

MAGAZINE

Wiskunde
Informatica
Studievereniging



'Christiaan
Huygens'

Symposium

BOOKREVIEW

Java Puzzlers

COMPUTERS EN EMOTIES

AI

HISTORISCH PERSOON

Kurt Gödel

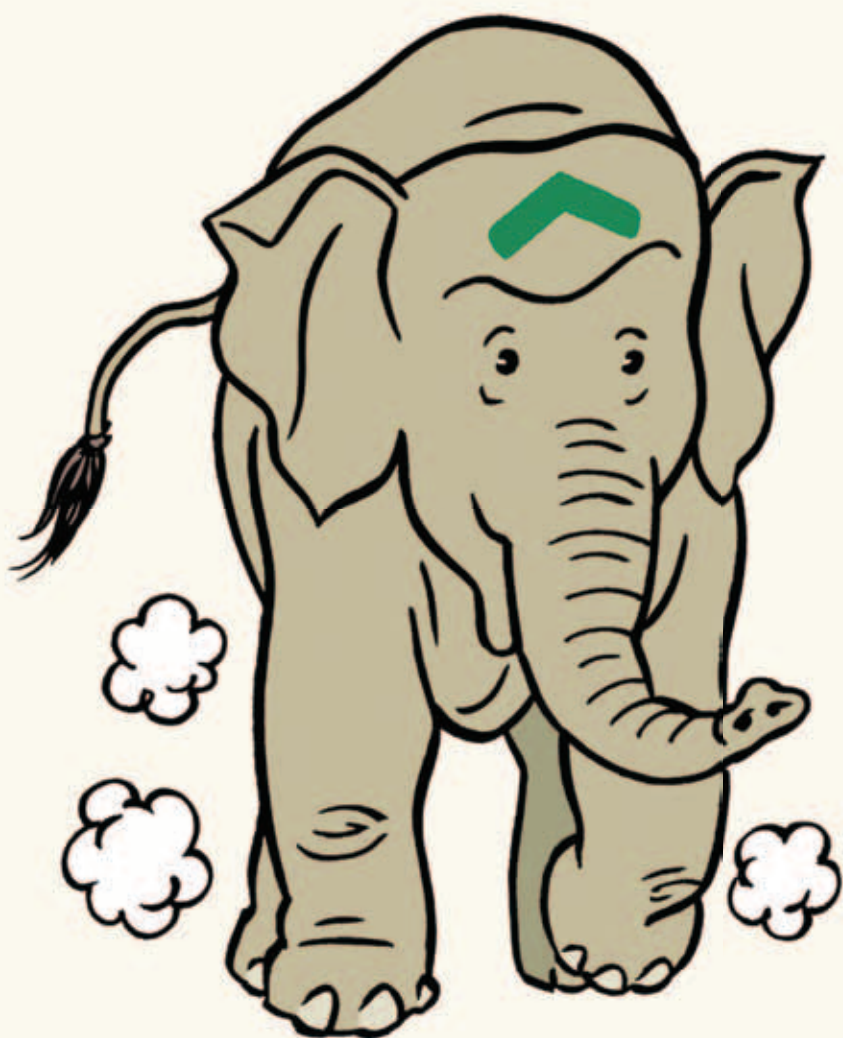
NEW

LAYOUT!

VASTE RUBRIEKEN:

Actueel | Vereniging | Informatica | Wiskunde | Supplementair

Je hoeft niet ziek te zijn om er beter van te worden



Zolang je gezond bent heb je geen zorgen en denk je meestal niet al te veel na over je zorgverzekering. Maar als je ziek wordt, besef je pas wat het waard is om gezond te zijn en hoe belangrijk het is dat zorg goed geregeld is. Dat doen wij. Wij van DSW. Goede zorg mogelijk maken.

DSW
zorgverzekeraar
goed voor je



inhoud en colofon

MACHAZINE

is een uitgave van

W.I.S.V. 'Christiaan Huygens'

Hoofredacteur

Dorien de Regt

Redactie

Max de Groot, Kees Boon, Wikash Sewlal, Michiel van Dam, Anne van Ee (QQ'er)

Beeldredactie

Bruno Scheele, Peter Pul, Bas van Sambeek

Redactieadres

Mekelweg 4, 2628 CD Delft

E: machazine@ch.tudelft.nl

T: 015-2782532, F: 015-2782690

Concept en ontwerp

G2O Kesteren

Drukker

DeltaHage bv

Cover

Owls photo - by hagit on sxc.hu

Ribbon image - taken from omnigifts.co.uk

Aan dit nummer werkten mee

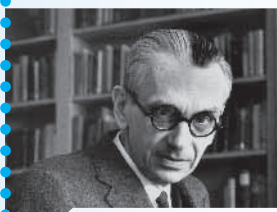
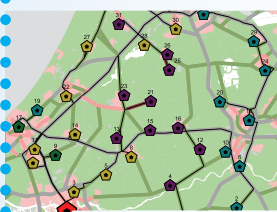
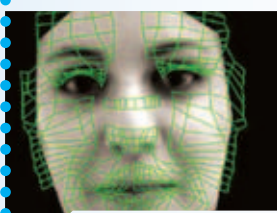
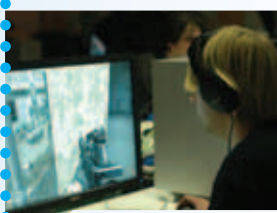
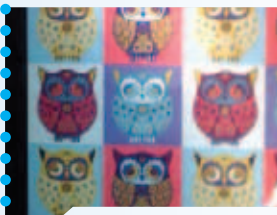
Arie Troebel, Tom Verhoeff, Mark Janssen, Sanne Aalbers, Anne van Ee, Ankur Sharma, Prof.dr. C. Witteveen, Bradley Galdey, Fred Vermolen, Jeremy Rauss, Paul Brussee, Prof. dr.ir. H.J. Sips, Marco Krikke, Dr. P.G. Kluit, Joost Broekens, Koen Lubbers, Joey van den Heuvel, A.S. Pruteanu, Dennis den Ouden, Prof. dr. J.M. Arts, Radboud Duintjer Tebbens, Tijmen P. Collignon, Ester Stegers, Rijn Buve, Marjon Ruijter

Algemene Voorwaarden

De MaChazine-commissie en het bestuur zijn verantwoordelijk voor de inhoud van het Machazine, met dien verstande dat de mening van een schrijver niet (noodzakelijk) de mening van de redactie of de vereniging weergeeft. Alle rechten voorbehouden. Niets uit deze uitgave mag worden vervoelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand of openbaar gemaakt, in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnamen of enige andere manier, zonder voorafgaande schriftelijke toestemming van de uitgever.

Adverteerdersindex

Thales	Achterkant Kaft
DSW	Binnenkant Kaft
ASML	14, 15
Technolution	20, 21
All Options	32
Quinity	49



Algemeen

Redactioneel	2
CH Twitter	2
Agenda	2
Arie Troebel	3
Van Bestuur 53 - Faalprojecten	4
Faculty Student Council	5
Column international student	5
Column - Reflectie	6
Announcement - Teacher of the year	6

Vereniging

Gamen tot je neervalt - LAN-Party on!	7
Fred Vermolen - Genomineerd voor 'beste docent van de TU Delft'	7
Symposium - Cloud your Identity, Distribute your Life	8

Informatica

Ken je Prof! - Prof.dr.ir. H.J. Sips	9
Developing the TU Delft 'My Timetables' for Blackboard application	12
Informaticapuzzel XXII	16
Oplossing informaticapuzzel XXI	16
Artificiele intelligentie - Computers en emotie	17
Zero emission racing! Greenchoice Forze	19
Book review - Java puzzlers	24
Clustering algorithm for highly mobile wireless sensor networks	25

Wiskunde

Mathematical modeling of particle nucleation and growth in metallic alloys	27
Wiskundepuzzel LI	29
Oplossing wiskundepuzzel L	29
Wiskunde en de paradox van rationele vaccinatiebeslissingen	30
Fast Iterative Solution of Large Linear Systems on Grid Computers	33

Supplementair

Historisch Persoon - Kurt Gödel	37
Alumnus - Rijn Buve, CTO TomTom	41
Reizen tijdens je studie - Stage bij de Schotten	43





Laat ik nu maar niets over de iPad zeggen! (vrij naar Ferdinand Peper, Stelling 11, 1989)

Beermile start! <http://twitpic.com/16k758> #chdies

Koffie/frisdrank/snoepautomaat op de #tudelft leeg? Geef het door en wij houden het bij: <http://bit.ly/c86pWr>

Ladies...Ask not what Barney Stinson can do for youAsk what you can do for Barney Stinson.

A new style == new opportunities?

So, isn't this a nice gift to you? A shiny new MaCHazine, and even in a shiny new jacket!

As a break of tradition, this editorial is written in English. Why in English all of the sudden? Wasn't Dutch good enough? Well, since the MaCHazine has made a new start with its new face, perhaps it's also time that everything else has a fresh reboot as well. Getting more English articles for our foreign readers is a great start.

There are a lot of great articles in this MaCHazine. We have our usual suspects in the form of information about activities and issues within the faculty and editorials by a lot of great guys. We have reports of activities that CH has to offer and we have a lot of nice articles about various subjects in the field of Computer Science and Mathematics. And just for a bit more flavor, we revisit our historical heroes and recount tales of studies of students in the world. But of course, everything can be improved. That's why we want the input of all our readers, including the English ones. Hence my choice of language for this editorial :).

So, please tell us what you think of the MaCHazine by sending us a mail addressed to machazine@ch.tudelft.nl. Like the new layout? Mail us a compliment. Hate it? Mail us your criticism. And of course the same goes for our articles. Are there particular subjects you would like to read about? Would you like to write an article yourself? Would you enjoy a single issue of the MaCHazine to cover only one topic, such as game design or financial mathematics? Or do you enjoy having a MaCHazine with articles about various topics?

We always enjoy hearing from our readers and there is no better time to give us your opinions than with this new start. 

Agenda

April

Thu 22 Excursion to Vanderlande & Quintech

Fri 23 International Food Festival

Tue 27 Symposium 'Cloud your Identity, Distribute your Life'

May

Thu 6 Members Lunch

Fri 7 Excursion to KLM

Wed 12 Siaarcie BBQ

Tue 18 General Assembly

Wed 19 MatCH: Games Afternoon

Thu 20 Party

Tue 25 Inhouseday at KPMG

Wed 26 Mathematics Lecture by Radboud Duintjer Tebbens

Wed 26 Lecturer of the Year Award Ceremony

June

Thu 3 Committee Thanks Day



Arie Troebel

The difference machine

Lieve schatjes in het mooie Delft,

Ik zat in de bioscoop, nou nee, ik moest naar de bioscoop met mijn

vrouw, en als ik ergens een hekel aan heb... ja dat dus!

Dus wat gaat een beetje wetenschapper dan doen?

Hij gaat zich uit verveling verdiepen in de Babyloniërs, die zo lekker aan het spelen waren met hun getalletjes en dat allemaal gebaseerd op het getal 60!!

Jaja, ze konden er wat van, wat te denken van de algebraïsche formule als $ab = ((a + b)2 - (a - b)2)/4$. Ja, je moet wat doen in zo'n donker collectieve pijnbank.

Maar goed ik dwaal even af.

Ik wil het met jullie even hebben over het heerschap Charles Babbage (1791-1871), en natuurlijk ook heel belangrijk Ada Lovelace (1815-1852)!

Charles was een vrij rustige man, die zich met volle overgave gaf aan 'nog niet eerder ontworpen dingen' te maken.

Hij zat op een zondagmorgen wat te prutsen en te sleutelen aan een machine wat hij zelf 'the difference Engine' noemde. 'Ja, je moet het beestje toch een naam geven', prevelde hij.

Het zag er mooi uit vond hijzelf. 'Ik had wel IO kunnen studeren, maar ja dat bestaat nog niet,' praatte hij wederom tegen zichzelf.

Charles was in een beste stemming, want een vriendin zou langskomen om 'the difference Engine' te zien.

Op de deur werd gebonkt en daar was ze, een plaatje zoals altijd.

'Hi Charles,' zei ze met een zwaar Cockney accent, 'wat ben je aan het freubelen?'

Charles: 'Ada, dit wordt een regelrechte hit!'

Ada: 'Ja maar wat is het dan Charlie?'

Charles: 'Jee, jij ook met je moeilijke vragen altijd Ada! Vind je het niet mooi dan?'

Ada: 'Mooi is het wel, maar is het ook functioneel, wordt het wel iets voor de toekomst? Ik bedoel: kunnen we er iets op gaan doen, of zetten we het gewoon in de hoek met een Feijoa Sellowiana plantje erop?'

Charles: 'Ada, op dit moment schiet me er even niets te binnen, maar als je wilt mag je ook wel iets doen, hoor. Kijk maar wat je uit gaat spoken!!'

Dat liet Ada zich geen twee keer zeggen, vele uren en dagen later kwam Ada weer terug op het honk van Charles, en ze zei: 'Charles, ik heb wat opgeschreven op wat perkament; mag ik er wat mee doen?'

Charles: 'Natuurlijk Ada,' je gaat je gang maar, doe wel voorzichtig want hij is net af!'

Ada ging aan de gang en de hele zware machine zette zich in beweging. 30 Minuten later kwam er wat uit rollen, en op dat moment liet ook de machine het afweten. Dus volgens de overlevering heeft hij het nooit gedaan, maar bij dezen is dat dus rechtgezet, en dat een vrouw de eerste was die ooit een computer programma schreef. Maar wat was het nu wat er uit de 'computer' kwam rollen?

Dit dus: Charles ma +2bier^{1wijn} 



Van Bestuur 53

Faalprojecten

Sanne en Anne, de Commissarissen Onderwijs

Sommige dagen zijn niet zo productief als andere dagen. Je probeert je best te doen, maar er gebeurt gewoon niet zo veel. Zo was afgelopen woensdag het geval voor ons. Er waren deze dag geen speciale CO taken voor ons weggelegd, zoals vergaderingen met de CRI's, CRW's, FSR, OCV, OCI, OCCEES of SVR-O. Jullie zien het, ons leven bestaat uit een grote verzameling afkortingen. Wel was er een meeting met de onderwijscoördinator waarbij het de bedoeling was om het juist niet over de problemen van de studies Wiskunde en Informatica te hebben, maar om eens een persoonlijk gesprek te voeren. Dit faalde al meteen.

Het gesprek met Julia begon als volgt: "Even nog kort een paar dingen over wat wij gisteren besproken hadden." Hoewel we elke dinsdag al een vergadering hebben, konden we het toch niet laten om vervolgens alsnog binnengekomen klachten te bespreken. Gelukkig konden wij daarna praten over hoe stom versus leuk kerels zijn en alle problemen die bij hen komen kijken.

Tijdens de lunch ging gelukkig niets fout en daarna konden we dan ook met een gerust hart op expeditie. De bedoeling was de uilenposter bij de H&M te claimen, die we al een tijdje terug hadden gespot in de etalage. Deze super-mooie poster zou namelijk erg goed staan op het CH hok of in het vergaderhok. Jammer genoeg was de H&M ons niet zo goed gezind, want wat blijkt: alle posters worden teruggestuurd naar een 'unknown location'. Zelfs het pruillipje hielp hierbij niet, want helaas werken er alleen maar vrouwen.

Huilend kwamen wij terug op het CH hok! Er stond niets anders op dan de normale actiepunten af te knallen. Sanne ging haar lezing promoten via het servicepunt en alle secretariaten. Anne moest een logo regelen voor het

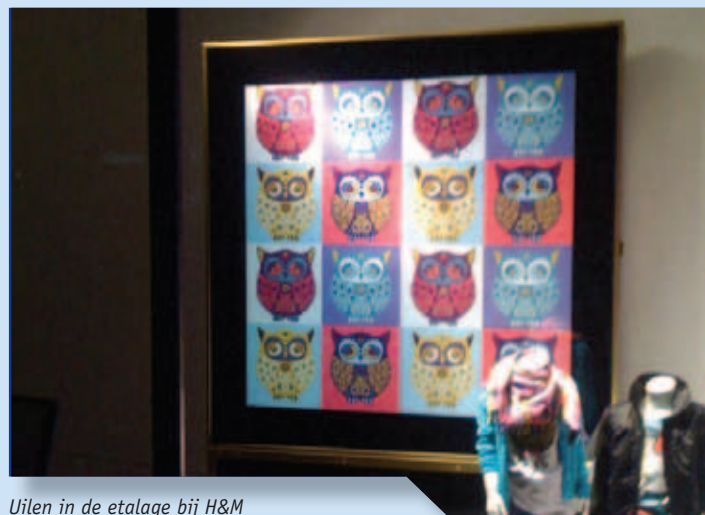
nieuwe CH voetbalteam, maar had meer zin in iets oncreatiefs na de grote uilenposter-domper. Alle CRI tussenevaluatie agenda's liggen nu een maand van te voren al klaar...

Het was tijd voor iets leuks! We besloten onze Pentagonvriendjes een bezoek te brengen om voor het Sjaarciefeest DDB Eindfeest kaartjes te verkopen. Bepakt met een grote kas gingen wij op weg. Onderweg zorgde Sanne voor een indrukwekkend rinkelconcert. Misschien hadden we dit kunnen waarderen als wij penningmeesters waren, maar nu was het gewoon irritant. Aangekomen bij de olifanten-lovers werden wij met enorme vreugde ontvangen. Al snel bleek grote onduidelijkheid over hoeveel mensen aanwezig konden zijn en daardoor werden er geen kaartjes gekocht. Gelukkig is er een toezegging gedaan dat vandaag iemand langskomt voor de kaartjes; of dit daadwerkelijk gebeurt, blijft de grote vraag.

Aan de overkant van het Mekelpark konden we de VvTP vinden. Gelukkig waren er nog twee bestuurders aanwezig die ons verhaal wilden aanhoren. Bij dit bestuur was ook al veel onduidelijkheid over het aantal aanwezigen, maar wij hebben de positieve verwachting een groot deel van hen op het feest te zien! Toen was het tijd om TG een bezoek te brengen. We hadden namelijk gehoord dat zij heel erg enthousiast waren om hun medebestuurder in zeer aangeschoten toestand te aanschouwen. Hier konden wij dan ook eindelijk wat kaartjes kwijt! Dit faalproject had gelukkig een goed einde.

Nu denken jullie misschien: waar is het vijfde Pentagonbestuur gebleven? Zij zitten ver in het zuiden en wij hadden het briljante inzicht om eerst te bellen. Zo hebben wij tijd en moeite bespaard, want een bezoek aan hen had geen resultaat opgeleverd. Helaas zit Sanne nog zonder handdoek, maar deze komt vast nog een keer naar haar toegevolgen!

Niet elke dag is zoals hierboven beschreven, maar je kunt natuurlijk niet altijd knallen. Op het moment zijn wij bezig met het maken van modulekaarten voor Wiskunde en Informatica. Ook baart het tweede instroommoment van de masters ons zorgen, omdat deze zowat niet bestaan. Ook al is, naar aanleiding van de invoering van de Harde Knip, toegezegd door het CvB dat een tweede instroommoment bij elke studie mogelijk moet zijn, onze faculteit heeft al aangegeven dat zo'n tweede instroommoment nooit nominaal studeerbaar zal worden. Genoeg zaken dus waar wij ons nog over kunnen ontfermen als we geen faaldag hebben. 



Uilen in de etalage bij H&M



Faculty Student Council

A new MaCHazine, a new report from the Faculty Student Council.

Tom Verhoeff & Mark Janssen

So what have we been up to the past few months? Apart from all the complaints we handled, there were some major points we worked on; let us take you through some of them.

First of all, let's talk about the faculty restaurant. We received a lot of complaints about the old situation: the available space wasn't used efficiently enough and long queues were a recurrent issue. We worked with the faculty to come up with a restructured layout, which has been realized last December. Because the changes might not be sufficient, we will be sure to continue our collaboration with the faculty to resolve all issues.

An improvement we have made on the side of education is the introduction of so-called module cards, which provide a clear and fast view of a study program. It is easy to see which courses are given when and the number of ECTS they are worth. We try to have the new module cards on the faculty website within the coming weeks. They will also be the main course overview starting next year, replacing the traditional study guide.

One of the first problems we encountered this year was the new timetable interface. The system introduced in September lacked a lot of features an EEMCS-student cannot live without, such as export to iCalendar files. Combining different free-elective courses on one page also proved to be an impossible task. The IT department promised that a new system would be available before the start of the second semester, but a progress check a few months later revealed that no progress had been made. Fortunately, they decided to offer the job of creating a new interface to four students from our faculty. You can read more about their work in this MaCHazine.

Do you want to make a difference on our faculty? New Faculty Student Council elections are coming up on the 26th and 27th of May and we are looking for skilled successors. If you are interested, please let us know at fsr@ewi.tudelft.nl. We will be sure to answer all your questions, and if you are up to it, add you as one of the candidates for next year. The application deadline is April 1st.

Column international student

Let it snow!

Ankur Sharma

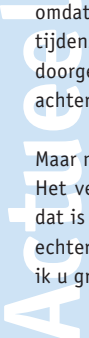
As I opened the door after a long and dreamy night, I was in a different world. For a second, it seemed as if I was colorblind, but I wasn't –pew what a relief! Everything was white and riding a bike to the library wasn't my best idea. It was close to the end of 2009, I had seen snow before, but as I discussed later with my Dutch friends, Delft hadn't seen such snow for a long time. So, as I walked dragging my bike on the 3 inches deep snowy roads, I found more snow enthusiasts and we ended up having a snow fight. I was 15 again, and I loved that.

The Snow fight made me tired, eventually, and I remembered where I was destined to. The TU Delft Central library, however, wasn't the same place where I used to go everyday. It looked like a white snowy mountain with people with snowboards, signboards and virtually anything that could be rolled on – Nice!

So I hopped on and I was still feeling 15, so it didn't matter. I found a few guys who got hold of a traffic sign board and were using it to slide on the snow - this is what I call innovation. People kept coming and soon we had kids, grandpas, guys and gals enjoying this beautiful day.

It was fun for both international and Dutch students and everyone found a good reason in the white snow to escape from the books and relax for a while and I am sure these were some moments that everyone there will remember for a long time. Eventually the 15 in me started feeling hungry and I started to look for my bike that was buried somewhere in the snow carpet. As I walked back and met more kids fighting with snow bombs, I weirdly felt that in the next days the snow is going to melt and will life will be colorful again, but I wanted the snow, I wanted to be 15 all the time. As I cooked a spicy hot meal at home, it started snowing again.

Actual



Reflectie

Zoals u weet, krijg ik van onze universiteit iedere maand een ruime vergoeding om problemen te bedenken en op te lossen. Zo op het eerste gezicht een ideale baan: zelfs als ik meedeel dat een oplossing niet bestaat of tijdens mijn leven niet gevonden kan worden, dan wordt mijn salaris toch keurig op mijn rekening gestort. Het enige nadeel aan deze baan is dat je soms heel hard en lang moet nadenken. (Ter toelichting: nadenken is een oude term voor mentaal googelen: in plaats van het internet stuur je queries door je hersenpan in de hoop dat je met de gegenereerde hits iets kunt doen.)

Maar nu blijkt dat ik het al deze jaren verkeerd heb gedaan. En hoe kwam dat? Het verrassende inzicht is dat ik op de oude manier had leren nadenken. En dat is heel funest. Een nogal dure, maar fantastische, cursus heeft mijn leven echter drastisch en fantastisch veranderd. Omdat ik niet de beroerdste ben, zal ik u gratis het recept van deze cursus uit de doeken doen.

Nu komt het moeilijkste gedeelte van de cursus: reflectie is stap één, maar je kunt natuurlijk ook weer reflecteren over je reflectie. Dat heet meta-reflectie. Ik geef een voorbeeld: ik had een probleem, daarover leer je reflecteren, zo kun je praten over het feit dat je een probleem hebt. Dan komt de volgende (meta) reflectiestap: Wat is nu eigenlijk mijn probleem als ik een probleem heb? Kom je nu tot de geniale conclusie (en dat was de bedoeling van de cursus) dat dat eigenlijk geen probleem is, dan heb je daarmee in een klap drie problemen opgelost. Kijk, dat noem ik nu een cursus die z'n geld op brengt.

Met een, twee stappen los ik daarom voortaan elk probleem op. Alleen maar door te (meta-)reflecteren. Daar kunt u toch moeilijk een probleem mee hebben? Toch wel? Dan moet u het recept hierboven toch nog eens aandachtig doorlezen. Veel succes. 

Teacher of the year

MaCHazine staff

Upcoming quarter, the annual teacher of the year contest will be held. Grade the teachers of the courses you took this year, instead of having them grade you! You can also add comments and motivation. Your opinion may get your favorite teacher to become teacher of the year of the faculty of EEMCS and perhaps even the whole TU. For more information about the teacher of the year award, please read the article by Fred Vermolen, last year's teacher of the year of this faculty. For more information about this year's election, please contact the Chief Commissioners of Education. 



Gamen tot je neervalt

LAN-Party on!

Bradley Galdey

Hoewel CH als studievereniging bovenal op de studies gericht is, bestaat er natuurlijk ook een tijd voor plezier. De LAN-party in het weekend van 30 en 31 januari is daar een uitstekend voorbeeld van.

Een bepaalde (om de één of andere reden niet totaal onverwachte) hoeveelheid CH-ers speelt graag een spelletje. Aangezien algemeen bekend is dat meer zielen meer vreugd tot gevolg hebben, is met anderen spelen populair, en dat maakte ook de afgelopen LAN-party een succes. Net als het jaar daarvoor was voor de locatie weer gebruik gemaakt van buurthuis 'Het Voorhof'. Hier kwamen op zaterdagochtend de eerste mensen binnen met hun pc's om zich te nestelen voor het weekend. In de loop van het weekend werden hier verscheidene spellen op gespeeld met de rest, waaronder bijvoorbeeld 'Unreal Tournament'. Ook was er even een Xbox 360 aanwezig, waarop fanatiek voetbal werd gespeeld, en de 'oeehh's en het harde gelach bleven dan ook niet uit. Later maakte de Xbox plaats voor een Wii, waarop vele hectische potjes 'Super Smash Bros.' zijn gespeeld.

Tussen al dat vechten en schieten door werden de deelnemers goed verzorgd. Op de zaterdag was er een lunch beschikbaar en op de zondag naast lunch ook een ontbijt voor iedereen. Daarnaast was er voor degenen die daar behoefte aan hadden een slaapruijme klaargemaakt, waar met behulp van een slaapzak of iets dat dezelfde functie vervult een welverdiende pauze kon worden genomen.

Verder werden er, zodat de deelnemers niet geheel op zichzelf waren aangewezen, een aantal toernooien georganiseerd. In teams van twee konden spelers het tegen elkaar opnemen in 'Call of Duty: Modern Warfare 2', en ook in 'Super Smash Bros.' konden de deelnemers laten zien wat ze echt waard zijn in één-op-één-gevechten.

In het geheel kan naar mijn mening gesproken worden van een wederom geslaagde activiteit, ik heb me erg vermaakt en zal er daarom volgend jaar weer bij zijn, die LAN-party houden we erin. 

Genomineerd voor de titel 'beste docent van de TU Delft'

Fred Vermolen

Tot mijn grote verbazing werd ik gekozen tot docent van het jaar 2008-2009 van de opleiding Technische Wiskunde. Deze titel werd me uitgereikt vanwege het vak Numerieke Methoden II dat aan derdejaars technische wiskunde studenten gegeven wordt. Voor mij was dit al een grote eer, zeker omdat er zo vreselijk veel heel goede collega-docenten in onze opleiding rondlopen. Een aantal collega's acht ik veel beter dan mezelf. Sterker, een echt goede docent vind ik mezelf eigenlijk helemaal niet. Het enige dat ik probeer is mijn werk goed te doen.

