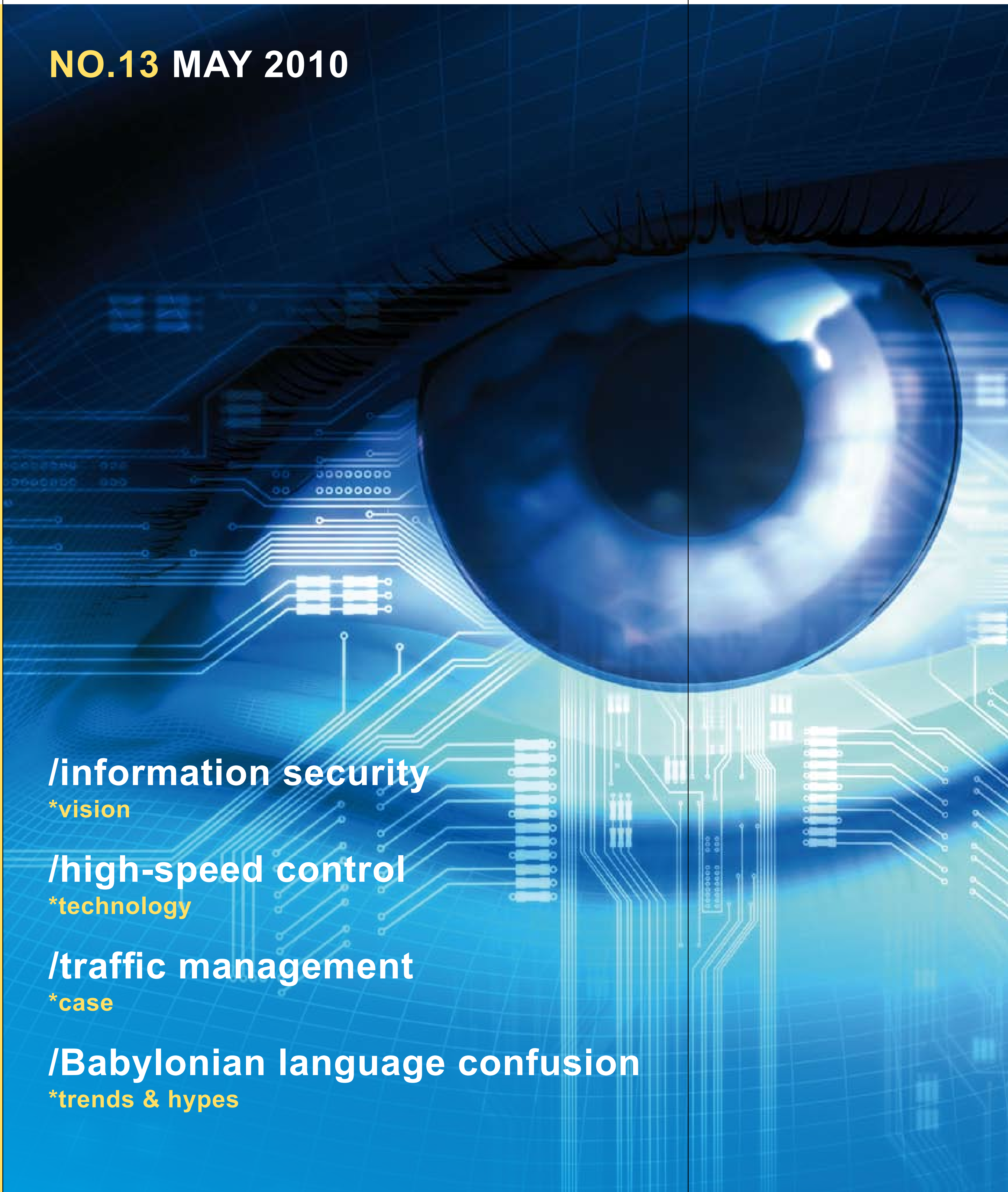
[OBJECTIVE

on innovation and technology

Technolution

NO.13 MAY 2010

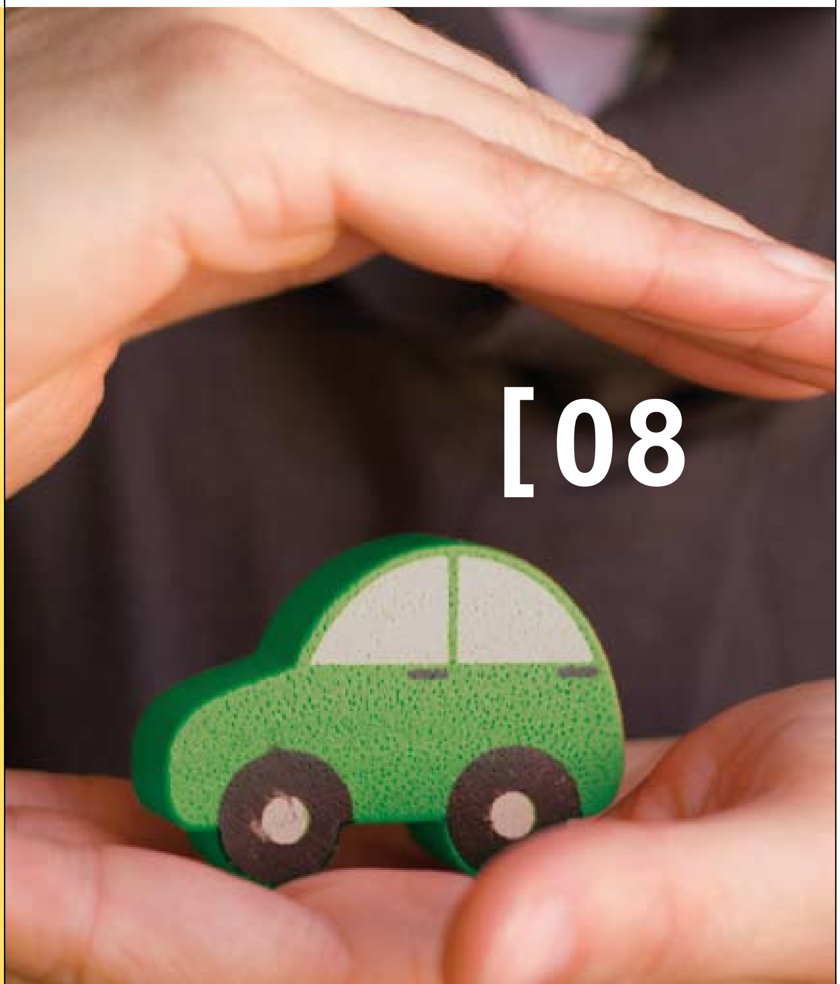


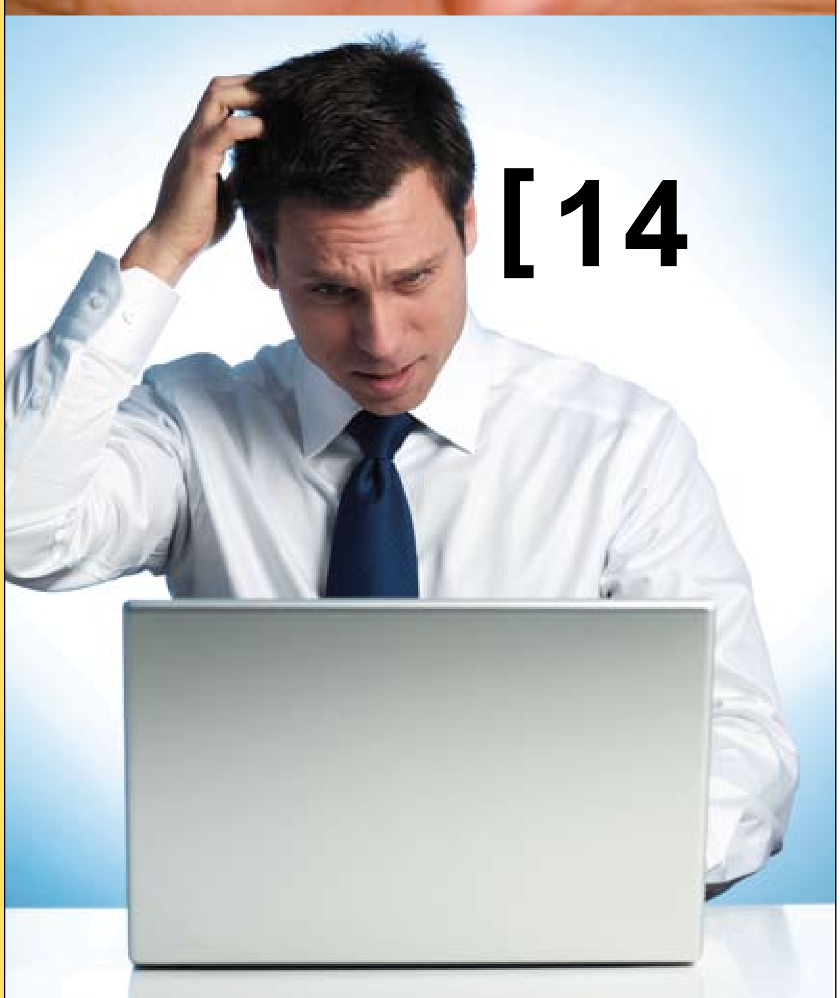
/information security
*vision

/high-speed control
*technology

/traffic management
*case

/Babylonian language confusion
*trends & hypes

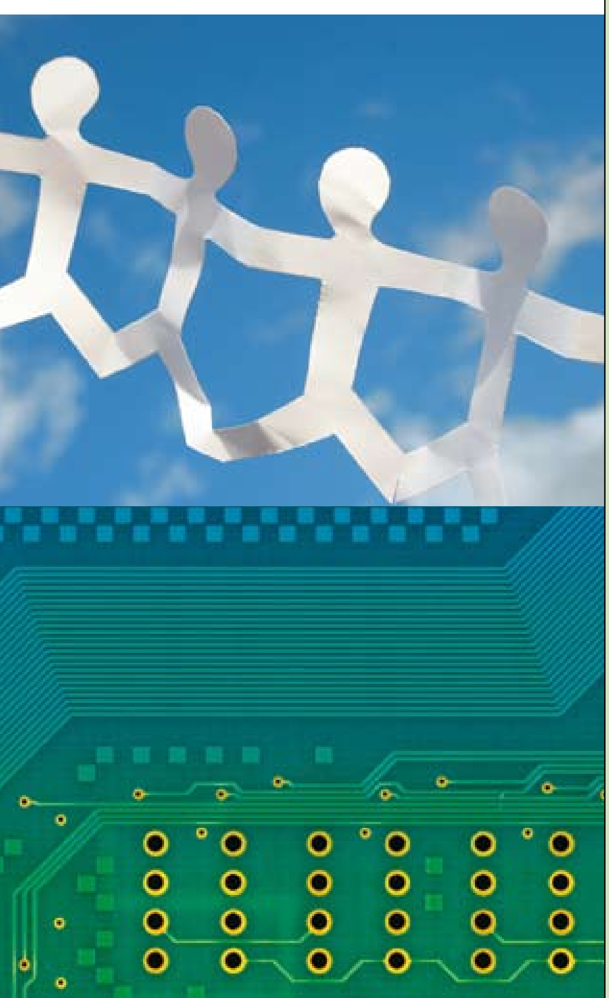




/contents

Jan van der Wel

CEO



/foreword

Computers are getting faster and smaller all the time and memory is constantly increasing. Combined with all the communication options available, this is creating ever more opportunities to build cooperative, intelligent systems. New interactive services are springing up. Measurement and control systems no longer just control the temperature in ovens and the like; they govern complete machines or even complete ecosystems such as road traffic. These rapid developments are impacting on society and us as a company.

>>everything is connected with everything else

As new trends emerge, so new names are coined to describe them. Take ‘cooperative traffic systems’, in which vehicles act as sensors and have ‘intelligence’ to help improve traffic flow and enhance road safety. Other examples are ‘collaborative workspace’ for cooperation between hospitals performing operations and ‘coordinated control’ of machines.

The great quantity and diversity of data generated makes combining and interpreting it all a challenge. Directing and taking decisions is also becoming ever more complicated. New approaches are needed if we are to keep an overview. Forms of artificial intelligence offer ways to facilitate this.

How do we overcome obstacles such as the need to secure our personal data? Objections come thick and fast whenever the impression is given that privacy might be compromised by these types of cooperative systems which can tell where somebody is to the nearest metre. Vociferous objections were raised against the Kilometer road pricing scheme, even though the technology underlying the concept had been well thought through. These kinds of responses can be avoided with clear and consistent communication towards users.

Coping with the technical challenges to achieve successful implementation of such systems demands knowledge of the application, the overall system, integration and architecture, along with the desire to get to grips with the detail. In this issue of Objective, we discuss a number of aspects of this vision. Among other things, we look at information security, complex decision-making in traffic management and the control of high-speed mechatronic applications.

Happy reading!

Information security

>>i've got nothing to hide, but everything to protect

Information systems play a crucial role in our modern society. Naturally, those systems need to be properly secured. But even more important is the question of what data we store in them and why. Privacy is best protected by having as little information as possible.

Information security as the finishing touch

More and more companies and public bodies use information systems for their operational management. From supermarkets to hospitals and from banks to the tax department, data about us are stored everywhere; purchasing behaviour, health problems, debts and income.

This is sensitive information that we would like to see secured properly. However, in the world of information systems there is a strange paradox. Although security is an inseparable part of any information system, it is often not incorporated into the design in an integrated manner. Security is only considered at the end. "The house is finished. Now all we need is a lock on the door." This separation between architecture and security has grown up over time. When securing data, we focus on the particular information architecture. But nobody asks why the architecture is the way it is.

