

Optimisation of MHN architecture2.doc
28 Feb 99 LAR, v1.7

Optimisation of Multimedia Home Network Architecture

By

**Leif Arne Rønningen
Telenor Satellite Services**

Extract

In this document a large number of issues about Multimedia Home Networks has been handled, including proposing new solutions.

The most important results are within:

- Iterated design methodology
- Generalised object oriented processing structures
- Architectures of Multimedia Home Networks
- Unified User Interface, service control
- Satellite Unit architecture
- Multimedia Processing Unit architecture,
- How to speed up Java
- HDTV MPEG-2 Picture-in-Picture decoder,
- High-performance graphics
- Hardware/Java co-design of Protocols
- Security strategy
- Personal, mobile IPv6 addressing in satellite networks
- Common Interface Modules

Contents

PART A.....	6
IN THIS DOCUMENT (1).....	7
WHAT IS MHN ?	7
DESIGN AND OPTIMISATION OF ARCHITECTURE OF MHN	7
COMMON INTERFACE MODULES	8
REFERENCES	9
LIST OF ABBREVIATIONS	15
THE COMBINED DIGITAL SATELLITE BROADCAST AND INTERNET SYSTEM – THE FUTURE NETWORK (2).....	17
INTRODUCTION	17
THE FUTURE TSS NETWORK.....	19
STANDARD TV AND HDTV, ENHANCED BROADCAST	19
ENHANCED BROADCAST, INTERACTIVE TV AND WEB/INTERNET ACCESS	21
PARALLEL PICTURE-IN-PICTURE (PIP) STREAMS AND WEB ACCESS	22
SCALEABLE, LAYER-STRUCTURED MULTIMEDIA APPLICATIONS.....	22
SYNCHRONISATION OF LAYER-STRUCTURED MULTIMEDIA PLAYS	26
GAMES	29
COMMERCE ON DIGITAL TV/ INTERNET/ WEB.....	29
<i>Pre-booked PPV and token PPV</i>	30
<i>Payment on Web/Internet</i>	30
MULTIMEDIA HOME PLATFORM.....	31
BUSINESS MANAGEMENT SYSTEM – BMS	31
SECURITY.....	32
ROUTING AND CACHING SCHEMES	33
DATA BROADCAST CAROUSELS AND PROTOCOLS	33
<i>DVB Data and Object Carousels</i>	33
<i>DVB Multiprotocol Encapsulation - MPE</i>	34
<i>TheTCP/ IPv6 protocol stack</i>	34
<i>The limitation of TCP via satellite</i>	34
<i>Transfer of Audio-Visual content using UDP</i>	34
<i>The HTTP1.1 Web protocol</i>	35
REMOTE DOWNLOADING OF SOFTWARE	36
PART B.....	37
DEVELOPMENT IN PHASES (3).....	38
INTRODUCTION TO DEVELOPMENT METHODOLOGY	38
DESIGN AS ONE PHASE IN THE DEVELOPMENT PROCESS	38
DISCUSSION ON PHASES	39
OPTIMISATION IN DESIGN (4).....	43
INTRODUCTION	43
OPTIMISATION IN DESIGN	43
HARDWARE/ SOFTWARE CO-DESIGN.....	45
HARDWARE /JAVA SOFTWARE CO-DESIGN	46
SOFTWARE DESIGN	47
<i>Design methodology</i>	47
HARDWARE DESIGN	49

<i>From Java classes to VHDL</i>	49
<i>ASIC design methodology and processing architectures</i>	49
<i>Standard processing architectures</i>	51
MHN FUNCTIONAL REQUIREMENTS (5)	52
INTRODUCTION, MHN SERVICES.....	52
STANDARD DIGITAL TV/ HDTV	53
SERVICE CONTROL	54
OPERATING SYSTEM FUNCTIONS	56
MPEG-2 PIP DECODING	58
GRAPHICS	58
SATELLITE RECEPTION AND TRANSMISSION.....	59
TERRESTRIAL RECEPTION AND TRANSMISSION	60
SECURITY FUNCTIONS	60
PROTOCOLS, DEMULTIPLEXING	60
EXTENDED USE OF IEEE 1394	62
EXTENDED USE OF COMMON INTERFACE	62
ARCHITECTURE AND DESIGN OF MHN (6)	65
INTRODUCTION	65
THE MHN ARCHITECTURE	65
THE IEEE 1394 BUS.....	66
PERSONAL, MOBILE, GLOBAL ADDRESSING WITH IPV6.....	66
THE GATEWAY & SECURITY UNIT	67
GENERAL AV UNITS.....	67
UMTS.....	67
THE SATELLITE UNIT	67
THE MULTIMEDIA PROCESSING UNIT.....	69
<i>IP Switch/ Router</i>	69
<i>Bootloader</i>	70
<i>Service Control and Applications</i>	70
<i>How to speed up Java</i>	71
<i>Demultiplexer & Protocols</i>	72
<i>MPEG-2 PIP decoder</i>	77
<i>Graphics</i>	77
<i>Common Interface</i>	77
<i>Smart-card, IR IO</i>	77
PART C	78
COMMON INTERFACE MODULES (7)	79
INTRODUCTION	79
JAVA COMMON INTERFACE MODULE.....	79
GENERAL PROCESSING	79
GRAPHICS PROCESSING	80
MEMORY EXTENSIONS	80
FILTERING.....	80
CACHING.....	80
FORMAT CONVERSION.....	80
VIDEO PHONE.....	80
DVD	81
DV	81
TUNER&DEMODULATOR	81
MIGRATION SUPPORT	81

Part A

In this document (1)

This document consists of three parts.

- **Part A, Chapters 1 and 2:** Gives an introduction to this document, and the TSS Combined Digital Satellite Broadcast and Internet System
- **Part B, Chapters 3-6:** Describes specification, architecture and design of the future Multimedia Home Network, MHN
- **Part C, Chapter 7:** Describes Common Interface Modules for 1st and 2nd generation IRDs

Focus is put on the design and optimisation of architectures, using software/hardware co-design. The total Quality of Service, QoS, plays an important role, and especially performance and cost are emphasised. Standards are used as far as possible, and are mostly referenced. This includes specifications from DVB, IEEE, W3C, ISO, ETSI and others. Where specifications are missing, extensions are proposed.

What is MHN ?

MHN, Multimedia Home Network, as proposed here, uses the IEEE 1394 serial buss to interconnect all kinds of electronic AV- (Audio-visual) and other equipment to be used in homes. This includes digital TVs, DVDs, DV cameras, PCs, PC peripherals, game machines, UMTS, video-telephones, telephones, and other equipment. The services supported are TV Enhanced broadcast, Interactive TV, Internet/Web access, radio, and multimedia services in general. Most multimedia services except video conferencing, use two-way, unbalanced communication, either via satellite, cable networks, telephone networks, terrestrial broadcast networks, or from local storage devices. Short-range, portable equipment like UMTS is also included.

MHN includes concepts of the Multimedia Home Platform, MHP, as defined by DVB MHP045r12 and the Satellite Interactive Terminal, SIT, as defined by the SIT ad hoc group.

Equipment like DVDs, DV cameras and PCs is already marketed with IEEE 1394 connectors and protocols.

Design and Optimisation of architecture of MHN

In Chapter 3 development methodology through phases and iterations is discussed. A Markov chain model is established.

Optimisation in design a very complex problem that is hard to formulate and solve analytically. A qualitative approach with focus on performance, scalability and cost, is followed.

A large number of projects concerning hardware/software co-design is in development throughout the world, and a few are shortly referenced. In the document hardware/Java

software co-design has been addressed and methodology is proposed. This includes presentation of a generalised processing structure for object oriented Java specifications. In Chapter 5 some MHN functional requirements are put forward. The basis is DVB standards, MPEG-2 HDTV and Internet/Web standards. A unified user interface for service- and equipment control is proposed, based on Web technology. Moreover, specifications for HDTV and advanced picture-in-picture facilities, advanced graphics, security and protocols are worked out. At the end, protocols for extended use of Common Interface are outlined.

Chapter 6 presents a service scalable architecture based on the serial IEEE 1394 bus. The architecture permits use of personal, mobile IP addressing. Furthermore, the functionality of several MHN Units is described. Finally, a hardware/Java software co-design methodology is demonstrated, using the Demultiplexer & Protocol Module as a case.

Common Interface Modules

A number of Common Interface Modules, intended for 1st and 2nd generation IRDs are proposed.

References

- Booch, G. Object-oriented Analyses and Design. Addison-Westly, 1998.
- Downing, T B. Java RMI: Remote Method Invocation. IDG Books Worldwide, 1998.
www.idgbooks.com
- Harmon, T. Using RMI in the Real World. JavaPro, November 1998, Vol.2, No. 6.
www.java-pro.com
- RMI. JDK documentation,
<http://java.sun.com/products/jdk/1.1/docs/guide/rmi/getstart.doc.html>
- Wirfs-Brock, A. Complex Java Applications. Article in JavaReport, Jan99.
www.javareport.com
- Haskel, B G, Puri, A, Netravali, A N. Digital Video: An introduction to MPEG-2. Chapman & Hall, 1997.
- Halfhill, T R. How to soup up Java. Article in Byte, May 1998.
- Cox, NN. Stochastic processes.
- P3. Håndbok for produktutvikling Del 1: Prosess i praksis. Telenor Intranet, not public.
- Sintef. Sintef-gruppens Kvalitetshåndbok. Sintef Intranet, not public.
- IEEE 1394-95
- Perry, D. VHDL. McGraw-Hill, 1998.
- Ashenden, P. The Designer's Guide to VHDL. Morgan Kaufman Publishers, 1995.
- Glaeser, G. Fast Algorithms for 3D-Graphics. Springer-Verlag, 1994
- 3D Studio. Inside 3D Studio, Release 4. S D Elliot, P L Miller. New Riders Publishing 1995.
- Rønningen, L A. Dr.Ing. thesis. 'Performance Evaluation in Design of Distributed Control for Switching Systems'. December 1982. ELAB report STF44 A83115.
- Rønningen, L A, ELAB. 'BUDOS and BUDOS INTERFACE Models. Technical Description'. January 1984. ELAB report STF44 F84120.
- Rønningen, L A, Stol, N. Modelling, implementation and use of a traffic simulator for NPDN. Paper presented at the Nordic Teletraffic Seminar 1984.

- Rønningen, L A. The combined digital satellite broadcast and Internet system. Telenor Satellite Services 1999.
- Rønningen, L A. A C-program for generating dependent stochastic variables. 1998. Not published.
- Birtwistle, G M. 'Discrete Event Modelling on Simula'. Macmillian 1979.
- Birtwistle, G M, O J Dahl, B Myhrhaug, K Nygaard. 'Simula Begin'. Studentlitteratur, Lund 1973.
- Nygaard, K, Dahl, O-J. 'The development of the SIMULA languages'. ACM Conference 'A history of programming languages', Los Angeles, June 1978.
- Luotonen, A. 'Web proxy servers'. Prentice Hall PTR 1998.
- Flanagan, D. 'Java in a nutshell'. O'Reilly 1997.
- Danesh, A. 'Internet med Java og JavaScript. Prentice Hall 1996.
- Huitema, C. 'IPv6 - The New Internet Protocol'. Prentice Hall PTR 1998.
- Thomas, S A. 'IPng and the TCP/IP Protocols'. Wiley Computer Publishing, Wiley & Sons, Inc 1996.
- Gai, S. 'Internetworking Ipv6 with Cisco Routers'. McGraw-Hill Series on Computer Communications. 1998.
- Siyan, K S. 'Inside TCP/IP'. New Riders 1997.
- Stern, J, Letteri, R. 'QuickTime and MoviePlayer Pro for Windows & Macintosh'. Peachpit Press 1998.
- Tanenbaum, A, Woodhull, A. 'Operating Systems. Design and Implementation'. Prentice-Hall 1997.
- Tanenbaum, A. 'Computer Networks' Prentice-Hall 1996.
- 'NorDig I. Digital Integrated Receiver Decoder Specification, for use in cable, staellite and terrestrial networks'. NorDig 1998.
- 'DAVIC 1.4 Specification. Application level software architecture'. Rev 1.1, 1998.
- Bliksås, K, maXware. 'Multimedia Channel Turbo Internet. Satellite communication protocol. Specification'. November 1997.

Ringbakken, S, maXware. 'Multimedia Channel Turbo Internet. End-user system architecture description.' Oct 97.

Ringbakken, S, maXware. 'Multimedia Channel Turbo Internet. Packetizer ARP update mechanism.' Nov 97.

Hovde, P, maXware. 'Multimedia Channel Turbo Internet. User Requirements Specification' Oct 97.

Hovde, P, Bliksås, K, maXware. 'Multimedia Channel Turbo Internet. Telenor DSM-CC Packetizer. Software Functional Specification'. July 97.

Hovde, P, maXware. 'Multimedia Channel Turbo Internet. Telenor DSM-CC Data Carousel. Software Architectural Design'. September 97.

Ringbakken, S, maXware. 'Multimedia Channel Turbo Internet. DVB/MPEG-2 satellite receiver PC card device driver'. 1997.

DSD - Nytt CD-format fra Sony/Philips. Lyd & Bilde nr. 4/98.

Conax Web page, <http://www.conax.com>

Petajan, E. 'The HDTV Grand Alliance System'. Proceedings of the IEEE, vol 83, No.7, Jul1995.

RFCs: can be found on <ftp.ripe.net/rfc>

RFC 1122

RFC 792

RFC 1883

RFC xx

RFC 1583

RFC 1587

RFC 1971

RFC 2131

RFC 1970

RFC 2068

RFC 1945 Hypertext Transfer Protocol - HTTP/1.0. <http://www.ripe.net/rfc>

RFC 2068 Hypertext Transfer Protocol - HTTP/1.1. <http://www.ripe.net/rfc>

W3Cs: WWW Consortium. <http://www.W3C.org>

prEN50221	Common Interface for Conditional Access and other Digital Video Broadcasting Decoder Applications
DVB TM 1324	Digital Video Broadcasting (DVB): Guidelines on implementation and usage of Service Information
ETS 300 421	Digital broadcasting systems for television, sound and data services: Framing structure, channel coding and modulation for 11/12 GHz Satellite services
ETS 300 468	Digital broadcasting systems for television, sound and data services: Specification for Service Information (SI) in Digital Video Broadcasting (DVB) systems (Edition 2)
ETS 300 472	Specification for conveying ITU-R System B Teletext in Digital Video Broadcast (DVB) bitstreams
DVB TM 1244	Final report from the Conditional Access Specialist group
PCMCIA	PC Card Standard, release 2.1. Personal Computer Memory Card International Association
A 011	DVB Common Scrambling Algorithm. DVB Blue Book A011.
EN 50 083-9	Interfaces for CATV/SMATV Headends and Similar Professional Equipment. Proposed European Norm prEN 50 083-9, Comité Européen de Normalization Electrotechnique CENELEC, January 1996.
EN 50 201	Interfaces for DVB-IRD. Proposed European Norm prEN 50 201, Comité Européen de Normalization Electrotechnique CENELEC, June 1996.
EN 50 221	Common Interface Specification for Conditional Access and other Digital Video Broadcasting Decoder Applications. Proposed European Norm prEN 50 221, Comité Européen de Normalization Electrotechnique CENELEC, June 1996.
ETR 154	Digital Video Broadcasting (DVB); Implementation Guidelines for the Use of MPEG-2 Systems, Video and Audio in Satellite Cable and Terrestrial Broadcasting Applications. European Telecommunication Report ETR

- 154, European Telecommunications Standards Institute ETSI, January 1996.
- ETR 162 Digital Broadcasting Systems for Television, Sound and Data Services; Allocation of Service Information (SI) Codes for Digital Video Broadcasting (DVB) Systems. European Telecommunication Report ETR 162, European Telecommunications Standards Institute ETSI, October 1995.
- ETR 211 Implementations Guidelines for DVB Service Information.
- ETR 289 Digital Video Broadcasting; The Common Scrambling System. Draft European Telecommunication Report ETR 289, European Telecommunications Standards Institute ETSI, May 1996.
- ETR 290 Measurement Guidelines for DVB Systems. Draft European Telecommunication Report ETR 290, European Telecommunications Standards Institute ETSI, May 1996.
- ETS 300 421 Digital Broadcasting Systems for Television, Sound and Data Services; Framing Structure, Channel Coding and Modulation for 11/12 GHz Satellite Services. European Telecommunication Standard ETS 300 421, European Telecommunications Standards Institute ETSI, December 1994.
- ETS 300 429 Digital Broadcasting Systems for Television, Sound and Data Services; Framing Structure, Channel Coding and Modulation for Cable Systems. European Telecommunication Standard ETS 300 429, European Telecommunications Standards Institute ETSI, December 1994.
- ETS 300 468 Edition 2 Digital Broadcasting Systems for Television, Sound and Data Services; Specification for Service Information (SI) in Digital Video Broadcasting (DVB) Systems. European Telecommunication Standard ETS 300 468, European Telecommunications Standards Institute ETSI, May 1996.
- ETS 300 472 Digital Broadcasting Systems for Television, Sound and Data Services; Specification for Conveying ITU-R System B Teletext in Digital Video Broadcasting (DVB) Bitstreams. European Telecommunication Standard ETS 300 472, European Telecommunications Standards Institute ETSI, May 1995.
- ETS 300 473 Digital Broadcasting Systems for Television, Sound and Data Services; Satellite Master Antenna Television (SMATV) Distribution Systems. European Telecommunication Standard

- ETS 300 473, European Telecommunications Standards Institute ETSI, May 1995.
- ETS 300 743 Digital Video Broadcasting (DVB); DVB Subtitling System. Draft European Telecommunication Standard prETS 300 743, European Telecommunications Standards Institute ETSI, May 1996.
- ETS 300 744 Digital Broadcasting Systems for Television, Sound and Data Services; Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television. Draft European Telecommunication Standard prETS 300 744, European Telecommunications Standards Institute ETSI, May 1996.
- ETS 300 748 Digital Broadcasting Systems for Television, Sound and Data Services; Framing Structure, Channel Coding and Modulation for Multipoint Video Distribution Systems at 10 GHz and above. Draft European Telecommunication Standard prETS 300 748, European Telecommunications Standards Institute ETSI, May 1996.
- ETS 300 749 Digital Broadcasting Systems for Television, Sound and Data Services; Framing Structure, Channel Coding and Modulation for Multipoint Video Distribution Systems below 10 GHz. Draft European Telecommunication Standard prETS 300 749, European Telecommunications Standards Institute ETSI, May 1996.
- ISO 13818 - 1 Information Technology - Generic Coding of Moving Pictures and Associated Audio Information. Part 1: Systems. ISO/IEC International Standard IS 13818, November 1994.
- ISO 13818 - 2 Information Technology - Generic Coding of Moving Pictures and Associated Audio Information. Part 2: Video. ISO/IEC International Standard IS 13818, November 1994.
- ISO 13818 - 3 Information Technology - Generic Coding of Moving Pictures and Associated Audio Information. Part 3: Audio. ISO/IEC International Standard IS 13818, November 1994.
- ISO 13818 - 6 Information Technology - Generic Coding of Moving Pictures and Associated Audio Information. Part 6: Extension for Digital Storage Media Command and Control. ISO/IEC International Standard IS 13818, November 1994.
- ISO/IEC 13522-5 Information Technology - Coding of multimedia and

hypermedia information-Part 5: Support for base-level applications.