Nog meer verbaasd was ik toen ik te horen kreeg over mijn nominatie als beste docent van de TU Delft. Ik moest de EWI-faculteit vertegenwoordigen en het opnemen tegen zes zeer goede kandidaten waarvan ik er twee persoonlijk ken. Er kwam een interview met een fotosessie van het TU blad Delta, wat resulteerde in een mooi artikel. Uiteindelijk heb ik de titanenstrijd niet gewonnen en is de prijs gegaan naar Susanne Rudolph. Deze docente is erg goed en daar kon ik het eenvoudigweg niet van winnen. Ze vertelde dat als de studenten haar stof in een college niet begrepen, dat ze dan een extra college inlaste. Ze is dus heel erg betrokken met haar studenten en dat wordt dan ook terecht gewaardeerd en ik feliciteer haar dan ook met de uitslag.

De prijs van Beste docent van de TU Delft is dit jaar voor het eerst ingevoerd. Mijn eerste indruk was: 'In wat voor een kermis ben ik nu weer beland?' Achteraf bezien, denk ik dat deze prijs een goede zaak is. Het onderwijs op universiteiten is lange tijd ondergesneeuwd geweest onder het onderzoek. Hoewel ik een gepassioneerd onderzoeker ben, hecht ik zeer veel waarde aan het onderwijs. Het is leuk, vernieuwend, leerzaam en motiverend om met studenten (en jongeren in het algemeen) te werken. Goed onderwijs levert op de langere duur goede technisch wiskundigen op waarvan het onderzoek en het bedrijfsleven dan kunnen profiteren. De grotere waardering voor onderwijs zet aan tot goed onderwijs. De tijd waarin mensen die veel onderwijs geven als half-zacht worden aangezien, is voorbij en dat is maar goed ook, want de TU Delft is en blijft op de eerste plaats een onderwijsinstelling. Ook hoop ik dat de TU Delft haar goede naam in het onderzoek en onderwijs behoudt of -nog liever- verbetert.

Tenslotte merk ik op dat ik van vrijwel iedere werkdag aan de TU Delft geniet, en dat ik de combinatie van onderzoek en onderwijs in mijn werk als een voorrecht beschouw. Ik hoop dan ook nog lang te mogen genieten van wiskunde, onderzoek (ook in aanverwante disciplines), onderwijs en studenten. 



Cloud your Identity Distribute your Life

Op 27 april 2010 zal W.I.S.V. 'Christiaan Huygen' een symposium neerzetten voor Technische Wiskunde- en Informaticastudenten. In het symposium 'Cloud your Identity, Distribute your Life' zal een overzicht gegeven worden van de huidige ontwikkelingen op het gebied van gedistribueerde systemen, Peer-2-Peer samenwerking, cryptologie en vooral de privacy-gevolgen en wetgevingsprocessen hiervan. In het symposium zullen de onderwerpen daarom niet alleen vanaf de wetenschappelijke kant, maar ook vanaf de filosofische en sociale kant benaderd worden.

Locatie

Het symposium zal gehouden worden in de MustSee bioscoop Delft. Het programma bestaat uit twee tracks, een toegespitst op informatica en een toegespitst op wiskunde en de meer maatschappelijke lezingen. Na elke lezing is er de mogelijkheid te wisselen tussen de twee tracks. Er is ook een lunch in de middag en een borrel achteraf.

Gedistribueerde systemen

Toepassing van gedistribueerde systemen is soms vereist, omdat er afstand zit tussen waar de data is opgeslagen en waar het gebruikt wordt. In andere gevallen zou een taak door een enkele computer uitgevoerd kunnen worden, maar om praktische redenen is een gedistribueerd systeem beter, bijvoorbeeld om kosten te drukken. In dit licht zal professor Wesley Petersen van de ETH Zürich een lezing houden over berekeningen op grafische processoren, game processoren en Intel multi-core. Hierbij zal hij ingaan op de uitdagende aspecten van het gedistribueerde element.

Peer-2-Peer

Aan de Technische Universiteit Delft wordt toonaangevend onderzoek verricht op gebied van Peer-2-Peer samenwerking. Johan Pouwelse zal een lezing geven over de nieuwe ontwikkelingen in alweer de vierde generatie van Peer-2-Peer samenwerking.

Cryptologie

Het beschermen van gegevens kan op veel manieren. Dit onderwerp zal behandeld worden met oog op de andere onderwerpen van het symposium. Hoofd van de Cryptology and Information Security groep van het Centrum Wiskunde & Informatica en professor Cryptologie aan de Universiteit Leiden, Ronald Cramer zal een lezing geven over Secure Multi-Party Computation. Ook maatschappelijke ontwikkelingen van cryptologie zullen aan bod komen in de lezing over pseudonimisering van persoonsgegevens door Eric Verheul.

Privacy en wetgeving

Naast de technische kant van de bovenstaande onderwerpen, zal ook de maatschappelijke kant aan bod komen tijdens het symposium. De schijnbare tegenstelling van aan de ene kant persoonlijke gegevens proberen te verbergen en aan de andere kant meedoen in maatschappij die efficient informatie deelt wordt aangesterkt door huidige ontwikkelingen in de techniek. Hier zullen wat voorbeelden van worden gegeven. Melanie Rieback zal een lezing houden over RFID Guardian, dat staat voor Radio Frequentie Identification. Deze techniek wordt onder andere toegepast in de OV-chipkaart. IT-journalist Brenno de Winter zal meer vertellen over de OV-chipkaart. 



Ken je Prof!

Prof.dr.ir. H.J. Sips

Jeremy Raes & Paul Brussee

Waar doet uw sectie onderzoek naar?

Het vakgebied van PGS valt uiteen in twee delen: parallel en gedistribueerd. Parallel is eigenlijk vrij simpel. Het houdt in dat alle processoren samen zo snel mogelijk een taak uitvoeren. Dit lijkt eenvoudig maar dit blijkt in de praktijk een weerbarstig probleem te zijn. Zeker met de moderne multi-core machines, wordt het ons software ontwikkelaars er niet makkelijker op gemaakt. Een aantal nare problemen die klassiek altijd verborgen bleven kunnen ineens in volle hevigheid naar voren treden. Denk hier bijvoorbeeld aan het probleem van latency of de beperkte omvang van je resources of hoe lang je de CPU nog bezig kan houden gegeven een bepaalde bandbreedte. Het moet allemaal blijven passen in je geheugen. Zodra het allemaal niet meer past krijg je gelijk een enorme performance penalty voor je kiezen. Er zijn voorlopig nog geen software tools die dit kunnen verhelpen. Twee machientjes in een dual core kan je nog wel bezig houden, maar zodra het er 40 worden, wordt het toch een heel ander probleem.

Denkt u dat een oplossing überhaupt mogelijk is?

Ik ben toch wat pessimistisch. Het is niet zo dat je het niet kan programmeren, maar het is wel heel arbeidsintensief. Portabiliteit, het overdragen van het ene systeem naar het andere, is ook niet altijd fantastisch. Daar zit toch het merendeel van de software ontwikkeling in tegenwoordig. Hoe zich dat verder zal uitkristalliseren, blijft voorlopig nog de vraag.

Het aantal cores blijft eigenlijk vooruitlopen op onze mogelijkheid om ze te benutten? Ja, eigenlijk wel. De focus is nu ook verschoven van een zo snel mogelijke CPU maken naar systemen om data aan te voeren. De CPU's zijn wel snel genoeg, maar ze krijgen hun data niet snel genoeg aangevoerd. Ze moeten veel te veel wachten, want alles wat afstand moet overbruggen is traag.

Er zijn wel enthousiastelingen die zeggen dat je de hardware niet plat moet maken, maar in 3D. Dit levert echter weer warmteproblemen op.

Is dit niet meer een probleem voor elektrotechnische ingenieurs?

Voor informatici vrees ik. Aan de hardware kan uiteraard nog wel wat worden verbeterd, maar fundamenteel zijn daar geen oplossingen voor. Zelf denk ik dat men de algoritme moet veranderen. Mensen gaan er ten onrechte van uit dat

ze over oneindige resources beschikken. Neem bijvoorbeeld MPEG4 met al die object coderingen. Vanuit een informatica standpunt lijkt dit fantastisch, maar vanuit een hardware technisch oogpunt is dit een ramp. Alle regelmaat is weg en je introduceert een hoop afhankelijkheden. Eigenlijk kan je niet anders constateren dat voor de moderne hardware deze algoritmes helemaal niet geschikt zijn, maar het zal nog wel even duren voordat men ervan overtuigd raakt dat er technologisch toch echt geen oplossing bestaat voor dit probleem. We moeten nieuwe vormen van oplossingen verzinnen. Hetzelfde geldt voor problemen die nu algoritmisch duur zijn vanwege de enorme complexiteit. Het kan wel eens zo zijn dat deze op nieuwe machines juist goedkoper worden.

Jullie doen ook nog onderzoek naar gedistribueerde systemen.

Bij gedistribueerde systemen werken we aan P2P-systemen, met de focus op video-on-demand over het Internet.

Het ultieme gedistribueerde is de interactie tussen een aantal autonome systemen die met elkaar communiceren, bijvoorbeeld over het Internet. Bittorrent met gedistribueerde trackers is daar een voorbeeld van. Daarin heb je tal van uitdagingen met bijvoorbeeld schaalbaarheid. Je krijgt ook te maken met alle negatieve kanten, zoals illegaliteit. Als je er puur technisch naar kijkt, zijn het slechts systemen die bits verplaatsen, maar het gaat allemaal zo snel dat je het ook op een manier kan gebruiken die de contentindustrie niet zo aanstaat.

Dat zou de wetenschap niet in de weg mogen zitten van de ontwikkeling van deze technologie?

Dat is bij ons altijd een hele discussie geweest. Toen wij met ons onderzoek begonnen, hebben we hierover lange gesprekken gevoerd met de stichting Brein. Zij vonden het helemaal niets.

Ik heb hier altijd het standpunt ingenomen dat wij geen illegale content willen op onze servers. Wat mensen met software doen is hun verantwoordelijkheid, maar hier aan de TU wil ik het niet. Nu zit daar soms een dun lijntje, want we doen soms wel eens experimenten waar we tijdelijk wel even deelnemen voor bijvoorbeeld metingen. Mijn standpunt is dat dit wel geoorloofd is, want anders zou je ook geen onderzoek naar welke vorm van illegaliteit dan ook kunnen doen. Soms moet je een beetje participeren om het systeem te meten. Zoiets moet natuurlijk wel strikt afgebakend worden in tijd en doel.



Een ander vraagstuk is anonimiteit op het Internet. Ook daar zitten positieve en heel negatieve kanten aan. Positief is uiteraard dat daarmee de hele censuur in China om zeep geholpen kan worden. Negatief is dan weer dat het medium ook gebruikt wordt om bijvoorbeeld kinderporno mee uit te wisselen. Dat zijn toch hele fundamentele kwesties.

Bent u daar zelf al uit?

Ik vind dat heel moeilijk. Technologie is agnostisch, net zoals een hamer. Je kan er hele positieve maar ook hele negatieve dingen mee doen. (Pauzeert, denkt even na) Ik ben er dus niet helemaal uit of we dataverkeer traceerbaar moeten maken en zo sommige zaken toch opspoorbaar moeten maken, of niet.

Dat verhaal uit de cryptografie kennen jullie waarschijnlijk wel: een onderzoeker had een cryptografisch algoritme gekraakt en daar een paper over geschreven. Maar hij durfde daarna niet meer naar de Verenigde Staten, omdat hij gearresteerd zou kunnen worden. Zoiets kan uiteraard niet de bedoeling van de wetenschap zijn. Je moet dat soort systemen kunnen testen.

Als je naar de geschiedenis van de media kijkt, zijn er altijd wel punten waarop de contentindustrie zich bedreigd voelt. Eerst was het de grammofoon, daarna de opkomst van de radio, vervolgens de televisie en nu het Internet.

De derde tak van ons onderzoek spitst zich toe op grid- en cloud computing. Ook op dit onderwerp zijn we al een aantal jaren actief. We willen nu uitbreiden richting de massieve multi-player online games. Denk hierbij aan een platform voor iemand die zo'n spel wil maken waarbij de entry costs enorm gereduceerd worden.

Waarin verschilt de opleiding hier van andere informatica opleidingen in Nederland?

Wat ik van Hans Tonino [opleidingsdirecteur informatica, red] begrijp, is dat de studenten de opleiding goed waarderen. We hebben goede docenten en hebben veel geïnvesteerd in vernieuwing de laatste jaren. Ik denk dat een student hier aan de TU meer mogelijkheden heeft om zijn eigen interesses tegemoet te komen met behulp van keuzevakken. En juist omdat je aan de TU zit kan je vakken hebben die bijvoorbeeld meer in de telecom zitten, of juist meer op hardware gericht zijn.

Elk curriculum is een compromis, want je kunt natuurlijk niet alles doceren. De afgelopen twee jaar hebben we rustig ons nieuwe curriculum gedraaid en dat draait goed. In september krijgen we een kleine vernieuwing om het een en ander wat moderner te maken.

Hoe was u zelf in uw studententijd? Waren er bijvoorbeeld vakken die u moeilijk vond? Wist u toen al dat u professor wilde worden?

Nee, dat was geen vooropgezet plan, dat is zo gelopen (lacht). Maar als je eenmaal het besluit genomen hebt om in de academische wereld te blijven, dan wil je natuurlijk wel hogerop. In mijn tijd kon je nog lang studeren. Zelf heb ik er in plaats van 5 jaar nominaal volgens mij ook 7 jaar over gedaan, waarvan ik een jaar ziek geweest ben. Maar ik had het ook in 5 jaar kunnen doen, dat was geen enkel probleem geweest. Doordat je de eerste 2 jaar veel strakker in het stramen gehouden werd was het makkelijker om nominaal te blijven studeren dan daarna. Want daarna had je allemaal keuzevakjes en dan gaat het logistiek wat rommelen en voordat je het weet schuift het uit. Ik vond mijn eigen studententijd erg mooi. In die tijd waren studentenvereningen ook ineens niet meer zo populair.

Ik ben als student wel actief geweest binnen het toen net opgerichte AAG en heb een tijd in een studentenraad gezeten hier op deze faculteit. De universitaire structuur was toen nog wel heel anders. Ook waren de studentenprotesten nog vrij heftig. Dat was de tijd van '68-'69, nog voor de Wet Universitaire Bestuurs Hervorming.

De studentenprotesten die nu spelen zijn van hele andere aard. Toen ging het echt over medezeggenschap. Er waren toen een aantal mensen heel erg actief, waarbij het grappig is om op te merken dat de meest actieve actievoerders later ook de beste banen in het bedrijfsleven gekregen hebben. Het bloed kruipt toch waar het niet gaan kan.

Vindt u dat studenten anno 2010, en Delftse studenten misschien in het bijzonder, relatief braaf zijn? In andere universiteiten worden gebouwen gekraakt.

Wat er nu gebeurt vind ik nog wel mild. Vroeger had je zelfs hier hele langdurige bezettingen van het bestuursgebouw, soms weken lang. De Universiteit van Tilburg die omgedoopt werd tot de Karl Marx universiteit en de Maagdenhuisontzuimingen waren ook in die tijd. Dat ging er toch wel veel heftiger aan toe dan nu.

Waar de studenten het nu over hebben is meer een achterhoede gevecht naar mijn idee. Het is bijna niet tegen te houden dat de basisbeurs wordt omgezet in een lening. De enige vraag is of dat geld ten goede zal komen aan het onderwijs, maar zelfs dat vraag ik me af gezien de budgettaire problemen van de overheid.

Ik denk dat je tegenwoordig niet echt meer op de barricades hoeft te gaan staan. Het onderwijs in Nederland is goed geregeld en heeft een behoorlijk drempelloze toegang, zeker wanneer je dit vergelijkt met Amerika waar je enorme bedragen moet betalen. Als je wilt heb je hier bijna alle mogelijkheden om met minors jezelf te verbreden en om naar het buitenland te gaan. In die zin is het helemaal veel flexibeler geworden. De student zal misschien wat uitgestelde financiële pijn gaan voelen, maar ik denk dat op een gegeven moment de bezuinigingen in de financiering van de universiteit zelf toch wel heel hard zullen gaan toeslaan. Je ziet namelijk bijna altijd dat zo ongeveer twee jaar na een crisis de universitaire wereld een klap krijgt. Dat is na voorgaande crisis ook zo geweest. Formeel zullen ze zeggen dat ze niet gaan bezuinigen, maar dan is er altijd wel een of andere parameter waar ze aan draaien voor wat meer efficiency hier of daar. Wat ze er natuurlijk niet bij vertellen is dat het heel moeilijk te realiseren is. Een universiteit is te vergelijken met een mammoettanker en die krijg je niet zo een twee drie om. Zelfs als de subsidie gelijk blijft is dat natuurlijk al een probleem vanwege de stijgende salarissen en pensioenpremies van zo'n 1 à 2 procent per jaar.

Zal het onderwijs in Nederland kunnen blijven concurreren met de massaproductie van ingenieurs in China?

Op het ogenblik is de Chinese academische wereld nog niet echt een concurrent qua niveau. Het duurt best wel lang voordat je een kwalitatief goede onderzoeksgroep van de grond hebt, die ook echt impact heeft. Van de andere kant moeten we ons ook realiseren dat de Chinezen nog geen gelegenheid gehad hebben om enorme fouten te maken. Stel bijvoorbeeld dat Toyota, dat nu in het nieuws is, een Chinese fabriek is. Hoe zouden de Chinezen dat probleem hebben afgehandeld? Kunnen ze dat überhaupt met hun cultuur? Dat zijn de wezenlijke vragen die we moeten stellen. Om economisch succesvol te zijn is er meer nodig dan alleen productie. Je moet als bedrijf bankroet kunnen gaan. Als je dat niet goed afhandelt kun je de malaise krijgen die Japan 10-15 jaar lang gehad heeft, vanwege het niet oplossen van slechte leningen. Of China dit werkelijk kan op een schaal waarbij ze een bedreiging vormen voor



ons, dat moeten ze nog maar aantonen. Dat ze kunnen produceren geloof ik wel, maar dat is wat anders dan zelf de markt dicteren met alle gevaren van dien. Want als iets mislukt, dan mislukt er ook een heleboel. En wie gaat dan de rommel opruimen.

De invloed van cultuur op de economie is gigantisch. En dat is iets dat vaak door economen niet wordt gezien. In geen enkel economisch model komt dat terug. Japanners houden bijvoorbeeld van dingen mooi verpakken terwijl de Chinezen dat helemaal niet hebben, alles wat ze maken is even lelijk. Dat soort kleine dingen gaan uiteindelijk de doorslag geven.

Hoe vertaalt u die invloed van cultuur naar het onderwijs hier op EWI?

Je moet kijken naar waar wij als Nederlanders goed in zijn, en dat is, zeker vergeleken met Zuid-Europa, de experimentele en empirische kant van het vak. Als wij hier studenten krijgen uit Zuid-Europa dan is de theorie wel in orde, maar die andere kant niet. Het denken over hoe je systemen maakt en dergelijke zit voornamelijk in Noord-Europa. Blijkbaar is dat een deel van onze culturele bagage. We proberen dit te vertalen naar ons onderwijs hier. Je moet geen dingen gaan doen die tegen je culturele sterktes ingaan. Toch zie je dit nog te vaak gebeuren bij stimuleringsprogramma's van de overheid.

Een andere trend is het lage aantal vrouwen aan onze faculteit. Waar ligt dat aan volgens u?

Ja, dat is een hele goeie! (veert op) Ik denk dat dit voor een deel te wijten is aan de Nederlandse vrouwencultuur. Je ziet dat, zeker in Zuid-Europa, meer vrouwen informatica studeren dan hier. Ik vind dat de manier waarop de profielen op de middelbare school zijn georganiseerd een kapitale fout is geweest. Men wilde de profielen een wat maatschappelijker gezicht geven richting de beroepspraktijk en hebben er toen wat fantasienamen aan gegeven die de lading niet dekken. Voor mijn dochter ben ik ook naar zo'n voorlichtingsavond geweest en ik wist niet wat ik hoorde. "Wat kun je nou worden met dit profiel?" wordt er dan gevraagd, "Je kan levensmiddelen-technoloog worden." Is dan het antwoord. Nou, daar staan die meiden om te juichen hoor. Bovendien is het verschil tussen het profiel N&G en N&T dermate klein dat ik het niet te rechtvaardigen vind dat die twee naast elkaar bestaan. Dat heeft allemaal een negatief effect op onze instroom. Maar het ligt niet alleen aan de manier waarop het wordt aangeboden. Er is ook sprake van onderlinge beïnvloeding door meisjes. Dat is een culturele component die we bijna niet kunnen beïnvloeden en die kennelijk zo sterk is dat ze toch iets anders kiezen.

We hebben toevallig net een vrouwelijke docente aangenomen binnen onze afdeling, dus daar ben ik wel heel blij mee. We hebben nu 4 vrouwelijke PhD studenten binnen de PGS-groep, dus dat is redelijk veel.

Men zegt dat techniek minder macho is in het zuiden van Europa, waardoor vrouwen daar sneller geneigd zijn dat soort studies te kiezen. Wie zal het zeggen. Wie heeft de wijsheid in pacht.

U heeft best veel verantwoordelijkheden. Houd u nog wel voldoende vrije tijd over voor uzelf?

Genoeg hoor (lacht). Je kan wel hard gaan werken, maar als mens ben je nu eenmaal maar beperkte tijd effectief.

Ik zit nu ook in een andere positie. Zelf echt onderzoek doen, dat doe ik bijna niet meer. Anderen schrijven, terwijl ik meer in een begeleidende rol zit en, bijvoorbeeld, advies geef over welke richting een onderzoek uit moet. Het echte actieve onderzoek gebeurt eigenlijk door de promovendi en PostDocs.

Mist u het niet om zelf onderzoek te doen?

Nee, want ik vind dat andere ook heel uitdagend. En dan kan je nog steeds heel enthousiast worden als iemand met iets leuks komt.

Het is bovendien praktisch onmogelijk om zelf nog te onderzoeken. De boel moet hier ook bestuurd worden en met externe managers gaat het ook niet, dus een aantal van ons zijn nu eenmaal de 'pineut' (lacht).

Besturen vind ik ook heel leuk. Het geeft een andere kijk op de organisatie en daar heb je toch bepaalde andere vaardigheden voor nodig dan om een goed onderzoeker te zijn.

Waar bent u beter in: onderzoek of bestuur?

Ik denk dat ik op de rand balanceer. Ik doceer bijvoorbeeld ook graag, maar er kan ook een moment zijn dat ik er helemaal geen tijd voor heb. Ik vind het ook heel belangrijk dat jongere mensen zich daarin goed ontwikkelen. Voor de studenten vind ik dit belangrijk, althans voor de goede studenten. Ik zal nooit vergeten dat ik in mijn eerste jaar les kreeg van een aantal instructeurs die ik écht fantastisch vond, maar de hoogleraar waaronder ze vielen, gaf waardeloos college. Het is niet noodzakelijkerwijs een vreugde om van een slecht docent les te krijgen, terwijl men soms hier wel eens doet alsof het andersom is. Die lijn hou ik niet aan. Het is juist belangrijk dat je inspirerende docenten hebt.

Ik vind dat we al heel wat verbeterd hebben aan onze werking van nieuw jong talent. Daar heb ik bij ST toch heel veel mee te maken gehad. Nu werken we met een systeem waarbij iemand voor 5 jaar een aanstelling krijgt en na die 5 jaar kan iemand doorgroeien, of gaat weg. Misschien een beetje rigide, maar het werkt. We hebben hele goede mensen aangetrokken. Dat is ook te zien in de resultaten van de onderzoeksvisite die we recentelijk gehad hebben en voor onze afdeling uitstekend verlopen is. Dat vind ik wel iets om trots op te zijn.

Ik probeer dat coöperatieve model ook een beetje in onze afdeling vorm te geven, maar verbetering blijft mogelijk.

Doet u aan sport?

Ja, ik heb jarenlang gevolleybald, ik zeil veel en doe ook aan tango dansen. Afgelopen november was ik voor de tweede keer in Buenos Aires. Écht een leuke stad! De tango is best een moeilijke dans en misschien wel een uitdaging voor technici, want als je wat van mechanica, bewegingsleer en geometrische patronen afweet kun je daar jezelf een beetje in uitleven. Dat vereist natuurlijk niet alleen wat coördinatie van jezelf, maar ook van je partner.

Hartelijk dank voor dit gesprek!

Woonplaats: Delft

Muziek: vroeger klassieke gitaar gespeeld

Boek: "Een verhaal van liefde en duisternis" Amoz Oz

Film: "The night Porter"

Favoriete drank: wijn

Favoriete OS: Mac OS X

Vakantieland: Frankrijk en Italië

Quote: "Ga doen wat je echt leuk vindt"

Aantal kinderen: 2



Developing the TU Delft 'My Timetables' for Blackboard application

Tom Verhoeff & Marco Krikke

Back in August 2009 the Delft University of Technology switched to a new application for building and maintaining schedules. This application, called Syllabus+, provided the schedule makers with an easy way of scheduling activities more efficiently across the whole campus. A new interface came for the system which is known as *timetables.tudelft.nl*. During the first weeks some startup problems occurred, but when all the problems were solved the system worked fine. Unfortunately 'worked fine' does not mean it works the way you want it too. While O&S was happy with all the new features, students were left with an interface not suiting their needs.

As chair of the faculty student council for the faculty EEMCS, I was one of the first to receive complaints from students. Students of our faculty know how to complain when the university delivers an imperfect IT system. Complaints varied from "I'm not able to get my elective courses in my schedule", "why do I have to select everything over and over again to check for changes" to "my phone uses a digital calendar so iCalendar export should be available". Workarounds by different students showed up within a few days, but none of them really worked perfect. All the complaints led to a commitment by the TU that a new, better interface would be available at the start of the second semester. TU Delft application developer 3x0 unfortunately lacked the manpower to build a new system in time, so in October word reached us that they were looking for students to get the job done. That's when I teamed up with Marco Krikke, Mike Noordermeer and Maarten van der Beek to build a new schedule interface from scratch, based on student needs.

Constraints

Before we even started working on the project, it was necessary to define the scope of the project. First of all we were required to integrate our new interface into BlackBoard. This forced us to make use of Java, and one of us had to study BlackBoard documentation, since nobody had any BlackBoard programming experience. All the data that we wanted to store also had to be saved in BlackBoard. All the timetable data was provided in a Microsoft SQL Server database. Basically our challenge was to build a Java based building block for Blackboard, which takes data from a MSSQL database and presents it to users in an easy and understandable format.

Requirements

The constraints did not seem to be anything more than a challenge, so we moved on to gathering requirements. Since we now had the opportunity, we were eager to fix the problem of the schedule interface once and for all. Mike and Marco had already assisted our faculty in building a list of requirements for the old Webber system, so we had something for a start. After interviews with students, O&S, 3x0, Oras and educational commissioners from the study

societies, a full list of requirements was created. I will not bother you with a full list of functional requirements, but summarized our system would provide the following features: add courses to a profile based on course name, course code, study program, BlackBoard enrollments, student number (for Architecture students) and staff member (for staff members). We went to a meeting with 3x0 with a complete requirements analysis document, and after some negotiating we came to an agreement. Now we could finally start with the real work.

Architecture

First the basic architecture of the system had to be defined. Since everything had to be done in Java, starting with Hibernate as an Object-Relational Mapper was an easy decision. The interface itself had to be clear, easy to use and highly interactive. The Google Web Toolkit (GWT) offers all the functionality to make this possible. The combination of GWT and Hibernate is the technical basis for the application. Some other libraries were also added to support common used functionality. Among this list are log4j for easy logging; ical4j, opencsv and itext for generating the different export files and some BlackBoard libraries to support the BlackBoard integration. We will discuss all the libraries in more detail later on.

Database

All timetable data is provided in a MSSQL database generated by a brand new module for Syllabus+. In the server layer of our application, Hibernate is used to map information from the database to usable Java-object. Due to the complexity of the database, performance proved to be an issue during development. The first basic implementation of Hibernate performed okay only for small data requests, but after some optimizing this also worked for larger data sets, such as full schedules. Performance issues returned when we implemented support for different student sets per course later on. The first time we saw a `java.lang.OutOfMemoryError` it came as a surprise, but after some new rounds of optimizing the whole ORM-layer performed good enough. Some tweaks to the Hibernate configuration ensure fast en reliable performance with a higher user load.