Privacy is not having information

Yet for the sake of privacy, it is important to pose the 'why question' at the very start of the project. The core question is: ought we really to be storing that data there? In the ideal case, we would have as little information as possible in order to safeguard privacy. Often, these kinds of questions are not considered in the design of a system. They are left to the implementation side. This results in partial technical solutions with protocols, passwords and physical security measures – all of them retrospective – rather than thinking properly about the concept beforehand.

Anonymous road pricing

Unfortunately we are still seeing more failures than successes, but luckily there are also some good examples where the security and privacy aspects have been considered from the very earliest stages. An example is the Kilometer road pricing scheme, which attracted a great deal of attention, albeit much of it negative. In terms of security and privacy, that was undeserved. The essence of road pricing is recording how many miles a car has driven on a particular type of road.

In-car equipment calculates the total amount owed from the GPS coordinates and the various rates that apply. The computations are carried out in the car – information such as time and location do not leave the vehicle. All that goes to the authorities are a few anonymous details. The same principle applies in other systems too: privacy risks are most easily and best solved on the user side. In this way, the system builder ensures that sensitive information does not become available in the first place.

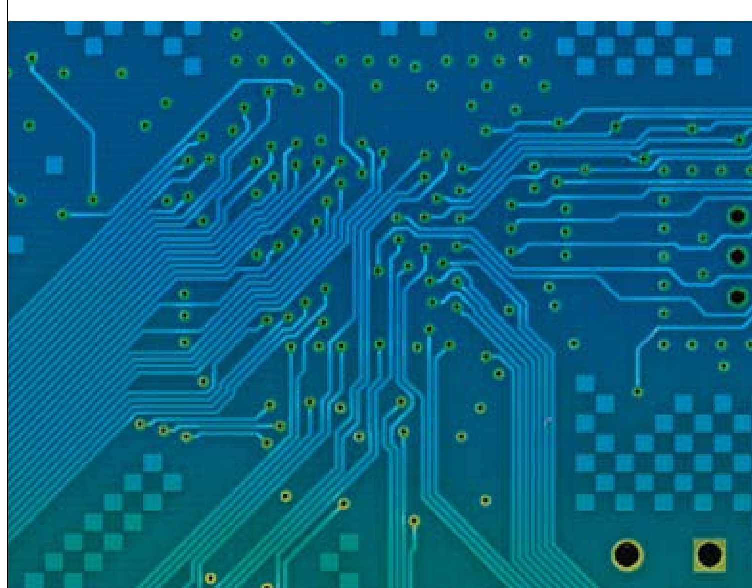
Function creep

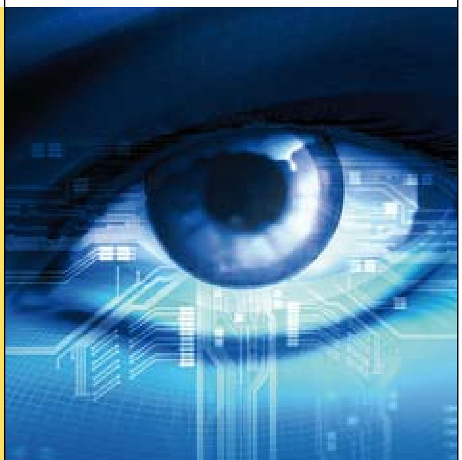
From a technical perspective, it can be attractive to gather as much information as possible and then combine it with as many functions as possible. Because you never know what you might need in the future. Minimalism is often lost on technical types. Their enthusiasm for technology instead makes them want more options and functions. Unfortunately, that also has risks associated with it, as we are now seeing with telephone and internet exchanges.

In the past, designers of new digital telecom exchanges built in storage functionality. This was useful for analysing the quality of the systems and networks. Data was kept for six weeks and then removed. Storage was not anonymised, because it was only meant for in-house technical analysis. However, the authorities thought the information stored would be useful for detection purposes: “We can use it to locate and track down criminals.” This is how function creep happens: built-in functions are later used or abused for other ends.

>>which information to store?

The Netherlands now top the world rankings for phone tapping. Moreover, European legislation requires telecom providers to store all data for all users for 6 months. It is fair to ask whether this legal provision would have been introduced had the technical possibilities not been available. But thanks to the extensive functionality of modern telephone exchanges, this infringement of our privacy is now possible.





What is information security?

In short, if the information is there, there is a strong temptation to use it. So a sensible designer ensures that a system requires as little information as possible. And of course that information needs to be properly secured in terms of availability, integrity and confidentiality. It is all about the total system that the information has to pass through: both the storage and transport of data.

Availability

The road pricing scheme would have involved every car making a permanent record of its route. That meant that the GPS signal had to be available everywhere and at all times. The in-car boxes would also have had to be operational at all times. Communication with the back office was less critical. If the connection is lost, the box stores its data so that it can try it again later. Solving these issues around availability requires keeping a good overview of the situation. How do all the subsystems work together? What happens if one system fails? How does the system restore itself? That is system engineering.

Integrity

Integrity is about the accuracy of data. Are the figures I am reading out correct, are they up-to-date, complete, not manipulated and from the stated source? Important questions for any information system, and certainly for systems that provide support for important decisions. Aircraft and ships need to be able to have complete faith in measurement data. Saving measurements in a “data log” allows us to demonstrate manipulation if there has been any and make a judgement on integrity. This allows the user to see – and prove – what has happened. In this case, data storage serves to provide proof, but it also enables us to analyse what went wrong if there are problems. An aircraft black box, yields indispensable information. In its operation, this type of system resembles financial systems. A bank wants to be able to trace every transaction from payee to recipient. Banking systems are by definition not built with a view to privacy. On the contrary, their security works by following and analysing payments. What is a customer paying for with his or her credit card, how much are they paying and what kinds of shops or companies are they paying the money to? If there is anything unusual, the bank spots it at once. This is how banks combat fraud. They are unlikely to abuse their data – a bank depends on its customers’ trust.

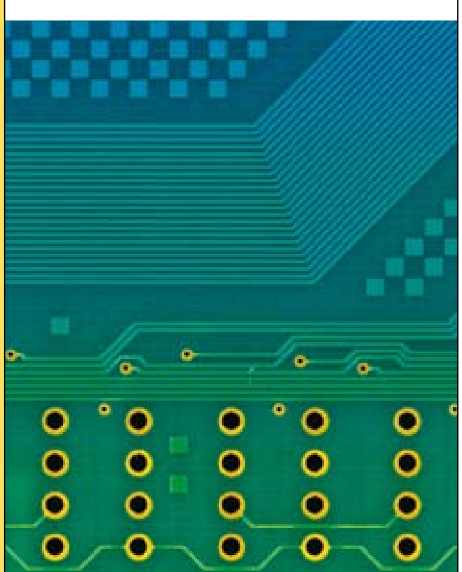
>>open and complete communication with user

Confidentiality

Confidentiality means only making information accessible to authorised persons and systems. So above all, confidentiality is about organisation. Who has access to sensitive information and how do we prevent unauthorised individuals from gaining access to it?

