THE EVOLUTION OF EMBEDDED SOFTWARE IN CONSUMER PRODUCTS

R H Bourgonjon

Rapporteurs: Robert Allen and Dr Paul Ezhilchelvan

The evolution of embedded software

in consumer products

R.H. Bourgonjon

© Philips Electronics N.V. 1995

Embedded Software in Consumer Products

Part 1.

- Growth of embedded software in consumer products.
- Evolution of embedded software.
- The role of software architectures.
- Multi-media and consumer products.
- Standardization.

Embedded Software in Consumer Products

Part 2.

- Technology and process.
- Software process improvement.
- Software development methodologies
- Conclusion

Embedded Software in Consumer Products

What are electronic consumer products?

- Electronic products meant for households and/or individuals manufactured in mass production.

Embedded Software in Consumer Products

Characteristics of embedded software in consumer products.

- Software is intrinsic part of functionality and is invisible for end-user.
- Software can usually not be repaired or replaced after sales (virtually error-free).
- Software is not independently priced.

Software in Electronic Products

Some facts:

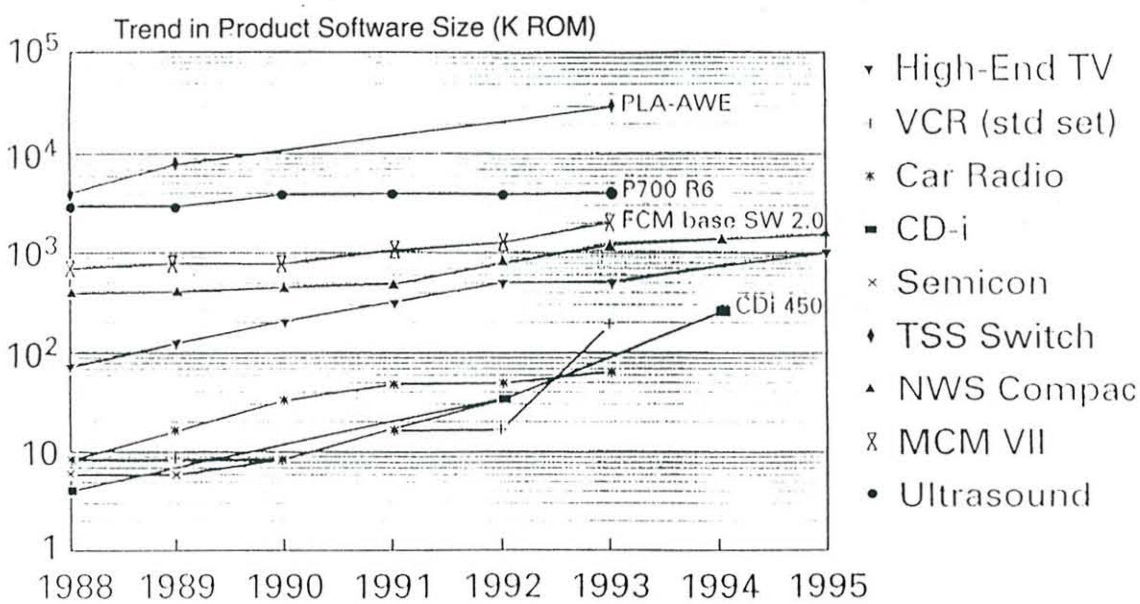
- Almost all (mass produced) electronic products contain increasingly larger parts of embedded software.
- The variety of such products is increasing.
- Functionality and complexity of products is increasing.
- Functionality is shifting between products.
- User interfaces become very important.

Software in electronic products

Consumer products with software (examples):

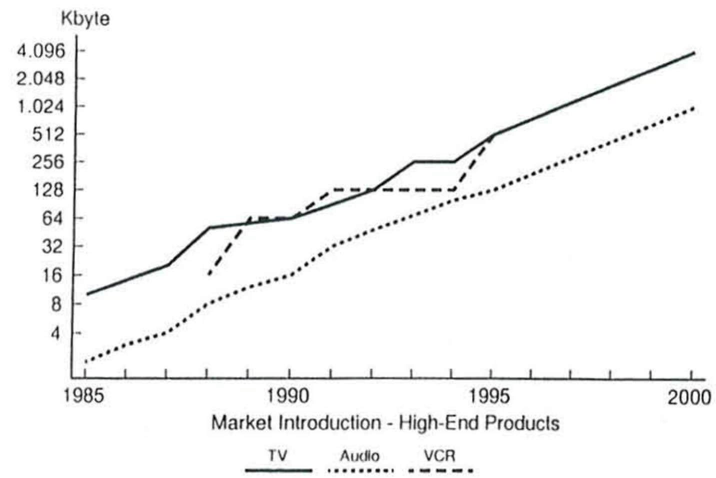
- Television (High-end about 600 K bytes of software);
- VCR (High-end about 265 K - 512 K bytes software);
- Car radio (64 K - 256 K bytes of software);
- Audio sets (64 K - 256 K bytes of software):
- GSM hand held telephone (512 K bytes software);
- Set-top box (512 K - 1000 K bytes of software);
- Fridges, micro wave ovens, washing machines, vacuum cleaners, shavers.

Software in electronic products



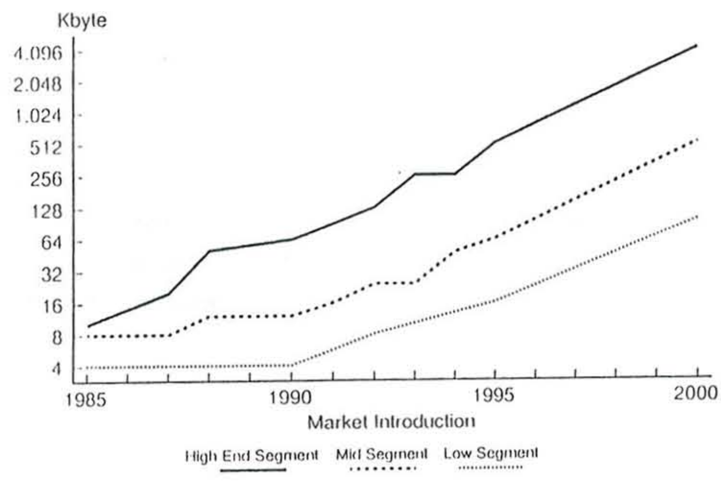
Software in electronic products

Exponential growth of ROM size.