User Interface

From a user point of view everything has to work fast and intuitive. Nobody wants to spend time learning how to create its personal timetable. GWT does not only provide easy to use and recognizable interface objects, but also a full framework to manage the client server communication. Communication is done with asynchronous remote procedure calls to the server, possibly resulting in so called delta updates at the UI. Delta updates only update certain elements of a page, so the page does not have to be completely refreshed after each user action. Development is done in Java, but all code is compiled to JavaScript by a special GWT compiler.

SmartGWT, an extension framework to 'normal' GWT, seemed to provide the best set of interface features for our application. After some implementation had been done everything seemed to work, but stability and fast performance proved to be difficult. Switching back to normal GWT was the only option left. This clearly improved the responsiveness of the whole interface but some of



the features had to be manually implemented, such as the graphical calendar view. Implementing the whole interface took more time than expected, but in the end everything worked out fine.

Since GWT compiles HTML and JavaScript for different sets of browsers to overcome rendering issues, our interface had to be tested for browser compatibility. The newest versions of all major browsers are able to handle the JavaScript without a problem, except for some strange layout bugs, but some of the older browsers proved to be a problem. Everybody knows JavaScript is not processed quickly in IE6, but getting a cup of coffee is the least you need to do when trying to run a GWT application in IE6. Due to some strange problems, GWT also recognized IE7 as IE6 when we first tested browser compatibility. Every TU Delft computer still runs IE7 so decent support was required. Fortunately some workarounds proved to be good enough to get everything going in IE7, but still IE8 is recommended.

Blackboard

Running a GWT application in BlackBoard did not require many adjustments. Since BlackBoard still uses frames, all you have to do is put the GWT application in its own frame. All TU Delft BlackBoard users will recognize the big top frame with the TU Delft logo. This causes no problems during normal usage, especially since the top frame is replaced by a small one when browsing course pages. The same small top frame is desirable when displaying someone's schedule on screen to use the most amount of screen space available, but this functionality is not offered by BlackBoard. It is always big, always small or only small when browsing courses. Clearly the last option applies to the TU Delft implementation. By using a nasty JavaScript hack, we are still able to resize the top frame, but this is something you should not even want to use in a serious application.

BlackBoard has some more 'unexpected' features. Let's take a look at the user authentication for example. Presenting someone's timetable is only possible when a user is logged on, so we called the `BbSession.isAuthenticated()` method before doing anything else. Soon enough we discovered this method returns `True`, even

when the user did not log on yet. In that case BlackBoard sees an user 'authenticated' as guest in that case. Solving this issue was done getting the required information from the BlackBoard 'context' (ctx) by using this code:

```
if (ctx.hasUserContext()) {
    User user = ctx.getUser();
    User.SystemRole role = user.getSystemRole();
    isLoggedIn = role != User.SystemRole.GUEST;
}
```

We came across another surprise when implementing the 'import from BlackBoard' feature for adding courses to your profile. Getting a list of courses a user is currently enrolled in takes more time than you would expect. Every course in Blackboard is defined as a Course object. Course and User are connected through a CourseMembership object. To get a list of active BlackBoard courses retrieving all CourseMemberships for a user should be enough. Unfortunately, this list contains all the courses a user has ever been enrolled in. Also a difference has to be made between 'normal' courses and 'communities'.

Export

While implementing the export functionality, we also came across some unexpected problems. For example generating an iCalendar file from a list of activities does not require rocket science. Unfortunately, it does when support for both Outlook 2003 and Google Calendar is necessary and you don't happen to be in the UK. Specifying a time zone for the iCal file does not work with Outlook 2003, but not specifying it makes Google Calendar think it is GMT instead of GMT+1. The only way to get this to work is defining a time zone per activity instead of per file.

Generating decent PDF output also proved to be difficult. Using the iText library for a clean text output is no problem at all, but generating a decent graphical week view is difficult. We do know the current graphical PDF export is not perfect yet, we will improve it if the TU requests an updated version.

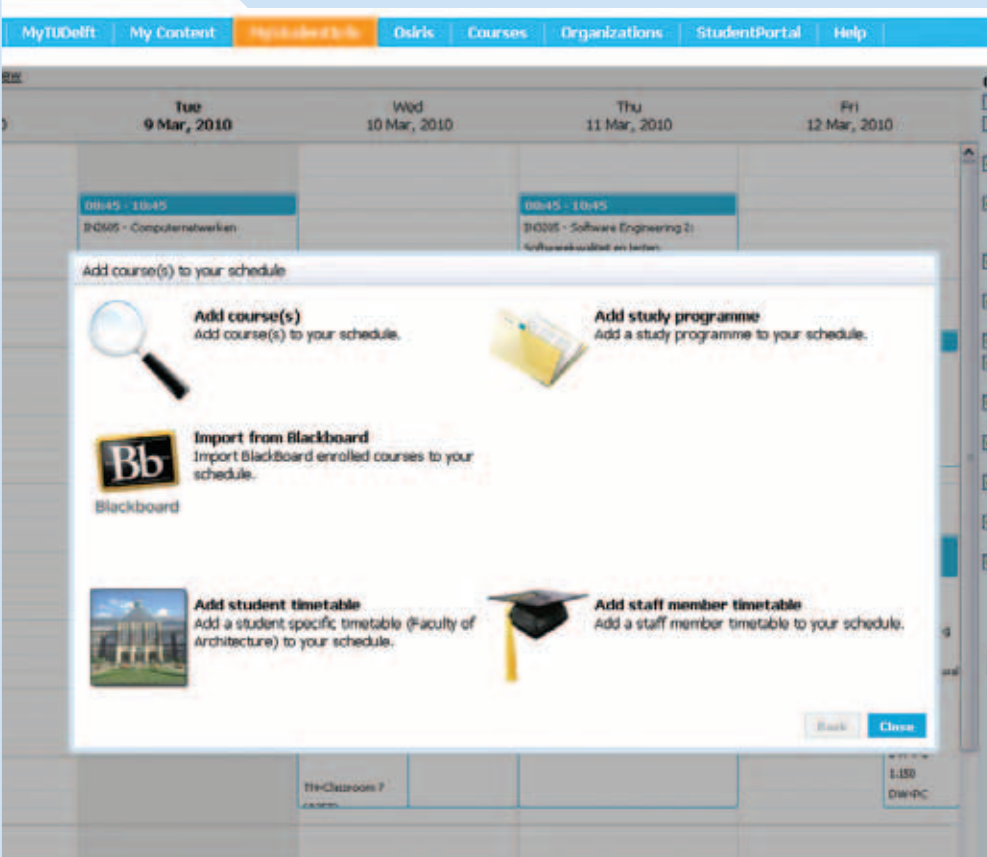
Testing

When the final release came near, testing became more important. Fortunately, a lot of students helped us testing the system to ensure it complied with all student needs. We have been able to solve a lot of the minor problems reported, but testing also brought us a list of 'like to have'-features for possible future releases. We do not plan on abandoning the project anytime soon, so when the first weeks of usage are a success, we will definitely look into some new features. Some of these features require adjustments require action from O&S, others might be implemented in a future release.

Conclusion

Developing the MyTimetable application has been quite an experience. Like all IT projects, we came across some unexpected problems, but in the end everything works fine. Almost all feedback we received has been positive so far, but we also got some interesting new ideas for possible future updates. All comments and suggestions are still welcome at roostercie@ch.tudelft.nl. Want to thank us for the new system? You can find us in the /Pub every Wednesday starting at 4.

Screenshot of the interface of the new lecture schedule



Tomorrow, we will be able to make chips faster. Today, you can tell us how.

Deep UV-light
(193 nm)

The race to increase the number of IC switches per square centimeter is not the only race that is underway in the chip world. Manufacturers are also aiming to accelerate chip production. But how do you boost a machine that needs to be accurate to the nanometer?

ASML is now working on chip lithography systems in which a disk of photo-sensitive silicium (the wafer) is illuminated at high speed.

The wafer lies on the so-called wafer stage, which weighs more than 35 kilos. It is passed back and forth under the light, with an extreme acceleration and deceleration of 33 m/s^2 .

33 m/s^2



Chips with 45 nm details can only be made if, between acceleration and deceleration, you illuminate the wafer precisely to the nanometer. One thousand sensors and 800 actuators control and, consequently, illuminate 180 wafers an hour. How much software and how many processes are required to do this? And how do you manage the necessary IT architecture?

Accelerating by 33 m/s^2 poses a challenge in itself. Which motors do you choose? Where do you find amplifiers with 100kW capacity, 120 dB SNR and 10 kHz BW? And that is just the beginning – because the heat itself distorts the accuracy of your system as well...

For engineers who think ahead

Profile: Worldwide market leader in chip lithography systems | Market share: 65% | R&D-budget: EUR 500 million | Opportunities for: Physicists, Chemists, Software Engineers, Electrotechnicians, Mechatronics and mechanical engineers | Discover: ASML.com/careers



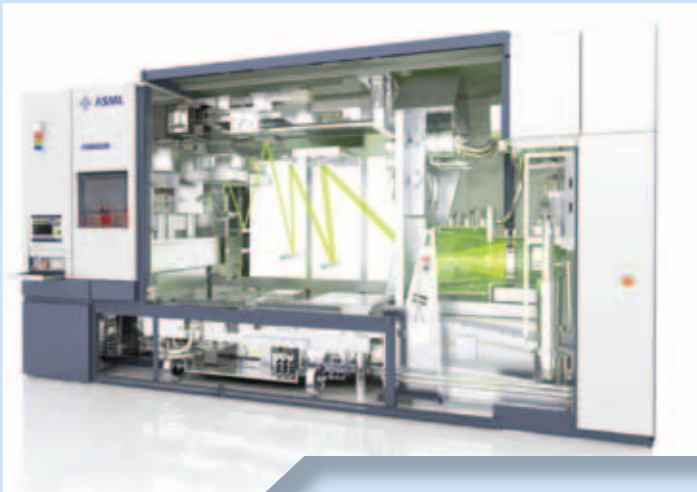
ASML



ASML

Make tomorrow today

Semiconductor technology is an essential part of our lives today. Your iPhone, your digital camera, your MP3 player, your car, your passport; they all contain chips. And that's just today. If you want to know how high tech our lives will be tomorrow, ask ASML; global market leader in chip production systems. Two-thirds of all the chips in the world are already manufactured on ASML machines. Our challenge is to develop machines which can produce even smaller chips even more quickly and more accurately. In short: technology which can further enhance the quality of our high tech lives. Do you want to help with that at the very highest level? Then there is only one employer in the world who matches your ambitions: ASML.



Steep curve

ASML is a relatively young company, which has developed into the global market leader at an unprecedented speed. In the space of twenty years we have grown from nothing into a multinational with more than 50 sites in sixteen countries and annual sales of 1,596 million euro in 2009. We set the international standard with the development of wafer steppers and scan systems which are used for chip production. In order to retain and further expand that leading position we have thousands of technicians exploring the frontiers of (nano) technology every day. That requires people who also constantly explore their own limits and try to push them further. Who consciously opt for the steepest learning curve.

Technological race

ASML works with an almost impossibly short time-to-market, driven by our customers' competitive market conditions. It is precisely that technological race against the clock that makes working here so utterly fascinating. Cutting-edge technology in the field of precision engineering, dynamics, optics, electronics and information technology all comes together within our company in order to develop systems which are more reliable, faster and more accurate than their predecessors in the shortest possible time.

It is only our very unusual development method whereby a large number of development processes are running in parallel that enables us to constantly meet the very ambitious demands set by the market and ourselves.

Drive to develop

Working at ASML requires a special attitude. An unbridled passion for technology is vital, but not enough. Having commercial insight, team spirit and being result-oriented are all equally important, since what makes ASML successful is the interplay between precision engineering, dynamics, optics, electronics and information technology. But perhaps the most important factor is a drive to develop which is deeply embedded in all our people. ASML invests a great deal in your development. For us that is a given. In return we also expect it of you. Since anyone who is working on creating tomorrow must take his talents seriously today.

Do you have a technical education up to graduate or hbo (higher vocational education) level? And do you want to help build a leading international high tech company tomorrow?

Then you know what you need to do today. Visit www.asml.com/careers

Advertorial





Chocola

Informaticapuzzel XXII

Dr. P.G. Kluit

Op 5 februari 2010 vond in Harbin China, de finale plaats van de jaarlijkse ACM-ICPC programmeerwedstrijden. Zie ook:

<http://cm.baylor.edu/welcome.icpc>. Het was erg koud in Harbin.

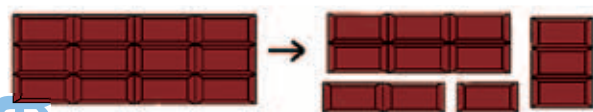
De volgende opgave behoort tot de minder moeilijke opgaven van de wedstrijd. Maar dat betekent niet dat hij gemakkelijk is.

Je hebt een groot stuk chocolade, en je wilt dat delen met je vrienden. Helaas zijn je vrienden nogal kieskeurig, ze weten precies hoeveel chocolade ze willen hebben. Het wordt steeds moeilijker om na te gaan of je wel aan alle wensen kunt tegemoetkomen. Het wordt tijd om een programma te schrijven om dit probleem eens en voor al op te lossen.

Je hebt een rechthoekig stuk chocolade, verdeeld in rechthoekige blokken die allemaal even groot zijn. Om de chocolade te verdelen breek je het stuk in twee delen, tussen twee rijen of tussen twee kolommen. Vervolgens kun je de twee overgebleven stukken op dezelfde wijze verdelen. Elk van je vrienden heeft aangegeven hoeveel blokjes hij wil hebben, en ze willen dat ook allemaal in een stuk hebben. Je wilt het stuk eigenlijk alleen verdelen als het geheel onder je vrienden verdeeld kan worden, zonder dat er iets over blijft.

Hieronder zie je hoe je een stuk chocolade van 3×4 blokken kunt verdelen in 4 stukken, die respectievelijk 6, 3, 2 en 1 blok bevatten. Je moet hiervoor drie maal breken.

Invoer



De invoer bevat verscheidene testgevallen.

Elk testgeval bestaat uit 3 regels. De eerste regel bevat het getal n ($1 \leq n \leq 15$), het aantal stukken waarin de reep moet worden verdeeld. De tweede regel bevat twee gehele getallen x en y , ($1 \leq x, y \leq 100$), de afmetingen van de reep. De laatste regel bevat n positieve gehele getallen, de grootte (het aantal blokken) van elk van de n delen waarin de reep moet worden verdeeld. De laatste regel van de invoer bevat het getal 0.

Uitvoer

Geef voor elk testgeval aan (door het afdrukken van "Ja" of "Nee"), of de gevraagde verdeling mogelijk is.

Inzendingen kunnen verstuurd worden naar machazine@ch.tudelft.nl tot en met 16 april.

Oplossing Informaticapuzzel XXI

Hamburger

Dr. P.G. Kluit

Het Hamburgerprobleem van de vorige keer heeft veel toetsenborden in beweging gezet.

We ontvingen oplossingen van Jeroen Wille, Thijmen Krebs, Misha Stassen, Robert Stuursma (allen in Java), van Charlotte Ipema (Haskell) en van Maurice Bos (C++). En dan was er ook nog een oplossing (buiten mededinging) van de hand van Cees Pronk, die de 'statistical model-checker' Prism gebruikt, en daarmee redelijke resultaten bereikt, voor 80 of minder kinderen.

Thijmen Krebs komt met de volgende formule, die de meeste andere oplossers expliciet of impliciet ook gebruiken ($2n$ is het aantal kinderen):

$$p = P(\text{laatste twee gelijk}) = 1 - P(\text{laatste twee ongelijk}) = 1 - C(2n-2, n-1) / 2^{(2n-2)}.$$

Hier staat $C(n, m)$ voor het aantal mogelijke combinaties om m elementen uit n te pakken (de binomiaalcoëfficiënt). De kans dat de laatste twee ongelijk zijn is gelijk aan de kans dat je bij $2n-2$ maal werpen met een zuivere munt precies even vaak kruis als munt krijgt.

Blijft de vraag hoe we nauwkeurig en efficiënt de expressie $C(2m, m) / 2^{(2m)}$ kunnen berekenen (met $m = n - 1$). Als je eerst alles vermenigvuldigt, of eerst alles deelt wordt de berekening al snel onnauwkeurig, of loopt stuk op factoren die 0 of oneindig worden.

Op het punt van nauwkeurigheid springt de oplossing van Thijmen Krebs er uit. Hij gebruikt zijn eigen klasse BigInt, met een vermenigvuldiging gebaseerd op de Fast Fourier Transform. Rond $n = 10^6$ wordt het programma traag, maar geeft wel antwoorden in ruim een miljoen cijfers achter de komma. Omdat mijn eigen oplossing met doubles werkt, kan ik de laatste $10^6 - 15$ cijfers niet controleren.

Minder nauwkeurig, maar wel efficient is de oplossing van Micha Stassen. Hij weet over- dan wel under-flow te voorkomen door in elke stap ongeveer evenveel te vermenigvuldigen als te delen. Ook rond $n = 10^9$ levert het programma van Micha binnen enkele seconden antwoorden die tot op de laatste decimaal nauwkeurig zijn (althans volgens mijn antwoorden).

Dit is zijn code:

```
double calcCompact(int n){
    double q = 1;
    for (int i = 2; i <= n; i++){
        double mul = 2*i - 2;
        double deel = i - 1;
        q *= mul*(mul - 1.0)/
            deel/deel/4.0;
    }
    return 1 - q;
}
```

Het kan zelfs nog iets compacter; de volgende code doet hetzelfde:

```
double calcCompacter(int n){
    double q = 1;
    for (int i = 1; i < n; i++){
        q *= (2.0*i - 1)/i/2;
    }
    return 1 - q;
}
```

Overigens kun je door de formule van Stirling voor $n!$ te gebruiken een benadering voor $C(2m, m)$ afleiden. De volgende code levert voor grote n heel snel goede benaderingen.

```
double burger1(int n){
    return 1 - (1.0/Math.sqrt(Math.PI * (n-2)/2));
}
```

Rest ons een winnaar aan te wijzen. Dat betekent de snelheid en compactheid van Micha Stassen af te wegen tegen de doorwrochtetheit en nauwkeurigheid van Thijmen Krebs.

Laten we niet kiezen, en een gedeelte eerste plaats toewijzen.



Artificiële intelligentie

Computers en emoties

Joost Broekens, MMI, Informatica

Computers en emoties hebben zo op het eerste gezicht weinig met elkaar te maken. Toch wordt er de laatste tijd veel onderzoek gedaan op het gebied affective computing, in Delft, in Nederland en internationaal. Dit onderzoek spitst zich toe op het gebruik van emoties in computersystemen om deze systemen beter te laten samenwerken met mensen en om deze systemen slimmer te maken. Wij mensen communiceren namelijk graag door middel van emoties, en we gebruiken emoties om ons gedrag en dat van anderen te beïnvloeden.

Onderzoek naar emotionele computersystemen neemt steeds grotere vormen aan, en er zijn tal van mogelijkheden om dit onderzoek in concrete producten te gebruiken. Mogelijke producten zijn bijvoorbeeld adaptieve games die proberen om de spelervaring leuker te maken, webwinkels die emotionele feedback gebruiken om producten beter te kunnen matchen met potentiële klanten, en “emotionele” robots zoals de iCat (Philips) die op een meer natuurlijke manier met mensen communiceren. Deze producten zijn zeer de moeite waard, en het onderzoek is uitdagend. Emotie is iets menselijks, soms nuttig, soms niet, en onze apparaten zullen dat uiteindelijk gaan begrijpen. In dit stuk vertel ik eerst over het vakgebied affective computing, en geef ik daarna twee voorbeelden van mogelijke producten.

Kunstmatige Intelligentie en affective computing

Het vakgebied kunstmatige intelligentie houdt zich bezig met hoe computers geprogrammeerd kunnen worden zodat ze zelfstandig problemen kunnen oplossen, of, zelf kunnen leren hoe ze problemen moeten oplossen. Denk bijvoorbeeld aan een stofzuigrobot die leert hoe je huiskamer er uit ziet, zonder dat je dat hoeft te vertellen. In dit geval is de robot niet geprogrammeerd op de vorm en inhoud van jouw huiskamer, de robot leert die huiskamer als een soort stofzuigplaattegrond. Een ander, klassieker voorbeeld, is het medische expertsysteem. Een dergelijk systeem heeft kennis over bepaalde ziektes (symptomen, reacties op medicijnen, e.d.) in de vorm van een grote verzameling regels. De arts kan informatie over een patiënt invoeren in dit systeem en vervolgens kan het systeem de arts helpen bij het analyseren van patiënten.

De “heilige graal” van kunstmatige intelligentie is tweeledig. Door computersystemen intelligent te maken kunnen zij bepaalde taken, zoals stofzuigen, van ons overnemen: een vorm van geavanceerde automatisering. Ten tweede is de basisaanname in de kunstmatige intelligentie dat het mogelijk is om menselijke en dierlijke intelligentie te simuleren met computers. Door dit te proberen kunnen we dus veel leren over hoe mensen en dieren denken en leren te handelen.

Vanuit een heel andere discipline, de mens machine interactie, is het besef ontstaan dat onze technologie goed met ons moet kunnen communiceren. Het is al heel lang bekend dat emoties een belangrijke rol spelen bij sociale interactie en communicatie tussen mensen. Verbale communicatie wordt meestal aangevuld met belangrijke niet verbale cues. Aan het einde vorige eeuw werd ook steeds duidelijker dat emoties een belangrijke rol spelen bij het kunnen maken van rationele beslissingen. Daarvoor werd vaak aangenomen dat emotie en ratio twee gescheiden werelden waren en dat emotie de ratio voornamelijk in de weg zat. Maar neurologisch onderzoek, bijvoorbeeld door Antonio Damasio, toonde steeds vaker aan dat de capaciteit voor het hebben, tonen en herkennen van emoties sterk gekoppeld was aan het kunnen maken van rationele beslissingen. Het is alsof het vermogen tot normaal emotioneel functioneren een basisfunctie is waarop rationeel denken gebouwd is. De ratio gebruikt als het ware de processen in onze hersenen die verantwoordelijk zijn voor de verschillende emoties.

Het besef dat computers en andere apparaten beter met mensen zouden moeten communiceren en de resultaten van onderzoek naar emoties en denken waren de doorslaggevende factoren in het ontstaan van het vakgebied affective computing. Dit gebied houdt zich bezig met hoe computers gebruik kunnen maken van emoties ter verbetering van de mens-machine interactie en ter ondersteuning van de kunstmatige intelligentie.

De eerste serieuze pogingen om na te denken over hoe computers emoties kunnen simuleren, en waarom dit nuttig zou kunnen zijn, komen uit de '80 jaren van de vorige eeuw. Wetenschappers zoals Aaron Sloman begonnen na te denken over hoe emotie gekoppeld kan worden aan kunstmatige intelligentie. Ook werd het vanuit de psychologie steeds duidelijker hoe emoties konden ontstaan uit denkprocessen. Een zeer invloedrijk boek geschreven door Andrew Ortony, Gerald Clore, en Allen Collins beschreef in detail de logische structuur van emoties en hoe verschillende emoties ontstaan uit de evaluatie van gebeurtenissen in relatie tot persoonlijke motieven en doelen. Een voorbeeld:

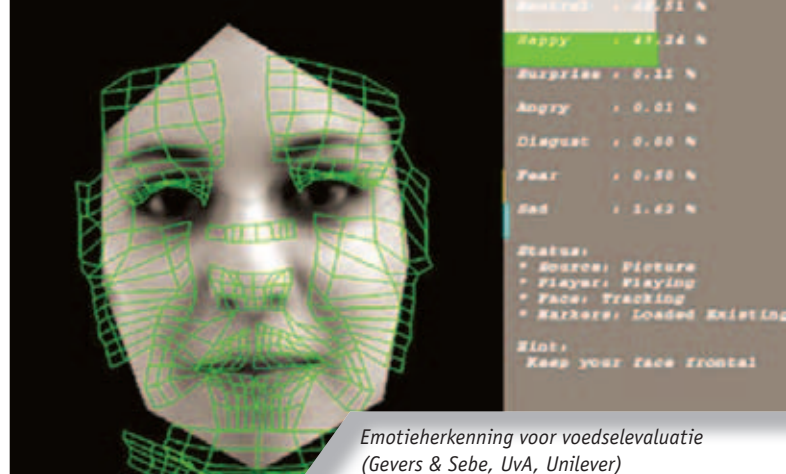
Informatica

als ik wil eten en ik sta in de file omdat iemand voor mij op domme wijze een aanrijding heeft veroorzaakt dan evalueer ik dit als volgt: "doel: eten, gebeurtenis: aanrijding, gevolg: niet eten, oorzaak: persoon voor mij, emotie: boos".

Er werd op dat moment in de KI wereld ook veel onderzoek gedaan naar agents; autonoom beslissende programma's die een eigen set van motieven en doelen hebben en zelf deze doelen moeten proberen waar te maken. Het "brein" van deze agents bestond uit doelen, feiten en intenties. Hieruit beslist de agent wat de volgende actie wordt. De structuur die voor emotie nodig is leek sterk op die van een agent: doelen, kennis over de wereld en intenties tot handelen. Het waren dan ook dit soort agents waarin als eerste op succesvolle wijze een grote diversiteit aan emoties werd gesimuleerd. In het begin waren deze emoties voornamelijk een bijverschijnsel: ze deden niets maar ontstonden gewoon uit het denkproces van de agents. Ontstaan betekent in dit geval dat uit een grote set regels, en op basis van een psychologische theorie een bepaalde emotie resulteert..

Latere aanpakken verschilden van de agent-georiënteerde aanpak. Men ging bijvoorbeeld kijken naar hoe een computer getraind kan worden op herkenning van menselijke emoties uit gelaatsuitdrukkingen en stem. Ook deden onderzoekers, zoals Cynthia Breazeal van het MIT in Boston, pogingen om een robot te bouwen die dit soort modellen hadden, waardoor deze robots op natuurlijker wijze met mensen om konden gaan. Deze robot, Kismet, herkent emotie aan de hand van stemgeluid en heeft een uitgebreid model van wat prettig en niet prettig is (Kismet schrikt bijvoorbeeld van je als je snel je hoofd voor Kismet's gezicht houdt). Kismet heeft ook een eigen emotie, die ontstaat uit alles wat er met Kismet gebeurt, en Kismet uit deze emotie via het gezicht. Doordat Kismet zo geloofwaardig met het herkennen en uiten van emoties omging kon er onderzoek gedaan worden naar de rol van emoties in de relatie tussen robot en mens. Vragen zoals "hoe leren we robots gedrag aan?" en "hoe reageren mensen op een angstreactie van een robot" konden nu gesteld worden.

Samengevat zijn er een aantal trends ontstaan uit emotionele agentgebaseerde modellen en interactieve emotionele robots en andere systemen. Ten eerste is het nu mogelijk om onderzoek te doen naar de rol van emotie in mens-machine interactie. Ten tweede is het mogelijk om onderzoek te doen naar de relatie tussen gesimuleerde emotie en kunstmatige intelligentie. Ten derde, omdat het "brein" en de "emoties" van deze agents ontworpen worden op basis van



Emotieherkenning voor voedselbeoordeling (Gevers & Sebe, UvA, Unilever)

psychologische theorieën kunnen de resultaten van de simulaties gebruikt worden om de voorspellingen van de theorieën te testen. De agent is in dit geval een soort simulatie van de onderliggende theorie.

Systemen met emoties in de praktijk

Afgezien van interessant onderzoek, is affective computing ook een bron van inspiratie voor de ontwikkeling van innovatieve producten, zoals de recente "Smile shutter" technologie voor het automatisch maken van een foto als iedereen op de foto lacht. Dit stuk eindigt met twee andere voorbeelden. De rol van affective computing in deze producten is voornamelijk het verbeteren van de interactie tussen mens en machine.

