I/O Vivat

Jaargang 18, nummer 2,
november 2002
ISSN: 1389-0468
Oplage: 1300

I/O Vivat is het drie maandelijkse orgaan van Inter-*Actief*, de studievereniging voor Informatica, Bedrijfsinformatietechnologie en Telematica.

Hoofdredacteur
Marc Maurer

Opmaak
Ilse Fokker, Marc Maurer, Ruben Smelik, Maks Verver

Redactie
Ilse Fokker, Marc Maurer, Ruben Smelik, Maks Verver, Matthijs Punter

Columnisten
Gerrit van der Hoeven, Maarten Donders, Ruben Smelik, Marc Maurer

Gastschrijvers
Mark Weiser, Paul Havinga, Hans Scholtem, Andy Hopper, Ilse Fokker, Erik van der Sluis, Oebele van Veen, Berteun Damman, Tjarda Koster

Drukker

PrintPartners Ipskamp

Adressen

E-mail:	vivat@inter-actief.utwente.nl
Post adres:	Inter-*Actief*
Telefoon:	Postbus 217 INF L216
	7500 AE Enschede
Internet:	www.inter-actief.net

Dank aan alle inzenders van kopij

De studievereniging wil de adverterende bedrijven bedanken voor de goede samenwerking.

Beste I/O Vivat lezer,

Onlangs heeft onze hoofdredacteur Ruben Smelik de last van een bestuursfunctie van Inter-*Actief* op zich genomen. Hierdoor was de functie van hoofdredacteur vrijgekomen binnen de I/O Vivat redactie. Aangezien een dergelijk functie CV technisch erg interessant is, heb ik deze taak geheel belangeloos op mij genomen.

Met een beetje fantasie komen we zo op het punt van de ledenwerving. Om onze Vivat op een hoog niveau te kunnen laten blijven concurreren met welbekende bladen als Nature en het Science Magazine, is het van vitaal belang dat onze redactie op volle oorlogssterkte kan opereren. En dat is nu niet (meer) het geval; een ernstige zaak. Wij doen daarom een ernstig beroep op onze leden om zich te melden en te zitting te nemen in onze redactie.

Tenslotte wil ik iedereen wijzen op de ongekende mogelijkheden die het I/O Vivat te bieden heeft. Zie het blad als de Russische (groei)markt voor het kapitalistische Westen. Zo is bijvoorbeeld het geplaatst krijgen van een artikel in ons aller blad aan te merken als een officiële publicatie, aangezien wij een heus ISSN hebben (dat nummertje in de colofon links). Profiteer hiervan en laat deze mogelijkheden niet onbenut!

Deze uitgave staat geheel in het teken van het Smart Surroundings symposium, wat 11 december gehouden zal worden op de Universiteit Twente. Smart Surroundings is een breed en veel omvattend onderwerp waar je zeer zeker, in meer of mindere mate, in de rest van je leven mee te maken zult krijgen. Zorg dat je voorbereid bent op de toekomst en schrijf je in op http://www.smart-surroundings.nl.

Marc Maurer
Hoofdredacteur I/O Vivat

Inhoudsopgave

Smart Surroundings

Inter-Actief

Algemeen

Columns

Wat zoekt Gerrit...

Het digitale geluk?

Pratend over de uitdagingen aan de ICT is het toevoegen van 'intelligentie' of zelfs 'emotie' aan de leefomgeving van de mens, zijn auto, zijn koelkast, zijn fornuis, zijn tv, zijn verwarming, zijn telefoon, zijn computer een terugkerend thema. Sterker, sommigen noemen de hoeveelheid chips die iemand bezit met de bedoeling zijn leefomgeving 'intelligent' te maken de nieuwe maat voor ware rijkdom.

De vragen die zich voordoen bij het creëren van een werkelijk intelligente omgeving zijn zeer intrigerend en uitdagend. Generaties informatici kunnen en zullen zich nog door die vragen laten inspireren. Laat dat ook vooral gebeuren.

Toch heb ik soms ook een onbehaaglijk gevoel bij al die intelligentie en 'smartness' die de informatica ons wil brengen. Dat ligt vast aan mijn vooroordelen. Ik heb altijd gedacht: intelligentie dient het debat, het debat zaait de onrust, de onrust drijft de vooruitgang. Dat is mooi. Maar de 'intelligentie' die de nieuwe rijken in hun chips met zich dragen dient niet het debat maar het gemak, het gemak zorgt voor rust, en rust is stilstand. Dat is helemaal niet mooi. Wat willen we nu eigenlijk?

Ik wil geen pleidooi houden voor het ontwikkelen van een koelkast die het debat uitlokt door te stoppen met koelen als het gedrag van zijn eigenaar hem niet aanstaat. Ik wil wel waarschuwen dat 'smart surroundings' in een ongunstige interpretatie betekent: het afschermen van het individu van ongewenste prikkels, daarmee het bevorderen van de gemakzucht en de snellere frustratie bij tegenslag, en het afslijten van het vermogen om om te gaan met fouten, uitdagingen en teleurstellingen.

In 1992 schreef ik voor I/O Vivat een verhaal over het onderzoek in taaltechnologie, waarin de nodige 'smart surroundings' de revue passeerden. De vragen op dit gebied dagen uit tot onderzoek, dat vond ik 1992, dat vind ik nu. Maar 'smart surroundings' zijn niet de sleutel tot het menselijk geluk. Integendeel bijna.

Goed, u weet wie het zegt. Als mensen enthousiast over 'global village' praten, hoor ík vooral 'village' en dan denk ik ook: geen debat, geen onrust, geen vooruitgang, wat willen we nu eigenlijk? Tijd voor nieuw optimisme, geloof ik.

...van der Hoeven?

What is the metaphor for the computer of the future? The intelligent agent? The television (multimedia)? The 3-D graphics world (virtual reality)? The Star-Trek ubiquitous voice computer? The GUI desktop, honed and refined? The machine that magically grants our wishes? I think the right answer is "none of the above", because I think all of these concepts share a basic flaw: they make the computer visible.

Mark D. Weiser

The World
is not a desktop

A good tool is an invisible tool.

By invisible, I mean that the tool does not intrude on your consciousness; you focus on the task, not the tool. Eyeglasses are a good tool -- you look at the world, not the eyeglasses. The blind man tapping the cane feels the street, not the cane. Of course, tools are not invisible in themselves, but as part of a context of use. With enough practice we can make many apparently difficult things disappear: my fingers know vi editing commands that my conscious mind has long forgotten. But good tools enhance invisibility.

I think the value of invisibility is generally understood. Unfortunately, our common metaphors for computer interaction lead us away from the invisible tool, and towards making the tool the center of attention.

Take multimedia.

The idea, as near as I can tell, is that people already spend hours a week at home watching television, so clearly television is attractive, and we want our computer interfaces to be attractive, so let's put TV into them. To mention a few things that may be wrong with this chain of reasoning: is everything we spend a lot of time doing attractive (sleeping? breathing? worrying?); will the attactiveness of multimillion dollar production TV translate to casual computer TV? And most importantly for this essay, should computer interfaces be attractive at all? Attractiveness is the opposite of invisible.

Take intelligent agents.

The idea, as near as I can tell, is that the ideal computer should be like a human being, only more obedient. Anything so insidiously appealing should immediately give pause. Why should a computer be anything like a human being? Are airplanes like birds, typewriters like pens, alphabets like mouths, cars like horses? Are human interactions so free of trouble, misunderstanding, and ambiguity that they represent a desirable computer interface goal? Further, it takes a lot of time and attention to build and maintain a smoothly running team of people, even a pair of people. A computer I need to talk to, give commands to, or have a relationship with (much less be intimate with), is a computer that is too much the center of attention.

Take magic.

The idea, as near as I can tell, is to grant wishes: I wish I was the person I am now, but richer; I wish my boyfriend were smarter and more attractive; I wish my computer would only show me what I am interested in. But magic is about psychology and salesmanship, and I believe a dangerous model for good design and productive technology. The proof is in the details; magic ignores them. Furthermore, magic continues to glorify itself, as Robin Williams' attention-grabbing genie in Aladdin amply illustrates.

Take virtual reality.

The idea, as near as I can tell, is that by moving to full-body-sensing and interaction we'll solve the user interface problem by maximally utilizing all of our body's input and output channels. Setting aside for a later time the appropriateness of the "input" metaphor to humans being-in-the-world, VR seems to have the goal of the invisible computer behind the scenes. But is it really true that the problem with our current user interfaces is that we don't have enough of them? Is it a quantity problem -- a little user interface is good, more is better? VR, by taking the gluttonous approach to user interface design, continues to put the interface at the center of attention, leaving the real world behind.

Take voice input.

The idea, as near as I can tell, is that if I could just talk to my computer it would finally understand me. The problem is, if I could talk to my computer today, I'd have to talk in C or Fortran or CSH, because that is what they understand. When I can send email to my computer and have it DWIM the answer, then I'll start to believe in voice computers for limited applications. Limited, because most of my life I am with other people, and I want to talk (or listen) to them, not to my computer. If I want to take notes, or glance at information, I want to do so unobtrusively. Voice command is so well-known in science fiction exactly because it is prominent and attention grabbing -- fiction is supposed to hold our attention. A good tool is not.

I do think that research on agents, speech recognition, and so on is important; the problem is that they are all in the domain of the conscious interaction. The result is that the research dialogue is restricted to a narrower-than-necessary set of problems, rather than the broader problem of good, invisible, tools. I believe we could use a lot more attention on techniques of invisibility, including abandoning computers as we know them.

It was the desire to build technology truer to the possibility of invisibility that caused me to initiate the ubiquitous computing work at PARC five years ago. The first phase of that effort incorporated existing projects, such as the wall-sized pen computer called LiveBoard, and added others, the inch-sized tab and the foot-sized pad, to create a panoply of devices that could be ubiquitous in the home or office -- hundreds per person, integrated with the everyday setting. Enabling the mundane computer put us on the way to the invisible computer (see the July issue of CACM for more information). But more work is needed.

To understand invisibility the humanities and social sciences are especially valuable, because they specialize in exposing the otherwise invisible. For instance, ethnography can teach us something of the importance of the details of context and setting and cultural background; feminist deconstructionism can teach us a little of the necessity of different, deeply lived, points of view to real understanding.

The clock, and the clockwork machine, are the metaphors of the past several hundred years of technology. Invisible technology needs a metaphor that reminds us of the value of invisibility, but does not make it visible. I propose childhood: playful, a building of foundations, constant learning, a bit mysterious and quickly forgotten by adults. Our computers should be like our childhood: an invisible foundation that is quickly forgotten but always with us, and effortlessly used throughout our lives.

"Why should a computer be anything like a human being?"

Dr. Mark Weiser, Chief Technology Officer at Xerox Palo Alto Research Center (PARC), was best known for his contributions to the field of mobile computing. He was often referred to as the father of "ubiquitous computing". He coined that term in 1988 to describe a future in which PCs will be replaced with invisible computers embedded in everyday objects. He believed that this will lead to an era of "calm technology," in which technology, rather than panicking us, will help us focus on what is really important to us.

Other research interests included garbage collection, operating systems and user interface design. Dr. Weiser, who held several U.S. and foreign patents, wrote or co-wrote more than 75 technical publications on such subjects as the psychology of programming, program slicing, operating systems, programming environments, garbage collection and technological ethics. He taught graduate and undergraduate courses on human factors, systems, and programming. He was a popular speaker at scientific symposia and conferences, and a frequent subject of media interviews.

Dr. Weiser, who founded three companies, was the drummer with rock band Severe Tire Damage, the first band to perform live on the Internet. He was born on July 23rd, 1952 in Chicago, Illinois. He was married with two children.

For more details, please see http://www.ubiq.com/weiser.

"Ambient intelligence" will become a network of hidden intelligent interfaces that recognise our presence and mould our environment to our immediate needs

John Horvath

Making friends

with big brother

Much has been made of the intrusion of computer technology in our lives. This should come as no surprise for, after all, one of the most potent symbols of George Orwell's police state in his novel "1984" was that of the telescreen. Although it was identified with the passive nature of television, the fact that Internet communications is interactive has naturally led many to identify telescreens with that of the computer monitor.