Software in electronic products

Exponential growth of ROM size TV-sets.



Technology evolution

- The cost of computers falls 90% every 6 years.
  - current production is 8000 transistors/person/day.
- The cost of storage falls 90% every 9 years.
- The cost of telecommunications falls 90% every 6 years.

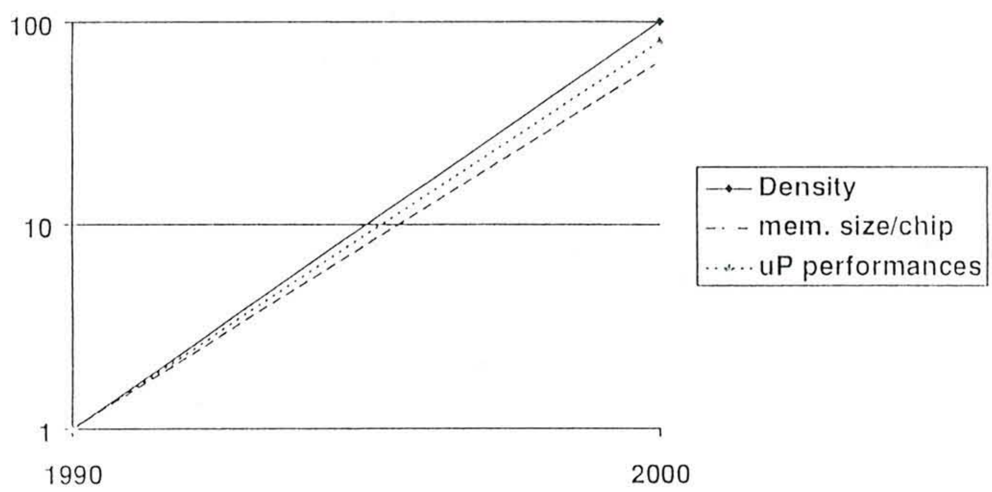
These changes are revolutionary!

- Relative costs fall dramatically.
- The unthinkable becomes commonplace in 25 years.

The fundamentals of software engineering remain the same, but seem to get forgotten and have to be relearnt.