- TM 1582 Draft Specification for DVB Interaction Channel through PSTN/ISDN. Internal Document of the Technical Module TM of the European DVB Project TM 1582, rev. 4, May 1996.
- TM 1594 Draft Specification - Network Independent Protocols for Interactive Services. Draft M. Internal Document of the Technical Module TM of the European DVB Project TM 1594, rev. 1, May 1996.
- TM 1631 Guidelines for the Use of DVB-SIS Specification - Network Independent Protocols for Interactive Services. Draft D. Internal Document of the Technical Module of the European DVB Project TM 1631, rev. 1, May 1996.
- TM 1640 Draft Specification for DVB Interaction Channel for CATV. Internal Document of the Technical Module TM of the European DVB Project TM 1640, rev. 4, May 1996.
- TM 1659 Draft Specification for Data Broadcasting in Digital Video Broadcasting (DVB) Systems. Internal Document of the Technical Module TM of the European DVB Project TM 1659, May 1996.

List of abbreviations

DVB	Digital Video Broadcasting
DVB-C	DVB-Cable
DVB-S	DVB-Satellite
DVB-T	DVB-Terrestrial
ETS	European Telecommunication Standard
FEC	Forward Error Correction
MPEG	Moving Picture Experts Group
PID	Program Identifier
CA	Conditional Access
CRC	Cyclic Redundancy Check
EBU	European Broadcasting Union
EIT	Event Information Table
EPG	Electronic Programme Guide
ETSI	European Telecommunications Standards Institute
IEC	International Electrotechnical Commission
IRD	Integrated Receiver Decoder
ISO	International Organisation for Standardisation

JTC	Joint Technical Committee
NVOD	Near Video On Demand
PID	Packet IDentifier
PSTN	Public Switched Telephone Network

The Combined Digital Satellite Broadcast and Internet System – the future Network (2)

Introduction

This chapter starts with a short description of the existing TSS satellite TV network, including ‘Turbo Internet’ for Web browsing on PCs, and then gives an overview of the future network.

[Figure 1](#) shows the present principal digital interactive TV broadcast network architecture of Telenor Satellite Services, with focus on the Multimedia Home Platform ([MHP](#)), the Interaction Network (telephone network and Internet) and the corresponding transmit-end servers. Note that presently the MHP is either an IRD for normal TV, or a PC using ‘Turbo Internet’. The system provides subscription and pre-booked PPV TV services, NVOD included, Turbo Internet via satellite for fast access to Internet Web pages, and [Data Carousels](#) for cyclic broadcast of heavily used applications. Note that all services provided by Web servers/Internet (commerce, banking, education, information retrieval, etc.) become available to the user. On the other hand, content and information providers, and service and business organisations have got the same possibilities to present and sell their information or products to very large populations.

As seen in [Figure 1](#), the MHP (PC or IRD) is connected to an Internet Access Server (part of the ISP’s domain) through the telephone network (PSTN, ISDN or other). Standard Internet protocols, TCP/IP/PPP are used on top of standard modems (V22bis/V32bis, ISDN or other). The MHP Web browser that supports HTML ‘talks’ to Web servers by means of the [HTTP/1.0](#) client-server protocol. The MHPs in the TSS network (currently Nokia Mediamaster 9610S) supports the TCP/IP/PPP protocols and use [Conax](#) pre-booked PPV protocols (proposed to DVB as standard) towards the Conax PPV Server.

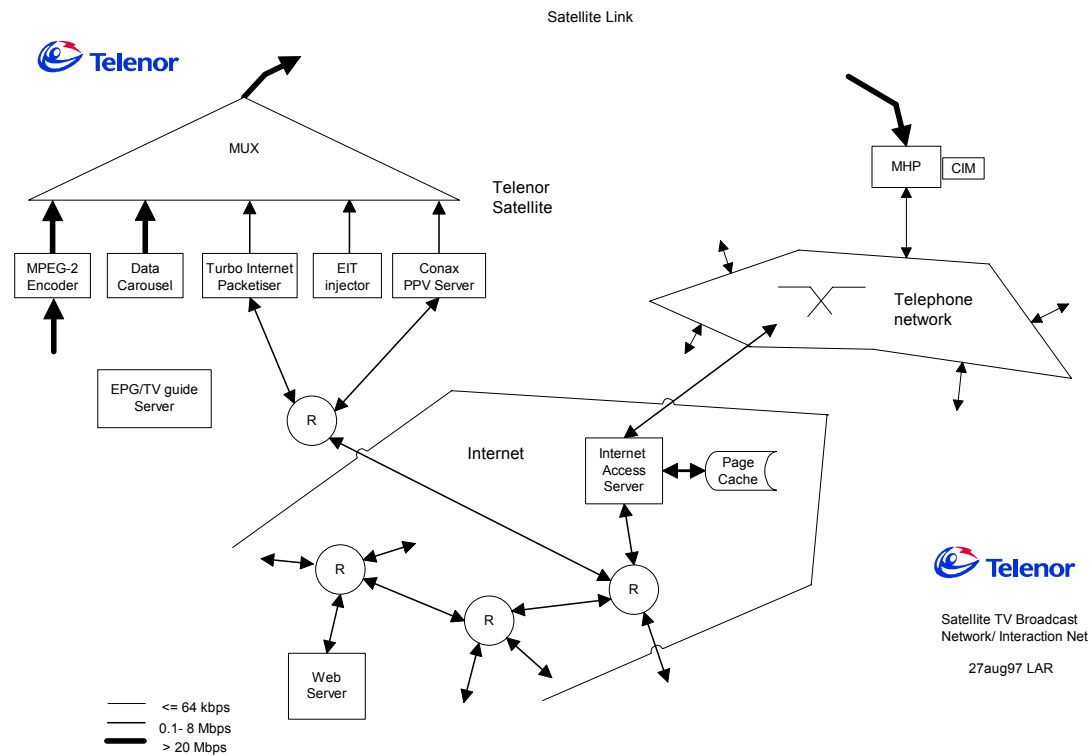
Turbo Internet requests from the user goes via the telephone network, while responses are routed via satellite. Functionally, the user will experience Turbo Internet just as terrestrial Internet, but with reduced delay. The ‘Turbo effect’ is obtained when high data rate per user all the way from where pages are cached to where the pages are shown is provided. [Figure 1](#) shows a special case of caching (to be extended), when the cache is part of the Internet Access Server. When Web pages are cached in the Access Server, the time delay for fetching pages becomes much smaller.

For the time being, payment for the Turbo Internet service is based on fixed subscription, independent of use.

On the broadcast link, the DVB [Multiprotocol Encapsulation](#) protocol is used under the TCP/IP protocols.

The DVB Carousel (DSM_CC based protocol) is used to cyclically output HTML pages, still pictures, video and audio clips, and other data. Seen from the user, applications from the Carousels are ‘offline interactive’ (like text TV).

The EPG/TV guide server is used to store [EIT](#) formatted data. The EIT Injector outputs the data cyclically to the multiplexer. This server is located close to the up-link site, and ‘Turbo effect’ can easily be obtained.



Abbreviations: MHP - Multimedia Home Platform (IRD)
 CIM - Common Interface Module
 R - Router

Figure 1. The Telenor Satellite digital TV and data broadcast platform. Functional description

For more details about the several subsystems shown in Figure 1, see [Conax](#), [W3C](#) and [maXware](#).

The future TSS network

[Figure 2](#) shows a proposal for the extended principal digital interactive multimedia/TV/data broadcast network architecture of Telenor Satellite Services (sub-systems will be explained in following sections). The figure shows only one up-link site, but several with same functionality (load-sharing and specialisation of content) will be placed in several countries in Europe, and connected together through Internet and satellite links. The network combines digital satellite broadcast, interactive TV and Internet/Web access in one network. The interaction network will be provided using an Intranet, either terrestrially or via satellite as shown in the Figure. Services and technical quality provided by the network shown in Figure 1 have been extensively enhanced. DVB broadcast standards, Web/Internet standards and new standard proposals are used.

Broadcast to cable networks or other terrestrial networks are not treated in this document, but are regarded as important services provided by TSS.

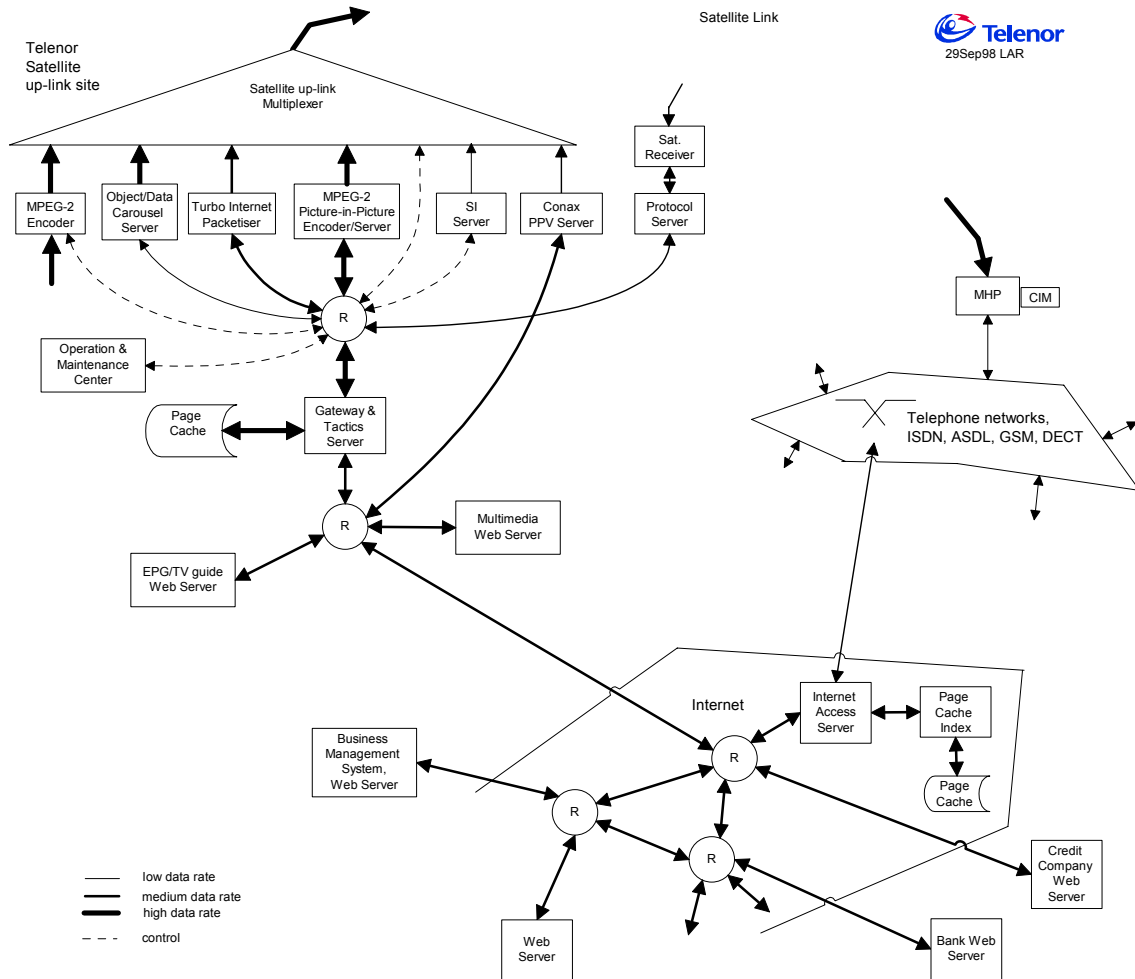
In order to perform services or sub-services, a number of subsystems have to play co-operative roles. This co-operation is treated in this chapter. Each subsystem is described in more detail in ([Rønningen99](#)).

Standard TV and HDTV, enhanced broadcast

Standard digital TV has been in operation for some time in several countries in Europe and America. All of these follow the [MPEG-2](#) standards. There are some differences in the RF part, Europe follows the DVB recommendations for both terrestrial, cable and satellite broadcast. The services include standard TV with video (mp@ml), sound, subtitling, text-TV, conditional access (CA), subscription, pay-per-view (PPV) and nearly-video-on-demand (NVOD)¹. The services also include a navigator with present/following EPG connected to pay-per-view. Several CA-systems are in use (Viaccess, Conax, etc.). The NorDig I specification specifies an IRD that includes the service repertoire referenced above. Some service operators also include an eight days EPG.

Each network operator (like TSS) has its own specification and rules of operation for the broadcast network.

¹ PPV and NVOD are defined as part of Enhanced Broadcast, even there are good arguments consider them as part of Interactive TV



Abbreviations:

- R router
- MHP Multimedia Home Platform
- GTS Gateway and Tactics Server
- ODC Object and Data Carousel
- TWS TSS Multimedia Web Server
- PIP MPEG-2 Picture-In-Picture Encoder/Server
- BMS Business Management System
- CIM Common Interface Module
- IAS Internet Access Server
- OMC Operation and Maintenance Center

Figure 2. Proposed future TSS platform for digital satellite broadcast and Web/Internet access.

USA has introduced HDTV in November 1998. The receivers are quite expensive, varying from about \$7,000 to about \$9,000 (Nov 98). The HDTV offer will be about five hours a week in ten of the most TV-successful cities. In a year or two, the price for the HDTV set is expected to get below \$2,000 for the cheapest sets.

A problem with the [GA-HDTV](#) (Grand Alliance) standard used in USA is the large number of variable parameters, which make it difficult to get the hardware cost down

For future [NorDig](#) standards, the proposal here is to limit the number of levels to three, and use ASICs for most functions:

- Spatial and temporal resolution
 - 1280(hor) x 720(vert) pels, 25 and 50 Hz progressive scan (Level I)
 - 1920 x 1080 pels, 25Hz (Level II), 50 Hz (Level III) progressive scan
- 16:9 aspect ratio
- 4:2:0 chrominance sampling
- 10 bits luminance and 10 bits chrominance resolution per pel
- I, P and B pictures
- Compressed data rate, 10-45Mbps

The specification shall include main profile @ main level, used for standard TV.

The audio shall support the Dolby C3 system. An extended [MPEG-2](#) 5.1 standard with 96 k samples per second, is optional.

Enhanced broadcast, interactive TV and Web/Internet access

The intention of TSS is to combine the basic TV service with interactivity and Web/Internet access. This can be done for both normal TV (NorDig I) and HDTV.

A new feature is the conversion of pictures and videos of any format in a Web page to parallel video MPEG-2 [PES](#) streams in real time at the satellite up-link site, and let the MHP handle the decoding in hardware (see next section).

Another new feature is to use object and data carousels in co-operation with up-link site caching of Web objects, in order to increase performance. The GTS sub-system handles this together with the ODC (Object and Data Carousel Server) sub-system.

Note that the TSS Web server, TWS, has a high-capacity link to GTS, which assures short response time for Web objects that are not cached by GTS.

The Internet access includes at least Web browsing, email and file transfer.

Parallel Picture-In-Picture (PIP) streams and Web access

The MPEG-2 standard permits several video PES streams, each occupying rectangles on the screen, to be transmitted in parallel. The receiver hardware shall be designed to decode this in addition to normal TV.

All Web objects routed via the broadcast channel shall be examined at the up-link site, and all pictures, videos and audio shall be converted to MPEG-2 PES before broadcast. The GTS (Gateway & Tactics Server) sub-system handles this together with the PIP (MPEG-2 Picture-in-Picture Encoder/Server) sub-system.

Both GTS and the TWS shall use MPEG-2 PES format for video and audio. GTS includes play-out equipment for MPEG-2 PES video, that can be accessed only by ‘turbo users’ (response always via satellite). The TWS can be accessed also through the terrestrial Internet.

Scaleable, Layer-structured multimedia applications

The basic idea behind scaleable and layer-structured multimedia applications is that they can run on all levels of Multimedia Home Platforms. The low-end digital TV receiver shows only the basic layer, while the high-end MHP can show all layers. The applications build upon a combined use of DVB carousels, Internet/Web access and MPEG-2 Picture in Picture playout systems. Applications can be written in several combinations of [HTML](#), [Java](#), [Javascript](#), [Quicktime](#), and others.

Typical layer-structured multimedia applications are:

A live TV program from studio, for example extended ‘Kontrapunkt’, where two teams (from two Nordic countries) compete on presenting details about musical compositions played to them, and one or more teams (established via Internet/TSS Web server) sit at home and take part in the competition using Interactive TV/Web. There may also be low-rate video channels from the homes to the studio, which may be shown in a window on the TV screen. All teams have the possibility to access Web to retrieve more information about the compositions etc. The progress/status of each team is presented as multimedia applications, controlled partly from the studio, partly by the home teams.

A live TV program from an event, say the European Song Contest where the singers, musicians and other actors are real, but the background is synthesised using virtual reality (VR) techniques. The background can be transmitted using vector representation and presented as graphics. This reduces the required data rate dramatically.