Security technology

A wide range of cryptographic techniques exists to protect the integrity and confidentiality of signals or communication. These techniques encrypt the information, so that no one can access it en route. The recipient has a key to unlock the information. He can also see if anyone has been tampering with the data on the way. Digital keys can come in the form of a Public Key Infrastructure (PKI) certificate, this is a worldwide technology that provides people and machines with digital passports (PKI certificates) and their own keys. PKI certificates are issued by specialist firms like VeriSign. They are the trusted third party in this process of encryption, a kind of ‘key master’ of the internet. It should be stressed that a digital passport does not offer absolute security; just like an ordinary passport, it can be forged.



Residual risks
However good your security is, there will always be residual risks. There is no such thing as a system that is 100% safe. So the residual risks must be dealt with at the organisational level as soon as they are encountered. For example, when residual risks occur, who is responsible and is the organisation capable of deciding on action and communication? This means that alongside technology, plans also need to be made in terms of prepared scenarios and processes to deploy additional technology

Communication
The hacking of public transport chip cards has received a lot of attention in the Dutch and English media. Perhaps this is because journalists do not know exactly what is going on. However, it does mean that the general public is getting a distorted picture of new technology. This problem also affected the Netherlands road pricing scheme. In technical terms, the road pricing scheme was well constructed, but there were problems in communication with the public. Citizens were being confronted with new and unfamiliar, and above all compulsory, technology. Unfamiliarity breeds suspicion and we find technology scary. Particularly if we think our privacy is at stake.

On the other hand, we constantly phone and surf the web without giving a second thought to privacy. But as soon as the government makes something compulsory, we get edgy. Privacy and information security are sensitive issues, which means that communication by government and companies needs to be accurate and persuasive precisely in those areas. And the timing has to be right: choose your moment to communicate, deliver a consistent story and have clear answers ready to all questions. And be honest about the residual risks. “In view of the state of the technology and the organisation, we can now do this – but no more than this.”

Trust and reputation
You can destroy a good project with poor communication. Inadequate communication damaged confidence in the road pricing scheme. The communication that there was came too late and was reactive, and it was in response to negative questions. If you lose the public’s trust, you will not win it back with a brilliant technical solution. Trust is a product of communication, not just with citizens but also with companies and among companies, in other words with all stakeholders.

/customer interview

Movenience: mobility as a service

Most drivers who use the Westerscheldetunnel automatically pay the toll by means of an in-car box, the t-tag. They can now also use the t-tag to pay at a number of car parks in Zeeland. If Movenience has any say in the matter, this will just be the start.

Movenience was set up in 2007 by the Dutch NV Westerscheldetunnel, infrastructure consultancy NedMobiel and the Portuguese toll company Brisa. The core activity of this young firm is handling car-related payments. “Our ambitions go further than just collecting tolls”, explains Dirk Grevink, business development director of Movenience. “As well as tolls, you could use the t-tag to pay for parking, fuel and even a drive-in McDonald’s meal. You have a box in your car; we make sure you can use it to pay for all kinds of things from a single account.” In itself, electronic toll payment is nothing new; toll roads abroad have been using it for years. “But there is a big difference”, emphasises Grevink. “For most toll operators, the in-car device is purely an instrument to collect their tolls. We want to use it to supply services to drivers.”

European toll system

The company is looking forward to the impending European market for tolls and road pricing. This is currently fragmented, with many countries having several toll operators. These are the companies that own the roads. They build a road, get funding for it and generally earn their money from tolls. There are different systems in place for collecting tolls electronically that do not always work together. Partly for this reason, Europe has decided to separate the two: electronic toll payments will be independent of road ownership. Interoperability requirements will mean that drivers can use a single box to pay throughout Europe. The system is meant to be in place for freight traffic in September 2012 and for all traffic two years later. This will create a market for suppliers of European Electronic Toll Services (EETS providers).These companies will distribute boxes (On-Board-Units or OBUs) that meet the EU requirements and enter into contracts with customers (consumers, drivers). “These service providers will also start collecting any road charges for the government. Movenience wants to do the same. You could compare it to telephone service providers. They give you a phone that will also work on different networks throughout Europe.”

Partnership

“For us, the relationship with our customers, drivers, is key. That means custom solutions; we really want to offer our own product. If we buy an existing system, we are tied to the limiting conditions set by the supplier. However, we are too small to develop a system ourselves, so we went looking for a partner. Someone who understands road pricing, who is aware of what is happening in the European market and who knows about interoperability. Technolution has that knowledge and that is very important to us. There are many parties who claim to understand this area, but if you dig a little deeper, you quickly discover the limits to their knowledge.”

TIPS

For the Westerscheldetunnel, Technolution developed the equipment that detects and classifies vehicles at the toll booths. Movenience subsequently asked Technolution to develop the Transaction and Information Processing System (TIPS). TIPS handles the electronic payments made using the t-tag. The great thing about this system is its simplicity, says Grevink: “Under the lid, TIPS is complex, but it is very well put together and the essence is simple. It supports our contract with the customer (driver); we have the t-tag which the customer uses to pay and we can make a link with companies (operators) that can supply services to the customer. It is transparent, straightforward and user-friendly for our helpdesk. We are very satisfied.”

Photo above:
Dirk Grevink
Business Development
Director Movenience

At the kitchen table

In the run-up to the European market for tolls and road pricing, Movenience is participating in accessibility projects. An example is SpitsScoren, which tempts people to drive outside the rush hour by paying them five euros each time they do. “We can use the project to test out new technologies and principles. We will be looking at whether it works in technical terms, but also whether we can influence people’s behaviour. In order to achieve that, the technology needs to be simple and close to the user. We want to be sat at the kitchen table with our customers. That is where people make their decisions about how to travel, not when they are in their cars: by then, the choice has already been made. That is why SpitsScoren uses a smartphone that you can use at any time, wherever you are. The customer uses the device to indicate their travel plans for the day. The built-in GPS checks whether they actually do what they say they are going to do. That smartphone takes us very close to the customer.””