There is no doubt recent advances in information and communication technologies have had a major impact on the way we live, work and interact with each other. Yet if computer technology of the present has raised some concern over privacy, that of the future should lead us to near panic. Research into "ambient intelligence", a network of hidden intelligent interfaces that recognise our presence and mould our environment to our immediate needs, could bring about an even more radical change.

Ambient intelligence refers to an electronic environment that is sensitive, adaptive, and responsive to the presence of people. In practical terms this means we will be surrounded by intelligent interfaces embedded in everyday objects such as furniture, clothes, vehicles and roads. As we move through our environment, these interfaces register our presence, automatically carry out certain tasks based on given criteria, and learn from our behaviour in order to anticipate our needs.

While this may sound like the stuff of science fiction, some serious effort has been put into making it a reality. For the past three years, a project on collaborative sensing has been undertaken by scientists and researchers at the Xerox Palo Alto Research Center (or PARC) in Silicon Valley. According to Feng Zhao, a senior researcher of the project at PARC, ambient intelligence technology is a key element in the post-internet revolution.

"The key challenge here," says Zhao, "is to instrument our physical world with all these tiny, dirt-cheap sensors wirelessly connected."

We are already surrounded by sensors -- devices ranging from the most basic thermostats, to sensors in cars which monitor everything from fuel consumption to vibrations. All of these capture physical information about their immediate space. But they can only do so much. Hence, they are often referred to as "dumb" sensors, in that they don't have onboard processing and don't network with each other.

The sensors of the future, however, known as "smart" sensors, are expected to wirelessly connect with similar sensors in their environment in a so-called "sensor network", or sensornet. Unlike their "dumb" counterparts, they would also contain a computer chip in order to process incoming information.

Supporters of the technology see the potential applications for smart sensors as wide-ranging. For example, smart sensors on roads could make travel safer and highways less congested by noting accidents, potholes and alternate routes. It could then relay the information to a car's global positioning system. What's more, the sheer volume of information able to be collected means a more powerful ability to make predictions, even on a seemingly mundane level.

Some of the settings to incorporate this new technology have already taken shape in so-called "smart houses". Philips Research, the R&D armof one of the world's largest electronic companies, has developed the first prototypes of an ambient intelligent home system.

Although much progress has already been made, it will be some time before the networked vision of collaborating sensors actually take-

sshape. Feng Zhao predicts that in 5-10 years from now sensors will be ubiquitous, and will be everywhere including refrigerators, microwaves, garden fences, and car tires. This is how a European Commission report imagines the use of ambient intelligence:

"The year is 2010. You've just got off a long-haul flight in a foreign city and you're looking forward to a few hours' sleep before your important presentation. First though, you have to pick up your hire car and find your way across town to your hotel. Your only stops between your plane and the airport exit are baggage claim and customs, because the communications device on your wrist dealt with passport and visa control the moment you stepped into the arrivals hall. Not only is your hire car waiting in its designated bay, but the doors unlock as you approach and it starts at the touch of a button - no queuing for keys. The built-in navigation system is already showing you the best route to your hotel and you can enjoy the drive knowing there is a parking space reserved for you at the other end. Forty minutes later, as you step into your hotel room and the lights and temperature automatically adjust to your personal preference, you silently thank the Ambient Intelligence (Aml) equipment that has monitored your progress ever since you stepped off the plane." (Euroabstracts, Volume 39, Number 4, August 2000, p.15)

While projections of a brave new world based on the use of smart sensors and ambient intelligence abound, the optimism about the technology and its applications are divorced from social reality. The premise is that the technology will help us to live a better life because of their ability to detect a lot of interesting things we can't -- or don't have the capacity -- to focus on. Yet they are also intrusive. Like all technology, it can have beneficial uses; however, more often than not, they are mainly developed and employed for reasons not so benign.

Politically, full acceptance of the technology is primarily focused on the economic benefits it supposedly would bring, with only scant consideration given to possible abuses. In Europe, members of the Information Society Technologies Advisory Group (ISTAG) have been making consistent efforts to not only demonstrate how such technology can impact our lives, but to promote key developments in the field so that it will lead to "positive" scenarios, such as the one described earlier.

"We are surrounded by sensors"

To this extent, the ISTAG has identified three major factors that will determine the successful implementation of ambient intelligence. The first is that it must facilitate human contact rather than replace it. The second is its impact on business practices, in that companies will need to create complex partnerships and adopt radical new business plans. The third is the need for significant, long-term focused research.

Along these lines, security and trust technologies -- privacy, safety and dependability -- are seen as "naturally important" aspects to the future of technological development. Yet for those concerned with the preservation of democracy, decency, and human rights, the fact that such concerns are "naturally important" is not enough; security and trust technologies must be seen as a crucial factor in the future development -- and deployment -- of smart sensors and ambient intelligence technologies.

If the pundits are correct as to the future course of research and technological development, we may be in store for a nightmare. Unfortunately, in much the same way most of us have voluntarily accepted the intrusion of big brother into our lives, either through fear or ignorance, our total loss of privacy will most likely come about through peaceful capitulation rather than brute force.

Recently, we have seen major progress in developing the new off-the-desktop computing paradigm that moves towards the notion of a pervasive, wearable, unobtrusive, disappearing, or invisible computer. "Ubiquitous Computing", a phrase which the late Mark Weiser described as "the calm technology, that recedes into the background of our lives", matures from the vision of the Nineties to reality of the young millennium, enabling increasing mobility and interaction of services and applications in a large variety of areas in daily life.

The Embedded Systems research group is pursuing research in this field on various aspects. The areas of activity include: wireless communication, sensor networking, home networking, real-time operating systems, and security. In this short overview two projects will be highlighted that try to solve problems around this theme: the EYES project (http://eyes.eu.org), and the At Home Anywhere project (http://wwwes.cs.utwente.nl/aha/).

Paul Havinga, Hans Scholten

Towards Ubiquitous Computing

Embedded Systems research in the Computer Science department of the University of Twente

Energy-efficient sensor networks: the EYES project

Imagine a world with smart machines that can self-diagnose and repair, predict aging components and proactively alert factories for replacement parts before the machine breaks down. Smart roads will make travel safer and highways less congested by noting accidents, potholes, alternate routes and reporting the information to a car's navigation system. Smart appliances, such as refrigerators, will understand families' dietary requirements or doctor's orders and take inventory of refrigerators to relay information to a shopping list on a personal digital assistant (PDA). Collaborative sensor networks will help realize this vision.

Sensors are tiny devices capable of capturing physical information, such as heat, light or motion, about an environment. Rapid advances in micro-electromechanical systems (MEMS), digital circuitry, and wireless communication have enabled a new generation of tiny, inexpensive, networked sensors. Embedding millions of sensors into an environment creates a digital skin or wireless network of sensors. These massively distributed sensor networks, communicate with one another and summarize the immense amounts of low-level information to produce data representative of the overall environment. From collaboration between (large) groups of sensor nodes, intelligent behavior can emerge that surpasses the limited capabilities of individual sensor nodes. Collaborative, smart sensor networks present information in a qualitative, human-interpretable form, which allows people (or computers) to respond intelligently. Sensor networks will change the way we work and live.

The European project EYES (IST-2001-34734) develops the architecture and the technology needed for building such self-organizing and collaborative energy-efficient sensor networks using smart sensor nodes, which are self-aware, self-reconfigurable and autonomous. The project runs from March 2002 till March 2005. The consortium consists of 6 partners from Italy, Germany, France, and the Netherlands. The CTIT is the coordinator of the project, and involves people from the Mathematical Sciences department, and the Embedded Systems group of the Computer Science department.

In our vision, sensor nodes collaborate to be able to cope with the environment: sensor nodes operate completely wireless, and are able to spontaneously create an impromptu network, assemble the network themselves, dynamically adapt to device failure and degradation, manage movement of sensor nodes, and react to changes in task and network requirements. Despite these dynamic changes in configuration of the sensor network, critical real-time information must still be disseminated dynamically from mobile sensor data sources through the self-organising network infrastructure to the applications (services). Furthermore, next generation sensor networks provide processing power to perform intelligent tasking. For example, power consumption may be minimised by intelligently trading high-consumption transmission for low-consumption processing.

We have two distinct key system layers of abstraction: the sensor and networking layer, and the distributed services layer. Each layer provides services that may be spontaneously specified and reconfigured.

- The sensor and networking layer contains the sensor nodes and the network protocols. Ad-hoc routing protocols allow messages to be forwarded through multiple sensor nodes taking into account the mobility of nodes, and the dynamic change of topology. Communication protocols must be energy-efficient since

sensor nodes have very limited energy supply. To provide more efficient dissemination of data, some sensors may process data streams, and provide replication and caching.

• The distributed services layer contains distributed services for supporting mobile sensor applications. Distributed services co-ordinate with each other to perform decentralised services. These distributed servers may be replicated for higher availability, efficiency and robustness. We have identified two major services.

"Most efforts 'till now concentrate on entertainment"

The lookup service supports mobility, instantiation, and reconfiguration. The information service deals with aspects of collecting data. This service allows vast quantities of data to be easily and reliably accessed, manipulated, disseminated, and used in a customized fashion by applications.

On top of this architecture applications can be built using the sensor network and distributed services. Communication in sensor networks is data-centric since the identity of the numerous sensors is not as important as the data they contain. We seek solutions to so called semantic addressing, for mapping terms like "all sensors in the living room" to network identifiers, possibly using multicast techniques.

Currently, we are working within the University of Twente with several staff members, five PhD students, and four master's students working on the project. We have a small prototyping network running already. At the end of the project we hope to demonstrate some sample applications on a large-scale system of more than 100 sensor nodes.

Home telematics: At Home Anywhere

At Home Anywhere (@HA for short) addresses home automation or 'home telematics'. Presently, a house has several separated distribution and communication infrastructures: telephone, cable TV and radio, satellite TV, PC network, connection between thermostat and central heating system, etc. In most cases these infrastructures are isolated islands that interconnect only in rare occasions. In general, these infrastructures can be divided in three classes:

1. entertainment: audio, video, games, etc. This class requires high bandwidth and real-time responses. Characteristic for this class is isochronous, streaming data.

2. sensors and control: sensors and actuators, e.g. central heating control, fire detection, burglar alarm. S&C uses low bandwidth, but requires a high degree of dependability. Some devices may need real-time services.

3. information: PC applications, World Wide Web browsing, etc. This class uses bandwidth in bursts of data, and only needs best effort responses.

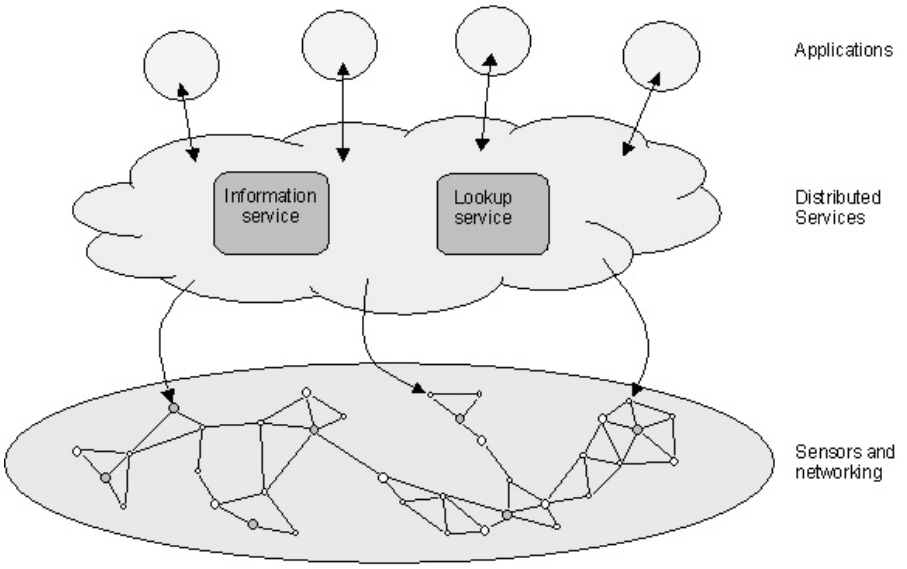


Fig. 1: Sensor network architecture.

In @HA we plan to connect appliances is one common, inexpensive infrastructure that supports entertainment, sensors and information. This infrastructure may incorporate different types of networks, including wireless networks.

