# Technology statement

The information below is intended as a "*Technology statement*". It simply touches on the subjects in those areas of technology in which Technolution is active, experienced and holds an opinion of its own. It demonstrates the involvement of Technolution in these areas.

## General technology subjects

**Company attitude towards technical challenges**

When we are faced with a complex problem, we solve this problem by simplification rather than by adding complexity. It requires a lot of hard work to keep things simple: simplicity is the soul of quality. Needless to say inherent, complex problems exist. They still need to be broken down into smaller, simpler pieces that together solve the complex problem.

**Internal technology projects**

To keep up-to-date with technology trends and to get hands-on experience with them, we continuously execute in-house R&D projects. By doing so we break down the; "you don't have experience with this technology" barrier that we sometimes face when acquiring new projects.

**Vendor independent platform selection**

Due to the fact that we are vendor independent, we look for the best suitable components, tools or platform for the project. For embedded systems, this means we choose between OS-less, Windows-CE, VxWorks, embedded Linux or some other (proprietary) solution. In relation to hardware for example the choice of the CPU brand is determined only by rational arguments such as price, performance, experience, portability issues, ease of use, vendor support, availability over time, etc.

**Vendor related partnerships**

Although it seems contradictory to the previous statement, vendor partnerships are solely on the basis of technology access. We currently participate in partnership programs with: Xilinx, Altera and Microsoft (WEP and MSDN).

**Public/Private partnerships**

Technolution is a partner in ERTICO, an organization which is a multi-sector, public/private partnership pursuing the development and deployment of Intelligent Transport Systems and Services (ITS).

**Moving functionality between off-the-shelf hardware, programmable logic and software**

With the availability of programmable-logic-devices (like FPGAs), the relationship between hardware and software is extended. We now have the choice of implementing functionality in hardware, programmable-logic-devices or software (i.e. on CPU cores, but also on CPU cores in FPGAs).

This offers new possibilities and requires close cooperation in the design phase between hardware and software specialists. In a number of projects we shifted functionality from the traditional software domain to the programmable logic domain, thus enabling innovative solutions.

Examples are the realization of a high-speed JTAG interface, dedicated UARTs and various signal / image processing systems. One of the advantages can be cost reduction: in one

case we even replaced a high-performance (expensive) CPU with a small CPU and a simple FPGA.

**Agile system development**

Our industry is frequently reinventing itself. When problems with large projects keep occurring, people will try to find a way to solve them with some kind of methodology. The agile system development paradigm is different (www.agilemanifesto.org) and is fully supported by us because it reflects the way we already work whenever possible.
It is not a method in the accepted sense. Rather, it is a framework focused on quickly delivering a quality solution. It supports methods such as DSDM (Dynamic Systems Development Method) and XP (eXtreme Programming) where customer satisfaction and working software are a central theme.

Sometimes customers demand a more formal way of working (usually the waterfall method). When this is the case we still transparently use most elements of the agile approach for the actual engineering phase.

**Certification**

Certification is a process in which implementations are verified against the specifications they were built upon. Certification is always done by an external party whose responsibility is to interpret the specification and build a certification environment to test the implementations. Certification tests should be done automatically as much as possible to prevent user errors. The ability to perform regression tests quickly and accurately is very important. To this end Technolution has developed its own flexible certification environment.

This environment provides a framework with common functionality such as development and execution of scripts and reporting. With this environment highly automated tests are possible, eliminating the tedious and error-prone manual tests. This results in quick, complete and reliable certification testing.


## Software development

**Software development and documentation tools**

When analyzing, defining and specifying a software system, we use well-established methodologies (like UML, Yourdon, Hatley and Pirbai). However, we always use them “fit to purpose” i.e. in the best possible and practical way, and not rigorously. In other words, they are tools, not targets.

**Programming languages**

Because we have been active in the computer industry for over 23 years, most programming languages have been used at one point in time. There is hardly a subject that has caused more debate among technicians than the choice of a programming language.

The more popular ones today are Java, C/C++/C#. When selecting a language for the job, the most efficient one in terms of speed, familiarity and maintainability should be chosen. For portability across server and desktop platforms Java and C# are the preferred choices, when developing hard real-time embedded systems, C is the only option with possible escapes to assembly language.

