
// daarin:
// - text -> tekst van de tweet
// - created_at -> datum en tijd
// Deze maps worden in de lijst geplaatst

List<Map<String, String>> data = new
    ArrayList<Map<String, String>>(20);

try {
// API URL
String url = "http://twitter.com/statuses/user_
    timeline.json?screen_name="+ username;
// Voer de request uit
HttpResponse response = HTTPHelper.doGet(url);
String content = HTTPHelper.
    getResponseContent(response);
// De teruggekregen content is in JSON-formaat.
// Parse deze content.
JSONArray array = new JSONArray(content);
// Voeg de tweets toe aan data.
for (int i = 0; i < array.length(); i++) {
    JSONObject tweet = array.getJSONObject(i);
    Map<String, String> map = new HashMap<String,
        String>(2);
    map.put("text", tweet.getString("text"));
    map.put("created_at", tweet.getString("created_
        at"));
    data.add(map);
}
} catch (IOException e) {
// Gooi de fout door als er iets mis is.
// Effect is dat het programma wordt afgesloten.
    throw new Error(e);
} catch (JSONException e) {
    throw new Error(e);
}
// Maak een nieuwe ListAdaper
ListAdapter adapter = new SimpleAdapter(this, data,
// Gebruik standaard listitem met twee velden
    android.R.layout.simple_list_item_2,
// Koppel de velden text en created_at...
    new String[] { "text", "created_at" },
// ... aan het eerste en tweede veld
    new int[] { android.R.id.text1, android.R.id.text2
});
// Koppel de adapter aan de lijst
list.setAdapter(adapter);
}

```

TERNET te selecteren.

Met deze methode is het eerste deel van de applicatie klaar. Het is dus nu mogelijk om van willekeurige gebruikers de tweets op te halen.

Volgende keer

In de volgende I/O Vivat het tweede deel van deze tutorial. Dan wordt het maken van een loginscherm en het posten van een twitter-bericht behandeld.

## Op je eigen telefoon

Eclipse genereert automatisch een apk-bestand in de bin-map van je project. Door deze te downloaden naar je telefoon kan je de applicatie installeren op je mobiele telefoon.

Een makkelijkere alternatief is om het programma Linda File Manager te installeren (uit de Android market), de applicatie op je sd-kaart te zetten en vervolgens via de filemanager te installeren.

Een vereiste is wel dat onder Instellingen > Toepassingsinstellingen de optie Onbekende Bronnen op 'aan' staat.

## Bediening emulator



Hiermee ga je terug naar het beginscherm en verberg je de applicatie.



Hiermee open je het menu van het huidige scherm.



Hiermee ga je terug naar het vorige scherm of verberg je het onscreen keyboard.



Het afsluiten van de telefoon met de aan/uit-knop werkt niet altijd. De AVD wordt ook afgesloten door het scherm af te sluiten.



Beweeg het slotje met de muis naar de rechterkant van het scherm om het scherm te ontgrendelen. Mogelijk moet je eerst op de aan/uit-knop drukken.

## Meer informatie

<http://developer.android.com/>  
<http://apiwiki.twitter.com/>

# Op bezoek bij: Topicus



## Op de werkvloer bij Topicus

**T**opicus, de ict-dienstverlener uit Deventer met ruim 180 medewerkers die het allemaal net even anders doet. In een van de karakteristieke Topicus panden maak ik kennis met Jan Hendrik Brouwer.

*Jan Hendrik, stel jezelf eens voor.*

Ik ben Jan Hendrik Brouwer, alumnus bedrijfsinformatietechnologie met een master Business Administration. Tussen oktober 2003 en oktober 2004 was ik voorzitter van stichting IAPC en sinds 2007 ben ik actief in het algemeen bestuur van stichting IAPC. Daarvoor heb ik nog in de introductiecommissie van de faculteit Informatica, de IKI, gezeten. In 2007 ben ik afgestudeerd en sinds oktober 2008 werk ik bij Topicus Care als informatieanalist.

*Hoe is het om als pas afgestudeerde bij Topicus te beginnen?*

Hartstikke leuk! Ik ben begonnen op het TisWeb-project binnen Topicus Care. Topicus Care is een afsplitsing binnen Topicus die zich bezighoudt met zaken rondom de verzorgende kant van de zorg, vandaar de naam 'Care'. Het TisWeb project waar ik mee bezig ben is voor een gebruikersvereniging waarin 17 thuiszorginstellingen zijn vertegen-

woordigd die gezamenlijk hun software inkoop.