Networking and interoperability

Most efforts till now concentrate on only one class of appliances, mostly entertainment. The first objective of the @HA network is to provide a network for entertainment, S&C and information that supports both real-time and non-real-time data, as well as streaming media. This network will be based on a new variety of a rotating token protocol, giving bandwidth to the appliance that has the token. @HA deploys a new type of real-time token protocol that can be used on top of low-cost network hardware, e.g. Ethernet. In contrast to other protocols, the token does not follow a fixed rotation path, visiting every node during each rotation. Instead, the token is scheduled and follows a dynamic route, visiting only those nodes that need attention.

Data storage redundancy and data incompatibility

A key building block for @HA system is a mixed-media storage server that will provide storage for all appliances and devices at home. The storage server must support streaming media, as this is one of the important data types in the system. One of the functions the storage server must offer is simultaneous real-time recording and playback of multiple video and audio streams on hard disk and optical disc

(DVD). Time shifting (the ability to pause live broadcasts) is one of the applications of such a storage server. Information is stored once and can be accessible by many. Efficient storage and retrieval of data is essential, as the amount of data in the file server is huge. E.g. one-hour video in DVD quality measures several gigabytes of data. @HA here builds on previous research done in our group.

Resources

Even small appliances (resource lean), like a temperature- or light sensors, should be connected to the network, but their size and price preclude 'heavy' processors to accomplish that. @HA investigates the concept of delegation, or controlled invocation: small systems use their limited processing power to implement at least a network stack to connect to the network and a small real-time operating system kernel to handle the lightweight protocols. But even such an implementation with a small footprint operating system kernel may be too resource rich. Depending on the type of appliance, a balance must be found between hardware and software. Hardware can range from simple dedicated hardware or micro-controllers to full-fledged processor systems, while software can range from simple runtime systems executing state machines to operating systems executing complex multi-tasking programs.

In @HA we distinguish three classes of appliances: 3C (3+ cent) appliance: simple devices that implement only a network stack to connect to the system; 3D (3+ dollar) appliance: medium complex devices that implement network stack and delegation protocols on a small, smartcard-like processor; 300D (300+ dollar) appliance: powerful devices, controlled by a complex embedded computer.

Delegation is the basis for location transparency: the user interface and underlying application of any appliance are available anywhere in the house. For instance, the settings of the thermostat of the central heating may be inspected and changed on any available display, be it TV set, PDA or PC. The concept of delegation is not new. The X Window System separates application and user interface, so each can execute on a different network node. But this system has a lot of information interchange between client and server and therefore is not suitable. HAVi does something similar. It distinguishes between controlling and controlled device. Applications and user interfaces (called havlets) are written in Java and allow for flexible and powerful extensions and modifications. But some devices we have in mind might be too small, even for Java virtual machines.

Sentient computing is the proposition that applications can be made more responsive and useful by observing and reacting to the physical world. It is particularly attractive in a world of mobile users and computers.

The paper presents a classification and quantification of sensor information together with a description of a method for altering the behaviour of arbitrary terminal devices. It also presents a framework for "programming with space" which can associate space-related events with actions. Consideration is given to the applications made possible by such systems.

Andy Hopper

Sentient Computing

The Royal Society Clifford Paterson Lecture, 1999

Sentient computing

Computers have become integral to our lives, but for many the gap between man and machine is so large as to be effectively unbridgeable. Central to any good working relationship is a degree of mutual understanding between the two parties. The problem with conventional human-computer interaction is that responsibility for understanding, or the lack of it, lies wholly with the user.

The office PC illustrates this. Outwardly it seems different from earlier models of computing, such as time-sharing systems or even the mainframe. But, take away the mouse-driven user interface and the fundamentals remain unchanged. The burden for understanding lies wholly with the user. Not so long ago computers came with a team of dedicated operators. In the PC era everyone is his or her own operator. No wonder so many PCs are only used to a fraction of their potential. It does not have to be this way.

Instead of bringing the user to the computer, let us take into account that people live and work in a real physical world, and make this notion - the concept of space - integral to the operation of our computer systems. We need to make computer systems aware of the physical environment - shifting part of the onus of understanding from user to machine. Awareness comes through sensing, and that implies the need for appropriate sensor technologies to collect status and location data. Applications can now be aware of their physical environment. They know where people and devices are and what devices can do. Crucially, the user interface is no longer based on some abstract metaphor of a physical object, but can be based on space itself. So, when I walk into my study at home, my PC seamlessly and automatically displays the desktop from my office machine - my proximity to my home PC is the prompt to the user interface. I pick up a CD cartridge in my office, and as I open it the appropriate sound track immediately starts to play. Again, a real, intuitive physical action initiates an appropriate response, made possible by an underlying computer system in which location and status data extends throughout the physical environment. I call this approach sentient computing.

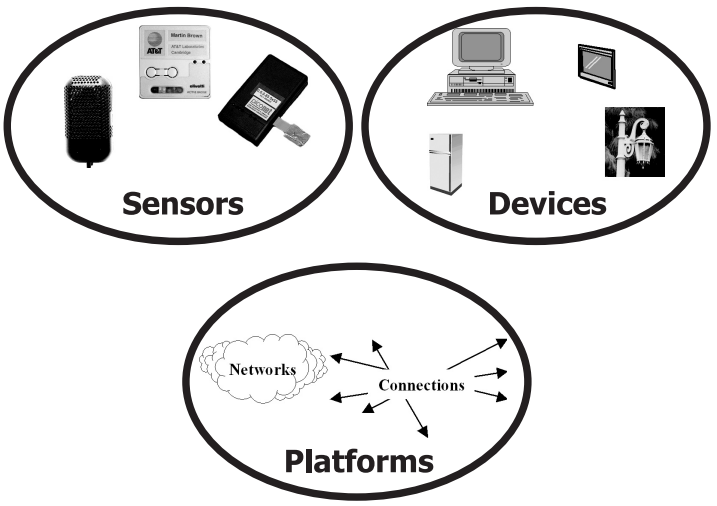


Fig. 1 Components for Programming with Space

The ultimate justification and test of sentient computing will be its capacity to deliver benefits to users, enabling them to interface directly to devices and expressing complex configuration requirements in a simple

way. Reaching this goal depends on our capacity to address a broad spectrum of conceptual and technical challenges.

The central requirement in any computer application is the need to achieve the right conceptual mapping between the physical and the logical. Applications are about physical things - people, PCs, telephones, printers, whatever. A computer program is ultimately a logical abstraction, and the art of the system designer lies in bridging the conceptual gulf between these two radically different domains. The first challenge in sentient computing is to determine the appropriate meeting point between the physical notation of space and the logical constructs of our computer system. Do we do it at the ad hoc, application-specific level? That would work, but at the expense of programmers constantly 're-inventing the wheel'. Different sentient computing applications will share common features and attributes, suggesting a systems-level approach might be more appropriate, with standardised support for programs that have to capture and express the concept of space. Alternatively, we could make space an integral part of the programming language. But that would necessitate the creation of libraries for representing standardised 3D objects. Any library comprehensive enough to be universally applicable would, almost certainly, be over specified for the great majority of applications.

Underlying all this, are twin problems of computational efficiency and performance. Our logical representation of space has to be appropriate to the application in hand. But what do we mean by appropriate? Too exact a representation will make the task of maintaining our store of spatial data difficult. We want systems that react to our actions with no perceptible delay, necessitating the updating of spatial information in real time, or near-real time. So how should we associate spatial properties with things? Do we have to use a 3D representation? Under what circumstances would a 2D representation suffice? Can we use regions around things? Above all, what are the basic properties of things and the logical constructs necessary to our computing models that will turn the vision of sentient computing into an everyday reality?

The sentient computing project at AT&T Laboratories Cambridge and the University of Cambridge is an experimental programme designed to provide answers to these questions. The work of the programme is structured around three basic themes: sensors (that tell us about the spatial properties of objects), devices (the PCs, printers, and other output devices used in our applications), and the platforms (that connect sensors and devices together). Surrounding these three basic elements we have the appropriate architecture that gathers all the elements into a complete functioning system (Fig. 1).

"Many PCs are only used to a fraction of their potential. It does not have to be this way."

Sensors

Sensors tell us about the location or position of things. To reflect the requirements of different applications, we take three different approaches to categorising the concept of position. First, there is containment, where we say that an object is within this container, e.g. a room, so that the application could register the fact that I'm in my office or my study at home. Secondly, there is proximity, where we register that we are close to something, and finally we use co-ordinate systems, which provide a point location in space, subject to some error value. These categories are not hard and fast and can blend together. Small containers are very similar to a co-ordinate system, and proximity has much in common with the concept of containment.

Existing systems not primarily designed as sensors can generate a valuable amount of spatial information. In the case of containment, a satellite telephony system might provide an initial containment within one twentieth of the world, which, depending on the number of antennas on the satellite, can then be further partitioned into up to 50 subsections. That is still very course granularity. With GSM, the digital mobile phone system, we can do significantly better, at the expense of worldwide coverage, realising a container some 25-km across. Third generation cellular systems, such as UMTS, will offer a very similar performance. Indoor wireless LANs give us even finer granularity and provide a container about 30m in diameter.

Our first experience of developing a sensor specifically to provide spatial

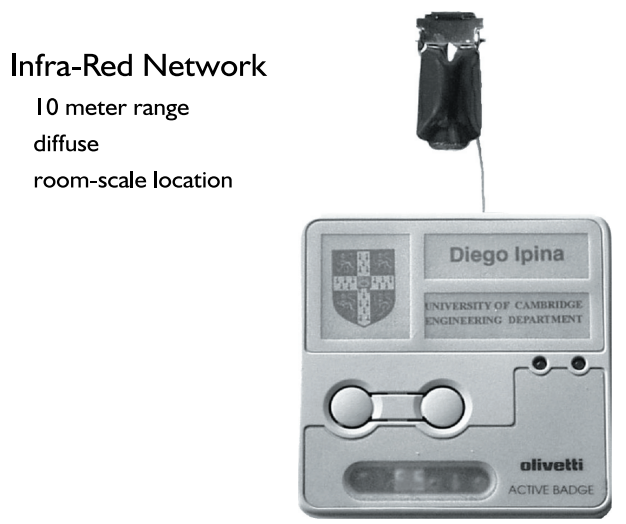


Fig. 2 Containment: Active Badge

information originated in the early 1990s in the form of the Active Badge (Fig. 2). Personnel and equipment can be tagged using the badge, which transmits a unique infrared signal every few seconds. The transmissions are diffuse and receivers in a room pick up the signal, so the badge gives room-scale containment. It tells us who and what is where, and the software system which makes this information available to others, is still very popular. The Active Badge has been the inspiration that got us started on this whole line of enquiry.

In the case of proximity - allowing, for example, a laptop and a telephone to exchange short dialling codes - promising commercial systems are starting to appear. The radio-based Bluetooth system offers a range of round 10 m, while for the infra-red based IrDA the range is more like 3 m. We have built our own wireless-based proximity system, which we call PICOnet.

PICOnet is envisaged as the minimalist building block of our system - the simplest of proximity sensors for the simplest of nodes. It appears to be always on, but uses little power, so the batteries may never have to be changed. Our PICOnet radio operates at 418 MHz, gives a data rate of approximately 5 kb/s, and has a range of around 5 meters depending on the antenna design. There are two modes of operation. In the basic mode, the PICOnet node operates as

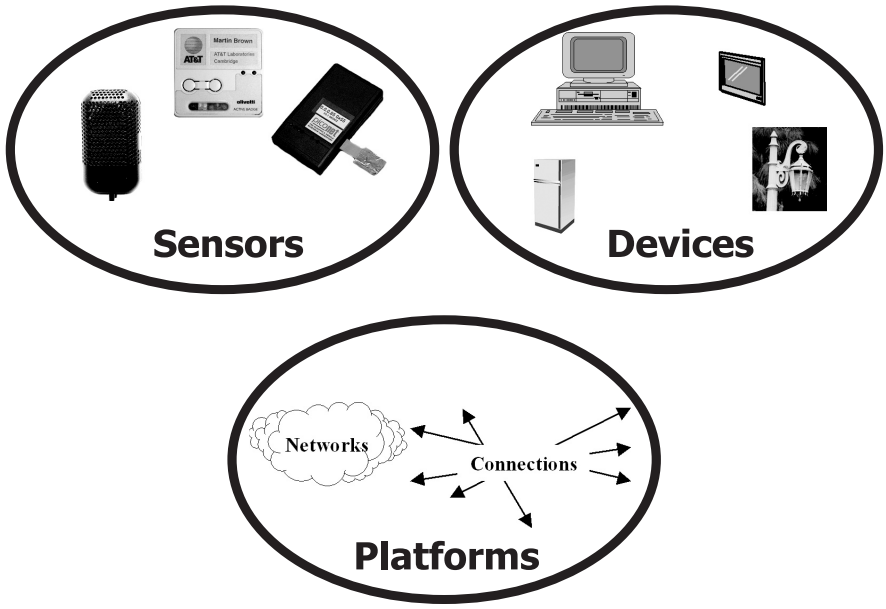


Fig. 3 Proximity: PICOnet Devices

a beacon, i.e. transmit only. The node has a very low power timing circuit, so it can count time with a minimal expenditure of energy. Then at regular intervals, it transmits a very short message, ending the message with an announcement of the time of the next transmission. A slightly more complex variant combines a transmitter and a receiver. This node also transmits at regular intervals, but after every transmission it starts listening for a short period, and then it shuts down. The challenge is to exploit the facilities offered by these simple operational modes in order to effect the seamless interworking of devices in close proximity.

