

# AsTeRICS Deliverable D2.4

## Report on State-of-the-Art

UCY

## Document Information

Issue Date	3 June 2010
Deliverable Number	D2.4: Report on State-of-the-Art
WP Number	WP2: System Architecture and Specification
Status	Final
Dissemination Level	RE PU Public PP Restricted to other programme participants (including the Commission Services) RE Restricted to a group specified by the consortium (including the Commission Services) CO Confidential, only for members of the consortium (including the Commission Services)

**AsTeRICS – Assistive Technology Rapid Integration & Construction Set**  
Grant Agreement No.247730  
ICT-2009.7.2 - Accessible and Assistive ICT  
Small or medium-scale focused research project

## Disclaimer

The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

The document reflects only the author's views and the Community is not liable for any use that may be made of the information contained therein.

## Version History

Version	Date	Changed	Author(s)
0.1	12 Feb. 10	First draft	UCY
0.2	22 Feb. 10	CEDO & UMPC outlines integrated	UCY
0.3	12 Mar. 10	New template, added content for 2.3 and Wiimote	KI-I
0.3.1	16 Mar. 10	Chapter 4 inputs	UPMC
0.4	29 Mar. 10	UCY input and latest template	UCY
0.5	02 Mar. 10	Chapter 4.1 – for review	CEDO
0.6	15 Apr 10	Section 2.3.5 added	SL
0.7	21 Apr 10	Chapter 4 and 5 inputs by FHTW	FHTW
0.8	27 Apr 10	Chapter 3 and Introduction by UCY	UCY
0.8.1	28 Apr 10	Chapter 2 updated	Harpo, Sensory
0.8.2	30 Apr 10	Added conclusion	KI-I, FHTW
0.9	15 May 10	Final reviews and contributions	FHTW, SL
0.99	31 May 10	WP leader review and refinement	UCY
1.0	03 June 10	Finalization	FHTW, KI-I
1.1	09 Dec 10	Section about related EU projects added as the reviewers suggested	UCY

## Table of Content

Document Information .....	2
Version History .....	3
Table of Content.....	4
Table of Figures .....	10
1 Introduction .....	13
2 State of the Art in Assistive Technology .....	14
2.1 Input Methods and Tools for People with Physical Disabilities.....	14
2.1.1 Headpointers and Mouthsticks .....	14
2.1.2 Special Mice, Keyboards and Alternative Input Systems .....	15
2.1.2.1 Mice .....	15
2.1.2.2 Joysticks .....	17
2.1.2.3 Keyboards.....	18
2.1.3 On Screen Keyboards .....	21
2.1.3.1 Click-N-Type .....	21
2.1.3.2 Wivik .....	21
2.1.3.3 Oska Keyboard .....	22
2.1.3.4 SymWord .....	22
2.1.4 Increasing Typing Efficiency.....	23
2.1.5 Head Mice and Eye Gaze Systems .....	23
Head mouse interaction styles .....	24
2.1.5.1 Infrared head mice .....	24
2.1.5.2 Blink Eyes Control.....	25
2.1.5.3 Head Tracking with Gyro Sensors.....	26
2.1.6 Voice input .....	26
2.1.7 Switches and scanning systems.....	26
2.1.7.1 Examples of currently available switches .....	27
2.1.7.2 Switches and users .....	29
2.1.7.3 Hardware interfaces for switches .....	30
2.1.7.4 Scanning.....	31
2.1.7.5 Scanning in existing applications.....	35
2.1.7.6 Mouse control by one switch (One switch mouse).....	36
2.1.8 Touch screens.....	37
2.1.9 Wiimote .....	38
2.1.10 New input hardware systems .....	39
2.1.10.1 Touchpads .....	39
2.1.10.2 3D mice.....	39
2.2 Enhancing input .....	40

2.2.1	Word completion and word prediction .....	40
2.2.2	Abbreviation expansion .....	40
2.2.3	Disambiguation .....	41
2.2.4	Dasher .....	41
2.2.5	FASTY Project .....	42
2.3	Assistive Control Systems .....	42
2.3.1	Environment Control Systems .....	42
2.3.2	Wheelchair Control.....	44
2.3.2.1	PG Drive Technology .....	44
2.3.2.2	Dynamic Controls.....	44
2.3.3	Pneumatic Gripper for Mouthsticks.....	45
2.3.4	Robots as assistive devices .....	46
2.3.5	Robots as rehabilitation tools. ....	47
2.4	Communicators .....	47
2.4.1	Basic communication devices .....	47
2.4.2	Advanced communicators .....	48
2.4.3	PC based communication software .....	49
2.4.4	Other mobile communicators.....	49
2.5	Open Source and AT.....	49
2.5.1	OATS .....	51
2.5.2	Project: Possibility .....	51
2.6	Accessibility interfaces .....	51
2.6.1	Microsoft Active Accessibility.....	51
2.6.2	UI Automation .....	52
2.6.3	Other interfaces.....	52
3	Electrophysiological signals to enable computer input.....	54
3.1	EMG based systems .....	54
3.2	EOG based systems .....	56
3.3	EEG based systems.....	58
3.4	BNCI pattern recognition techniques .....	60
3.4.1	SoA in Pattern Recognition for BNCI based on BCI competitions.....	60
3.4.2	Computational Intelligence Methodologies for BCI .....	62
3.4.2.1	Feature Extraction and General Purpose Classification for BCI.....	63
3.4.2.2	Projection Techniques for BCI.....	64
3.4.2.3	Supervised Projection for BCI .....	65
3.4.2.4	Classifier ensembles for BCI .....	66
3.4.3	EEG and BCI toolkits.....	66
3.4.3.1	BCI2000.....	66
3.4.3.1.1	Stimulus Presentation .....	67
3.4.3.1.2	Feedback Demo.....	67

3.4.3.1.3	Cursor Task.....	67
3.4.3.1.4	P300 Speller.....	67
3.4.3.2	SPM8.....	68
3.4.3.3	EEGLAB.....	68
3.4.3.4	OpenEEG.....	69
3.4.3.5	Biosig.....	70
3.4.3.6	EEG/MRI Matlab Toolbox.....	70
3.4.3.7	EMEGS.....	70
3.4.3.8	OpenVibe.....	70
3.4.3.9	BCPy2000.....	71
3.4.3.10	Conclusions.....	71
4	Computer Vision for Assistive Technologies.....	72
4.1	Tracking of fingers, limbs or shoulder.....	72
4.1.1	Finger tracking.....	72
4.1.2	Upper Limb tracking.....	73
4.1.3	Shoulder & elbow tracking.....	74
4.2	Head and face tracking.....	74
4.2.1	Tracking of facial features.....	75
4.3	Eye- and gaze tracking.....	76
4.3.1.1	Remote eye trackers.....	76
4.3.1.2	Head-mounted eye trackers.....	77
4.3.2	Methods for computer vision based eye tracking.....	77
4.3.2.1	Limbus technique.....	77
4.3.2.2	Hirschberg technique.....	78
4.3.2.3	Video-oculography.....	79
4.3.3	Techniques for computer vision based gaze estimation.....	81
4.3.3.1	Feature-based gaze estimation.....	82
4.3.3.2	Appearance based gaze estimation.....	83
4.3.4	State-of-art in eye trackers.....	83
4.3.4.1	ISCAN ETL-500 remote eye-gaze tracker.....	84
4.3.4.2	Eyegaze Edge.....	84
4.3.4.3	MyTobii P10.....	85
4.3.4.4	I4Control System.....	86
4.3.4.5	OpenEyes.....	86
4.3.4.6	EyeWriter.....	87
4.3.4.7	ITU Gaze Tracker.....	88
4.3.4.8	EyeSeeCam.....	89
4.3.4.9	MobileEye.....	90
4.3.4.10	Intelligaze IG-30.....	90
4.3.5	Summary and comparison of eye trackers.....	91

4.4	Software support for Computer Vision .....	93
4.4.1	MATLAB related Computer Vision libraries .....	93
4.4.2	Image processing and Computer Vision Libraries .....	94
4.4.2.1	OpenCV .....	94
4.4.2.2	CVD Library .....	95
4.4.3	Numerical Libraries .....	95
4.4.3.1	LAPCK/BLAS .....	95
4.4.3.2	GSL.....	95
4.4.3.3	TooN Library .....	96
4.4.4	Computer Vision software for embedded platforms .....	96
4.5	Conclusion .....	97
5	State of the Art in Hardware .....	98
5.1	Hardware for the AsTeRICS embedded platform .....	98
5.1.1	IGEPv2.....	98
5.1.2	ISIS XL.....	99
5.1.3	Gumstix Overo Fire .....	100
5.1.4	BugLabs “BUG” platform .....	101
5.1.5	Bitsy-G5 .....	102
5.1.6	VIA EPIA-P710.....	103
5.1.7	Asus EeePC T101MT Netvertible.....	104
5.1.8	WePad Table PC .....	105
5.1.9	Kontron pico ITX-SP.....	106
5.2	Microcontrollers for interfacing sensors and actuators.....	107
5.2.1	PIC18 series of microcontrollers.....	108
5.2.2	Atmel AT90USB64/128 microcontrollers with USB support .....	109
5.2.3	Cypress microcontrollers with USB support.....	110
5.3	Wireless Interfaces.....	112
5.3.1	ZigBee and 802.15.4.....	113
5.3.2	BlueTooth.....	114
5.3.3	WiFi.....	114
5.3.4	Comparison of wireless technologies .....	114
5.4	Summary and Proposal.....	115
6	State of the art in Software Architectures .....	116
6.1	Introduction .....	116
6.2	Architectural Patterns .....	116
6.2.1	Layers .....	117
6.2.2	Component-based software .....	118
6.2.3	Service-Oriented Architectures – SOA .....	120
6.3	Middleware.....	122
6.4	Software Frameworks .....	122

6.4.1	Enterprise JavaBeans .....	123
6.4.2	OSGi .....	125
6.4.3	FRACTAL.....	127
6.4.4	Microsoft .NET Framework.....	128
6.5	Serialisation.....	130
6.5.1	XML-RPC.....	130
6.5.2	Protocol Buffers.....	130
6.5.3	Thrift.....	131
6.5.4	Comparison of Technologies.....	131
7	State of the Art in Relevant Areas of Software Engineering.....	132
7.1	Operating Systems for Embedded Platforms.....	132
7.1.1	Real-time Operating Systems and tiny RT-Kernels .....	132
7.1.2	Windows Derivatives .....	135
7.1.2.1	Windows Embedded Compact .....	135
7.1.2.2	Other Windows Embedded Versions.....	137
7.1.3	Palm-OS .....	137
7.1.3.1	Access Linux Platform + Garnet OS.....	137
7.1.3.2	Palm webOs.....	137
7.1.4	Symbian-OS.....	138
7.1.5	Linux Derivatives.....	139
7.1.5.1	Android .....	139
7.1.5.2	Maemo.....	141
7.1.6	Build Frameworks for Embedded Linux and Cross Compilation .....	141
7.1.6.1	BuildRoot .....	142
7.1.6.2	Chroot / Portage.....	142
7.1.6.3	Scratchbox.....	143
7.1.6.4	OpenEmbedded.....	144
7.1.6.5	Angström Linux .....	145
7.1.6.6	Poky Linux .....	145
7.2	Java Support and Java Virtual Machines.....	146
7.2.1	Java SE for Embedded Use .....	147
7.2.2	OpenJdk.....	148
7.2.3	JamVm.....	148
7.2.4	Cacao .....	149
7.2.5	PhoneME .....	149
7.2.6	Summary of Java Virtual Machines .....	150
7.2.7	Real-time Issues in Java .....	150
7.2.7.1	JamaicaVM .....	151
7.2.7.2	Aonix PERC .....	152
7.2.7.3	WebSphere Real Time .....	152



7.2.7.4	JAVA SE Real-Time Systems .....	152
7.2.7.5	Javolution.....	152
7.3	Graphical Modelling of Dynamic Systems and Signal Processing .....	153
7.3.1	Simulink (MathWorks) .....	154
7.3.2	LabView (National Instruments) .....	156
7.3.3	BioExplorer (CyberEvolution) .....	157
7.3.4	BioEra (Proatech LLC) .....	159
7.3.5	BrainBay .....	161
7.3.6	OpenVibe .....	165
7.3.7	Summary and Comparison.....	166
8	Relevant European Projects.....	167
8.1	TOBI: Tools for Brain Computer Interaction.....	167
8.2	BRAIN: Brain-computer interfaces with Rapid Automated Interfaces for Nonexperts 168	
8.3	BrainAble: Autonomy and social inclusion through mixed reality Brain-Computer Interfaces: Connecting the disabled to their physical and social world .....	169
8.4	DECODER .....	169
8.5	MUNDUS: Multimodal Neuroprosthesis for Daily Upper limb Support .....	170
8.6	Future BNCI .....	170
8.7	COGAIN: Communication by Gaze Interaction.....	171
8.8	I2HOME: Intuitive interaction for everyone with home appliances based on industry standards.....	171
9	Conclusions.....	173
	References.....	175

## Table of Figures

Figure 1:	Adjustable headpointer and clear view headpointer	15
Figure 2:	Mouthstick and telescope mouthstick	15
Figure 3:	3M ergonomic mouse .....	15
Figure 4:	Zero tension mouse .....	15
Figure 5:	Infogrip BIGTrack .....	16
Figure 6:	Traxsys Roller II.....	16
Figure 7:	NoHands Mouse.....	17
Figure 8:	LifeTool Integramaus.....	17
Figure 9:	nAbler Joystick .....	18
Figure 10:	The ErgoSplit keyboard .....	18
Figure 11:	Kinesis Advantage Pro .....	18
Figure 12:	Large size keyboard from Gorlo & Todt .....	19
Figure 13:	The BigKeys LX.....	19
Figure 14:	Magnetic keyboard .....	19
Figure 15:	Minkeyboard .....	20
Figure 16:	Maltron Single Handed Keyboard and Twiddler 2 .....	20
Figure 17:	Multi-purpose keyboard Lucy.....	20
Figure 18:	Darci USB.....	21
Figure 19:	The Click-N-Type on screen keyboard.....	21
Figure 20:	Wivik on screen keyboard.....	21
Figure 21:	Oska keyboard .....	22
Figure 22:	SymWord in virtual keyboard mode .....	22
Figure 23:	The word prediction system EMU .....	23
Figure 24:	SmartNav 4 .....	24
Figure 25:	TrackerPro from Madentec .....	24
Figure 26:	Blink-It .....	25
Figure 27:	b-Link keyboard.....	25
Figure 28:	Big Red Twist .....	27
Figure 29:	Micro Level Switches <sup>36</sup> .....	27
Figure 30:	Leaf Switch <sup>36</sup> .....	27
Figure 31:	Pillow Switch <sup>36</sup> .....	28
Figure 32:	Grasp Switch <sup>36</sup> .....	28
Figure 33:	Ribbon Switch <sup>36</sup> .....	28
Figure 34:	Switch 3.5 mm jack plug interface.....	30
Figure 35:	Sweety!.....	30
Figure 36:	PowerLink3 <sup>36</sup> .....	30
Figure 37:	Simple scan.....	31
Figure 38:	Row-column scan .....	32
Figure 39:	Column-row scan.....	32
Figure 40:	Nested scan .....	32
Figure 41:	Directed scan.....	33
Figure 42:	Autoscan .....	33
Figure 43:	Userscan .....	34
Figure 44:	Userscan with two switches .....	34
Figure 45:	Stepscan .....	34
Figure 46:	Userscan with one switch .....	35

Figure 47:	Column-row scanning in Oska keyboard.....	35
Figure 48:	Nested-group scanning in SymWord .....	35
Figure 49:	Magnified cell in simple scan in The Grid.....	36
Figure 50:	Moving One Switch Mouse left .....	36
Figure 51:	One switch mouse. Choosing left click.....	37
Figure 52:	The coordination system of the Wii Remote.....	38
Figure 53:	SpaceExplorer – 3D mouse from 3Dconnexion .....	39
Figure 54:	SymWord with prediction .....	40
Figure 55:	Tapir .....	41
Figure 56:	Dasher.....	42
Figure 57:	VSI and R-net Controls from PGDT .....	44
Figure 58:	DX-G90 and DX2 control form Dynamic Controls .....	45
Figure 59:	The prototype of the pneumatic gripper equipped mouthstick [3] .....	45
Figure 60:	The e3 gripper in use.....	46
Figure 61:	Handy, CEA (France) .....	46
Figure 62:	Manus (CEA, France, and Exact Dynamics Netherlands).....	46
Figure 63:	Virtuose robot for upper limb rehabilitation (CEA, France) .....	47
Figure 64:	BIGmack communicator .....	47
Figure 65:	GoTalk 20+.....	48
Figure 66:	SymWord.....	49
Figure 67:	UI Elements in UISpy.exe.....	52
Figure 68:	Main facial muscles .....	54
Figure 69:	A person wearing electrodes attached above the corrugator supercilii muscle site.....	55
Figure 70:	Ternary keyboard for electroocular spelling interface.....	56
Figure 71:	Hierarchical Clustering Procedure .....	57
Figure 72:	Overall accuracy for 13 users plotted over spelling 6 five-letter words .....	57
Figure 73:	Matrix of symbols used for the P300 protocol [177].....	61
Figure 74 :	(a) Large-scale electrode pairs between the SMA area and left/right. M1 areas. (b) Local-scale electrode pairs located over SMA and M1 areas [179]. .....	61
Figure 75:	Averaged ERD spatial mappings of (a) left hand and (b) right foot in the training set [180] .....	62
Figure 76:	Basic structure of ABI BCI software modules.....	69
Figure 77:	In-air typing system (University of Tokyo) .....	72
Figure 78:	Nose mouse, University of Ottawa and CNR, Canada. ....	75
Figure 79:	LipMouse actions.....	76
Figure 80:	Vision Key system (VisionCan, Canada).....	77
Figure 81:	VOG technique principle used in University Hospital of Munich eyetracking ..	79
Figure 82:	Eye detection methods .....	80
Figure 83:	Gaze estimation methods .....	82
Figure 84:	ISCAN ETL500 eyetracker built by University of Florida. ....	84
Figure 85:	Eyegaze Edge and Eyegaze Analysis .....	84
Figure 86:	MyTobii P10.....	85
Figure 87:	I4Control System, <i>Czech Technical University</i> (MEMREC project and EU FP6 IST COGAIN NoE). ....	86
Figure 88:	OpenEyes eyetracker, Iowa State University .....	87
Figure 89:	EyeWriter, Alhambra Hospital, USA.....	88

Figure 90:	ITU Gaze Tracker (COGAIN NoE, FP6, IST): remote (a) and mouth-mounted (b) setup .....	89
Figure 91:	EyeSeeCam .....	89
Figure 92:	ASL Mobile Eye .....	90
Figure 93:	IGEPv2 OMAP3 computing board .....	98
Figure 94:	ISIS XL Atom-based computing board .....	99
Figure 95:	Gumstix Overo Computer-On-Module .....	100
Figure 96:	BugLabs OMAP3-based BUG platform.....	101
Figure 97:	Bitsy-G5 MX31-based computing platform.....	102
Figure 98:	VIA EPIA P710 computing platform .....	103
Figure 99:	Asus EeePC T101MT netvertible.....	104
Figure 100:	WePad tablet PC .....	105
Figure 101:	Kontron Atom-based Single Board Computer .....	106
Figure 102:	Low pin count USB development kit including PICKit2 debugger [36] .....	109
Figure 103:	AT90USBkey development board [38].....	110
Figure 104:	CY3664 encore3 development kits [41] .....	112
Figure 105:	Diagram of the IEEE 802 wireless space .....	112
Figure 106:	The relationships of reference models, architectural patterns, reference architectures, and software architectures [112]. .....	116
Figure 107:	Visual overview of a Three-tiered application.....	118
Figure 108:	Collaborations in a SoA [124] .....	121
Figure 109:	Basic EJB Architecture .....	124
Figure 110:	OSGi Class Loader Graph .....	125
Figure 111:	The OSGi Bundle lifecycle .....	126
Figure 112:	An example of a FRACTAL component [122] .....	128
Figure 113:	Overview of the Common Language Infrastructure <sup>159</sup> .....	129
Figure 114:	Illustration of preemptive scheduling [139].....	134
Figure 115:	Windows Embedded Compact Kernel Architecture [145].....	136
Figure 116:	OAL Architecture [146] .....	136
Figure 117:	Architecture of Palm webOs [150].....	138
Figure 118:	Android OS Architecture <sup>170</sup> .....	140
Figure 119:	Screenshot from BuildRoot configuration menu .....	142
Figure 120:	Sato Desktop managed by Matchbox window manager.....	146
Figure 121:	Simulink graphical editor and oscilloscope window [155] .....	154
Figure 122:	Interfacing Matlab/Simuling with embedded systems [157].....	155
Figure 123:	LabVIEW VI editor for system design [159].....	156
Figure 124:	LabVIEW Arm Cortex-M3 (Luminary Micro) and Arm7 (NXP) prototyping boards .....	157
Figure 125:	Bioexplorer design and monitoring windows .....	158
Figure 126:	BioEra design window and element properties .....	160
Figure 127:	BrainBay design and runtime windows .....	161
Figure 128:	BrainBay camera element properties, live camera view with feature detection.....	162
Figure 129:	BrainBay camera mouse design .....	163
Figure 130:	Using the Autonomy environmental control system via BrainBay camera mouse.....	164
Figure 131:	BrainBay speller and word prediction elements .....	164
Figure 132:	2D- and 3D-topographical mapping of EEG data using OpenVibe [171] .....	165
Figure 133:	P300 BCI application designed with OpenVibe [170] .....	166

# 1 Introduction

The purpose of Work Package 2 of the AsTeRICS project -“System Architecture and Specification” - is to define in detail the components of a construction set for enabling easy-to-use Assistive Technology, based on user requirements and needs. In particular, new developments in the sector of Information and Communication Technologies for assistive tools to enhance motor capabilities and accessibility in the digital realm will be explored and integrated into the project. This document aims to identify state-of-the-art hardware and software components and development methodologies found in related systems and the literature, and to evaluate their reusability for the AsTeRICS project.