A movie - the viewers can influence the coming scenes (a number of pre-programmed choices) using a window (HTML ‘forms’) on the screen to input their choice.

A football match – the user may divide the screen between the main picture and several other pictures and text windows, showing associated or other information.

News - the user get the headline news live, and he can if he wants, fetch explanatory text and short videos from carousels, and even more details from Web, and show in separate windows on the screen.

Flash advertisements – A tour operator sells out the last aeroplane seats/hotel rooms at special prices via Web, first come first served.

Other advertisements designed for TSS network, using combinations of TSS play-out systems.

The TSS Web Mall, using different TSS servers in co-operation, see Chapter 7.

Multimedia applications for PC, fetched from Web/Internet, adapted for TSS network

Multimedia applications for PC, fetched from CD-ROM and Web/Internet, adapted for TSS network

Standard TV programmes as sent today.

Radio channels playing different type of music, with multimedia presentation of composers, compositions, singers, orchestras, etc.

Synchronisation of defined events in random time is required. The studio control can use the Web/Internet and/or the broadcast channel for synchronisation of events. A proposal is to use primitives defined by simulation programming languages like [Simula](#) and [Demos](#). More will be said about this in later sections.

The basic layer

In [Figure 3](#), the basic layer consists of a number of HTML pages only with text, and multi-dimensional hyperlinks to other pages within the same layer and layers above. The pages are played out from a DVB Carousel. There is no interaction channel, but the user can interact (click on a link symbol) with the MHP to select other pages from the Carousel (just like text TV). This is denoted ‘interactive offline’.

The basic TV control or Navigator, and standard digital TV programmes belong to this layer.

From the basic layer, the user can also activate the PPV (including NVOD) server (see Chapter 7) and order and pay for the wanted service.

With HTML 3.2, the look and feel of the text cannot be altered by the MHP, it is determined by the page itself. However, by linking in local MHP files with different backgrounds, the appearance can be personalised. The ‘style sheet’- feature of [HTML 4.0](#) enables the MHP to define its own look and feel of text, background and layout.

For existing (low-end) IRDs with text based EPGs (EIT or other format) it is simple to locally convert the HTML format into EIT, or an internal IRD format (download the converter using bootloader or module loader). The look and feel can be as before.

Layer 1

Layer 1 can be reached via links from the basic layer. This layer contains a number of HTML pages with text and JavaScript, and multi-dimensional links to other pages within the same layer and layers above and below. The text pages can be played out from a DVB Carousel, or since an interaction channel is available, can be fetched from Web/Internet. This is denoted 'interactive online'. This layer has got links to still pictures, video and audio clips, and applets.

Layer 2

From layer 1 still pictures and applets can be loaded automatically, or the user has to go to Layer 2 by selecting a link. The stills (files) can be stored on a Web server, or on a Carousel. Since video and audio clips require much more capacity than text and stills, the user that wants them has to click on their link symbols. Stills and videos can be resized and placed arbitrary by the user. Video and audio files are normally fetched from the Web, but very popular clips can also be put on the Carousel. The accepted still formats are JPEG, GIF, and MPEG-2. The video format shall be MPEG-2 and others as options.

Layer 3

On this layer, MPEG-2 still pictures, video and audio clips, and applets can be activated. The advanced PIP features as described above can be used. The user can resize and replace all picture windows. With HTML 4.0 the MHP screen look and feel can be determined locally (using style sheets).

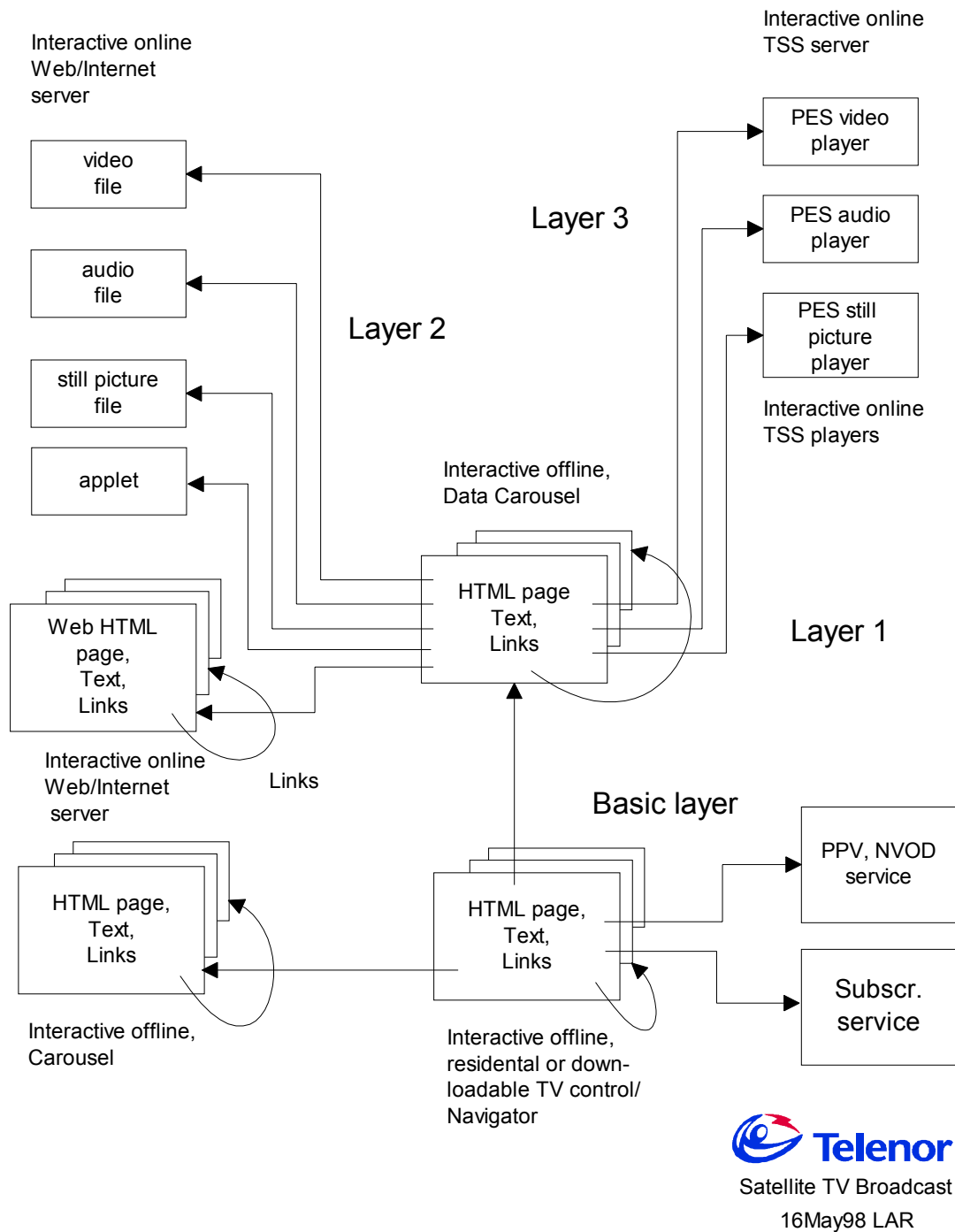


Figure 3. Layer-structured interactive multimedia applications, using carousel, Web/Internet servers and MPEG-2 PES play-out systems in co-operation

Synchronisation of Layer-structured Multimedia Plays

As pointed out in the previous section, the components of multimedia applications have to be synchronised. For synchronisation of Web/Internet multimedia components, several software support packages are available, e.g. [Quicktime](#) and [Java Media Framework](#). These tools support only parts of what we are looking for; a general tool for synchronisation of plays performed by geographically distributed actors using multimedia resources. Such synchronisation has always been practised in TV studios and theatres, but only with locally distributed actors, and considerable manual support.

Figure 4 shows a very general situation. In TV studio there are one studio control group and several studio acting groups. In addition, several remote acting groups participate. The remote acting groups may be formed of normal TV viewers, that 'log into' a group before start of programme (rules TBD). The remote acting group may also be a TV/multimedia team acting at geographically distributed sites, together with viewer groups. Each group has a range of multimedia resources available. Each group normally consists of persons, computers or other technical systems that have the ability of acting. Together, the acting groups perform a play, directed by the studio control group. During the play there are a lot of discrete and sometimes random, event points where synchronisation have to take place. The author's experience from discrete event modelling and simulation ([Ronningen82](#)) has shown that the following synchronisation events often are encountered, and may be adapted for our purpose:

- Resources (R) are produced by one group and consumed by other groups. Two primitives are needed,

give(R) The producing group puts the resource into a resource queue, and wakes up any waiting groups.

take(R) The consuming group tries to take a resource out of the resource queue. If the queue is empty, the group is put into a wait queue until a resource has been produced.

- Resources (R) stored in a pool are requested by a group. Two primitives are needed,

acquire(R) The group tries to get a resource. If no available, the group waits passively in a wait queue until a resource is released by another group.

release(R) A group releases a resource to the pool, and wakes up any waiting group.

- All groups have an input message queue, which all groups can put messages into. Three primitives are used,

wait (que) Messages wait passively in the queue until fetched

fetch(que) A group fetch the message from the queue, or wait passively until a message has arrived

find(que, m) A group search the queue for a message m, and leaves if m not found.

- Groups wait until a certain condition is fulfilled. The group that sets the condition, wakes up any waiting groups with a signal.

waituntil(que, cond) A group waits in a queue until the condition is fulfilled.

signal(que) A group sets the condition, and wakes up any waiting groups with signal.

- Groups are initially activated by the primitives *schedule(group)* and passivated with *cancel(group)*.
- The groups may use *sequencers*, which are devices that can output messages, set conditions and send signals according to a configured sequence. Timers and ‘tick clocks’ are special cases of this. The overall sequence of the play has to be controlled by the studio control group.
- Passive waiting can in general be avoided by using an *empty_queue* test before calling the synchronisation primitive.

Note that audio/visual perception of events are special cases of the above. Pressing a key or pointing to a hyperlink, etc, are also special cases of the above.

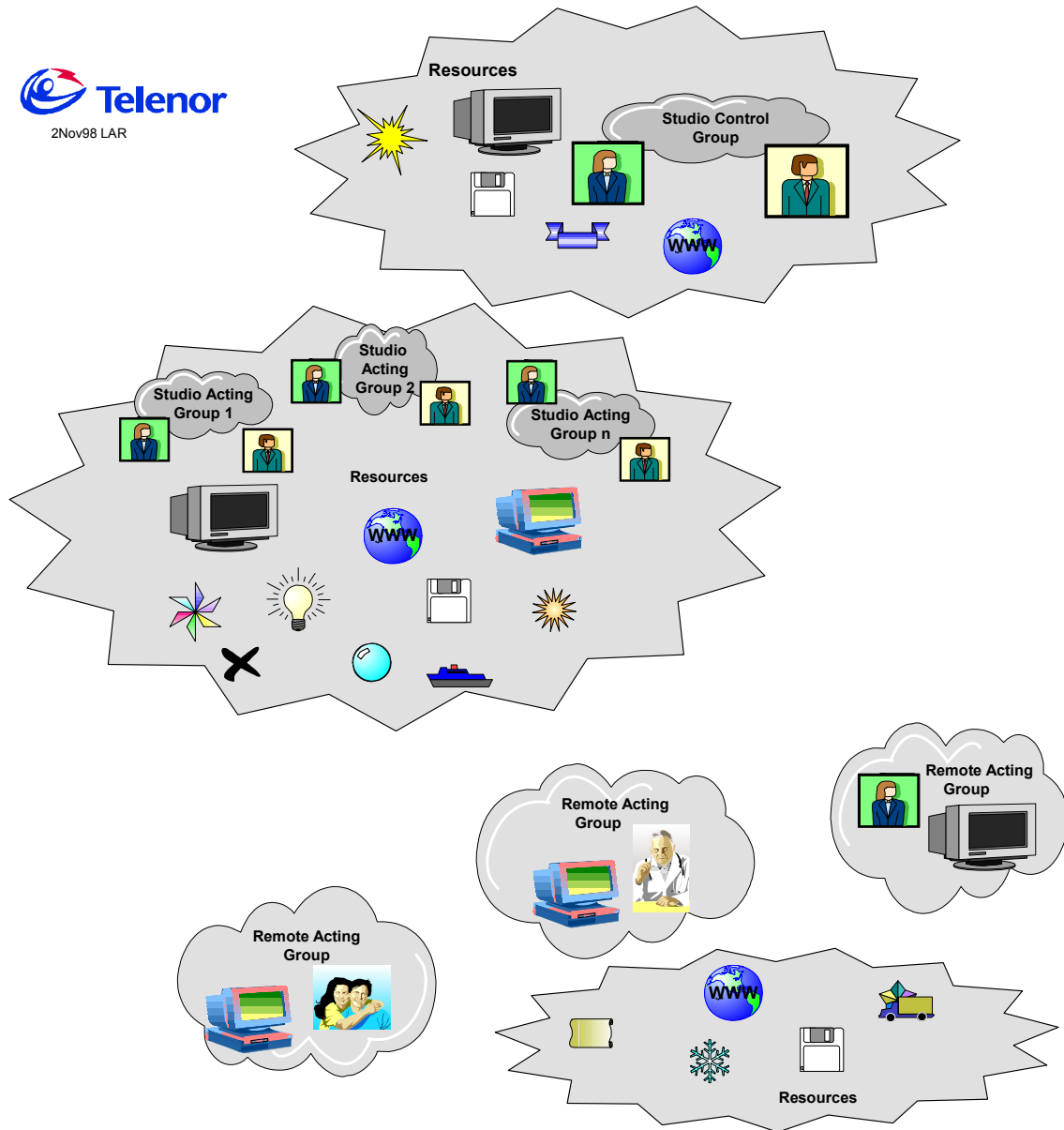


Figure 4. Multimedia control group, distributed acting groups and resources.

Games

Games are presently distributed via CDs for multimedia PCs, or via Web/Internet as downloadable software. In addition, there are special play machines in the market designed for games only (e.g., Nintendo 64 and Sony Playstation), and with private cassette or diskette solutions for storage of the game firmware. The PC may or may not have hardware support for 2D and 3D games, which result in varying and unsatisfactory performance. The play machines have very efficient hardware for 'playing around' with 2D and 3D objects, but when connected to standard PAL TV, the resolution is not impressive.

With digital TV this can be changed. It is possible to use progressive scan, and MPEG-2 video data rates of up to at least 15 Mbps for normal TV. When HDTV is introduced, both the data rate (10 - 45 Mbps) and the resolution can be substantially increased, giving excellent quality for graphics and games. Instead of placing the extra cost of advanced graphics processing on all viewers, plug-in modules or 'side-cars' to the set-top box or TV will be proposed. The Common Interface, with PCMCIA or IEEE 1394, is ideal for this purpose.

There are several options using Common Interface Modules (CIM) or IEEE 1394 Plug-ins:

- Modify existing play machines so they can output MPEG-2 with up to HTDV quality to Common Interface. The input from the user can be direct to the play machine or via the MHP.
- Build new play machines with MPEG-2 output, and that can download parallel MPEG-2 streams and other data streams from satellite (from TSS Web servers, Web in general (maybe cached) and from carousels). The input from the user can be direct to the play machine or via the MHP.
- Same as above, but using low-level transfer of processed graphics data from play machine into the Command Bus of Common Interface, and close co-operation between the MHP and CIM graphics processors.
- Other solutions

Commerce on digital TV/ Internet/ Web

Money in the near future will be bits and bytes in networks and databases. Money is leaving the physical world and entering virtual reality. Several potential problems, e.g. regarding fraud and copy-write protection, have to be addressed. But, the introduction of e-money gives new possibilities for adequate international value-measurement of products and services, and for example, automatic payment of value added taxes (my proclamation, to be followed up).

Separately, digital TV and Web provide a lot of possibilities for remote, electronic commerce. On the Web, several common frameworks for small and large firms called Web malls, have been in operation in USA for some time, with great success. See Chapter 7. Together, they are unrivalled (my proclamation), because the customers can be made aware of products and

services through flash presentations on TV, and then given the hyperlinks to the malls. The proposal is to apply existing and coming Web mall technology, secure Web payment technology, commerce on TV and conditional access schemes for TV. The commercial part of the TV services can be handled by the Business Management System, BMS, that in turn can be regarded as part of a TSS Web mall server. The mall server will be placed on TSS Web servers and carousels, and with hyperlinks to the general Web.

The basic TV services have for some time been offered as combinations of:

- Subscription
- Impulse and token Pay-Per-View (PPV)
- Free of charge – financed by commercials.

Invoices for payment of subscribed services are today sent by post to the users. But in addition, payment can already now be made using secure Web technology. Both the marketing, ordering and payment can be included in a Telenor Web Mall. Below, this is described for PPV.

Pre-booked PPV and token PPV

With pre-booked PPV the users can buy TV programmes online (from BMS) in advance. Most users buy just before start of programme (Canal+), and all sub-systems have to be dimensioned to handle the peak traffic. An advantage is that the operator immediately can get figures of the popularity of programmes

With the extended token PPV, tokens or coins can be withdrawn from the users bank account and deposited in the user's smart-card in advance (see below). The time to receive service becomes short, and the user discounts his smart card as he views programmes. Payment can be made to BMS on request when an agreed credit and/or time limit is reached. An alternative to credit is the 'prepaid' scheme. These schemes effectively spread the extremes of the interaction traffic, which in turn reduce the need for equipment investment. The viewing rate for each programme can be estimated by storing the viewing history and using classical statistics on the data. The accumulated data can be transferred to the service operator in low-traffic periods. Transactions are secured using authentication and encryption.

Payment on Web/Internet

Internet access is normally offered as subscription by an access provider, with or without limited access time per month. Content and service operators handle the commercial process. This process can be implemented as multimedia presentations/ searches/ordering (using HTML 'forms') combined with different payment schemes, placed on the TSS Web mall servers.

Service, content and other product providers can use the system to market any product, tangibles and non-tangibles. This is similar to selling TV services. The TSS Web mall server shall handle the payment.