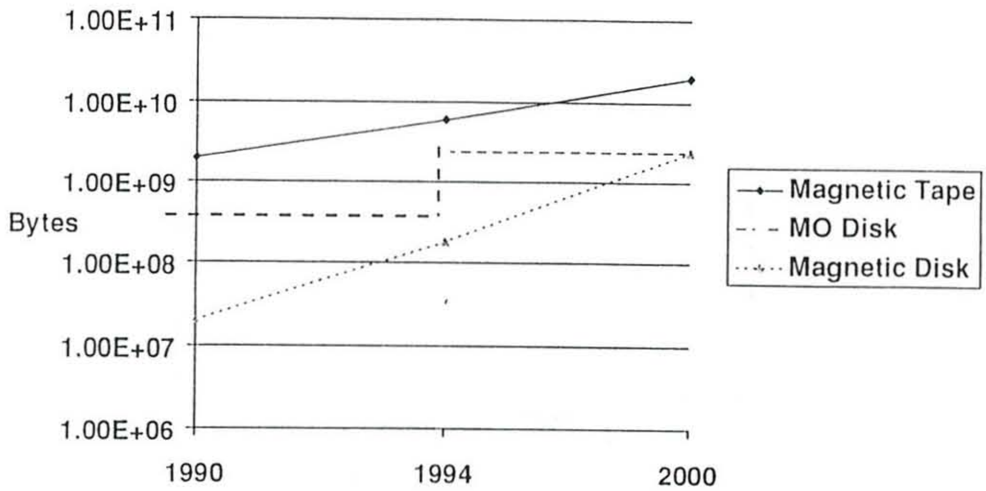
Technology evolution

Technology Trends: Silicon



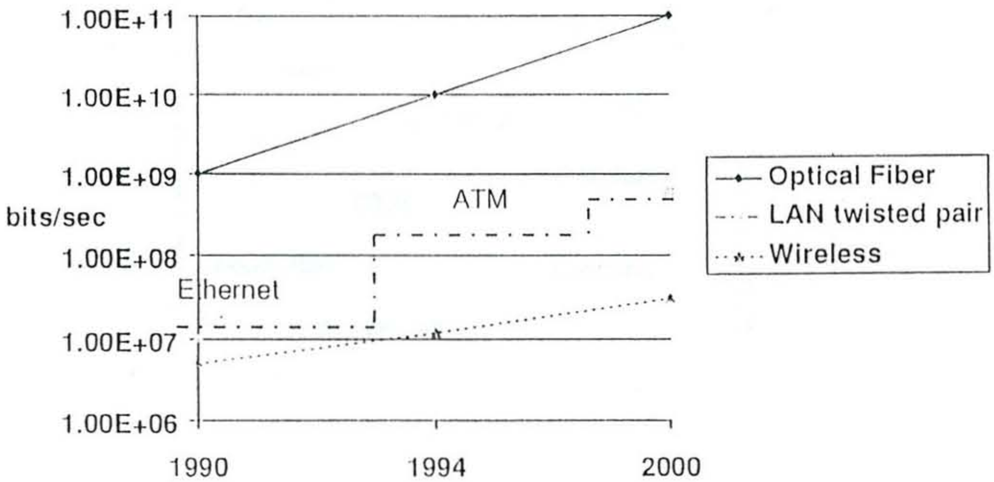
Technology evolution

Technological Trends: Storage

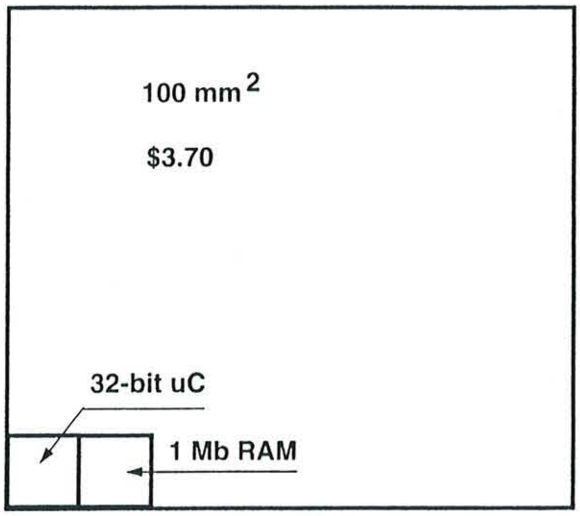


Technology evolution

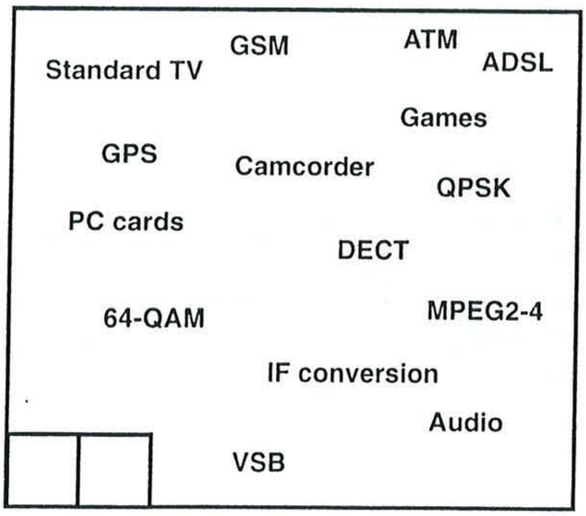
Technology Trends: Transmission



The year 2001: Technology



The year 2001: Systems



Software evolution

Software productivity

- For some applications: 15% compound annual improvement 1955 - 1990. (source: Applied Software measurement, Capers Jones)
- But features grow faster.
- Application size increases at 15 - 25% compound annual growth rate (Microsoft Word 27 K LOC 1st version, 2000 K LOC now).

Software evolution

Evolution of embedded software

- Implementing existing functionality (hardware-bound software).
- Adding new features (monolithic closed software).
- Growing in size and complexity (propriatory architectures).
- Increased "system" aspects (open architectures).

Software evolution

Implementing existing functionality (hardware-bound software).

- Software is considered as more easy alternative for hardware.
- Software is treated as pure development issue (= cost).
- Hardly any effect on the organisation.

Software evolution

Adding new features (monolithic closed software).

- Possibilities of software are exploited.
- Software features become opportunities.
- Starts impacting other parts of the organisation (marking, logistics, etc.)
- Out-sourcing needs mature subcontracting management.

Software evolution

Growing size and complexity (propriatory architectures).

- Dramatic growth in size and complexity.
- Need for improving user interfaces.
- Growth in team size and lead time.
- Dramatic growth in management and supporting activities
  - software engineering tools;
  - software development environment;
  - quality aspects;
  - change management and control.

Software evolution

Increased "System" Aspects (open architectures).

Products become systems:

- interdependencies and interfaces
- modularity; global architecture
- design for change
- system lifetime exceeds product lifetime
- (de-facto) standards

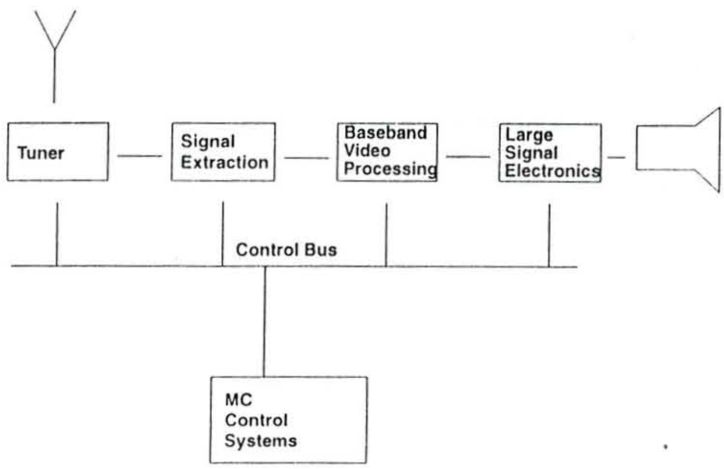
Software evolution

Multi-media is the most striking example.

- Multi-media platforms.
- Multi-media networks.
- Multi-media servers.
- Multi-media content provision.

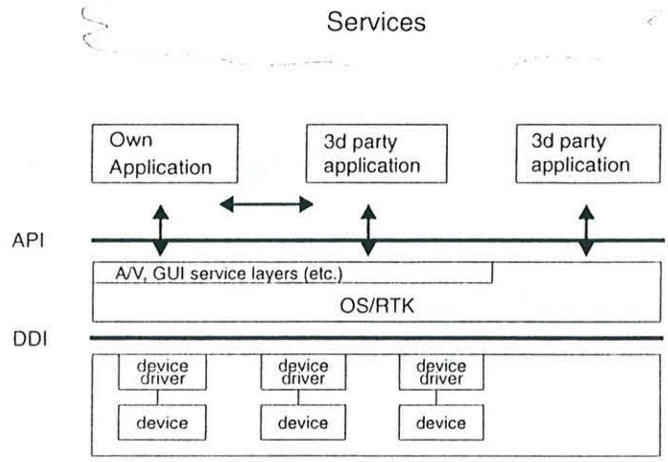
Software evolution

Modelling Current TV Architectures



Software evolution

Modelling future TV Architectures



Feature TV Platform

Market evolution

Markets are uncertain.

- Technology evolves fast (partly predictable).
- Potential products and markets grow faster (very unpredictable).
- Uncertainty is increasing.
- Cross-relations become dominant (instability).

Embedded Software in Consumer Products

The role of Software Architectures

Software Architecture

Why an Architectural Approach is needed.