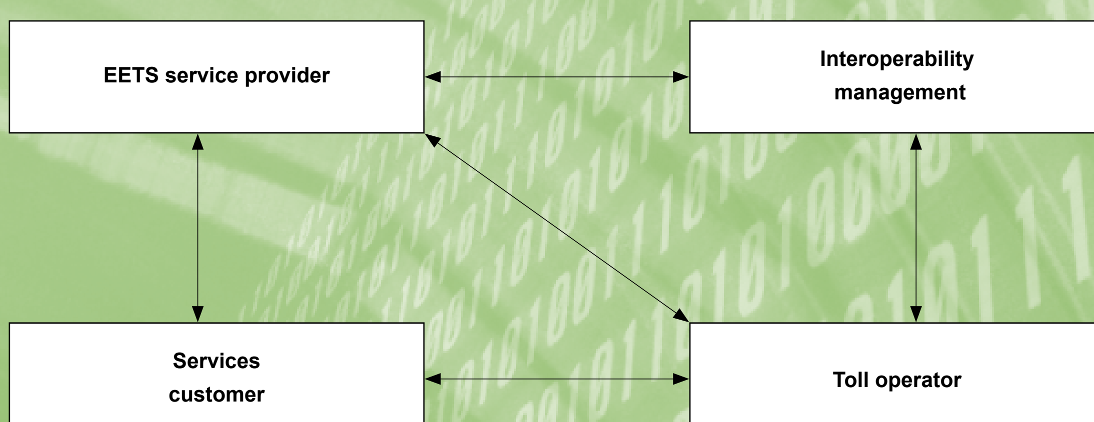
Smartphone as platform

“The smartphone is a great platform for new applications”, explains Grevink. “It means you are not dependent on the launch of new equipment for new services. Downloading an application on the smartphone is enough – to pay for car parking, for example. In the background, we make a link between the parking payments system and our back office. That is another important aspect of TIPS: scalability. We can keep on making new interfaces which suppliers can link different systems to. That flexibility is quite unique in this field.”

Mobility as a service

It looks as though road pricing has been shelved again for political reasons. That is a shame, believes Grevink: “Mobility shouldn’t be an issue for the individual any more. As it stands, we spend a lot of time planning and making arrangements. Really, the whole mobility issue should be taken out of our hands. That way, road pricing becomes not a charge but a service. You want to make a journey by train or car and you pay by the mile to use that service. With the advent of user-friendly in-car payment systems, such as the t-tag, public acceptance of road pricing will increase. Projects such as SpitsScoren and SpitsMijden will also contribute to acceptance. I can imagine a scenario in which those projects get extended further and further and that road pricing is introduced on the back of them, perhaps as a regional charge initially. Road haulage could also be an important driver. Trucks already pay lots of small tolls across Europe. For them it makes things simpler if they can pay everything with one box.”

.



/technology

High-speed control

>>electronics with software properties

Control engineers like to use software to make their controllers. Such controllers are flexible and often cheap, but not always fast enough. Electronics are many times faster and more accurate but inflexible. Technolution has designed a high-speed electronic controller with the flexibility of software.

When we talk about speed in *Objective*, we are usually talking about electronics. But mechanical systems are becoming ever faster too. For example, pick & place machines that can place components on a PCB at lightning speed, or wafer-steppers that can move the wafer into the right position under the lens in a fraction of a second. In an electron microscope, speed is less important, but the movements need to be made with extreme precision and as little vibration as possible. All of these are mechanical systems that demand fast and accurate control. A software controller cannot achieve the required speeds: the processing power required would result in a very expensive controller. An electronic controller with programmable logic is fast enough and cost-effective but not flexible. Once the design is fixed in the FPGA, there is nothing left to adjust. Or is there? Technolution has devised an electronics-based method with the flexibility of software. All the settings you want to be able to change in software are now also available in electronics. Moreover, the control frequencies are much higher than in software controllers and they are 100% predictable.

From transfer to controller
Control technology is the field of the mechatronic engineer. The first stage in his work is to describe the process that needs to be controlled. For example, a living room with a heater and a thermostat. If the thermostat goes up by a degree, how long does it take before the whole room is a degree warmer? It takes a while. Everything in the room has to heat up, and there is leakage from poor insulation, windows and doors. All of which results in the transfer: intervention versus result. For a simple process, this can be a mathematical formula. However, practical situations tend to be more complex and difficult to describe in a single formula. So control technology offers general formulas (PID) to describe the process. You test things out by introducing a reference signal at the input and then measuring how the system or process behaves at as many points as possible. For example, using a hammer to strike the axle of an engine and a sensor to pick up the resulting vibrations. This is how you determine the transfer of the system experimentally, which can result in a formula with a number of parameters. Using the experimental data, the mechatronic engineer can design a controller to govern the system. In his design, he wants to be able to measure what is happening and adjust it at as many points as possible, because in complex systems and machines, the controller settings will never be exactly right straight away. So you want to be able to adjust those settings.

Fast and predictable
A software package like Simulink offers that flexibility. The user can join all kinds of functions together as blocks. As long as the controller remains on the software side, there is enormous flexibility to measure and adjust all kinds of things. However, the great limitation of software is its speed and predictability. Software often suffers from jitter. The timing of software is not accurate enough for fast control processes. At low frequencies, that is not really a problem, but at around 10 kHz, the jitter in software is a serious obstacle. Electronics, by contrast, are predictable and accurate to the nearest clock pulse. In theory, electronics are also affected by jitter, but it is many orders lower than in software and therefore negligible. The clock frequency can easily be a factor of a thousand higher than the control frequency: control at 300 kHz, clock at 200 MHz. The accuracy of the overall control system is more likely to be limited by the sensors.

So if you want to design a fast controller, you will need to switch over to electronics (programmable logic in field-programmable gate array FPGAs). Simulink offers the option of downloading the software controller to an FPGA. The programme then converts the software into one large chunk of Visual Hardware Descriptive Language (VHDL) code which is synthesised and downloaded to the FPGA. Generating a new controller for the FPGA can take an hour. And once on the chip, the result is fixed: an electronics controller which cannot be altered and in which there is nothing left to measure between the separate controller blocks. A controller without adjustments and measurement points, in other words. Changes are only possible in Simulink, after which new

VHDL code needs to be generated to overwrite the FPGA. This results in time-consuming repetition.