## Relationship to other AsTeRICS deliverables

The deliverable is related to the following AsTeRICS deliverables:

- D2.1 – System Specification and Architecture (for Prototype 1)
- D2.2 – Updated System Specification and Architecture (for Prototype 2)

The results reported in this deliverable will be considered during design and implementation of the system, which will be documented in D2.1 and D2.2.

## Relationship to the Description of Work

According to the Description of Work (DoW)<sup>1</sup>, AsTeRICS will provide a flexible and affordable construction set for user driven Assistive Technologies (AT) or assistive functionalities. The technologies envisioned by AsTeRICS require harmonious interaction of various software and hardware components. Therefore, this deliverable aims to provide the required technology analysis and evaluation of related state-of-the-art systems and technologies. The results reported in this deliverable shall be used as guidelines for the design and implementation decisions reported in D2.1 and D2.2.

## Structure of this document

The rest of this deliverable is structured as follows: Section 2 provides an overview of the state-of-the-art in Assistive Technology. Section 3 presents solutions for enabling computer input through Electrophysiological signals, while Section 4 presents the state of the art in computer vision for Assistive Technologies. Section 5 presents the state-of-the-art in hardware platforms considered relevant to the projects hardware requirements. Section 6 presents relevant system architecture principles and evaluates existing software architecture technologies. Section 7 provides a comprehensive overview of software technologies in the sectors of signal processing and operating systems for embedded platforms. Section 8 reviews relevant European projects related to AsTeRICS so we can avoid overlap and maximize knowledge reuse during the project. Finally, Section 9 concludes this document with an overview of the main contributions of this deliverable to WP2 and to the AsTeRICS project in general.

---

<sup>1</sup> Description of Work, technical annex 1 of the AsTeRICS Grant agreement

## 2 State of the Art in Assistive Technology

Assistive technology (AT) can be defined as any item, equipment or application that is used to increase, maintain, or improve the functional capabilities of disabled people. AT products can help people with disabilities in performing daily living tasks, communication, education or work. AT products give these people alternative access to advanced electronic devices like computers or mobile phones. They help them to achieve greater independence and improve their quality of life. This chapter describes, mainly, current trends in AT for motor disabled people. Motor disabled people have difficulties moving their body which could range from limitations in movement of a particular limb to total paralysis. . Motor disability can be caused by problems with power or tone of the muscles, control of complex voluntary movements, coordination of voluntary movements, mental functions responsible of sequencing complex movements. Motor disability may be the result of such diseases as: spinal cord injury, traumatic brain injury, neurological disorders, cerebral palsy, multiple sclerosis, muscular dystrophy, stroke, ALS (amyotrophic lateral sclerosis), cranial trauma or paralysis (see AsTeRICS Description of Work<sup>1</sup>).

Assistive technology products should meet all of the needs of disabled users but, despite of a wide range of AT products on the market, there are user needs which are not fully met by AT technology. AT developers are still improving their products and seeking a new way to help the users.

An important field in AT technology is to develop access to computers for people who are not able to use the computer mouse and keyboard in a standard way, as computers and networking technologies can greatly enhance the potential of people with disabilities to fully participate in modern society. This chapter describes various AT systems for alternative human computer interaction (HCI), e.g.: switch systems, headpointers, head mice, eye gaze systems, voice input, brain systems (EEG), which aim to meet this need.

### 2.1 Input Methods and Tools for People with Physical Disabilities

#### 2.1.1 Headpointers and Mouthsticks

Headpointers and mouthsticks are very simple but helpful tools mostly on a low technical level. Most of them are pure mechanical devices. While headpointers are mounted on the head of a person, mouthsticks have a mouthpiece adapted to the mouth and teeth of the user. Depending on the capabilities of the using person, the pointers and sticks can be used to use

- standard mice/keyboards,
- adapted mice/keyboards,
- trackballs and
- touchscreens.

The following figures show a selection of several headpointers and mouthsticks:



Figure 1: Adjustable headpointer and clear view headpointer<sup>2</sup>



Figure 2: Mouthstick and telescope mouthstick<sup>3</sup>

## 2.1.2 Special Mice, Keyboards and Alternative Input Systems

### 2.1.2.1 Mice

Many persons with physical disabilities cannot use standard mice. Therefore, many alternatives are available. Beside adapted mice, the usage of trackballs is often very helpful.

#### Ergonomic Mice

Ergonomic mice have the advantage of avoiding exhaustion and allowing gently usage. Because of their special shape, they sometimes can be used by persons with problems at their hands or wrists (e.g. Repetitive Strain Injuries - RSI). Two, of many interesting, products are the 3M ergonomic mouse (see Figure 3) and the Zero-tension mouse (see Figure 4).



Figure 3: 3M ergonomic mouse<sup>4</sup>



Figure 4: Zero tension mouse<sup>5</sup>

<sup>2</sup> <http://www.wisdomking.com/headpointers>

<sup>3</sup> <http://www.wisdomking.com/mouth-sticks>

<sup>4</sup> [http://solutions.3m.com/wps/portal/3M/en\\_US/ergonomics/home/products/ergonomicmouse/](http://solutions.3m.com/wps/portal/3M/en_US/ergonomics/home/products/ergonomicmouse/)

<sup>5</sup> <http://www.zerotensionmouse.com>

## Trackballs

Trackballs can be used within a very small movement area. Some of them can be used with the whole hand, others with a mouthstick.

### Infogrip BIGTrack

The ball to the BIGTrack has a diameter of 7.5 cm, so it can be used by a finger, by the whole hand or with a mouthstick. The two buttons are separated from the ball, so an unintentional button press will be avoided. Additionally, external sensors can be connected in one model. The case is very robust. The BIGTrack will be connected via an USB connection, for the usage and configuration, the standard mouse drivers and controls are used.



Figure 5: Infogrip BIGTrack<sup>6</sup>

### Traxsys Roller II Trackball

Beside the 5cm ball and the two standard buttons, the Traxsys Roller II has an additional “drag&drop” button, there drag&drop operations can be permitted without pressing this button the whole time. Similar to the BIGTrack, it is connected by USB and uses the standard mouse controls.



Figure 6: Traxsys Roller II<sup>7</sup>

<sup>6</sup> <http://www.infogrip.com/>

<sup>7</sup> <http://www.traxsys.com/AssistiveTechnology/Trackballs/tabid/1404/Default.aspx>



## NoHands Mouse

The NoHands Mouse is a foot operated mouse that consists of two separate pedals. One pedal operates the pointer, while the other is used for button clicking. Similar to other mice and joysticks, it is recognised by the PC as a standard mouse.



Figure 7: NoHands Mouse<sup>8</sup>

### 2.1.2.2 Joysticks

Joysticks give the opportunity of using them with very low effort and force and need just a very small area of movement. So called zero-way joysticks need hardly any force to be used and can therefore also be used by persons with very limited mobile abilities. They also can be used in a lying position.

### Lifetool Integramaus

The Integramaus is a zero-way joystick, controlled with the mouth. The mouse buttons are triggered by sipping and puffing. It will be connected via USB to the PC and will be recognised as standard HID mouse.



Figure 8: LifeTool Integramaus<sup>9</sup>

<sup>8</sup> <http://www.footmouse.com/nohands.php>

<sup>9</sup> [http://www.lifetool.at/show\\_content.php?sid=123](http://www.lifetool.at/show_content.php?sid=123)

## nAbler Joystick

The nAbler joystick is a very massive and robust device which substitutes the mouse and provides additional functions.



Figure 9: nAbler Joystick<sup>10</sup>

### 2.1.2.3 Keyboards

Beside the standard keyboards, a huge amount of adapted or special keyboards are available in order to fit the requirements and needs of the user.

They are categorised in

- ergonomic keyboards,
- big keys and large size keyboards,
- mini and macro keyboards,
- one hand keyboards,
- virtual or on-screen keyboards,
- other special keyboards

### Ergonomic Keyboards

Ergonomic keyboards have the advantage of avoiding exhaustion and allowing gently usage. Additionally, some of them have been developed for persons with hand or wrist problems.



Figure 10: The ErgoSplit keyboard<sup>11</sup>



Figure 11: Kinesis Advantage Pro<sup>12</sup>

<sup>10</sup> <http://www.pretorianuk.com/Language/UK/assistive.html>

<sup>11</sup> <http://www.comfortkeyboard.com/ErgoSplit%20Keyboard.html>

## Big Keys and Large Size Keyboards

This kind of keyboards will be used for persons with coordination problems, no proper fine motor skills, tremor or strong spasm. Many of these keyboards have additional track support to prevent unintended multiple key presses. Some keyboards have colored keys, so that persons with perception disorders or cognitive impairments can use them more easily.



Figure 12: Large size keyboard from Gorlo & Todt<sup>13</sup>

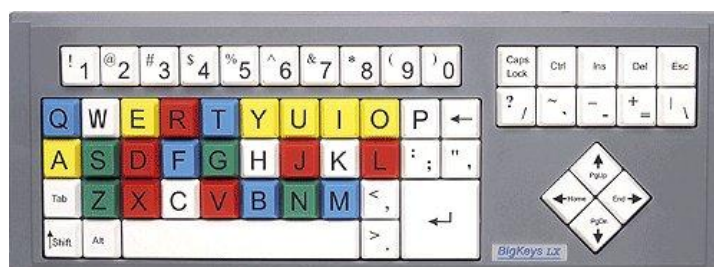


Figure 13: The BigKeys LX<sup>14</sup>

## Mini and Macro Keyboards

Mini and macro keyboards are for persons with good fine motor skills, but with no big range. Users might be persons with dystrophy, atrophy, multiple sclerosis, rheumatism, arthritis and users of mouthsticks. Pressing the keys needs a very low effort; some special magnetic keyboards can be used with a magnetic pointer.



Figure 14: Magnetic keyboard<sup>15</sup>

<sup>12</sup> <http://www.kinesis-ergo.com>

<sup>13</sup> <http://www.gorlo-todt.de/grossfeldtastaturen.htm>

<sup>14</sup> <http://www.bigkeys.com/>

<sup>15</sup> <http://www.gorlo-todt.de/produkte.htm>

## Finger-operated Minikeyboard



Figure 15: Minikeyboard<sup>16</sup>

## One Hand Keyboards

These keyboards have been developed for persons who just can use one hand. A lot of different products are available, some are optimised to use all keys with one hand, others just have a few keys for entering key combinations (“chords”).



Figure 16: Maltron Single Handed Keyboard and Twiddler 2<sup>17</sup>

## Lucy

Lucy is a multi-purpose and flexible keyboard, which can be controlled by laserpointer, pointer, joystick or two switch scanning; the user chooses the best fitting method for her/him. Lucy can be used as an input system, as environmental control and as communication device.

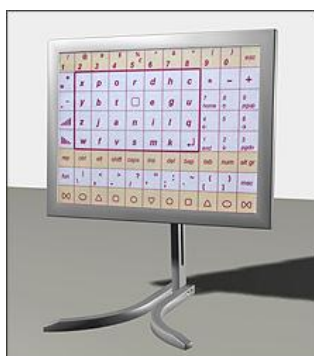


Figure 17: Multi-purpose keyboard Lucy<sup>18</sup>

<sup>16</sup> <http://www.computer-fuer-behinderte.de/produkte/2tast-minitastatur.htm>

<sup>17</sup> <http://www.maltron.com/maltron-kbd-single.html> and <http://www.handykey.com/>

## Darci USB

Darci USB is a “kind of keyboard”, using Morse codes as input channel. Two external sensors can be connected to Darci, encoding the Morse code and transmitting the characters to the PC.



Figure 18: Darci USB<sup>19</sup>

### 2.1.3 On Screen Keyboards

On screen keyboards are virtual keyboards, being presented on a display and controlled with assistive technology. Most of these keyboards are very adaptive and configurable, fitting the user needs.

#### 2.1.3.1 Click-N-Type



Figure 19: The Click-N-Type on screen keyboard<sup>20</sup>

Click-N-Type<sup>20</sup> is a freeware product supporting graphical scanning. The scanning process will be controlled with the two mouse buttons. The layout and scanning settings are quite adaptable and a basic word prediction system is included.

#### 2.1.3.2 Wivik

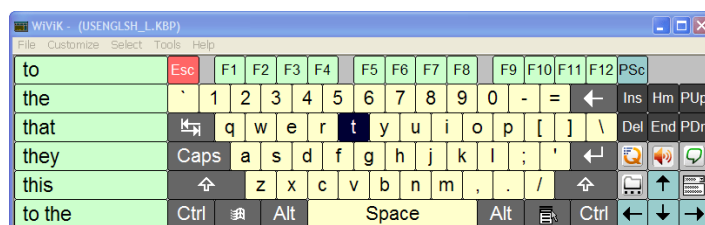


Figure 20: Wivik on screen keyboard<sup>21</sup>

<sup>18</sup> [http://skilate.com/fs\\_assisttech\\_prod5\\_readable.html](http://skilate.com/fs_assisttech_prod5_readable.html)

<sup>19</sup> <http://www.westest.com/darci/usbindex.html>

<sup>20</sup> <http://cnt.lakefolks.org/>

The on screen keyboard Wivik<sup>21</sup> is very popular and well supported by assistive technologies. Besides pointing devices, several scanning modes are supported, including the usage of up to six switches. Several languages are supported. Additionally, the advanced word prediction system WordQ is included.

### 2.1.3.3 Oska Keyboard

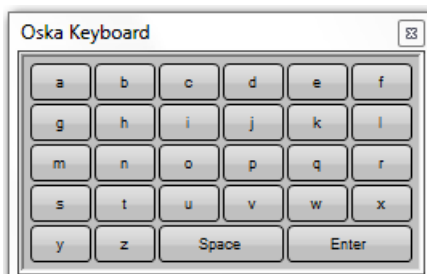


Figure 21: Oska keyboard

The Oska keyboard<sup>22</sup> is an on screen keyboard solution which is a part of the Oska Suite, a software solution range designed to assist people who have difficulty using traditional computer input methods. The Oska keyboard software allows the use of multiple input devices including mouse, joystick, head pointer and switch/scanning methods to generate keyboard input. Using the Oska Editor, a user is able to create his own keyboard layout. The user is able to control the choice of keys on the keyboard using scanning and alternative interfaces like: numeric keypads, mice, gamepads and touchpads.

### 2.1.3.4 SymWord



Figure 22: SymWord in virtual keyboard mode

SymWord<sup>23</sup> is a talking word processor for younger learners and computer users with literacy difficulties. It can work also as an on screen keyboard in virtual keyboard mode. In this mode the contents of SymWord cells are sent to another application. Users can set a transparency option for SymWord allowing users can see applications located under the SymWord window. Using the additional application GridEd, the user can create his/hers own grids. The cells on the grids can contain not only the letters but also words, sentences and control keys. The user can change the grids in an easy way and choose the grid which he/she needs. SymWord contains an adaptive scanning system, prediction dictionary and can also be used as an email client.

<sup>21</sup> <http://www.wivik.com/>

<sup>22</sup> <http://www.oskaworld.com/>

<sup>23</sup> <http://www.clarosoftware.com/index.php?cPath=372>

### 2.1.4 Increasing Typing Efficiency

Persons with mobility disabilities are notwithstanding optimal fitting AT often not very fast input text to an electronic system. Word prediction systems can increase the input speed by suggesting words, depending on the already captured letters. Some systems have this functionality already integrated, but there are also stand alone systems, for example the word prediction system EMU, which is a multi language system with the possibility of using different dictionaries (including the option to expand them)

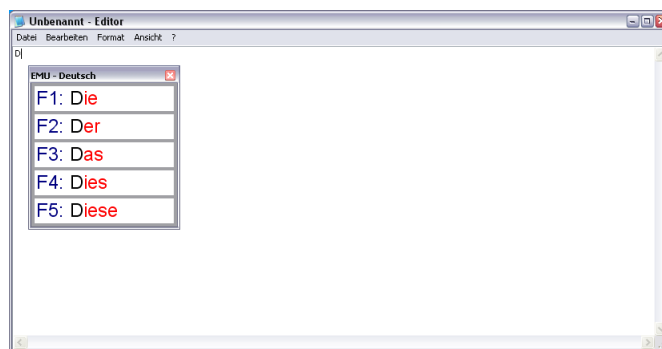


Figure 23: The word prediction system EMU<sup>24</sup>

### 2.1.5 Head Mice and Eye Gaze Systems

The head mice and eye gaze systems are alternative ways of operating computers for people who cannot use their upper limbs or hands. These systems detect head movement, face movement, eye movement or eye-blinks and utilize this information to control the computer. A head mouse replaces the standard mouse and is particularly useful for upper limb impaired subjects. This peripheral can connect to a (fixed or portable) host system via an USB port or wirelessly. Common mouse operations are performed with a head mouse as follows:

- Mouse spatial displacements are deduced from head movements (orientation and scale)
- Mouse clicks can be implemented via adaptive switches such as sip/puff switch (many head mice have the option to connect external sensors, which are used to trigger mouse presses.)
- The usage of dwelling: a mouse press will be triggered, if the user is staying for a defined time on an object (additional software is needed)

Used in connection with an on-screen keyboard, head mice allow head-controlled access to all important functions of a computer, including internet access. The user can thus perform tasks as drawing, gaming, graphics applications such as Computer Aided Design (CAD). To determine head- or eye movements, various technologies are utilized, in particular:

- Acceleration and tilt sensors (Inertial Measurement Units, IMUs)
- IR-sensors and IR cameras
- Computer vision techniques and algorithms
- Measurement of bioelectric signals

<sup>24</sup> <http://www.is.tuwien.ac.at/emu/>

Systems using computer vision for head- or eye tracking are covered in section 4, whereas systems using bioelectric measurement are covered in section 3 of this document. The following section gives a general overview on head mouse operation and interaction styles and covers IR-based systems and IMU-based systems.

### Head mouse interaction styles

A head-operated device can be used in two ways: as an absolute pointing device or as a relative pointing device. In the absolute pointing mode each orientation of the user's head corresponds to a specific mouse pointer position on the screen. If the mouse pointer is in the middle of the screen and the user wishes to select an icon in the top left of the screen, the user points his/her head in that direction. When the pointer is over the icon, the user keeps his/her head in that position until the selection is made. The relative pointing device works similar to a hand-operated joystick. Like a joystick there is a neutral position, called deadband. When the user's head is in a central position it is in the deadband. In this position the mouse pointer does not move. To move the mouse pointer, the user moves his/her head out of the deadband in the desired direction and holds his/her head in this position until the mouse pointer is in the desired position. At this point the user returns his/her head to the deadband. According to [4] the relative pointing device may be preferred by many users.

#### 2.1.5.1 Infrared head mice



Figure 24: SmartNav 4<sup>25</sup>



Figure 25: TrackerPro from Madentec<sup>26</sup>

Infrared mice use infrared (IR) sensors to track user body movements. They reflect IR light from the reflective dot and then send the appropriate instructions to computer to move the mouse cursor. The dot can be placed anywhere on the user body for example: forehead, hand or glasses.

Some examples of Infrared mice:

- SmartNav<sup>27</sup>
- HeadMouse extreme<sup>28</sup>
- TrackerPro<sup>29</sup>

<sup>25</sup> <http://www.naturalpoint.com/smarnav/products/4-at/>

<sup>26</sup> <http://www.madentec.com/products/tracker-pro.php>

<sup>27</sup> <http://www.naturalpoint.com/smarnav/>

<sup>28</sup> <http://www.orin.com/access/headmouse/>

<sup>29</sup> <http://www.madentec.com/products/tracker-pro.php>



The mouse clicking can be done by additional software, switches connected to the mouse (SmartNav and TrackerPro), voice (SmartNav) or dwell time (SmartNav). Infrared mice typically use absolute position interaction; however, SmartNav is also able to use relative position interaction.

### 2.1.5.2 Blink Eyes Control

If the motor condition of the user does not allow the user to control head mice, the user can try to control the computer by blinking the eyes – here the blinks are used in a similar way to switch presses. Examples of such systems are: Blink-IT<sup>30</sup> and b-Link<sup>31</sup>.