Three different categories for electronic payment on Web are used:

'The credit card category':

The customer validates the purchase by sending his encrypted credit card number and details about the payment electronically to the vendor. The vendor forwards an electronic invoice to the credit-card company, who can decrypt the credit card number, and send a secured electronic invoice to the user. Then the user pays and gets a receipt, all electronically. Note that the credit-card company controls the transactions, but the vendor or a third person does not get hold of the credit card number.

'The cheque category':

A person X sends a secure electronic cheque to another person Y, and informs his bank about the transaction. Y can then cash the check or make an electronic transfer to his bank account. The bank controls the transactions.

'The cash category':

A person X withdraw some tokens (cash money) from his bank account and deposit in his electronic purse/smart-card using secure electronic transfer. When a bargain is closed between X and another person Y, X sends secure tokens to Y. Y may deposit the tokens in his bank account, or use them in a bargain with a company Z. This is very much the same as using physical cash. The advantage of the cash scheme is that the transactions only can be traced if someone tries using the tokens more than once.

Multimedia Home Platform/ Network

The MHP shall support Enhanced Broadcast, Interactive TV and Web/Internet access, that is, multimedia services of any layer-structured combination as described above. In the future, the MHP will be extended/interpreted as a Multimedia Home Network, as described in later chapters. The basic TV control/Navigator is also a Web browser that sends requests to Web Servers terrestrially, using [HTTP/1.1/TCP/IPv6/PPP](#) as described in later sections, and receives responses via satellite (selected by user) by means of the HTTP/1.1/IPv6/[MPE](#)/MPEG-2 protocols. Several TV or multimedia PIP MPEG-2 PES streams can be synchronised with the browser actions. The MHP shall decode PIP MPEG-2 PES in hardware. The services have to be signalled in SI.

Together with fast Web/Internet via the broadcast channel and the advantages of broadcast carousels, the TV set becomes a real contender to the PC (performance and price).

Business Management System – BMS

For the time being, the role of BMS includes only that of the SMS (subscriber management system) operators. BMS functions beyond the level of SMS are defined in ([Rønningen99](#)).

With existing practice, the user must have direct business relationship with SMS operators, local ISPs and/or local PSTN-providers and other companies, to search into, order and pay for use of different TV channels, telephone, Internet access and products in general.

To maximise user-friendliness, it is important to reduce the number of user/business access points (for ordering, payment, info, etc.) to the combined TSS system. The user shall receive *one* bill for all sub-services, however with a hierarchical specification of used resources and corresponding payment.

The functionality in BMS can be based on Web server technology and extended to handle the combined user/business access point. This involves implementation of TCP/IP_{v6} and HTTP/1.1 protocols. BMS can also be regarded as part of the TSS Web server system.

The user shall have the choice if he wants to interact with BMS using a Web browser, the telephone with 'push-a-number-guide' and automatic answering, the telephone and conversation with a 'customer care person', or by visiting a customer care desk in Telenor's or representative's retail shops.

The payment system of services shall be based on combinations of fixed monthly fees, resource usage fees (for processing, storage and transport), and acceptance of advertisements. If the user do not want to see advertisements, he pays maximum price for the service. In the other end, the service is free of charge if the user accepts to receive all sent commercials. The user shall also have to possibility to choose something in between.

The resource usage shall be registered in each actual sub-system, and reported to BMS regularly.

The BMS organisation shall also take care of marketing of services, in co-operation with the service providers involved.

Security

Security is concerned about how to protect a system against any type of attack. Cryptographic systems are an important part of security. Normally, it is very difficult to build a system that can withstand every type of attack.

The experience has shown that secure algorithms are secure only for a certain period of time. Both symmetric encryption algorithms and public-key crypto systems can be attacked systematically, and have been broken. The consequence is that security systems should be substitutable, low-cost plug-in modules

Conditional Access is applied to assure that only those who are permitted for a service get access. Authentication is used to assure that the source of messages etc., has the expected identity. Encryption is used to make content readable only for wanted persons. Encryption also prevents unwanted persons to change the content of messages.

Ari [Luotonen](#) has described several solutions of firewalls. The simplest is a single router that can filter out unwanted IP packets, and a more sophisticated solution consist of a proxy with a router on each side. The latter, applied in Figure 2, gives security both on network, transport and application levels.

The cryptographic schemes used in addition to standard TV conditional access in the TSS network, are Ipv6 authentication and encryption.

Routing and Caching schemes

The present “Turbo Internet” see Chapter 2, uses IPv4 for addressing and routing, and the DVB Multiprotocol Encapsulation for transport via satellite. Dynamic allocation of user (source) IP addresses is used, due to shortage of available addresses. There is no caching in the system, so the “Turbo effect” is obtained only for Web pages stored in Telenor Nextel’s Web server.

In this document, the new IPv6 protocol with extended IP address lengths (128 bits) is applied. Each user and subsystem can be allocated several fixed IP addresses, and it is shown how controlled routes for requests (terrestrially) and responses (via satellite) can be realised using existing features of IPv6.

HTTP/1.1 has got a lot of support for caching, which will be used for satellite up-link caching, together with controlled routing as mentioned above. A new feature described in this document is the tactical use of up-link caching and carousels in combination. The scheme is based on estimation of traffic and standard caching schemes in Internet, and when a URL object requested by a large number of users simultaneously, the object is put on a carousel and broadcast to their user group.

Data Broadcast Carousels and protocols

DVB Data and Object Carousels

The DVB Data and Object Carousels are specified in [TM1659](#) Rev 5. NorDig specifies a one-layer data carousel for MHP software bootloading. The data carousel cyclically plays out data structures, e.g., files, organised in a one-level structure. When the data structure has IPv6 packet format, then authentication, message encryption and message integrity can be handled as defined in Chapter 4 of this document.

An object carousel is included in the DVB MHP API specification (not finalised). The object carousel cyclically plays out objects, structured as e.g., a hierarchical file catalogue structure,

streaming data or data organised as described in [TM 1659](#). The objects can also be MPEG-2 PES packets, UDP/IPv6 or IPv6 packets. When the objects contain IPv6 packets, then authentication, message encryption and message integrity can be handled as defined in Chapter 4 of this document. Conditional access for PES packets can be handled as for normal TV.

DVB Multiprotocol Encapsulation - MPE

The TSS implementation of DVB Multiprotocol Encapsulation, specified by DVB in [TM 1659](#), is documented by [maXware](#), and supports IP packets. In the steady state, only one MPE packet is sent for each IPv6 packet. The encapsulation involves translation of 128 bits IPv6 addresses into 48 bits MAC addresses. When the payload is IPv6 packets, then authentication, message encryption and message integrity can be handled as defined in Chapter 4 of this document.

The TCP/ IPv6 protocol stack

TCP is an end-to-end, client - server, transport protocol, providing window flow- and error control. Even if TCP often is used together with IP, it is a stand-alone protocol. TCP is used in Internet and other systems. The UDP protocol is the unreliable 'twin' to TCP. It is a datagram protocol, without flow- or error control, but it's 'ports' can be used for sub-addressing of applications in MHP.

IPv6, the Internet Protocol version 6, is a protocol intended for routing and addressing of datagrams. No flow- or error control is supported. The most important improvements over version 4, are the increased IP address space from 32 to 128 bits, the performance enhancement and the caching support.

The extended IP address space permits each user to have several IP addresses. This is convenient for individual MHPs and broadcast to groups of MHPs: Use fixed IP addresses, one for terrestrial internet access, one for satellite transmission and several broadcast group IP addresses. Since the satellite channel is (for the time being) one-way for responses, requests have to be sent terrestrially.

The limitation of TCP via satellite

As shown in ([Rønningen99](#)), the window flow control mechanism, together with propagation and processing times, limits the average data rate per TCP connection via satellite to about 400 kbps.

Transfer of Audio-Visual content using UDP

Audio-visual streams can be transferred via satellite using UDP. The scheme is described in ([Rønningen99](#)). The actual transfer rate is limited by the satellite channel and the receiver capabilities.

Note that the UDP layer can be omitted using IPv6 ‘flow’ directly. See (Rønningen99).

The HTTP1.1 Web protocol

The HTTP/1.1 protocol is documented in [RFC 2068](#) titled “Hypertext Transfer Protocol – HTTP/1.1”.

Since the HTTP/1.1 protocol is applied in an extended mode for caching and transfer to carousels in this document, parts of it are described in detail.

The HTTP/1.1 protocol is quite new and has been introduced by several vendors in their Internet/Web equipment, but is not yet widespread. There are several good reasons for using version 1.1 (and not 1.0) in the TSS network: The mechanisms for caching control, and that each connection (normally TCP) permits pipelined HTTP/1.1 requests.

HTTP/1.0 is currently the most widespread protocol for transferring Web documents.

See Chapter 4 in ([Rønningen99](#)) for application of HTTP/1.1 in the TSS network.

The main characteristics are:

HTTP/1.1 is an ASCII text based request/response protocol. The client sends a request to the server, and the server sends back a response. Most often, HTTP/1.1 uses a TCP/IP connection through a network of routers and proxy servers, between the client and server.

Typical HTTP/1.1 requests and responses go like this:

A Web browser client (type Mozilla/4.0) requests the TSS Web Server for an HTML page. Images of type gif and jpeg are accepted. Note that the destination IP address (www.tssws.no) is part of the URL.

```
GET http://www.tssws.no/mydir/mypage.html HTTP/1.1
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
```

The server (type Netscape,,,) sends back a HTML page of length 3417. The status-code 200 means Ok. The explaining text ‘Ok’ follows.

```
HTTP/1.1 200 Ok
Server: Netscape-Enterprise/3.0
Date: Sun, 5 Jun 1998 07:33:24 GMT
Content-type: text/html
Content-length: 3417
```

The HTTP/1.1 specification defines an number of different headers, that describe the content of request and resource entities (content), respectively. Headers important for caching are treated in ([Rønningen99](#)).

Remote downloading of software

In this section the focus is on issues of the actual downloading of software from one sub-system to another. It is assumed that the quality of the software itself is controlled before the actual download takes place.

The main issues of downloading of software are about security and versions. Software is normally structured in a hierarchy of modules, and the configurations of versions of modules shall have a well-defined behaviour. The set of allowable configurations of software shall be approved a priori.

The download of software into sub-systems (MHPs included) shall be controlled from OMS. OMS shall maintain a database on existing configurations, configuring history and approved configurations for all sub-systems. Each sub-system shall also itself maintain existing and approved software and hardware configurations.

Before a download of software modules takes place, the configuration shall be checked by OMC. When the modules are received by the sub-system the configuration shall be approved by the sub-system, and a status report shall be sent to OMC. Not all MHPs shall do this, only a selection (TBD). Some MHPs will not perform download they are expected to. If a user have problems with his MHP, it shall be possible to request the MHP via Internet to send a copy of the current configuration to OMC. It shall also be possible to download all software via Internet, if the user accepts this. [See NorDig I.](#)

To prevent that unwanted software modules are downloaded, authentication shall be applied (e.g. Ipv6 authentication). In some cases, the content/payload can be encrypted for privacy. Only authorised personnel shall have access to OMC download equipment (password, physical secured equipment

Part B

Development in phases (3)

Introduction to development methodology

Development of information and communication technology products is usually carried out in phases, starting out from more or less accurate user requirements. In the design phase the decision to use software, hardware or a combination is taken. The methodologies for hardware and software development are very well described in numerous textbooks, and normally the following phases and module tasks are involved:

- | | | |
|----------------------|-------------|---------|
| 1. User requirements | U1, U2, ,Um | |
| 2. Specification | S1, S2, ,Sn | modules |
| 3. Design | D1, D2, ,Dp | “ |
| 4. Implementation | I1, I2, ,Iq | “ |
| 5. Test | T1, T2, ,Tr | “ |

(m, n, p, q, r – integers)

In each phase parallel and serial hierarchically arranged tasks are carried out. Normally, the quality assurance system, QA (see ISO9000 series) (Sintef), requires approval of one phase before the next is started. There are normally dependencies between tasks, that is, one task has to be carried out before another can be started. In the planning process, the critical path is determined, however, the estimated times for each task in the critical path are normally considered independent and just summed up.

In most cases it is not advantageous to develop a new system from scratch, but rather buy a system or at least reuse parts of existing systems.

The management and realisation of a development project involve technical, economic, time, human, social and organisational issues. Industrialisation of the results should also be considered. Development projects may be considered as a special case of innovation (West). All these issues are very well described in the literature, see for example (P3), and will not be repeated here. However, the method of phases, which is widely accepted, will be scrutinised in the last section of this chapter.

Design as one phase in the development process

Design is part of the development process, and may consist of the two sub-phases functional description² and architectural description. The functional description consists of functional structures (of blocks) and behavioural descriptions, but yet nothing is said about the realisation. The architectural description involves the hardware/software trade-off and describes the behaviour of hardware and software structures and modules.

² The functional description may also be considered as part of a specification phase. This is a matter of definition.

Discussion on phases

The discussion carried out here is based on the authors own experience over the last two decades of development projects. Formally, it should be considered as a set of not-tested hypotheses.

Development is usually carried out in phases as outlined above. The first question to ask is why are phases used? One answer is probably that this gives good overview of the development process, it makes it manageable. Who needs the phases? Certainly project managers, maybe project members, the firm/organisation management, the economy-, purchasing-, marketing- and production departments. The problem is that the phased model in many cases does not give a particular good description of how a project actually is carried out.

There are certainly a lot of information and communication technology development projects that can be considered as ‘carpentry’. The resource, time and cost estimates can be based on earlier experience, and the risk for not following plans is reasonable. But for basic research projects the phased model does not fit very well. The more research oriented, the more difficult it is to plan the project. The number of estimated hours and costs are normally exceeded, and the expected goals are uncertain.

The first step should be to take the time dependency between project tasks into account. In ([Rønningen88](#)) such a model is developed. The critical path task times are modelled as stochastic uniformly distributed variables. When the task time is uncertain, the minimum and maximum time should have a ratio of 1:3, 1:10 or even larger. Normally, the number of tasks in the critical path is large, and if the individual task times are just summed up, the resulting sum approaches normally distributed (the central limit theorem) with a small variance. What we then have lost, is the dependency between tasks. The experience is that if the complexity of one task is underestimated, then the other tasks are underestimated as well. This can be taken care of by introducing a covariance matrix that gives the dependencies between tasks. The result of this, as verified by simulation in ([Rønningen88](#)), is that the central limit theorem is not valid when strong dependencies are introduced. If a large number of uniformly distributed task times are summed, and there is strong positive correlation between times, the sum also becomes uniformly distributed, and the variance of the sum becomes large.

Another observation is that advanced research projects involve a large number of iterations within phases and between phases. The iterations in the critical path can be modelled as shown in Figure 5. The time in each phase will be stochastic. As a first approximation, the figure can represent a homogeneous Markov chain, where the ‘bubbles’ represent the finite number of states, and the arrows the transition between states. The time scale is assumed discrete. In practice, it is probably of no value to use a smaller time unit than one day. Furthermore, the Markov chain is assumed to possess a stationary transition mechanism, and all states are aperiodic, irreducible and recurrent. With the transition probabilities as the input to the model, the state probabilities, and the distribution and moments for the passage time from state ‘short user requirements’ to state ‘end’ can be calculated.

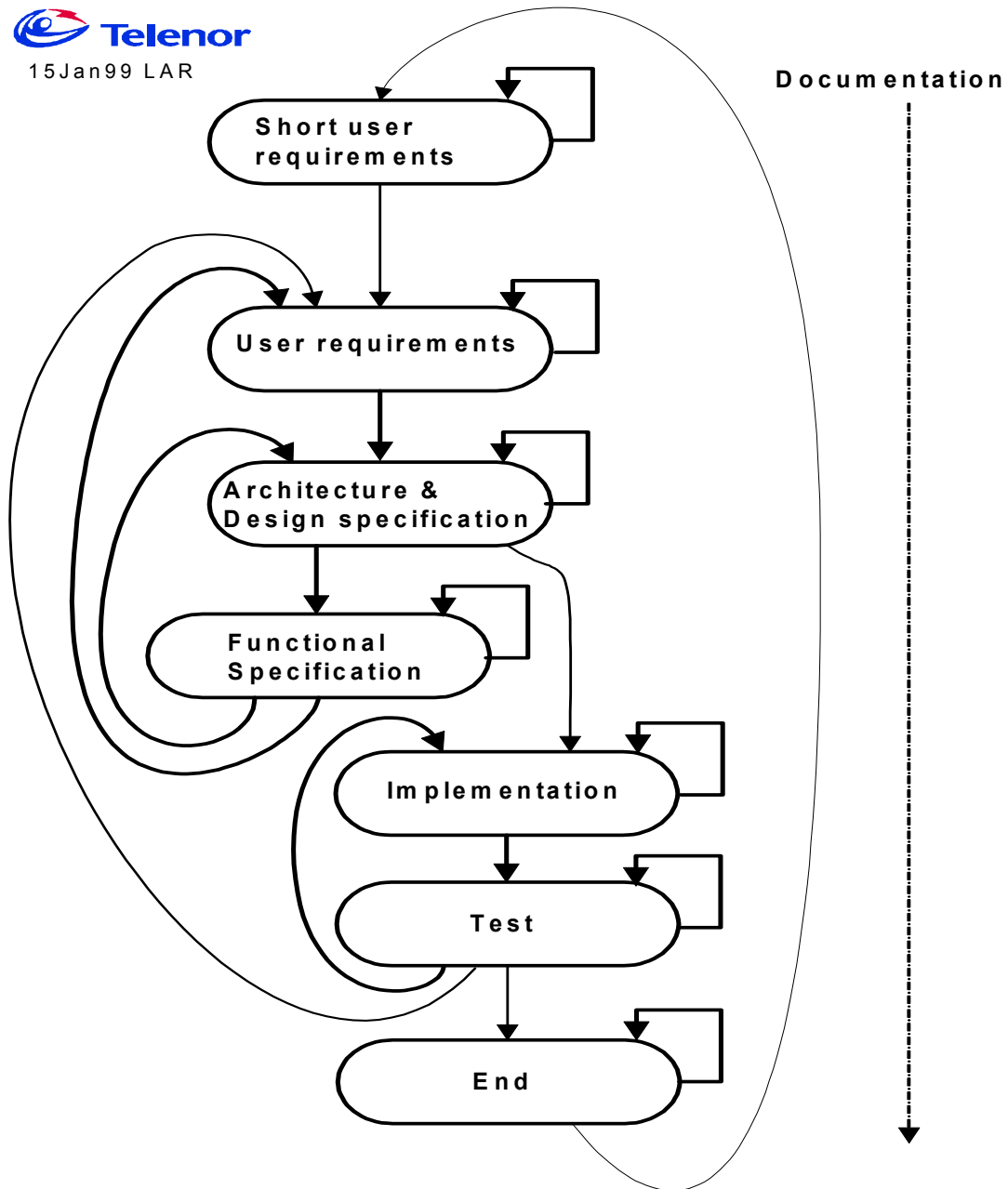


Figure 5. The development process with focus on iterations.