**Script languages**

Script languages are used for gluing together all sorts of development tasks. Tools and conversions help when either installing or testing software and systems. Python, Perl, PHP, and JavaScript are some of the more frequently used script languages. We also use them to automatically perform stress tests during electronics/programmable logic/embedded software prototype testing. Scripts and make files are also used to build and simulate programmable logic (nightly builds and self-checking test benches).

**Software development tools**

Tools can shorten development time. A few of the ones we have used are: Eclipse, Borland J/C++ Builder, Microsoft Visual Studio and GNU tool chains. Care must be taken when tools are considered that ”generate” code. Today, no such tools exist with satisfactory results in the embedded field. Performance, size, predictability and maintainability (vendor tool lock-in) are compromised. We follow these developments, but do not yet see any added value.

## Embedded Software Development

Creating embedded solutions requires fundamentally different skills compared to those required for standard IT solutions. Therefore, many technicians at Technolution have solid technical engineering backgrounds. We design and develop embedded applications, hardware and production test software, board-support-packs (VxWorks, eCos), OS porting (Windows-CE and embedded Linux), device drivers and protocol stacks (proprietary, TCP/IP). To do so we use cross-development tool chains and -often JTAG-based- software debuggers that match the chosen hardware platform.

**Target operating systems**

Operating systems come in many styles, forms and shapes. As an independent development company we select the most suitable one for the job. Because not all situations are the same we make a choice from all mainstream operating systems. However, situations exist in the embedded world where time is critical and the functionality of the system is fixed and limited, in that case there is always the choice of no operating system at all.

Desktop: Windows Vista, XP, NT, 2000, Linux
Server: Windows 200x, Sun Solaris, Linux
Real-time: Windows XPe (embedded), WinCE.NET, embedded Linux, eCos, VxWorks, QNX

**Wireless**

Wireless technologies will become increasingly important in the years to come. We have already created solutions integrating and designing the various wireless technologies:

Short-range: Bluetooth, WIFI (802.11x), Zigbee, UWB, RFID, ISM, 868, 434, DSRC, IrDA.
Long-range: GSM, GPRS, UMTS, Mobitex, FM broadcast (RDS / DARC).

**Industrial field buses**

To qualify as an industrial field bus the protocol must be robust and real-time. The industrial automation industry never succeeded in defining the definite standard. We have worked with most of the popular de facto specifications like: CAN (with various sub-protocols such as CAL and Devicenet), Real Time Ethernet (SERCOS-III), Precision Time Protocol (IEEE 1588), Profibus and Echelon.

## Technical Information Systems development

The construction of Technical Information Systems (TIS) is a logical extension of the activities we have been undertaking since 1987. TIS are often IT intensive and characterized by the use of general purpose computing platforms such as PCs, servers and other COTS (Commercial Off The Shelf) system components.
TIS are built by Technolution using databases, middleware, application frameworks (J2EE, .NET), GUIs, web services and so on. Although many principles of sound technical design and implementation are common, the skills needed to design and implement TIS are beyond the embedded domain and therefore such projects are staffed by specialized people. Because of our strong technical background we are able to provide a sound technical architecture where performance and flexibility are built-in from the beginning.

**Middleware**

Nowadays, middleware is almost mandatory when developing distributed, load balanced, fail-over capable applications. Middleware eases the development burden of these applications by providing a solution for all or part of the required features. We have used open standards like Web services (WS-*), CORBA and RPC, as well as proprietary systems like COM/DCOM in several projects.

Recently two concurrent technologies have emerged that can be used for distributed application development: .NET and J2EE. Both offer roughly the same features, both of them having their strong points. We do not make a definite choice for either of these technologies. As a vendor-independent development company we select the best technology for the job at hand.

**Databases**

During the course of application server implementation the use of databases is inevitable. High-end databases like MS SQL2005 server and Oracle were used in several projects. For applications that do not need such heavyweight database servers we resolve to more lightweight products like: MySQL, PostgreSQL, Interbase/Firebird or Paradox. Connections to the databases can be JDBC, .NET dataproviders, ODBC or native, depending on the application.

**Web 2.0**

Web 2.0 is not a single technology in itself. The phrase Web 2.0 has been invented to indicate that a new generation of online collaboration and distributed applications is afoot. Examples of technologies that are commonly associated with Web 2.0 are AJAX, ATLAS, SOAP, Web services (WS-*), RSS feeds, weblogs and wikis. Many of these technologies are not new by themselves and have been used by Technolution as part of our solutions. Regardless of popular phrases we shall continue to choose the best technology fit for the problem at hand with Simplicity and Quality in mind.