#### Blink-It

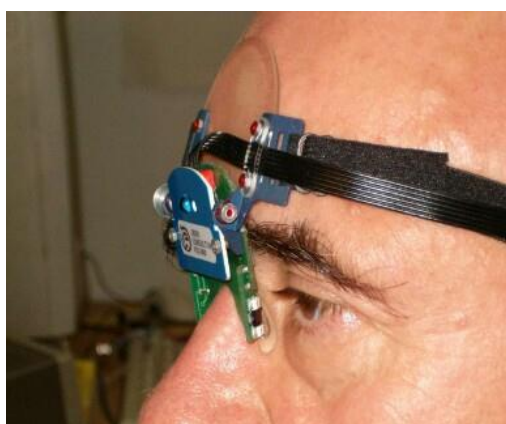


Figure 26: Blink-It

The Blink-It system is the result of the co-operation between the Institute of Biocybernetics and Biomedical Engineering PAS and Ober Consulting Poland. The Blink-It detects eyelid blinks using IR sensors. It works with the Mrugopis and the MrugoMysz applications. Mrugopis is the program for taking notes. The user, making blinks, can insert letters from the scanned grid to the text field. Using MrugoMyszka the user is able to move the mouse, simulating the mouse click, and to insert text into another application. When the user is using the MrugoMyszka the icons representing the action are scanned to the right of the mouse cursor. When the desired icon is displayed the user blinks to execute the icon action.

#### b-Link



Figure 27: b-Link keyboard

<sup>30</sup> <http://www.ober-consulting.com/product/blink-it/>

<sup>31</sup> [http://www.tp.pl/prt/pl/o\\_nas/edukacja/687688/](http://www.tp.pl/prt/pl/o_nas/edukacja/687688/)

The b-Link is a free and open source application. It detects user blinks using the standard webcam. The user, making blinks, selects letters for the column-row auto-scanned grid. Using b-Link a user is also able to simulate mouse moves and mouse clicks.

### 2.1.5.3 Head Tracking with Gyro Sensors

One or several gyro sensors are fixed on the head, receiving the movement of the head and transmitting it to the computer, where software calculates the movements and controls the mouse cursor. GyroTrackPro<sup>32</sup> is an example for such a product. [5] gives more information on this topic.

## 2.1.6 Voice input

Voice input can be a convenient alternative method of computer control for people with motor disabilities who have good speaking skills. Voice recognition software converts sentences spoken by the user into text. In addition many speech recognition applications allow users to control the computer with their voice, for example they can: move the mouse, emulate the mouse click, start and closes application, minimize and maximize applications, change applications, browse the web or send emails. Current speech recognition systems use different adaptive algorithms to learn the user voice and increase its accuracy. Windows Vista and Windows 7 include speech a recognition option, which allows users to dictate text and control the windows environment with his/hers voice. The Windows speech recognition option currently is available only for several languages: English, Spanish, German, French, Japanese and Chinese.

Example of other speech recognition applications, which give the user option to control the computer environment by voice are:

- Simon<sup>33</sup> is free and open-source application
- E-Speaking<sup>34</sup>
- Dragon NaturallySpeaking<sup>35</sup>

## 2.1.7 Switches and scanning systems

For the past three decades switches have been used by people with severe physical disability and also by people with cognitive disabilities to access computers. For people with certain physical disabilities switches (normally between 1 and 5, with appropriate software) can enable them to control their computer. For people with severe cognitive disabilities one or two switches can provide basic access to educational software. This section explains the types of switches available; and the (physical) disabilities for which different switches may be useful. It then goes on to explain current hardware interfaces for switches and how switches can be used with scanning software to facilitate alternative methods of keyboard entry and mouse control. Finally this section describes how scanning is implemented in some existing systems.

---

<sup>32</sup> <http://www.vrealities.com/gyrotrackpro.html>

<sup>33</sup> <http://simon-listens.org/index.php?id=128&L=1>

<sup>34</sup> <http://www.e-speaking.com/index.htm>

<sup>35</sup> <http://www.nuance.co.uk/naturallyspeaking/>

Quite simply, a switch is a device for making an electrical connection between two contacts. It is packaged in many forms and, for a physically disabled person, can replace a computer keyboard and mouse.

Depending on the specifics of their impairments some users are able to operate just one switch, others can operate more switches. Switches can be activated with different body parts, not only with the hands, for example using the head or legs. There are also switches activated using sip-and-puff technology, small muscle movements, eye blinks, finger movements and other forms of movement.

### 2.1.7.1 Examples of currently available switches



Figure 28: Big Red Twist<sup>36</sup>

The most common switches are activated by pressing anywhere on the top surface. They may vary in colours, size and the pressure needed to activate. Examples include:

- Big Red twist (AbleNet)<sup>37</sup>
- Jelly Bean twist (AbleNet)<sup>38</sup>
- Moon Switch (AMD)<sup>39</sup>
- The Micro Level Switch (AbleNet), which is activated by pressing down on the top of the switch with a feather light touch<sup>40</sup>.



Figure 29: Micro Level Switches<sup>36</sup>

- Leaf Switch (AbleNet) is activated by pressing lightly against the leaf in one direction<sup>41</sup>.



Figure 30: Leaf Switch<sup>36</sup>

<sup>36</sup> <http://www.ablenetinc.com>

<sup>37</sup> <http://www.ablenetinc.com/Store/tabid/205/Default.aspx?CategoryCode=138>

<sup>38</sup> <http://www.ablenetinc.com/Store/tabid/205/Default.aspx?CategoryCode=142>

<sup>39</sup> <http://www.amdi.net/store/moon-switch.html>

<sup>40</sup> <http://www.ablenetinc.com/Store/tabid/205/Default.aspx?CategoryCode=151>

<sup>41</sup> <http://www.ablenetinc.com/Store/tabid/205/Default.aspx?CategoryCode=148>

- Pillow Switch (AbleNet) has a smooth and soft foam surface making it suitable for head or cheek activation.



**Figure 31: Pillow Switch**<sup>36</sup>

- Grasp Switch (AbleNet) is activated with a squeeze or pinch. User must be able to release the grip in approximately 2 seconds to prevent the switch from latching<sup>41</sup>.



**Figure 32: Grasp Switch**<sup>36</sup>

- Ribbon Switch (AbleNet) is activated by bending in either direction<sup>42</sup>



**Figure 33: Ribbon Switch**<sup>36</sup>

- SCATIR Switch (AbleNet) can be controlled with an eye-blink, eyebrow movement, finger movement, head movement or facial muscle movement<sup>43</sup>.
- IntegraSwitch (LifeTool) is activated by sip and puff<sup>44</sup>.
- Picture Plate Switch (AMDi) requires only a few grams of pressure to activate. Its target area has a slot for pictures<sup>45</sup>.
- The Twitch Switch (Enabling Devices) is activated by small muscle movement, such as wrinkling the forehead. It can be easily adjusted to increase or decrease the amount of muscle movement needed to activate it<sup>46</sup>.
- Eye Blink Switch (Enabling Devices) is activated by eye blink<sup>47</sup>.

#### Some of the switch manufacturers:

- AbleNet<sup>36</sup>
- AMDi<sup>48</sup>
- Enabling Devices<sup>49</sup>

<sup>42</sup> <http://www.ablenetinc.com/Store/Ribbon+Switch/tabid/205/Default.aspx?ItemCode=55870>

<sup>43</sup> <http://www.ablenetinc.com/Store/tabid/205/Default.aspx?CategoryCode=146>

<sup>44</sup> <http://www.ablenetinc.com/Store/tabid/205/Default.aspx?CategoryCode=141>

<sup>45</sup> <http://www.amdi.net/store/picture-plate-switch.html>

<sup>46</sup> [http://enablingdevices.com/catalog/capability\\_switches/hand-finger-body-switches/the-twitch-switch](http://enablingdevices.com/catalog/capability_switches/hand-finger-body-switches/the-twitch-switch)

<sup>47</sup> [http://enablingdevices.com/catalog/capability\\_switches/hand-finger-body-switches/eye-blink-switch](http://enablingdevices.com/catalog/capability_switches/hand-finger-body-switches/eye-blink-switch)

<sup>48</sup> <http://www.amdi.net/>

Some switches provide feedback after activation. Feedback may be auditory, for example when the user hears a click sound, or tactile, for example when the user feels resistance or vibrations. The feedback is often important for people with cerebral palsy because they often have poor proprioception. They do not have a sense of where their limbs are and what they are doing. The feedback helps the user to know if the switch is activated without having to look at it.

### 2.1.7.2 Switches and users

Because of the nature of certain disabilities, many switch users needs are not static. It is important to note that at some point the individual may need one switch, and then after some time, the same individual would need a different switch. A good example is muscular dystrophy: in the beginning users may need a switch that is activated in the standard way, by pressing it, but when the dystrophy is in later stages, the same user may no longer able to use this switch and needs to change it. The timing of the change will usually depend on the stage of the disease.

Also, the methodology of special education points out that at first an individual needs something simple, that will give him a feeling of success and motivate him to work. That is why the first switches and software provided may vary from those provided at subsequent stages. This gives more and more opportunities for the users and leads to a more independent use of computers.

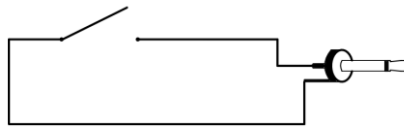
The following table presents examples of switches and their potential users. The key to the success of any switch is often individualization, i.e. finding the right solution for a given individual – hence this table should be treated only as an outline showing different possibilities:

Disability	Main indication for using switches	Switches
Cerebral palsy – athetoid	Involuntary motions, difficulty putting the hand in certain positions	Pillow Switch, Leaf Switch, Ribbon Switch, Micro Light Switch, Jelly Beamer
Cerebral palsy – spastic	Difficulty opening the hand and using a finger to point, strabismus, uncontrollable shaking	Grasp Switch, Pillow Switch, Leaf Switch, Big Switch, Moon Switch, Specs Switch, Jelly Beamer, Micro Lever Switch
Muscular dystrophy – mainly Duchenne and Becker type	Muscular weakness, difficulty with pressing and pushing objects, difficulty raising hands	Sip Puff Switch (IntegraSwitch), Micro Light Switch, Jelly Beamer, Pillow Switch, Twitch Switch, Eye Blink Switch
Down's syndrome	Coordination troubles, impairment of cognitive ability, delayed fine motor skills	Moon Switch, Jelly Bean, Big Switch, Picture Plate Switch
Autism	Hypotonia (poor muscle tone) apraxia (poor motor planning)	Big Switch, Jelly Bean, Moon Switch, Picture Plate Switch
Intellectual disability – severe and moderate	Impairment of cognitive ability	Picture Plate Switch, Moon Switch, Jelly Bean, Big Switch
Paralysis	Inability to move hands and/or legs	Sip Puff Switch (IntegraSwitch), Pillow Switch, Twitch Switch, Eye Blink Switch

**Table 1: Switches and users**

<sup>49</sup> <http://enablingdevices.com>

### 2.1.7.3 Hardware interfaces for switches



**Figure 34: Switch 3.5 mm jack plug interface**

There are switches which can be connected to computer directly using a USB interface, but the most common switches have a 3.5mm jack plug. To connect them to the computer you have to use a switch interface. The simplest interface is the mouse with two sockets for switches. Switches replace the left and right mouse buttons.

There are also more advanced switch interfaces, for example Sweety or IntelliSwitch:

Sweety (by AsTeRICS partner Harpo) uses a Bluetooth interface to connect to the PC. Sweety can handle up to five switches. Using the Sweety software you can set actions for each switch, for instance to simulate keyboard key press or to simulate a mouse button click<sup>50</sup>.



**Figure 35: Sweety!**

IntelliSwitch (Madentec) can be connected to the PC by using a ZigBee transmitter or by USB cable. IntelliSwitch has five jack sockets for switches. Using the IntelliSwitch software you can configure switches in one of twenty six possible ways. Additional software extends the switch's configuration capabilities<sup>51</sup>.

There are also switch interfaces which can control electrical devices, for example PowerLink3.



**Figure 36: PowerLink3<sup>36</sup>**

<sup>50</sup> <http://www.isable.eu/isableshop/shopexd.asp?id=16>

<sup>51</sup> <http://www.madentec.com/products/intelliswitch.php>

The PowerLink4 (AbleNet) gives the user the ability to turn on and turn off two electrical devices with single switches. Depending on settings, the user can activate a device for as long as the switch is activated, or each activation will turn the device on and off sequentially. The user can also turn on the devices for a defined time, after which the device is turned off automatically<sup>52</sup>.

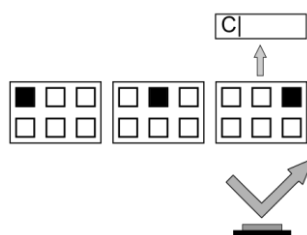
#### 2.1.7.4 Scanning

The term scanning in AAC technology is used to describe the process of sequential display of a set of items on a screen, and selection of an item from the set using a small number of switches. The scanning is often implemented in the grid based programs or in the virtual keyboard programs. The active item or group of items is highlighted. Highlighted items are usually visually different from the rest of items. They may have a different colour or a border. Scanning can be described with scanning movements and scanning methods.

#### Scanning movements

The scanning movement defines how the highlighter moves from one item or group to another. There are three common scanning movements: simple scan, group scan and directed scan.

- Simple scan



**Figure 37: Simple scan**

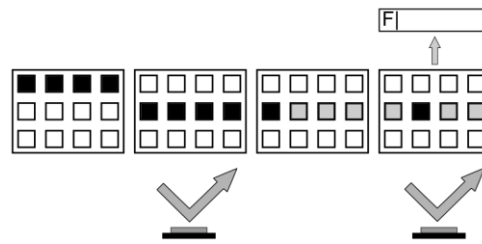
In this scan movement items are highlighted in turn, one item at one time. When the desired item is reached the user can select it using a switch. The items can be arranged in rows or columns. Simple scan is good for small numbers of items. With larger sets of more than 12 items it is slow [30]. Simple scan is suitable for people with a learning disability because it is easy to operate.

- Group scan

In the group scan the selection set is divided into groups of items and these are then individually highlighted. The user first selects the group containing the desired item, then, depending on the group scan type, items in the group are scanned one at time (simple scan) or the selected group is divided into further subgroups which are still individually highlighted. The subgroups are scanned until the user reaches the level of single items. When the item is selected the scanning will go back to the top level groups. The group scan is faster than the simple scan for larger sets of more than 12 items [30]. There are several types of group scan: row-column, column-row and nested.

<sup>52</sup> <http://www.ablenetinc.com/Store/tabid/205/Default.aspx?CategoryCode=117>

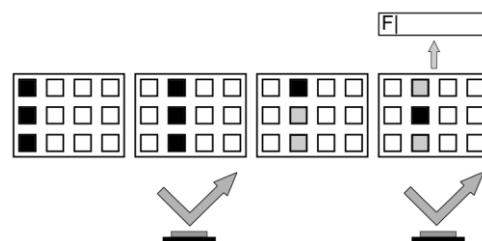
- Row-Column scan



**Figure 38: Row-column scan**

Row-column scanning is widely used in AAC technology. To select an item the user first scans rows to the one containing the desired item, selects it, then scans to select the item in that row.

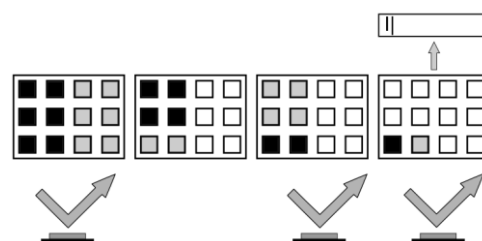
- Column-Row scan



**Figure 39: Column-row scan**

Column-row scanning is often used for sets containing word lists because many people find it easier to spot a particular word in a column rather than a row [30]. To select an item the user first scans columns to the one containing the desired item, selects it, then scans to select the item in that column.

- Nested scan

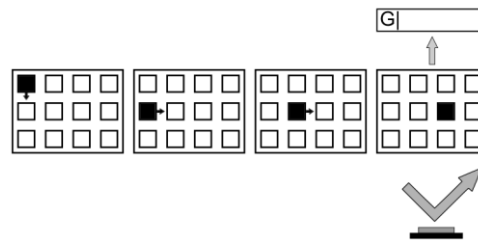


**Figure 40: Nested scan**

Nested scan can be faster than other types of scan, especially for larger sets but usually it needs a larger number of selections than other scan types. To select an item the user scans to the group containing the desired item, selects it, and then scans subgroups in the same way until the desired item is reached.



- Directed scan



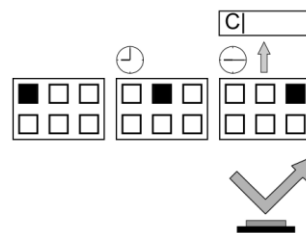
**Figure 41: Directed scan**

In the directed scan the user directs the highlighter movement freely in a number of directions (left, right, up, down). In the most cases, this type of scanning movement can be applied only to a rectilinear grid.

### Scanning methods

The scanning methods define how the highlighter is controlled with the switches. There are several different methods of scanning: auto scan, user scan, critical over scan. These methods are commonly used but not the only ones available.

- Autoscan

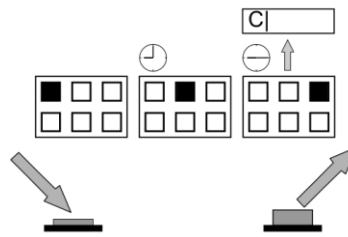


**Figure 42: Autoscan**

In this scan method the highlighter moves automatically from item to item (or group to group) at a set rate. When user activates a switch the currently highlighted item is selected. The time that an item is highlighted usually varies between a tenth of a second and 10 seconds. This method can be very fast, some experienced users can operate with a scan time of 0.1 seconds [30].

Autoscan usually requires a single switch, so it is good for users who can control only one switch. Autoscan is also good for people who cannot hold a switch in the on position. A user who does not have good switch timing should consider other scan methods.

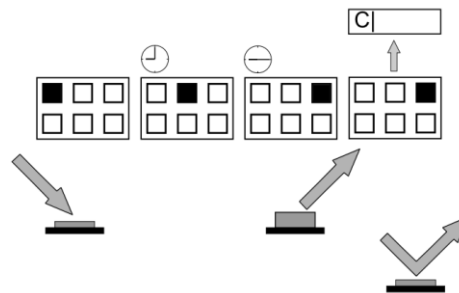
- Userscan



**Figure 43: Userscan**

In this method when the switch is activated and held in the on position, the scan starts and moves with autoscan. When the switch is released the scan stops and the currently highlighted item is selected, either immediately on releasing the switch or after dwell time.

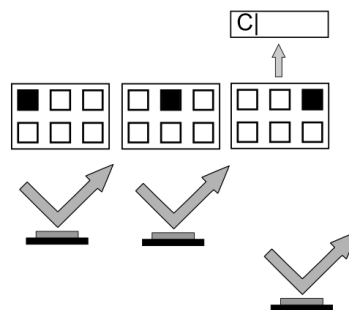
This method can be also controlled by two switches. In this case the item is selected with a second switch.



**Figure 44: Userscan with two switches**

User scan is good for the user who can keep the switch activated and release it with good timing. It should also be considered for the user who becomes anxious or stressed waiting to press the switch for autoscan.

- Stepscan



**Figure 45: Stepscan**

In this method pressing and releasing the first switch progresses the highlight by one item. The second switch is used to select the highlighted item.

Stepscan can be also used with one switch and dwell time.

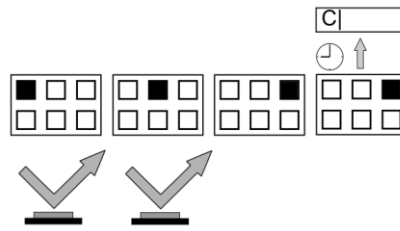


Figure 46: Userscan with one switch

This method should be considered for the user who cannot keep a switch activated and release it with good timing, or who has difficulty waiting and activating the switch with good timing.

### 2.1.7.5 Scanning in existing applications

Examples of scanning in Oksa keyboard, SymWord and TheGrid.

- Oksa keyboard



Figure 47: Column-row scanning in Oksa keyboard

In the Oksa keyboard there is implemented the autoscan and stepscan with column-row and row-column moves. User can control scanning using mouse buttons, joystick, or keyboard. The highlighted keys have a red frame.

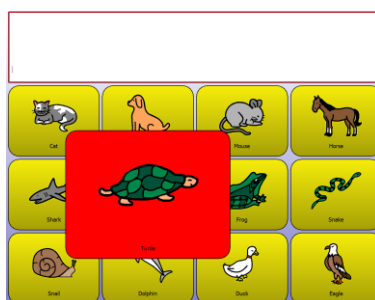
- SymWord



Figure 48: Nested-group scanning in SymWord

In SymWord there is implemented the autoscan and stepscan with simple and nested-group moves. Using the included grid editor (GridEd) the user can set scan order for the cells or assign the cells to the scan groups. The user can control scanning using either mouse or keyboard. The highlighted cell has a frame with a colour chosen by the user.