Binnen het project wordt een bestaande desktopapplicatie vervangen door een webbased oplossing, die eventueel in saas-vorm kan worden aangeboden. De applicatie verzorgt de administratieve processen rondom de AWBZ-zorg. Zorginstellingen krijgen voor mensen die recht hebben op AWBZ-gelden vanwege een chronische ziekte een indicatie binnen van de zorg die ze nodig hebben. De applicatie kan deze berichten inlezen en retourberichten sturen. Ondertussen wordt er voor de cliënten een dossier opgebouwd op basis van de informatie die binnenkomt. Tenslotte kan de geleverde zorg worden gefactureerd aan het zorgkantoor, de cliënt of zijn zorgverzekeraar.

Ik ben binnen het project verantwoordelijk voor het functioneel ontwerp van de applicatie en het onderzoeken van de requirements voor de applicatie. Deze requirements komen tot stand in gebruikersbijeenkomsten, met afgevaardigden van de zorginstellingen die aangesloten zijn.

Jan Hendrik liet me de applicatie zien. De applicatie is van een veredelde PROGRESS-interface getransformeerd naar een applicatie die beter de procesflow

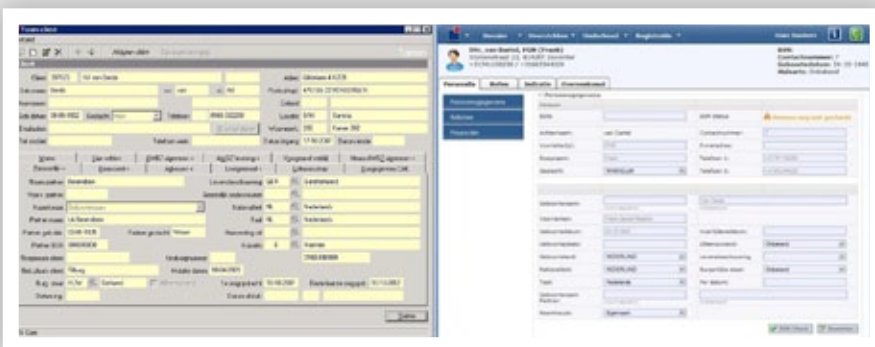
van een zorginstelling modelleert. In de applicatie wordt bijgehouden hoeveel en welke zorg mensen nodig hebben. Medewerkers van zorginstellingen kunnen vervolgens bijhouden welke zorg er daadwerkelijk geleverd is. Voor dat laatste heeft Topicus ook een mobiele applicatie ontwikkeld, voor de Windows Mobile-pda's van de medewerkers van de zorginstelling.

*Waar ga je in de toekomst aan werken binnen Topicus?*

Voorlopig zijn we nog met de afronding van het cliëntdossier voor de thuiszorggebruikers bezig. Daarna volgt een uitbreiding op de applicatie met managementinformatie ten behoeve van de zorgmanagers die gebruik maken van de applicatie. We willen met behulp van een datawarehouse een aantal standaardrapportages aanbieden in de applicatie, maar daarnaast ook ondersteuning bieden voor diverse tools. Op die manier dragen we ons steentje bij aan de efficiëntie van de zorg.

*Je bent als Business Administration-student dan op een heel andere plek terecht gekomen dan dat je zou denken?*

Dat valt best mee. Daarnaast heb ik natuurlijk ook wel een BIT achtergrond vanuit de bachelor. Binnen Topicus is voldoende ruimte om aan de ene kant technisch de diepte in te gaan, en om aan de andere kant meer de management of ondernemerskant op te gaan. Voor een uitbreiding van het huidige project ga ik me bezighouden met de complete planning van het project. Bij een projectmatige aanpak als bij Topicus kun je makkelijk verschuiven naar andere rollen in het project.

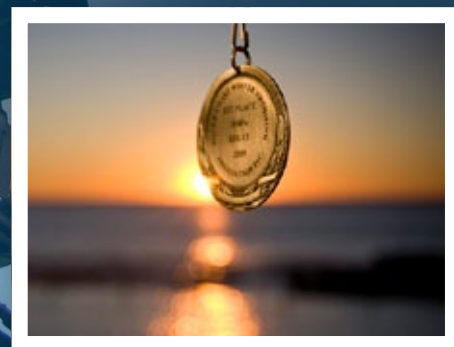


Figuur 1: De oude en de nieuwe client respectievelijk links en rechts.



# VOLGENDE KEER IN I/O VIVAT

- IMPLEMENTATIE VAN DE OV CHIPKAART
- VERVOLG ANDROID TWITTER TUTORIAL
- WINNAAR AV11 WEDSTRIJD



*Achterkant/  
Advertentie  
Belasting-  
dienst*