There are three issues we have to deal with: discovery, description and communication. Consider a situation where there are billions of PICOnet devices all over the world. They are mostly inactive, but nevertheless, should by chance two nodes happen to pass, then they have to wake up and register each other's presence. This fundamental discovery problem can be addressed in a number of ways. Probably the simplest is the bilateral rendezvous, where the node that operates as a receiver/transmitter switches on its receiver and listens for a possible transmission from an adjacent node. Once a transmission is detected, the time for the next transmission will be known, and the two nodes can then operate in a deterministic manner. Another possibility is third-party rendezvous where a node is held permanently in receiver mode, possibly drawing power from a plentiful source. This node acts as a source of information on all local transmissions, and can therefore facilitate the discovery process for other node-pairs in its neighbourhood.

"We need a platform for connecting and displaying on all these interesting devices in a ubiquitous way."

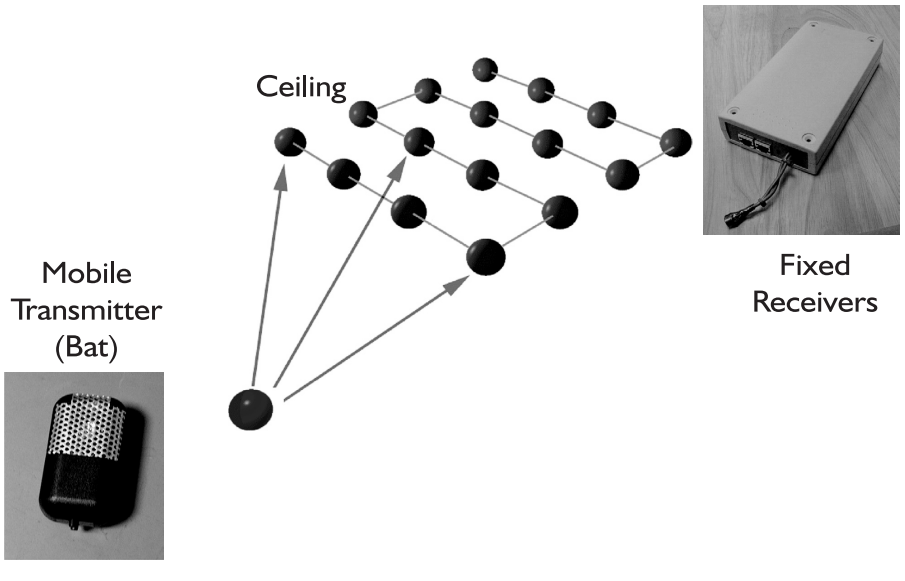


Fig. 4 Co-ordinate: Active Bat

All the nodes in a PICOnet system are completely general purpose, with

every node responsible for describing its services and requirements to the rest of the world. This description function is provided by a node's attribute store.

For communication between PICOnet nodes we can use an attribute store as a sort of bulletin board whereby node A posts messages to node B. There is no support for hop-by-hop routing which is the simplest way of maintaining the objective of location by proximity. If node A receives a message from node B, and node B describes itself as a fan, then node A knows it is close to a fan - there is no other way it could have got the original message.

Effecting a consistently reliable rendezvous between nodes, which spend most of their time in a deep sleep mode, remains a fundamental problem. Another challenge is how to attach such systems to those that can only operate by placing much stricter timing constraints on communicating devices, for example the Internet Protocol.

We have built a whole series of PICOnet-enabled devices (Fig. 3) to help us understand issues like the functioning of a complete system and the interoperation of PICOnet with other systems. My CD cartridge demo uses PICOnet. There is a PICOnet node in every cartridge. You pick it up, you open it and the right music starts to play from nearby speakers. It is simple, easy to grasp, and users understand it immediately.

Co-ordinate systems provide our final approach to categorising position. Outside the Global Positioning System (GPS) can be used, which, when used in combination with maps, has given rise to a large number of applications.

"Applications are the mechanism through which we can test the principles underlying sentient computing."

GPS gives an error value of around 30 meters most of the time, although greater precision can be achieved. At the Cambridge Laboratories we have been working on a co-ordinate system for indoors. This uses a tag, which incorporates ultrasonic transmitters and an array of ceiling-mounted detectors. A detector on the far side of the room will register a pulse later than a detector directly above an object. Using this differential timing information, we can calculate the position of objects to within a few centimetres almost all the time. Bats find their way around using much the same principle, so we call our system the Active Bat (Fig. 4). Fix two transmitters on a rigid object and you can work out its orientation. The Active Bat is a very versatile system; clearly there will be many sentient computing applications that do not require this level of precision and refinement. However, as a research tool, it is providing us with valuable information on what can be done when you have very detailed in positional data.

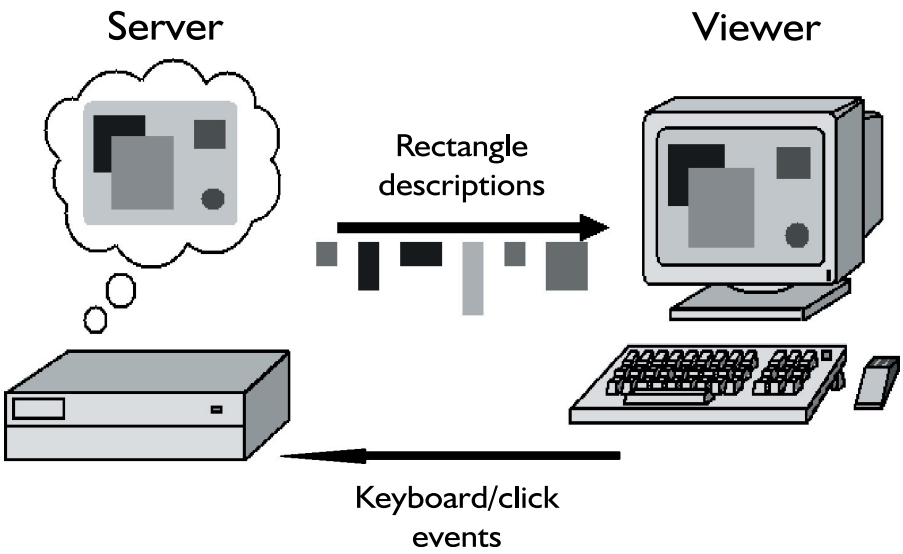


Fig. 5 VNC - The Platform

Devices and platforms

In sentient computing, a device can be anything that takes output from the distributed computing environment. Naturally, this includes conventional computing devices like workstations, PCs or the various forms of personal digital assistant (PDA), but it also embraces consumer products like refrigerators and microwaves, and new devices into the future. We need a platform for connecting and displaying on all these interesting devices in a ubiquitous way.

One way to do this is to tunnel connections to all devices using a simple device-independent protocol. We have devised one such ubiquitous platform called the Virtual Network Computer (VNC). In our approach the viewer, at the receiving end of the connection, has no state, it is just something that visually displays information. All the processing is centralised on a server at the sending end of the connection. Because the viewer has no state, it does not matter if it crashes. The application carries on running, and the user can simply switch to another display device. The other direction, viewer to server, is also stateless - it is just key strokes and clicks - making our viewer a particularly simple version of the so-called thin client (Fig. 5.)

The absence of state eliminates any requirement for synchronisation. You can leave your desk, go to another machine, whether next door or on the other side of the world, reconnect to your desktop and finish the sentence you were typing. Even the cursor will be in the same place. The appearance

is of total mobility, although all we are doing is showing a display in different locations.

The technology underlying the VNC is a version of the remote frame buffer protocol. At the server end of the connection everything we want to display is decomposed into a series of rectangles, with every rectangle characterised by its size, colour and position on the screen. The rectangle descriptions are sent to the viewer, which recreates the original image by redisplaying the individual rectangles. As the viewer requests the next set of updates the protocol can cope with servers and clients of varying speeds. It is a bit like the old character-based dumb terminal, only now we are displaying rectangles rather than characters.

The low-level nature of the protocol is the key to device independence, providing a platform that supports the connection of any device to anything (Fig. 6). The connections can be one-to-one (fixed or mobile), and the streams can be split giving one-to-many, many-to-one, and many-to-many.

There is a potential difficulty associated with this model of stateless viewers in which everything is potentially connected to everything else. We have already established that timing constraints mean that there is a fundamental problem in effecting a rendezvous, or connection, between pairs of PICOnet nodes. Now we are postulating a model of computing built on the premise of universal interconnection.

Architecture

We have sensors generating a wealth of location information; we have devices and a platform for connecting to any device. Now we need to glue everything together, providing our applications with suitable abstractions to support space-aware programming.

Our sensors provide raw spatial facts about objects. They tell us where an object is, and, possibly, the direction

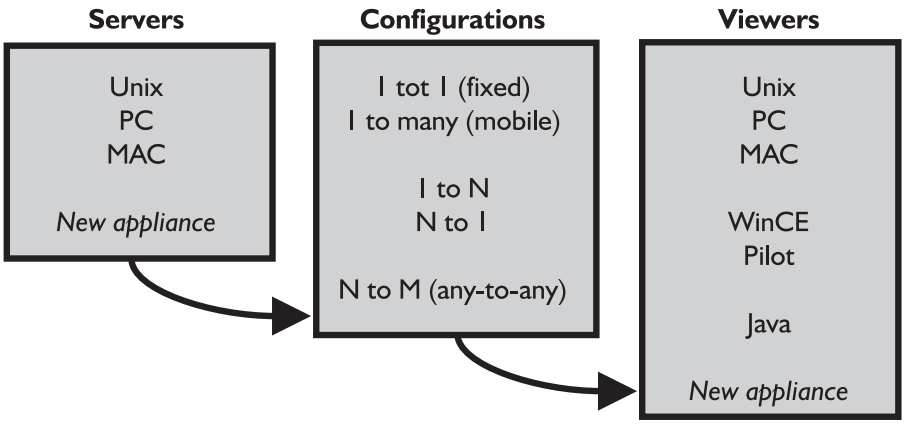


Fig. 6 VNC - Configurations

in which an object is pointing. Location-aware applications need more than raw spatial data, they need to be notified of spatial relationships between objects that are significant in terms of advancing the execution of the application. But how do we decide whether a spatial relationship is significant? The approach we have adopted operates on the basis of zones of containment surrounding objects. In Figure 7 (image on the left) X represents a person and K a keyboard. Now suppose we have an application that needs to be notified when person X is in a position to use keyboard K - when X is possibly 'holding' K. If the zone of confinement of K overlaps the zone of confinement of X, then the holding condition is held to be true and the application receives the appropriate trigger. The situation on the right of Figure 7 indicates how this principle could be applied to support a multi-camera video conferencing system, giving participants the freedom to look in different directions while talking, or even walk around their offices.

Note Figure 7 is a 2D representation of what in reality would be a 3D environment. This simplification can be made because, in general, people and objects tend to remain relatively fixed in the vertical plane. However, the principle can be extended to 3D if required.