- TheGrid



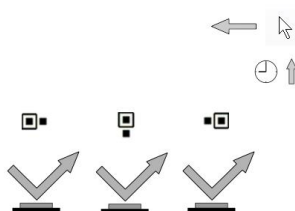
**Figure 49: Magnified cell in simple scan in The Grid**

The Grid has implemented the autoscan, userscan, stepscan and more complex scan methods. The user can choose simple, row-column, column-row scan moves. The scanning can be controlled with F1-F8 keys, joystick or different switch interfaces. The highlighted cell can have a frame, different colour or can be magnified.

### 2.1.7.6 Mouse control by one switch (One switch mouse)

The One Switch mouse is a very interesting technique that was invented by Professor Hatakeyama in Waseda University (Japan)<sup>53</sup>. It is especially interesting as it uses a combination of tapping (to step through directions) and holding (to choose functions such as left click, right click etc) of a switch. It is especially useful for people with Muscular Dystrophy - a degenerative condition whose later stages leaves people as almost completely immobile but still retaining good temporal control between their thumb and first finger. A particularly useful feature was the acceleration of the mouse movement enabled users to “fly” around the screen at great speed. Hardware systems based on this technique have been built in Japan<sup>54</sup> and, after coming across these systems in 2008 and being impressed at the level of control this technique afforded user’s in a Japanese hospital, permission was sought to “clone” the system using software and existing switch input hardware. The first software version was produced by Sensory Software Ltd in the UK<sup>55</sup>.

#### Using PC version of One Switch Mouse, moving the mouse



**Figure 50: Moving One Switch Mouse left**

To move the mouse the user taps the switch to select the desired mouse direction, after a brief pause the mouse starts to move in that direction. The icon which describes the direction appears at the left of mouse cursor.

<sup>53</sup> [http://homepage2.nifty.com/htakuro/paper/1SW\\_Mouse/1SW\\_Mouse.html](http://homepage2.nifty.com/htakuro/paper/1SW_Mouse/1SW_Mouse.html)

<sup>54</sup> <http://homepage3.nifty.com/kikiroom/sub1.htm>

<sup>55</sup> <http://www.oskaworld.com/category/one-switch-mouse-for-pc.php>

## Clicking with One Switch Mouse

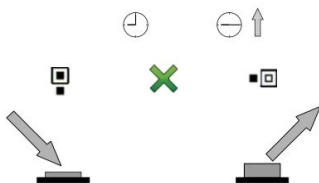


Figure 51: One switch mouse. Choosing left click

The user holds the switch down to scan between the mouse options in order (move, stop, left click, right click, double click, drag). To choose the desired action, the user must release the switch.

### 2.1.8 Touch screens

Touch screen have been used with disabled adults and children for many years [176]. However, more recently with the advent of Windows 7 and hand held devices such as the iPod Touch and iPhone and most recently the iPad they have gained much more interest and acceptance in the population as a whole especially as these systems now use Mutitouch techniques. The concept of touching where you look is very attractive, hand eye coordination being one of our earliest learned behaviours, but recent touch screen devices take this further with Multitouch capabilities. Multitouch<sup>56</sup> enables users to perform different functions with two or more fingers which perform gestures on the display, e.g. pinching with the finger to perform a resize operation.

Within the AsTeRICS project there two categories of devices that are of particular interest:

1. A second multitouch screen to complement the user's normal screen. An example of one of these is the Mimi 720<sup>57</sup>. This is a unit which just plugs into the USB port of a computer and provides a second (or shadow) display powered by the user's computer. We see this being of potential use in two areas:
  - The user's on-screen keyboard can be placed on the touch screen which removes clutter from the main screen. A disadvantage of this approach may be in asking the user to change their focus between two screens. It is not clear currently whether this approach is helpful or not.
  - In a communication scenario one screen can be placed facing the user and the other placed facing the person they are communicating with. This technique can be found in the Lightwriter<sup>58</sup>.
2. Using a Smartphone as an input system for the desktop computer. This could either be used in a similar way to a separate display as in 1 (above) or by using the Smartphone as a mobile touchpad/mouse (see section 2.1.10.1). For the iPhone/iPod touch such software is already available<sup>59</sup> and various open source project show how

<sup>56</sup> <http://en.wikipedia.org/wiki/Multi-touch>

<sup>57</sup> <http://www.slashgear.com/nanovision-mimo-720-s-touchscreen-display-review-2053124/>

<sup>58</sup> <http://www.toby-churchill.com/en/content/lightwriters-0>

<sup>59</sup> <http://www.mobileairmouse.com>

to achieve this with other devices<sup>60,61</sup>. Although the functionality of these existing systems is not necessarily ideal, they provide a starting point for further exploration.

### 2.1.9 Wiimote

The Wii Remote (also Wiimote) is the main input device for the Nintendo Wii game console with motion sensing capability. It is a stick "handset" of the size of a TV remote control including: three accelerometers (x, y, and z axis, see Figure 52: The coordination system of the Wii Remote), several buttons and an infrared (IR) camera for motion tracking, a vibration unit, a speaker and 4 LEDs for output capability. The accelerometers provide continuous position data, not only movement data. The IR camera is used in combination with an IR light source (called sensor bar) for positioning issues in the 3D environment. The communication with the Wii console is made via Bluetooth.

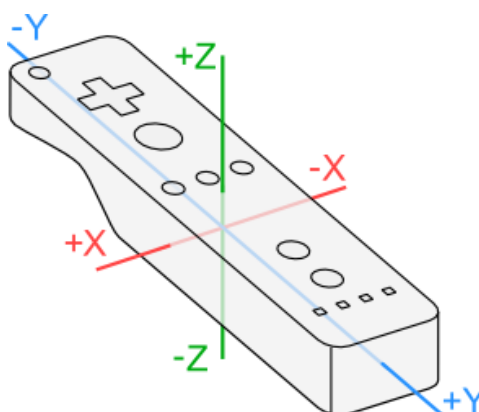


Figure 52: The coordination system of the Wii Remote<sup>62</sup>

The combination of the motions sensing capability, the connection via the wireless Bluetooth and the price (about € 30 in March 2010) makes the Wii Remote to a very versatile useable device; see [31], [32] and [33] for example. The listed attributes of the Wiimote make it a proper fitting AT device for rapid prototyping and for the usage as general input device: it is robust, handy, wireless, provides 6 degrees of freedom and additional buttons.

The connection between the Wiimote and the host device (PC, game console, embedded system, etc.) will be established by a standard Bluetooth stack. On the host device, the Wiimote can be accessed via special application (e.g. Glovepie<sup>63</sup>) or libraries (e.g. libwiimote<sup>64</sup>) for programming languages. Furthermore, the Wiimote has already been connected to the Gumstix embedded platform<sup>65</sup>. In 2009 Nintendo released the MotionPlus extension for more quickly and accurately recognition of motions.

<sup>60</sup> <http://sensorapi.codeplex.com/>

<sup>61</sup> <http://www.codeproject.com/KB/mobile/MobileTouchPad.aspx>

<sup>62</sup> <http://wiibrew.org/wiki/Wiimote>

<sup>63</sup> <http://sites.google.com/site/carlkenner/glovepie>

<sup>64</sup> <http://libwiimote.sourceforge.net/>

<sup>65</sup> <http://www.tango-controls.org/embedded/gumstix/device-servers/wiimote>

## 2.1.10 New input hardware systems

Many new input hardware systems can be used for meeting the individual needs of disabled people. This chapter describes adaptations of touchpads and 3D mice.

### 2.1.10.1 Touchpads

These days most laptops come with a touchpad. However, it is also possible to buy keyboards with Touchpads<sup>66</sup> or just USB touchpads<sup>67,68</sup>. In most cases, from the computer's perspective, these touchpads are seen as hardware mice. However, some manufacturers provide SDKs that enable developers to control the touchpads directly:

Synaptics, who produce Touchpads for many laptops have an SDK<sup>69</sup>. However, it is often not clear if a particular laptop has a Synaptics touchpad, and some recent tests by the authors indicate that not all Synaptics touchpads can be controlled using their SDK (e.g. the authors tested on some recent netbooks and found that the SDK access was not available).

Cirque produces a range of external touchpads. The authors have recently been in contact with Cirque and have obtained a beta SDK which gives complete access to touchpad information. The advantage of having such access is that the touchpad can then be used to explore non-classical user interface techniques. Areas that are to be explored in the AsTeRICS project include:

- Using the touchpad like a joystick where mouse movement is controlled by using the relative movement from an initial touch – this could be useful for people with good (but limited) fine motor control.
- Applying tremor reduction to finger movements on the touchpad.
- Exploring text input and mouse control using gestures on the touchpad.

### 2.1.10.2 3D mice



Figure 53: SpaceExplorer – 3D mouse from 3Dconnexion

3D mouse from 3Dconnexion<sup>70</sup> is a navigation device designed for navigating 3D CAD and graphic applications. It provides six degrees of freedom.

Movements used to operate 3D mouse can be adapted to meet some disabled people's needs. An example application from The Oska Suite<sup>71</sup> uses a 3D mouse to emulate standard mouse movements.

<sup>66</sup> <http://www.logitech.com/index.cfm/keyboards/keyboard/devices/192&cl=us,en>

<sup>67</sup> <http://www.cirque.com/desktoptouchpad/touchpad-mouse-overview.aspx>

<sup>68</sup> <http://www.maplin.co.uk/Module.aspx?ModuleNo=264584&C=Froogle&U=264584&T=Module>

<sup>69</sup> <http://www.synaptics.com/search/node/sdk>

<sup>70</sup> <http://www.3dconnexion.com>

## 2.2 Enhancing input

Typing, these days, is a very important process in computer and mobile devices operating. People with motor disabilities often type slowly, especially when they use equipment like: switches, trackballs, headsticks, mouthsticks and scanning systems. To make typing for them faster and more comfortable Assistive Technology developers use technologies like: word prediction, abbreviation expansion, disambiguation or dasher.

### 2.2.1 Word completion and word prediction

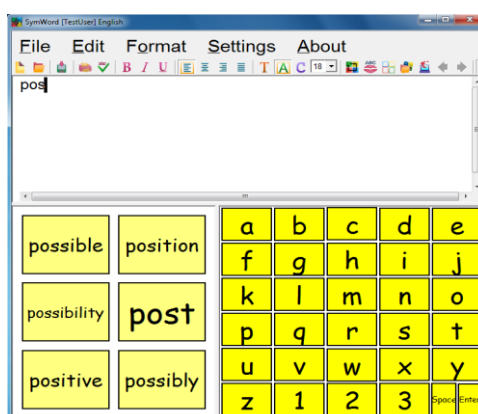


Figure 54: SymWord with prediction

Word completion is a process of complementing words based on the initial letters of words. As the user types, completion software monitors his/hers input and generates a list of words beginning with the letter sequence typed by user. Each time a letter is added, the list is updated. When the desired word appears in the list, the user can choose it and insert it into the text in a single action. Word prediction is the process of predicting words based on the previous words written by user. Word completion and word prediction can increase the typing speed by reducing the number of selections necessary to write a word. Completion and prediction applications can use different adaptive algorithms for example place the words which user uses more higher in the list.

Examples of assistive technology applications, which use prediction:

- SymWord<sup>72</sup>
- Prototype<sup>73</sup>
- WordQ<sup>74</sup>

### 2.2.2 Abbreviation expansion

Abbreviation expansion is a process of replacing the typed abbreviations with text assigned to them, e.g. the user type “ast” and the word “AsTeRICS” is produced. Abbreviation expansion applications run in the background and monitor a user’s keyboard input. When the user types one of the defined abbreviations, the abbreviate application recognizes it and

<sup>71</sup> <http://www.oskaworld.com/>

<sup>72</sup> <http://www.clarosoftware.com/index.php?cPath=372>

<sup>73</sup> <http://www.sensorysoftware.com/prototype.html>

<sup>74</sup> <http://www.wordq.com/>



replaces the abbreviations with defined text. The user can define his own abbreviations for the phrases, which he/she uses frequently and add them to the application. The abbreviation expansion process reduces the number of keystrokes needed to complete sentences.

Example of applications that use abbreviation expansion:

- Abbreviate<sup>75</sup>
- WiViK<sup>76</sup>
- Microsoft Word (i.e. The autocorrect feature)

### 2.2.3 Disambiguation

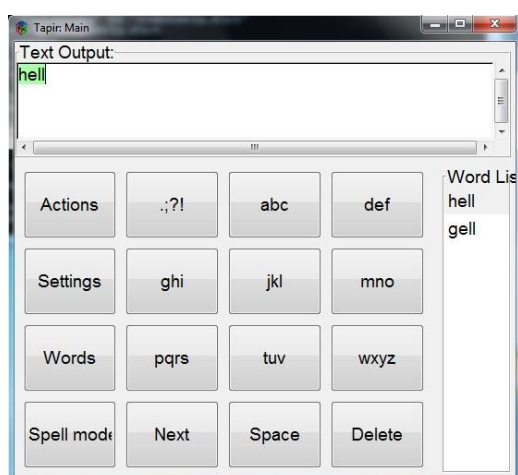


Figure 55: Tapir

Disambiguation is similar to the technology T9 used in mobile phones. This method combines groups of letter on each key using dictionary of the words which will be used for typing. The disambiguation is especially convenient for head mice and scan systems users because the keyboard in disambiguation applications has less keys than the keyboard with one letter per key. The number of keystrokes needed to write text, in most cases, is the same as in keyboard contains keys with one letter. An example of application which uses the disambiguation method is Tapir<sup>77</sup>.

### 2.2.4 Dasher

Dasher it is text-entry interface driven by natural continuous pointing gestures. The user points towards the rectangle with letters, which he/she want to type and the dasher zooms towards them and inserts them into the text box. Dasher can be driven using a mouse, trackpad, touchscreen, rollerball, joystick or any two-dimensional pointing device (including head mice and gaze trackers). It is claimed that an experienced Dasher user can type as fast as normal handwriting (29 words per minute) and even faster (39 words per minute). Dasher is free and open-source software and can be downloaded from the project's website<sup>78</sup>.

<sup>75</sup> <http://www.abbreviate.cc/index.html>

<sup>76</sup> <http://www.wivik.com/>

<sup>77</sup> <http://www.inference.phy.cam.ac.uk/tapir/>

<sup>78</sup> <http://www.inference.phy.cam.ac.uk/dasher/>

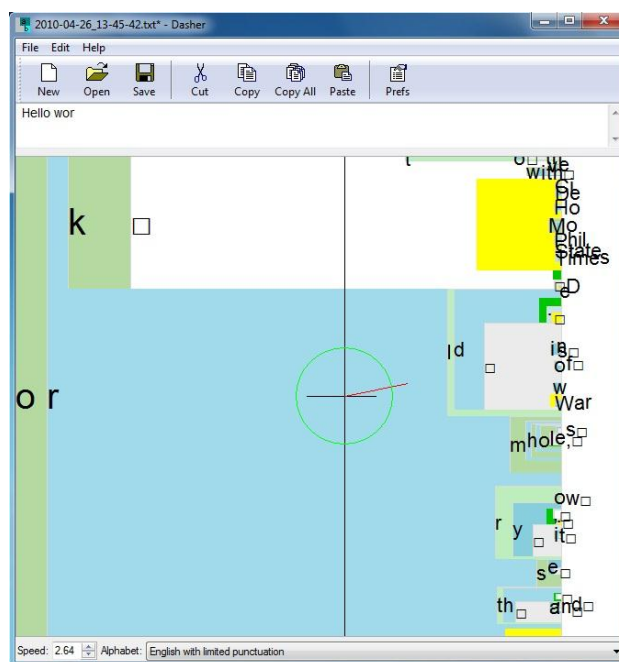


Figure 56: Dasher

## 2.2.5 FASTY Project

FASTY (Faster Typing for Disabled Persons) Project is a European Project in IST (Information Society Technologies) part of the Fifth Research and Technological Development Framework Programme (January 2001 to March 2004). FASTY created a system for increasing the text generation rate of disabled persons by predictive typing and dedicated advanced input devices. The system was developed for the Dutch, French, German, and Swedish language, the concept, however, can be used for most European languages. FASTY project<sup>79</sup> was created to reduce the number of keystrokes more than 50%.

## 2.3 Assistive Control Systems

Control systems and electromechanical systems in the sector of assistive technology can help people with disabilities in many areas of their daily life, enhancing their autonomy and mobility. Among these systems are Environmental- and Smart Home Controllers, wheelchair controllers with support for different input methods and electro-mechanical aids for grasping and manipulating the environment. The following sections give a brief overview on existing technologies.

### 2.3.1 Environment Control Systems

Environment control systems are central controllers, with the possibility to actuate with communication devices (e.g. telephone), with entertainment systems (TV, radio, etc.), environmental controls (light switches, door control, etc.) and other devices. Different companies provide products with widespread functionality. Table 2 shows the major products divided in the categories of user interface, configuration, communication and other.

<sup>79</sup> <http://www.is.tuwien.ac.at/fortec/reha.e/projects/fasty/fasty.html>

	FST James II	GEWA Control Prog	GEWA Progress	SiCare Basic	SiCare Pilot Senior	SRS Intellec	Proteor Nemo+	Proteor Keo	Easy by Voice	HMC Easy Rider	Apollo 15 dlx	Admiral 2
<b>User Interface</b>												
User interface is adaptable to the user	-	-	+	+	-	+	+	+	+	+	-	+
Number of different menus	17	15	∞	∞	1	∞	∞	∞	∞	10	10	?
Visual Feedback	+	+	+	+	+	+	+	+	-	+	+	+
Audio-Feedback	+	+	+	+	-	+	+	?	+	+	+	+
Speech recognition	-	-	-	+	-	-	+	-	+	-	-	-
Speech output	-	-	-	+	-	+	+	-	+	+	+	+
Scanning	+	+	+	+	+	+	+	+	-	+	+	+
Number of internal switches	24	18	∞	2	15	5	3	0	0	0	15	10
Number of possible external switches	1	5	2	8	1	2	1	?	0	6	1	5
Joystick possible	+	+	+	-	-	+	+	+	-	+	-	+
<b>Configuration</b>												
User can configure the environmental control device	+	+	+	+	+	+	+	+	+	+	+	+
Basic Configuration can be done without Computer (structuring of menus, learning of IR-codes, ...)	+	+	+	+	+	+	+	+	+	+	+	+
Configuration sharing	+	+	+	+	-	-	-	+	?	?	?	?
<b>Communication</b>												
IR-Control	+	+	+	+	+	+	+	+	+	+	+	+
Support of Bluetooth	-	-	-	-	-	-	-	+	-	-	-	+
Support of WLAN	-	-	-	-	-	-	-	-	-	-	-	-
Support of communication technologies like GSM	-	-	-	-	-	+	-	+	+	+	-	-
Support of bidirectional communication with devices	-	-	-	-	-	+	-	+	-	+	+	+
<b>Other</b>												
Number of possible functions	289	241	256	∞	15	> 2500	200	?	∞	200	150	200
Modular – Extendable via PlugIns	-	-	-	-	-	-	-	-	-	+	-	-
Based on special hardware	+	+	+	+	+	+	+	+	+	+	+	+

Table 2: Major Environmental Control Systems and their Functionality

## 2.3.2 Wheelchair Control

Electric wheelchair control systems are developed by several companies. This section will describe two of the most common systems and possibilities for external interaction with these control systems.

Wheelchair Company	Control System
Sunrise Medical	PG Drive Technology (VSI, VR2)
Invacare	Dynamic (DX)
Otto Bock	PG Drive Technology (VSI)
Meyra / Orthopedia	PG Drive Technology (VSI)
Vassilli	N.N.
Alber	Alber
Permobil	PG Drive Technology (R-net, VR2)
Etac / Balder	Dynamic (DX)

**Table 3: Major Electronic Wheelchair Companies and their Control Systems**

### 2.3.2.1 PG Drive Technology

The company PG Drive Technology (PGDT<sup>80</sup>) develops several wheelchair control systems. The most popular are VSI and VR2, the most modular and extendable system is the R-net (it provides an I/O-module to connect switches or other input devices). The communication between the components will be done by a custom software communications protocol developed by PGDT specifically for powered wheelchair applications. The underlying hardware link utilizes the CAN bus standard.



**Figure 57: VSI and R-net Controls from PGDT**

### 2.3.2.2 Dynamic Controls

The company Dynamic Controls<sup>81</sup> has developed several wheelchair control systems, the A-series, Shark and DX/DX2 controls. The DX2 is the most extendable and flexible system, with additional connection modules. For the communication between the modules, a communication protocol called DX is used. There is no further information about this protocol available.

<sup>80</sup> <http://www.pgdt.com>

<sup>81</sup> <http://www.dynamiccontrols.com/>



**Figure 58: DX-G90 and DX2 control form Dynamic Controls**

Due to the fact that the major companies using proprietary communication protocols, the effort to connect an external control system (e.g. AsTeRICS) is quite high and depends on the cooperation of these companies.

### 2.3.3 Pneumatic Gripper for Mouthsticks

Persons with severe physical disabilities (limited or no hand control) often use mouth- or headsticks to handle things (pressing buttons, pushing things, etc.) in the nearer environment including computers. Standard mouthsticks do not allow the grabbing of objects. Such functionality would considerably improve the benefit of sticks and enhance abilities and independence of users.