Het emotionele robot huisdier is een product dat bestaat, bijvoorbeeld in de vorm van de robot hond Aibo of zeehond Paro. Deze "diertjes" zijn ingezet in verpleeginstellingen voor de verbetering van de kwaliteit van leven. Het blijkt dat bejaarden het inderdaad een leuk ding vinden, en graag met het "diertje" in contact zijn. Natuurlijk zitten er allerlei haken en ogen aan het op grote schaal inzetten van dit soort robotjes ter ondersteuning van de sfeer in verpleeg- en bejaardentehuizen. Het toont echter wel aan dat er een mogelijkheid is voor dit soort producten, en, niet onbelangrijk, dat leeftijd niet noodzakelijkerwijs een probleem is. Een eenvoudige toevoeging zou kunnen zijn dat het robotdiertje niet alleen emotioneel gedrag vertoont, maar ook emoties kan meten van de gebruiker. Dit vergt nog onderzoek, maar is zeker niet onhaalbaar. Het robot huisdier kan nu gebruikt worden als observatie-instrument, en kan dan gebruikt worden als interface tussen cliënt en zorgverlener.

De emotionele webwinkel is een concept waarbij niet alleen op basis van eerder gekochte artikelen een suggestie voor een nieuw te kopen artikel gedaan wordt, maar ook op basis van de stemming van de persoon of het emotionele profiel. Stel, de gebruiker geeft aan hoe hij/zij zich voelt op het moment van het kopen van een artikel, en wat hij/zij vindt van het product. Met deze informatie kan de webwinkel een model opbouwen van de koppelingen tussen producten en emoties. Een simpele extra service kan nu zijn dat de webwinkel een suggestie doet voor een product bij het inloggen. De gebruiker klikt aan hoe deze zich voelt, en de webwinkel koppelt dit aan een verzameling producten. De onderliggende technologie voor het meten en interpreteren van emoties en het opbouwen en matchen van emotionele profielen is aanwezig.

Er zijn vele mogelijkheden voor het verbeteren van de interactie tussen mens en techniek. De technieken die door affective computing worden aangeleverd zijn daar voorbeelden van. Het is mogelijk en nuttig om deze technieken toe te passen. Natuurlijk zijn er ook risico's en problemen. Wat te doen met, bijvoorbeeld, privacy? Wie krijgt mijn emotionele profiel allemaal te zien? Ook bestaat het gevaar dat de software verkeerde conclusies trekt uit de emotionele informatie van de gebruiker. Hierdoor kan de service of het product niet goed meer functioneren. Als systemen goed beveiligd zijn en er geen automatische, onomkeerbare beslissingen door worden genomen zijn de problemen veelal overzichtelijk. Affective computing maakt technologie menselijk, robots sociaal en games extra spannend.

Meer info: <http://mmi.tudelft.nl/~joostb> en het boek Affective Computing (Picard)



Aibo (Sony), Paro (ISRI-AIST) en iCat (Philips), entertainment en companion robots.



Zero emission racing!

Koen Lubbers, Teammanager Greenchoice Forze

Burning rubber and still being sustainable? This seems to be contradictory, but it is possible by using hydrogen fuel cell technology.

Greenchoice Forze participates in the Formula Zero Championship, the racing class of the future. This racing class is all about high speed racing while producing no other emissions than pure water!

The greenhouse effect is one of the biggest problems of the 21st century. A big contributor to the greenhouse effect is the automotive industry. Though, millions of people enjoy motorsport and mobility in general every day. Wouldn't it be great if motorsport and sustainability would go hand in hand?

Formula Zero

In 2003, a couple of engineers from Delft founded Formula Zero BV. The philosophy of Formula Zero is 'fun + eco = future'. Motorsport has always played a pioneering role in developing new technology, so it is a great way to give the development of hydrogen technology a boost.

Formula Zero is the driving force behind the 2010 Formula Zero Cup, a championship for hydrogen fuel cell powered karts. This is a zero emission racing class, where the only exhaust is pure water. The 2010 Formula Zero Cup will be the third season of hydrogen races.

The championship consists of four race events. Each race event consists of a sprint race and a main race. For each race, there is qualifying round, a semi-final and a final. Last year, the races were organized all over Europe. The start of the season was in England. After England, the Netherlands, Belgium and Spain were also visited. This year, most likely, there will be races in the USA too.

The Formula Zero Championship started as a racing class for hydrogen karts. Formula Zero's vision on the future is to scale up the competition. The first step is to grow into a hydrogen Formula 3 racing class. Later on, the objective is to become a hydrogen Formula 1 racing class!

As an official closing of the second season, Formula Zero organized a GP in Turin on the 10th and 11th of October 2009. Even after crashing into the barriers during the second qualifying round, Greenchoice Forze, the Formula Zero team from Delft, managed to win the GP!

This year, Greenchoice Forze will organize the Dutch race in cooperation with Formula Zero. It will be a race in the centre of The Hague, at the Spuiplein. The race will take place on July 8th and be part of a big sustainability event!

Greenchoice Forze

Greenchoice Forze wrote history by winning the first hydrogen race ever! On the 23rd of August 2008, the team defeated five teams from international top universities. The Formula Zero team from Delft is also the first world champion ever.

Greenchoice Forze was founded in 2007, to show the world that sustainability and motorsport do not have to be contradictory. The team wants to showcase this to a broad audience by racing on hydrogen! Except for racing, Greenchoice Forze also visits a lot of fairs and events to showcase the potential of the sustainable technology already available.

The team consists of about 50 highly motivated students from every faculty of the TU Delft. Besides designing and engineering the kart, the team members also take care of management and marketing of the team.

The current kart

Normally a kart has a gasoline tank, a small combustion engine and an exhaust. The Forze II, however, is far more complicated. Instead of using a simple combustion engine that runs on gasoline, the Forze II has two DC electro motors which are powered by a proton exchange membrane (PEM) fuel cell. The gasoline tank is replaced by a 5L tank in which hydrogen is stored at a pressure of 200 bars.

Basically the Forze II works as follows: Hydrogen is fed to the fuel cell. In the fuel cell the hydrogen and oxygen, taken from the ambient air, react and produce electricity. The voltage produced by the fuel cell is then brought to a higher level by using a DC/DC boostconverter. Afterwards the electric energy is stored in boostcaps. When the driver kicks down the pedal, the electric motors subtract their energy from these boostcaps.

*Award ceremony Formula Zero GP Torino
Copyright: Giorgio Giroto*





Testing in the wind tunnel

The fuel cell consists of several devices, of which the most important one is the stack. In the stack, the hydrogen (H_2) and oxygen (O_2) are separated by a membrane. At the anode side, a catalyst splits the hydrogen molecules into protons and electrons ($2\text{H}_2 \rightarrow 4\text{H}^+ + 4\text{e}^-$). The protons can go through the membrane to the oxygen at the cathode side, but the electrons cannot and have to go through an external circuit, resulting in an electrical potential difference. The reaction on the cathode side is then as follows: $4\text{H}^+ + 4\text{e}^- + \text{O}_2 \rightarrow 2\text{H}_2\text{O}$. Because the potential difference is low, multiple cells have to be connected in series to get some decent voltages. All those cells stacked together form the stack.

The voltage coming from the fuel cell is not high enough to charge the boostcaps, so a DC/DC converter is implemented in the kart. It converts the voltage coming from the fuel cell to the desired voltage for the ultra capacitors. This device is rather complex, because the potential over the boostcaps depends on the amount of energy stored. Another important property of the DC/DC converter is that it works as a diode. The DC/DC converter makes it impossible to have a current flowing back into the fuel cell.

For the energy storage system, 32 Maxwell Boostcaps are used. A boostcap is an ultra capacitor, a device which can store and release a lot of energy really fast. The capacity of one boostcap is 3000 Farad, which is incredibly high compared to capacitors used in electronics (in the order of micro or even picoFarads).

The two motors used are Perm 132 DC electro motors. These motors are not connected directly to the boostcaps, because a throttle is needed. They are controlled by a separate motor controller for each motor. This way an electronic differential is possible. We are currently developing a motor controller which can control both motors, but is half the weight and just a third of the size of the two off-the-shelf controllers. This controller will also have a lot more features, like a reverse mode.

All this technology has to be protected by bodywork. This is made of a special composite, which is far more environmentally friendly than conventional composites. A natural fiber, flax, is used for the bodywork of the Forze II in

The next kart

The Forze III, the hydrogen kart which is currently being developed, will be totally different than the Forze II. Since the new regulations permit the teams to buy or develop a fuel cell system and hydrogen storage system of their own, Greenchoice Forze can finally start innovating the hydrogen technology itself!

Because of the new fuel cell system that is being developed, the current DC/DC converter no longer suffices. Since there is no market for DC/DC converters in the power range of 7 – 15 kW, this component has to be developed in-house by Greenchoice Forze.

These developments make it more necessary than ever to program our micro-controllers ourselves. Segger Microcontrollers GmbH agreed to supply us with embedded software to make our software far more reliable and to make it possible to realize these developments in the short timeframe available.

Next to these developments, we are also busy developing our own motor controller. The current motor controllers are too heavy, too big and have too little features. The motor controller we are developing is just half the size of the current one and just a third of the size of the two controllers combined.

For the bodywork we thought of a new production method, which is far less time consuming. We use a wireframe as a mould. The frame will be covered with a stretch fabric, over which flax fibers en bio-resin can be applied. This state of the art material is the same composite as used on the Forze II.

ICT systems in the Forze III

An important part of the technology in the hydrogen racekarts consists of ICT systems. In figure 1 you can find an overview of these systems in the Forze III.

Our electronics system is a modular system consisting several printed circuit boards (PCB's) manufactured by Cyner. This is done to enhance reliability and to make it possible to improve subsystems without adapting the whole system.

On the PCB's, microcontrollers are implemented equipped with ARM7 processors from NXP. These microcontrollers are programmed in C-code in such a way that together they fulfill all the necessary tasks for the kart to be working. The different PCB's communicate via a CAN-bus network, the standard in the automotive industry.

Racing without harmful emission; A Utopia?

Greenchoice Forze is already doing it!

By participating in the first hydrogen racing class ever, called the Formula Zero Championship, they are promoting this clean technology. Greenchoice Forze is building their third kart, with which they plan to win the 2010 Formula Zero Cup!



Copyright: Richard van 't Hof

Forze I in the first hydrogen race ever, Rotterdam August 23rd

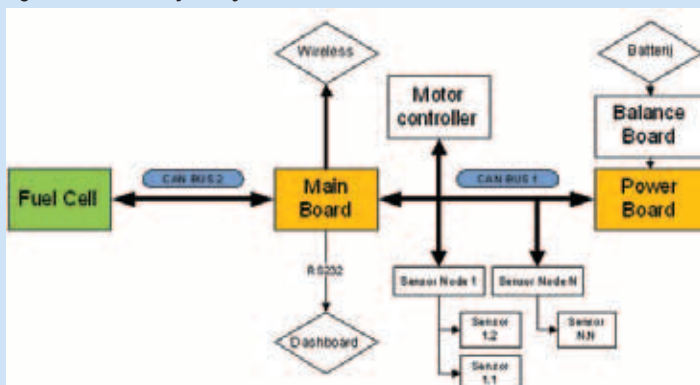


The main board is the nerve system of the kart, equipped with a LPC2388 microcontroller on which a real time operating (RTOS) runs. The main board handles the communication with the fuel cell and logs all data acquired on a SD-card. It also controls the telematics system. The software used is SEGGER embOS, emFile and embOS/IP.

The main function of the power board is supplying all electrical components with energy. Furthermore, all emergency protocols are implemented on this PCB.

The sensor nodes take care of the A/D conversion of the different sensors in the kart and sends this information via the CAN-Bus network to the main board.

Figure 1: Overview of the systems in the Forze III



The dashboard is the hardware interface between the driver and the kart. It consists of the start and stop buttons and the display for all critical data. At the moment we are investigating what kind of display is best for the Forze III.

The balance board controls the charging of the LiFePo4 battery package. The code for this PCB was already on the microcontroller implemented.

As stated earlier, the motor controller controls the power supply of the motors. It has to be swift and reliable. On the motor controller there is no RTOS running, but a superloop. The code is executed line after line.

The future

Greenchoice Forze is promoting sustainable technology by racing. Formula Zero is the first hydrogen racing class ever, but will definitely not be the last. There have been a lot of developments over the last years in hydrogen technology, but it is important to realize that this technology, in comparison to the combustion engine, is still in its infancy. By racing on hydrogen, the development of this new technology will gain a huge boost.

There are still some big logistical problems to be solved for the application of hydrogen cars, but there are no real technical barriers left to implement fuel cell technology in the automotive industry on a large scale.

Being part of the future

If you would like to have some more information or if you are interested in being part of this project, please go to our website www.greenchoice-forze.nl or send an e-mail to formulazero@tudelft.nl.

Brute kracht inzetbaar voor meer dan alleen beelden?

In de jaren negentig kwamen steeds krachtigere videokaarten voor de consument op de markt. En een steeds groter deel van de rendering pipeline werd in hardware uitgevoerd. De bewerkingsschappen in de rendering pipeline zijn de zogenaamde shaders. Bijvoorbeeld een vertex shader voegt bepaalde 3D-effecten toe aan objecten, een geometrieshader genereert nieuwe objecten vanuit al gedefinieerde objecten en een pixelshader berekent de kleur van een pixel. Een belangrijk nadeel was dat met het vastleggen van de specifieke shaders in hardware ook de rekenkracht per shader vast lag. Dit terwijl verschillende spellen verschillende eisen stelden aan de verdeling van de rekenkracht tussen de verschillende shaders.

Een GPU is een parallelle processor: hij voert zeer veel parallelle bewerkingen uit. Dat is bijvoorbeeld te zien aan de brede geheugenbussen, tot wel 256 bits. Ter vergelijking: de CPU is nog niet zo lang geleden van 32 naar 64 bits gegaan.

Een GPU werkt dus met veel meer data tegelijk. Maar het ontwerp is bedoeld voor eenrichtingsverkeer: er komt een brok data binnen met elementaire beeldinformatie, voor elke pixel berekent hij het een en ander, vervolgens stuurt hij dat er aan de andere kant uit richting het beeldscherm.

Een GPU is slecht in keuzes maken, zoals in het doorlopen van typische “if ..., then ..., else ...”-softwareconstructies. Daar wordt hij traag van. Een GPU is juist gebouwd om voor alle parallele paden hetzelfde te doen. Dat heet single instruction, multiple data (SIMD). Er zijn een instructiedecoder en een hele groep executie-units. Zodra er een keuze komt, kan die maar voor een executie-unit gelden en kunnen de anderen niets effectiefs doen. Een gewone processor is veel beter in het oplossen van if-then-else-problemen. Die kan slim door een reeks van voorwaarden en afhankelijkheden heen slompen en zich veel rekenwerk besparen, als een skiër die in de afdaling continu kiest en stuurt om de snelste weg te vinden. Een GPU is als een roeiboot met bijvoorbeeld acht roeiers die tegelijkertijd samen een doel proberen te bereiken. Als elke roeier een ander parcours zou moeten varen, dan kan dat alleen maar door acht parcoursen achter elkaar te varen met telkens één roeier aan het werk. Een GPU is dus goed in het parallel verwerken van rekenkundige bewerkingen volgens vaste formules.



Berekeningen aansturen met API

Als een GPU gebruikt gaat worden, moet dat gebeuren met speciale instructies, die bij elk merk GPU kunnen verschillen. Hierbij zijn de standaard API's niet bruikbaar. Het is dus hard nodig om toegang tot de GPU's te standaardiseren, zodat de functies die moeten worden uitgevoerd op een redelijk abstract niveau kunnen worden aangeboden. Op dit moment zijn de belangrijkste twee: CUDA van NVIDIA en de merkonafhankelijke variant OpenCL (Open Computing Language). OpenCL is initieel gedefinieerd door Apple Inc., maar is door hen via de Kronos Groep tot een open standaard gemaakt. Door deze API's te gebruiken kunnen de ingewikkelde details van een GPU worden beheerst en kan de programmeur zich richten op het onderhavige probleem. Ook zullen er door standaardisatieprogramma's (bv. Matlab) bibliotheken ter beschikking komen met GPU-ondersteuning voor veel voorkomende problemen, zodat niet iedereen opnieuw het wiel hoeft uit te vinden.

Rekenkracht GPU anders aanwenden

Door de genoemde API's is het, zoals eerder aangegeven, mogelijk om goed passende rekenkundige problemen, zonder uitvoer naar een beeldscherm, met succes sneller op een GPU uit te voeren dan op een CPU. Gedacht moet worden

aan o.a. matrixberekeningen en Fouriertransformaties, zoals die bijvoorbeeld voorkomen in ruimtelijke problemen als 3D-simulaties van stromingen of elektromagnetische velden. Dit zijn verschijnselen die zich laten analyseren door de wereld in kleine vakjes of kubusjes te verdelen om per vakje een natuurkundige vergelijking op te lossen (ook wel de eindige elementenmethode genoemd). Het oplossen van deze vergelijkingen vergt heel veel 'simpele' rekenstappen op een grote dataset, maar ook andere grootschalige berekeningen (vooral floating point) en beeldgeneratie voor de professionele en wetenschappelijke gebieden.

Voor dit soort toepassingen heeft NVIDIA een professionele GPU ontwikkeld onder de naam Tesla. Een onderzoeker heeft zo voor een relatief bescheiden bedrag een desktop die qua rekenkracht kan wedijveren met een supercomputer.

GPU in embedded systemen?

Voor professionele technische toepassingen is een GPU alleen maar bruikbaar in zijn natuurlijke habitat: op een grafische kaart, ingebouwd in een pc, met standaard software. Mochten GPU-chips al los verkrijgbaar zijn, dan nog zijn ze moeilijk in embedded systemen in te passen. Het probleem moet passen bij de architectuur van de chip, de chip moet passen in het systeem en het businessmodel moet kloppen. Zo heeft de GPU voor zijn inzetbaarheid bij grafische toepassingen een korte levensloop en is daardoor slecht verkrijgbaar op de lange termijn. De professionele embedded wereld is vrijwel altijd beter uit met een FPGA. Die zijn wel lang genoeg leverbaar. De keus in FPGA's is enorm: voor elk denkbaar probleem is wel een specifieke FPGA te vinden. Deze zijn ook nog eens uiterst flexibel. De ontwikkelaar kan namelijk zelf de balans kiezen tussen parallel en serieel, tussen data load en rekenkracht. Soms zijn er niet zoveel afzonderlijke berekeningen nodig, maar moet het wel op 10Gb datastromen gebeuren, zoals in een hoge resolutiecamera. Die genereert een bulk data die eerst wordt bewerkt voordat een pc er iets mee kan. Misschien zou dat met een GPU kunnen, maar voor bewerkingen in het embedded domein kiest Technolution er altijd voor om de datastroom met een FPGA te bewerken. Juist vanwege voorgenoemde zaken.

Toekomst GPU

Een van de grootste beperkingen van vandaag in het parallel rekenen is de doorvoersnelheid van het geheugen. Larrabee is de codenaam van een ontwikkeling bij Intel, die een hybride van een X86 CPU en een GPU ineen smeedt tot een GPGPU (General Purpose GPU). In dit geval wordt gebruik gemaakt van vele volwaardige CPU-kernen met SIMD-support en een conventionele cache. Door in de toekomst een CPU- en een Larrabeechip te integreren komen beide typen processors op een plak silicium te zitten en kunnen de verbindingen erg kort en dus snel zijn. Dit geeft veel flexibiliteit en performanceverbetering voor een breder scala rekenintensieve toepassingen. De performanceclaims van Intel worden door de bestaande GPU-specialisten met de nodige scepsis benaderd. Als deze combinatiechip slaagt, zou hij het einde van de losse grafische kaart kunnen inluiden en dus geheel nieuwe toepassingen voor de GPU in beeld brengen.

Dit artikel is overgenomen uit Objective 12 - november 2009, het relatiemagazine van Technolution over innovatie en technologie.

Java puzzlers

Traps, pitfalls and corner cases

Joey van den Heuvel

In this book review I would like to give you an insight in the wonderful world of Java Puzzlers. The book Java Puzzlers is written by Joshua Bloch and Neal Gafter which both have worked on the java core and thus know a lot about the traps, pitfalls and corner cases of java.

They have created a book full of java puzzles. Short programs that appear to do one thing but actually they are doing something else. And I thought that the best way to get you interested in reading this book is to just show you two brain teasing puzzles. So here are two examples out of the book and remember: "First try to solve the puzzle before skipping to the solution!"

Puzzle: Long Division

This puzzle is called Long Division because it concerns a program that divides two long values. The dividend represents the number microseconds in a day; the divisor, the number of milliseconds in a day. What does the program print?

```
public class LongDivision {
    public static void main(String[] args) {
        final long MICROS_PER_DAY = 24 * 60 * 60 * 1000 * 1000;
        final long MILLIS_PER_DAY = 24 * 60 * 60 * 1000;
        system.out.println(MICROS_PER_DAY / MILLIS_PER_DAY);
    }
}
```

Solution: Long Division

This puzzle seems reasonably straightforward. The number of milliseconds per day and the number of microseconds per day are constants. The number of milliseconds per day differs only in that it is missing the final factor of 1000. When you divide the number of microseconds per day by the number of milliseconds per day, all the factors in the divisor cancel out, and you are left with 1000, which is the number of microseconds per millisecond. Both the divisor and the dividend are of type long, which is easily large enough to hold either product without overflow. It seems, then, that the program must print 1000. Unfortunately, it prints 5. What exactly is going on here?

The problem is that the computation of the constant MICROS_PER_DAY does overflow. Although the result of the computation fits in a long with room to spare, it doesn't fit in an int. The computation is performed entirely in int arithmetic, and only after the computation completes is the result promoted to a long. By then, it's too late: The computation has already overflowed, returning a value that is too low by a factor of 200. The promotion from int to long is a widening primitive conversion, which preserves the (incorrect) numerical value. This value is then divided by MILLIS_PER_DAY, which was computed correctly because it does fit in an int. The result of this division is 5.

So why is the computation performed in int arithmetic? Because all the factors that are multiplied together are int values. When you multiply two int values, you get another int value. Java does not have target typing; a language feature wherein the type of the variable in which a result is to be stored influences the type of computation.

Puzzle: No Pain, No Gain

This program prints a word, using a random number generator to select the first character. Describe the behavior of the program:

```
import java.util.random;

public class Rhymes {
    private static Random rnd = new Random();

    public static void main(String[] args){
        StringBuffer word = null;
        switch(rnd.nextInt(2)) {
            case 1: word = new StringBuffer('P');
            case 2: word = new StringBuffer('G');
            default: word = new StringBuffer('M');
        }
        word.append("ain");
        system.out.println(word);
    }
}
```

Solution: No Pain, No Gain

At the first glance, this program might appear to print out the word Pain, Gain, and Main with equal likelihood varying from run to run. It appears to choose the first letter of the word, depending on the value chosen by the random number generator: M for 0, P for 1, and G for 2. The puzzle's title might have provided you with a clue that it doesn't actually print Pain or Gain. Perhaps more surprisingly, it doesn't print Main either, and its behavior doesn't vary from run to run. It always print ain.

Three bugs conspire to cause this behavior. Did you spot them all? The first bug is that the random number is chosen so the switch statement can reach only two of its three cases. The specification for Random.nextInt(int) says: "Returns a pseudorandom, uniformly distributed int value between 0 (inclusive) and the specified value (exclusive)" [Java-API]. This means that the only possible values of the expression rnd.nextInt(2) are 0 and 1. The switch statement will never branch to case 2, which suggests that the program will never print Gain. The parameter to nextInt should have been 3 rather than 2.

The Second bug is that there are no break statements between the cases. Whatever the value of the switch expression, the program will execute that case and all subsequent cases caused by the fall-through principle. Each case assigns a value to the variable word, and the last assignment wins. The last assignment will always be the one in the final case (default), which is new StringBuffer('M'). This suggests that the program will never print Pain or Main but always Main.

The last and most subtle bug is that the expression new StringBuffer('M') probably does not do what you think it does. You may not be familiar with the StringBuffer(char) constructor, and with a good reason: It does not exist. There is a parameterless constructor, one that takes a String indicating the initial contents of the string buffer and one that takes an int indicating its initial capacity. In this case, the compiler selects the int constructor, applying a widening primitive conversion to convert the char value 'M' into the int value 77. In other words, new StringBuffer('M') return an empty string buffer with an initial capacity of 77. The remainder of the program appends the characters a, i, and n to the empty string buffer and prints out its contents, which are always ain. 

Clustering algorithm for highly mobile wireless sensor networks

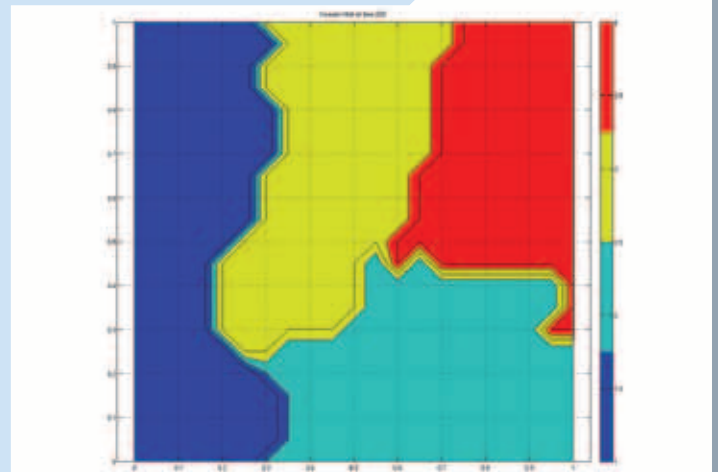
A.S. Pruteanu

The increasing complexity and dynamicity of distributed information systems present new challenges for designers requiring new engineering paradigms. Self-Adaptation and Self-Organization are emerging as highly promising approaches.

Biological systems, from embryos to social insects, get tremendous mileage using large decentralized systems of comparatively simple and unreliable components to achieve complex goals. Recent technological advances have made possible man-made systems composed of vast numbers of components, ranging from sensor networks to self-assembling modular robots. A key challenge is controlling such systems: how does one harness the power of self-organization to achieve user-specified global goals from the local interactions of identically programmed, simple agents? Domain-specific “compilers” systematically solve complex global tasks by composing a remarkable small set of robust, analyzable, bio-inspired local behaviors. Furthermore, several languages can encode complex notions of self-repair, thus relieving the user from explicitly managing fault-tolerance. The system is able to disseminate the local interaction of the agents with respect to the predefined behavior of the ensemble. A wireless sensor network is a highly distributed system where the transmission of information in a reliable and energy efficient way is its most important requirement. We propose a new technique for solving the problem of routing efficiency in case of increased mobility of the nodes. Our approach is inspired by cellular automata algorithms and uses simple, local rules to construct and maintain stable clusters (global behavior). We evaluated the idea through simulations and showed its validity.

Mobility has become one of the most important requirements for wireless sensor networks applications. Due to its negative effects (frequent and random topology changes leading to breaks in the communication links) we end up with higher power consumption and lower quality of service. To solve these drawbacks an overlay network can be used to reduce the effects of mobility. As a consequence, a highly dynamic topology appears as almost static to the upper layers. It has been proven that the packet delivery delays are reduced in a hierarchical structure compared to a flat one, thus resulting in better routing efficiency. In large mobile scenarios, the flat hierarchical structure produces excessive route-maintenance traffic that can saturate the network. Clustering in mobile networks can be seen as a virtual partitioning of the moving nodes into various groups with respect to the distance to others or their motion correlations. The highly dynamic nature of mobile ad-hoc networks results in frequent and unpredictable changes of the topology. Additionally, the classic routing techniques (either proactive or reactive) in large-scale, dynamic MANETs do not perform well and are inefficient in maintaining accurate paths. Introducing hierarchy is a technique to increase the network performance by delegating to a subset of nodes the task of performing long distance communications. The basic idea behind clustering is to form groups (clusters) of neighboring nodes from the flat structure of the network. Within each cluster, a node (leader) is selected to communicate with other clusters on behalf of its neighbors. In addition to packet routing and forwarding, cluster heads can also control the channel scheduling and aggregate the data within its domain.