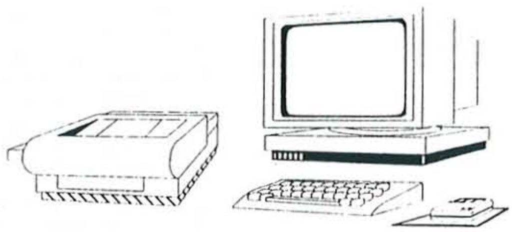
Electronic products are changing:

- From Boxes-only to System Solutions.
- From Hardware-only to Software-controlled Products.
- Convergence of Consumer Electronics, Computer and Communication Industries/technologies.
- Subject to rapid technological developments.
- Mastering complexity.

Software Architecture

You do not need an architect for:

You do need one for:



Architecture: Partitioning and Well-defined interfaces.

Software Architecture

A good architecture offers

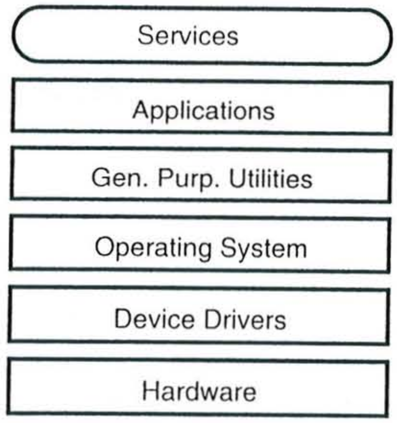
- Flexibility — design for change
- Efficiency — design for reuse
- Product family support — design for scalability
- Consistency — predictability from partial knowledge
- Orthogonality — consists of independent functions
- Completeness — all functions to given class are provided
- Transparency — information hiding
- Interoperability — support of open systems

and allows to

- protect and re-use previous investments
- establish and maintain standards
- allow flexibility and speed in product development.

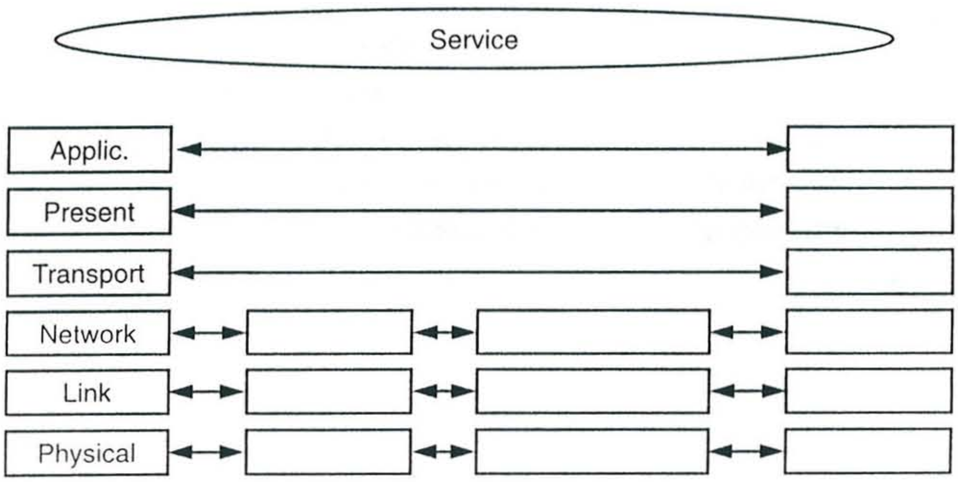
Software Architecture

Diagram 1: Layered processing



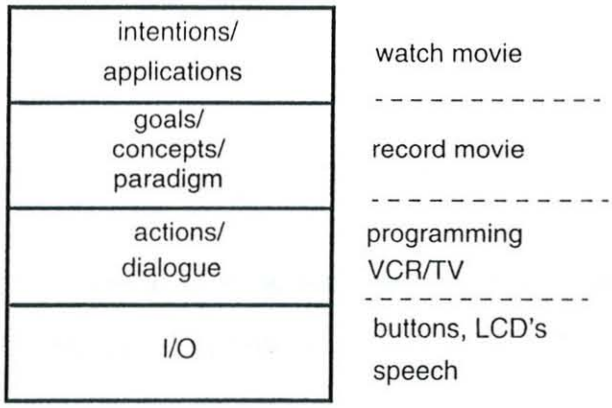
Software Architecture

Diagram 2: A layered view of communication



Software Architecture

Diagram 3: A layered view of user interface.



Embedded Software in Consumer Products

Multi-media and consumer products.

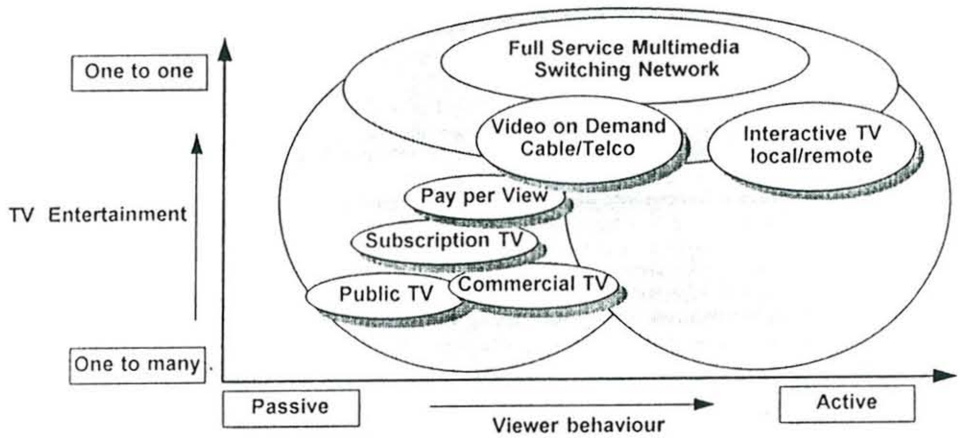
Multi-Media and Consumer Products

The future of Television

- TV watching is changing towards increased individualisation and interactivity.
- This trend is enabled and supported by progress in digital compression, micro-computers, and software.
- The delivery chain will be longer be determined by traditional players, but by companies that
  - provide new services like video-on-demand, and
  - control different parts of larger heterogeneous systems.