The principle of turning raw spatial data into application-significant events through geometric containment and overlapping is reasonably straightforward. You can think of it as the mouse/desktop metaphor mapped onto the real world. However, once you start thinking about real applications, with a population possibly comprising hundreds of thousands of objects, then there is the problem of how to implement this principle in a computationally efficient manner. Every time an object moves, a calculation must be done to identify possibly significant overlaps and send the appropriate application callbacks. In a realistically sized system, there could easily be a large number of object moves every second. It is thus necessary to represent the containment regions with flexibility of precision together with reference counts of how applications have registered interest at a particular level.

Now we can put our architecture together to see how it supports applications. It starts with the sensor events, which are related to the movement of real objects. Applications register the set of objects in which they have a particular interest, and are fed callbacks indicating the occurrence of significant spatial relationships between objects. These callbacks are generated via geometric containment

and overlap. When an application receives a callback, it executes an appropriate action as specified by the application program.

The operational system that has been built uses a variety of sensors; allows space representations to change quickly; provides an appropriate governing event logic; uses caches and proxies to handle large volumes of data quickly; and executes in real time to satisfy a human in the loop.

Applications and the future

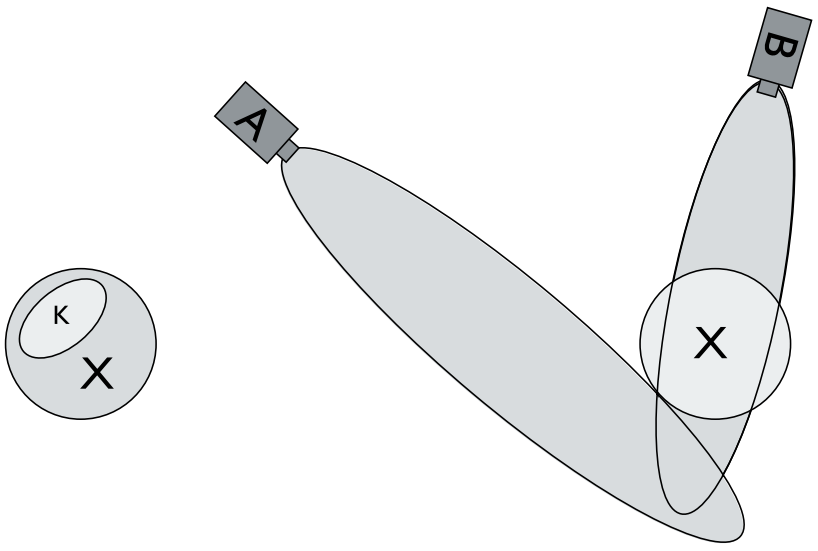
Applications are the mechanism through which we can test the principles underlying sentient computing. The automatic generation of my office desktop on my home PC is made possible by a variant of an application we call the 'follow-me desktop'. This uses an Active Bat desktop sensor, with the ability to register significant spatial relationships between the desktop and the viewer, i.e. it can determine whether the viewer is facing the desktop and vice versa. Once this containment overlap has registered, then the application tunnels the user's desktop onto the new device, be that a workstation, a PC, the refrigerator door, or some new device yet to be invented. The platform that makes this possible is the VNC technology, with its capacity to re-route whatever desktop, to wherever you like and then display it on whatever you like. I do not have the Active Bat system in my home, but I do have the Active Badge, and I can use this to register my co-location with my home PC. Four steps take me across my study to my desk, and in this time my desktop is on my screen ready for me to start work.

Much of the potential attractions of video conferencing can be undermined by the need for speakers to address a single camera throughout the duration of the call. It may seem unnatural, people want to feel free to look around, maybe even get up and walk around their room. A possible solution is to have multiple cameras in a room, combined with some technique for determining which camera to use at any instant. Machine vision and scene analysis is one of the most difficult and challenging research areas so using such technologies is unlikely to provide the same level of robustness as a tagged system. The 'follow-me video phone' uses a sensor, the Active Bat, and a display device that is fast enough to provide a VNC moving video image.

So where is this entire research theme heading? Initial applications of sentient computing will almost certainly be within vertical markets. It is possible PICOnet-based guidebooks will enrich our visits to museums and art galleries; while the VNC-based follow-me desktop has obvious attractions as a means of distributing personal desktops throughout a closed working environment such as the hospital or factory. However, it would be surprising, and not a little disappointing, if the long-term role of sentient computing was confined to such geographically restricted and application-specific domains.

We live in a world in which computing, and the technology to interconnect computers, becomes cheaper year by year. In due course, it is likely that there will be hundreds of communicating devices for each one of us. How then will all these devices be administered? How will they interoperate? And how will they be personalised to that we know how to use them? It may be this will be done automatically, through a process in which physical information about the position of objects is likely to be as important as logical information about their relationship. In short, programming with space - possibly the key to ubiquitous, pervasive, sentient computing and the communications world of tomorrow.

I thank the following who have been in involved in the sentient computing project or have otherwise helped me with this lecture: MD Addlesee, F Bennett, DJ Clarke, R Dettmer, AC Harter, SE Hodges, AH Jones, TJ Richardson, Q Stafford-Fraser, PJ Steggles, AMR Ward, MV Wilkes, KR Wood.



Person X is "holding" keyboard K

Person X can be "seen" by camera B but not by camera A

Fig. 7 Evaluating Spatial Facts

References can be found at: www.uk.research.att.com/~hopper/publications.html

A Hopper, Phil. Trans. Roy. Soc. Lond. A358, pp. 2349-2358 (2000)

Programma

Woensdag 11 december 2002

09.30 - 10.00 Inloop en ontvangst met koffie
10.00 - 10.10 Opening door de dagvoorzitter
10.10 - 11.10 Plenaire sessie 1
11.10 - 11.30 Koffie
11.30 - 12.30 Plenaire sessie 2
12.30 - 13.30 Lunch & demonstraties
13.30 - 14.15 Parallelle sessie 1A & 1B
14.30 - 15.15 Parallelle sessie 2A & 2B
15.15 - 15.35 Koffie
15.35 - 16.35 Plenaire sessie 3
16.35 - 17.05 Forumdiscussie
17.05 - 17.15 Afsluiting door de dagvoorzitter
17.15 - 18:00 Borrel

Inschrijven kan op: www.smart-surroundings.nl

Plenaire sessie 1: Sentient Computing

Prof. Dr. Andy Hopper, Cambridge University

Sentient computing stelt dat (mobiele) applicaties nuttiger kunnen worden en beter kunnen functioneren door de fysieke wereld te observeren en op ontwikkelingen hierin te reageren. De oplossing hiervoor vereist omgang met gegevens van sensoren gecombineerd met het veranderen van het gedrag van willekeurige apparaten.

Plenaire sessie 2: Unobtrusively Augmented Environments

Prof. Hans Gellersen, Lancaster University

Ambient intelligence heeft de potentie om ons huidig dagelijks leven te verbeteren. Deze sessie behandelt onder andere netwerksystemen die verschillende voorwerpen met elkaar verbinden en het toevoegen van embedded systems aan gewone gebruiksvoorwerpen. Ook wordt aandacht besteed aan het gebruik maken van rapid prototyping om nieuwe ideeën uit te werken door middel van het Smart Its platform.

Parallelle sessie 2A: Wireless Sensor Networks

Dr. Ir. Koen Langendoen, TU Delft

Grote technologische vooruitgang is geboekt in de ontwikkeling van low-cost sensoren met een eigen draadloze netwerkinterface (Wireless sensor networks). Netwerken met daarin honderden of zelfs duizenden sensoren bieden grote mogelijkheden voor applicaties voor bijvoorbeeld de industriële sector, de zorgsector en voor thuiswerken.

Parallelle sessie 2B: Design Methoden in Context

Dr. Gerrit van der Veer, Vrije Universiteit Amsterdam

Met behulp van een aantal Design Methoden en scenariotechnieken wordt het proces van ontwerpen van modellen voor Ambient Intelligence vergemakkelijkt. Deze sessie gaat in op een aantal verschillende Design Methoden en legt tevens de link tussen dergelijke methoden en Ambient Intelligence.

Plenaire sessie 3: Multimodale Interactievormen

Dr. Msc. David Keyson, TU Delft

Bij Ambient Intelligence worden multimodale interactievormen, zoals touch screens en speech, als alternatieven gebruikt voor de klassieke interactiemiddelen, zoals toetsenbord, muis en monitor. Volgens deze sessie moeten systemen die over multimodale interactievormen beschikken niet in huidige producten verwerkt worden, maar van de grond af opgebouwd worden.

De tijden veranderen, jullie ook, en wij, het nieuwe bestuur van de studievereniging Inter-Actief, surfen mee. Als Vice-Voorzitter en Promotie Functionaris werd mij gevraagd om iets te schrijven voor al onze leden. Jullie hebben immers ook het recht om te weten waar Ivo bijvoorbeeld jullie met pijn en moeite opgebrachte contributie aan spendeert en waarom een ander bestuurslid, Wouter, verdacht minder op de campus aanwezig is dan voorheen.

Ilse Fokker

Het bestuur

stelt zich voor...

Wegens chronisch gebrek aan tijd en geld bij de meeste studenten was het in eerste instantie nogal moeilijk een compleet bestuur te vormen. Na een laatste noodactie ("Inter-*Actief* op Non-*Actief*?"), die de meesten van jullie hoogstwaarschijnlijk niet mee zullen hebben gekregen, kwamen er dan toch plots 2 mensen bij. Met veel genoegen bestond het kandidaatsbestuur nu uit zes mensen. Net op tijd: het 24e bestuur van Inter-Actief van het jaar 2002/2003 was gevormd en ging hard aan de slag met het maken van het beleidsplan. Maar daarover zo meer.

Wie zijn die 6 helden in groene blousjes? Ze luisteren naar de namen Ruben Smelik, Edwin van der Ham, Ivo Grondman, Wouter Wiegmans, Ilse Fokker en Remy Smeets. Volgens velen toch een combinatie van kleurrijke persoonlijkheden. Als eerste onze voorzitter Ruben. Deze 3e jaars INF-er doet als student-assistent op het moment flink zijn best anderen aan studiepunten te helpen, maar voor hemzelf staat dat het komende jaar op een wat lager pitje. Hopelijk voor hem komt zijn ervaring als Voorzitter bij de I/O Vivat nog goed van pas. Edwin daarentegen heeft het al wat langer rustiger aan gedaan. Hij is inmiddels 5e jaars Informatica, houdt zich bezig met tafeltennis, en kan zich dit jaar helemaal uitleven op onder andere de notulen als Secretaris. Derde persoonlijkheid in deze groep is voor het eerst in de geschiedenis (jawel!) een TEL-er. Ivo is derdejaars en volgt Penningmeester Erik op in het bijhouden van de financiële zaken van Inter-*Actief*. Deze taak is hem op het lijf geschreven aangezien hij zelfs in het weekend nog bezig is met geld als croupier in het lokale casino. Wouter, eveneens TEL, ontfermt zich over de externe zaken en zoekt de contacten meer in het bedrijfsleven. Zeer belangrijk werk, want zonder onze sponsoren komen we niet ver met Inter-Actief! Hierna belanden we bij de schrijvster van dit stukje. Als enige BIT-er zorg ik voor de promotie en allerhande activiteiten daaromheen. Mocht Ruben last krijgen van RSI, dan kan ik ook nog de hamer overnemen als vice-voorzitter. Hopelijk hoeft dat niet, want samen met Remy, die Interne zaken regelt, moet er wel veel gedaan worden! Remy is de "jongste", hij is eerstejaars Informatica student. We verdenken hem er echter van al langer op de campus rond te lopen.

Welnu, we waren nog maar net rond of de eerste grote uitdaging kwam al in zicht: het schrijven van een beleidsplan, en wel snel! Geen tijd voor ontspannen kennismaken, kansloos uitgaan of zelfs maar slapen, want er wachtte ons veel werk.

We gaan ons dit jaar richten op het meer betrekken van alle leden bij al onze uiterst vermakelijke activiteiten op het gebied van ontspanning en sociale ontwikkeling!

Ons hoopvolle beleidsplan moet uitgevoerd worden. De borrelkelder moet verbouwd worden, meer uitjes, meer reizen, meer boeken, meer EWI (de kersverse faculteit), noem maar op. Allemaal hebben we er erg veel zin in, en met een grote dosis enthousiasme hopen we er een heel productief jaar van te maken. Dat mag ook wel, er is veel te doen!