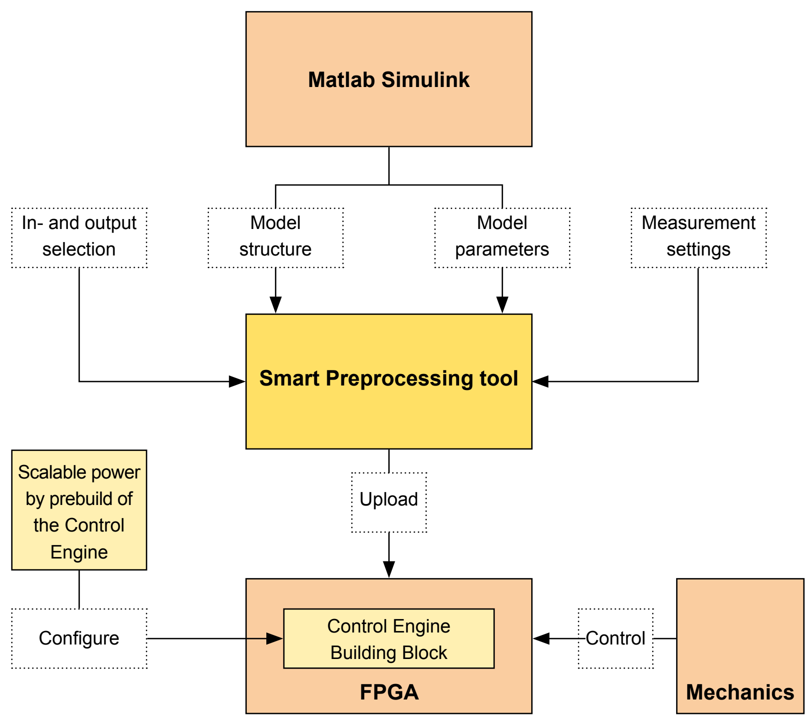
Modified library

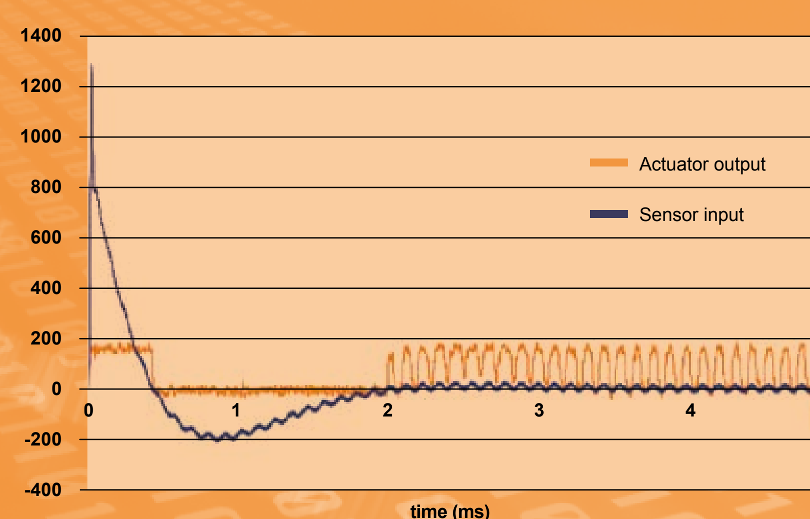
In order to retain the flexibility of a software controller, Technolution has developed its own library of building blocks for Simulink. These blocks are already present, part-configured, on the FPGA but have been modified in such a way that they can be used in Simulink. Each has specific functionality and can be joined together on the FPGA in the appropriate way. Now the controller can also be transferred to an FPGA using Simulink. Not as one big chunk of VHDL code but simply with some configuration data. These configuration data are determined using a tool developed by Technolution. The tool reads and interprets the Simulink files and generates the configuration data for the controller. You could compare it to a navigation system: instead of downloading a complete high resolution map to the FPGA, you merely download a route onto a map that is already there.

FPGA controller as prototype and end product

The controller thus created in the FPGA has the same measurement and control points as in Simulink. It is an electronic controller which the user can use for real-time measurement and control at all the same points as in the simulation – which is very useful for the mechatronic engineer. This system offers him a handy additional tool to define his processes with. He devises a controller and tests theoretically using Matlab and Simulink. When he is satisfied with the simulation, he transfers it to electronics in a matter of seconds. He can then link these electronics to the real process in order to see if they really do offer the control he is looking for. He can now make any adjustments to fine-tune the controller ‘on-the-fly’ and sees the results immediately. So he does not have to go back to the computer to change the parameters in Simulink and wait another hour for a new configuration of the FPGA to be generated.

This gives the user a handy tool which can achieve operating frequencies that are not possible in software, both for R&D and production. Even when the controller is mass produced, it retains its flexibility, and at a much lower price than comparable software controllers.





/case

Traffic management:

from motorway to city centre

The Dutch road network relies on traffic management. Increasing urbanisation and economic activity in the Netherlands mean that the pressure on an already heavily used road network continues to grow, which means there is no alternative to traffic demand management. The road network has grown in size and the number of traffic management measures has increased. The work of monitoring and directing all these measures is concentrated in traffic centres. Rijkswaterstaat (the Directorate-General for Public Works and Water Management) has five regional traffic management centres to cover its trunk road network and the major cities, and increasingly the provinces, have their own centres too.

Because traffic management measures on the roads continue to proliferate, coordinating them is becoming ever more important. The aim is to find synergies between them and promote traffic flow and traffic safety across the entire road network, instead of at one point or on one route. This is called network-wide traffic management. It means adding a management concept to the individual systems – a management concept that is able to transform policy aims with respect to traffic flow, traffic safety and environment into targeted coordination of the individual traffic management systems.

Network-wide traffic management
Coordinating systems requires an administrative and technical change. In administrative terms, the parties involved must want to and be able to work together in order to open the way for the coordination of systems. The systems will then need to be linked together, in order to actually coordinate them. This is a step-by-step process, starting with cooperation in defined fields such as parking, traffic regulation, motorways and incident management.

For municipalities, this means working together with those in charge of car parking and lifting bridges. Once this has been arranged, a parking guidance system can be set up that has up-to-date information on the availability of parking spaces, whether bridges are up or down and traffic volumes on the various available routes. In this way, the system can tell the user where the available parking spaces are and what the best route to take is.

A logical next step is to coordinate traffic management systems along each route. To support dynamic routing, the timing of traffic lights can be adjusted; semi-dynamic green waves can even be set up.

In a broader context, guaranteeing good traffic flow and safety at roundabouts and crossroads is very important to many municipalities and other road managers. Regionally, a good start on administrative cooperation between road managers has been made with the ‘Gebiedsgericht Benutten’ (‘Area-focused Use’) approach. This involves road managers meshing their policy principles together and together translating them into a management strategy for the regional road network. The management strategy indicates for which part of the road network the current traffic situation needs to be adjusted, after which the relevant roadside systems are deployed in a coordinated manner with the help of a management scenario.

These are all first steps towards network-wide traffic management. Based on where we are now and what we have learned so far, work is now ongoing on a management concept that transcends and connects the different fields. Together with Rijkswaterstaat (the Directorate-General for Public Works and Water Management), Amsterdam municipality, TU Delft and traffic science advisers, Technolution has designed a management concept that can manage traffic autonomously over a greater part of the road network under normal conditions.

Road traffic manager indispensable

The road traffic manager is the indispensable link in the coordination of systems. He needs to choose the right regulation strategy at the strategic level, apply (and choose) the right regulation scenarios at the tactical level and take measures at the operational level. Examples are safely opening rush-hour lanes, closing lanes for road works, putting up or removing red crosses in case of incidents, or operating bridges.