There already has been done work to provide people with limited or no hand function with an aid for manipulating and grasping things. There are several approaches:

- Hand orthoses which are controlled by EEG, EMG or other signals. [1][2]
- Hand mounted grippers (electric or pneumatic) which are controlled by switches or sensors. An example of a pneumatic gripper system controlled by capacitive touch sensors is the off-the-shelf system Gripability<sup>82</sup>
- Low tech solutions which are mechanical. There exist various solutions which are made for specially tasks. An example for a mouthstick with mechanical gripper is the tongue operated Pincer Mouthstick<sup>83</sup>

A new approach is the combination of a pneumatic gripper with a mouthstick. Biosignals (EEG/EMG) are transmitted to a PC, processed and the output used to control electro-pneumatic pressure regulator for the gripper. [3]



**Figure 59: The prototype of the pneumatic gripper equipped mouthstick [3]**

The company Gripability offers pneumatic grippers, not on a mouthstick, but as gripping support for persons with limited hand functions. The grippers are controlled by one to several switches, being adaptable to the user's needs.

<sup>82</sup> The Gripability System. <http://www.gripability.de>

<sup>83</sup> Adlib incorporated: Pincer Mouthstick. <http://www.adlib-inc.com/AD8018.htm>



Figure 60: The e3 gripper in use<sup>84</sup>

### 2.3.4 Robots as assistive devices

The assistive robotic is a readaptation tool as it is supposed to substitute (or at least to supplement) the impaired human function. In this case, the robot interacts with the environment on behalf of the user, what offers him/her a certain autonomy and independence.

Figure 61 shows the Handy robot, developed by the CEA, France, which is available in United Kingdom. This robot aims to make eating the IMC patients.



Figure 61: Handy, CEA (France)

Manus (see Figure 62), the world wide known robot issue of the CEA, France, is commercialized by the Exact Dynamics (Netherlands). It is an articulated arm with 6DOF with a simple motorized griper. The arm is attached to a wheelchair.



Figure 62: Manus (CEA, France, and Exact Dynamics Netherlands)

<sup>84</sup> [http://www.gripability.de/gripability\\_eng/produkte\\_e3\\_info.html](http://www.gripability.de/gripability_eng/produkte_e3_info.html)

Ongoing research aims to increase the end-user interactivity with the environment: a vision system and a tactile screen will be added and ported by the wheelchair.

The vision system will film the environment and localize the object to grasp selected by the end-user via a wheelchair attached screen.

### 2.3.5 Robots as rehabilitation tools.

Robots as rehabilitation tool (or gesture assistance) memorize the gesture to be learned and in function of the degree of the impairment, it assists the user in gesture execution or it completes the imperfect gesture.

Figure 63 shows a Virtuose robot (CEA, ONG Approche and Fondation Caisse d'Epargne, France; Kinéhaptique project) which is used for upper limb rehabilitation purposes. A virtual reality environment (developed by Haption) using 3D vision navigation will offer a relaxing assistance for rehabilitation exercises which will be easier supported by the end users.



Figure 63: Virtuose robot for upper limb rehabilitation (CEA, France)

## 2.4 Communicators

Communicators are augmentative and alternative communication (AAC) devices. They are used mainly by people with speech disorders. Communicators are designed to compensate problems of these people and help them with communication in society.

### 2.4.1 Basic communication devices



Figure 64: BIGmack communicator

Basic communicators, in the most cases, include one switch and a recording and playback module. First, the message is recorded on the device and after each switch is activated the message is played. There are also devices which are able to record series of message and each press of the switch will play next message. The communicators are usually powered from batteries and can be moved with user.

Example of basic communicators:

- GoTalk Button<sup>85</sup>
- Clip Talker<sup>86</sup>
- BIGmack® communicator<sup>87</sup>
- No Touch Talker<sup>88</sup>

## 2.4.2 Advanced communicators



Figure 65: GoTalk 20+

Advanced communicators work in a similar way as basic communicators but contain more messages, usually from 4 to 32. Before they are used, the message should be recorded and the overlays prepared with a “description” of the messages. Usually, these devices can contain recorded messages for more than one overlay. Some of the advanced communicators contain scanning systems or can switch on and off electrical devices.

Example of advanced communicators:

- Tech/Talk<sup>89</sup>
- GoTalk Pocket<sup>90</sup>
- GoTalk 20+<sup>91</sup>
- SuperTalker progressive communicator<sup>92</sup>

<sup>85</sup> <http://www.attainmentcompany.com/product.php?productid=16149&cat=338&page=1>

<sup>86</sup> <http://www.attainmentcompany.com/product.php?productid=16502&cat=338&page=1>

<sup>87</sup> <http://www.ablenetinc.com/Store/tabid/205/Default.aspx?CategoryCode=105>

<sup>88</sup> [http://www.inclusive.co.uk/catalogue/acatalog/No\\_Touch\\_Talker.html](http://www.inclusive.co.uk/catalogue/acatalog/No_Touch_Talker.html)

<sup>89</sup> <http://www.amdi.net/techtalk/>

<sup>90</sup> <http://www.attainmentcompany.com/product.php?productid=16585&cat=0&page=1>

<sup>91</sup> <http://www.attainmentcompany.com/product.php?productid=16146&cat=0&page=1>

<sup>92</sup> <http://www.ablenetinc.com/Store/tabid/205/Default.aspx?CategoryCode=71>



### 2.4.3 PC based communication software

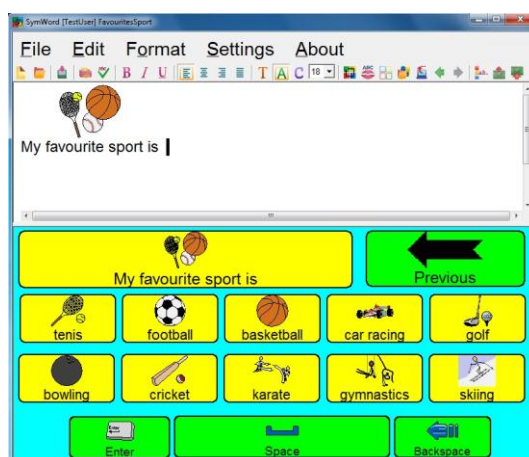


Figure 66: SymWord

Communication applications have grids with the cell which are equivalent to communicator keys. The user selects the cell from the grid and inserts the content to text box. Most of these applications have scanning options and they are integrated with speech synthesizers. Some communication applications have advanced options such as: send/receive email, environmental control or control of the other applications running on the computer.

Examples of communication applications:

- SymWriter<sup>93</sup>
- The Grid 2<sup>94</sup>
- SymWord<sup>95</sup>
- Symbols for Windows<sup>96</sup>

### 2.4.4 Other mobile communicators.

The popularity of devices with touch screen (e.g. PDAs and smart phones) has resulted in AAC developers starting to create communication applications for them. A recent example of such an application is Proloquo2Go<sup>97</sup> which can be installed on iPhone, iPod touch, and iPad. Some of the communication applications installed on the mobile phone can also use phone options such as sending SMS.

## 2.5 Open Source and AT

Open source software is becoming more popular also in the field of Assistive Technology. The list below shows examples of free or open source applications for motor disabled people:

<sup>93</sup> <http://www.widgit.com/products/symwriter/index.htm>

<sup>94</sup> <http://www.sensorysoftware.com/thegrid2.html>

<sup>95</sup> <http://www.clarosoftware.com/index.php?cPath=372>

<sup>96</sup> <http://www.symbolforwindows.eu/en/index>

<sup>97</sup> <http://www.proloquo2go.com/>

**Alternative interfaces:**

- Joystick 2 Mouse<sup>98</sup> – Control of mouse and keyboard using joystick or gamepad
- Dwell Clicker<sup>99</sup> - Enable mouse click by dwell time
- Morseall<sup>100</sup> - Linux application that allows control of the computer using the mouse button only by tapping Morse codes
- Mouse Smoothing Software<sup>101</sup> - improved control of the mouse for people with hand tremor
- Mouse Gestures<sup>102</sup> - User control of Mozilla Firefox by just the mouse

**Voice control applications:**

- Shoot<sup>103</sup> - Translates voice commands into keystrokes.
- Open Mind Speech<sup>104</sup> - Speech recognition tools.
- Game Commander<sup>105</sup> - Voice recognition program designed for games commands

**Web cam mices:**

- Face Mouse<sup>106</sup> – Web cam mouse
- Enable Viacam<sup>107</sup> - Web cam mouse

**Blink eye control applications:**

- b-link<sup>108</sup> - It allows to operate the computer by user eye blinks

**Text input interfaces:**

- Word Predictor<sup>109</sup> – Prediction system for users with limited mobility
- Dasher<sup>110</sup> - a text-entry interface driven by natural continuous pointing gestures

**Screen Keyboards:**

- Click-N-Type<sup>111</sup> – It is an on-screen virtual keyboard
- GOK<sup>112</sup> - GNOME onscreen keyboard

---

<sup>98</sup> <http://atzitznet.no-ip.org/Joy2Mouse3/index.html>

<sup>99</sup> <http://www.sensorysoftware.com/dwellclicker.html>

<sup>100</sup> <http://morseall.org/>

<sup>101</sup> <http://www.alphaworks.ibm.com/tech/mousesmoothing>

<sup>102</sup> <http://optimoz.mozdev.org/gestures/installation.html>

<sup>103</sup> <http://clans.gameclubcentral.com/shoot/>

<sup>104</sup> <http://sourceforge.net/projects/freespeech/>

<sup>105</sup> <http://www.tucows.com/preview/364262>

<sup>106</sup> <http://www.oskaworld.com/category/free-face-mouse.php>

<sup>107</sup> <http://eviacam.sourceforge.net/eviacam.php>

<sup>108</sup> <http://b-link.sourceforge.net/>

<sup>109</sup> <http://www.projectpossibility.org/viewProject.php?id=5>

<sup>110</sup> <http://www.inference.phy.cam.ac.uk/dasher/>

<sup>111</sup> <http://www.lakefolks.org/cnt>

<sup>112</sup> <http://www.oatsoft.org/Software/gok>

### One switch games:

- Orb<sup>113</sup> – One switch game
- Orbit<sup>114</sup> – One switch game

There are two popular web services which host open source AT applications: OATS and Project: Possibility.

#### 2.5.1 OATS

Open Source Assistive Technology Software (OATS)<sup>115</sup> is a web site which is a repository for open source assistive technology. OATS is also platform to facilitate communication between developers and AT users to create better AT software. The OATS site is a place where the AT users can find a wide range of free AT applications.

#### 2.5.2 Project: Possibility

Project: Possibility<sup>116</sup> is a web service, which aims to support a community of people with disabilities and software developers to work together on AT software

### 2.6 Accessibility interfaces

Accessibility interfaces expose system values and events to Assistive Technology applications which can then represent the application's UI in alternative formats such as speech or Braille and make them more accessible for people with vision and motion disabilities.

#### 2.6.1 Microsoft Active Accessibility

Microsoft Active Accessibility (MSAA)<sup>117</sup> is a COM based technology that provides access to the user interface (UI) elements of the other applications in Microsoft Windows. MSAA is based on a client-server architecture. Applications (e.g. screen readers) which use MSAA to access the UI elements of other applications are called clients. Applications (e.g. word processor) which use MSAA to provide the information about their UI elements to clients are called servers. Clients can get basic information about server controls, such as name, location on screen, or type of control, and state information such as visibility, enabled, or selected. If the user wants his controls to be available for MSAA client he should implement MSAA server API for his controls. There is no need to implement MSAA server API for system controls like buttons or check-boxes, because they are available for MSAA by default. Currently MSAA is part of the Windows Automation API.

---

<sup>113</sup> <http://www.projectpossibility.org/viewProject.php?id=7>

<sup>114</sup> <http://www.projectpossibility.org/viewProject.php?id=8>

<sup>115</sup> <http://www.oatsoft.org/>

<sup>116</sup> <http://www.projectpossibility.org/>

<sup>117</sup> <http://msdn.microsoft.com/en-us/library/dd373592%28VS.85%29.aspx>

## 2.6.2 UI Automation

Microsoft UI Automation (UIA)<sup>118; 119</sup> is a similar technology to MSAA and, in fact, is being promoted by Microsoft as the successor to MSAA. It also provides access to the UI elements of applications and allows manipulating them. UIA is a newer technology, provides a much richer object model and exposes more information about the UI than MSAA. It is recommended to be used today. UIA is available for Win32 and NET. Framework. It was designed to be used in other platforms than just Microsoft Windows. In the UIA terminology, the program which uses UIA to get information about the other applications UI is called the client. The program which provides information about its UI elements is called the provider. UI Automation exposes every piece of the UI to client applications as an automation element. These elements are contained in a tree structure, with the desktop as the root element. Objects that are next to each other visually should also be next to each other in the UI Automation tree (see Figure 67).

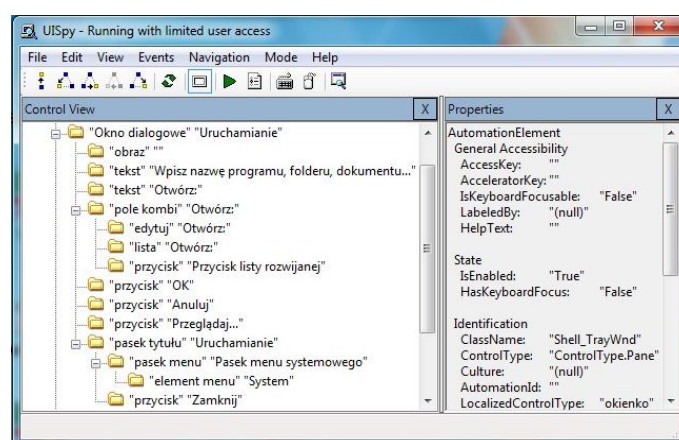


Figure 67: UI Elements in UISpy.exe

UI Automation provides interface for the navigation, filtering, and supports alternative views of the tree. Currently UIA is part of the Windows Automation API.

## 2.6.3 Other interfaces

### IAccessible2

IAccessible2<sup>120</sup> is currently a part of the Linux Foundation and is a royalty free and open multi-platform standard. It was created to provide an accessible OpenDocument Format (ODF) based office suite for the Commonwealth of Massachusetts. IAccessible2 is designed to augment MSAA with more information. It can be easily integrated with MSAA applications without serious changes in implementation and semantics of these applications. It provides access to rich document applications such as the IBM Workplace productivity editors and web browsers such as Firefox. It supports Web 2.0 technologies like AJAX and DHTML. The future of this work (for Open Office at least) is now unsure since Sun was taken over by Oracle and Sun's accessibility team was reduced<sup>121</sup>.

<sup>118</sup> <http://msdn.microsoft.com/en-us/library/ee684009%28VS.85%29.aspx>

<sup>119</sup> <http://msdn.microsoft.com/en-us/library/ms747327.aspx>

<sup>120</sup> <http://www.linuxfoundation.org/collaborate/workgroups/accessibility/iaccessible2>

<sup>121</sup> <http://ostatic.com/blog/oracle-cuts-affect-gnome-accessibility-work>

### **GNOME Accessibility project**

The GNOME Accessibility project<sup>122</sup> provides open source accessibility solutions for the GNOME desktop.

### **Mono Accessibility**

The Mono Accessibility project<sup>123</sup> enables many .NET Framework and Silverlight applications to be accessible on Linux. It implements UIA for the Mono. Mono is an open source implementation of Microsoft's .NET Framework

---

<sup>122</sup> <http://projects.gnome.org/accessibility/>

<sup>123</sup> <http://www.mono-project.com/Accessibility>

### 3 Electrophysiological signals to enable computer input

Although the measurement of bioelectric signals is important in physiology and rehabilitation technology and a lot of prototypical systems for assistive technologies based on electrophysiological signals exist, only few products are commercialized at present. We found prototypes of human computer interfaces based on electromyography (EMG), electrooculography (EOG) and electroencephalography (EEG). A summary of the most relevant papers including the algorithms used by the authors is included at the end of this chapter.

#### 3.1 EMG based systems

Different papers can be found in the literature that translate electromyographic signal EMG of facial muscles (Corrugator supercillii is the main responsible of frowning, zygomatic of smiling Figure 68) into commands for human-machine interfaces. Ville Rantanen [6] describes a system to detect frowns and eyebrow lifts based on the muscular electrical signal measured with capacitive electrodes. Experimentation with 10 test subjects proved that, under controlled circumstances, the movements are detected with good efficiency, but characterizing the movements into frowns and eyebrow lifts is more problematic. Testing with real patients should be performed to determine the viability in real world conditions.

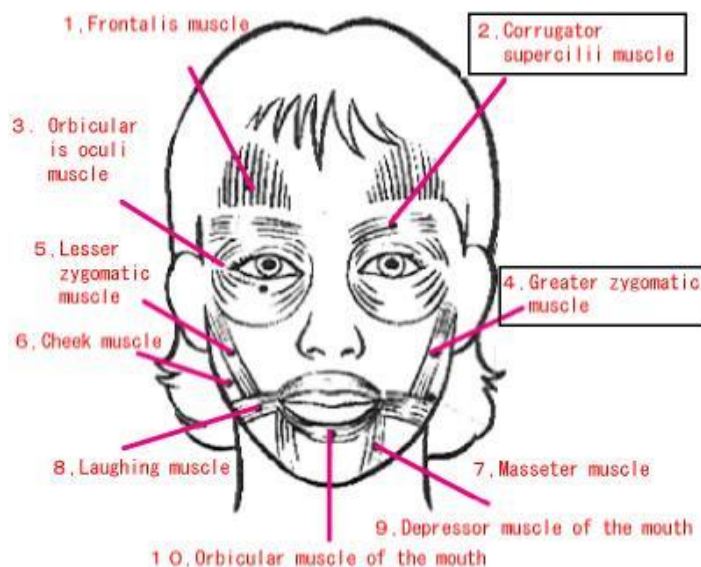


Figure 68: Main facial muscles

Most of the papers revised in this section describe the use of voluntary facial muscle activation and associate the EMG measures to build an on/off switch for mouse-click, (e.g [7], [8]). Chin et.al [9] combine facial EMG and eye gaze in order to enhance the performance of the eye gaze tracking (EGT). The number of freedom degrees of this system is 5, as shown in Table 4. The precision of the task performance is enhanced compared to the use of EGT, but user's speed in performing cursor control tasks was considerably reduced.

Muscle Contraction	Facial Movement	Resultant Cursor Action
Left Temporalis	Left jaw clench	Left increment
Right Temporalis	Right jaw clench	Right increment
Right Frontalis	Eyebrows up	Up increment
Procerus	Eyebrows down	Down increment
Left and Right Temporalis	Left and rightjaw clench	Left-click

**Table 4: Relationship between muscle contractions and resultant cursor actions**

Surakka [10] combines corrugator supercilii muscle EMG (see Figure 69) to solve Midas touch problem within a EGT system. The midas touch problem is the name given to the false selection that happens with EGT systems when an object is gazed for too long but the user did not want to select it, i.e. the system presents a large false positive rate (FPR). Surakka's system [10] uses eye-gaze tracking for movements of the cursor and EMG signals to produce a reliable click operation. The upper and lower parts of the following figure represent a face without and with voluntary activity, respectively.



**Figure 69: A person wearing electrodes attached above the corrugator supercilii muscle site**

Pekka-Henrik [11] uses a Neural Networks to detect voluntarily produced changes in electromyographic signals measured from two facial muscles related with frowning and smiling; corrugator supercilii and zygomaticus major. The neural network system was able to correctly recognize more than 95% of all muscle activations.

Larsen [12] studied the relation between positive and negative affective reactions and facial EMG. Pleasant stimuli typically elicit greater electromyographic (EMG) activity over zygomaticus major and less activity over corrugator supercilii than unpleasant stimuli do. To provide a systematic comparison of these 2 measures, the authors examined the relative form and strength of affective influences on activity over zygomaticus major and corrugator supercilii. Self-reported positive and negative affective reactions and facial EMG were collected as the subjects were exposed to series of affective pictures, sounds, and words. Zygomaticus major pulls the corners of the mouth back and up into a smile while corrugator supercilii draws the brow down and together into a frown.

## 3.2 EOG based systems

The EOG signal results from the variation in the standing potential between the retina and the cornea [13]. The angle of the eye relative to the head changes the electrical potential that can be measured in the surroundings of the orbit of the eye. In the 1960's and 1970's people experimented with the EOG as a means of determining where people are looking [14]. Currently most approaches to sensing point of gaze use an infrared-sensitive camera. Anyhow some EOG based Human-Machine interfaces prototypes have been developed in the few recent years [15], [16].

Gips [17] developed a 5 electrodes EOG based tracking system. The output of this flexible system can be adapted to different applications as for instance a virtual keyboard. He focuses on the feedback to train the user. The many ways the user can affect the signal is considered an advantage rather than a disadvantage. During initial training and skill development people experiment and arrive at their own optimal method of controlling the cursor. It's a semi-conscious skill like riding a bicycle or skiing. Blinking and winking eyes have also been used in implementing clicking by detecting them with electro-oculographic (EOG) measurements.