Figure 1



The concept of clustering has had a lot of attention from two research communities: MANETs and Wireless Sensor Networks. The former relied on more capable devices in terms of available energy, transmission range and propagation properties. The latter had to deal with a lot more restrictions: the nodes are asleep most the time to save energy; they communicate on much shorter ranges with a lot less throughput. Information loss due to radio interference and battery depletion is a common characteristic of the network. A short enumeration of well-known algorithms is presented. Weight-based clustering algorithms assign to each node in the network an application specific measure, which usually includes a number of parameters related to how suitable the node is for the cluster head role. Examples of such algorithms are: LCA [5], DCA [17], DMAC [2], G-DMAC [1], K-CONID [9], WCA [4], C4SD [13]. Time-based algorithms enable the election of cluster head nodes based on the order in time when the potential nodes announce their candidacy. Examples of algorithms are: Passive clustering algorithm [6] and ACE clustering algorithm [3]. Probabilistic protocols are designed mainly for static networks (e.g. static Wireless Sensor Networks), as they run in synchronized rounds for balancing the energy consumption. Representative algorithms are: LEACH [7], HEED [11], MOCA [12], EEMC [8]. Protocols with decisions based on semantic information: Smart Clustering algorithm [10], Tandem [13]. One of the newest algorithms are: GBL [14], DDVC and DDLC [15]. They are using either Doppler shift of GPS to measure the relative motion. The first approach is using a new technique but is not accurate enough for low-speed estimates. The second one uses a strong assumption: GPS.

The algorithm

Our technique is inspired by cellular automata algorithms where simple, local only interactions lead to complex global behavior of the system. In our case, d-hop neighborhood is defining the local interaction. Since the topology of the network is dynamic (mobile nodes), the overlaying layer has to hide to the upper ones the increased mobility. The authors of related papers presume the existence of additional information like GPS coordinates, Doppler shift of GPS, RSSI readings, mobility estimations, speed values of the nodes etc as helping points. Although for some application scenarios (outdoor deployments) this is a feasible approach, in most of the cases, the lack of a connection to the satellites or the unreliable radio measurements caused by interferences or propagation problems, prove that such strong assumptions are impractical.

Mobility estimation can be obtained when there is a degree of correlation within the trajectories of the nodes. Due to all these reasons, our main contribution is to create an algorithm that relies only on the information provided by the distributed local interactions between the nodes. In this way we are not limited to outdoor deployments with nonrandom trajectories and we do not need expensive equipments in terms of energy consumption (GPS). We now can extend the applicability of our approach to any scenario: indoors, outdoors or mixed.

Cluster formation and maintenance

The cluster formation technique is inspired by the physical laws (the principle of equilibrium of gasses inside a container [16]). They balance each other so that in time they stabilize while their molecules move randomly. We use the concept of pressure (shared variable between the nodes of a cluster) to describe the probability of a node to impose its cluster ID assignment to its neighbors. Local rules that maintain the equilibrium are: 1. pressure diffusion inside the clusters, 2. pressure exchange between the clusters and 3. Cluster election rule. The later one is based on either majority voting (counting the number of neighbors belonging to different clusters) or pressure imposing. These simple interactions are able to maintain almost fixed network domains (clusters) while the actual nodes are moving.

Cluster head election

The decision of which node to use as a cluster head is taken based on domain change frequency of the nodes and their d-hop neighbors: the higher the value, the less the probability of becoming a cluster head and the vice versa. In this way, nodes that change less times their domain(cluster assignment), meaning they are either less mobile or are located further away from the cluster border, have a higher chance of becoming cluster heads. Their stability is crucial for the packet loss metrics.

Simulation results

To validate our technique we simulate the mobility scenario using a Random-Waypoint model with Matlab and MIXIM (Omnnet based network simulator). As shown in the Figure 1, the clusters are stable even after 200 points in time and move slowly. A node can cover the space from a minimum of 0.01 [unit]/time to a maximum of 0.1 [unit]/time. The sum of all pressures inside a domain is constant through simulation time. The average number of nodes per cluster is 125 out of a total of 500 (Figure 2). Compared to GBL[14] and DDVC[15] where nearby nodes with similar mobility patterns (detected through GPS or Doppler shift), are clustered, we do not use any information about the motion of individual nodes. In this way, our technique can be used for random motion patterns. In DDVC[15] a cluster head is in charge of its corresponding cluster formation (acceptance of nodes as members) and has knowledge of all its members, acting as a central authority.

Figure 2

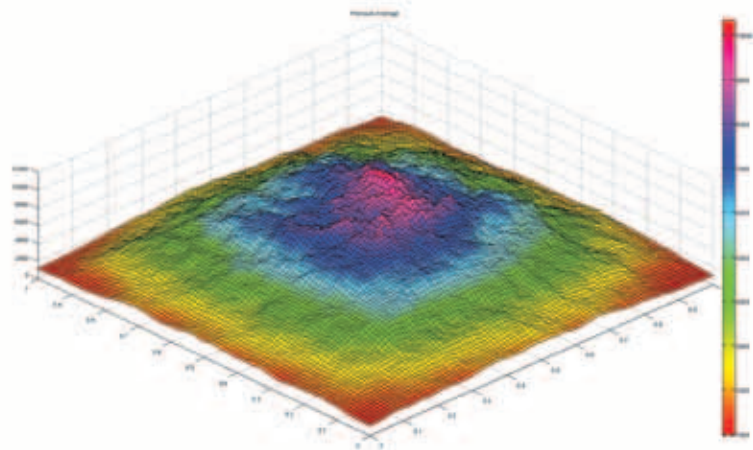
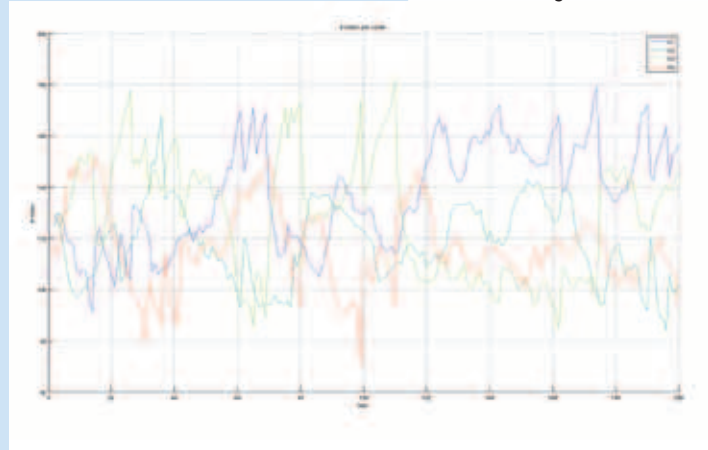


Figure 3

Such an approach is not fully de-centralized and although it does not need coordination from the gateway still involves a lot of micro-management by the cluster-head. This approach is feasible when groups of nodes are easy to track. When randomness occurs in the motion such an approach is not feasible. As a consequence, a fully distributed approach is needed, where gradients established in the network are able to direct packets toward the cluster-heads. Additional knowledge about all the cluster members is not needed. The cluster change frequency graph shows which nodes are potentially good candidates as cluster heads (Figure 2). The nodes in the corners did not change often their cluster assignment so they have a high election probability. Nodes in the center have less probability due to their increased changing frequency.

Conclusion

Our new technique constructs a static overlay network on top of the mobile infrastructure. We hide to the upper layers (routing) the increased dynamic topology. Our preliminary simulation results validate our starting approach as highly promising. Our solution has three major improvements compared to other algorithms: 1. it is a fully de-centralized (distributed) approach (it uses only the local interaction between the nodes); 2. It uses no hard-assumptions like GPS coordinates or correlated movement of the nodes; 3. It uses no motion prediction (we can apply our methodology to totally random motion scenarios). The obtained freedom allows us to create application scenarios like: indoor spaces, mixed indoor-outdoor spaces, random motion of the nodes, high node density. 

References

- [1] S. Basagni. Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks. In Proceedings of Vehicular Technology Conference (VTC), pages 889893. IEEE Computer Society, 1999.
- [2] S. Basagni. Distributed clustering for ad hoc networks. In Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN), pages 310313, Washington, DC, USA, 1999. IEEE Computer Society.
- [3] H. Chan and A. Perrig. Ace: An emergent algorithm for highly uniform cluster formation. In Proceedings of the First European Workshop on Sensor Networks (EWSN), pages 134171. Springer, January 2004.
- [4] M. Chatterjee, S. K. Das, and D. Turgut. Wca: A weighted clustering algorithm for mobile ad hoc networks. Cluster Computing, 5(2):193204, 2002.
- [5] A. Ephremides, J. Wieselthier, and D. Baker. A design concept for reliable mobile radio networks with frequency hopping signaling. Proceedings of the IEEE, 75(1):373, 1987.
- [6] M. Gerla, T. J. Kwon, and G. Pei. On-demand routing in large ad hoc networks with passive clustering. In Wireless Communications and Networking Conference (WCNC), pages 100105. IEEE Computer Society, 2000.
- [7] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan. An applicationspecific protocol architecture for wireless microsensor networks. Wireless Communications, IEEE Transactions on, 1(4):660670, 2002.
- [8] Y. J., L. W., Y. Kim, and X. Yang. Eemc: An energy-efficient multi-level clustering algorithm for large-scale wireless sensor networks. Computer Networks, 52(3):5132, 2008.
- [9] F. G. Nocetti, J. S. Gonzalez, and I. Stojmenovic. Connectivity based k-hop clustering in wireless networks. Telecommunication Systems, 22(14):205220, 2003.
- [10] M. Strohbach and H. Gellersen. Smart clustering - networking smart objects based on their physical relationships. In proceedings of the 5th IEEE International Workshop on Networked Appliances, pages 131 135. IEEE Computer Society, 2002.
- [11] O. Younis and S. Fahmy. Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Transactions on Mobile Computing, 3(4):366379, 2004.
- [12] A. M. Youssef, M. F. Younis, M. Youssef, and A. K. Agrawala. Distributed formation of overlapping multi-hop clusters in wireless sensor networks. In Proceedings of the Global Telecommunications Conference (GLOBECOM), pages 16, 2006.
- [13] Marin-Perianu, R.S. (2008) Wireless Sensor Networks in Motion -Clustering Algorithms for Service Discovery and Provisioning. PhD thesis, University of Twente. CTIT Ph.D.-thesis series No. 08-130 ISBN 978-90-365-2745-3
- [14] G. H. K. Lam, H. V. Leong, and S. C. Chan. Gbl: Group-based location updating in mobile environment. In 9th International Conference on Database Systems for Advanced Applications (DASFAA), pages 762774, 2004.
- [15] E. Sakhaee and A. Jamalipour, "Stable clustering and communications in pseudolinear highly mobile ad hoc networks," Vehicular Technology, IEEE Transactions on, vol. 57, no. 6, pp. 3769-3777, Nov. 2008.
- [16] Cellular Automata Modeling of Physical Systems by Bastien Chopard (Author), Michel Droz (Author)
- [17] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. Selected Areas in Communications, IEEE Journal on, 15(7):1265-1275, 1997.

Mathematical modeling of particle nucleation and growth in metallic alloys

Dennis den Ouden

As a student of Applied Mathematics I wanted to specialize in the field of Numerical Analysis, with a particular focus on the application of mathematics to real life phenomena. As such I choose a subject of metallurgy for my Master thesis project. In this article I will elaborate on the problems I will model, the model itself and of course the results I obtained.

Metallurgical phenomenon

In the field of metallurgy two phenomena are considered of great importance in the manufacturing of alloys. First there is the behavior of the alloy on an molecular level and the influence of this behavior on the strength of the material. Second there is the deformation of the material during hot processing and the effect on the molecular dynamics.

The behavior of alloys on a molecular level can best be described by considering a quasi-binary alloy, e.g. an alloy consisting of two or more elementary elements, such as Cu-Co (copper-cobalt) or Al-Mg-Si (aluminum-magnesium-silicon). This alloy can be seen as a solidified solution of the solute element, e.g. Co or Mg+Si, in the solvent element, e.g. Cu or Al. In a certain temperature range, which can vary from alloy to alloy, it is possible that some Co atoms or Mg_2Si molecules form large groups, either due to defects in the material or spontaneously. These groups are called particles or precipitates and deform the alloy structure locally. In theory these particles can have any size, but are in most cases in the order of nanometers (10^{-9} meters). The local deformation of the material due to particles can severely change, both positive and negative, the overall behavior of the material under hot processing. Therefore I focused on a model that predicts the particles present in an alloy. The alloy under I took under investigation in this project was a simplified version of the alloy AA 6082, an aluminum alloy. This alloy consist of the solvent element aluminum and the two solute atoms Magnesium (Mg) and Silicon (Si). For this alloy it is assumed that only particles consisting of Mg_2Si are formed. The overall concentrations of the elements are 0.63wt% Mg, 0.9wt% Si with the remainder Al. Here wt% stands for weight percent.

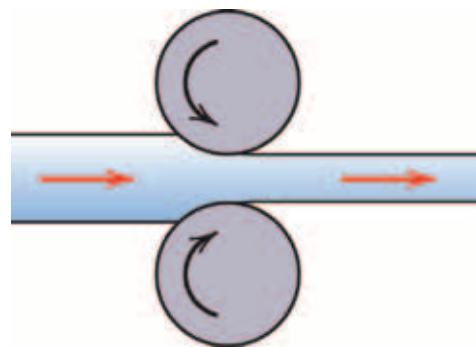


Figure1: Flat rolling. Image from Wikipedia.

Hot processing of alloys happens at very large scales in the industry. An example is the rolling of alloys after they have been cast, see figure [ref{hot}]. During this process a certain amount of pressure is exerted on a exterior pint of the material causing two types of deformations. First there are the deformations that disappear after the force is released, the so called elastic deformations. Second there are the deformations that are permanent and stay after the force has been released, the so called plastic deformations. Although plastic deformations are those wanted during hot processing of alloys, I focused on elastic deformations as this behavior is more easily modeled and still can give significance information on the influence of hot processing on the molecular behavior of the alloy.



a) The undeformed cylinder

b) The deformed cylinder

Figure 2



Modelling particle nucleation

In the field of metallurgy several approaches are possible to obtain information about the theoretical amount of particles in an alloy. A well-known approach is the KWN-model [1]. I compared to specific variants of this model, one by Myhr and Grong [2], and another by Robson et al. [3]. I will not state the comparison itself, but I will state the resulting model.

The KWN-model is based on the evolution of the particle number density for spherical particles. This density is expressed as $N(r, t)$ where r is the amount of particles with radius r per cubic meter at time t . The behavior of this quantity is best described using the continuity equation

$$\frac{\partial N}{\partial t} = -\frac{\partial[Nv]}{\partial r} + S,$$

where $v \equiv v(r, t)$ is the rate at which a particles with radius r grows or shrinks (if negative) at time t and $S \equiv S(r, t)$ a source function predicting the number of newly formed particles with radius r at time t . Both v and S depend explicitly on the value of N .

From a known value of N the value of several variables can be obtained, such as concentrations. The value of the growth rate can be expressed as

$$v = \frac{\bar{C} - C_i}{C_p - C_i} \frac{D}{r},$$

where $\bar{C}$ is the mean concentration in the matrix (i.e. outside the particles) of the solute, C_p the concentration of solute inside the particle and C_i an equilibrium concentration at the edge between the matrix and the particle. This last concentration depends explicitly on the radius of each particle. Finally D is the diffusion constant of the solute in the solvent. The mean concentration $\bar{C}$ can be derived from the value of N .

The source term S can be seen as a point function [4] depending on the critical radius $r^* \equiv r^*(t)$ for which $v = 0$ and the nucleation rate $j \equiv j(t)$:

$$S = \begin{cases} j & \text{if } r = r^* \\ 0 & \text{else.} \end{cases}$$

The largest difference between the two models compared was in the formula of j . Using a combination of both models I assumed that the following formula holds:

$$j = j_0 \exp\left(-\frac{\Delta G_{\text{het}}^*}{RT}\right) \exp\left(-\frac{Q_d}{RT}\right),$$

with j_0 a empirical constant, R the gas constant, T the absolute temperature, Q_d a constant related to diffusion and ΔG_{het}^* an energy barrier that should be overcome before nucleation happens. This energy barrier is a function of other energies:

$$\Delta G_{\text{het}}^* = \frac{\bar{A}_0^3}{(\Delta G_v + \Delta G_s)^2}.$$

The energy ΔG_v is related to the value of the concentrations and reflect the need of the system to reach a certain stable state. In this project I focused on the other energy ΔG_s , the strain energy. This term is the sum of energies due to all possible strains, but we will restrict our selves to a binary case, in which only strain energy due to misfit between the matrix and a particle with radius r^* , ΔG_s^m and strain energy resulting from elastic deformations, ΔG_s^{el} . This latter term will be defined below.

Modelling elastic deformation

As mentioned my Master thesis project focused solely on elastic deformations. For simplicity we assume a cylindrical region which is deformed such that rotation symmetry is maintained. This means that no forces or deformations in tangential direction are considered. Using a simple force balance, two partial differential equations are derived:

$$\begin{aligned} \frac{\partial \sigma_{\eta\eta}}{\partial \eta} + \frac{\partial \sigma_{\eta z}}{\partial z} + \frac{\sigma_{\eta\eta} - \sigma_{\theta\theta}}{\eta} + b_\eta &= 0 \\ \frac{\partial \sigma_{\eta z}}{\partial \eta} + \frac{\partial \sigma_{zz}}{\partial z} + \frac{\sigma_{\eta z}}{\eta} + b_z &= 0. \end{aligned}$$

In these equations η stands for the independent radial variable, z for the independent axial variable and $\sigma_{\alpha\beta}$ for the stress acting on region with α constant and in the direction of β . These stresses are related with the strain in the system by the cylindrical Hook's Law:

$$\sigma_{\alpha\beta} = \delta_{\alpha\beta} \lambda (\varepsilon_{\eta\eta} + \varepsilon_{zz} + \varepsilon_{\theta\theta}) + 2\mu \varepsilon_{\alpha\beta},$$

which are in turn related to the deformations present in the system:

$$\begin{aligned} \varepsilon_{\eta\eta} &= \frac{\partial u_\eta}{\partial \eta} & \varepsilon_{\theta\theta} &= \frac{u_\eta}{\eta}, \\ \varepsilon_{zz} &= \frac{\partial u_z}{\partial z} & \varepsilon_{\eta z} &= \frac{1}{2} \left(\frac{\partial u_\eta}{\partial z} + \frac{\partial u_z}{\partial \eta} \right), \end{aligned}$$

If the deformations are known using the above equations and some set of prescribed boundary conditions, the strain energy of the system can be calculated at each point in the system, by using the formula

$$\Delta G_s^{\text{el}} = \frac{1}{2} \underline{\sigma} : \underline{\varepsilon},$$

where $:$ is the Frobenius inner product. This value can then be used to calculate the nucleation rate j as described above.

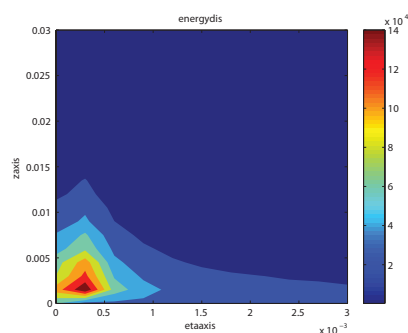


Figure 3

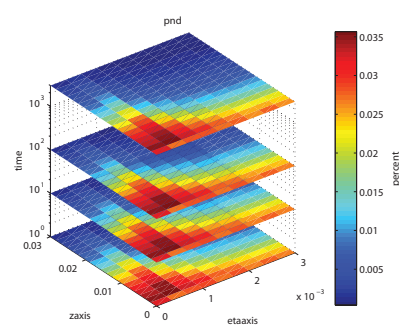


Figure 4



Used numerical methods

To simulate the behavior of the combined model, two systems need to be discretized. For the model predicting the number of particles I used a combination of the familiar upwind method and an IMEX- θ method. These methods give an order 1 accurate approximation in both place and time. For the discretization of the elastic equations I applied a basic finite element method using linear triangles in the (η, z) domain.

Results from simulation

To illustrate the effect of incorporating I simulate a so called tension test on a solid cylinder with a radius of 3mm and a height of 30mm . This test consists of fixing the bottom of the cylinder in both the radial and axial direction, fixing the top of the cylinder in the radial direction and applying a force of

$$F = 6 \times 10^9 \frac{\text{N}}{\text{m}^2},$$

in the axial direction. This specific force should physically cause plastic deformations, but for the moment we assumed only elastic deformations occur. A picture of the undeformed and deformed cylinder can be seen in Figure 2.

The resulting deformations caused a strain energy with the typical form as can be seen in Figure 3. The effect of these energy levels can be seen back in all variables that we can calculate from the results. A good influence can be seen from the total particle present in the system, as depicted in Figure 4.

Conclusions

From the various results obtained with the combined models, the conclusion can be drawn that the influence of elastic deformations on the nucleation and growth of particles can be modeled in a straightforward manner, resulting in predictable difference with the results from nucleation and growth of particles in absence of elastic deformations.

If you are interested in knowing more about the nucleation and growth of particles in alloys, elastic deformations and the application of numerical methods, this Master Thesis can be found at repository.tudelft.nl. 

References

- [1] Kampmann, R., Eckerlebe, H., and Wagner, R., in: Materials Research Society Symposium Proceedings, volume 57, page 525. MRS, 1987.
- [2] Myhr, O. R. and Grong, O., *Modelling of non-isothermal transformations in alloys containing a particle distribution*, Acta Materialia, 48(7):1605-1615, 2000.
- [3] Robson, J.D., Jones, M.J., and Prangnell, P.B., *Extension of the N-model to predict competing homogeneous and heterogeneous precipitation in Al-Sc alloys*, Acta Materialia, 51(5):1453-1468, 2003.

Oliebollen eten

Puzzel LI

Prof.dr J.M. Aarts

Beschouw de open bol met straal l , maar zonder rand. Precies gezegd: de open bol met straal 1 en middelpunt O is de verzameling van alle punten X uit de ruimte met de eigenschap dat de afstand van O tot X kleiner is dan 1 : $OX < 1$. Daarin (in die open bol met straal 1) leg ik een configuratie van n punten met onderlinge afstand ≥ 1 . Dan is $n \leq 19$.

Wie helpt me om dit te bewijzen of te weerleggen. Ik heb een tijd geprobeerd dit probleem (vind een bovengrens voor n) aan te pakken door te kijken naar ingeschreven veelvlakken en ook de truc van Van Kan te gebruiken. Ik kwam niet verder dan ik op Oudjaar al was. Als ik zo'n configuratie van n punten heb dan kan ik iedere punt van de configuratie een bolletje met straal $\frac{1}{2}$ weg laten eten zonder dat die punten van elkaars oliebolleendeel snoepen. Zo kwam ik op die 19. Wie kan dit bewijzen (ik had op oudejaarsavond al aardig wat op, olieballen bedoel ik) en/of wie kan de bovengrens naar beneden bijstellen?

Ook nu is een goede (of nog betere) oplossing weer 10 punten waard. De inzendingen kunnen tot en met 16 april naar machazine@ch.tudelft.nl gestuurd worden. 

Van Kan verslaat alles en iedereen

Oplossing puzzel L

Prof.dr J.M. Aarts

Er waren twee inzendingen die veel op elkaar leken, namelijk van Thijmen Krebs en van Misha Stassen. Zij ontvangen beide 10 punten voor de ladder. Hier is de oplossing. Kijk eens naar een die (open) cirkelschijf D met straal 1 . Verdeel D als een taart in zes gelijke punten. Voor iedere taartpunt geldt: twee punten op die taartpunt hebben afstand kleiner dan 1 . Ik beschouw nu een verzameling van n punten in D met onderlinge afstanden groter dan of gelijk aan 1 . Ik zal laten zien dat $n \leq 6$; dat betekent dat bij een configuratie van 7 punten er altijd 2 van die 7 punten zijn met onderlinge afstand < 1 , hetgeen te bewijzen was. Welnu, als $n \geq 7$ en ik dus 7 punten op 6 taartpunten moet leggen, dan zegt het pigeon hole principle (das Schubladeprinzip) me dat er twee punten op eenzelfde taartpunt terecht komen en dus onderlinge afstand kleiner dan 1 krijgen, hetgeen verboden is.

Jos van Kan, een uiterst slimme collega, kwam met een heel knappe verbetering van de uitspraak van de puzzel. Gelukkig voor de puzzelaars doet hij buiten mededinging mee, anders maakten zij geen kans meer op een prijs. Jos van Kan bewijst dat bij een configuratie van 6 punten er altijd 2 van die 6 punten zijn met onderlinge afstand < 1 . Het bovenstaand bewijs volgend moet bewezen worden dat $n \leq 5$ is. Om dat te bewijzen deel je de taart niet zo maar in 6 punten, maar zorg je er ook voor dat één van de sneden door één van de punten gaat; dan blokkeert dat ene punt 2 taartpunten, en moeten de overige 5 over 4 taartpunten verdeeld worden en dat kan niet. De goede oplossers van deze week scoren geweldig op de ladder en stijgen met stip. Thijmen Krebs staat meteen weer bovenaan en ook Misha Stassen (een oude bekende) zit weer hoog in de ladder, welke door ruimtegebrek op te vragen is bij de MaChazine. 



Wiskunde en de paradox van rationele vaccinatiebeslissingen

Radboud Duintjer Tebbens (Risico-en beslissingsanalyse)

Vaccinaties redden wereldwijd ieder jaar ongeveer 3 miljoen levens.¹ Niet alleen zijn ze duidelijk één van de meest succesvolle tools om ziektes te voorkomen, vaccinaties zijn doorgaans ook relatief goedkoop.

Een groot aantal studies heeft aangetoond dat verschillende vaccinatieprogramma's enorm kosteneffectief zijn, wat ruwweg wil zeggen dat de kosten per gewonnen levensjaar in goede gezondheid heel laag zijn. Sommige vaccins blijken zelfs kostenbesparend als je de besparingen van medische kosten meeneemt. Zo hebben wij laten zien dat het vaccinatiebeleid voor polio in de V.S. sinds 1955 \$180 miljard (!) meer aan medische besparing dan aan vaccinatiekosten heeft opgeleverd.² Op grond van deze argumenten lijkt het een *no-brainer* om te vaccineren. Toch is er groeiende beweging die zich verzet tegen vaccinaties. Het is interessant dat deze beweging zich deels manifesteert in de beter opgeleide groepen van rijkere landen, bijvoorbeeld in progressieve gemeenschappen in de V.S. en Duitsland. Ook in Nederland heeft alle ophef rond de Mexicaanse griep hevige gevoelens over vaccinaties losgemaakt en komen er zo nu en dan weer onnodige epidemieën (bijv. kinkhoest, mazelen, polio) voor in bevolkingsgroepen die zich tegen vaccinatie verzetten. Gezien de sterke argumenten vóór vaccinatie klinkt de beweging tegen vaccinaties irrationeel. Wiskundige modellen kunnen ons echter helpen om te begrijpen wat precies ten grondslag ligt aan de antivaccinatie gevoelens. Hierbij moet ik wel zeggen dat ook irrationele argumenten en foutieve informatie een rol spelen,³ en dat ik in dit stukje slechts de top van de fascinerende ijsberg van de vaccinatiepolitiek zal beschouwen.