Future

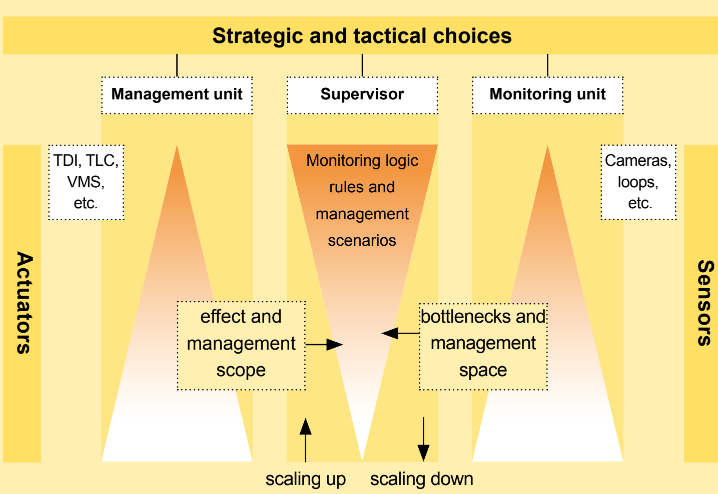
Technology is not standing still either. We are familiar with the digital traffic reports (RDS/TMC) that can be picked up by the car radio or navigation system. With the advent of mobile communication systems, there is a growing desire to be able to send messages direct to vehicles as the logical successor to signs beside and above the road. Conversely, as vehicles drive around they can ‘tell’ us what the traffic situation is. This leads to so-called cooperative systems, in which vehicles and infrastructure communicate with each other (and eventually also vehicles with vehicles).

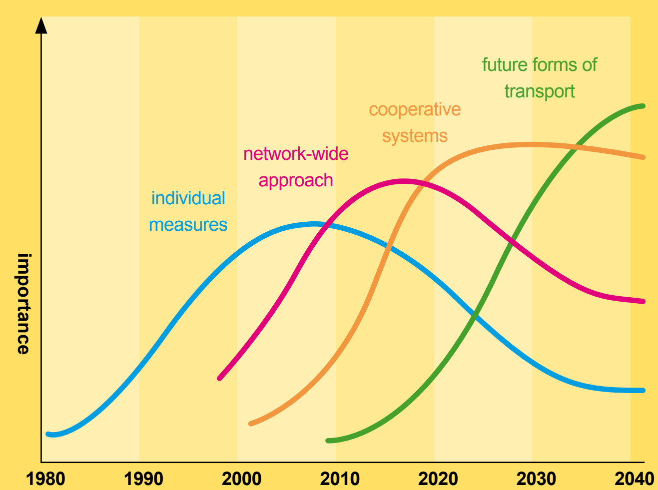
The first step towards cooperative systems has consisted mainly of exploring the technologies that will be tasked with projecting traffic management onto individual vehicles. Technolution was a partner in the European R&D project ‘cooperative vehicle infrastructure systems’ (CVIS). Now the R&D results need to be turned into products: Connected Cruise Control (CCC) is an example. Within CCC, a vehicle selects a target speed based on its own observations and an electronic horizon and it receives a target speed from the traffic centre. The vehicle converts these target speeds into a recommended speed for the driver.

MobiMaestro

MobiMaestro is the solution for network-wide and integrated traffic management. It consists of applications and products that have been developed according to a single philosophy and that all support part of the traffic management domain. A single philosophy means that there is uniformity between the associated software modules, so that together they form a single integrated traffic management system. This offers great benefits, both in process terms because operation and working methods are the same for users, and in technical terms due to the interchange-ability of modules, interfaces and data.

Examples of MobiMaestro applications are dynamic parking and route guidance in cities like Rotterdam, Utrecht, Breda and Houten, but also the TLC management expansion and linkage to in-car systems. In addition, there is a process manager which road traffic managers can use to design their own workplaces and Technolution’s BOSS-Online system supports road traffic managers in making decisions.





Babylonian language confusion

The abacus is a counting frame known to have been in use in Egypt 2500 years ago. This arithmetical aid may even date back to 3000 BC. The trail leads to Babylon – the original source of the computer, but also of language confusion.

New computer languages regularly become available. The question for every company is what to do with them. Keeping abreast of everything is impossible; there is neither the time nor the money. It would not be sensible either, because many new languages will not survive for long. Evaluating and comparing languages requires understanding and knowledge of application, software and vocabulary. Strong and weak typing are not the same as static and dynamic typing. But what exactly are they? And what do they tell us about a language, about the productivity of programmers and about the flexibility, quality and shelf life of the end product? This article aims to answer those questions, in the hope of making the task of choosing the right language easier. Because renewal is essential in order to be able to compete in the eternal race towards ever bigger and more complex systems. Consolidation/standardisation simply is not an option.

Static and dynamic typing

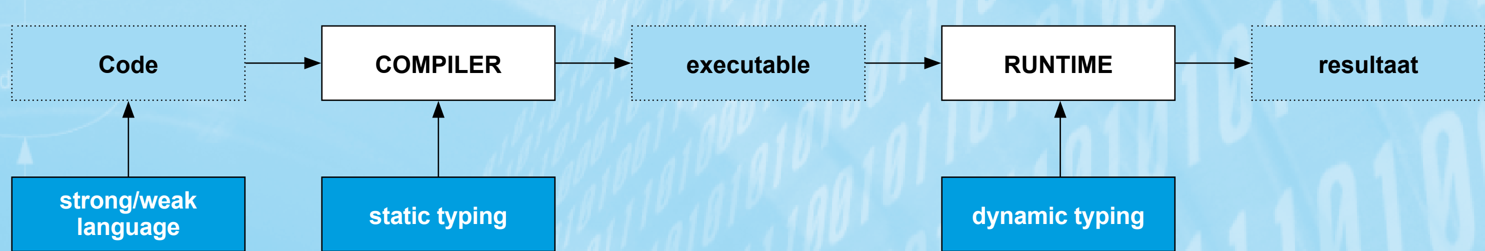
EA language is a bridge between the human being and the computer. The human being uses it to tell the computer what to do. The basis of every language is the same: the user writes a programme as a chunk of text (source code). That is done by the compiler, who translates the text into computer language (the executable). In some languages, it is primarily the compiler who checks for errors in the code. This is called static typing. In other languages, the compiler does less checking, but we come across the errors primarily when the code is run. That is a runtime error. A language that largely leaves error checking to the runtime is a dynamically typed language.

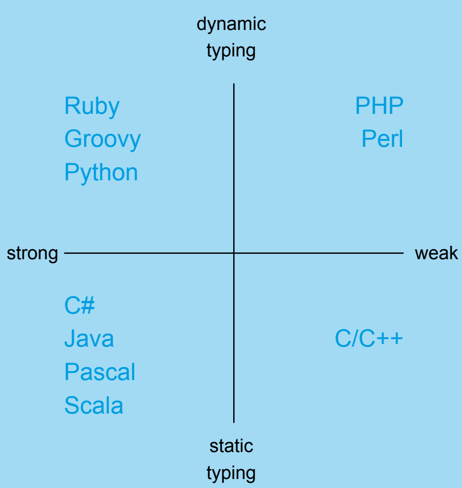
What is a type?

The term ‘type’ refers to the kind of data a programme processes. A programmer cannot simply apply any operation to any kind of data. How strictly a language deals with types is indicated by strong or weak typing. So static and dynamic typing is separate from the terms strong and weak typing. That sometimes causes language confusion of Babylonian proportions: you can have strong and weak dynamic typing, and strong and weak static typing. An analogy may be useful to clarify the differences, lest we end up in our own Babylonian language confusion.