A property of the homogeneous Markov chain is that the distribution of time in a state is geometric, that is, memory-less. Only dependencies between neighbour states are permitted, and other correlation is ignored. More advanced stochastic processes can be introduced, the first step could be to relax on the requirement that the Markov chain shall be homogeneous. However, the mathematical treatment soon becomes intractable.

Establishing the state probabilities and the moments of the passage time from state 'Short user requirements' to state 'end' for a homogeneous Markov chain is quite easy (Cox), but will not be carried out here. A more valuable approach (TBD) is establish a more general model, and solve it by simulation or mathematical means.

Optimisation in design (4)

Introduction

Optimisation in design encompasses hierarchical system design. The fundamental starting point is the geographical distribution of the users of the system. For large cities, terrestrial networks may provide the lowest cost for a certain Quality of Service, QoS. When groups all over Europe shall be reached, satellite is probably the cheapest. However, in this chapter we shall be concerned about the optimisation of networks and equipment used in homes.

It is assumed that a Multimedia Home Network shall be made available to all homes, and the user access points shall be available in every room of the house. Most houses are not prepared for this, cables have to be installed, or some wireless home network has to be used. But when the homes are built of concrete, radio transfer becomes difficult. In this document it is assumed that cables according to the [IEEE 1394](#) bus standard are installed.

The home equipment providing the services will normally be built using a combination of hardware and software.

Formal mathematical optimisation (like mathematical programming) in design of home equipment is probably intractable, but it helps a lot to have a list of the most important QoS issues, and apply heuristic or qualitative approaches to the optimisation process and decision making.

Optimisation in design

A qualitative approach to optimisation in design will be outlined below. The focus will be on performance, cost and human factors. Wide sense performance evaluation in design of information systems, have been focused for many years, e.g., see ([Rønningen82](#)), and has now been placed under the QoS umbrella.

Below are listed a number of important QoS issues seen from the user's point of view:

- Traffic and usage performance
- Cost of purchase, installation, maintenance and up-grade
- Functionality
- Reliability
- Security
- User-friendliness
 - Easy to buy and pay
 - Easy to install, Plug-and-play
 - Easy to use correctly

- Easy to upgrade

- Easy to read documentation
- Easy to get technical support

These QoS issues are largely determined by a number of factors faced by designers:

- Performance
- Cost of development, production, marketing, installation, maintenance and up-grade
- Functionality
- Reliability
- Security
- Modularity
- Scalability
- Software re-usability
- Software boot-loading possibility
- Software module loading possibility
- Plug-in hardware possibility
- Development friendliness
- Upgrade friendliness
- Implementations friendliness
- Test friendliness
- Production friendliness
- Marketing friendliness (easy to learn for retailers)
- Competition, time-to-market

The first to do when the requirements to the system/ equipment are known, is to see if the system or parts of the system can be purchased. It is normally much cheaper to buy and reuse modules than develop from scratch. This again means lower prices to the user.

If development shall take place, several implementation options are available:

- Software + standard hardware
- Application specific hardware
- Software + standard hardware + application specific hardware

After [VHDL](#) has become widespread, the design times for software and hardware are comparable. This is true if the hardware solution ends up with an FPGA, but if an ASIC shall be produced, 3-6 months extra are needed.

Software and standard hardware architectures should be preferred if the performance requirements are moderate to low, or if standards are lacking. It is quite easy, cheap and fast to download new software into equipment, locally or remotely.

If the performance requirements are severe, the only solution is ASIC implementation. But also if standards are fixed, and volumes are large ASIC implementation should always be considered. It is possible to design the ASICs to be programmable to some extent.

The last bullet combination is quite common, and can be a good compromise between functionality, performance, cost aspects and human factors.

Hardware/ Software Co-design

A large number of research projects aimed at tools for hardware/software co-design is running all over the world, see

<http://lester.univ-ubs.fr:8080/~baganne/recherches/codesign/codesign-sites.html>.

Here, the focus is on hardware/Java software co-design, but a few ongoing activities will shortly be referenced.

Time

Time is a methodology for development of communication and information systems, worked out at Sintef. Time and is an extension of SOM (SDL oriented modelling) that has been applied successfully in a number of Sintef projects and other projects over the last three decades.

Time uses the hierarchical functional structure, FS, for describing functional, static system aspects, while signal sequence diagrams, SSD, and state-transition diagrams, STD, are used to describe behaviour for functional modules on system level and sub-system level. To implement STDs, the Finite-state machine, FMS, concept is used. Emphasis is put on using the same languages for hardware and software.

PMOSS

The University of Paderborn and the University of Tübingen, co-operate to develop the co-design tool denoted 'the Paderborn Modular System for High-level Synthesis and HW/SW-codeSign, PMOSS'. See <http://www.uni-paderborn.de/fachbereich/AG/campoag/>.

The phases included are high-level description, partitioning, optimisation and synthesis.

The C language is used for high-level descriptions. Then PMOSS automatically partitions the C-description into hardware and software parts based on 'expert rules'. The hardware part is then optimised, synthesised and described in e.g. VHDL format.

Harpoon

Harpoon is aimed at compiling Java code into silicon. Several persons at universities in America and Europe are engaged in the development. The source code is licensed under the GNU GPL. See <http://www.flex-compiler.lcs.mit.edu/Harpoon/>.

In the Harpoon Web page the compiler concept is described as follows:

'The Harpoon project compiler front-end translates Java bytecode files into a class-oriented intermediate representation which is intended to be easier to analyse and manipulate than bytecode assembly language. The intermediate representation is control-flow-graph

structured, with all control flow explicit. It is also maximally factored and in static single assignment (SSA) form'.

The SSA form vs conventional form is shown below

Conventional form	SSA form
$i = 0$	$i0 = 0$
$i = i + 1$	$i1 = i0 + 1$
$j = \text{func}(i)$	$j0 = \text{func}(i1)$
$i = 2$	$i2 = 2$
.....

The QuadSSA refers to quadruples (the four components) going into SSA operations. Example, $a \leftarrow b \oplus c$. Harpoon has identified Quad statements needed to generate control-flow structures from Java bytecode.

The theory behind the compiler implementation is described by Appel.

INSYDE

The ESPRIT III project INSYDE had the objective to define, implement, validate and demonstrate methodology for software/hardware co-design. The phases in the methodology are analysis, system design, detailed design, and validation.

The analysis uses an Object Oriented (OO) approach ([Rumbaugh](#)) to establish the functional requirements of system. In the system design- and the detailed design phases, a semiautomatic translation from OO to SDL and VHDL for software and hardware respectively, is carried out. The next step is to co-simulate the software and hardware modules, in order to validate the architectural model.

Hardware /Java software Co-design

Figure 6 shows some possible routes from high-level descriptions like schematics, C, C++, Ada, [Java](#) and [VHDL](#) languages into hardware platforms. To obtain standardised solutions, it is desirable that schematics and C, C++, Ada and other languages shall first be translated into VHDL or Java. From VHDL and Java there shall be direct compilation into ASICs, firmware/RISC-on-chip, Java bytecode running on a Java VM and a general processor or a Java processor, or Java bytecode compiled to native code running on a general processor. Note that the ASICs can be made more or less programmable.

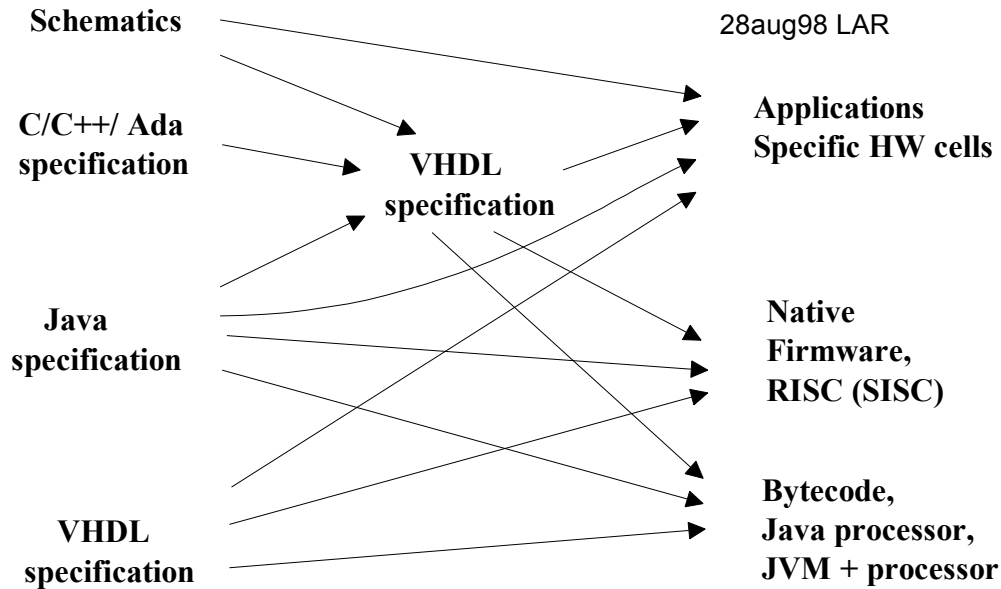


Figure 6. From high-level descriptions to implementation

For the time being, there is (to my knowledge) no commercial tools available for optimising the tasks shown in Figure 6, but there are several research projects addressing the problems, see below.

An important issue is reuse of software and hardware modules. The hardware description can be either cells that are already compiled into silicon, or VHDL descriptions. Software modules written in other languages than Java, can be translated to Java.

The hardware implementations suggested in Figure 6, may be thought of as separate ICs, but can as well be regarded as different realisations of cells on a single IC.

Software design

Design approaches may be procedure- (C, Pascal), object- (Simula, Java), logic- (Prolog), rule- or constraint-oriented ([Boch](#)). Here, object-orientation is focused.

Design methodology

The TIME development methodology has been developed at Sintef during the last three decades. TIME uses the Functional Structure (FS) as the hierarchical structuring description, and Signal Sequence Diagrams (SD) and State Transition Diagrams (STD) for behavioural description. The STDs are implemented as Finite State Machines (FSM). The combination of SOM (SDL oriented methodology), the predecessor of TIME, and object oriented design using Simula/DEMOS has been demonstrated in several projects ([Rønningen84](#)) for about twenty years ago.

The history of object oriented design goes back to 1967 when the first version of Simula was released. Today, Java represents the state-of-the art of object oriented programming

languages. Even if new functions are introduced, Java uses the main concepts from Simula. The concepts of object, class and context (inheritance) are the most important.

An object is used in Simula to describe the structure and behaviour of modules of an actual system. Objects with the same structure and behaviour are defined by one class declaration.

A context can be thought of as a library of predefined classes defined for a system. By 'prefixing' a program block (including class) with the context name, all classes of the 'library' are made available within the program block. The prefixing can be repeated in a hierarchical way (also denoted inheritance). A class at a certain point in the hierarchy may then have super-classes above and sub-classes below.

Another important concept is that of polymorphism. A reference can point to several unequal objects, if these objects have a common super-class.

Today, the concept of Object Model, OM, is widely used. An OM includes the following sub-concepts ([Booch](#)):

- Abstraction: Simplified description of a system, some details are emphasised, others are suppressed.
- Encapsulation: Hiding details of objects from the surroundings.
- Modularity: Possibility to decompose a system into smaller modules that co-operate in a defined manner.
- Hierarchy: Ranked order of abstracted modules of a system

The software development includes the following:

OOA - Object Oriented Analyses. Involves establishment of an Object Model

OOD - Object Oriented Design. With an OM established, design is carried out using objects, object-oriented decomposition and other OM sub-concepts.

OOP - Object Oriented Programming. Coding in a language like Simula or Java, using classes, objects, and other OM sub-concepts.

Related to OOD are several approaches used to develop software and hardware for telecom systems, e.g., SDT . SDT defines a hierarchy of functional processes (parallel objects), and each functional process is thought of as an (Extended) Finite State Machine, FSM. The behaviour of the FSM can be described by means of the SDL language.

The software may compiled to native code for a certain hardware processor, or be compiled to some intermediate code (e.g., Byte Code in the case of Java) and run on a virtual machine (software).

Operating system are used to administer hardware and software resources, see Chapter 5.

Hardware design

Hardware design may involve standard ICs and/or ASICs. The performance, development cost, production cost and time-to-market may decide which approach is selected.

From Java classes to VHDL

A large number of ASIC modules or cells exist as VHDL descriptions. To reduce the overall cost and time, such cells should be reused.

In telecom and Information systems, Java has become the preferred language. But as pointed out above, the efficiency is rather low. It is desirable to automate the task of converting Java programs into hardware as described. Java can then be considered as a design specification, and optimal partitioning between hardware and software can be carried out.

ASIC design methodology and processing architectures

The common phases in digital ASIC design with iterations (normally) from each *review*, is outlined below:

- Customer requirements
- System architecture, analysis, optimal partitioning, specifications
 - *Review, approval*
- VHDL (or equivalent) entry (optionally schematics)
- RTL simulation
 - *Review, approval*
- Logic synthesis
- Gate-level simulation, test vector generation
 - *Review, approval*
- Floor-planning
- Automated layout
- Layout check
- Timing- and power analysis
 - *Review, approval*
- Prototype production
- Test
 - *Review, approval*
- Production

For FPGA, the production is only downloading of configuration into existing chips. A number of tools are available in the market, supporting the various phases.

A generalised object oriented processing architecture (extended MIMD) is proposed in Figure 7. The part below the switch consists of several groups of objects. The objects in one group are instances from the same class, but the number of objects may vary. Above the switch,

there are several groups of different processing units. The processing units in one group are equal, but the number can be varied. There is also an object that is responsible for dynamically dispatching objects to processing units in an optimal way.

The processing unit needs an IO, and may have to its disposal RAM, ROM or Flash memory. The processing unit itself can be a general-purpose processor, a Java processor, an FPGA, a special RISC processor, an programmable ASIC, or non-programmable ASIC.

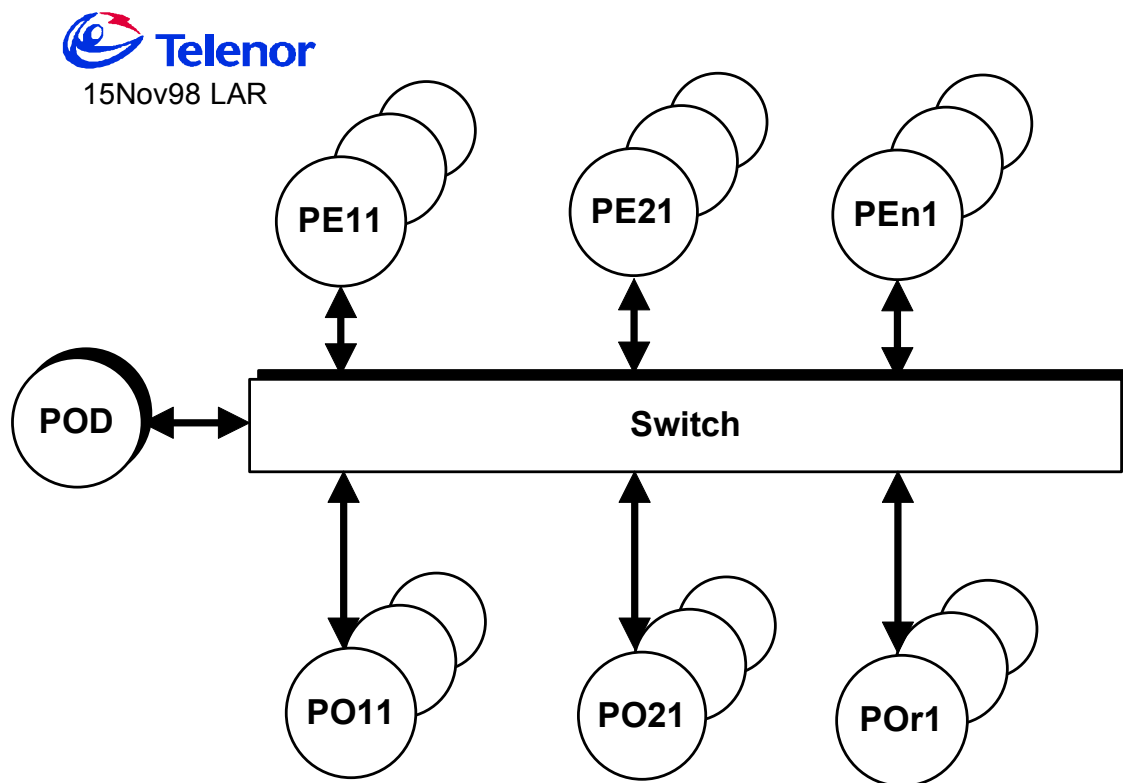


Figure 7. Generalised processing structure.