**Advanced communications**

Traditionally there always was a distinction between datacom and telecom. Today, from a technological transport point of view, these differences have disappeared, while the architectural differences between packet-switched and circuit-switched networks still remain. Because we work with both architectures we can better accommodate cross-border projects such as Packet-Over-SDH and Voice-Over-IP (VOIP).

Short-range:

RS232, RS422, RS485, 802.x, IEEE 1394 (FireWire), USB (OTG), 10/100/1000/10000 Megabit Ethernet over copper wires (UTP CAT5/6 cable), POF (Plastic Optical Fiber) and glass fibers.

Long-range:

PSTN (modems), ISDN (terminal adaptors), xDSL (modems), HDLC, X-series, V-series.

Long-range transport:

PDH (multiplex / line), SDH (multiplex / line).

**Protocols**

Although not the only protocol available, TCP/IP and all its associated standards and products are the backbone of our modern, internet-based society. Therefore it is one of our core competences. We implement TCP/IP from small 8-bit micro controllers to high-performance server systems. Network design, configuration and troubleshooting are our second nature. We used this knowledge to create routers, bridges, gateways, embedded web servers, I/O networks, PoS networks, and many other systems that simply need to communicate.

Some internet related keywords:

RFCs, TCP, IP, Echo, PAP, CHAP, SNMP, MIB, SMTP, PPP, PPTP, VPN, RSVP, SSL, IPSEC, HTTP the list is endless and on top of that stateless RPC to full streaming architectures.

Other protocols:

X.25, MASC, Q931 (ISDN call set-control).

**Information security**

The purpose of information security is to protect and authenticate information and services. A risk analysis uncovers the information and services to be protected. Based on the risk analysis, technology and methods can be put in place to minimize risk. Some examples of

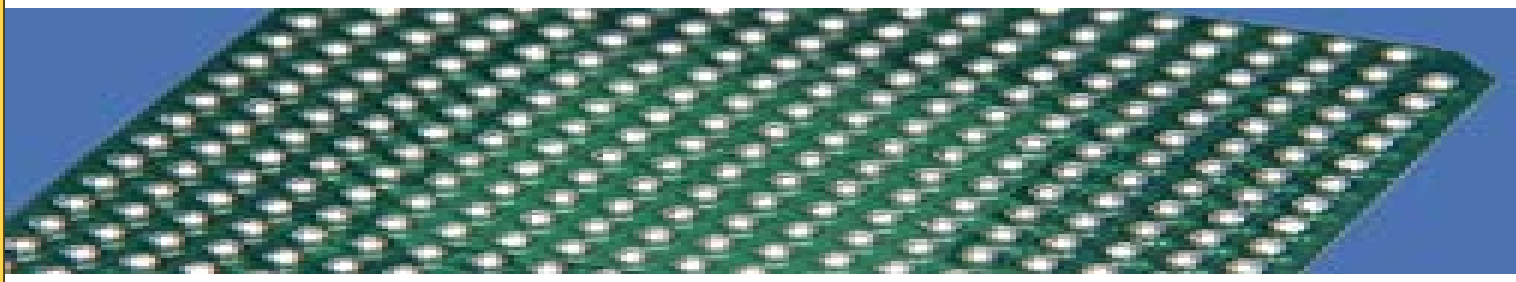
technologies we have used are: SSL, IPsec, HTTPS, PPTP, Kerberos, PGP and PKI. We have knowledge of ITSEC and Common Criteria methods for certification.

**Cryptography**

The main reason why cryptographic technologies are used is to protect data integrity, protect data against unauthorized access and to prove the trustworthiness of data (See information security). We have used AES, Elliptic Curve, RSA, DES, 3DES, SHA1 and MD5.

**Smartcards**

We have a long relation with smart cards. Since 1996 we have implemented E-Purse terminals for Chipknip, Geldkarte, Proton, Moneo, Mondex, Visacash, and EMV (CTAP). In the field of GSM SIM we have used Javacard to implement various applications among which are: Chipknip payments, Mobile CEPS payment and reload. Furthermore we helped with the definition and implementation of several electronic chip card based transaction systems (i.e. vending machines and parking systems).