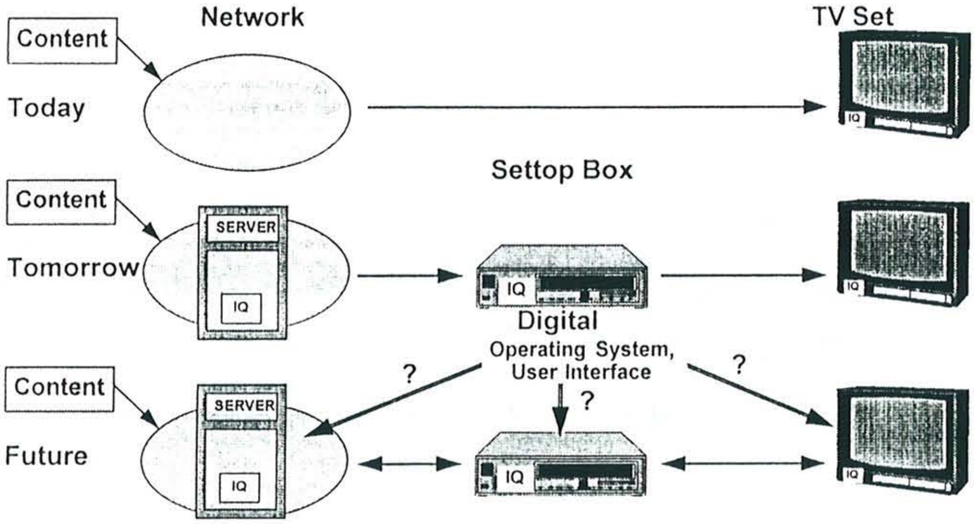
Multi-Media and Consumer Products

Media Services Development.



Multi-Media and Consumer Products

Where will added value be?



Multi-Media and Consumer Products

Stakeholders and Value Chains

- Content and rights owners, service providers:
  - Hold today about 50% of revenue stream,
  - Eager to expand their business via new delivery media.
- Telecom and cable network operators;
  - Hold today about 35% of revenue stream,
  - Need to expand because of declining revenues in traditional business.
- Computer companies;
  - Hardly present in today's delivery chain,
  - Consider residential and server market as an extension of today's business.
- Consumer equipment manufacturers;
  - Hold today only 10-15% of revenue stream,
  - Want to expand from "brown goods" to "systems".

Multi-Media and Consumer Products

Relevance of Architectural Approach.

- In a rapidly changing environment a stable reference frame is needed to preserve consistency.
- Existing interfaces are changing, therefore the total delivery system has to be considered.
- Increasing part of the video delivery chain (server, cable box, TV set, CD-i) will be software controlled.
- Added value will shift to higher layers in architecture, and its physical location is uncertain.

Multi-Media and Consumer Products

Basic System Architectures.

SERVICES

APPLICATIONS	TV watching, info retrieval, games, video on demand, ordering, billing
USER INTERFACE	graphics, menus, navigation
GENERAL PURPOSE UTILITIES	audio/video functions, communication, data handling, administration
REAL TIME OPERATING SYSTEM	scheduling, memory management, I / O management
DRIVERS	tuning, decoding, demodulation, noise reduction, scanning format
I / O DEVICES	display, speakers, tuners, microprocessors, remote control

Microsoft Windows dominates this part of architecture in PC world

Multi-Media and Consumer Products

Basic architecture	Dedicated MM Player	Dig. cable box	TV set 2000
APPLICATIONS	games, info retrieval, etc.	video on demand, ordering, billing	TV watching, games, info retrieval
USER INTERFACE	graphics, menus, navigation	source selection, navigation	image & sound rendering
GENERAL PURPOSE UTILITIES	audio/video functions	audio/video functions, communication	audio/video functions, teletext, communication
REAL TIME OPERATING SYSTEM	RTOS	RTOS	RTOS
DRIVERS	decoding	decoding, scanning format	demodulation, noise reduction
I / O DEVICES	remote control, tracker ball	power supply, remote control	display, speakers, remote control

Embedded Software in Consumer Products

STANDARDISATION

Standardisation

Changing role of standardisation.

- Formerly only standardisation of interfaces between equipment (boxes) of potentially different suppliers. Not concerned with internal architecture.
- Borderline of physical boxes less and less relevant also for consumer products.
- Emerging standards based on architectural considerations.

Standardisation

Standard categories.

- Platforms;
  - General purpose (WIN32, POSIX.1, TALIGENT, POSIX.4);
  - Domain specific (DAVIC, TINA).
- Data:
  - Formats (MPEG2-VIDEO, AC3, MHEG-5, GIF, JPEG);
  - Manipulation (SQL3).
- Communication
  - Distributed computing (CORBA 2.0, OLE-2);
  - Data communication (SDH, ATM, AAL5, MPEG2-TS);
  - Management & Control (TAPI, SNMP, MPEG2-DSMCC).

Standardisation

Standardisation approaches

- Digital Video Broadcast (DVB)

  Bottom-up by extending existing analogue broadcast standards from non-computer arena.

- DAVIC

  Building on-top of existing computer and telecommunication standards (ATM, TCP/IP, CORBA, DSM-CC, MPEG2).

Standardisation

Standardisation Caveats

- Terms are often loosely used

  (organisations and their standards are mixed-up).

- Compatibility and/or overlap difficult to detect.
  - Standards are often grouped (CORBA-2.0, MPEG2);
  - Standards are often stacked (e.g. DAVIC);
  - Different application domains have overlapping and conflicting standards (e.g. encoding rules for data transmission).

Embedded Software in Consumer Products

TECHNOLOGY AND PROCESS

Technology and Process

Software Engineering

- Computer based applications are increasingly complex.
- We choose to put that complexity into the software.
- More and more software is business critical.
- Quite often, software is safety critical.

Constructing complex and important systems is engineering.

Technology and Process

Software 2000

People employed

Software categories	1994	2004
Software business	400 000	1 M
Embedded Software	2 M	10 M
Automation (IT)	2 M	1 M ?
Intermittent Programming	20 M	200 M

Technology and Process

Three major challenges:

- Predictability / quality;
- Mastering complexity;
- Productivity, development leadtime, flexibility.