Deng [18] proposes a multi-purposes eye-movement based HMI. Different directions of eye-movements are detected using EOG signal allowing 5 different commands. The precision of the system after adjustment could get more than 90%.

Kherlopian [19] developed an EOG based system for computer control known as Telepathix that allow users to interact with a computer only by moving their eyes. The system prompts the user to make eye movement commands to navigate the ternary keyboard shown in Figure 70. With each successive eye movement command (Left, Center, or Right) the keyboard iteratively splits into thirds until a letter is chosen. The letter selection process ends when the user initiates the stop sequence by "typing" two blank spaces.

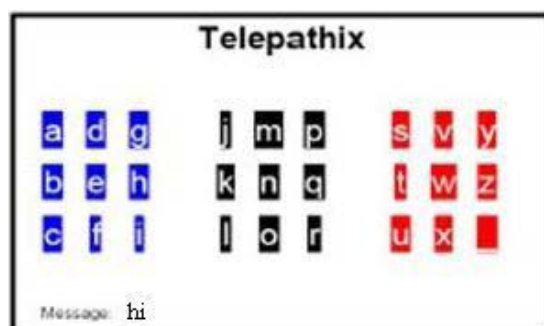


Figure 70: Ternary keyboard for electroocular spelling interface

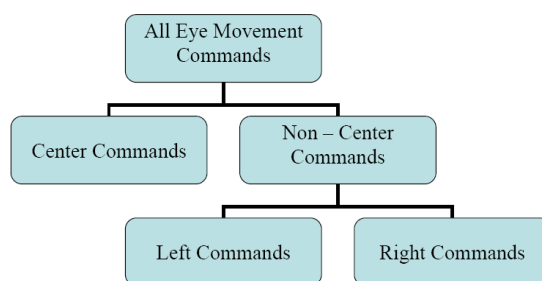
The system measures EOG signal with two bi-polar electrodes placed on the user's forehead, it records windows of 3 seconds and extract the following features to classify the different command options:

- Peak Polarity – Left and Right commands correspond to positive and negative potentials, respectively.
- Slope – In comparison to noise and artefacts, there exists a characteristic slope for potential changes due to eye movement.



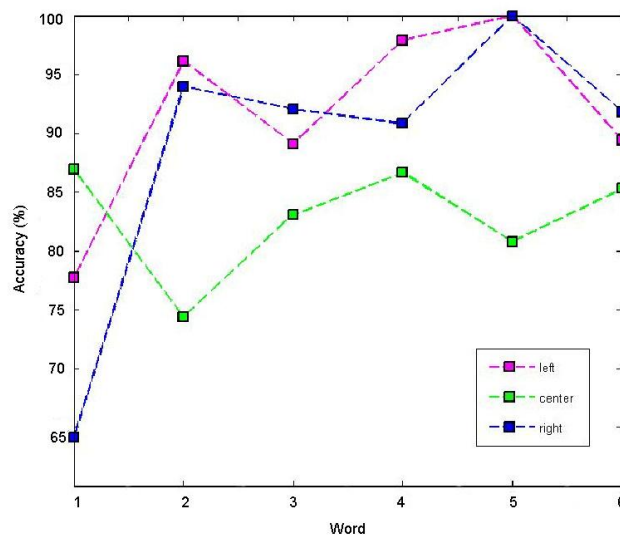
- Threshold – Left and Right commands generally exceed a set value in their respective directions.
- Mean Value – Since the Center command is given by no eye movement, the mean for those commands should be near zero.
- Command Timing – Since the user is prompted and then a three second command recording window opens, the true command signal start point should be expected with a slight delay due to human response time.
- Peak Duration – In comparison to noise or interference, true command signals have characteristic peak durations.
- Correlation of Model Peak – A characteristic peak can be created and used to serve as a model to find similar peaks in other signals.

The classification of the features is performed with a hierarchical clustering procedure (see Figure 71).



**Figure 71: Hierarchical Clustering Procedure**

The accuracy of the system is shown in Figure 72.



**Figure 72: Overall accuracy for 13 users plotted over spelling 6 five-letter words**

The disadvantage of this system is that, the user has to make so many commands through eye movements to select a single letter. Thus this system is less comfortable to the patient. Akam et. al [20] propose to navigate the desired letter using eye-gaze direction and select the navigated letter using an eye blink. As blinking is a natural protection system which defends the eye from environment exposure Dhillon [21] proposes a system that obtains the eye gaze direction (angle) from EOG signals and EMG signal is recorded from eyebrow

muscle activity to select the letter gazed. The gaze angle is proportional to the change of the electrical potential in the EOG when the angle is in the range of  $\pm 50^\circ$  horizontally and  $\pm 30^\circ$  vertically.

To build any EOG based system it has to be taken into account that the retina-corneal potential of the eye changes with light conditions and other variables for calibration purposes. Guven [22] uses neural networks to classify EOG for diagnose of subnormal eyes. The authors point out that the potential of the eye dipole increases with the luminosity. The fact that this phenomenon is not referred in the papers reviewed in this document is probably due to the fact that the signal has many sources of artifacts and thus the change in the retina-corneal potential is not one of the main issues. Huckauf [23] revises gaze gestures and anti-saccades to find optimal actions to elicit commandments. In this case an EGT with an infra-red camera is used, but some results can be generic for ergonomics and usability. Authors of [24] review pros and cons of classical selection methods and novel metaphors, which are based on pies and gestures (anti-saccades, On- and off-screen buttons, pEYEs, Special selection methods). No clear conclusion is stated, the feedback seems to be an important element but may also vary with the selection method used. Therefore, not only the method, but also the circumstances (task, feedback provided, experience of users) should be taken into account when evaluating certain methods.

### 3.3 EEG based systems

The electroencephalography signal can also be used to control a computer or another device. This type of systems is commonly called Brain computer interfaces or Brain machine interfaces. Most of these systems use advanced signal processing and computational intelligence techniques to map different patterns of brain activity with commands of a device. An extended study of these systems is performed in Section 3.4 that reviews the State of the Art of BCI systems based on pattern recognition and computational intelligence based classifiers. Here we will review simple systems that use a direct mapping of some direct measures of EEG and the controls of devices. This is the case of the sensory-motor cortex mu rhythm BCI. In this type of BCI the Mu rhythm amplitude across different sites on the sensory-motor cortex where traditionally mapped to commands.

Wolpaw [25] uses Mu rhythm amplitude assessed by on-line frequency analysis and translate it into cursor movement: larger amplitudes moved the cursor up and smaller amplitudes moved it down. The subjects learns over several weeks, to change mu rhythm amplitude quickly and accurately, so that the cursor typically reached the target in 3 seconds. Farlan and Wolpaw [26] improved this system, by using digital signal processing techniques. The movement was defined in the following way: if  $\Delta V$  is the cursor movement,  $S$  is the control signal (a linear sum of the spectral band-channel combinations used as control features, that is the weighted sum of the frequencies of interest of every channel),  $b$  is the gain, and  $a$  is the mean control signal for the user's previous performance.

$$\Delta V = b(S - a) \quad (1)$$

McFarland in [27] found that the signal to noise ration of Mu and Beta rhythm in BCI systems similar to the one described in [26] could be increased using a Common Average Reference (CAR) or Large Laplacian (that is using 3 cm offset to surrounding electrodes) compared with ear referencing or small Laplacian (using 6 cm to set of surrounding electrodes). [28]

introduces in the same system a linear regression to enhance the performance of the system. To calculate the S in the previous formula he uses a weighted sum of the features:

$$S = \sum w_i x_i \quad (2)$$

where x is the i th feature and w is the weight given to that feature. Zhao [29] describes a system that detects Alpha-block phenomenon that appears when closing the eyes. From EEG recording in occipital regions of the scalp Alpha wave amplitude is measured, thresholded and used for activation of the controls of a robot. A summary of the discussed algorithms and its references is given in Table 5.

Author, year	Modality	Summary of functionality	Processing used
Ville Rantanen 2009	EMG	2-commands (eyebrow-frown): low reliability 1-Command: good reliability	PSD, thresholding
Barreto 2000	EMG-EEG	Left-Right. Click	Spectral power summations over discrete windows over several frequency bands between 8 and 500 Hz
Chin 2008	EGT-EMG	Mouse like system with 5 degrees of freedom. EMG was used to enhance performance of EGT system. The fusion lead to smaller error but speed decrease in the cursor task performance	Spectral power summations over discrete windows over several frequency bands between 8 and 500 Hz camera based eye-gaze information fusion operators
Surakka 2004	EGT-EMG	EGT for pointing+frowning, EMG for click selection.	Algorithm not explained. Users where trained to click with visual feedback of the task performance
Jeff T. Larsen 2003	EMG	positive and negative affective reactions classification using facial EMG	Level of activity in zygomaticus major and corrugator supercilii muscles
Gips J. 1998	EOG	Different applications are cited. Mapping of potential to position. Blinking and winking for mouse-click function.	Processing is not explained
Lawrence Y. Deng 2009	EOG	Mapping of the EOG signal related to eye-movements into 5 commands	Unspecified Fuzzy system.
Armen R Kherlopian 2006	EOG	Virtual keyboard. It is used with 3 commands related with eye movements (left,center, right)	Feature extraction from the raw signal and classification with hierarchical clustering.
Batu Akam 2007	EOG	Added blink-selection command to Kherlopian system.	-
Hari Singh Dhillon 2009	EOG-EMG	Mouse-like system: EOG is used for eye gaze and EMG for click selection.	In the range of $\pm 50^\circ$ horizontally and $\pm 30^\circ$ vertically eye-gaze seems to be a direct mapping of the potential with the angle. Click is extracted from high frequencies 50-500 but processing is not explained.
Li Zhao 2007	EEG	on/off switch controlled with alpha activity that arise when closing the eyes	Detection of Alpha blocks by applying a threshold on the energy.
Wolpaw 1991	EEG	up/down control with Mu rhythm	Threshold Mu rhythm on the motor cortex to select up and down.
Mc Farland and Wolpaw 1997a	EEG	up/down control with Mu rhythm	Cursor movement is linearly mapped to the amplitude of Mu and Beta rhythm at certain sites of the scalp.

Author, year	Modality	Summary of functionality	Processing used
Mc Farland 1997	EEG	up/down control with Mu rhythm	Add to previous system CAR and large Laplacian filter.
Mc Farland and Wolpaw 2005	EEG	up/down control with Mu rhythm	Use a linear regression to enhance performance compared to the liner mapping of Mu and Beta rhythm
Li Zhao 2007	EEG	on/off switch detecting Alpha-block elicited by closing the eyes.	Use a threshold on the amplitude of alpha bursts energy.

**Table 5: Summary of the most relevant papers and the corresponding algorithms implemented.**

### 3.4 BNCI pattern recognition techniques

In the last years the research on Brain and Neural Computer Interfaces (BNCI) has continually increased and this trend seems to stay. This section attains to get an updated overview on the State of the Art of pattern recognition for BCI applications.

First we review employed methodologies within the BCI competitions. This might give a good overview of cutting edge research on the field. Second we review other papers related with the field. The goal of these two tasks is to review the requirements of the toolkit implementation in order to include breakthrough methodologies in the field.

Several freely available toolkits can be downloaded from Internet. Some of those toolkits are meant to be used for general EEG/ECOG data analysis, but some others are more focused on BCI applications. In this document we shall review those toolkits to identify its major features and possibly try to identify their weak points. The idea within AsTeRICS is to develop our own toolkit for BCI applications. This review is then fully justified: before creating our own BCI toolkit, we shall understand what already exists.

#### 3.4.1 SoA in Pattern Recognition for BNCI based on BCI competitions

The current section reviews the techniques and algorithms for pattern recognition used by the winners of the BCI competitions iii 2004 and 2008. In our opinion the BCI competition has become in recent years a recognised benchmark platform for the evaluation of pattern recognition technologies in the BCI research field<sup>124</sup>. As it can be observed in the following paragraphs associated papers have been published recently. None of the winners of the competition iv in 2008 have already published the algorithms they used in this competition (they are all from the same group in the Institute for Infocomm Research, Singapore). Some of these papers, which can give cues to the algorithms used in the 2008 competition, are reviewed in Section 3.4.2.

The work by Rakotomamonjy et al. [177] presents the algorithm that provided the best classification performance on the dataset I corresponding to a P300 protocol. P300 is an evoked potential elicited by infrequent, task-relevant stimuli. A 6x6 grid of letters is presented to the subject (see Figure 73). Columns and rows are highlighted randomly; the subject is focussing his attention to the letter he wants to select, and a P300 ERP is elicited when this letter is highlighted. Ideally a minimum of 12 highlights are required (6 rows and 6 columns) to make a selection, but to enhance the performance the cycle is repeated 15 times. An epoch is extracted after each stimulus, filtered from 0.1 to 10Hz and decimated giving a time

<sup>124</sup> <http://www.bbci.de/competition/?document=competition>

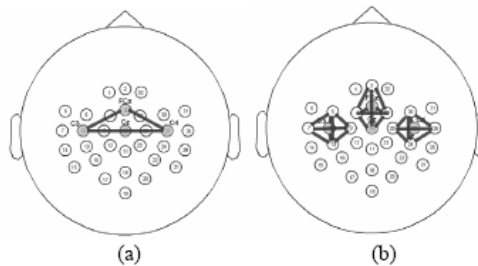
series vector of 14 samples. The 15 epoch corresponding to the same stimulus are averaged to enhance the signal to noise ratio. The system performs with this feature a first binary classification using SVM to decide if P300 is present or not. A fusion of these results is then applied by averaging SVM outputs to decide which one of 36 symbols from the grid is the selected one. To achieve a selection of the optimal channels for each subject a recursive channel elimination is applied<sup>125</sup>.



**Figure 73: Matrix of symbols used for the P300 protocol [177]**

Moreover Lemm et al. [178] designed the winning algorithm for dataset iib. It corresponds to a classical motor imagery protocol with 2 classes recorded with 2 bipolar EEG channels bandpass filtered 0.5-30 Hz. A Morlet wavelet is used to estimate energy for mu and Beta rhythm in the somatosensory cortex to estimate the ERD and MRP. A probabilistic classification model based on Bayesian decision theory is then applied.

Wang et al. [179] describe a single trial classification based on short-range (local-scale) synchrony and long-range (large scale) synchrony (see Figure 74). Local-scale synchrony is found between adjacent areas corresponding to a single sensory modality, i.e., electrodes are in the same region, while large-scale synchrony occurs between a bit widely separated brain regions.



**Figure 74 : (a) Large-scale electrode pairs between the SMA area and left/right. M1 areas. (b) Local-scale electrode pairs located over SMA and M1 areas [179].**

Phase locking value was calculated on three pairs of electrodes, i.e., Fcz-C3, FCz-C4, and C3-C4 for large-range synchrony and four neighboring electrodes of FCz, C3 and C4 were combined to form a five-electrode group respectively for the measurement of local-scale synchrony. The synchronization feature is calculated as follows:

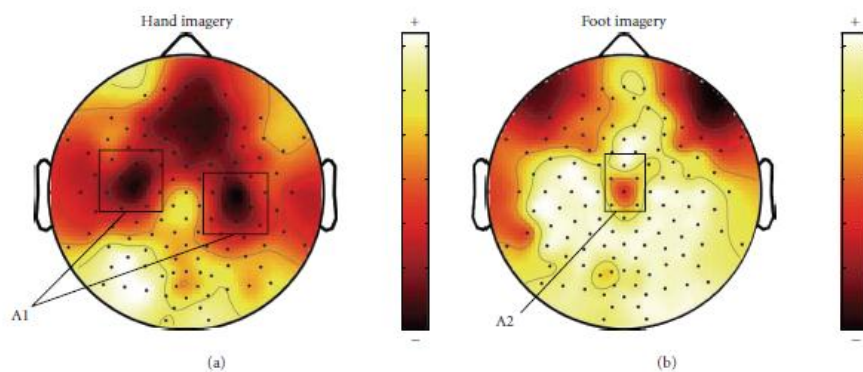
$$PLV = \left| \left\langle e^{j\Delta\Phi(t)} \right\rangle_t \right|$$

where  $\Delta\Phi(t)$  is the phase lock at each time  $t$ . If 2 signals are completely synchronized, is a constant and  $PLV=1$ . If the signals are unsynchronized, then  $\Delta\Phi(t)$  follows a uniform

<sup>125</sup> The code is available at <http://asi.insa-rouen.fr/~arakotom/code/bciindex.html>.

distribution and  $PLV=0$ . Linear classifiers were used to classify local-scale features and large-scale features respectively. A linear classifier was defined by a normal vector  $w$  and an offset  $b$ .

Zhang et al.'s [180] algorithm was applied to this is a Motor imagery task with 2 classes for the training data set (left hand, right foot) and 3 classes for the test data set (left hand, right foot, relax), which is included in dataset IVc. The data were recorded with 118 EEG channels (0.05-200Hz). 37 EEG channels were selected according to ERD distribution (see Figure 75), so only brain regions A1 and A2 are taken into consideration. A common spatial subspace decomposition, as described by Wang et al. [179], is applied to left hand and right foot EEG trials. The method calculates the eigenvalues of the projection in a way that the eigenvector with largest eigenvalue for SF (components of subspace of foot subspace) has the smallest eigenvalue for SH (components of subspace of hand task). A Fisher LDA is used as classifier for the ERD extracted features, whose computation is based on the application of common spatial subspace decomposition (CSSD).



**Figure 75: Averaged ERD spatial mappings of (a) left hand and (b) right foot in the training set [180]**

Galán et al. [181] developed a system to classify a 32 EEG channels recording of a protocol with 3 classes movement: left hand, right hand, and word association (dataset v of the competition). The data were recorded with an asynchronous protocol, i.e. EEG data is not splitted in trials since the subjects are continuously performing any of the mental tasks. The system applies canonical variates transformation (CVT) to project these features into a space that maximizes the separation between the class centroids and minimize the intra-class variance. The classification is then done using distance based discriminant analysis (DBDA) combined with a mental tasks transitions detector (MTTD) based on a dissimilarity function, to classify spontaneous mental activities.

From the former descriptions it is clear that discriminant analysis classifiers and support vector machines are the most used classification methods in the current SoA. Moreover, projection techniques are used frequently. Implementations in the AsTeRICS project in the realm of BNCI will take into account these conclusions.

### 3.4.2 Computational Intelligence Methodologies for BCI

In this section we describe a review of the State of the Art of computational intelligence (CI) techniques employed in Brain Computer Interfaces. There are some works in the analyzed literature that undertake a general survey on BCI [182]. Only a few of those works are based

upon computational intelligence. Therefore we attain a more complete description of the utilization of these kind of techniques in the BCI application field.

Computational Intelligence, also known as Soft-Computing, is a branch of Pattern Recognition that is characterized by the combination of different complementary techniques for the implementation of real applications. In this context CI techniques are grouped in different types of techniques, each of them with its own characteristic function [183]:

- Neurocomputing, which groups different neural network techniques.
- Fuzzy Computing, which groups fuzzy logic, fuzzy sets, and fuzzy aggregation.
- Evolutionary Computation, which is formed by Genetic Algorithms, Genetic Programming and Swarm Intelligence.
- Probabilistic Computing, which includes several statistical techniques [184].

Some authors add to this subset the research fields of Machine Learning, which in this context deals with classifier ensemble systems, and Chaos Computing, which includes some techniques based on Chaos Theory mainly employed in feature extraction. In the following sections we describe the application of CI to BCI.

### 3.4.2.1 Feature Extraction and General Purpose Classification for BCI

Different classification techniques have been used in the BCI application field. They are described in the following paragraphs.

First Qin et al. [185] propose the employment of kernel machines, particularly Support Vector Machines (SVM), for classifying data of BCI competition (see Section 3.4.1) in two applications for non-invasive cursor control and invasive motor imagery. They claim this kind of system, which is qualified as semi-supervised, can be used in order to reduce the need of training data that characterizes spatial filtering techniques (see Section 3.4.2.3).

One further work making use of data acquired with invasive electrodes is this presented by Sanchez et al. [186]. They make use of a combination of signal processing and classification algorithms for invasive BCI. One interesting statement of the paper relates to the necessity for adaptive methodologies to go down in the scale of analysis in order to get into the spike domain. Since we are mainly interested in the non-invasive BCI applications we do not further analyze their approach.

A further work we briefly mention has been presented by Herman et al. [187], where the performance of different spectral features, namely power spectral density (PSD) techniques, atomic decompositions, time-frequency (t-f) energy distributions, continuous and discrete wavelet approaches, for two-level motor-imagery classification is analyzed in terms of classification accuracy (CA). The performance of different classifiers, namely (to be completed), is analyzed.

It is worth mentioning as well the feature extraction stage of Bin et al. [188], which describes a BCI system based on visual evoked potentials (VEPs). The paper compares the three stimulus modulation approaches used in current VEP based BCIs:

- In time modulation (t-VEP) different targets can be focussed by the user. Each one of them flashes asynchronously. Every flash elicits an evoked potential with a constant delay from the flash. These potentials can be detected by averaging various epochs.

This system requires synchronization between the stimuli presentation and the EEG recording.