Veel te doen, ook voor jou! Zoals bij wel meer verenigingen geldt ook bij Inter-Actief het "voor en door leden". Dus we zijn nu eigenlijk wel benieuwd naar jullie meningen. Wat vind jíj ervan? Wil je een magnetron in de kamer? De totale macht aan het bestuur? Dit, en serieuzere vragen zullen één dezer dagen aan jou gesteld gaan worden. Je kunt er nog een prijsje mee verdienen ook.

Mocht je na het lezen van dit stuk meer over ons willen weten, kom dan gerust langs in de Inter-*Actief* kamer. Ondanks alle moderne communicatietechnologie zien we jullie toch het liefst in levende lijve om jullie onze plannen toe te lichten, of nieuwe ideeën op te doen.

Slimme omgevingen

Het symposium van 11 december gaat over slimme omgevingen. In de woonkamer, de keuken en op het werk zullen complexe systemen worden geïnstalleerd, waarmee met behulp van informatie- en communicatietechnologie een slimme omgeving wordt gecreëerd. Eigenlijk doet de Universiteit Twente dit al jaren, alleen dan op een geheel andere wijze.

Zo'n 41 jaar geleden werd hier in Enschede namelijk de campus gebouwd. Een campus waar een groot deel van de studenten en medewerkers kan wonen en werken. Het is dus één grote omgeving zijn met slimme mensen, de bewoners hebben immers óf vwo óf hbo afgerond en studeren/onderzoeken nu op een academische niveau. Een redelijk slimme omgeving dus.

Naar de wenselijkheid van deze slimme omgeving zal even niet gekeken worden. Het levert immers goede onderzoek- en onderwijsresultaten op, die zeker de moeite waard zijn. Studenten die afgestudeerd zijn, vallen echter in een diep zwart gat. Gewoon weer terug in de grote normale mensenwereld, waar alles toch een tikkeltje anders gaat.

Veiligheid op straat is daar bijvoorbeeld een hot item, internet is veel minder snel en mensen hebben een naam, geen icq-nummer! Bier wordt over het algemeen niet 's ochtends al genuttigd, rond die tijd staan normale mensen namelijk in de file naar het werk. Daarnaast staan straatnaambordjes aan het begin van elke straat en bevinden deze zich expliciet niet in een willekeurige woonkamer van desbetreffende straat.

Daar moet de zojuist afgestudeerde student natuurlijk aan wennen. Na vijf, zes, zeven of zelfs meer jaar isolatie van deze normale wereld kan dat namelijk best een schok zijn. Het is daarom lovenswaardig dat Inter-Actief hiervan het belang in ziet en een symposium organiseert over deze slimme omgevingen, Smart Surroundings.

Wij zouden geen voorzitters zijn, als ons, al dan niet per ongeluk, de uitkomst van het symposium niet ter ore was gekomen. De planning is dat binnen 2 à 3 jaar overal in het land zogenaamde Smart Surroundings locaties worden gebouwd, die alleen toegankelijk zijn voor studenten en oud-studenten van de Universiteit Twente.

Zo'n Smart Surrounding kun je zien als een soort dagcafé. Even helemaal weg van de dagelijkse beslommeringen en met lotgenoten in een nagebootste slimme omgeving praten hoe het vroeger was. Uiteraard met de mogelijkheid tot het nuttigen van een biertje of een sterke bak koffie. En terwijl je op internet aan het chatten bent met een lotgenoot in een andere lokale Smart Surrounding ergens in het land, valt je op dat de catering er zelfs aan heeft gedacht op de achtergrond Skyradio na te bootsen, om het optimale 'ohja'-gevoel te beleven.

Dit zal de symposiumcommissie waarschijnlijk niet voor ogen hebben gehad, toen ze een thema bedachten voor haar symposium. Het zou ook gek zijn dat speciaal voor dit onderwerp sprekers uit het buitenland worden overgevlogen, terwijl twee simpele voorzitters het ter plekke bedenken.

Het creëren van slimme omgevingen is echter niets mis mee. In een wereld waar gemakzucht, luxe en comfort zege vieren, is dit een gat in de markt. Onze vraag daarentegen is of de mensen van deze Smart Surroundings al dan niet slimmer of dommer worden. Zo hoeven ze bijvoorbeeld niet meer te onthouden dat de melk over datum is, het systeem heeft deze melk allang verwijderd en nieuwe besteld.

Wij wensen u een leerzaam symposium toe.

Maarten Donders en Ruben Smelik

Maarten Donders & Ruben Smelik

De GameCom is een commissie van Inter-Actief. Met 5 actieve mensen zijn ze nu bezig met het ontwikkelen van een realtime-strategy spel. In dit artikel lichten ze het een en ander toe.

Wil je meer weten? Mail dan naar gamecom@inter-actief.utwente.nl.

Eric van der Sluis
Oebele van Veen

GameCom

In het golvende heuvellandschap staan de grazende witte schapen in de ochtend mist tussen de eeuwenlang ongeschonden vegetatie. Het ochtendlicht weerkaatst in een gele gloed op de bladeren en het bedauwde gras. De scherpschutter ligt tussen een struik en rots op de koude grond wolkjes uit te blazen. De geur van olie drijft op de lucht als de voorbode van mechanisch geweld. Hij luistert naar de ruisende, trappelende brekende geluiden van ergens achter de heuvels en stelt zijn vizier bij. De schapen grazen. Door de telelens kijkend ziet hij de beestmassa over de groene heuvels razen, dichterbij en dichterbij. Tussen de glimmende klauwende massa lopen grotere bevelende ondingen.

De schapen grazen niet meer, waar ze waren is nu een grote wriemelende bewegende massa die zich rond het onding opdringt. De scherpschutter richt. De geur van olie neemt toe. Een stampend geluid mengt zich van achteren met de knarsende geluiden. Achterom gekeken ziet de schutter tegen het licht een grote strakke mechanische vorm en houd zich stil. De zwerm in het dal reageert agressief op een nieuw ratelend zoemend geluid en begint de heuvel op te spurten naar de vorm.

Een aardeverscheurende herrie overstemd alles. De voorsten worden verscheurd en uiteengespreid over de achterlopers. Beesten glijden uit. Gekrijs, gespat, gebreek. De heuvel flank verandert langzaam in een glijbaan van ingewanden waartegen wanhopige beesten omhoog proberen te kruipen. De herrie houdt aan, het gekrijs van de beveler is niet te horen als hij uiteengereten wordt door brandend lood. Drukkende voelbare stilte als de geweren ophouden. De geur van olie en verbrand vlees drijft tussen de eeuwenoude vegetatie. De overwinnaar vertrekt, de scherpschutter blijft verdoofd achter.

Waar leidt het toch allemaal toe? Elk jaar weer worden er spellen ontwikkeld die de rest voorbij streven in grafische prestaties, spelconcepten en gebruikersinterface. Professionele bedrijven zijn jaren bezig om een dusdanig goed spel te ontwikkelen dat het met de juiste marketing tot nieuwe verkooprecords zal leiden. Hierbij is meer en beter steeds een belangrijk onderdeel. Toch is duidelijk op te merken dat er slechts een aantal echte toppers zijn in verkoopcijfers. Opvallend is dat deze spellen op technisch gebied geen echte uitblinkers zijn.

Het ontwikkelen van een spel biedt een uitdaging voor een commissie. Natuurlijk willen wij het beste leveren en natuurlijk willen wij dat alles wat bestaat in het niet valt bij wat wij produceren. Al snel kom je er achter dat optimaler programmeren geen directe grote verbeteringen in het spel tot gevolg heeft. Dus verdiep je je in de doelgroep, de mensen die het spel zullen gaan spelen. Zij zijn immers de mensen die het eindproduct tot zich zullen nemen als een hongerige met chitine bepantserde monstruositeitenmassa op een kudde grazende schaapjes.

Wat wil de huidige spellenspeler? Eigenlijk wil hij gewoon even een spelletje kunnen spelen, het liefst met een huisgenoot om dan met een gevoel van intens geluk al zijn agressie botvieren op zaken die geen weerstand bieden aan absolute vernietiging. Duidelijke benodigdheden hiervoor zijn massa's en bloed, die na enige inspanning best op het scherm te tonen zijn. Verder wil de huidige generatie steeds minder te hoeven doen om tot dit doel te komen, maar toch alles tot in de details in hun hand te hebben. Het klinkt bijna alsof men tegenwoordig alles wil, maar gelukkig is ook dit geen probleem voor de goeroes die er nu mee aan de slag zijn.

GnuPG, een afkorting voor "GNU Privacy Guard", is een hulpmiddel om veilig gegevens te kunnen verzenden en op te slaan. Het kan daarom gebruikt worden om bestanden te versleutelen of om ze van een digitale handtekening te voorzien.

Berteun Damman

GnuPG

Een versleutelingshulpmiddel voor persoonlijke communicatie

Geschiedenis

Mensen hebben altijd al geprobeerd hun gegevens te versleutelen en in de loop van de tijd zijn de methodes steeds geavanceerder geworden. Echter lange tijd waren de hulpmiddelen om gegevens goed te beveiligen slechts in handen van overheden. In 1991 ontwikkelde Philip Zimmermann "Pretty Good Privacy", afgekort wel PGP genoemd. Het doel van PGP was om iedereen goede versleutelingsmogelijkheden te bieden. Zimmermann ontwikkelde PGP in de Verenigde Staten, daar stond de wet echter niet toe om krachtige versleutelingsgereedschappen te exporteren en daarom werd hij in 1994 aangeklaagd. In 1996 hebben de VS de zaak echter geseponeerd. Ondertussen is PGP opgekocht door Network Associates en weer afgestoten en momenteel is het in handen van de PGP-corporation.

GnuPG is opgestart als een vrij alternatief voor een gedeelte van de PGP-verzameling, die meer hulpmiddelen bevat, zoals bestandssysteemversleuteling. Het wordt ontwikkeld buiten de Verenigde Staten zodat de ontwikkelaars geen problemen hebben met exportrestricties en verder wordt het gebruik van gepatenteerde algoritmen vermeden. GnuPG is, zoals de naam al doet vermoeden, een project van de Free Software Foundation en de broncode is beschikbaar onder de GPL-licentie.

Werking

GnuPG werkt met de inmiddels redelijk bekende "public/private-key encryption". Dit houdt ongeveer in dat iemand over een sleutelpaar beschikt: een publieke of openbare sleutel en een geheime sleutel. De geheime sleutel - de naam zegt het al - moet geheim gehouden worden en de openbare sleutel kan openbaar gemaakt worden. Stel dat Alice en Bob berichten willen uitwisselen, dan geeft Alice haar openbare sleutel aan Bob en Bob geeft zijn openbare sleutel aan Alice. Ze kunnen nu elkaars openbare sleutel gebruiken om een bericht te versleutelen. Zodra dit is gebeurd kan het bericht alleen nog met de bijbehorende geheime sleutel ontsleuteld worden. Aangezien de geheime sleutel in principe nooit verstuurd wordt, kan deze niet onderschept worden door kwaadwillenden en is de communicatie veilig.

Er zijn echter nog meer doeleinden voor de geheime en openbare sleutel. Deze kunnen ook gebruikt worden om bestanden van een handtekening te voorzien, als Alice een programma heeft gemaakt voor Bob dan kan zij dit programma vergezeld doen gaan van een digitale handtekening die alleen met haar geheime sleutel gemaakt kan worden. Als Bob het programma dan krijgt dan kan hij verifiëren dat er niet met het bestand geknoeid is door met Alice' publieke sleutel de handtekening te controleren.

"GnuPG is een vrij alternatief voor PGP"

De veiligheid van het algoritme is erop gebaseerd dat het ontzettend veel tijd kost om de geheime sleutel te kraken wanneer je deze niet hebt. Ter vergelijking: het distributed.net-project is onlangs afgelopen. Dit project wist na 1757 dagen hard rekenen een 64-bit sleutel te kraken. De rekenkracht van het project was op z'n top vergelijkbaar met 45.998 2GHz AMD Athlon XP computers, met die snelheid had de sleutel in 790 dagen gevonden kunnen worden. Kortom: Het zijn geen sleutels die je zo even kraakt en GnuPG gebruikt meestal ook nog moeilijker te kraken sleutels.