Technology and Process

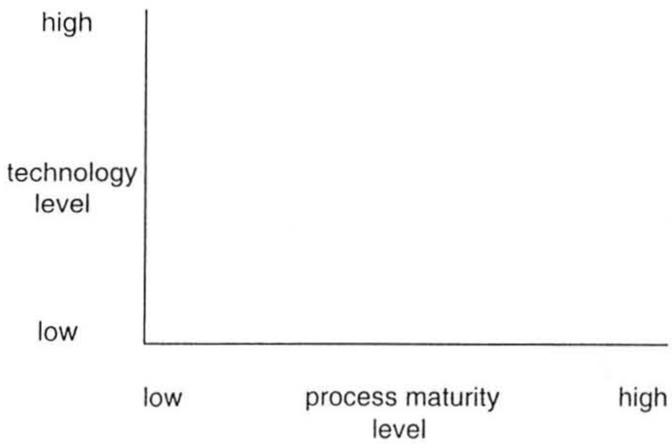
In more technical terms:

- Mastering the software engineering process;
- Systems, design, system architecture;
- Methods, tools, standard components.

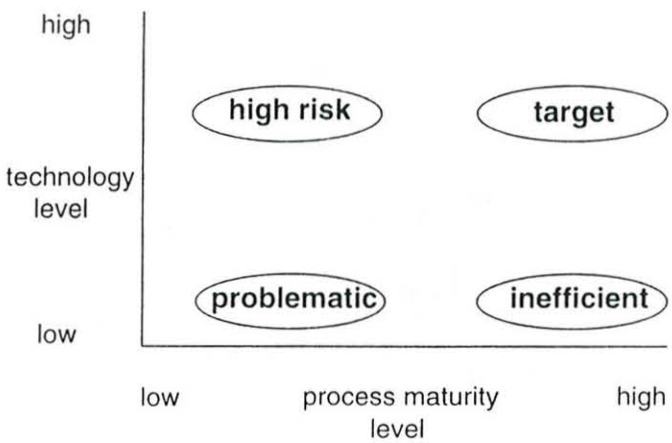
Technology and Process

- Software Process
  - organisation;
  - management;
  - measurements (metrics);
  - control.
- Software Technology
  - methods;
  - tools;
  - environments.

Technology and Process



Technology and Process

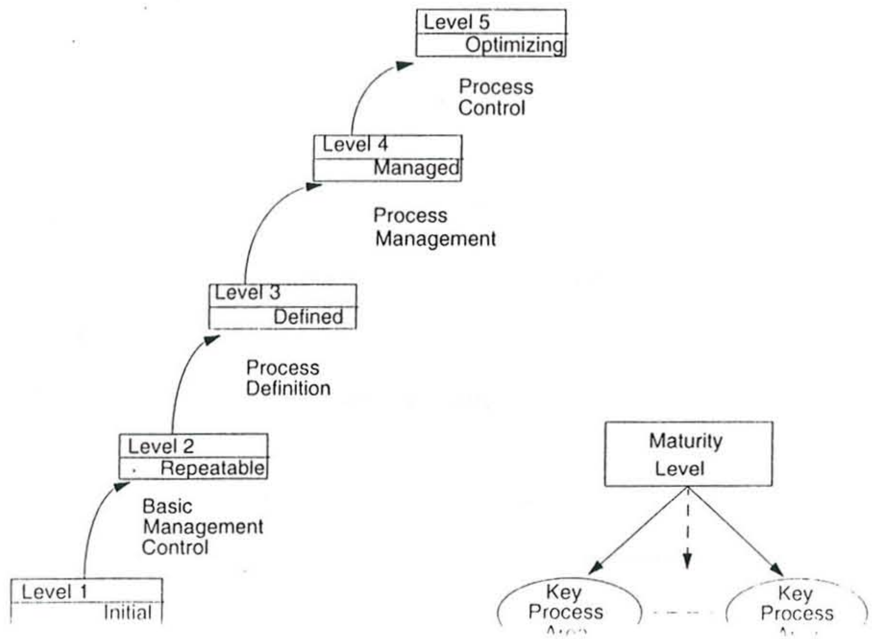


Technology and process

Key areas Software Engineering Process

- Requirements management.
- Configuration management.
- Change management.
- Project planning.
- Project tracking.
- Quality assurance.
- Subcontracting management.

Technology and Process



Technology and Process

Software productivity

- The future demand for embedded software cannot be met with just more people.
- Current improvements in software productivity are not sufficient either.
- The growth of embedded software will be determined by
  - De-facto standard architectures allowing for domain specific reuse;
  - Domain specific development environments (Domain Specific Formalisms)

Design Technology

Architectural models

- object libraries;
- object frameworks;
  - architectural framework,
  - component libraries.

Design Technology

Domain Specific Formalisms & Applications

Aim

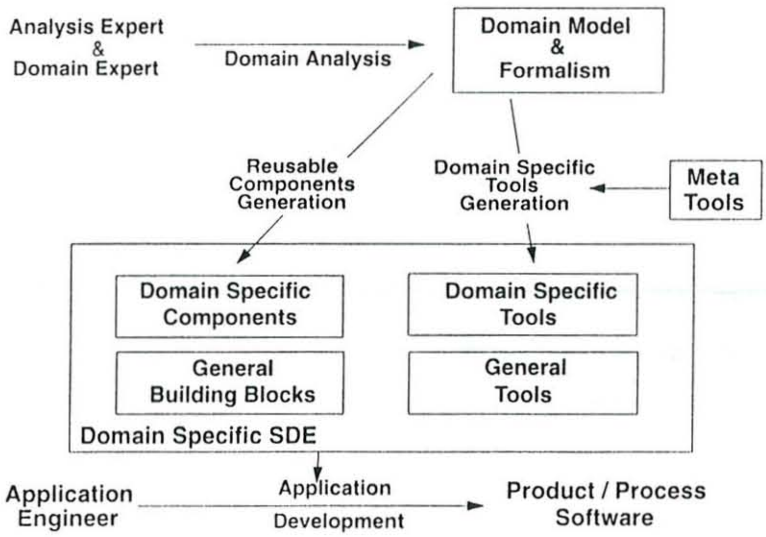
The fast generation of dedicated easy-to-use software-based production tools.

Relevance

For specific domain:

- high quality (embedded, re-usable) software
- compactness of generated code
- shorter development lead time less computing background required.