Apples and pears

Let us imagine a computer programme as a fruit stall. In a weak language, the fruit can all be mixed up together. In a strong language, all the apples must be in the apple crate and all the pears in the pear crate. In this example, a crate is the equivalent of an array, which in the weak case you can fill with anything and in the strong case only with real numbers, say. In a weak language, problems can arise when all the fruit in a mixed fruit crate has to undergo the same operation. That cannot happen in a strong language like Java. This first requires all the types of fruit to be defined, before assigning the operation ‘peeling’ only to bananas and ‘cutting’ only to apples and pears. In a weak language (such as C), the programmer can easily mix all the different types together, but this in turn can have all kinds of unintended effects and eventually result in a programme crash.





Language	Web framework
C#	ASP.NET
Java	JavaEE, Tomcat, Struts2, Tapestry, Wicket
Python	Django
Ruby	Rails
Groovy	Grails
Scala	Lift
Perl	Catalyst
PHP	Zend Framework

Hello world

One way of getting a first impression of how user-friendly a language is to see how much code is needed to put the text “hello world” on the screen. In one language, one line is enough, whereas other languages require pretty much a full page of code. It is an indication of the ease of entry. Can I go straight to work, or do I have to learn all manner of things first? Dynamically typed languages have a low entry threshold and are much more flexible in use. Statically typed languages offer a better guarantee of error-free software because the compiler has already tracked down most of the errors. But they are also complex and laborious. The programmer first has to type everything (“this is a crate of apples, those are bananas”), and that takes a lot of work. It demands precise knowledge of the grammar, which raises the entry threshold. So the choice is all about weighing up confidence, flexibility and productivity. A predominantly dynamically typed language allows you to build applications quickly, but the results are more sensitive to errors, so you will need to do much more testing of the end product. So much of the work shifts from coding to testing.

Tools

Every professional needs good tools. That starts with a good working environment. What the statute book and jurisprudence are to the lawyer, libraries and frameworks are to the programmer. In order to write code, the developer uses an advanced editor similar to Word. It automatically completes text, checks syntax as you type and compiles directly. This type of editor is known as an Integrated Development Environment (IDE). It contains help functions, support and built-in debuggers. In short, an indispensable tool.

Library and framework

A language usually also has a standard library, in which frequently-used functions are already completely worked out. The programmer can easily add these functions to his work using the Application Programmers Interface (API). A standard library is a part of the language. Alongside this, there are frameworks. These are libraries developed for a special purpose. Again, they contain complete bits of functionality to add a particular concept at a stroke. Some languages have enormous frameworks, such as Java EE (Java Enterprise Edition).

The skill of a programmer lies above all in his familiarity with that mountain of functionality and knowing where to find what. A good framework, good documentation and good support from an IDE are highly conducive to productivity.

Popularity and frontrunner drag

The choice of a language determines which frameworks and which development environments you have at your disposal. Conversely, a good IDE and a large framework can make a language popular. But the popularity of a language or an enormous framework can also act as a drag. The result can be that the language barely gets improved or a framework has a very steep ‘learning curve’ and ultimately acts as a drag on productivity. New languages take advantage of that. They can start from scratch and throw all the ballast overboard. Often a new language is more or less a copy of an existing, successful language. A mix of the good things from other languages, just a bit slicker. On the other hand, that does mean that experience and history are lost, including a framework.

Standardise or renew?

The driving force behind the innovation and evolution of languages is the drive for higher productivity and quality. Sometimes that can be achieved with a good and extensive framework. Other times, it is better achieved with a new language. Pushing a language to make it a standard does not work, or at best it works only temporarily. Languages and frameworks age. We call this ‘legacy’. They cannot keep up with constant innovation in hardware, software and operating systems. The maintenance costs of legacy systems will always tend to increase. It is an eternal onward progression in which no one can permit themselves to stand still. When it comes to games computers, it is accepted that a game for the PlayStation 1 will not work on a PlayStation 3. Similarly, a company that is entirely dependent on its IT systems for its core business will now and then be forced to make a drastic decision and take a big step forwards.

/employee interview

Winifred Roggekamp

*brainstorming with the customer

As a project manager, Winifred Roggekamp sets the boundaries of a project. For example in a workshop, together with the future users. His technical background remains indispensable for his role as project manager. "You need to know what you are talking about."

"I started out as an embedded software programmer, but I pretty soon moved up to project leader. That involves managing a project team, handling daily affairs and contributing ideas on content. I recently became a project manager, which means I have final responsibility for the project. I now deal with overarching matters, finances and contacts with the customer. Together with the project leaders, I make sure that projects run smoothly. Beside management skills, you also need technical know-how in order to do the job well. Customers appreciate it if you can talk to them at the relevant level and also have answers to technical questions.

Dredgers

I am currently working on a system for monitoring dredgers. Rijkswaterstaat (the Directorate-General for Public Works and Water Management) has a monitoring system that records where the ships have pumped up silt, how much and where it has been dumped. But this system is old. We are currently designing a new system that is faster and more flexible. A specific issue for dredgers is how you determine the volume of slurry. Slurry is soil plus water, but you are interested in the quantity of soil that comes on board. Shipping is new territory for us. In order to get a picture of that new world, we organised a number of workshops. We made prototypes and models and presented them to the users. Together we spent a whole day talking about our ideas. New questions, tips and ideas emerged and by the end of the day, we had grown much closer together. Workshops like these are very important to help you get up to speed. They help you to learn a lot about the customer's domain. And you involve the customer in the project team's thought processes.

Collaboration

I helped to write the quotation for the dredgers. We always do that in a quotation team, certainly on larger projects. We look at the technology together. And in terms of finances, we always make an evaluation with at least two people: do the figures add up and if not, why not? From the moment we got the project, I took responsibility for the project management. For me, the key parts of a project are the beginning and the end. I initiated the project; put together the team and I maintain contacts with the customer. This is a big project, so I work together with the project leader a lot. It is also why I attend the workshops. I review the designs: does what I am reading in the design tally with what the customer told me? After everything has been properly laid down at the start, my role diminishes. Then it is above all the project leader who keeps the project moving while I look on from the sidelines. At the end comes the testing and delivery phase. Then I go to see the customer to make sure everything has been delivered to his satisfaction. Are there still things that he wants to change? Does our solution properly match his requirements? What does he want for the future?

Organic growth

Becoming a project manager was a gradual process. In a project team, you get the responsibility that suits you and that you choose to take on. As such, I have performed all kinds of project management tasks over the years. Until at last I got so close to being a project manager that I could finally say "now I am a project manager". Personally I think that's a nice way for it to happen, growing within your role and acquiring ever more responsibility; naturally finding a place where you perform well.

/colophon

Objective is a publication of Technolution B.V.
All previous editions of Objective are available for download from www.technolution.eu/objective

Distribution
Controlled circulation for connections of Technolution

Technolution B.V.
Zuidelijk Halfrond 1
P.O. Box 2013
2800 BD GOUDA
The Netherlands

T +31(0)182 59 40 00
E info@technolution.eu
I www.technolution.eu