Het grootste gevaar bij cryptografie ligt dan ook niet in het al dan niet te kraken zijn van sleutels maar of een sleutel wel daadwerkelijk van degene

is van wie die sleutel beweert te zijn. Als Alice meent dat de sleutel van Bob is, maar deze blijkt stiekem door Eve gemaakt te zijn stuurt ze allerlei geheime zaken naar Bob die alleen voor hem bestemd zijn, maar Eve kan ze dan lezen.

Een wiskundiger uitleg van het algoritme staat bijvoorbeeld in het Algebra voor Informatica dictaat op pagina 27 tot en met 31.

Toepassingen

De meest gebruikte toepassing van GnuPG is het versleutelen van e-mail die je verstuurt. Dit lijkt wellicht erg paranoïde gedrag, maar sinds april 2001 is er in Nederland een aftapverplichting voor providers van kracht. Dit houdt in dat ze e-mail moeten aftappen als de overheid hierom vraagt. Het Echelonnetwerk is ook geen onbekende naam voor de meeste mensen. Er is dus een gerede kans dat jouw e-mail afgezocht wordt op verdachte woorden of ergens opgeslagen wordt in een groot archief. Wil je liever het zekere voor het onzekere nemen, of ben je gewoon paranoïde, dan is het dus verstandig om in ieder geval je e-mail van een handtekening te voorzien en als het even kan ook te versleutelen. Versleutelen kan echter alleen als de ander ook over een openbare sleutel beschikt. Tekenen kan altijd, en daarmee toon jij aan dat je e-mail daadwerkelijk door jou geschreven is. Daarmee voorkom je dat iemand jou woorden in de mond kan leggen, maar aan de andere kant betekent het natuurlijk ook dat je later niet meer kunt beweren dat iemand met je e-mail geknoeid heeft.

Een andere erg nuttige toepassing van digitale handtekeningen ligt bij het verspreiden van software. Het komt de laatste tijd geregeld voor dat een FTP-server gekraakt wordt en dat de programmatuur die erop staat van zogenaamde backdoors voorzien wordt. Je ziet vaak wel een MD5-som bij een bestand waarmee je kunt controleren of het bestand correct verstuurd is, maar als het bestand moedwillig is veranderd dan kan de dader zonder problemen een nieuwe MD5-som genereren. Dit kan niet bij een digitale handtekening die op een geheime sleutel is gebaseerd, als dan het bestand vervangen wordt klopt de handtekening niet meer en zonder geheime sleutel kan de dader geen nieuwe handtekening maken. Hij zo eventueel wel een hele nieuwe sleutel kunnen maken op naam van de oorspronkelijke eigenaar, dus je moet wel van te voren te sleutel kunnen vertrouwen.

Software

Nu er een scala aan nuttige toepassingen voorbij is gekomen rijst waarschijnlijk de vraag hoe je dit zelf kunt gebruiken. Een vraag waar de rest van het artikel op in zal gaan. Het richt zhoals gezegd specifiek op GnuPG en niet op vergelijkbare software als PGP. Beiden moeten echter onderling uitwisselbaar zijn omdat ze op dezelfde standaard gebaseerd zijn, namelijk de OpenPGP standaard. [Zie: RFC 2440]

De software zelf kan gedownload worden op http://www.gnupg.org. De meeste linux-distributies hebben zelf echter wel een package gemaakt en bij de BSD's zit het in de ports, dus dan is het ook daar te vinden. De software draait sowieso goed op Linux, BSD en Windowssystemen. Op veel andere besturingssystemen zoals GNU/Hurd, Irix en OS/2 draait het ook, maar daar zitten wat haken en ogen aan het gebruik in verband met de toevalsgetallengenerator.

Nadat je de juiste distributie voor je favoriete besturingssysteem hebt gevonden en je ergens een uitvoerbaar programma op je schijf hebt kunnen we verder.Dit kan al naar gelang van het besturingssysteem en het al dan niet zelf compileren makkelijk of minder makkelijk zijn, maar dit valt buiten dit artikel. Het programma is een programma dat bij voorkeur in tekstmodus werkt. Onder Windows dien je derhalve een DOS-prompt te regelen waarin je met het programma kunt werken. Dit kan wellicht even schrikken zijn voor sommige windowsgebruikers. Er zijn echter allerhande programma's beschikbaar die het een en ander in een grafische schil verpakken, een daarvan voor Windows is WinPT, te vinden op http://www.winpt.org.

Zelf aan de slag

Het artikel gaat verder met alleen GnuPG zonder hulpprogramma's. Het eerste doel is om een eigen geheime sleutel aan te maken hiertoe type je in: ($ stelt de prompt voor, dit zou ook C:\> kunnen zijn)

```
$ gpg --gen-key
```

Het programma vraagt nu wat voor soort sleutel je wilt (Oudere versies bieden optie 5 niet):

```
Please select what kind of key
you want:
(1) DSA and ElGamal (default)
(2) DSA (sign only)
(4) ElGamal (sign and encrypt)
(5) RSA (sign only)
```

De standaardkeuze voldoet voor de meeste gebruikers. Met optie 1 wordt een DSA sleutel gemaakt om handtekeningen te maken en een ElGamal sleutel om bestanden te coderen. Je kunt later altijd nog extra sleutels toevoegen om te gebruiken bij coderen of het genereren van handtekeningen. De standaardkeus voldoet hier. Optie 3 ontbreekt omdat je daarmee geen handtekeningen kunt maken.

"De meest gebruikte toepassing is het versleutelen van e-mail"

Nadat de keuze voor (1) gemaakt is - maak je een andere keuze dan zul je verder jezelf moeten redden - komt de vraag hoe groot de sleutel moet zijn.

```
About to generate a new ELG-E
keypair.
minimum keysize is 768 bits
default keysize is 1024 bits
highest suggested keysize is
2048 bits
```

```
What keysize do you want?
```

De grootte kan variëreren van 768 tot 2048 bits. Een grotere sleutel is uiteraard veiliger maar ook langzamer bij het versleutelen en ontsleutelen. Ook wordt de lengte van de handtekening er groter door. Het gerucht gaat echter dat Dan Bernstein (de auteur van o.a. QMail) een snellere manier heeft uitgevonden om sleutels te kraken (zie: http://cr.yp.to/papers/nfscircuit.ps).

Wellicht is het dus toch verstandig om voor 2048 bits te kiezen, alhoewel

"Een grotere sleutel is veiliger maar ook langzamer"

anderen zeggen dat Bernsteins manier toch niet zo revolutionair is. Nadat er eenmaal gekozen is zit je aan je keus vast tenzij je helemaal een nieuwe sleutel maakt!

Daarna moet er gekozen worden wanneer je sleutel verloopt:

```
Please specify how long the key
should be valid.
 0    = key does not expire
<n>   = key expires in n days
<n>w = key expires in n weeks
<n>m = key expires in n months
<n>y = key expires in n years
```

Als je geen speciaal doel voor de sleutel voor ogen hebt, zoals een tijdelijke sleutel om te gebruiken tijdens de Pandora spygame, is het meestal goed om geen vervaldatum te kiezen. Als je echter van jezelf weet dat je de neiging hebt om wachtwoorden te vergeten of je harde schijf te formatteren en daardoor je sleutel te verliezen zou je wellicht toch kunnen overwegen om vervaldatum te nemen. Deze datum kan achteraf bijgesteld worden, maar de kans bestaat danda niet iedereen dat op tijd doorkrijgt.

Als je deze keus gemaakt hebt moet je een User-ID opstellen voor de sleutel. Deze ID bestaat uit je naam, e-mailadres en een eventuele opmerking en dit proces ziet er ongeveer zo uit:

```
You need a User-ID to iden-
tify your key; the software
constructs the user id from
Real Name, Comment and Email
Address in this form:
   "Heinrich Heine (Der Dichter)
<heinrichh@duesseldorf.de>"

Real name: Berteun Damman
Email address: berteun@dds.nl
Comment:

You selected this USER-ID:
"Berteun Damman
  <berteun@dds.nl>"
```

Als je nog meer e-mailadressen hebt dan is het mogelijk om deze later ook toe te voegen (dat gaan we ook doen). Hierna moet voorlopig echter een wachtwoord voor de sleutel gekozen worden. Denk goed na over je wachtwoord, gebruik geen voor de hand liggende combinaties met namen van familieleden, je geboortedatum of combinaties van bestaande woorden. Dit wachtwoord kan later ook veranderd worden.

Hierna wordt de sleutel daadwerkelijk gegenereerd, om over genoeg willekeurige data te beschikken helpt het om wat met je muis te bewegen en om wat op je toetsenbord te typen. Er verschijnt nu zoiets in beeld ten teken dat je sleutel gemaakt is:

```
pub 1024D/6EBAEAB0
  2002-10-29 Berteun Damman
  <berteun@dds.nl>

Key fingerprint =
  01AE 671D BFF7 DFD2 C465
  66E2 EF23 01B9 6EBA EAB0

sub 2048g/D70FF7BE 2002-10-29
```

Bij deze sleutel is 6EBAEAB0 het id van de sleutel. Zoals gezegd gaan we nu nog enkele USER-ID's toevoegen aan de sleutel. Start gpg als volgt op:

```
$ gpg --edit-key 6EBAEAB0
```

Waarbij je 6EBAEAB0 uiteraard door jouw id vervangt. Er komt nu de volgende prompt in beeld:

```
Command>
```

Door help in te typen krijg je een overzicht met beschikbare opdrachten. Met passwd kun je bijvoorbeeld je sleutel wijzigen. We gebruiken nu adduid

om een extra gebruikers-id toe te voegen. Dit gaat ongeveer als volgt:

```
Command> adduid
Real name: Berteun Damman
Email address:
  damman@cs.utwente.nl
Comment:

You selected this USER-ID:
"Berteun Damman
  <damman@cs.utwente.nl>"

Change (N)ame, (C)omment,
  (E)mail or (O)kay/(Q)uit? O
```

Hierna kan dit proces eventueel nog een keer herhaald worden om nog meer USER-ID's toe te voegen. Er wordt uiteraard elke keer om je wachtwoord gevraagd. Ben je klaar, dan type je quit in en beantwoord je de vraag of de veranderingen opgeslagen moeten worden bevestigend.

Nadat je je sleutel zo voorzien hebt van de nodige identiteiten is het moment gekomen om een herroepingscertificaat te maken. Mocht je onverhoopt je wachtwoord vergeten of mocht je computer gecompromitteerd worden dan kun je met dit certificaat je sleutel herroepen. Dan weten mensen dat ze geen gebruik meer moeten maken van de sleutel. Als je je wachtwoord nog wel weet kun je nog wel oude berichten decoderen met je geheime sleutel, maar deze kan niet meer gebruikt worden om berichten van een handtekening te voorzien.

Dit certificaat wordt gemaakt door het volgende in te voeren (waarbij je natuurlijk weer je eigen ID invoert voor 6EBAEAB0):

```
$ gpg --output certificaat.txt
  --gen-revoke 6EBAEAB0
```

Het is verstandig dat het certificaat niet in handen valt van anderen, want iedereen kan hiermee je sleutel herroepen. Print het daarom uit, overschrijf het bestand en stop de uitgeprinte kopie in een verzegelde envelop en leg deze in een kluis. Behoort dit niet echt tot de mogelijkheden, berg het dan gewoon goed op.