Een belangrijk aspect is hoe we omgaan met onzekerheid. Het is niet alleen onmogelijk aan te tonen dat een vaccin absoluut 100% veilig is, maar ook heel moeilijk om *a priori* statistisch aan te tonen dat een vaccin bijvoorbeeld 99.9% veilig is omdat dat een studie van praktisch onmogelijke (of te dure) omvang vereist.⁴ Hierdoor kunnen integere voorstanders vaak niets beter zeggen dan dat er geen gegevens bestaan die aantonen dat een (nieuw) vaccin niet veilig is. De juiste interpretatie van een dergelijke uitspraak is dat we op grond van resultaten in laboratoria weten dat het vaccin hoogstwaarschijnlijk heel veilig is, maar dat we geen harde gegevens hebben om de exacte veiligheid te bepalen. Het gebrek aan goede gegevens geeft echter ruimte aan allerlei

speculaties en angsten. Wat doorgaans beter bekend is, zijn de risico's van niet vaccineren. Vaccinatiebeslissingen komen dus in praktijk vaak neer op een afweging tussen onzekere risico's. In de formele beslissingstheorie is bewezen dat iemand die volledig rationeel handelt altijd zal kiezen voor die optie die het hoogste verwachte nut oplevert, waarbij het verwachte nut gebaseerd is op subjectieve kansen op verschillende uitkomsten. Dat geeft weer aan dat het effectief communiceren van de onzekerheid een belangrijke *public relations* kwestie is voor een overheid die vaccinaties wil doorvoeren. Wiskundige technieken zoals *structured expert judgment* kunnen tot de communicatie bijdragen door op wiskundig verantwoorde wijze de meningen van verschillende experts te combineren in een "consensus" verdeling, die de gezamenlijke onzekerheid van de experts weergeeft.⁵ Ook het doorrekenen van onzekerheid in een wiskundig model via een probabilistische gevoeligheidsanalyse of onzekerheidsanalyse, is een belangrijke manier om de uiteindelijke onzekerheid waarop we beslissingen kunnen baseren te bepalen.

Maar zelfs als we volledig rationeel handelen en alle onzekerheid over kosten, baten en risico's van vaccins in acht nemen, kan op het eerste oog irrationeel anti-vaccinatie gedrag ontstaan. Beschouw een vaccin waarmee we de kans p_z om een ziekte te krijgen kunnen reduceren tot 0. Voor het gemak nemen we even aan dat we het nutsverlies bij het krijgen van de ziekte in een geldbedrag van C_z euro kunnen uitdrukken. Het vaccin kost C_v euro en er bestaat een kans p_b op een ernstige bijwerking waarvan het nutsverlies C_b euro is. Volgens beslissingstheorie zouden we dus rationeel kiezen voor vaccinatie zolang het verwachte nutsverlies bij vaccinatie kleiner is dan het verwachte nutsverlies zonder vaccinatie, oftewel:

$$C_v + p_b C_b < p_z C_z \Leftrightarrow p_z > \frac{C_v + p_b C_b}{C_z}$$

We kunnen deze beslissingsregel meenemen in een zogenaamd ziekteverspreidingsmodel. Een dergelijk model verdeelt de bevolking in *susceptibles* (mensen die nog ontvankelijk zijn voor een ziekte), *infecteds* (mensen die de besmetting dragen), en *removeds* (mensen die niet deelnemen aan de verspreiding omdat

Figuur 1: Stock-and-flow diagram van het beslissingsmodel voor vaccinaties en grafiek van de vaccinatie-belissing en jaarlijks aantal nieuwe gevallen als functie van de tijd.

Het SIR-model en de waarden van variabelen zijn gebaseerd op Edmunds WJ, Medley GF, Nokes DJ. Evaluating the cost-effectiveness of vaccination programmes: a dynamic perspective. *Statistics in Medicine* 1999;18(23):3263-82, m.u.v. de recovery rate (γ) van 12 i.p.v. 50 per jaar en het weglaten van de vaccine efficacy. Verder nemen we aan dat $C_v=100$, $C_b=C_z=20\ 000$, en $p_b=0.005$



ze al besmet zijn geweest of een vaccin hebben genomen).⁶ De verandering in omvang van deze groepen beschrijven we met een eenvoudig stelsel differentiaalvergelijkingen:

$$\begin{cases} \frac{dS}{dt} = (1-k)bN - \beta IS - \mu S \\ \frac{dI}{dt} = \beta IS - \mu I - \gamma I \\ \frac{dR}{dt} = kbN + \gamma I - \mu R \end{cases} \quad (*)$$

Hierbij is (in goed Nederlands) S =susceptibles, I =infecteds, R =removeds, N =total population, b =birth rate, μ =mortality rate (we nemen voor het gemak aan dat de bevolking constant is door te stellen dat $b = \mu = 1/\text{levensverwachting } \lambda$), γ =recovery rate (gelijk aan $1/\text{infectietijd}$) en β =transmission coefficient (geeft aan hoe snel de ziekte zich kan verspreiden). De variabele k is gelijk aan 1 als we rationeel gezien beslissen ons kind bij geboorte te vaccineren en 0 als we beslissen dat niet te doen, en hangt volgens de ongelijkheid hierboven dus af van de subjectieve kans om ziek te worden. Die kans kunnen we uitrekenen door de jaarlijkse kans op ziek worden enigszins naïef te benaderen door het jaarlijks aantal nieuwe ziektegevallen per hoofd van de bevolking (de *incidence*, deze is gelijk aan $\beta IS/N$ als de eenheid van het model jaren is). Als we dan voor het gemak nog aannemen dat de kansen om in verschillende jaren ziek te worden onderling onafhankelijk zijn (gegeven dat je nog niet besmet was), dan is op ieder tijdstip t de kans dat het kind gedurende zijn leven niet ziek wordt gelijk aan $(1 - \beta I(t)S(t)/N)^\lambda$. In werkelijkheid zal men het exacte aantal nieuwe ziektegevallen per jaar op het tijdstip t niet meteen kennen, maar zal men beslissingen nemen op grond van de geobserveerde ziektegevallen in het verleden, waarbij we kunnen verwachten dat observaties uit het recente verleden zwaarder meetellen dan die uit het verdere verleden. Dit kunnen we modelleren door de *exponential smooth Inc*(t)* van de functie $\beta I(t)S(t)/N$ te gebruiken.⁷ Dit alles leidt tot de volgende vergelijking voor de perceptie van het risico p_z :

$$p_z(t) = 1 - (1 - \text{Inc}^*(t))^\lambda$$

We krijgen dus $k=1$ als $p_z > (C_v + p_b C_b)$, en anders $k=0$. Het stelsel vergelijkingen kan vanwege de niet-lineaire termen niet analytisch opgelost worden, maar is wel eenvoudig numeriek op te lossen door bijvoorbeeld gebruik te maken van Euler integratie. Figuur 1 geeft het model schematisch weer in

een *stock-and-flow diagram*, met daaronder een grafiek van het aantal nieuwe ziektegevallen per jaar en de waarde van k ($k=1$ op de grijze gebieden en $k=0$ elders). Het model begint in evenwichtstoestand.⁸ De perceptie van het risico is in die toestand hoog, zodat men besluit kinderen te vaccineren. Door het succes van de vaccinatie wordt het risico echter spoedig kleiner, waardoor het rationeel gezien beter wordt om niet te vaccineren. Dat gaat een tijd lang goed, tot zo veel ontvankelijke kinderen geboren zijn dat er een nieuwe epidemie ontstaat. Hierdoor gaat de perceptie van het risico weer omhoog tot vaccineren opnieuw de rationele keus wordt. Zo schommelen we door naar een oscillerende evenwichtstoestand. Deze toestand komt voort uit rationele vaccinatiebeslissingen en leidt, in verwachting, tot een economisch optimale graad van vaccinatie.

Maar vanuit het perspectief van de volksgezondheid is het natuurlijk zeker niet wenselijk om regelmatig wederkerende epidemieën te moeten bestrijden. In het bijzonder is deze uitkomst niet optimaal als het mogelijk is een ziekte volledig uit te roeien. Als uitroeiing voltooid is worden immers zowel de toekomstige kosten als de toekomstige ziektegevallen (bijna) gelijk aan 0, terwijl het resultaat uit de figuur tot in het oneindige tot ziektegevallen en kosten leidt. We zien dus dat rationeel gedrag, weliswaar gebruikmakend van een eenvoudig mentaal model voor de risico's⁹ tot antivaccinatie gevoelens kan leiden en dat deze gevoelens paradoxaal genoeg een direct gevolg zijn van het succes van vaccinaties zelf. Als gevolg hiervan kunnen ziektes die we door vaccinaties onder controle hadden gekregen terugkeren en kunnen uitroei-initiatieven falen. Tot nog toe is uitroeiing van 5 menselijke ziektes ondernomen. Slechts bij één is het gelukt (pokken). Twee initiatieven zijn afgeblazen (*malaria* en *framboesia*), deels doordat de prioriteit van de initiatieven afnam na aanvankelijk succes. Twee initiatieven zijn al jarenlang dicht bij hun doel (*polio* en *dracunculiasis*) maar onder voortdurende financiële druk, onder meer vanwege de perceptie dat de kosten per vermeden ziektegeval te hoog worden.¹⁰ Wiskundige modellen kunnen een positieve bijdragen leveren op dit gebied door op het eerste gezicht onintuïtief gedrag te verklaren en beleid te identificeren dat, met inachtneming van alle onzekerheid, op de lange termijn het meest wenselijk is. **U**

Voetnoten

¹ http://www.euro.who.int/features/2007/featureiw07/20070410_12

² Het bedrag is in Amerikaanse dollars van het jaar 2002 en afkomstig uit Thompson KM, Duintjer Tebbens RJ. Retrospective cost-effectiveness analyses for polio vaccination in the United States. *Risk Analysis* 2006 December;26(6):1423-40

³ Als je bijvoorbeeld onder "vaccination" zoekt via google, kom je na Wikipedia op <http://www.whale.to/vaccines.html> en daarna op de Vaccine Liberation Homepage (<http://www.vaclib.org/>). Beide barsten van de misleidende informatie, uit het verband gehaalde opmerkingen en suggestief materiaal tegen vaccinaties.

⁴ Om bijvoorbeeld met 80% zekerheid te kunnen vaststellen of een vaccin minder dan 0.1% kans heeft op ernstige bijwerkingen bij vrouwen in het eerste trimester van hun zwangerschap (met een 5% significantie niveau), moet je meer dan 12 000 bereidwillige proefpersonen uit die demografische groep in je studie zien te vinden.

⁵ Cooke RM. *Experts in uncertainty: opinion and subjective probability in science*. New York: Oxford University Press, 1991

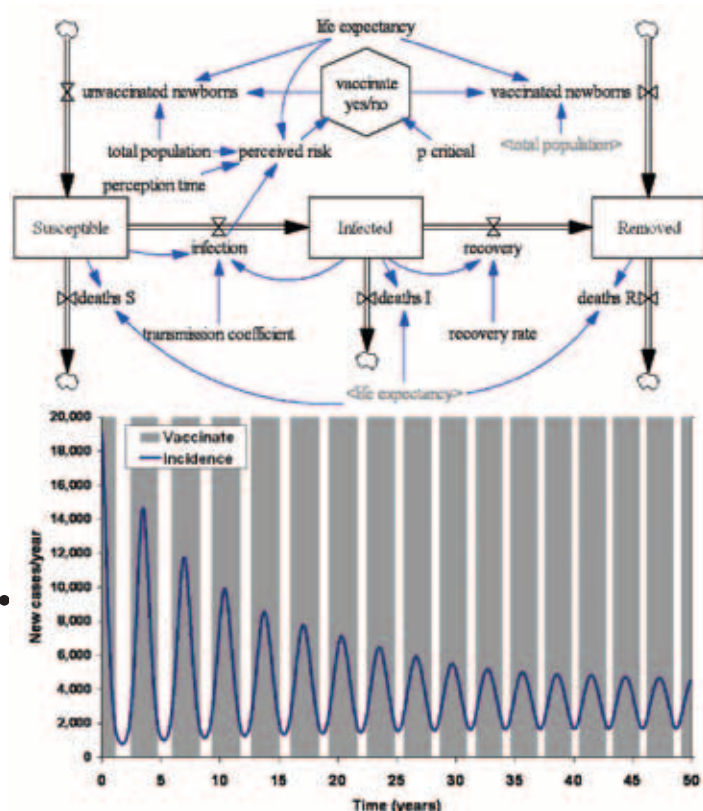
⁶ Dit eenvoudige model heet het SIR model. Er bestaat een groot aantal variaties op dit model om de verspreiding van specifieke ziektes in meer detail of realistischer te simuleren.

⁷ De *smooth* volgt uit $d\text{Inc}^*(t)/dt = (\beta I(t)S(t)/N\text{Inc}^*(t))/\tau$, waarbij τ de gemiddelde tijd is waarover we ons beeld van het aantal nieuwe gevallen vormen (de *perception time*, deze kiezen we voor het gemak gelijk aan 1 maand).

⁸ Deze is gemakkelijk uit te rekenen door de drie afgeleiden in het stelsel (*) gelijk aan 0 te stellen en S , I en R uit het verkregen stelsel op te lossen.

⁹ Zo is de jaarlijkse kans om ziek te worden voor *susceptibles* hoger dan het jaarlijks aantal nieuwe gevallen per hoofd van de bevolking die immers ook *removeds* bevat.

¹⁰ Duintjer Tebbens RJ, Thompson KM. Priority shifting and the dynamics of managing eradicable infectious disease. *Management Science* 2009 April 4;55(4):650-63





All Options

What do you see?

A rock climber?



**We see focus, confidence, teamwork and eagerness.
Exactly what successful traders at All Options need.**

Take the next step.

www.alloptions.nl



Fast Iterative Solution of Large Linear Systems on Grid Computers

Tijmen P. Collignon

Solving extremely large sparse linear systems of equations is the computational bottleneck in a wide range of scientific and engineering applications. Examples include simulation of air flow around wind turbine blades, weather prediction, option pricing, and Internet search engines. Although the computing power of a single processor continues to grow, fundamental physical laws place severe limitations on sequential processing.

This fact accompanied by an ever increasing demand for more realistic simulations has intensely stimulated research in the field of *parallel and distributed computing*. By combining the power of multiple computers with sophisticated numerical algorithms, simulations can be performed that perfectly imitate physical reality.

Traditional parallel processing was and is currently performed using sophisticated supercomputers, which typically consist of thousands of identical processors linked by a high-speed network. They are often purpose-built and highly expensive to operate, maintain, and expand.

A poor man's alternative to massive supercomputing is to exploit existing non-dedicated hardware for performing parallel computations. With the use of cost-effective commodity components and freely available software, cheap and powerful parallel computers can be built. The *Beowulf* cluster technology is a good example of this approach [1]. A major advantage of such technology is that resources can easily be replaced and added. However, this introduces the problem of dealing with *heterogeneity*, both in machine architecture and in network capabilities. The problem of efficiently *partitioning* the computational work became an intense topic of research.

The nineties of the previous century ushered in the next stage of parallel computing. With the advent of the Internet, it became viable to connect geographically separate resources -- such as individual desktop machines, local clusters, and storage space -- to solve very large-scale computational problems. In the mid-1990s the SETI@home project was conceived, which has established itself as the prime example of a so-called *Grid computing* project. It currently combines the computational power of millions of personal computers from around the world to search for extraterrestrial intelligence by analysing massive quantities of radio telescope data [2].

The fact that in Grid computing resources are often geographically separated implies that *communication* is very expensive. As a result, *global synchronisation* is the critical bottleneck operation, since this requires communication between *all* the processes. In particular, Grid computing is naturally suited for so-called *embarrassingly parallel* applications where the problem can be broken up easily and tasks require little or no interprocessor communication. An example of such an application is the aforementioned SETI@home project.

For the numerical solution of linear systems of equations, matters are far more complicated. One of the main reasons is that inter-task communication is both unavoidable and abundant. For this application, developing efficient parallel numerical algorithms for dedicated homogeneous systems is a difficult problem, but becomes even more challenging when applied to heterogeneous systems. In particular, the heterogeneity of the computational resources and the variability in network performance present numerous algorithmic challenges.

Classical Iterative Methods

Our main goal is to efficiently solve in parallel large algebraic linear systems of equations $Ax = b$ on large and geographically separated networks of computers. That is, we want to obtain A^{-1} , where A denotes the coefficient matrix, b represents the right-hand side vector, and x is the vector of unknowns.

As the demand for more realistic simulations continues to grow, the use of so-called *direct methods* for the solution of linear systems becomes increasingly infeasible. This leaves *iterative methods* as the only practical alternative.

The main characteristic of iterative methods is that at each iteration step, information from one or more previous iteration steps is used to find an increasingly accurate approximation to the solution. To be more precise, given an initial guess to the solution $x^{(0)}$, the classical iteration for solving the system $Ax = b$ is

$$x^{(t+1)} \leftarrow x^{(t)} + M^{-1}(b - Ax^{(t)}), \quad t = 0, 1, \dots \quad (1)$$

where M^{-1} can be chosen freely and serves as an approximation for A^{-1} . The matrix M is called the *preconditioner* for the iteration process (1). Setting $M^{-1} = A^{-1}$ would give the solution after a single iteration step, but this would require computing the inverse of the matrix A , which is exactly what we want to accomplish!

Computing the inverse of the matrix M should be both cheap and effective, which is reflected in the different choices for M . The simplest option would be to choose the identity matrix for M , which results in the classical *Richardson* iteration. Another variant is the classical *Jacobi* iteration, which is obtained by taking for M the diagonal matrix having entries from the diagonal of A .

Choices that in some sense approximate the matrix A more accurately naturally result in methods that converge to the solution in less iterations. However, inverting the matrix M will be more expensive in most cases and it is clear that some form of trade-off is necessary.

Efficient preconditioning on grid computers

Finding an efficient and effective preconditioner is crucial for fast convergence of iterative methods. Generally speaking, a preconditioner is the most difficult part to parallelise, especially in heterogeneous environments as found in Grid computing.

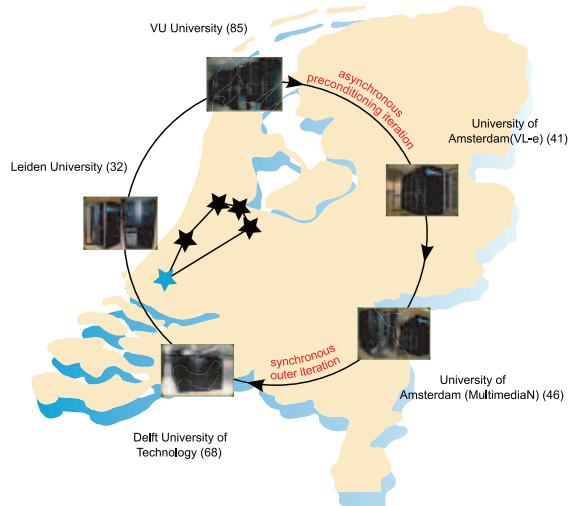


Figure 1: DAS-3: five clusters, one system. The physical locations of the clusters is indicated by the stars.

Classical iterations such as (1) can be generalised quite easily to so-called *a-synchronous* iterative algorithms, which *lack* global synchronisation points and are therefore perfectly suited to Grid computing. However, similar to classical iterative methods, they are extremely *slow to converge* to the solution.

Iterative *subspace* methods are a class of iterative methods that exhibit significantly improved convergence speeds. Some popular subspace methods are: the Conjugate Gradient method, GCR, GMRES, Bi-CGSTAB, and the recently proposed $IDR(s)$ method, which was developed in the Numerical Analysis group at Delft University of Technology. The $IDR(s)$ method is a highly efficient new subspace method for solving large nonsymmetric linear systems [3].

An inherent characteristic of all subspace methods is that they have a large number of global synchronisation points, which makes them less suitable to Grid computing. We propose using an asynchronous iterative method as a *preconditioner* in a *flexible* subspace method, where the preconditioner is allowed to change in each iteration step. By combining a slowly converging asynchronous inner method and a fast converging synchronous outer method, we aim to reap the benefits and awards of both techniques [4, 5, 6].

De-synchronising the preconditioning phase in this manner has the advantage that: (i) the preconditioner can be easily and efficiently parallelised on Grid computers, (ii) no additional synchronisation points are introduced, and (iii) by devoting the bulk of the computational effort to the preconditioner, the computation to communication ratio can be improved significantly, while reducing the number of expensive (outer) synchronisations considerably.

Experiments on the DAS-3 multi-cluster

Our target hardware is the distributed ASCI Supercomputer~3 (DAS-3), which is a cluster of five geographically separated clusters, located at four academic institutions in the Netherlands [7]. The DAS-3 is designed for dedicated parallel computing and although each separate cluster is relatively homogeneous, the system as a whole can be considered heterogeneous. Figure 1 shows the locations of the clusters in the Netherlands, as well as the number of nodes in each cluster. Since the five clusters are geographically separated, global synchronisation is very expensive.

Figure 2 shows total computing times on the DAS-3 for a realistic 3D test problem. Here, the number of equations per node is *fixed* so that the total problem size grows with the number of nodes. The nodes are divided equally between the five clusters. In this experiment, our new asynchronous preconditioning approach (blue lines) is compared to (standard) synchronous preconditioning (red line). The results clearly show that for *larger* number of nodes (or equivalently, for a larger problem size), synchronous preconditioning is *outperformed considerably* by asynchronous preconditioning.

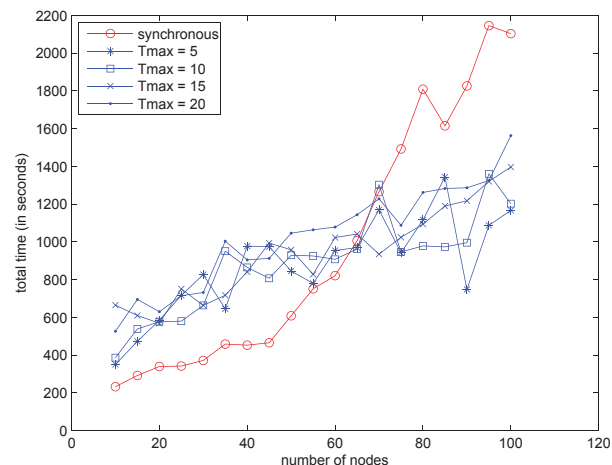


Figure 2: The red line is (standard) synchronous preconditioning, while blue lines are the (new) a-synchronous preconditioning. The different values for T_{max} indicate how much asynchronous preconditioning is performed.

Concluding remarks and further reading

Designing efficient numerical algorithms for Grid computing is a complex process that brings together many different scientific disciplines. By using an asynchronous iterative method as a preconditioner in a flexible iterative method, an algorithm is obtained that has the potential to reap the benefits and awards of both cluster and Grid computing.

For the interested reader, the book by Dimitri Bertsekas and John Tsitsiklis contains a wealth of information on parallel asynchronous iterative algorithms for various applications [8]. Furthermore, more extensive discussions on various aspects of parallel scientific computing may be found in the excellent book by Rob Bisseling [9]. For a comprehensive discussion on iterative methods for solving linear systems, the classic book by Gene Golub and Charles van Loan is greatly recommended [10], as well as the more recent book by Henk van der Vorst [11].

The work presented here has been performed under the supervision of Martin B. van Gijzen.

References

- [1] Thomas Sterling, Ewing Lusk, and William Gropp, editors. Beowulf Cluster Computing with Linux. MIT Press, Cambridge, MA, USA, 2003.
- [2] David P. Anderson, Jeff Cobb, Eric Korpela, Matt Lebofsky, and Dan Werthimer. SETI@home: an experiment in public-resource computing. *Commun. ACM*, 45(11):56–61, 2002.
- [3] Peter Sonneveld and Martin B. van Gijzen. $IDR(s)$: a family of simple and fast algorithms for solving large nonsymmetric linear systems. *SIAM J. Sci. Comput.*, 31(2):1035–1062, 2008.
- [4] Tijmen P. Collignon and Martin B. van Gijzen. Solving large sparse linear systems efficiently on Grid computers using an asynchronous iterative method as a preconditioner. Technical report, Delft University of Technology, Delft, the Netherlands, 2008. DUT report 08–08.
- [5] Tijmen P. Collignon and Martin B. van Gijzen. Fast iterative solution of large sparse linear systems on geographically separated clusters. Technical report, Delft University of Technology, Delft, the Netherlands, 2009. DUT report 09–12.
- [6] Tijmen P. Collignon and Martin B. van Gijzen. Parallel scientific computing on loosely coupled networks of computers. In B. Koren and C. Vuk, editors, *Advanced Computational Methods in Science and Engineering*, volume 71, pages 79–106. Springer Berlin Heidelberg, 2010.
- [7] Frank J. Seinstra and Kees Verstoep. DAS-3: The distributed ASCI supercomputer 3, 2007. <http://www.cs.vu.nl/das3/>.
- [8] Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Englewood Cliffs, N.J., 1989. republished by Athena Scientific, 1997.
- [9] Rob H. Bisseling. *Parallel Scientific Computation: A Structured Approach Using BSP and MPI*. Oxford University Press, 2004.
- [10] Gene H. Golub and Charles F. Van Loan. *Matrix Computations* (Johns Hopkins Studies in Mathematical Sciences). The Johns Hopkins University Press, October 1996.
- [11] Henk A. van der Vorst. *Iterative Krylov Methods for Large Linear systems*. Cambridge University Press, Cambridge, 2003.



Vehicle Routing Problems with time-dependent travel times

Ester Stegers

Congestion creates a substantial variation in travel speeds during morning and evening rush hours. This is problematic for all vehicle routing models that rely on a constant value to represent vehicle speeds.

Urban route designs that ignore these significant speed variations result in inefficient, unrealistic and suboptimal solutions. These poorly designed routes that lead freight vehicles into congested urban traffic, result in some cases in a waste of valuable time, and customers have to wait unreasonably long without having any reliable information about the actual arrival times of the vehicles. In these circumstances, it becomes difficult to satisfy the time windows during which the customers must be visited. This increases supply chain and logistics cost.

The Vehicle Routing Problem

The Vehicle Routing Problem (VRP) is one of the most challenging combinatorial optimisation problems. It consists of designing the optimal set of routes for a fleet of vehicles parked at a central depot in order to service a given set of customers with a fixed demand within given time windows. The VRP belongs to the category of NP-hard problems, meaning that the computational effort required to solve this problem increases exponentially with the problem size in the worst case. A small VRP (up to 100 customers) is solvable with exact algorithms, but for a large VRP (more than 100 customers) it is desirable to obtain approximate solutions using heuristics. These methods often give routes of sufficient quality within reasonable computation time when exact algorithms are not fast enough.

TNO Business Unit Mobility and Logistics has designed a logistics optimisation model to handle VRPs, called RESPONSE™. The objective of this model is first to minimise the number of vehicles and secondly to minimise the total route durations. The model is based on an evolution strategy proposed by Homberger and Gehring [1]. The evolution strategy is a technique inspired by

Darwin's principle of natural selection. From a population of solutions, a new generation will be created. Like in the evolution theory, the idea is that the better solutions will survive. However, there are a lot of other techniques to handle the VRP.

Problem investigation

In order to obtain a more realistic solution for the VRP, a method should be created that deals with time-dependent travel times. Van Rijswijk [2] reported on the need for complex optimisation algorithms and the need for time dependent measurement data. This research focuses on the first. Not only does the optimisation algorithm need to calculate a low cost feasible solution, it also has to do this efficiently. Efficiency becomes critical for the large problems often occurring in a logistics company.

Solution method

From a literature review, interesting algorithms are explored to handle the VRP with time-dependent travel times. It turns out that the evolution strategy is not so efficient any more. The most promising solution method, a Construction and Route Improvement Algorithm (CRIA), based on the work of Figliozzi [3] is chosen for further investigation. The advantage of this method is that improvements are done on route level. In the evolution strategy, local improvement heuristics are used, where orders will be interchanged. Without time dependencies this is no problem, since interchanging two orders has no effect on the rest of the route. However, with time-dependencies, the whole route after the changed order may have been affected and has to be recalculated.

To handle time-dependent travel times, time-zones are introduced, periods that classify the day (morning rush hours, off-peak hours, evening rush hours). Every time-zone has its own speeds over the network. Note that the use of 1 time-zone equals using no time-dependent travel times. The more time-zones used, the better the model reflects reality.