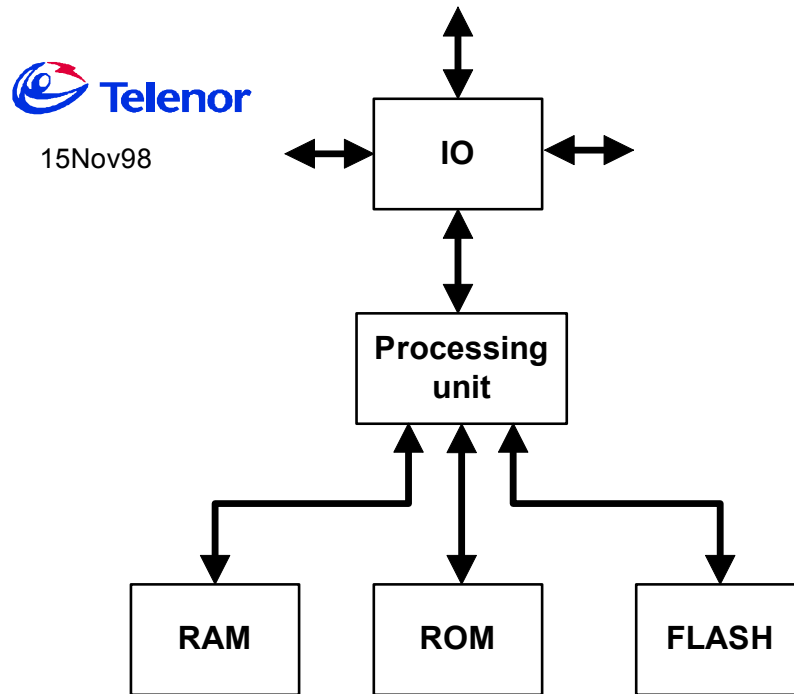


Figure 8. A modular, scalable and configurable processing unit.

Standard processing architectures

Standard hardware processing architectures are described in-depth in text-books, and will not be treated here. See ([Haskell](#)) for a more detailed discussion.

MHN Functional requirements (5)

Introduction, MHN services

The Multimedia Home Network, MHN, shall support the services, 'Enhanced TV', 'Interactive TV', 'Internet access' and a number of other multimedia services, see Chapter 4, and ([Rønningen99](#)). In addition, AV-devices like DVD, DV, PCs and storage devices using the IEEE 1394 bus can be connected to MHN.

DVB, NorDig, Internet/Web, IEEE and other standards are used as basis, and several extensions are proposed. Only extensions are treated in some detail, existing specifications are referenced

The IEEE 1394 bus standard (400Mbps) shall be used as local network. However, for HDTV video component transfer, an extended '1394 (at least 3.6 Gpbs) or other (TBD) shall be used.

When software solutions are preferred, the software modules can run on general purpose processors, or more specialised processors, e.g., Java processors. In every case an operating system is needed. Boot-loaders and module loaders shall be provided for remote update of software modules.

The MHP encompasses of a number of system functions and applications. Several Programming Interfaces, PIs, will exist between the functional blocks. If a PI is available for applications, it is part of an Application Programming Interface, API.

The consequence of developing applications using an API, is that applications become portable, i.e. applications can be downloaded to or plugged into different types of MHNs without any changes, and run immediately.

The DVB Java specifications shall be applied. This is the special variant of Java defined for DVB Multimedia Home Platform, MHP, and is based on the API(s) of PersonalJava and Java Media Framework from Sun Microsystems, Java Media Framework extensions and other Java classes standardised by Davic, NorDig, DVB members, and own proposed extensions.

To support Internet/Web access a Web browser shall be included. The browser shall support at least the HTML 4.0 specification, and JavaScript or equivalent.

Advanced functions as requested by small user groups, shall not increase the cost of the basic, low-end MHN. Therefore, equipment shall be provided as plug-in modules, both to the network and each physical box. Software shall also be provided as plug-in modules. The Common Interface or IEEE 1394 I/O can be applied for each box.

Standard digital TV/ HDTV

Standard TV

NorDig I specifies the hardware for satellite, cable and terrestrial tuners and demodulators, demultiplexers, MPEG-2 [mp@ml](#) decoding, graphical support, general processing and storage capacity, with necessary external interfaces. Furthermore, NorDig I specifies a basic navigator, text-TV and subtitling, and shall include all APIs as defined by DVB MHP.

HDTV

High Definition TV was introduced officially in USA 1 Nov 98. The GA-HDTV (Grand Alliance) specification is composed of higher profiles and levels from the MPEG-2 standard. The main requirements can be summarised as follows:

- The spatial and temporal resolution is increased.
1280(hor) x 720(vert) pels, 24, 30 and 60 Hz progressive scan
1920 x 1080 pels, 24 and 30 Hz progressive, and 30 Hz interlace scan
- 16:9 aspect ratio
- Chrominance sampling 4:2:0
- I, P and B pictures
- Compressed data rate, 10-45Mbps
- 8/9/10 bits DCT Coefficients

For more details, see ([Haskell](#)).

A problem with the GA-HDTV standard is the large number of variables that make it difficult to reduce the hardware cost.

The proposal here for Nordig ng (new generation) is to limit the number of levels to three, and use ASICs for most functions:

- Spatial and temporal resolution
1280(hor) x 720(vert) pels, 25 and 50 Hz progressive scan (Level I)
1920 x 1080 pels, 25Hz (Level II), 50 Hz (Level III) progressive scan
- 16:9 aspect ratio
- Chrominance sampling 4:2:0
- I, P and B pictures
- Compressed data rate, variable 10-45Mbps
- 10 bits DCT Coefficients
- The audio part shall support Dolby C3, and optionally MPEG-2, 5.1 with sample rate extended to 96 k samples per second.

The specification shall include MPEG-2 main profile @ main level.

The interface between the decoder box and monitors has to be extended to include signalling of configuration parameters. For HDTV, this shall be based on '1394 extended for data rates up to at least 3.6 Gbps (TBD).

Service control

The service control includes everything a user have to do to operate his MHN units, and in addition a number of high-level service functions not directly visible to the user.

The service control shall configure the tuner, that is, select frequency (channel), FEC, baudrate, polarization, etc., for satellite reception. Terrestrial tuners shall be configured according to NorDig I.

The service control shall scan and measure the frequency spectrum when searching for channels/services. It shall also control the BER on the channels.

A graphical user interface (GUI) shall determine the look and feel (ergonomics) of the MHN service control, supported by monitors and inputs devices.

At least n (TBD) resizable and movable windows can be placed side by side, overlapping or on top of each other.

The graphical tasks can be implemented as software/ hardware combinations. Making the hardware/software trade-off is part of the design.

Part of the user interface is the service control that shall control all multimedia services for all MHN units.

The service control shall be available on all monitors/input controls.

The user interface shall be common to all services and units, and it shall be possible to control all units connected to the MHN from all monitors /control inputs.

The service control shall maintain complete service lists, derived from SI.

The service control shall implement the TV navigator functions as specified by NorDig I, present following EPG with PPV linking and channel selection.

The service control also includes a Web browser handling at least HTML 4.0.

The service control shall handle MHN configuration and IPv6 address allocation.

The service control shall communicate with Web Servers using the hyper text transport protocol HTTP/1.1.

The HTTP/1.1 protocol shall use the TCP/IPv6 transport and network protocols through Internet (using a terrestrial or satellite Interaction Channel).

The HTTP/1.1/TCP/IPv6 protocols shall be carried via the broadcast channel using the Multiprotocol Encapsulation method defined by DVB.

The service control shall be able to fetch files from inside MHN. The files can be written by object carousels, data carousels, Common Interface, etc.

The service control shall be able to access object carousel streams directly to fetch files or streams.

The service control shall be able to fetch data from Common Interface Command Bus.

The service control shall be able to present a number of (TBD) MPEG-2 video streams (PES) carrying pictures of any size and form, and belonging to a received HTML page. The browser shall check the position of the pictures, and if necessary replace them according to the HTML page. The user shall be able to replace the window to any position on the screen. See ([Rønningen99](#))

The service control shall be able to present a number of (TBD) MPEG-2 I-pictures (PES) carrying pictures of any size and form, and belonging to a received HTML page. The browser shall check the position of the pictures, and if necessary replace them according to the HTML page. The user shall be able to replace the window to any position on the screen. See ([Rønningen99](#)).

The service control shall allow the user to control the volume of MPEG sounds belonging to video streams, still pictures or HTML pages displayed by the browser.

The service control shall be able to synchronise MPEG-2 video, picture and sound, as specified for PES.

The service control shall be able to control the AV streams (MPEG-2, DV, etc) to and from MHP units.

The script language JavaScript 1.2 or equivalent (Java) shall be included in the browser to handle local, (client) dynamic behaviour in Web pages.

Plug-in functions, or plug-ins for short, are software or hardware modules that are called to perform demanding tasks like image, video and audio decompression.

Plug-ins can be downloaded into the MHN using a loader, plugged into Common Interface, or be connected via '1394.

The plug-ins shall be made available to applications through APIs.

The following Plug-ins shall be implemented:

Plug-in for telephone, H.xxx on IPv6.

Plug-in for video telephone, H.323 on IPv6.

Plug-in for video conferencing, H.xxx on IPv6.

Plug-in for video camera, DV standard.

Plug-in for GIF still pictures /execute .gif files/
(GIF 1989 Revised Specification,
<http://www.w3.org/pub/WWW/Graphics/GIF/spec-gif89a.txt>)

Plug-in for JPEG still pictures /execute .jpg files/
(ISO 10918 JPEG Specification,
<http://www.iso.ch/isob/switch-engine->)

Plug-in for PNG still pictures /execute .png files/
(PNG Specification,
<http://www.w3.org/pub/WWW/TR/WD-png>)

Plug-in for AVI video stream (option) /execute .avi stream/

Plug-in for WAV audio (option) /execute .wav files/

Operating system functions

To manage hardware resources and software, an operating system shall be included.

Kernel functions

The kernel shall provide basic operating system functions as listed below. A detailed specification of each function is not covered in this documentation (See [POSIX](#)). Software, hardware and combined implementations are possible.

- Thread
- Clock
- Timer
- Interrupt
- Inter process message
- Signal
- Condvar
- Mutex

Remote software loader

The MHN shall implement a software/ data module loader for both system software and applications. The transfer follows the DVB object carousel specification TM1827. Three different download modes are defined:

Remote software downloading via satellite or cable

An object carousel sends out objects, e.g., software module(s) or other data with a data- and repetition rate agreed between TSS and the MHN software provider. The addressing is based on PIDs. The user has to initiate the actual download (or set default) in his receiver, or if the MHN itself discovers that the software is corrupt, it may initiate a download

For sub-addressing, the objects are IPv6 packets, using IPv6 addresses. An even finer sub-addressing is obtained by using UDP/IPv6 packets as objects. UDP ports can then address functional modules in each MHN unit.

The download stream can be encrypted, using the conditional access system as defined by DVB.

The download stream can be encrypted, using the scheme defined for IPv6, see (Rønningen99).

The stream of software /data modules can be compressed using the (TBD) standard.

Local downloading from a Common Interface Module - CIM

The CIM shall include a buffer for software/ data modules received from satellite. The CIM also sorts out IP packets as ordered by the MHN. The MHN asks the CIM to forward data via the Control bus. Several transfer schemes between the MHN and the CIM are possible (TBD).

The Common Interface Module can be provided with a ROM for software modules (to be sold in shops, etc).

Local downloading from other MHN inputs

This modus is a private solution for each MHN provider.

Memory management

The MHN shall provide a comprehensive memory manager. Detailed specification is not part of this documentation. This includes garbage collection for Java.

Process scheduling

This function shall support pre-emptive scheduling, combined with priority- and 'round robin' scheduling. Details are not part of this document.

File management

The MHN software shall include a file and directory system as defined by java.io.

MPEG-2 PIP decoding

The decoding shall meet the DVB and NorDig I requirements, plus the specification above for HDTV. Furthermore, the decoder shall be able to handle at least six (TBD) independent, parallel MPEG-2 video PES streams (Rønningen99). The video streams can be shown as videos in rectangular windows of any size with HDTV resolution or lower. The windows can be placed anywhere side-by-side or overlapped. In addition, text-TV and subtitling shall be processed according to NorDig I.

Decoding of object oriented video streams is optional (TBD).

The audio decoding shall support one Dolby C3 stream, and three MPEG-2 stereo streams simultaneously.

Input:
MPEG-2 PES over IPv6.

Output:
Video component format over IPv6 (TBD).

Graphics

The graphics processing shall support the services described in Chapter 2, and the service and control functions specified in this Chapter, including games. The MPEG-2 HDTV decoder and monitors gives a possible new enhancement of graphics.

The spatial and temporal resolutions shall follow the specifications for standard TV and HTDTV in this Chapter.

2D and 3D shall be supported

The graphics processing shall support at least (TBD) rectangular windows of any size shall be supported. The windows can be placed anywhere side-by-side or overlapped.

Object oriented window shapes are optional.

It shall be possible to overlay graphics on MPEG-2 video. The number of windows shall be at least (TDB). Rectangular windows of any size shall be supported. The windows can be placed anywhere side-by-side or overlapped.

It shall be possible to run at least 5 MPEG-2 video PIPs (Picture-in-Picture) as part of a graphical window (e.g. a HTML page).

Processing capacity shall be determined by the response time to user inputs. The maximum contribution to the response time from the graphics system shall be (TBD) milliseconds.

The NEC/ VideoLogic Power VR Second Generation, PVRSG, single-chip graphical IC is used as model, but has to be extended to support the HDTV requirements.

The graphics module shall support the following:

- Z-buffer, 32 bit, access rate at least 1 Gbps
- Frame buffer, access rate at least 3.2 Gbps
- Vector quantisation texture compression
- Volumetric shadows
- Multiple fog modes
- Deferred rendering, trilinear filtering
- Translucency sorter
- Bump mapping
- Full scene anti-aliasing
- Anisotropic texture filtering
- 3D in a window
- At least 128 Mbyte RAM addressing
- DVD assistance (MPEG-2)

Input:

MPEG-2 PES over IPv6

Output:

SDTV video component format over IPv6 (TBD)

HDTV video component format over IPv6 (TDB)

Satellite reception and transmission

The satellite unit shall comprise a down-link module and an up-link module. The down-link module shall include LNB, RF amplifier, filter, tuner, demodulator, adaptive antenna control (TBD), and a protocol converter.

The unit shall be able to receive simultaneously three (or more) different MPEG-2 transport streams from one to three (or more) different frequencies.

The local up-link module shall include RF amplifier, up-converter, filters, modulator, adaptive antenna control and a protocol converter.

The input/output to/from the satellite unit shall follow the IPv6 protocol over '1394 for all transfers.

Terrestrial reception and transmission

The default terrestrial input/output shall be ADSL. xDSL, ISDN, cable-TV (DVB-C, NorDig I), terrestrial TV (DVB-T, NorDig I), DECT and ATM connections shall be provided by plug-in modules.

Security functions

Authentication, data privacy, digital signatures, conditional access and copyright monitoring/control shall be provided by IPv6. Conditional access to MPEG-2 streams shall be based on PES packets, following the DVB standards.

Protocols, demultiplexing

The MHN shall handle demultiplexing and protocols as follows:

PID filtering

PID filtering shall be accomplished as specified by NorDig I.

Section filtering

Section filtering shall be carried out as specified by NorDig I.

TCP, UDP and IPv6.

These protocols shall be processed and used as specified in earlier sections.

ADSL

ADSL is mandatory and the default protocol for terrestrial communication.

xDSL

Several versions of xDSL may be included as plug-ins.

ISDN

ISDN shall be included as plug-in.

Cable TV

Cable TV shall be connected as specified by NorDig.

ATM

ATM protocols may be used for the interaction channel over satellite. The ATM protocol formats shall be converted to IPv6/'1394 in MHN.

PSI/SI

PSI/SI shall be processed according to NorDig I.

Data carousel

The processing of data carousel data is according to NorDig I.

Object carousel

The processing of object carousel is according to NorDig I.

Multiprotocol encapsulation

This protocol shall be used to transfer IPv6 via satellite.

Processing of HTTP/1.1 is mandatory. The protocol shall be applied as described in more detail in document documents by Rønningen and Luotonen.

The IPv6 protocol has been described by [Gai](#), Huotonen and Rønningen99. Processing of IPv6 is mandatory.

Figure 9 shows the protocol stacks and internal relations, assuming service control, Web browser and applications on top.

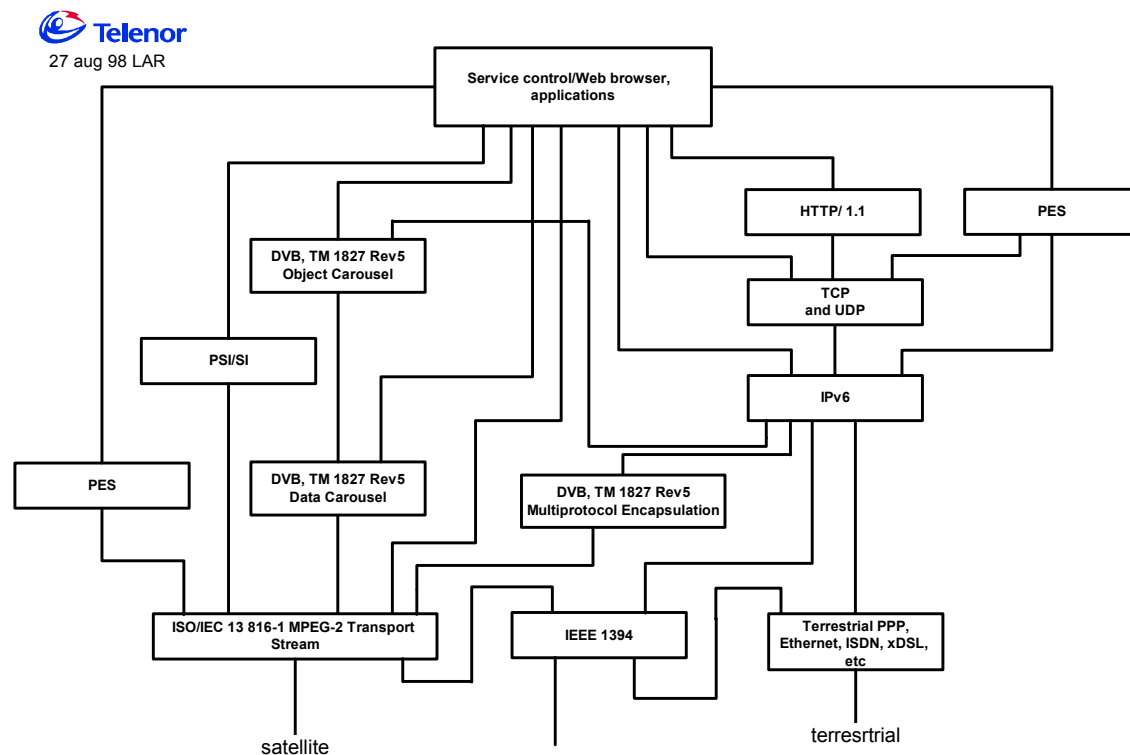


Figure 9. The MHN protocols

Not shown in the figure are different interfaces between the protocols. E.g., on top of TCP Corba, socket, Java RMI or other can be used.