Je sleutel verspreiden

Nu is je sleutel op zich klaar voor gebruik. Als niemand anders echter je sleutel heeft heb je er natuurlijk ook weinig aan. Je zult je sleutel moeten uitvoeren en aan anderen geven. Als ik mijn sleutel wil exporteren doe ik:

```
$ gpg --armor
  --output mijnsleutel.gpg
  --export 6EBAEAB0
```

Overigens kun je van iedereen de openbare sleutel exporteren, daarom is de sleutel immers openbaar. Deze sleutel staat nu in mijnsleutel.gpg. Omdat de optie --armor is meegegeven is het een ASCII-bestand geworden dat overal neergezet kan worden. Nu kun je dit bijvoorbeeld op een diskette zetten om aan je huisgenoot te geven. Als je huisgenoot dan jouw diskette in ontvangst neemt zal hij of zij de sleutel moeten importeren, dit gebeurt als volgt:

```
$ gpg --import mijnsleutel.gpg
```

Vervolgens zal gpg de melding geven dat de sleutel is geïmporteerd. Nu is het verhaal nog niet afgelopen, want je huisgenoot zal moeten kijken of het daadwerkelijk goed is gegaan. Eerst kan dit gecontroleerd worden door

```
$ gpg --list-keys
```

in te typen. Als het goed is staat de zojuist geïmporteerde sleutel erbij. Daarna doet je huisgenoot

```
$ gpg --edit-key 6EBAEAB0
```

Bij command typt hij of zij fpr (fingerprint) in. Dit heb je zelf ook uiteraard gedaan en je hebt die vingerafdruk opgeschreven. Nadat deze zijn vergeleken en deze hopelijk overeenkomen kun je er waarschijnlijk vanuit gaan dat niemand met je sleutel heeft geknoeid. Nu kan je huisgenoot ervoor kiezen om je sleutel te tekenen, ook wel 'signen' genoemd. Dit moet alleen gedaan worden als de fingerprint daadwerkelijk vergeleken is en je er zeker van bent dat je de juiste sleutel voor je hebt. Om te tekenen moet er "sign" worden ingetypt op de "Command>"-regel.

Het tekenen wordt gebruikt om het valideren van sleutels te vergemakkelijken. Als Alice de sleutel van Bob heeft getekend en Bob heeft op zijn beurt de sleutel van Carol getekend dan is het mogelijk dat Alice de getekende sleutel van Carol direct accepteert mits ze Bob voldoende vertrouwt. In het geval van jou en je huisgenoot zou je huisgenoot (die nog steeds gpg niet heeft afgesloten) "trust" intypen. Nu krijgt hij een aantal verschillende keuzes:

```
1 = Don't know
2 = I do NOT trust
3 = I trust marginally
4 = I trust fully
```

Optie één spreekt voor zich. Als je iemands met twee merkt dan betekent dit dat je weet dat hij of zij weleens sleutels wil tekenen die niet overeenkomen met de juiste persoon, kortom, zijn signatures zijn waardeloos. Bij optie drie zeg je dat hij of zij wel goed controleert, maar dat je hem of haar net iets minder vetrouwt dan jezelf. Door vier te kiezen geef je aan dat je je huisgenoot net zo goed vertrouwt als je jezelftro In de praktijk kunnen deze gegevens gebruikt worden om een zogenaamd "web of trust" te bouwen. Je kunt instellen dat jij sleutels die bijvoorbeeld door drie mensen die jij redelijk vertrouwt zijn getekend ook zelf als geldig beschouwt (en dus kunt gebruiken om te versleutelen). Bij een sleutel die door iemand is getekend die jij volledig vertrouwt kun je kiezen dat jij ook die sleutel zondermeer volledig vertrouwt.

Je kunt deze waardes zelf instellen, al naar gelang je iemand op z'n eerlijk gezicht vertrouwt of dat je al moeite hebt jezelf door dik en dun te vertrouwen.De crux blijft bij GnuPG namelijk het vertrouwen dat je hebt in een sleutel. De sleutels zelf zijn technisch gezien vrijwel onkraakbaar, zoals eerder gemeld, echter als je denkt dat een sleutel van Alice is terwijl deze door Eve gemaakt blijkt te zijn, dan kom je natuurlijk bedrogen uit. Het is daarom van belang om er zeker van te zijn dat degenen die jij vertrouwt ook verantwoord omgaan met het tekenen van sleutels, anders valt je hele web-of-trust in duigen.

Nu ongeveer duidelijk is hoe sleutels geïmporteerd en geëxporteerd moeten worden komt het er natuurlijk op aan om ook daadwerkelijk gegevens te versleutelen met GnuPG en om versleutelde gegevens te ontsleutelen.

Keyservers

Het beheren van sleutels wordt al snel een heel gedoe. Je moet er voor zorgen dat je eigen sleutel goed te verkrijgen is en zodra je een sleutel tekent wil je ook graag dat in principe iedereen die die sleutel heeft die kopie krijgt etc. Voor dit doel zijn er keyservers uitgevonden. Mensen kunnen hun openbare sleutel naar een keyserver sturen en ze kunnen daar openbare sleutels van anderen afhalen. Zodra je dan een sleutel tekent kun je deze weer uploaden naar de keyserver en deze verwerkt dit dan.

"Tekenen wordt gebruikt om het valideren van sleutels te vergemakkelijken"

Het zenden en versturen gaat als volgt:

```
$gpg
 --keyserver certserver.pgp.com
 --send-key 0x6EBAEAB0

$gpg
 --keyserver certserver.pgp.com
 --recv-key 0x6EBAEAB0
```

In het eerste geval verstuur ik mijn

sleutel naar de keyserver en in het tweede geval vraag ikijn sleel op.

Versleutelen en ontsleutelen

In het bovenstaande gedeelte hebben we de hele tijd met getallen gewerkt om de id's aan te geven van een sleutel, maar dit kan ook met het e-mailadres, dat maakt het bij versleutelen en ontsleutelen wat doorzichtiger. Stel ik wil een document naar mijzelf sturen en ik wil dat graag versleuteld doen. Dan zorg ik dat het origineel heb, bijvoorbeeld "verslag.tex". Ik voer dan het volgende commando in om het te versleutelen:

```
$ gpg --output verslag.tex.gpg
  --recipient berteun@dds.nl
  --encrypt verslag.tex
```

Het bestand "verslag.tex" wordt nu versleuteld met de openbare sleutel die bij "berteun@dds.nl" hoort en het versleutelde bestand wordt in "verslag.tex.gpg" gestopt. Nu kan ik een e-mail opstellen en ik kan dit document eraan hangen en versturen. Aangekomen bij mijzelf zal ik het, als ik het wil lezen als volgt ontsleutelen:

```
$gpg --output verslag.tex
  --decrypt verslag.tex.gpg
```

Het maken van handtekeningen is iets ingewikkelder, je kunt namelijk kiezen of de handtekening in een apart bestand moet of dat het in hetzelfde bestand staat. Dit laatste wordt vaak gebruikt bij e-mail en nieuwsgroepen, dan zie je een bericht met wat regels aan het begin en eind toegevoegd.

Een los bestand maak je als volgt:

```
$gpg --output verslag.sig
  --detach-sign verslag.tex
```

Het bestand "verslag.sig" bevat nu de signature, dit is echter geen gewone ASCII tekst, wat vervelend is als je de signature op een website wil zetten, om zo'n signature te maken gebruik je:

```
$gpg --output verslag.sig
  --armor --detach-sig
  verslag.tex
```

Hier geeft "--armor" weer aan dat het gewone tekst moet zijn. Deze handtekening kun je dan samen met het originele bestand op een website zetten zodat gebruikers het kunnen controleren. Het controleren gebeurt als volgt:

```
$gpg --verify
  verslag.sig verslag.tex
```

Je kunt ook, zoals gezegd, de handtekening en het originele document in één bestand stoppen. Nu is er weer de keuze om het wel of niet als ASCII te doen. Geen ASCII gebeurt simpelg met het commando "--sign":

```
$gpg --output verslag.sig
  --sign verslag.tex
```

Om gewoon een ASCII document op te leveren dat je bijvoorbeeld als e-mailbericht kunt gebruiken doe je:

```
$gpg --output verslag.sig
  --clearsign verslag.tex
```

Ook in deze gevallen werkt de opdracht "--verify" wel, maar daarmee krijg je alleen te zien of de signature goed is en krijg je niet je oorspronkelijke bestand terug, daarvoor moet je dezelfde opdracht gebruiken als bij het decoderen van een bestand dus:

```
$ gpg --output verslag.tex
  --decrypt verslag.sig
```

Bovenstaande opties kunnen ook gecombineerd worden. Dus dat je zowel het bestand versleuteld en van een handtekening voorziet. Want anders zou een kwaadwillend persoon of nog steeds je bericht kunnen lezen (als je alleen een handtekening gebruikt) of hij of zij zou jouw bericht kunnen vervangen door een ander versleuteld bericht zonder dat dit direct hoeft op te vallen. Dit gebeurt, hopelijk weinig verrassend, met:

```
$gpg --output verslag.gpg.sig
  --sign --encrypt --recipient
  berteun@dds.nl verslag.tex
```

Integratie met (e-mail) programma's

Bovenstaande methodes werken uiteraard, maar zijn toch vrij moeizaam. Als je even snel een e-mail wilt typen dan ga je niet eerst je bericht apart opslaan, vervolgens versleutelen en van een handtekening voorzien om het daarna te versturen. Je wilt het typen en met een paar toetsaanslagen versleuteld en getekend hebben. Het handigste hierbij is dat het in je e-mailprogramma integreert.

Dit gaat over het algemeen iets soepeler onder Linux dan onder Windows. De meeste Windowsgebruikers gebruiken Outlook Express, dit programma staat al niet erg bekend om het goed omgaan met Internetstandaarden of het goed samenwerken met plugins, dus op zich is GnuPG een goede reden om over te stappen naar een fatsoenlijke e-mailer.

Het voert te ver om voor elke client te bespreken, maar voor Linux heeft Mutt uitstekende ondersteuning evenals KMail en Evolution. Voor Windows zijn er voor Eudora plugins en The Bat! ondersteunt GnuPG uit zichzelf. Ook voor Mozilla is er een plugin beschikbaar, namelijk Enigmail. Voor Outlook Express is er ook een oplossing gevonden die redelijk goed integreert, deze oplossing is echter verre van optimaal als je deze met andere programma's vergelijkt. Voor een volledige overzicht en links naar de programma's kun je het best op: "http://www.gnupg.org/frontends.html" kijken.

Sommige e-mailprogramma's kunnen ook bijvoorbeeld automatisch sleutels opvragen als je deze mist en ze kunnen gebruikt worden om een bericht dat je naar meerdere mensen stuurt zo te versturen dat iedereen het kan lezen (dit kan overigens natuurlijk ook met het standaardprogramma) en ze kunnen binnengekomen berichten automatisch decoderen en verifiëren.

Verder zijn er nieuwslezers die GnuPG ondersteunen en kun je ook in Miranda ICQ GnuPG gebruiken om je berichten te ondertekenen. IRC is wat problematischer omdat je moeilijk elk bericht kunt encrypten naar alle, misschien wel 100+, namen in je kanaal

en een handtekening van 5 regels wordt ook irritant bij elke regel tekst, ik heb nog geen scripts gevonden die GnuPG gebruiken voor IRC. Wel is er een IRC-netwerk dat gebaseerd is op public/private key encryptie (zie: "http://www.invisiblenet.net/iip/").

GnuPG en PGP zijn op het moment nogal hulpmiddelen voor de liefhebbers en het meestgangbare e-mail-programma, Outlook Express, biedt uit zichzelf nog niet veel mogelijkheden voor GnuPG ondersteuning. Dit programma heeft nog een groot nadeel, het kent niet alle standaarden waardoor het berichten die van een losse handtekening zijn voorzien weleens niet goed herkent waardoor de ontvanger alleen een e-mailtje met twee attachments ziet. Vooral bij mensen die wat minder goed kunnen omgaan met computers kan dit tot grote verwarring leiden. Dit probleem treedt vooral op als de verzender Mutt gebruikt, maar het probleem ligt duidelijk bij Outlook Express.

Verder is het handig om een e-mail die je verstuurt altijd ook met je eigen sleutel te versleutelen, want anders kun je later je eigen e-ail niet mer lezen omdat deze alleen met de sleutel van de ontvanger is versleuteld.

Tot slot kan mijn eigen sleutel gevonden worden op "http://www.berteun.dds.nl/pubkey.txt". De fingerprint van deze sleutel moet zijn: "FAD2 4B03 55E4 41C3 DB6F 7D0F E4D2 CC59 8789 AC3E". Echter, het kan zijn dat de redactie natuurlijk het adres en fingerprint heeft veranderd waardoor het niet veilig is. Of dat de voorgaande zin er niet staat. Of dat het hele artikel anders is. Dus vertrouw deze sleutel niet zo maar.

Bronnen / verder lezen:
[I] http://www.gnupg.org/gph/en/manual.html GnuPG Privacy Handbook
[II] http://www.rsasecurity.com/rsalabs/faq/ RSA Cryptography FAQ
[III] http://www.pgp.com/ Homepage van de PGP Corperation