Design Technology

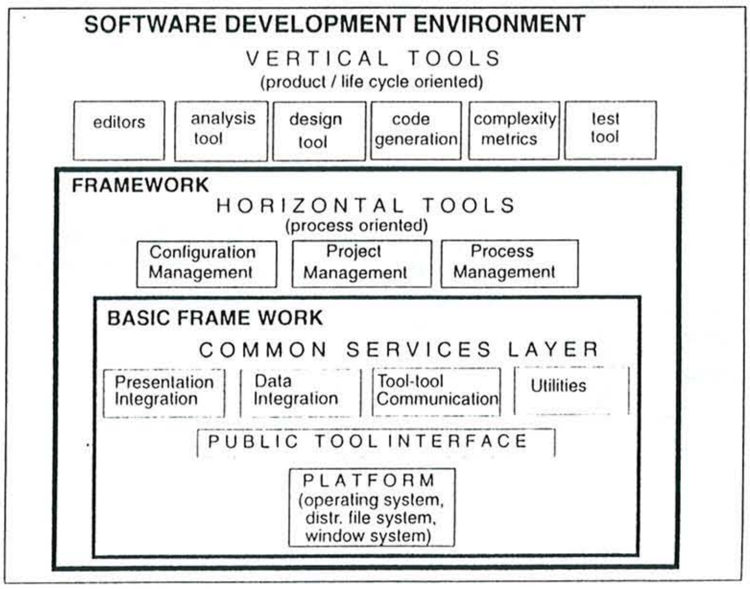


Design Technology

Software Development Environments

- A set of tools for supporting the development
- A framework for integration these
  - presentation integration;
  - data integration;
  - tool communication.

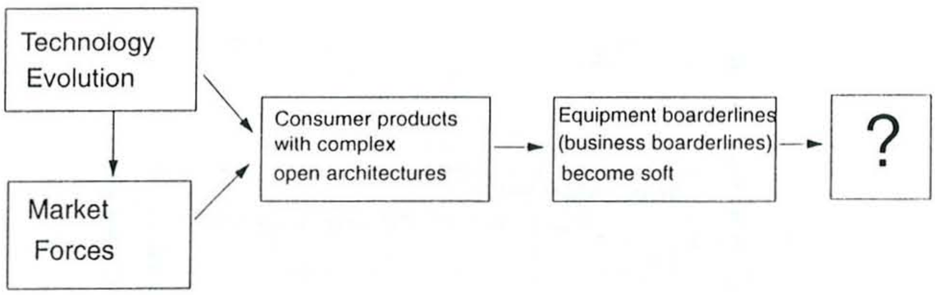
Design Technology



Embedded Software in Consumer Products

CONCLUSION

Conclusion



Conclusion

The old scientific management was about ensuring control.

The new will be about make sense out of chaos.

Quote Harverd Business Review.

DISCUSSION

Rapporteur: Robert S Allen

Professor Katzenelson referred to the issue of software sizes and asked if these were the sizes of the software inside the products.

Dr Bourgonjon replied that yes, it wasn't the supporting software or the environment, but the actual software inside the products.

Dr Herbert wanted to know if Dr Bourgonjon had any feel for how the software ROM for televisions was split between different functions. He proposed that teletext handling, logic control for different channels and VCRs, and basic television controls must be components.

Dr Bourgonjon explained that a large part of the software was involved with the user interfaces including on screen displays. Another significant factor was the provision of features such as auto switch off. He commented that televisions are rapidly changing.

Dr Herbert commented on discussions from the software 2000 workshop, about the development curve for both silicon and software. He pointed out that the time it took to build a new fabrication line, and the cost, is so large that companies can't afford to bring new products to market. He also remarked on the difference between the market side and the technology.

Dr Bourgonjon agreed, describing the increasing use of standards in memories and processors, and the use of standard component libraries. The software, however, is a different story, he said.

Dr Herbert remarked that he wished that agreement could be reached on standards for batteries, to avoid transformer problems in consumer products.

Professor Randell asked if Dr Bourgonjon's comments on open standards referred to interfaces defined within Philips or across different companies.

Dr Bourgonjon replied that he was talking about both.

Dr Lesk asked what could be done to encourage open standards, when some products relied on proprietary software paradigms, and others were public such as Philips audio CD timing signals.

Dr Bourgonjon explained that open standards were already emerging, and it was happening automatically. Standardisation was occurring between boxes not inside, and more and more groups were coming together to reach agreements.

Dr Lesk continued that compilers are designed to accept other formats, but the reverse wasn't true, and the same applied to word processors. At the same time, companies wanted their format to become standard.

Dr Bourgonjon thought that this was a necessity, as software is often not developed by one company, and there is a requirement to incorporate third party software. He anticipated more open standards in the near future.

Professor Kopetz stated that there were two ways to structure such an application, layering and partitioning. The former had been mentioned, but the partitioning approach was gaining recognition in consumer products, with autonomous components with clear communication protocols and interfaces between them.

Dr Bourgonjon agreed that partitioning was becoming more important. He gave the example of audio and video functions, and the use of logical building blocks. This was an area of development.

Professor Kopetz asked if these were implemented in dedicated chips.

Dr Bourgonjon replied that it depends on the application. He explained that some of the functionality of say the audio/video products are implemented in dedicated chips, but that this could also be done on a general purpose machine.

Dr Herbert expressed an interest in the move towards 'soft' embedded electronic products. He compared software with the design methods of hardware, with its well understood modularity, boards and circuits. He wondered if any lessons from this engineering background were being brought into software design.

Dr Bourgonjon suggested that this was difficult, in that in engineering, the boundaries and physical parts were fixed, and the flexibility was intentionally left to software. The hardware remained rigidly modular. Companies were now trying to stretch this to the limit, exploiting the maximum possible flexibility.

Dr Lesk asked if there was a move in Philips towards self describing components, such as the TV set controller detecting options. Suggested an example that a large amount of effort might be wasted in trying to design a phone switch which could discover what kind of device was connected to it.

Dr Bourgonjon didn't think this was currently done for televisions, but a large effort was going into the next generation, with lots of different developments.