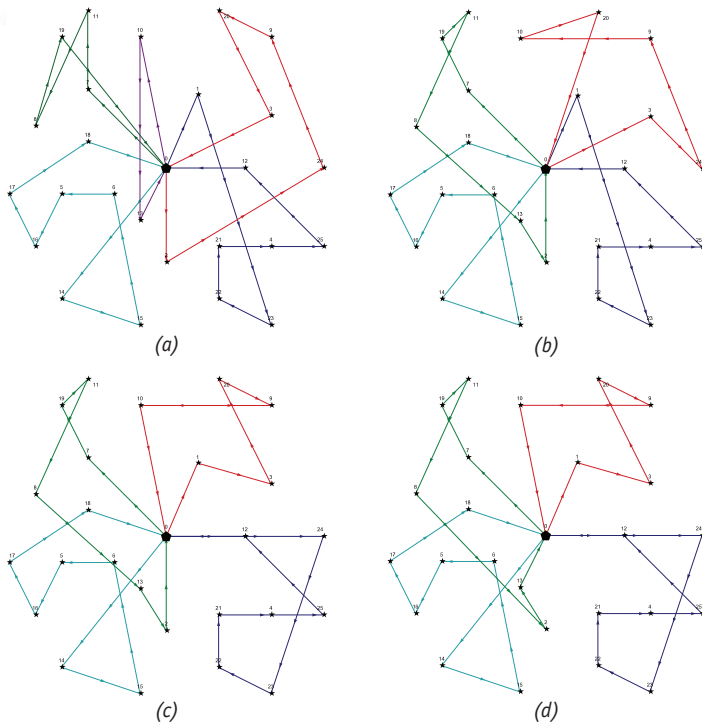


Figure 1: Solution of a VRP created by the CRIA. a: first solution created by the Construction Algorithm; b: improvement on the first solution; c: improvement on the second solution; d: final solution.

The CRIA is integrated with a self developed time-dependent shortest path algorithm that will be executed during the VRP calculation. Due to varying travel times, paths (with different distances) may result in a shorter path (in time) between customers at different times of the day. Using the time-dependent shortest path algorithm, these shorter paths may be found. In practise, this approach allows for the creation of routes that avoid congested areas during rush hours. No other published work using a similar algorithm is known to exist. Most researchers execute Dijkstra's shortest path algorithm before the start of the VRP calculation using average speeds. The computational complexity of the CRIA is analysed and experimental results indicate that the computation time increases proportionally to the number of time-zones, as well as to the square of the number of customers. The use of the time-dependent shortest path algorithm gives more realistic solutions, but for large problems (more than 100 customers and 5 time-zones) the computation time becomes a critical factor.

The CRIA integrated with the time-dependent shortest path algorithm is implemented and tested on one of the Solomon's benchmark problems (R103) proposed by Solomon [4]. Figure 1 gives an example of how CRIA finds a solution. A first solution, created by the construction algorithm can be like Figure 1(a), where each coloured line indicates a route. The next step is to take orders of several routes together and optimise only these orders. As can be seen in Figure 1(b), the two routes at the bottom remain the same, but the three other routes are reordered, such that all orders can now be served in merely two routes. Next, all orders of the two most right routes can be reordered leading to the more efficient solution shown in Figure 1(c). Finally, improvement of each individual route is tried, and this gives the final solution as represented in Figure 1(d).

Results real-life problem urban agglomeration The developed solution method is also tested on a real-life problem. This problem contains a part of The Netherlands, i.e., the urban agglomeration that stretches from Rotterdam to Amsterdam with a surface of about 60 km². The network contains 1,000 links and the problem contains 1 depot and 35 customers. Because there is no dynamic traffic data available (yet) that can be read by the model, the travel

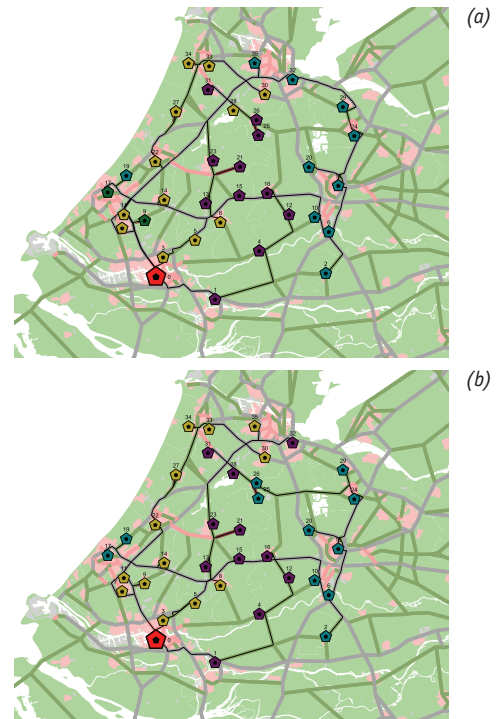


Figure 2: Results real-life problem urban agglomeration. a: calculated using 1 time-zone; b: calculated using 5 time-zones.

speed variations are simulated. As can be seen in Table 1, the solution method gives a quite good solution. When the average speed of the single time-zone is split into five time-zones with peak and off-peak speeds, the solution becomes better, i.e. less vehicles and less travel time.

Figure 3 gives a visualisation of the solution, by drawing the travelled links. The preference for highways is clear for the solution created with one time-zone as well as for the solution created with five time-zones, but for the five time-zones solution also the avoidance of congested areas can be seen. A part of the Dutch highways A1 and A4 are avoided and secondary roads, which are less congested are used.

Conclusions and recommendations

This research shows that when dealing with time dependent scenarios, the solutions generated by a non time-dependent algorithm may become infeasible. Moreover there is a possibility to find solutions that are better than the single time-zone solutions. The advantages of the developed solution method is clear: the possibility to generate good quality solutions of the VRP that better reflect reality. With the development of the CRIA, it will be easier to extend the VRP model in other related areas, like optimising for minimal cost by taking kilometre-pricing into account. When time-dependent measurement data connected to the developed model becomes available, the model can be used in practise. Then, by taking into account varying traffic situations, companies can reduce their logistics cost. 

References

- [1] J. Homberger and H. Gehring, *Two evolutionary metaheuristics for the vehicle routing problem with time windows*, INFOR, 1999.
- [2] H. van Rijswijk, *Dynamisch netwerk RESPONSE™*, TNOReport, TNO Built Environment and Geosciences, Business Unit Mobility and Logistics, the Netherlands, 2008.
- [3] M.A. Figliozzi, *A route improvement algorithm for the vehicle routing problem with time dependent travel times*, Paper under review for journal publication and also available in the proceedings of the 88th Transportation Research Board annual meeting, Washington DC, USA, 2009.
- [4] M. Solomon, *Solomon's benchmark*, <http://web.cba.neu.edu/~msolomon/problems.htm>, 1983.



Historisch Persoon

Kurt Gödel

Michiel van Dam

Kurt Gödel was een Oostenrijks-Amerikaanse logicus, wiskundige en filosoof, die in de twintigste eeuw een enorme invloed heeft gehad in de wetenschap en het filosofisch denken. In de tijd dat hij leefde waren andere wiskundigen het gebruik van logica en verzamelingenleer aan het verkennen, om daarmee de basis van de wiskunde te kunnen begrijpen.

Ein Österreicher im Exil in der Tschechoslowakei

Kurt Friedrich Gödel werd op 28 april 1906 geboren in Brno, een stad in het zuiden van Tsjechië, destijds Oostenrijk-Hongarije. Hij was de zoon van Rudolf en Marianne Gödel en zijn vader was bedrijfsleider bij een textielfabriek. Hij groeide op met de Duitse taal omdat zijn stad een kleine meerderheid Duits-talige inwoners had. Volgens zijn broer leed Kurt op jonge leeftijd aan reuma, wat later volledig is genezen. Toch heeft hij altijd gedacht dat er daardoor blijvende hartschade is ontstaan.

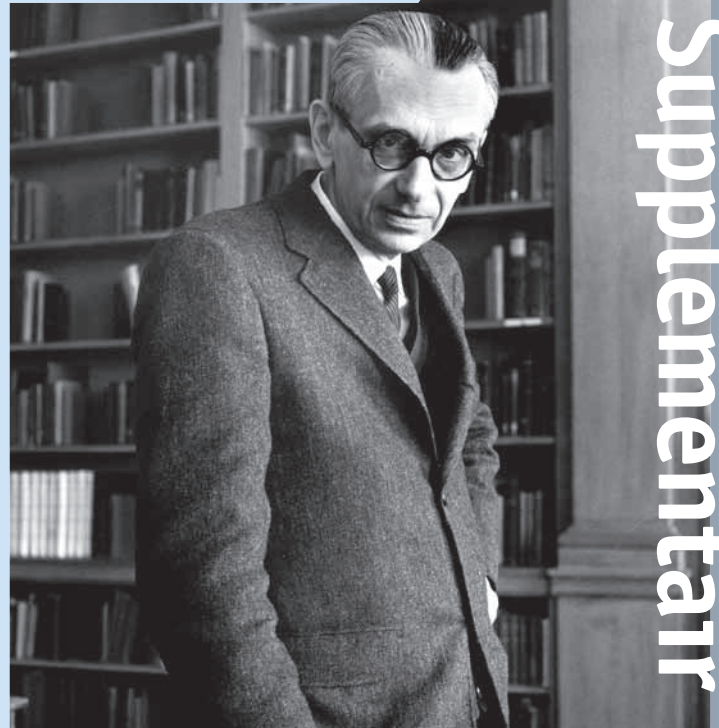
Op het moment dat Oostenrijk-Hongarije ophield te bestaan, aan het einde van de Eerste Wereldoorlog, werd hij automatisch Tsjecho-Slowaaks staatsburger, waar hij niet blij mee was. Hij voelde zich als een "verbannen Oostenrijker in Tsjecho-Slowakije." ("ein Österreicher im Exil in der Tschechoslowakei") Toen hij 23 was koos hij er daarom voor om Oostenrijks staatsburger te worden. Toen aan het begin van de Tweede Wereldoorlog Oostenrijk geannexeerd werd door Duitsland, werd hij daarmee automatisch Duits staatsburger, toen hij 32 was. 10 jaar later, na de Tweede Wereldoorlog, werd hij ook nog Amerikaans staatsburger. Hij trouwde in 1938 met Adele Nimbursky, waar hij geen kinderen mee gekregen heeft.

In zijn jonge jaren stond Kurt bekend in zijn gezin als Herr Warum, vanwege zijn onverzadigbare nieuwsgierigheid. Van 1912 tot 1916 ging Gödel naar de Evangelische Volksschule in Brno, een Lutherse school, en van 1916 tot 1924 ging hij naar het Deutsche Staats-Realgymnasium waar hij over de hele linie uitblonk, in het begin vooral in talen en godsdienst, maar later kreeg hij meer belangstelling voor geschiedenis en wiskunde. Zijn belangstelling voor wiskunde nam nog meer toe toen zijn oudere broer Rudolph in 1920 naar Wenen vertrok om medicijnen te gaan studeren aan de Universiteit van Wenen. Tijdens zijn tienerjaren bestudeerde Gödel al de Gabelsberger stenografie, Goethe's kleurenleer, de kritieken van Isaac Newton en de geschriften van Immanuel Kant.

Studie

Toen Kurt 18 geworden was, ging hij net zoals zijn broer naar de universiteit van Wenen, hoewel hij het universitaire wiskundeniveau op dat moment al meester was. Hij was aanvankelijk van plan om theoretische natuurkunde te gaan studeren, maar ging ook naar wiskunde- en filosofiecolleges. In de tijd dat hij daar studeerde verdiepte hij zich in wiskundig realisme, hij las Kants 'Metaphysische Anfangsgründe der Naturwissenschaft' en deed mee met de 'Wiener Kreis' met Schlick, Hahn en Carnap. Gödel bestudeerde toen ook getal-

Figuur 1: Kurt Gödel, 1962



theorie, maar nadat hij een cursus van Moritz Schlick gevolgd had, die het boek 'Introduction to Mathematical Philosophy' van Bertrand Russell behandelde, raakte hij geïnteresseerd in wiskundige logica.

In 1928 publiceerde Hilbert en Wilhelm Ackermann 'Grundzüge der theoretischen Logik', een inleiding tot de logica van de eerste orde, waarin het volledigheidprobleem aan de orde werd gesteld: Zijn de axioma's van een formeel systeem voldoende om daar elke bewering uit af te leiden, die waar is in alle modellen van het systeem? Dit was het onderwerp dat Gödel koos voor zijn

Wiener Kreis

De Wiener Kreis was een groep filosofen en wetenschappers die logisch positivisme of logisch empirisme propageerde. Deze groep hanteerde hierdoor een tweetal regels:

1. Ervaring is de bron van alle kennis.

2. Logische analyse door middel van symbolische logica is de beste manier om filosofische problemen op te lossen.

Eenheid

Het belangrijkste en laatste doel van de Wiener Kreis was om de wetenschap te unificeren, een basaal systeem op te zetten waarbinnen elke geldige uitspraak teruggebracht wordt naar een lager niveau, waarbinnen die uitspraak direct op ervaring terug te voeren valt. Onderdeel hiervan was een zoektocht naar verduidelijking en een symbolische taal die niet onderhevig was aan alle onduidelijkheden van natuurlijke taal.

Gödels onvolledigheidsstellingen bouwen op een vrij eenvoudig idee. In de stelling wordt een formule geconstrueerd die van zichzelf zegt dat hij onbewijsbaar is. Op het moment dat die stelling bewijsbaar zou zijn, zou hij zichzelf tegenspreken en een paradox opleveren. Hierdoor weet je zeker dat de stelling waar is, maar desondanks nooit bewezen kan worden. Dat vertaalde zich in een meer algemene stelling: voor elke berekenbare opeelbare groep axioma's voor de rekenkunde, bestaat er een formule die geldig is in die rekenkunde, maar niet bewijsbaar is.

De Principia Mathematica is een boekwerk over de absolute basis van wiskunde, gepubliceerd door Whitehead en Russel in 1910, 1912 en 1913. Het probeert alle wiskundige waarheden af te leiden van een gedefinieerde en beperkte set axioma's, door middel van inferentie en symbolische logica. Het bleek echter, dankzij Gödels werk, dat de Principia Mathematica onmogelijk consistent en compleet tegelijk kon zijn. Er zal altijd een uitspraak zijn die waar is binnen het systeem, maar niet binnen het systeem bewezen kan worden.

Desondanks heeft Gödel in zijn ‘volledigheidsstelling’ aangetoond dat semantische waarheid en syntactische bewijsbaarheid samengaan in predikaatanalyse van de eerste orde. Dit houdt in dat het mogelijk is om alle logisch geldige consequenties van een predikaatstelling te vinden, simpelweg door alle mogelijke afleidingen af te gaan, gebruikmakend van axioma’s in de theorie. Deze volledigheidsstelling kan bestaan naast de onvolledigheidsstelling, doordat de onvolledigheidsstelling over de predikaatstelling zelf gaat en de volledigheidsstelling slechts over alle logische gevolgen van deze stelling. Alle logische gevolgen zijn dus in dit geval bewijsbaar, de stelling zelf hoeft dit niet te zijn.

Gödel bleef werken in Wenen, en in 1931 publiceerde Gödel zijn beroemde onvolledigheidsstellingen in 'Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme'. In dat artikel bewees hij voor elk berekenbaar axiomatisch systeem, dat krachtig genoeg is, dat het systeem niet tegelijkertijd consistent en volledig kan zijn, maar ook dat de consistentie van axioma's niet bewezen kan worden binnen het systeem. Deze stellingen zorgden ervoor dat een halve eeuw aan inspanningen vergeefs bleken te zijn, waaronder het werk van Gottlob Frege, de Principia Mathematica van Whitehead en Russel en Hilberts formalisme. Deze stellingen houden ook in dat niet alle wiskundige vraagstukken berekenbaar zijn.

Gödel behaalde in 1932 zijn habilitatie aan de Universiteit van Wenen, waardoor hij zich vanaf dat moment privaattoecent mocht noemen. In 1933 kwam Adolf Hitler aan de macht en kregen de Nazi's steeds meer invloed in Oostenrijk en bij de Weense wiskundigen. Toen in juni 1936 Moritz Schlick vermoord werd door een pro-Nazi student, kreeg Gödel een ernstige zenuwcrisis en werd hij paranoïde. Vanaf dat moment was hij altijd bang om vergiftigd te worden en verbleef hij een aantal maanden in een sanatorium voor zenuwziekten.

In 1933 ging Gödel voor het eerst naar de VS en hield daar een toespraak op de jaarlijkse bijeenkomst van de American Mathematical Society. Tijdens dat jaar kreeg hij ook zijn ideeën over de berekenbaarheid en recursieve functies, en het idee waarheid. Dit werk werd daarna in de getaltheorie verder ontwikkeld, gebruikmakend van Gödelnummering.

In 1934 gaf Gödel een serie lezingen in Princeton, New Jersey, Over onbeslisbare stellingen van formele wiskundige systemen. Het verslag van die lezingen bij het Institute for Advanced Study (IAS) is later gepubliceerd. In 1935 wilde hij nog een keer naar het IAS, maar het reizen en harde werken putte hem dusdanig uit, dat hij daarna een jaar niet heeft gewerkt, maar uitrustte. In 1937 ging hij verder met zijn college en werkte hij ook aan het bewijs van de consistentie van het keuzeaxioma en van de continuümhypothese (CH), hij ging aantonen dat deze hypothesen niet weerlegd kunnen worden binnen het gebruikelijke systeem van axioma's van verzamelingenleer.

Gödelnummering is een manier om een uniek getal aan ieder symbool of iedere formule toe te kennen, binnen een formele taal. Aan ieder symbool wordt een getal toegekend, waardoor elke serie uitspraken gerepresenteerd kan worden als een reeks natuurlijke getallen.

Gödel zelf gebruikte een systeem van Gödelnummering met ontbinding in priemfactoren. Hij kende aan elk symbool een opeenvolgend getal toe. In de wiskundige uitspraak nummerde hij de opeenvolgende symbolen met de priemgetallen, en verhefte die priemgetallen tot de macht van het getal dat dat type symbool eerder kreeg. Bij a als 1, b als 2 en $=$ als 3, wordt de uitspraak $a = b$ dus $2^{1^1} 3^{3^2} 5^{5^2}$, dus 1350. Door ontbinding in priemfactoren kan de oorspronkelijke berekening teruggekregen worden.





Figuur 2: Kurt Gödel en Albert Einstein bij Princeton

In 1938 bracht hij opnieuw een bezoek aan het IAS, en het voorjaar van 1939 bracht hij door aan de Universiteit van Notre Dame. Doordat Nazi Duitsland de titel privaattoecent afschafte, moest Gödel solliciteren naar een andere baan in de nieuwe orde. Hierbij vermoedde zijn vroegere omgang met Joodse leden van de Wiener Kreis de procedure, waardoor de universiteit van Wenen zijn sollicitatie afwees. Het werd nog erger toen het Duitse leger hem oproep voor militaire dienst. Daarom vertrok Gödel in 1939, kort na het begin van de Tweede Wereldoorlog, met als bestemming Princeton. Om de blokkades te omzeilen reisde Gödel via de trans-Siberische spoorweg en Japan naar San Francisco, en reisde binnen de VS vervolgens per trein door naar Princeton, waar hij een baan kreeg bij het IAS.

De zomer van 1942 bracht hij door in Blue Hill, in Maine, toen hij bij het IAS vakantie had opgenomen. Hij had echter een zeer productieve zomer, want uit zijn werkschriften blijkt dat tijdens het verblijf in Blue Hill Gödel een bewijs ontdekte voor de onafhankelijkheid van het keuze-axioma vanuit de eindigetype-leer. Het bewijs wordt in zijn werkschriften niet gegeven, maar alleen een uitgebreide behandeling van het probleem.

In 1946 werd Gödel een vast lid van de IAS en in 1953 kreeg hij een volledig hoogleraarschap aan het instituut. Tenslotte werd hij in 1976 emeritushoogleraar.

Door zijn werk heeft hij in 1951 de eerste Albert Einstein Prijs gekregen, en in 1974 kreeg hij tevens de National Medal of Science.

Einstein

Bij zijn eerste reis naar de VS leerde Gödel Einstein kennen, en ze ontwikkelden een legendarische vriendschap. Wat er besproken werd tijdens de vele gezamenlijke wandelingen van en naar het IAS was voor andere leden van het instituut een geheim. Einstein vertelde aan het eind van zijn leven tegen de

econoom Morgenstern dat 'zijn eigen werk niet veel meer betekende, maar dat hij nog alleen maar naar het instituut ging... om het voorrecht te hebben samen met Gödel naar huis te kunnen lopen.'

Gödel werd in 1947, bij zijn Amerikaans burgerschapsexamen, vergezeld door Einstein en Morgenstern als getuigen. Gödel had aan hen meegedeeld dat hij een tegenstrijdigheid in de grondwet had gevonden die ervoor kon zorgen dat de VS een dictatuur werd. Toen Gödel gevraagd werd of hij dacht dat een dictatuur zoals het Nazi regime mogelijk zou zijn in Amerika, begon hij zijn ontdekking uit te leggen, waarmee Einsteins en Morgensterns zorgen over het onvoorspelbare gedrag van Gödel uit bleken te komen. Gelukkig kende de rechter Einstein, en dus werd Gödel snel de mond gesnoerd en werd er overgegaan naar andere vragen.

Gödel heeft in 1951 Einstein voor zijn 70e verjaardag de uitwerking gegeven van het bestaan van paradoxale oplossingen van diens veldvergelijkingen in de algemene relativiteitstheorie. Dit zou tijdreizen mogelijk maken en hierdoor ging Einstein twijfelen aan zijn eigen theorie.

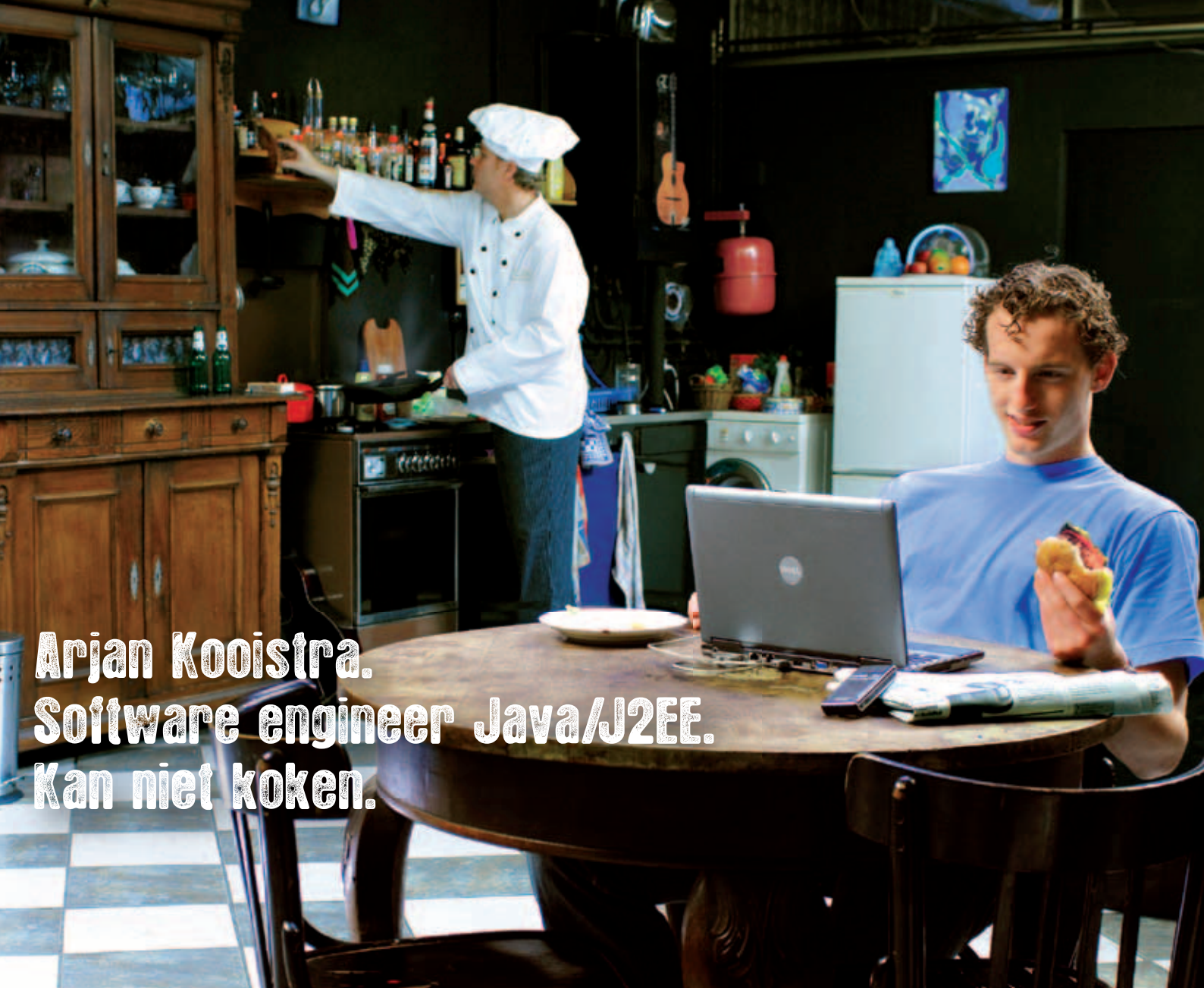
Overlijden

Aan het eind van zijn leven was Gödel geestelijk instabiel en ziek. Zijn paranoïa had zich ontwikkeld tot een obsessieve angst om vergiftigd te worden en hierdoor at hij niks meer, tenzij zijn vrouw het voorgeproefd had. In 1977 werd zijn vrouw Adele zes maanden in het ziekenhuis opgenomen, en in die tijd hongerde Gödel zichzelf dood. Hij overleed op 14 januari 1978 aan 'ondervoeding en uitputting'.

Gödel was zijn hele leven christen. Hij verwierp het idee van een onpersoonlijke God, waar Einstein in geloofde. Hij geloofde in een leven na de dood en stelde: 

"Ik ben overtuigd van een hiernamaals, onafhankelijk van de theologie."

"Als de wereld rationeel in elkaar zit, moet er een leven na de dood zijn."



Arjan Kooistra.
Software engineer Java/J2EE.
Kan niet koken.

Quinity
_com

**Ook zin in een succesweekend
met een privé-kok?**

Als je bij Quinity komt werken, werk je mee aan het ontwikkelen van eBusiness-applicaties. Dat doen we voor grote, financiële organisaties en met goede resultaten. En boeken wij succes, dan boek jij ook succes. Sterker nog: we garanderen je een carrière waarin je veel successen op je naam kunt zetten. Ook als je nog maar net bent afgestudeerd.

En om je daarvan alvast te laten proeven, krijg je van ons een geweldig succesweekend naar keuze aangeboden als we het met elkaar eens worden.

Kijk meteen op www.werkenbijquinity.nl voor alle details en mogelijkheden. En ontdek dat je bij Quinity net zo succesvol kunt worden als je ambities reiken.

Upload meteen je cv.

Quinity zoekt **software engineers Java/J2EE**, **projectleiders**, **functioneel ontwerpers** en **consultants/informatie-analisten**. Als je zo'n baan én een succesweekend wilt, upload dan snel je cv. Ook al heb je nog geen ervaring. Op www.werkenbijquinity.nl vind je uiteraard ook alle andere informatie en wetenswaardigheden over een baan bij ons bedrijf.

Quinity B.V. – Maliebaan 50 – Postbus 13097 – 3507 LB Utrecht
Telefoon +31(0)30 2335999

Werken bij Quinity.
Succes gegarandeerd.



Alumnus

CTO TomTom

Rijn Buve

In 1985 was het zover. Er moest een belangrijke keuze worden gemaakt.