**Hardware/software virtualization**

An old technology, already used by IBM in the 1960s, makes a re-entry in the sever, PC and embedded computer markets. Hardware virtualization allows multiple OSs to run simultaneously on a single platform, as if they run on their own system.

Applications in the server market are driven by security issues; in the PC and embedded market applications have to make optimal use of the available CPU cycles and reuse of (legacy) software. Virtualization also comes down to consumer devices, like cell phones and set-top boxes. Technolution currently uses VMware for typical PC platforms and VirtualBox for use on high-end embedded systems.

## System engineering

System engineering is the process of translating requirements into a system design. The key is to come to an overall system design in which the different disciplines are integrated and working together. We have experience in combining disciplines like hardware, (embedded) software, programmable logic, software engineering and mechanics into a product or system.

## Programmable logic development

PL (Programmable logic) has been around for over 20 years. In the early days it was a tool for hardware engineers to optimize their hardware designs with respect to component count (glue logic). This has changed dramatically in the last few years, as FPGAs now contain tens of thousands of logic cells that allow complete applications to be implemented in PL. Because of this phenomenon PL has become a separate discipline within Technolution alongside SW and HW. Many designs have already been made because modern hardware platform development cannot exist without PL.
PL within Technolution (and in most of Europe) is mainly implemented using VHDL.
Technolution actively records its experience with the language and the development tools in a document called “Programmable Logic Design Guide and Best Practices”. This document is a guideline for producing high quality designs. As the design size increases, we recognized the need for a clear view on the design methodology with strong focus on verification.
Some examples: PCI(e) interface, DDR(2) SDRAM controllers, various image processing algorithms (including real-time rotation without loss of quality), HD-SDI interfacing, floppy controller core with digital data separator (for obsolescence avoidance), DMA controllers, multi-gigabit optical link protocols, complete high-speed applications and numerous interfaces.

**Programmable logic design environment**

Visual VHDL capture:

Visual Elite HDL Pro from Summit, HDL Designer from Mentor Graphics

Simulation:

Modelsim from Mentor
GHDL and GTKWave (open source)

Synthesis:

ISE for Xilinx and QuartusII for Altera

Debug:

ChipScope for Xilinx and SignalTap for Altera

**Programmable logic devices**

We have designed hardware that includes the following types of logic devices: Xilinx (from Spartan to Virtex 5), Altera (Cyclone and Stratix), Lattice and Actel.

# Electronics development

At Technolution hardware development means designing and engineering electrical circuits on printed circuit boards. Over the years we have developed our own unique hardware development methodology. This methodology allows us to design complex digital hardware in a predictable fashion. This means that we can design "first time right". The essential parts of the methodology are:

* Realistic rapid prototyping of critical parts of the design.
* Timing verification on a 'worst case' basis instead of 'typical'.
* Signal integrity by design (see next item).
* Standard verifications on: cost price, availability, signal function, signal load, signal level, power dissipation, thermal behavior, surface, MTBF, etc.
* Design For Manufacturing and Testability (DFM/DFT).
* We realize our environmental responsibility and design according European RoHS (Restriction of Hazardous Substances) and the WEEE (Waste of Electrical and Electronic Equipment) directives.
* Professional Mentor Graphics tool chain for: design capture, simulation and layout.

**Signal integrity**

When high-speed electrical signals are transferred between electronic components, physical phenomena play some tricks that make it hard to interpret the signal correctly, jeopardizing the robustness of the entire system. The most important of these phenomena are:

* Reflections are caused by changes in impedance along the signal path, deforming the shape of the signal causing it to be misinterpreted. Impedance control over the entire signal path must be addressed. This is done by determining tuning parameters for pins, traces, vias, PCB material, connectors and terminations.
* Crosstalk occurs when inductive (dI/dt) and capacitive (dU/dt) coupling between PCB traces deform the shape of the signal. Crosstalk calculations are relatively simple and good design rules prevent most problems.
* Attenuation makes signals weaker causing noise margins to be reduced leading to possible misinterpretation of the signal. Attenuation increases with distance and frequency. Again, good design rules and simple calculations prevent most problems.
* Timing jitter is introduced by all phenomena above and by supply ripple and by PLLs in the clock path. This decreases the timing margin on received signals. Clock generation and decoupling must be carefully designed to minimize this jitter.