- In frequency modulation (f-VEP)based BCI, each target is flashed at a different frequency generating a periodic sequence of evoked responses with the same fundamental frequency as that of the flickered stimulus as well as its harmonics.
- In pseudorandom code modulation (c-VEP) flash instants follow a pseudorandom sequence.

The paper focuses on f-VEP and c-VEP, since t-VEP has a lower performance. The methodology is based on channel selection first. The channel with maximal VEP amplitude is selected as signal channel. This results in Oz electrode signal selection. Very interesting is the selection of the reference channel, which is based on this electrode delivering training accuracy. No classifier is used but a template matching based on the correlation coefficient between the template and the current signal spectrum. f-VEP BCI has the advantage of little or no training and simple system configuration, while the c-VEP based BCI has a higher communication speed. The results of both systems are the following:

- f-VEP: ITR 39.7 bit/min for 88% train and 85% test accuracy
- c-VEP: ITR 92.8 bit/min for 95% train and 91% test accuracy

Larger accuracy of c-VEP is due to its wider spectrum (5-25 Hz). This makes it more robust in front of interferences of basal brain activity (which is characterized by narrow band signals). From a PatRec point of view this can be understood as well. Features in f-VEP are formed by vector almost 1 and the rest 0 activations, while c-VEP presents real valued vectors in all components. Given the similar performance of both systems, ITR is what rules. f-VEP is suitable for less option systems, e.g. wheel chair, whereas c-VEP for more options, e.g. speller. It would be interesting to know if we can reproduce this by putting electrodes in the occipital part, since they use a unique electrode in Oz.

The most complete review of classifiers for BCI applications has been presented by Lotte et al. [189]. Extensive lists of different approaches can be found in this paper grouped by BCI paradigm. Moreover performance levels are given. References to feature extraction methodologies are given as well.

### 3.4.2.2 Projection Techniques for BCI

In this section we analyse unsupervised projection techniques used in BCI. These are mostly focused on ICA employment. Kachenoura et al. [190] explore different methods for parameterizing ICA and its applications to BCI. They give an introduction to the widely used algorithms within ICA, namely SOBI, COM2, JADE, ICAR, FastICA, and INFOMAX. The application of these techniques in the BCI field is also explained. They claim that an appropriate selection of the ICA algorithm significantly improves the BCI performance.

Lin et al. [191] describe a system to detect drowsiness and distraction in drivers. Although the paper does not explain directly a BCI application its methodology can be of interest for such application field. They systematically show the application of ICA as a preprocessing stage. Then the components are separated between signal and artifact by applying 3 different methodologies: neural network / SVM classifiers, adaptive feature selection mechanism (AFSM), and k-means. In the second case the selected features are mapped into a drowsiness estimation through the application of a neuro-fuzzy system denoted as ICAFNN.



### 3.4.2.3 Supervised Projection for BCI

Beyond unsupervised projection techniques currently most systems use a supervised feature extraction stage, mostly based on Common Spatial Patterns. These are claimed to be presented for the first time in [194][196]. In this context, [195] compares the performance for BCI classification of different types of Common Spatial Pattern algorithms. The paper describes in detail CSP filters and variants; theoretical background and implementation. It focuses particularly on single-trial classification, revealing tricks of the trade for achieving a powerful CSP performance. A multi-class approach of CSP called One Versus the Rest is presented by Xu et al. [192].

Other supervised projection approaches are used beyond CSP and described in the following paragraphs. Coyle et al. [196] present neural networks for a brain-computer interface (BCI) application. Features are extracted from the EEG in a two-class motor imagery paradigm. Features are extracted based on morphology of the time series with a supervised approach based on neural networks attaining the class separability. Interesting is that they use two NN, one per class (instead of using a multi-class approach). Lastly linear discriminant analysis (LDA) is used in the classification stage.

Coyle [197] proposes to use a third type of filtering for BCI processing besides the usual spatial and spectral filtering. It illustrates how neural networks are employed in a prediction based preprocessing framework, referred to as neural-time-series-prediction-preprocessing (NTSPP), in an electroencephalogram (EEG)-based brain-computer interface (BCI). NTSPP has been shown to increase feature separability by mapping the original EEG signals via time-series-prediction to a higher dimensional space. The paper implements this temporal filtering through two different approaches a self-organizing fuzzy neural network and a multilayer neural network trained through back-propagation. Both types of networks are trained in order to answer to particular classes in a feature space of larger dimensionality than the input one, i.e. for  $M$  channels and  $C$  classes projects into at least  $M \times C$  space. After this temporal filtering, CSP is applied. Interesting is the employment of a projection space taking into account the eigenvectors of maximal eigenvalues (as usual) but these of minimal eigenvalues as well. The results obtained are comparable to these obtained at Starlab with a simpler approach.

Poli et al. [199] describes an interesting application based on P300 evoked potential, which was first described by Citi et al. [198]. The system implements a so-called analogue approach. Four flashes placed on the 4 cardinal points of a screen are used to elicit P300. Wavelet Transform is applied to the signal. The selection the right features from a total set of 20000 is done through the application of Genetic Algorithms. The selected features are then linearly combined to deliver the movement into one of the possible directions. The movement is computed differentially, i.e. movement up is the activation of up p300 minus the activation of down p300. The coefficients of the linear combination are as well optimized by the GA. The resulting approach is single trial and the mouse cursor moves once per second. In our opinion they average various trials since they obtain 1 answer per second (we can have 10 to 15 flashes per second and there are only 4 targets). They talk about an analogue approach because the amount of movement is proportional to the result of the difference former mentioned, and no binary decision is taken. It should be tested how well the user can voluntarily control this difference.

### 3.4.2.4 Classifier ensembles for BCI

One classification paradigm being currently very popular is this of classifier ensembles or multi-classifier systems. This is a paradigm originated within the Machine Learning community that has flowed into other research areas. In this kind of system different classifiers are applied to a data set and then the results are fused through an operator. Some works taking into account the application of this paradigm in BCI applications can be found in [189] and in the following paragraphs.

The first approach we found in this context is this related to the BCI competition iii (see section 3.4.1) by Shangkai Gao and colleagues [192]. Although we have not found any paper in the literature describing the ensemble approach Cester et al. have analyzed its structure [193]. They make use of three different classifiers (LDA, fuzzy C-means, and SVM) in a bagging [184] structure. The fusion operator is the average. Performance is only acceptable when dealing with trial classification and not on a sample basis (as it would be desirable for a BCI system).

Hammon et al. [194] present a further ensemble classifier approach for BCI. ICA is used as feature extraction stage and multi-classifier stage in thence applied. In this case the classifier is used for each of the extracted features and adapted to each of the users. Average is used as fusion operator for the overall so-called meta-classifier. Fazli et al. [200] tune as well the classifiers to subject-specific training data in a 45 subjects database. In this case the ensemble is formed by subject-specific temporal and spatial filters. They claim such a system is able of real-time BCI use without any prior calibration (aka training).

### 3.4.3 EEG and BCI toolkits

Since one of the activities in AsTeRICS attains the extension of a toolkit for BNCI, we give in the following sections a state of the art of the existing toolkits for BNCI. We have included some content of the corresponding web sites for the sake of completeness.

#### 3.4.3.1 BCI2000

As it can be found in their web site<sup>126</sup>, “*BCI2000 is a general-purpose system for brain-computer interface (BCI) research. It can also be used for data acquisition, stimulus presentation, and brain monitoring applications.*”

To the best of our knowledge this is the most used toolkit related with BCI. It is quite complete, meant to work real time (although off-line data analysis can also be performed) and the source code is available, bringing the opportunity to the researchers to have full control over the parametrization of the applications. Its programming language is C++. A lot of documentation can be found in their web site, and also a useful user forum.

A user can download the executable and begin to perform BCI experiments. Through the GUI the user can change some parameters, but of course does not have full control over the BCI2000 software. In order to have full control over the BCI2000 platform, the source code needs to be downloaded, modified for some specific purpose and then compiled. That way it is possible to add functionalities to the BCI2000 platform, such as extracting different sets of features, implement different classifiers, fully modify the scenarios and so on.

---

<sup>126</sup> <http://www.bci2000.org/BCI2000/Home.html>

### 3.4.3.1.1 Stimulus Presentation

This executable allows the user to record data while a visual and/or an auditory stimulus is presented. No feature extraction or classification is done in this task.

### 3.4.3.1.2 Feedback Demo

In this task the user can control the vertical movement of a dot in order to hit a target on the right side of the screen. This dot has a constant speed (from left to right). In order to perform this task it is recommended that the user performs a training first, i.e. perform the Stimulus Presentation task. After the training is performed, the best parameters for each specific subject are extracted and used in the Feedback Demo task.

The classifier used in this case is linear and it is quite simple. The LinearClassifier computes a projection of a high-dimensional signal feature space onto a low-dimensional classification space. Thus, each dimension of classification space is a linear combination of signal features. Input data has 2 indices (N channels x M elements), and output data has a single index (C channels x 1 element), thus the linear classifier acts as a  $N \times M \times C$  matrix, determining the output after summation over channels and elements:

- In a BCI based on demodulation of a periodic EEG component (such as the Mu Rhythm BCI), the Linear Classifier's input is the time-varying amplitude or power spectrum for a number of spatially filtered EEG channels. Its output is later normalized with respect to mean and variance, and then used as a control signal to determine cursor movement.
- In an ERP-based BCI (such as the P300 BCI), the Linear Classifier's input is a sequence of averaged EEG time courses obtained in response to a number of stimuli, and its output is considered to represent the likelihood for each of these responses to be an ERP.

### 3.4.3.1.3 Cursor Task

Similar to the Feedback Demo task, but in 2 dimensions. The idea is now to control the vertical and horizontal movement of a cursor to hit a target.

### 3.4.3.1.4 P300 Speller

The user watches a matrix of flashing letters. When the letter the user is thinking of flashes, then an ERP called P300 appears. The idea of this task is to input letters in the computer by paying attention to those flashing letters.

The P300 Speller use the SWLDA classifier. The idea behind this classifier is to obtain a linear model that fits a data-set of stimuli by using multiple linear regressions and iterative statistical procedures. These two methodologies attain the selection of the significant variables that are included in the final regression.

The SWLDA algorithm can be summarized in the following steps:

- STEP 1: Compute the autocorrelation matrix to select the variable most highly correlated to the observations. The selected variable is included in the linear model.
- STEP 2: Fit an initial model by performing a linear regression equation with the included variable using least squares.

- STEP 3: Check if the variable added is that one which makes greatest improvements in "goodness of fit." That is, remove the added variable if its variance contribution in the regression is insignificant at a specified F level (F-test). If the added variable is removed go to Step 5, else go to Step 4.
- STEP 4: Add the variable to the model if there is a significant variance reduction at a specified F-level.
- STEP 5: Remove the added variable from the correlation matrix. Go to step 1.

The method terminates when no single step improves the model.

### 3.4.3.2 SPM8

Statistical Parametric Mapping refers to the construction and assessment of spatially extended statistical processes used to test hypotheses about functional imaging data. These ideas have been instantiated in software that is called SPM. The SPM software package (free Matlab Toolbox) has been designed for the analysis of brain imaging data sequences. The sequences can be a series of images from different cohorts, or time-series from the same subject. The current release is designed for the analysis of fMRI, PET, SPECT, EEG and MEG.

Although SPM is more focused on fMRI data, it can also be used for for MEG/EEG analysis, in particular about Source Reconstruction and Dynamic Causal Modelling (DCM), which is a spatio-temporal network model to estimate elective connectivity in a network of sources. All these new methods, mostly based on a Bayesian approach, have been implemented in SPM8. We provide a range of tools for the full analysis pipeline, i.e., you can take your raw data from the MEG or EEG machine, and put it through SPM, starting from the conversion of the data through to a statistical analysis of sensor level or source reconstructed multi-subject data or Dynamic Causal Modelling. It does not have classifiers implemented. In fact SPM is not focused on BCI at all. It is more intended to be an off-line analysis software package.

### 3.4.3.3 EEGLAB

As they explain on their web site<sup>127</sup>, "*EEGLAB is an interactive Matlab toolbox for processing continuous and event-related EEG, MEG and other electrophysiological data incorporating independent component analysis (ICA), time/frequency analysis, artifact rejection, event-related statistics, and several useful modes of visualization of the averaged and single-trial data. First developed on Matlab 5.3 under Linux, EEGLAB runs on Matlab v5 and higher under Linux, Unix, Windows, and Mac OS X (Matlab 7 recommended).*"

EEGLAB is missing a collection of classification routines. It is quite useful to visualize data, extract feature and read different formats of data, but we cannot perform classification tasks with EEGLAB. On the other hand, EEGLAB is meant to be used in an off-line mode, so no real time functions are implemented in this toolbox. As a conclusion I would say that EEGLAB is a very powerful toolbox for EEG data analysis, but not useful for BCI applications by itself. In this context it very worth commenting the existence of the ICA methodology, which can be used for data pre-processing in BCI applications. There is a submodule of EEGLAB called MatRiver which is meant to be used in real time applications (BCI), but it is still beta and not much documentation can be found yet.

---

<sup>127</sup> <http://sccn.ucsd.edu/eeglab/>

### 3.4.3.4 OpenEEG

OpenEEG is a big project mostly related with EEG hardware (build your own) but also has interesting software packages. As they explain in their own web site, “Unfortunately, commercial EEG devices are generally too expensive to become a hobbyist tool or toy. The OpenEEG project is about making plans and software for do-it-yourself EEG devices available for free (as in GPL). It is aimed toward amateurs who would like to experiment with EEG. However, if you are a pro in any of the fields of electronics, neurofeedback, software development etc., you are of course welcome to join the mailing-list and share your wisdom.”

ABI BCI software<sup>128</sup> is a simple software for the OpenEEG<sup>129</sup> hardware, that implements an experimental Brain Computer Interface (BCI). Nowadays, BCI research is a highly active field, but the existing technology is still immature for its use outside of a lab's settings. The ABI software tries to provide a simple tool for hobbyists to do experiments on its own with BCI technology.

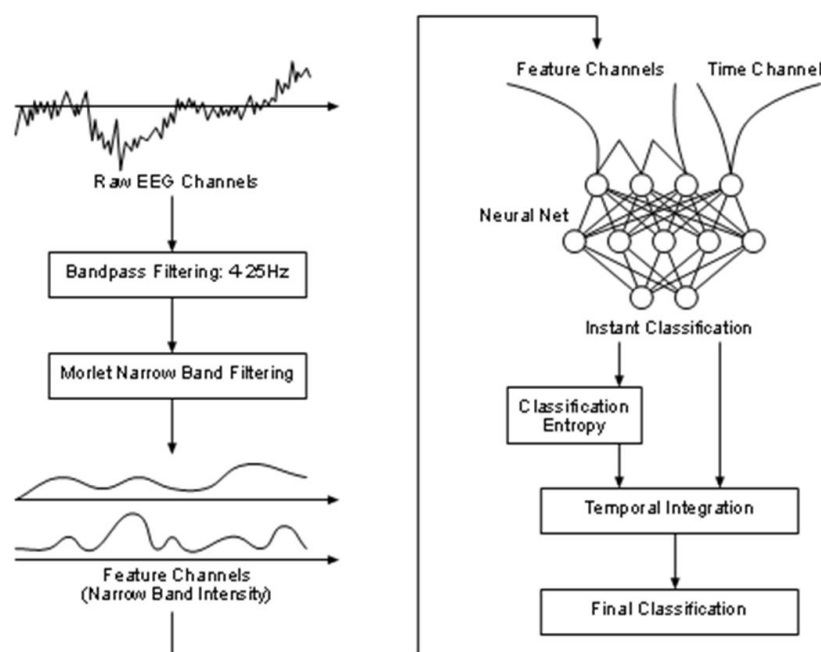


Figure 76: Basic structure of ABI BCI software modules.

The design of the classifier is heavily based on Christin Schäfer's design used for the Dataset III of the BCI Competition II. Instead of using a Gaussian multivariate Bayesian classifier, here we use a neural net to obtain the classification for each time instant  $t$ . Those outputs are then integrated in time using a weighted sum. The idea is simple: outputs with low confusion should have higher weights.

These steps are described as follows:

- Acquire raw EEG data. Filter the EEG channel using a bandpass filter between 4 and 25 Hz.
- Use Morlet Wavelets to extract local frequency information. Compute their absolute value. These are the feature channels.

<sup>128</sup> <http://www.dcc.uchile.cl/~peortega/abi/>

<sup>129</sup> <http://openeeg.sourceforge.net>

- Feed a two layer feedforward neural net with the frequency information and an additional time channel (restarts at zero at the begin of every trial). The neural net has two layers: the first weight layer uses the tanh activation function, the second a normal logistic activation. The net is trained using the cross-entropy error as the optimization criterion. The output of the neural net is the estimated instant classification.
- The final classification is obtained after performing a weighted time integration of the instant outputs, where individual weights are higher for low entropy outputs.

### 3.4.3.5 Biosig

BioSig<sup>130</sup> is an open source software library for biomedical signal processing, featuring for example the analysis of biosignals such as the electroencephalogram (EEG), electrocorticogram (ECoG), electrocardiogram (ECG), electrooculogram (EOG), electromyogram (EMG), respiration, and so on. Major application areas are: Neuroinformatics, brain-computer interfaces, neurophysiology, psychology, cardiovascular systems and sleep research. The aim of the BioSig project is to foster research in biomedical signal processing by providing open source software tools for many different applications. Generally, many concerns have to be addressed in this scientific field. BioSig handles this by providing solutions for data acquisition, artifact processing, quality control, feature extraction, classification, modeling, data visualization, and so on. Everything in this project is freely available under the GNU General Public License.

### 3.4.3.6 EEG/MRI Matlab Toolbox<sup>131</sup>

Firstly, this is not a signal processing toolbox. Of course, once the data is loaded, there are many matlab functions available for data processing, but few of them are integrated into a GUI interface here. At present, there are no specific functions for processing raw EEG, such as filtering, averaging, etc. For examples of signal processing tools, see the matlab signal processing toolbox and the links below, especially EEGLAB.

This toolbox has been developed to facilitate quick and easy import, visualisation and measurement for ERP data. The toolbox can open and visualise ERP averaged data (Neuroscan, ascii formats), 2D/3D electrode coordinates and 3D cerebral tissue tessellations (meshes). All the features can be explored quickly and easily using the example data provided in the toolbox. The GUI interface is simple and intuitive. The following lists the features already available and some items that could be developed. There are no classifiers available in this toolbox.

### 3.4.3.7 EMEGS

EMEGS<sup>132</sup> (ElectroMagnetic EncaphaloGraphy Software) was programmed to analyse data collected with high resolution EEG and MEG. Therefore there are no classifiers implemented.

### 3.4.3.8 OpenVibe

OpenVibe<sup>133</sup> is an open source framework for Brain Computer Interfaces. OpenVibe is described in more detail in section 7.3.6.

---

<sup>130</sup> <http://biosig.sourceforge.net/>

<sup>131</sup> <http://eeg.sourceforge.net/>

<sup>132</sup> <http://www.emegs.org/>

### 3.4.3.9 BCPy2000

This toolbox originated from porting BCI200 into the Python programming language. The classification used within BCPy2000<sup>134</sup> comes from the SciPy library<sup>135</sup>. It contains few classifiers, based on radial basis functions and on genetic algorithms. Deeper description of the toolbox is given in the BCPy2000 web site<sup>136</sup>.

### 3.4.3.10 Conclusions

Not many BCI toolboxes are available. We have identified 9 different toolboxes, 4 of which focus on BCI, the other 5 are just signal analysis or visualization toolboxes. In general not many of these modules contain classification functions. For instance, the BCI toolboxes are not very much focused in the classification part. Usually only one classification approach is implemented per toolbox. Two widely used classification approaches have been identified: LDA and NN.

On the other hand a very complete classification routine is implemented in the BIOSIG signal processing toolbox. But in this case it is meant to work in off-line mode. The main conclusion is that there is a lack of classification routines for working in real-time in the BCI toolboxes analyzed. The reasons are not very clear but it can be due to:

- The real-time implementation of classifier functions might be not trivial for some of the classification approaches
- The focus of the BCI toolboxes is to offer an open platform for the researcher to implement their own routines (feature extraction and classifiers) and thus the functionalities are not very extended but are very flexible.

Most of the implemented classifier functionalities are integrated in the BIOSIG toolbox. Therefore it looks reasonable to take this toolbox as starting point. First it would be interesting to establish common interfaces with the toolbox to be developed. Moreover the functionalities of BIOSIG could be extended in the following directions in the Evaluation Suite to be implemented within AsTeRICS project works:

- Projection methodologies including a working version of CSP and other ones like: PCA, ICA, kernel PCA, and non-linear projections, e.g. based on fuzzy integrals
- Decision trees and similar tree methodologies
- Multi-classifier ensembles and other complex classifier functions based on re-sampling methods: bagging, boosting
- Data fusion operators for undertaking multi-modal and multi-classifier BCI approaches
- Genetic algorithms for system optimization
- Fuzzy control algorithms, i.e. Mandami, Takago-Sugeno

In this context the work of Lotte et al. [189] can serve as the theoretical basis for the further extension of the toolkit. The approaches given in the former list are not included in that review.