Extended use of IEEE 1394

The MHN may include a number of functions that will be used by a fraction of the customers. The cost for introducing these functions should not be charged all users. The obvious way around this is to use plug-in modules into the '1394 bus.

The basic MHN transport mechanism shall be IPv6 over '1394, which can carry MPEG-2 PES, UDP, TCP, or any other protocol or format. Video and Audio Serial Component form, Y, Ca, Cb over IPv6 shall be supported. MPEG-2 PES packets and DV formats shall be carried by IPv6. Other formats to be supported are Quicktime and H.323 video.

Extended use of Common Interface

As outlined above, DVB is specifying a MHP API based on DVB Java. It is natural to use Java also for several CIM module functions. Different CIMs require different protocols, especially because of the performance requirements.

CIMs will be used to introduce new function and more capacity. This is valid for existing MHPs, and coming MHNs.

Below, some Java based Common Interface extensions are proposed.

Read/Write Buffers

Nothing gives better performance than accessing hardware-near routines, e.g. written in assembly, directly from the applications. However, this generally means that the application programmer must have comprehensive hardware-near skills, which often is not the case. Normally, such interfaces are difficult to use for application programmers.

For Common Interface the low-level I/O routines from java.io shall be adapted. Java provides several hardware-near classes for this. Some Java.io candidates are java.io.BufferedInputStream, java.io.BufferedOutputStream, java.io.BufferedReader, java.io.BufferedWriter.

File transfer

Communication takes place via files. Standard Java file- and directory schemes shall be applied. Such an interface is reasonably easy to use, and gives reasonable performance. The Java file system shall be adapted to Common Interface. Some Java.io candidates are java.io.FileInputStream, java.io.FileOutputStream

The files may be stored in the MHN or in the CIM. Both options shall be supported.

Sockets

Standard sockets can be used on top of CI protocol. Sockets are reasonably easy to use, and give good performance. Sockets as specified for TCP, UDP and IPv6 shall be included.

Java Remote Method Invocation

Java Remote Method Invocation, RMI, is Java's answer to RPC (remote procedure calls) used e.g. in Unix environments, and to some degree Microsoft's DCOM and CORBA. As the title indicates, a Java program can access any method of any other Java object, whether it runs on a different JVM on the same computer, or on a remote computer. The main advantage of RMI is that its API is very simple to learn and use, much simpler than COM, CORBA or sockets. The application programmer is totally off-loaded regarding details of the communication. A major drawback is that implementations known so far have low performance (Dow1). However, as pointed out by Harxxx, the performance can be enhanced considerably by simple means.

RMI carries out remote method invocation through a reference to a remote object. Specifically, the remote object has to implement the `java.rmi.Remote` interface. On the application layer the client and server communicates via their respective APIs. The stub layer provides the API for the client, while the skeleton layer provides the API for the server. In turn, the stub and skeleton layers communicate via a remote reference layer. The remote reference layer provides an abstraction layer above the transport layer, and also supports replicated objects, semantics and lost communication recovery. The transport layer may be a TCP connection, or a Common Interface transport connection.

CORBA

CORBA (Common Object Request Broker) is designed to support communication between applications written in any language, and running on any hardware platform. CORBA includes the Interface Definition Language, IDL, and IIOP protocol, defined by the OMG (Object Management Group). The software is complex and not so easy to learn and use. The user must learn IDL. CORBA gives reasonable performance. CORBA is supported by Java 1.2, and a full discussion can be found on www.omg.org.

Architecture and design of MHN (6)

Introduction

Based on the draft specification in Chapter 5, a distributed, service-scaleable architecture for the Multimedia Home Network is proposed. The concept is based on the IEEE 1394 serial digital bus. AV (audio-visual) equipment already uses '1394. '1394 meets the requirements for high-quality video streaming.

The MHN Architecture

The DVB Multimedia Home Network, MHN, system consists of a varying number of communicating units in the home, a possibility is shown in Figure 9. The architecture is highly scaleable, that is, the actual configuration may vary from two or three units to more than those shown in the figure. The network can communicate via satellite (two-way), cable, terrestrial wireless, or via the 'telephone' lines (ADSL, ISDN, ATM, etc). The flexibility adds cost to large configurations, especially related to mechanics and power supply, but gives low cost for small configurations. The architecture is scaleable with respect to services/functionality as well as processing capacity/performance. In short, the emphasis is put on being 'future-proof'.

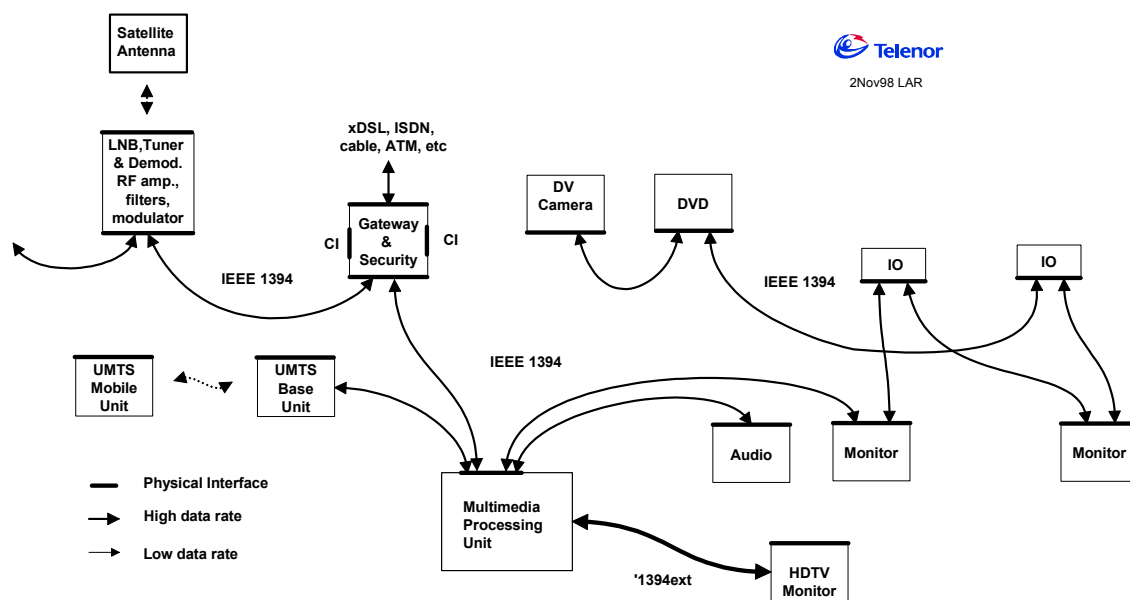


Figure 9. The scaleable Multimedia Home Network

The IEEE 1394 bus

The '1394 bus can address 64 units (without bridges etc.), and handle data rates close to 400 Mbps. The maximum cable length between units is presently 4.5 meters. Power can be supplied to units over the bus (maximum 40 V, 1.5 A). Over '1394 the IPv6 protocol will be used (xx). Each unit can reserve the necessary capacity for its service on '1394 from the Bus Master.

Personal, Mobile, Global addressing with IPv6

The Ipv6 addressing space makes it possible to address MHN units directly (this is not possible with IPv4, due to limited address space). Each user should be allocated a range of (256) IP addresses on permanent basis, where one is reserved for unique user identity. The other addresses can be allocated any user unit. In addition, each user can be allocated to user groups, permanently and/or temporarily. The allocated IP addresses shall be mobile, i.e. follow the user. The roaming concept used for the GSM mobile telephone system, with a home location register and temporary location register, could be applied as basis (TBD). To be located, the user must plug in his security card into the Gateway & Security unit, and can then operate the MHN units and get access to services. The Multiprocessing Unit orders the Gateway & Security Unit to allocate temporarily the actual user's IP addresses to MHN units (TBD).

IPv6 shall carry several protocols and formats:

- PES packets
 - Video
 - Audio
- Section packets
 - PSI
 - SI
 - Data Carousel
 - Object Carousel
- TCP protocol
 - HTTP/1.1
 - E-mail
- UDP protocol
- YUV video components (xx)
- Audio components (xx)
- ICMP messages
- H.3xx protocols (xx)
- DV (Digital Video) formats
- Other

The Gateway & Security Unit

This unit shall always be included in MHN, and shall handle conditional access to all services, authentication, data privacy, digital signatures and copyright protection. IPv6 supports all this except digital signatures. This applies to both incoming and outgoing services. This unit shall also handle all IPv6 addressing resolution as mentioned above. The unit contains the gateway function for ADSL as default, and provides ISDN, ATM, cable TV connections using plug-in modules. The unit shall have three or more Common Interface slots for CA-modules and for other-purpose CI modules, CIMS. CIMS can be used to enhance capacity or functionality, see next chapter.

MPEG-2 transport packets received from either terrestrial or cable networks that do not belong to any service signalled in PSI/SI shall be filtered out and stopped in this unit. The intention is to minimise the load on the '1394 bus.

General AV units

The units on the left-hand side of the figure are standard AV units, including DVD and DV camera. The intention is that a monitor, audio equipment and an IR input for service control can be placed in any room of the home. It shall be possible to watch several different TV programmes in parallel. The IR input can be the normal TV IR control, a keyboard or speech.

Other devices like digital still picture cameras, printers, mass storage devices, PCs, etc., supporting '1394 can be connected as well.

When the number of units exceeds 64, an IPv6 router can interconnect a number of '1394 buses.

UMTS

Of special interest is the introduction of UMTS in the MHN. In the CTS mode two units are needed, the base unit, and the mobile unit. The UMTS-CTS can provide wireless, mobile, in-house services like, telephone, video telephone, telephone, Web access, email and other multimedia services. UMTS is planned to be operational in 2002.

The satellite unit

The satellite unit shall handle satellite down-link and up-link, see Figure 10. An ad-hoc group, comprising Eutelsat, Hispasat, Intelsat, SES-Astra, Telenor, Telesat, Teracom and ESA, has drafted a standard for two-way satellite communication in homes. The home terminal is denoted SIT, Satellite Interactive Terminal. For down-link, the proposal is to use DVB Multiprotocol Encapsulation or alternatively, ATM-based formats, and for up-link only

ATM-formats. In every case, IP is used as network protocol. The primary AV streams will use MPEG-2 PES as before.

In our case, the satellite unit shall include the following:

- Duplexer and antenna adapter
- Adaptive antenna control
- Proxy, i.e., protocol converter

Down-link

- LNA & down-converter
- Filters
- Tuner
- Demodulator

The unit shall be able to receive simultaneously three (or more) different MPEG-2 transport streams (frequencies).

Up-link

- RF amplifier
- Up-converter
- Filters
- Modulator

The input/output to/from the satellite unit shall follow the IPv6 protocol over '1394 for all transfers.

MPEG-2 transport packets received from either terrestrial or cable networks that do not belong to any service signalled in PSI/SI shall be filtered out and stopped in this unit. The intention is to minimise the load on the '1394 bus.

MPEG-2 PES packets and Sections, and TCP and UDP packets shall be sorted out and packet into IPv6/ '1394.

IP packets intended for satellite up-link, shall be put on top of the ATM protocols as specified by SIT before transmitted.

Most functions shall be implemented as ASIC(s) to get the component cost down. Only high-level control functions shall be implemented by means of Java software and general-purpose micro- controllers, see next section.

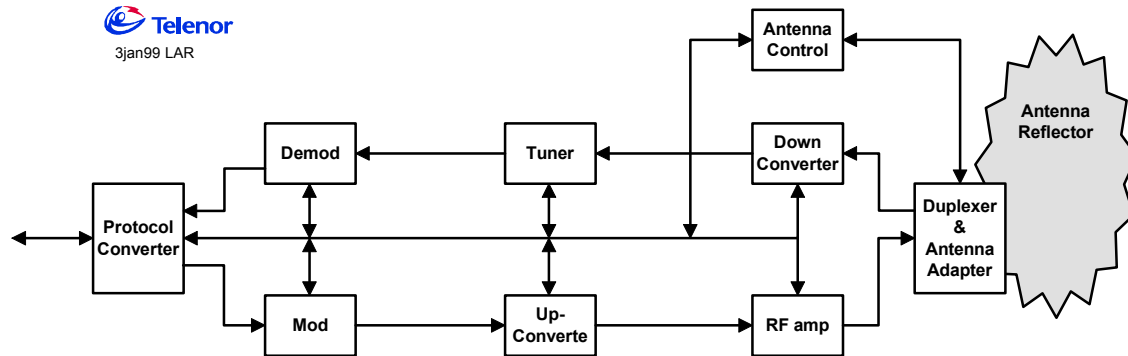


Figure 10. The satellite unit

The Multimedia Processing Unit

This unit can be an MPEG-2 decoder with graphics, a multimedia PC, a graphical play-station, or anything in between. Figure 11 shows a possible architecture. Several processing sub-units communicate through a switch. One sub-unit implements protocols and demultiplexing. The '1394 is the main IO port of the unit. Infrared IO, Smart-card IO and Common Interface are optional. The unit also implements graphics support and service control functions. The unit shall be able to decode three TV programmes, or three MPEG-2 AV streams for any (PIP) video window size and form, simultaneously. Note that several Multimedia Processing Units can be connected to the bus. Normally, the unit is remotely controlled from one of the monitors and IO control units.

IP Switch/ Router

The IP Switch/Router shall be able to handle at least eight simultaneous, full duplex connections at a serial data rate of at least 1 Gbps. The extended '1394 bus, '1394ext', may be used (TBD). Alternatively, IP routers using the 8 lsbs of the IPv6 address, together with a simple serial bus for the link- and physical layer (see below), can be used. The latter alternative will be followed here.

Note that for HDTV and high-performance graphics, direct connections are provided from the Demultiplexer & Protocols Unit to the MPEG-2 PIP Decoder, and from there to the Graphics Unit. There is also a direct connection from the Graphics Unit to the HDTV monitor.

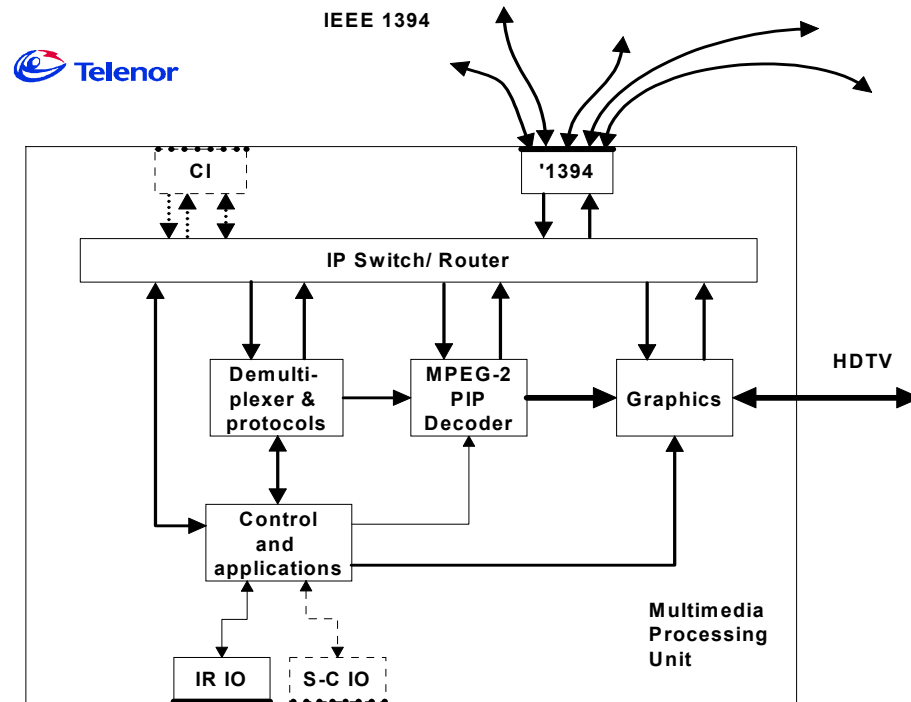


Figure 11. The Multimedia Processing Unit

Bootloader

The bootloader is considered as part of the hardware, and is used for full system- and application software upgrade. The actual upgrade shall be controlled by the Gateway & Security Unit and the user service control.

Service Control and Applications

This module implements a common user interface for controlling all services and units connected to MHN. Included is the configuration of the Gateway & Security Unit (e.g., local IPv6 addressing). A number of Java classes, comprising the Service Control (see Chapter 5), has been and shall be defined to cover this (TBD), (DVB MHP API).

When tools become available for partitioning Java classes into respectively VHDL code for ASIC synthesis, and bytecode for Java processors or JVMs on general-purpose processors, then the software/hardware optimisation task becomes straight-forward. For the time being, this is a manual task, but several tools are in development (Harpoon).

To control hardware and software resources of the Multimedia Processing Unit, an operating system as specified in Chapter 5 shall be installed.

The Java platform provides loading of software modules, see Java [JDK](#) documentation.

When a general-purpose microprocessor is applied, Java uses a JVM (Java Virtual Machine) to run the bytecode. To give a satisfactory performance, a 32 bits processor of 300-400 MIPS will probably (TBD) be needed. The memory requirement (TBD) could be higher than 128 Mbytes of Flash memory and 128 Mbytes of RAM.

Enhancing performance is especially important for the demanding plug-ins as specified in Chapter 5.

How to speed up Java

To speed up the Java execution rate on general-purpose processors, several steps can be taken.

The straight forward is to introduce a Java processor that executes bytecode directly. This may speed up the execution up to twenty times. However, there are also possible to enhance the execution of Java software considerably, as shall be seen below.

The efficiency of Java is quite low compared to e.g. C language execution. The Java code size is also much larger than corresponding C code. This can partly be explained by the objectives of 'write once, run anywhere' and high programmer productivity.