Experience has taught us that a good set of design rules is the basis for a solid high-speed design, followed by verification through simulation of critical signals during pre-layout and post-layout.

Parameters are verified using simulation with IBIS and (encrypted) SPICE and S-parameter models. Other tools are used for PCB stack-up and trace width calculations.

Measurements will finally verify signal integrity when the actual prototype is available (measurement probes are incorporated in the verification models).

In specific cases crosstalk and signal integrity calculations are performed with the help of 3D-field solving techniques.

The methodology described above enables us to design signal paths up to 10 Gigabits per second using ordinary FR4 PCB material. With the use of low εr PCB materials even higher speeds can be achieved.

**Hardware design environments**

| | |
|---|---|
| Design capture and analysis: | eProduct Designer by Mentor |
| Layout & signal integrity: | Expedition PCB Pinnacle by Mentor |

**Measurement equipment**

Even the best calculated and most simulated hardware designs need verification in the end. In order to be able to do these verifications we need measurement equipment that is suitable for the high-speed designs we create. Our top model digital oscilloscope is a 'Le Croy' SDA-6000 for signals up to 6 GigaHertz (20 GHz sample rate) Including an optical to electrical converter and a differential probe with 7.5 GigaHertz bandwidth

**Microprocessors**

| | |
|---|---|
| Intel: | 80x86, 8051/96,i960, StrongARM |
| Marvell: | Xscale |
| National Sem./AMD: | 4(86) and 5(86) series |
| Freescale: | PowerPC, ColdFire, 68k |
| NXP: | LPC 213x, LPC 229x |
| Atmel: | AVR related |
| Texas Instruments: | TMS320C-series (DSP), MSP430 series |
| Cirrus, NetSilicon: | ARM 7, ARM 9 |
| Xilinx: | PowerPC, Microblaze |
| Altera: | NIOS, NIOS II |

**System buses**

We have made boards and backplanes for a number of parallel buses:

VME64, PCI and Compact-PCI, Multibus II, PC104 and PC104 plus.

**On-Chip buses**

On-Chip Buses are used to interconnect IP blocks. Examples are OPB, AMBA and Wishbone.

**PC-buses**

PCI, PCI-X, PCI Express, USB, FireWire, Ethernet, SATA, PCMCIA (PC card), ETX, ETX-E and COM-E.

**CE, UL and EMC**

We have a core competence in testing and problem solving CE-related issues. We have assisted many companies in improving their designs to meet the constraints for the CE

qualification. It goes without saying that all our electronics projects benefit from this knowledge.

**Analogue circuit design**

- high bandwidth, low-noise amplifiers;
- radio frequency applications;
- 16-bit and 20-bit high-accuracy ADCs and DACs;
- numerous sensor interfacing projects.

**Signal processing**

Some technologies we have used are: compression, filtering, edge detection, FFT, radon, wavelets and transformation (1D, 2D, 3D) of digital images. We have created many applications using signal processing principles such as audio matching, echo cancellation, number plate recognition, complex impedance measurements and vehicle classification.

**Switching power supplies**

We understand the issues that make switching power supplies work. This enables us to design active Power Factor Correction, synchronous step down converters, Fly back converters, forward converter, Cuk converters and Buck-Boost converters for stand-alone applications or to integrate them in on-board solutions. We also do high voltage supplies (more than 1500V) and backlight inverters.

**Optical data transmission**

High-speed optical links that are used for telecom and interconnect between system parts in complex machines have been realized a number of times. Some highlights:

- multimode lasers at 850 nm (VCSELs, or Vertical Cavity Surface Emitting Lasers); 16-bit and 20-bit high-accuracy ADCs and DACs;
- Pluggable Parallel Fiber Optic Link Transmitters, arrays comprised of 12 VCSELs with associated control, using a 12-fiber parallel MPO/MTP connector (SNAP12) for high-density cabling and easy interconnection;
- transceivers with integrated control and amplifiers (SFF and SFP) with LC connectors;
- single mode lasers at 1350 and 1550 nm;
- wavelength-tuned single mode lasers for CDWM (Coarse Division Wavelength Multiplexing) which allow multiple data-channels to be optically multiplexed over one single fiber.