Wat zou ik na mijn Gymnasium doen? Afgezien van een grote persoonlijke

interesse in muziek, was ik ook altijd al door techniek gefascineerd en ik

zag hoe computers de wereld zouden veranderen en zelfs nieuwe werelden

zouden kunnen creëren. Mijn keuzepalet van studiekeuzes had ik daarmee

gereduceerd tot Conservatorium of Informatica. Het werd Informatica

met de stille belofte aan mezelf om toch vooral de creatieve kant daarvan

te bedrijven.

De realiteit van de studie Informatica aan de TU Delft bleek wat anders dan ik me aanvankelijk had voorgesteld. In plaats van het bezig zijn met computer-techniek en programmeren, bestond een groot deel van de studie uit wiskunde en theoretische informatica: twee vakken die op het eerste gezicht zeer ver van het destijds zo lonkende toetsenbord afstonden. "Op het eerste gezicht" zeg ik, want ik weet nog goed hoe docenten als Ruud Sommerhalder en Cees Witteveen keer op keer hun uiterste best deden deze vakken tot leven te laten komen bij de daarvoor ontvankelijke student. Probleem-complexiteitsklassen P en NP, symboolmanipulatie en het ruimte-tijd spanningsveld werden gekoppeld aan alledaagse situaties en actuele politieke problemen om de stof te kunnen relateren aan wat minder abstracte zaken dan de doorgaans Griekse symbolen waarin deze gespecificeerd worden.

Helaas waren de examenresultaten stille getuigen van het feit dat niet alle studenten hiervoor ook inderdaad even ontvankelijk waren. Vele studenten, waaronder ik, konden zich verheugen op een zware onvoldoende voor het eerste tentamen, als bleek dat de stof niet alleen gehoord, maar ook nog begrepen diende te worden. Was dat even pech. Het deed me denken aan de eerste week van het Gymnasium, waarin ik nog naïef hoopte dat als ik het Griekse alfabet eenmaal kende, ik ook daadwerkelijk Grieks kon lezen. Toch ging dat werkelijk begrijpen van de stof me hoe langer hoe meer boeien. Deze docenten waren buitengewoon intelligent. Er moest toch een reden zijn dat ze erin volhardden studenten dermate abstracte stof bij te willen brengen? En langzaam aan werd me duidelijk dat echte revolutionaire technische doorbraken vrijwel altijd gebaseerd zijn op het begrip en de toepassing van deze, soms uiterst abstracte, basiskennis. Je kunt bestaande oplossingen van problemen vaak lange tijd incrementeel verbeteren, maar wil je echt een grote stap maken, dan zul je de basis volledig moeten beheersen om tot nieuwe oplossingen te komen. Met die mooie gedachte heb ik me de theorie eigen gemaakt en daarna besloten dat ik me toch vooral wilde toelagen op de concrete toepassing ervan. Ik nam compilerbouw in mijn vakkenpakket. Dat bleek later een zeer gelukkige keuze te zijn.

Na mijn studie kwam de volgende grote beslissing: wat nu? Zoals ik al eerder zei, zag ik al vroeg hoe computers niet alleen de wereld zouden veranderen, maar ook hoe ze eigen werelden zouden kunnen creëren. Tegenwoordig worden miljoenen mensen meegesleurd door de virtuele 3D wereld van Avatar en worden miljarden verdiend in de vele werelden die in platforms als de Xbox en PlayStation huizen, maar 20 jaar geleden kreeg een aantal mensen, waaronder ikzelf, al kippenvel als ze "You are in a narrow damp tunnel whose walls are fungi encrusted. Enter command?" lazen bij het starten van de text adventure Dungeons. Toch kon dat beter. Visueel. Ik ging daarom werken bij een klein bedrijf in Amsterdam dat destijds zeer geavanceerde 3D programmatuur ontwikkelde. Drie jaar lang ben ik toen met niets anders bezig geweest dan "rendering" en tools maken om virtuele werelden te creëren. Totdat onverwacht iets totaal anders op mijn pad kwam.

In 1995 had het bedrijf CMG (later Logica) de opdracht van Rijkswaterstaat gekregen om de besturingssoftware voor de in aanbouw zijnde stormvloedkering in de Nieuwe Waterweg bij Hoek van Holland te ontwikkelen. Die stormvloedkering was het sluitstuk van het Deltaplan na de watersnoodramp in 1953 en was één van de meest prestigieuze softwareprojecten van die tijd, vooral door de enorm hoge faalkanseisen die eraan werden gesteld. De naam van het project was: "BOS": Beslis- en Ondersteunend Systeem van de Stormvloedkeringen in de Nieuwe Waterweg en het Hartelkanaal.

De projectleider van het BOS-project kende ik goed en ik vond het intrigerend te horen hoe ze dachten deze klus te gaan klaren. Er was een high-level design gemaakt en het echte werk, het volledig op detailniveau ontwerpen en bouwen van alle subsystemen kon bijna beginnen. In het hart van het systeem was een interpreter bedacht, voor een taal die nog ontworpen moest worden. En hoewel dit nog van voor de tijd van Java en virtual machines was, leek het concept er achteraf gezien wel een beetje op. Deze nieuwe taal voor de besturing van de kering, de "Procedure scripttaal" genaamd, diende twee doelen. Ten eerste was het op dat moment nog niet bekend hoe de exacte procedures en draai-boeken voor het sluiten en openen van de keringen eruit zouden zien. Daar zou Rijkswaterstaat nog zeker anderhalf jaar voor nodig hebben. En ten tweede waren de faalkanseisen aan de besturingssoftware zo hoog, dat gemeend werd dat men deze procedurescripts in een taal zou moeten kunnen specificeren die inherent veiliger is dan talen als C of C++. Er was alleen één probleem. Ze hadden nog niemand voor het projectteam kunnen vinden, die een dergelijk taal kon ontwerpen en bouwen. De eerdere keuze voor het vak compilerbouw kwam me nu goed van pas en ik besloot de uitdaging aan te gaan. Zowel het ontwerp van de nieuwe taal als het ontwerp en de bouw van de compiler, de

Figuur 1: 3D Rendering





Figuur 2: Stormvloedkering Oosterschelde

interpreter (die feitelijk de taak van virtual machine op zich nam) en later een debugger werden mijn deel in dit mission critical systeem. En dat werd een zeer leerzaam traject.

Bij de ontwikkeling van het BOS-systeem draaide alles om faalkansen. Een belangrijke politieke afweging voor wel of geen kering, was destijds ofwel te kiezen voor het verzwaren van de dijken over vele tientallen kilometers lengte, met alle kosten en landschapsvervuiling die daarmee gepaard zouden gaan, ofwel het bouwen van een stormvloedkering met gelijkwaardige beschermings-eigenschappen voor het achterland. In Nederland bouwen we al eeuwen dijken en alle veiligheidsaspecten van dijken zijn tot in de puntjes gespecificeerd. Een dijk in Nederland heeft een faalkans van 1 op 1.000. Dat wil zeggen dat op elke 1.000 stormen (van een welgedefinieerde sterkte), een dijk maximaal eenmaal kan bezwijken. Bij een geschatte frequentie van optreden van dergelijke stormen van eens in de 10 jaar, komt dat neer op maximaal één dijkbreuk in 10.000 jaar. Dat was dus ook het uitgangspunt voor de veiligheid en de faalkans van de stormvloedkering.

De stormvloedkering bestaat uit veel, en vooral grote en imposante hardware, zoals de enorme kerende wand, de armconstructies en het kogelgewricht waar de arm om scharniert. Maar daarnaast bestaat hij dus ook uit software. De faalkans van de kering moest derhalve verdeeld worden over deze twee aspecten van het systeem, hardware en software, die beide de betrouwbaarheid beïnvloeden. Faalkanstechnisch geldt dat twee seriegeschakelde componenten gezamenlijk een faalkans hebben die een orde slechter is dan de faalkans elk component afzonderlijk. Hierdoor werd gesteld dat zowel de hardware als de software aan een faalkans moest voldoen die nog een orde hoger was: 1 op 10.000 dus, wat neerkomt op een betrouwbaarheid van 99,99%. Maar daarmee was men er nog niet voor de software. Want deze bevat zowel een component die de kering sluit in geval van storm, als een component die hem weer opent, als de storm voorbij is. En falen de kering te openen na een storm is nog erger dan deze niet te kunnen sluiten. Door de kering niet tijdig weer te openen zou het achterland namelijk alsnog vollopen, maar dan met zoetwater, waarna de kering het uiteindelijk zou begeven en het achterland onbeschermd zou laten liggen voor een volgende storm. De faalkans voor dit scenario werd daarom aangescherpt en gesteld op 1 op 100.000, oftewel een betrouwbaarheid van 99,999%, vergelijkbaar met eisen aan software voor kerncentrales en de Space Shuttle.

Tijdens dit project, en later tijdens een vergelijkbaar project voor de vernieuwing van de besturing van de stormvloedkering Oosterschelde in Zeeland, werd voortdurend duidelijk hoe belangrijk het is om bij kritieke software ontwikkeling niet alleen te letten op de functionaliteit van het systeem ("wat moet het doen?"), maar ook vooral op de niet-functionele aspecten van het systeem ("hoe goed moet hij het doen?"). En dat geldt ook bij minder kritieke software. Immers, tijdens de life cycle van een systeem wijzigen de functionaliteiten die het systeem moet bieden doorgaans veel vaker dan de eisen die aan de kwaliteit van het systeem worden gesteld.

Een andere belangrijke les was hoe je software betrouwbaar en voorspelbaar binnen budget kunt ontwikkelen. Hoe weet je of een systeem inderdaad zo betrouwbaar is als geëist werd? En hoe voorkom je dat je eindeloos blijft testen bent om dat te halen of aan te tonen? Bij "normale" faalkansen kom je een aardig eind met uitgebreid testen, maar bij een systeem met een betrouwbaarheid van 99,999% is dat per definitie onmogelijk. Dat zou veel te veel testtijd vergen. De crux ligt dan in het voorkomen van het injecteren van problemen in het systeem door het toepassen van zeer strikte ontwikkelprocessen. Verificatie en formele validatie van het ontwerp spelen daarin onder meer een belangrijke rol. Zo is het ontwerp van het BOS systeem gespecificeerd in ongeveer 2.000 regels Promela en 20.000 regels Z, zeer compacte valideerbare formele talen. Deze specificatie werd uiteindelijk handmatig, aan de hand van strikte vertaalregels, omgezet in ongeveer 600.000 regels C++ code, met vrijwel 100% testcoverage van alle code. Zo konden we uiteindelijk de zo belangrijke basis-kennis van beschrijfsystemen en verzamelingenleer toepassen in het behalen van de betrouwbaarheid van de software voor de stormvloedkering.

Tegenwoordig ben ik nog steeds met software architectuur en ontwikkelprocessen bezig, maar nu op een totaal ander vlak. Bij TomTom maken dagelijkse miljoenen mensen gebruik van onze navigatiesystemen, navigatie gerelateerde diensten en location based services. En die diensten moeten niet alleen zeer betrouwbaar zijn, voortdurende innovatie is ook ongelooflijk belangrijk. Als een van de grootste en meest vernieuwende navigatiebedrijven ter wereld, moeten we niet alleen inspelen op wensen uit de markt, we moeten op de markt vooruitlopen. Ook hier wordt weer teruggegrepen op die abstracte basiskennis, in dit geval om methoden te ontwikkelen om aan de hand van de enorme hoeveelheid informatie, zoals anonieme GPS posities, die we ontvangen van miljoenen gebruikers, nieuwe nuttige en aantrekkelijke diensten te ontwikkelen. Een mooi voorbeeld daarvan is HD Traffic: de beste verkeersdienst ter wereld. Onze systemen genereren op basis van bewegingen van TomToms en miljoenen mobiele telefoons real-time verkeersinformatie. Een ander voorbeeld is IQ Routes: een zeer complex systeem dat de triljarden snelheidsmetingen, die we in de loop der tijden verzameld hebben, gebruikt om voor iedere steegje, straat of weg ter wereld te bepalen hoe snel daar op ieder tijdstip van elke weekdag gereden wordt, of zal worden; statistisch bepaald en door het enorme aantal datapunten zeer betrouwbaar. Met die informatie kunnen we zelfs daar waar geen real-time verkeersinformatie voorhanden is, toch zeer betrouwbaar de beste routes bepalen en het verkeersbeeld voorspellen, niet alleen nu, maar ook over een paar uur of een aantal dagen.

Het kunnen beschikken over dergelijke informatie en het bij TomTom mogen bouwen aan systemen die daar gebruik van maken, heeft voor mij weer een hele nieuwe wereld van mogelijkheden geopend. Al met al is mijn keuze voor Informatica een mooie geweest. Muziek blijft mijn dierbare hobby. 

Achtergrond: GPS posities (1 pixel = 1 datapunt), TomTom

Reizen tijdens je studie

Stage bij de Schotten

Marjon Ruijter

Vorig jaar heb ik van eind juli tot begin november stage gelopen in Glasgow, Schotland. Dat was een hele leuk tijd en het was een bijzondere ervaring om zomaar een aantal maanden ergens anders te wonen en te leven. Ik zal jullie vertellen over mijn belevenissen en hoop jullie daarmee aan te moedigen om ook een deel van je studietijd in het buitenland door te brengen. En dat hoeft niet per se Glasgow te zijn hoor, er zijn genoeg stagemogelijkheden over de hele wereld!

Dat ik stage wilde lopen in het buitenland stond voor mij al langere tijd vast. Ruim een half jaar voordat het zover was ben ik begonnen met me te oriënteren. Stagecoördinator Jan de Vries wist precies te vertellen wat de mogelijkheden waren en wie waar contacten heeft. Hij vertelde me dat Hans van der Weide contact heeft met Tim Bedford van de University of Strathclyde in Glasgow. Dat sprak me aan en een stageplek was via Hans met één mailtje geregeld.

Begin augustus zou mijn stage beginnen. Daarvoor ben ik met mijn vriend twee weken op vakantie geweest in Schotland om het land beter te leren kennen. Nadat we waren aangekomen in Glasgow kon ik wat spullen dropen bij m'n stageafdeling. Zo had ik alvast kennis gemaakt met de universiteit en stad waar ik 3,5 maanden zou blijven.

Glasgow is de grootste stad van Schotland. Het is een oude industrie- en handelsstad, maar de laatste jaren is er veel gerenoveerd en is het meer toeristisch geworden. Heel veel musea en kerken in Glasgow zijn gratis te bezichtigen en verder kun je heel vaak studentenkorting krijgen. Een greep van wat er aan toeristische dingen te zien is: de Barony Hall, het oudste huis van Glasgow Provand's Lordship, St Mungo's Museum of Religious Life&Art, een horrorfilm-achtige Necropolis (begraafplaats), de Glasgow Cathedral, het veel te grote Kelvingrove Art Gallery & Museum en de stiekem veel mooiere gebouwen van de University of Glasgow (de andere grote universiteit van Glasgow).

Maar als je naar Schotland gaat, kom je niet echt voor de steden, maar meer voor het landschap van de Highlands. We zijn met de trein naar het kleine stadje Oban gegaan, aan de westkust van Schotland. Het openbaar vervoer is buiten de steden een stuk beperkter dan in Nederland, maar je kunt een heel eind komen. Vlakbij Oban zijn verschillende eilanden, waarvan wij op Kerrera

en Mull zijn geweest, waar we hebben gewandeld en gefietst. In het pittoreske vissersstadje Tobermory staat een van de vele whiskydistilleerderijen van Schotland, waar we een korte rondleiding en kleine proeverij (om 12 uur 's middags) hebben gehad.

Gedurende de zomer worden er, verspreid over Schotland, vele Highland games georganiseerd. Dit is een echte aanrader om heen te gaan. Wij zijn naar de Highland Games in het dorpje Taynuilt geweest. Hier waren onder andere Highland dansen, hamerwerpen, steenstoten en boomstam gooien te zien. Alle mannen natuurlijk in geruite kilt en met op de achtergrond de hele dag doedelzakmuziek!

Na twee weken zijn we terug gegaan naar Glasgow. Deze grote stad vormt een groot contrast met het dunbevolkte noordelijke Schotland dat we hebben gezien.

Boomstammen gooien



En toen begon mijn stage aan de University of Strathclyde, aan het Management Science department. Ik heb hier gewerkt aan twee verschillende opdrachten. De eerste maand was ik bezig met een opdracht over de onderhoudskosten van vliegtuigen en helikopters, dit was vooral statistiek. Ik heb regressie-modellen voor de onderhoudskosten onderzocht en gekeken wat de invloed is van variabelen zoals gewicht, complexiteit en bouwjaar. De tweede opdracht ging over stoombuizen die gebruikt worden bij de stroomopwekking van een energiebedrijf. Ze willen beter inzicht krijgen in hoe vaak en wanneer bepaalde buizen kapot gaan en hier betere voorspellingen over doen, reliability and maintenance dus. Ik heb een soort vooronderzoek gedaan en heb informatie verzameld over modellen en processen om de toestand van de buizen te beschrijven. Er kwam hier wel minder wiskunde bij kijken dan bij m'n eerste opdracht. Het was erg toegepast, ik moest rekening houden met wat het bedrijf zou willen en het wiskundig niet te moeilijk voor ze maken. Wel wat anders dus dan wat ik meestal bij de studievakken te zien krijgen en daarom ook wel interessant.

De kamer op mijn afdeling deelde ik met twee promovendi. De promovendi en afstudeerders daar doen regelmatig dingen samen. Zo gaan ze bijna iedere week een keer naar de bioscoop en we zijn een aantal keer uit eten geweest of naar een pub. Het was een echte "vrouwenafdeling", wel iets anders dan ik in Delft gewend ben, maar erg leuk. Zo gingen we met de meiden tijdens de National Chocolat Week lekker chocoladetaart eten en een romantische komedie kijken. Meestal lunchten we ook samen. Het was erg leuk om te zien hoe verschillend iedereen luncht. Een Hollandse lunch met brood is wel erg simpel eigenlijk en het kruidige warme eten van de Thaise Noh rook veel lekkerder.

Voor internationale studenten werden er regelmatig activiteiten in het weekend georganiseerd. Daarnaast wordt er iedere woensdagavond een pub uitgekozen om elkaar te ontmoeten. De eerste weken dat ik er was, was het collegejaar nog niet begonnen en waren we met ongeveer 30 studenten. Maar toen het collegejaar begon waren er maar liefst 500 studenten. Dat maakte het erg makkelijk om veel mensen te leren kennen.

Eén van de trips die ik met de internationale studenten heb gemaakt was naar de opening van het Edinburgh Fringe Festival. Edinburgh is dus hoofdstad van Schotland en er zijn gedurende de zomer veel grote festivals, de Fringe is de grootste daarvan. Tijdens dit bijna vier weken durende festival zijn overal op straat artiesten te vinden, goochelaars, dansers, acrobaten, karikaturisten, muzikanten, ... Dat was heel speciaal en de stad was daardoor ook erg gezellig.

De week daarna was er in Glasgow een groot festival: de hele week was er het International Piping Festival, inclusief de World Pipe Band Championships 2009, oftewel een heel groot Doedelzak-festival! Nu denk je misschien "International? Ik dacht dat de doedelzak alleen in Schotland voorkomt." Maar dat blijkt niet zo te zijn, daar kwam ik ook achter toen ik naar het Nationale Doedelzak Museum ging. In de stad, met name op het plein George Square, waren de hele week optredens te zien en ook op de campus van de universiteit werd overal muziek gemaakt!

Accommodatie in Glasgow moest ik zelf regelen, maar dat is gemakkelijk gegaan. De eerste vier weken kon ik terecht op de studentencampus. Dat was ongeveer 5 minuten lopen naar m'n stageplaats, maar het waren wel ongeveer 100 traptreden op en af. De universiteit ligt op een heuvel en daar moest ik overheen. Op een aantal steile heuvels na is de stad verder redelijk vlak. Na die vier weken kon ik voor de rest van mijn verblijf terecht bij een andere studentenaccommodatie. Deze lag niet in het centrum van de stad, maar er reden veel bussen naar toe. En met mijn mountainbike, die ik op een beetje rare bazaar had gekocht voor 20 pond, was het ook goed te doen. Eerst woonden we er met z'n drieën, later met 6 internationale studenten. Naast mij twee studenten uit Litouwen, twee Duitse wiskundestudenten en een meisje uit Slowakije. Vanuit



Huisgenoten

mijn kamer kon ik het rugbystadion van de Glasgow Warriors zien. Met twee rugbyfanaten van m'n afdeling ben ik naar een wedstrijd geweest. Als er een wedstrijd was kon ik de supporters altijd goed horen, het was leuk om het ook een keer gezien te hebben.

In Delft ben ik lid bij een hardloovereniging en ik had snel uitgezocht dat er ook een leuke studenten-hardloovereniging in Glasgow is. In de stad kwamen we altijd veel stoplichten tegen, maar eenmaal bij de rivier of het Glasgow Green Parc aangekomen viel het verkeer mee. Iedere zaterdag werd er in een ander park een hardloewedstrijd georganiseerd. Daar heb ik een paar keer aan gedaan. De eerste keer werd ik als Nederlander gewaarschuwd dat het parcours wel erg heuvelachtig zou zijn, het is daar nou eenmaal niet zo vlak als bij ons.

Glasgow ligt op ongeveer een uur vliegen van Nederland. Zo konden mijn ouders, Jeroen en drie vriendinnen uit Delft makkelijk een weekend langs komen. Zij hadden lekker stroopwafels en drop meegenomen, die ik ook had uitgedeeld op mijn afdeling. De stroopwafels vielen in de smaak, maar drop vonden enkelen vies en medicijnachtig smaken. Met Lieke, Annegreet en Dorien hebben we een bustour naar de Highlands gemaakt. Via Loch Lomond en Fort William reden we naar het beruchte Loch Ness. Tijdens een boottocht hebben we Nessie helaas niet gezien, wel apart om te zien hoe dat monster het toerisme daar zo heeft doen opbloeien.

De Schotten hebben een mooi accent, wat het wel eens lastig maakte om te communiceren. Vooral buschauffeurs en mijn hardlooptrainster daar kon ik moeilijk verstaan. Op de universiteit en met studenten gaf dit gelukkig geen problemen.

Het laatste wat ik kwijt wil over Schotland is dat de mensen er bijzonder vriendelijk en behulpzaam zijn. Als je met een kaart op straat staat om de weg te vinden, word je spontaan geholpen. En bij een Ceidlidh-avond (Schotse volksdans) wordt je gewoon de dansvloer op meegenomen en worden de pasjes geleerd. Mijn tijd in Glasgow is snel voorbijgegaan en ik zal er nog vaak aan terug denken. 

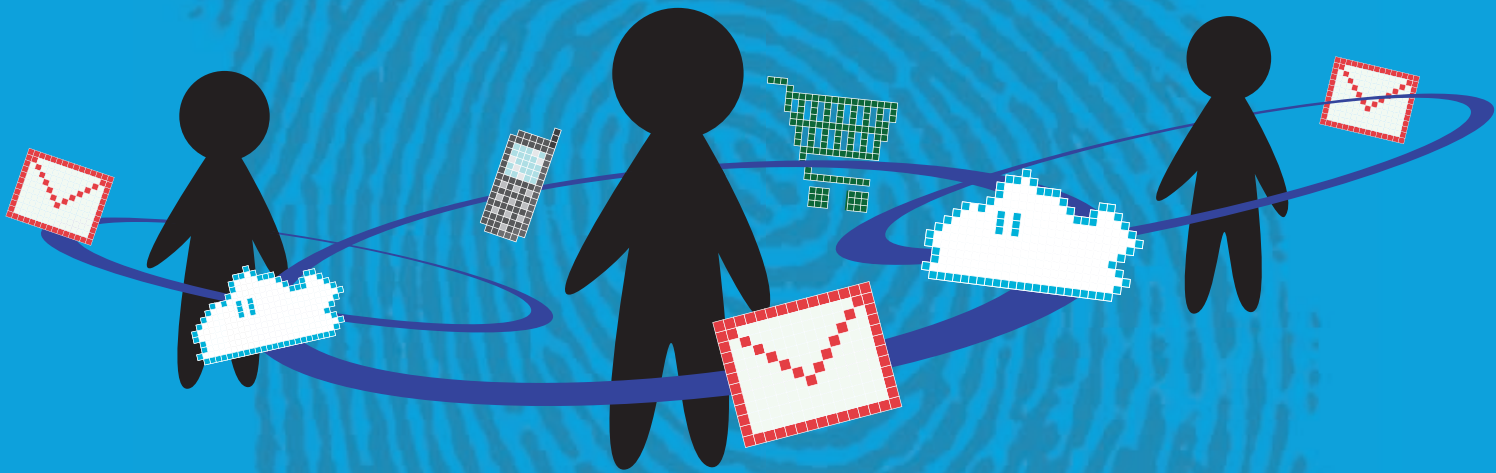
Symposium

Tuesday 27 April 2010
MustSee - Delft
10:00 - 18:00

Lunch & Drinks included
More information & Registration at
symposium2010.nl

Distributed systems
P2P
Cryptology
Law and Privacy

Cloud your Identity, Distribute your Life



**Quantum Cryptography &
Identity Access Management**
Siemens

OV-chipkaart Beveiliging & Privacy
Brenno de Winter

Secure Multi-party Computations
Prof. dr. R. Cramer

Angle: 4th generation P2P
Dr. Ir. J.A. Pouwelse

Pseudonimisering van Persoonsgegevens
Prof. dr. E.R. Verheul

RFID Guardian
Dr. M. Rieback

**Computations on Graphics, Game &
Intel multi-core processors**
Prof. dr. W.P. Petersen

Policy Laundering
Jérémie Zimmermann

And more...

THALES NEDERLAND

Toonaangevend in de sectoren Defence en Security. Met circa 2.000 gedreven medewerkers de toonaanbieder van hightechbanen. Onze primaire focus is het innoveren van onze producten en het ontwikkelen van de nieuwste technologieën. Spraakmakende voorbeelden van onze cutting edge technologieën zijn radar-, communicatie- en command & controlsystemen voor marineschepen en communicatie-, beveiligings- en betaalsystemen voor het bedrijfsleven. Thales Nederland (hoofdkantoor in Hengelo) is onderdeel van de internationale Thales Group.

OUR CAREER FEATURES

HIGHTECH

Je werkt aan unieke en zeer complexe producten.

MULTIDISCIPLINAIR

Je werkt als specialist in gevarieerde, multidisciplinaire teams.

INTERNATIONAAL

Je werkt voor opdrachtgevers over de hele wereld door systemen te ontwikkelen, te verkopen, te installeren, te testen en te onderhouden.

DYNAMISCH

Je werkt in de dynamiek van de multinational Thales. Onze vestigingen vind je overal ter wereld. Je internationale doorgroei kansen dus ook.

UITDAGEND

Je werkt aan je persoonlijke ontwikkeling in een omgeving die je constant uitdaagt.

EMBARK ON A THALES ADVENTURE IN THE NETHERLANDS

Heb je een afgeronde opleiding in Informatica, Elektrotechniek, Technische Natuurkunde, of vergelijkbaar dan kan jouw loopbaan zich in verschillende richtingen ontwikkelen. Bijvoorbeeld in een specialistische functie als Software Engineer, Radar Engineer, System Engineer of Software Architect. Ambieer je een Managementfunctie dan is je volgende carrièrestap Project Manager, Commercial Manager of Program Manager met mogelijk een doorgroei naar Executive Management.



By courtesy of Royal Schelde Group



Worldwide: with 68.000 employees, a presence in 50 countries, Thales is a global leader in aerospace, defence and security.



Apply straight away

Thales komt graag in contact met jou om samen jouw mogelijkheden te bekijken en je carrièrepad uit te stippelen. Ook bieden wij studenten met een technische achtergrond meer dan 100 uitdagende stage- en afstudeeropdrachten. Vraag onze gids aan of bezoek onze website. Je kunt daar ook solliciteren.

www.thalesgroup.com/netherlands

START YOUR THALES
ADVENTURE