Object oriented languages like Java support inheritance, polymorphic function (method) invocation, and very general libraries where only a fraction is used. This gives large code and slow execution.

Furthermore, Java normally uses a interpretative Java Virtual Machine (software). This means that before any instruction can be executed, decoding of bytecode must take place, and the performance becomes low.

It is possible to introduce just-in-time compilation, JIT, and this gives faster execution than the interpreter, but is still much slower than corresponding C native implementations. The JIT requires additional software, and code optimisation takes too long time to be used.

The solution to the performance problem but still maintain the 'write once, run anywhere', is to use static compilation of bytecode to native code. This eliminates the JVM. Furthermore, a special 'OO optimiser' shall be applied to remove most of the OO overhead during compilation.

JOVE is an example of a modern Java compiler that optimises the code in an excellent way. The code size and code execution speed approaches that of C.

Demultiplexer & Protocols

Most of the DVB and Internet/Web protocols specified in Chapter 5 can be implemented in hardware, but part of it should be implemented with Java statically compiled to native code for a RISC processor, or directly to silicon.

Tools for partitioning Java classes into either VHDL code for ASIC synthesis, or bytecode for Java processors or JVMs on general-purpose processors in an optimal manner, should be developed.

Figure 12 shows a proposal for a software/hardware architecture for the Demultiplexer & Protocol module. The intention is to increase the performance of handling DVB and Internet protocols compared to what is obtained with PCs and IRDs today. At the same time, the cost should not increase. The intention is to obtain a one-chip solution for the Demultiplexer & Protocols module.

The switch shown in Figure 11 is implemented as an IPv6 router. IPv6 packets coming in from the '1394 bus are automatically routed to their destination by means of the eight lsbs of the IPv6 address.

The physical transfer between hardware modules can be implemented by means of a simple two-way serial bus with the following lines/signals:

- Common earth
- Send
- Send strobe
- Send bit-clock (optional)
- Receive
- Receive strobe
- Receive bit-clock (optional)

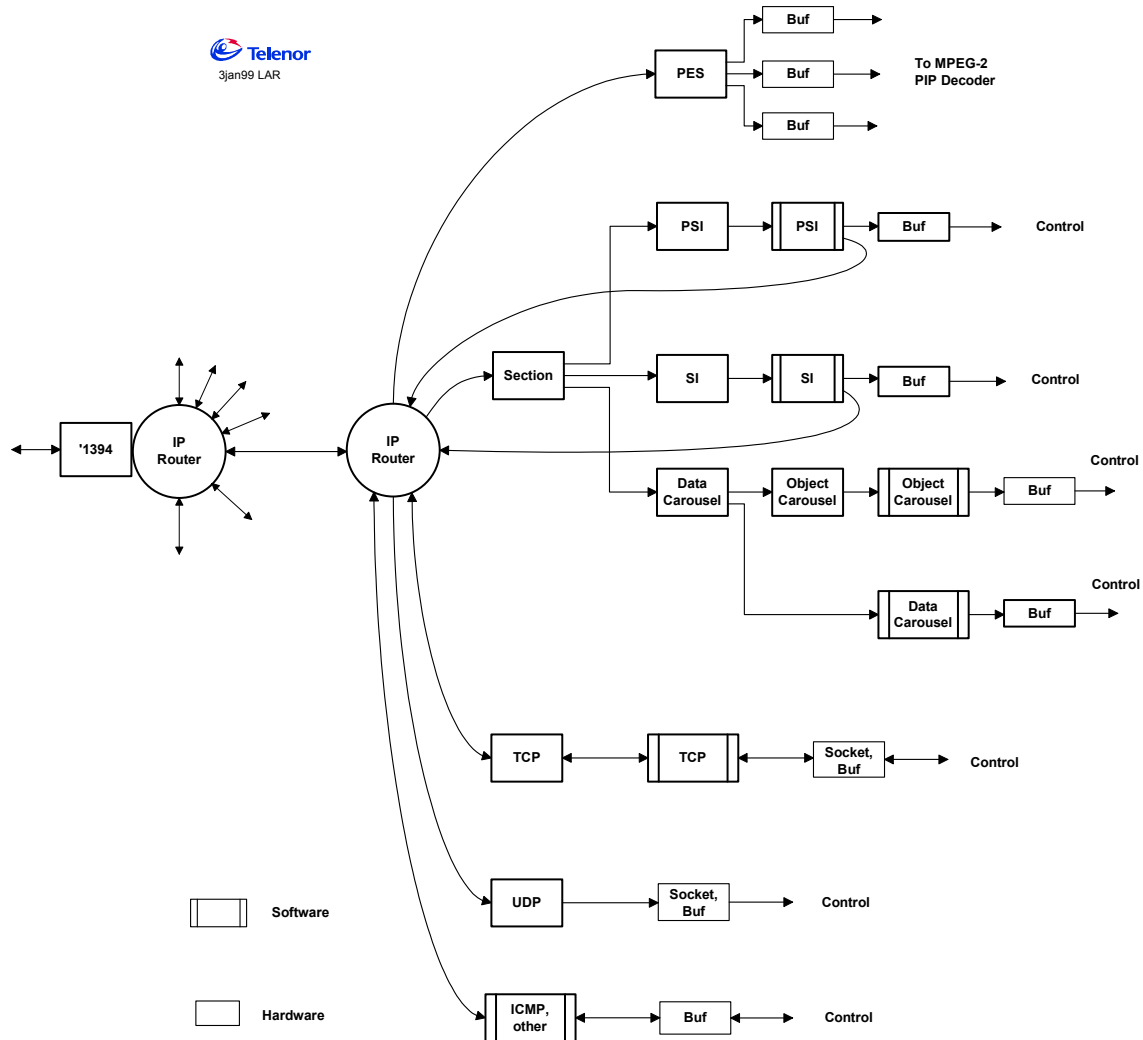


Figure 12. The Demultiplexer & Protocol modules

Processing of PES

The data rate (average) of incoming PES/IPv6 streams is high, up to 40 Mbps for each HDTV video. PES packets for different audio and video streams have to be sorted out and put into separate buffers, which can be read directly by the MPEG-2 PIP Decoder. Due to the high data rates, and that the PES packet format is completely specified and stable, the implementation shall be carried out by means of ASICs. It is assumed that Java bytecode exists, and a compiler like Harpoon shall be used for the ASIC design.

Processing of PSI

The Section module receives IPv6 packets, sorts out the section containing the PSI information and sends it to the PSI module. The PSI module consists of hardware and software. PSI consists of four tables (NIT, PAT, PMT and CAT) that will be maintained in the PSI buffer. From there, the tables are available from the service control. The PSI module shall be configurable, so that information about which transport packets (by PIDs) to be sorted out can be sent directly to the Satellite Unit and the Gateway & Security Unit.

It is assumed that a Java implementation of PSI exists, and the selection of type of processing units has to be made manually (this will be automated when tools become available). It is proposed to use a RISC processor with IO, flash memory for program and RAM for the PSI buffer. The Java bytecode shall be compiled and optimised into native code (firmware) and put into the flash memory.

Processing of TCP

The TCP protocol behaviour can be described as a finite state machine, FSM, see (Siyan). The FSM can be implemented in Java software as shown in Figure 13. The TCP module shall be implemented using a RISC processor with flash memory, RAM and IO. The Java bytecode shall be compiled and optimised into native code (firmware) and put into the flash memory. The TCP buffer shall be two-way, and shall satisfy the socket interface.

The FSM can be implemented in several ways, as described in (Rønningen 84). A table oriented software version, see Figure xx, goes as follows:

State-interpretor:

- Wait for in-message
- Read current state
- Take in-message out of inqueue
- Search in state-description for match on in-message
- Send out-message(s) as given by state-description
- If any, activate task(s)
- Set next state
- Repeat

Example of state-description, state LISTEN:

- | | | | |
|----------------|-----------|---------|-----------|
| • State | LISTEN | | |
| • In-messages | rcvSYN | SEND | CLOSE |
| • Out-messages | sndSYNAck | sndSYN | deleteTCB |
| • Next state | SYNRCVD | SYNSENT | CLOSED |

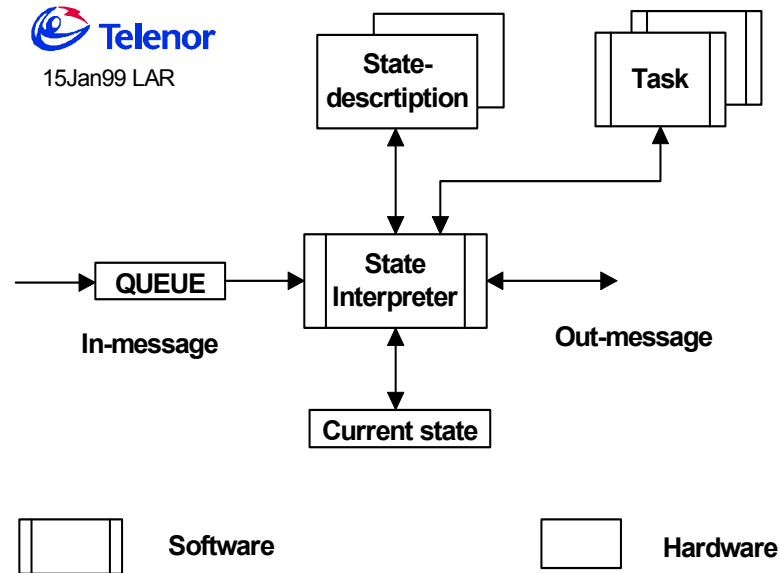


Figure 13. Table oriented FSM implementation

Other modules

It is envisaged that that other modules of the Demultiplexer & Protocols module also shall be implemented using a Java native code/RISC/RAM/flash combination.

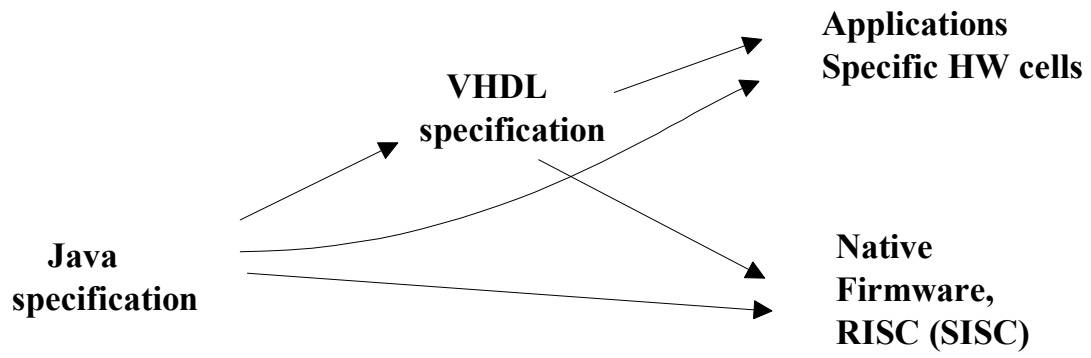


Figure 14. Optimal Java compilation to hardware or native firmware

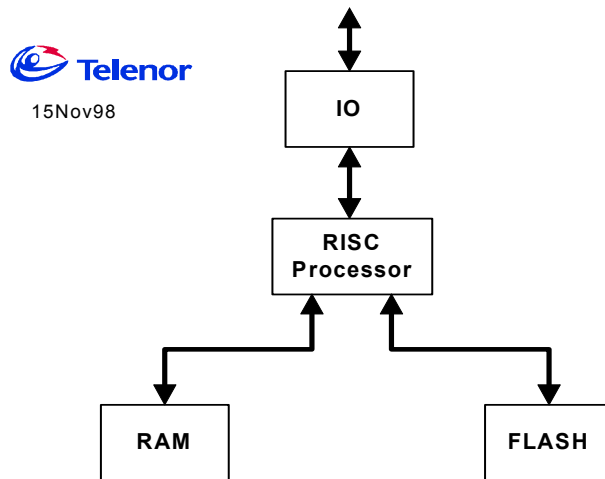


Figure 15. RISC processor module

MPEG-2 PIP decoder

HDTV ASIC vendors should design this module, according to the specifications in Chapter 5.

The IO shall be based on IPv6. Incoming packets of PES format shall be processed, and the output shall have YUV video component format

Graphics

The NEC/ VideoLogic ASIC, the Power VR Second Generation, PVRSG, meets most of the requirements, but has to be extended to handle the HDTV- and IO requirements from Chapter 5.

The input to this unit shall have YUV video format or bitmap. The output shall have YUV video format. For standard TV, the IPv6/ '1394 combination will be used. For HDTV a serial high-rate (3.2 Gbps) direct connection from the graphics module to the monitor is proposed.

Common Interface

See NorDig I specification.

Smart-card, IR IO

See NorDig I specification.

Part C

Common Interface Modules (7)

Introduction

The Common Interface modules, CIMs, proposed in part C, are intended for 1st and 2nd generation IRDs, but can also be used in the Multimedia Home Network, MHN.

Using the Common Interface slots, modules that enhance functionality, performance or both, can be plugged in. This means that the low-end users are not burdened with the cost of features wanted by the high-end users. On the other side, the high-end users get cheap products, because the basic box is the one used in the volume market.

CIMs are an important means to obtain a smooth migration from existing IRDs to the coming MHN. The CIM may consist of any combination of processors, memories and specialised hardware used to run IRD software tasks, sharing load and/or functionality.

The intention of this Chapter is not to specify or design CIMs, but rather to outline ideas.

Java Common Interface Module

A Java CIM will typically include a Java processor, IO, flash memory and RAM. A bootloader, used to download all software into the module, shall also be part of the hardware. The Java software shall be translated to bytecode form before downloaded. Note that the Java Virtual Machine, JVM, now has been substituted with a Java Real Machine. A tiny operating system is needed to control the CIM resources and scheduling.

A Java processor can be 10-20 times faster than a JVM running on a general purpose processor with the same clock rate. Since the maximum data rate over the Common Interface Command Bus is limited, the co-operating tasks in the CIM and the IRD should be loosely coupled.

In Chapter 5, several approaches for intercommunication between a Java CIM and the IRD are described. If ease-of-use is the most important, high-level approaches like RMI can be used. If high performance is needed, low level Java IO shall be applied.

This CIM can, e.g., be used for filtering and caching, see below.

General processing

General purpose CIM can include a general purpose processor, RAM, Flash memory, IO and an operating system like QNX or Linux. A bootloader shall also be included. Any software (C, C++, Java, etc) can in principle be downloaded. If Java is preferred, a JVM can be supplied. If high performance is needed, C software gives the fastest execution. But as described in Chapter xx, Java can be speed up by using static compilation and optimisation. The CIM can be used for several of the applications described below, e.g., filtering and caching.

Graphics processing

It is possible to use a CIM as graphical support for games, but then the output from the CIM should be MPEG-2 PES/ transport stream, (the Command Bus is too slow) to utilise the MPEG-2 IRD decoder capabilities. If graphical operations shall take place on uncompressed material, then a MPEG-2 encoder is needed. The cost of the encoder is more expensive than an IRD itself (at least for a long time), so this solution is not attractive. But if graphical operations can be made in the 'MPEG-2 compressed space' then the graphical CIM approach can be valuable. Research on these issues is needed.

The PVRSG graphical chip designed by NEC/VideoLogic, probably represents the state-of-the art in 2D and 3D graphical hardware support.

Memory extensions

Using Common Interface for memory extensions is not attractive because the address bus has been limited to 16 bits, and the Command Bus data rate is slow. However, if a Common Interface mode that corresponds to the original PCMCIA can be defined, then it is interesting to apply existing or coming PCMCIA RAM and flash memory modules.

Filtering

Since the whole MPEG-2 transport stream passes through the CIM, it is possible to sort out packets, process the content and output to the IRD (this is exactly what is done with conditional access). The output can be placed on the transport stream or the Command bus.

Caching

Caching of Java classes, IPv6 packets, HTML pages, stream objects, configuring data, etc. coming from Internet (via satellite), carousels or other can be accomplished. The caching can be combined with filtering or any other processing.

Format conversion

Actual format conversions are between DV and MPEG-2, H.323 to MPEG-2 and others. For this specialised hardware (ASIC) is needed, that could be combined with a Java CIM or a general processing CIM.

Video phone

Video conferencing and video phone are now expected to be popular services (after earlier trials that was not successful). The video camera could be connected directly to the CIM, that

converts to H.323 format. At least 64 kbps is needed for acceptable video quality (own judgement). This means that an ISDN connection (or better) to the IRD is necessary. Incoming H.323 content should be converted to MPEG-2 in the CIM and sent on the transport bus to the IRD decoder and further the TV monitor.

DVD

DVD uses MPEG-2 PES as data format, but the interface protocol does not support the MPEG-2 transport stream. A CIM that can be connected to the DVB directly and can convert to/from MPEG-2 transport stream, is a valuable module. The same module can be used for other storage media using the DVD interface.

DV

DV (Digital Video) equipment (and other AV equipment) uses the IEEE 1394 serial bus for transport of content. The compression techniques used for DV are similar to MPEG-2, but not equal (based on discrete cosine transforms, etc). Providing a CIM with an external '1394 connector and a DV-MPEG-2 format converter could be an interesting product, giving better quality than the PAL converted output.

Tuner&demodulator

To have the possibility to plug different tuners/demodulators (for satellite-, cable- and terrestrial broadcasting) into the IRD, special CIMs are under development. See the Nordig I specifications.

Migration support

There is a large number of first generation IRDs sold, that should not be thrown away when new generations arrive. If the IRDs have Common Interface and a bootloader, a lot can be done using one of the CIMs outlined above. The obvious support from CIMs is the extension of processing capacity and functionality.

A problem for all actors in the digital TV race, is the number of different EPG formats needed for different systems (MediaHighway, OpenTV, D-box). In an open, horizontal market, this means that every IRD should be able to decode a number of EPG formats, and that several EPG streams has to be transmitted in parallel, even if the content more or less is the same. The simulcasting of EPG formats increases the cost for the user, and makes the service less attractive. A way around this is to transmit only one EPG format (preferably HTML) and introduce CIMs that convert from the sent EPG format into the format used by the IRD. The limited graphical performance of the 1st generation IRDs can be enhanced by the same CIM.