Dr Herbert pointed out that some televisions were already aware that they were talking to a VCR, to decide on channel switching.

Dr Bourgonjon agreed that this did occur at the interface, but not a software package looking around for the types of equipment, at least not yet.

Dr Herbert concluded that the information was available, should the software be developed.

Professor Randell commented on the earlier point about the relative lack of concern about software development cost, and wondered how much concern there was about software development time.

Dr Bourgonjon explained the difficulties of predicting both development time and when the product is finished. He emphasised that products aimed at the Christmas market if not completed on time would become a year late. A secondary point is time to market.

Mr Rigg asked if there was any move towards intelligence in televisions, for example to automatically detect which side of the video signal contained the sound. This would be useful to make UK and continental videos compatible.

Dr Bourgonjon stated that these devices already exist, albeit on more expensive models. Voltage detection between different countries was already in place in universal television sets.

Professor Randell commented on the continuing problem of the different plugs!

Dr Bourgonjon continued that all manner of features could be imagined, the question is whether they would sell or not. He gave the example of speech control as one which does sell.

Mr Evans commented that inside a VCR there were several processors, not just one. He asked if this was due to the use of different software development teams, or was it a legacy issue.

Dr Bourgonjon replied that it wasn't different teams, but originally different, independent micro controllers were used, controlling different physical engines, for which the software size grew. He explained the use of distributed programming and a central controller in televisions.

Dr Herbert said that he had difficulty extrapolating the process. Looking at general purpose workstations, he could see more and more video features being added. He stated, however, that the model of programming used there is different from the embedded products model. Third party products seem to have been mentioned as an afterthought, and he asked if personal computers and televisions in ten years time would be distinguishable, and if so would it only be by the method of delivery of the software.

Dr Bourgonjon thought that this issue was one of the biggest debates that his company was having. He described a general consensus within the company from a technology point of view, that the systems will converge, with interactive television. He put forward a marketing argument that the environments within which the two are used are very different, in that television is a community activity, sitting in front of the set, whereas the belief is that using a computer is an individual task, associated with being 'behind' the PC. As a researcher he thought that this must be clarified.

Professor Katzenelson proposed that a community game would bridge these two environments.

Dr Bourgonjon maintained that although the two uses were different, the technology would converge.

Professor Randell commented on the lack of integration between different devices, such as scanner, fax, copier and printer. He expressed his surprise that such combined devices had not been developed, and his view that technological issues were often only a small part of the overall problem.

Professor Whitfield asked if there was likely to be a movement towards consumer led devices, such as washing machines checking the existence of the water supply. He expressed his desire for more flexible interfaces.

Comments followed that such a wide ranging interface would be too complex, and it would be difficult to anticipate the needs of individual users. An example was given of a washing machine that switches off when the phone rings, but this would not be to everyone's satisfaction.

Mr Martin joked that a false assumption was being adopted - that the products were intended to meet the consumers requirements.

Dr Bourgonjon described market studies which refute the claim that such wide flexibility was wanted by consumers. He gave an example of television which provided a facility of selecting your own programme number and your own name for different channels. He proposed that with very modular systems in the future, new features which were required could be added easily. He stated further that current devices were being produced with many more features than were used, and that 90 percent of people lose their instructions within six months.

Dr Dicker proposed a 'domestic network' or 'central nervous system for the household'. He remarked that many devices currently used, even if made by the same manufacturer were in no way connected. He gave an example of going to bed without switching on his washing machine, and wondering why he can't use the remote control for his VCR to do this. He expressed his view that there was a need to connect such devices, and he thought that there was no evidence of any progress or even proposals in this area. He proposed another scenario, whereby he could phone up his preset cooker to notify it of a delay in coming home. He suggested that this would not occur for a long time.

Dr Bourgonjon replied that such ideas have been around for a long time, such as over the phone systems. He thought that technologically, it would be easy to implement such systems within buildings.

Professor Randell extrapolated the scenario to expose problems whereby you couldn't get to a flooded area because the locking system had sealed it off, and the lighting system was disabled.

Professor Kopetz commented on the differences between user interfaces of different devices, and how different companies have addressed similar control problems. He asked if there was any tendency towards a standard user interface between products - perhaps like Windows!

Dr Bourgonjon explained that many activities involved in standardisation were being carried out, particularly for basic audio visual functions. However, he suggested that companies also wanted to give devices their own flavour, or added value, which helps them to sell. He thought that some basic controls have similarity though.

Professor Randell thanked the participants and wondered how many of them could program their own VCR.

DISCUSSION

Rapporteur: Paul Ezhilchelvan

Lecture Two

When Dr R Bourgonjon (the speaker) was presenting an architecture for embedded software, Professor Randell expressed concern over the speaker's emphasis on the required independence between blocks. The speaker shared the concern expressed and admitted that in certain cases it may not be possible to have the said degree of independence.

On the DAVIC standards, Professor Randell wondered whether the standards are international and whether distinct European and American standards are emerging in parallel, as they did in other areas. The speaker confirmed that DAVIC standards are unique and International. Dr Herbert commented that Europe no longer produces its own standards in these areas and expressed a cynical view that some standards are being produced too fast and may be found in near future not sufficiently comprehensive.

Professor Lesk wondered whether any metric is being used (within the Philips Research Labs) to monitor and measure the progress of software production. The reply was that any measurement feature that is being used will not strictly qualify to be a metric in the way the term metric was meant in the question. The speaker then went on to explain the nature of measurements carried out within his institution. More often, the degree of deviation between the original estimated cost and the actual cost was measured; also monitored were the number and nature of bugs found after testing.

The speaker was then asked whether the existing complexity present in the marketed products is indeed necessary and whether much of this complexity could have been eliminated. He responded by saying that the presence of avoidable complexity is in response to market forces. He cited examples of customers preferring seemingly-complex products with embedded software. This answer led to a discussion about the impact of considering or not considering human factors in product design. Professor Gladman cited an example where a system of sophisticated design performed rather poorly because the designers did not understand the technical abilities of the customers.

Professor J Katzenelson wondered about the delays and difficulties which could have been caused by the personnel changes during code generation process. The speaker pointed out that more difficulties are caused by changes in the applied technology.