---

<sup>133</sup> <http://openvibe.inria.fr/>

<sup>134</sup> <http://mloss.org/software/view/145/>

<sup>135</sup> <http://www.scipy.org/>

<sup>136</sup> <http://www.bci2000.org/wiki/index.php/Contributions:BCPy2000#Location>

## 4 Computer Vision for Assistive Technologies

Computer Vision is a promising technology for new Assistive Technology solutions, because vision based tracking of body features enables computer usage or interaction with the environment for people with severe impairments by a non-invasive technique. In recent years, the computer vision technologies and algorithms have constantly increased, and the cost of the equipment decreased significantly. In the subsequent sections, existing alternative communication modes via vision processing of different parts of a human body are addressed. In particular, computer vision based head-, limb- and eye tracking systems will be introduced.

### 4.1 Tracking of fingers, limbs or shoulder

Different body parts could be used as pointer, but it should be observed that to our best knowledge there is no commercially available technology today for supporting any alternative to arm-finger pointer.

The system's usual set-up includes an optical camera with the targeted body part located in camera's field of view. The targeted body part can be detected with an image edge detector (Sobel, Canny, Laplace). During recognition, a matching of detected edges with the targeted body's simplified 2D model (top-down approach) is done. The following methods are frequently used: MCMC (Markov Chain Monte Carlo, [46]), belief propagation [47], simple heuristics [48], or affine and scale invariant edge matching. Frequently, the body anatomic (articulated) model is used for image extracted feature and body part model guided matching [49].

Pose estimation (pose could be used as pointer) is frequently based on optical flow, MCMC, HMM (Hidden Markov Models) [50], Viterbi algorithm, particle filter (Sequential Monte-Carlo method) or through "tracking-by-detection" framework. Hereafter, some devices which introduce new concepts for alternative object pointing (selection) tasks are presented.

#### 4.1.1 Finger tracking<sup>137</sup>

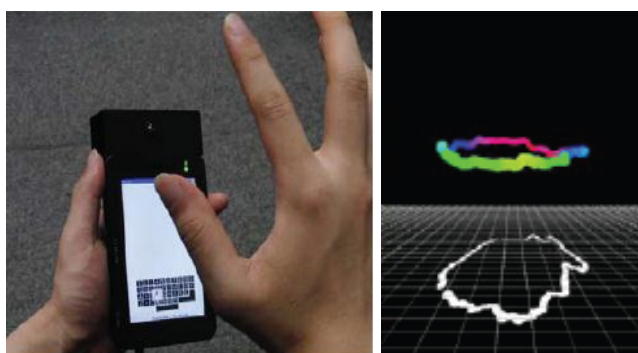


Figure 77: In-air typing system (University of Tokyo)

This Ishikawa Komuro Laboratory, University of Tokyo created a system which implements the concept of a distant ("in-air") non-physical touch: fingertip spatial movements are interpreted as different gestures linked to a usage of a standard mouse. Fingers are detected

<sup>137</sup> [http://www.k2.t.u-tokyo.ac.jp/vision/typing\\_system/index-e.html](http://www.k2.t.u-tokyo.ac.jp/vision/typing_system/index-e.html)



with a single high-frame-rate optical camera attached to a portable device (named “In-air typing system”) having a small (PDA-size) screen (see Figure 77).

The binarized fingertip image is used to estimate its 3D motion and posture. This device can simulate a standard PC mouse and could be useful for upper limb impaired subjects. Classic mouse operations are implemented as follows:

- Mouse spatial displacements follow changes of finger spatial position
- Mouse click gesture is implemented by detecting finger temporal movement over almost the same spatial position (dwelling time)

The system allows the execution of the following actions

- Writing a text through virtual keyboard displayed on the screen
- Displaying a picture (with a click action)
- Picture zoom and scroll (via specific finger positions)
- Drawing any 3D lines according to 3D fingertip position

Advantages and drawbacks of finger tracking are described in Table 6:

Finger tracking advantages	Finger tracking disadvantages
There is no restriction of finger spatial position as long as they stay in the field of view of the tracking camera	The only available system is a laboratory prototype
Finger detection is stable	No data on system evaluation with impaired subjects are available
The system is a portable device	
Precision of finger position detection is high (one pixel)	

**Table 6: Advantages and disadvantages of finger tracking**

#### 4.1.2 Upper Limb tracking

The University of Southern California works on a system which uses the concept of a pictorial structure (not fully connect graph) [47] to represent the investigated part of human body [51], [46]. The system detects and tracks human upper limbs from video monocular stream.

The tree structure of the upper limb (represented by pictorial structure) is matched with the acquired image. A combinatorial exploitation of the solution space over affine transforms and scales is guided by body global model. The limb tracking is done via the kernel based tracking [52] optimized with gradient-descent approach.

This project aims towards an adaptation of the system for motor impaired people. Advantages and drawbacks of limb tracking are described in Table 7:

Limb tracking advantages	Limb tracking disadvantages
There is no restriction of upper limb spatial position as long as it stays in the field of view of the tracking camera	The only available system is a laboratory prototype
The system performance is around 1 second for detection and in the order of milliseconds for tracking and seems to be compatible with motor impaired people	No data on system evaluation with impaired subjects are available
This method can be applied to the detection of any body part	

**Table 7: Advantages and disadvantages of limb tracking**

### 4.1.3 Shoulder & elbow tracking

In the frame of EU 6<sup>th</sup> FWP IST IP AMI (Augmented Multi-party Interaction) project, the University of Twente has proposed an approach to estimation and recognition of poses [53] which generalizes detection of different body parts, and in particular different limbs (shoulder, elbow), proposed by Sidenbladh & Black [54]. The system detects and tracks human limbs from video monocular stream. A combination of image descriptors (edges, skin color and texture, motion, etc.) is used in order to make the detection more robust. Advantages and drawbacks of shoulder/elbow tracking are described in Table 8:

Shoulder/elbow tracking advantages	Shoulder/elbow tracking disadvantages
There is no restriction of limb spatial position as long as it stays in the field of view of the tracking camera	For precise limb tracking, the system uses markers attached to the human body
This method can be applied to the detection of any body part (hands, elbows, shoulder, face, head) or their spatially allowed combination	No data on system evaluation with impaired subjects are available
The method location error is about 3-5cm	The only available system is a laboratory prototype

**Table 8: Advantages and disadvantages of shoulder/elbow**

## 4.2 Head and face tracking

In computer vision supported head tracking systems, head movements are tracked using a dedicated sensor operating in visible or infrared spectrum attached to a host computer. The movement information is translated into proportional movements of a standard interaction device such as a computer mouse (mouse emulation). If a vision sensor is used, almost any face characteristic can be utilized for mouse emulation. If an infrared camera is used, a tiny target placed in a predefined head position (forehead, glasses, hat, nose, etc.) must be used as this target is tracked.

A head mouse replaces the standard mouse and is particularly useful for upper limb impaired subjects. This peripheral can connect to a (fixed or portable) host system via an USB port or wirelessly. Common mouse operations are performed with a head mouse as follow:

- Mouse spatial displacements are deduced from head movements (orientation and scale)
- Mouse click (or object selection) can be implemented via adaptive switches such as sip/puff switches and mouse button software; a software click is frequently implemented in two steps: positioning the pointer on the target and keeping it there for a predefined (user profile defined) time period (dwell time)

Used in connection with on-screen keyboards, head mice allow head-controlled access to all functions for both these peripherals including Internet. The user can thus perform such tasks as e.g. drawing, gaming and graphics applications such as Computer Aided Design (CAD).

Some systems allow mouse cursor control using image analysis from a standard webcam. Examples are Face Mouse<sup>138</sup>, HeadMouse2<sup>139</sup> and Mouth Mouse<sup>140</sup>:

<sup>138</sup> <http://www.oskaworld.com/category/free-face-mouse.php>

<sup>139</sup> <http://robotica.udl.es/headmouse/headmouse2/headmouse2e.html>

<sup>140</sup> [http://www.ydp.eu/en/specialneeds/mouth\\_mouse](http://www.ydp.eu/en/specialneeds/mouth_mouse)

*Face Mouse* is non-commercial and open source software. It uses a non commercial version of FaceAPI from Seeing Machines<sup>141</sup>. There are two versions of Face Mouse: Face Mouse and Face Mouse with Click. The second version can do clicks using dwell time. The Face Mouse uses relative position movement.

*HeadMouse 2* is free software, designed specifically for motivating disabled people. It uses relative position movement. The mouse click can be emulated by dwell time, blinking the eyes or opening the mouth.

#### 4.2.1 Tracking of facial features

Different facial features are used for man-computer communication. There are: nose, global face, eyebrows, and their combination, etc. Frequently, the targeted characteristics are detected in the image taken during the calibration and tracked during the interaction. Classic mouse operations are performed with such a system as follows:

- Mouse spatial displacements are deduced from nose/head eyebrows movements (orientation and scale).
- Mouse clicks (or object selection) are implemented through the left or right eye double blinks or through dwelling time.

#### Examples of systems tracking facial features:

- The Institute of Information Technology, Ottawa, Canada [55] and CNR, Canada, investigated the nose mouse (nose-steered mouse) concept and system (see Figure 78). NosePaint is a graphics editor which uses only the nose and eyes.



Figure 78: Nose mouse, University of Ottawa and CNR, Canada.

- Facial mouse ([57], [58]) combines eye and nose and recognizes the movement and gestures of the face.
- A new method for information input to a PC system which combines mouth opening detected through image processing and weak hand motor coordinated action has been proposed in association with a PC keyboard [59] (for alphabetic and syllabic writing system).
- MouthMouse/LipMouse is a commercial product [56] that moves the mouse cursor using a webcam. Mouth Mouse uses relative position movement. The mouse clicking is simulated by moving the mouth. To operate mouse clicks the user can make

<sup>141</sup> <http://www.seeingmachines.com>

actions such as: opening the mouth, showing the tongue, forming lips as for kissing (see Figure 79).

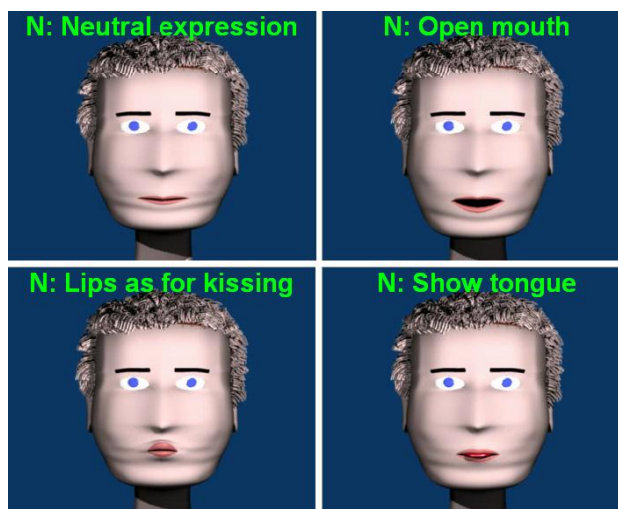


Figure 79: LipMouse actions

### 4.3 Eye- and gaze tracking

One of the most salient characteristics of man, and especially his/her face, are the eyes. The eyes play a fundamental role for man-environment interaction; this is also the most promising means for new man-machine interactions modeling, especially for upper limb impaired subjects. Therefore, non intrusive, robust and fast eye detection and tracking are the basis for development of new technologies for interactions and for attention responsive technology (basis for design of indoor/outdoor gaze-based environmental control accessible for impaired subject).

Eye-gaze systems allow a user to control a computer using only the movement of his or her eyes. Most of these systems use a monitor where the user is focusing, and move the cursor to that point. The selection is normally made using dwell time.

In general, two configurations for eye- and gaze trackers are possible:

- Remote eye trackers
- Head-mounted eye trackers

These two configurations are also respectively known as *passive* and *active eye monitoring*.

#### 4.3.1.1 Remote eye trackers

Remote eye trackers are not well-suited for applications where mobility is required (i.e. the relative speed should be very slow and the maximal displacement very limited for remote systems). Moreover, a planar surface is used as a mediation tool for interaction. In fact, one of the cameras is always placed in a manner that it can track the face, especially the eyes, of the user while the person is moving in his environment. A computer with a built-in or USB connected camera is the most popular implementation. The camera moves in the same frame of reference as the user does.

Desktop/remotely mounted systems are designed to accurately track pupil diameter and eye position relative to one flat surface (supposed to be parallel to the camera plan). The ASL desktop/remote solution<sup>142</sup> is referenced in more published papers than any other desktop/remote eye tracking system.

A desktop/remotely mounted solution is an excellent choice when participants will be viewing one flat surface or when wearing a headband is not practical. It will estimate a participant's line of gaze with respect to a single stationary surface in the environment. As with all of ASL EYE-TRAC®6 Series options<sup>143</sup>, the measurement is displayed as a cursor superimposed on the image from remote mounted scene cameras or other video sources showing the participant's field of view. A digital recording of this image can be recorded for presentations and data analysis.

#### 4.3.1.2 Head-mounted eye trackers

Even if there has been significant emphasis on remote eye tracking, head-mounted eye tracking systems are experiencing a renaissance due to new challenges [66]. They are built around a camera (or two in case of a binocular configuration) fixed on the head which tracks the movements of one eye (or both eyes). In such situations, head-mounted devices are used for interaction via a screen (a mediated interaction via a 2D support). New head-mounted eye trackers use two cameras: one for (dominant) eye tracking and another for scene scanning. These new eye trackers will be used for direct interaction with a 3D world.

### 4.3.2 Methods for computer vision based eye tracking

The basic techniques for computer vision based eye tracking are the limbus technique (Torok), the Hirschberg (corneal reflect) technique and image processing techniques (VOG, video-oculography); all these measurement techniques are based upon image- and computer vision processing in illumination controlled environments and should be applied on the dominant eye. Other techniques, as exploitation of bioelectric signals (EOG) for eye movement detection are described elsewhere in this document (see Section 3).

#### 4.3.2.1 Limbus technique

The iris-sclera boundary is named limbus[64]. The eye position is linked to the ratio of sclera and iris surfaces registered by a measurement system. Therefore, if the limbus is illuminated, this ratio determines the quantity of light reflected by this eye. Figure 80 shows the Vision Key system implementing limbus principle.



Figure 80: Vision Key system (VisionCan, Canada)

<sup>142</sup> <http://asleyetracking.com/site/>

<sup>143</sup> <http://www.asleyetracking.com/site/ProductOverview/EYETRAC6Series/tabid/56/Default.aspx>

Advantages and drawbacks of limbus eye tracking are described in Table 9:

Limbus eye tracking advantages	Limbus eye tracking disadvantages
The system is inexpensive as it requires only a simple light source and light detector	The system environment must be well controlled (scene surface reflection properties and scene illumination) as the ambient light contains multiple specular reflections and diffuse components which influence the precision
	The light detector must be in-line with the eye
	Measures are subject to head movements
	Only the horizontal measures are reliable as the vertical movements are disturbed by eyelids closing
	Measures are sensitive to the important variations of illumination
	High price (7000\$)

**Table 9: Advantages and disadvantages of limbus eye tracking**

#### 4.3.2.2 Hirschberg technique

Hirschberg (1885) has proposed the gaze orientation estimation method which is largely used in today's eyetrackers [65]. Hirschberg has linked the position of the corneal reflection of an IR light source to the position of eye pupil centre. The IR camera system measures this distance when the eye explores a planar image. These eye trackers start to be used in assistive devices for handicapped people.

Commercially available eye trackers differ by the reference used in measures:

- the allo-centred eye trackers use a sensor and a designation screen having a common fixed relative position and facing the user (many eye trackers use this as reference),
- the ego-centred eye trackers are wearable/portable (helmet type); the sensor and designation screen are attached to the system and include a head tracking device (example : Eye Command, Handicom, Sweden),
- both frame of references.

Advantages and drawbacks of Hirschberg gaze tracking are described in Table 10:

Hirschberg gaze tracking advantages	Hirschberg gaze tracking disadvantages
The pupil (rather than the limbus) is the eye feature that is easier to detect than a contour as both the sclera and the iris strongly reflect infrared light. Tracking the pupil contour is preferable as the pupil contour is smaller and induces a higher gradient than the limbus. Due to its size, the likelihood of pupil occlusion by eyelids is low	The infrared image technique cannot be used outdoors during daytime due to infrared components of sun light
The method uses a IR light source which eliminates uncontrolled specular reflections by active eye illumination with a uniform and controlled infra-red light not perceivable by the user	This method is computationally very heavy and requests a very powerful computer
This non-invasive method allows measuring eye position independently of the head-movements (as far as the head stays in the camera's field of view)	Allo-centred systems measure can be disturbed by uncontrolled head/body movements (the eyes are out of the camera's field of view); a fixed head is usually requested in clinical measurements
Its precision is high (especially if a 30 point calibration method is used; a simplified (5 points) calibration method also leads to good results	Ego-centred eyetrackers suffer from weak robustness to the illumination changes, their size and reliability

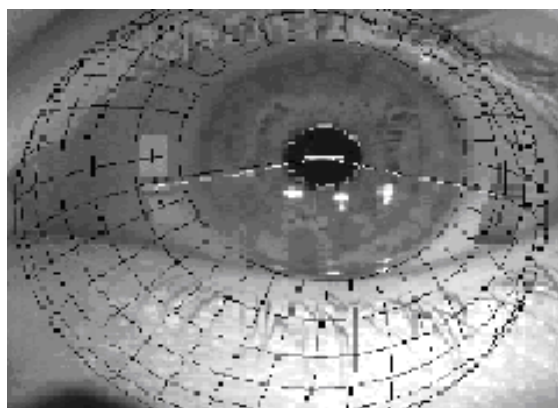
Hirschberg gaze tracking advantages	Hirschberg gaze tracking disadvantages
	Both types suffer from spurious reflections due to optical glasses and their metallic branches. The usage of systems is not easy for bearers of contact lenses and subjects having dry eyes syndrome
	Professionals using such systems have to pay attention to real conditions in which the system is used
	High price (device ranges from 10,000€ to several hundreds)

**Table 10: Advantages and disadvantages of Hirschberg gaze tracking**

#### 4.3.2.3 Video-oculography

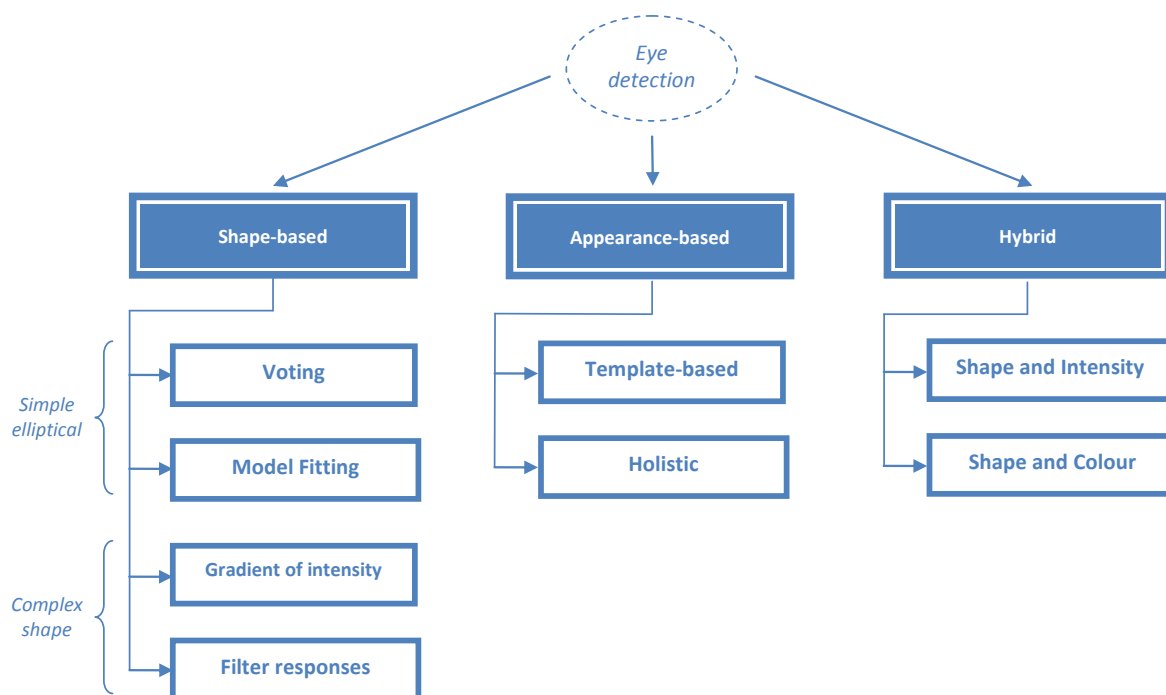
Video-oculography (VOG) is a non invasive technique, inherited from clinical research on vestibular (mainly vertigo) patients, for eye tracking using image processing technology [66] For example, the position of pupil and iris are used for 3D rotation angle estimation of the eye, see Figure 81.

The few available systems are usually head-mounted devices equipped with one or more optical cameras (with attached mirrors or not). Very few wearable eye trackers have been adapted for motor impaired persons. Two control approaches are possible: (1) a computer screen direct control or (2) eye tracking through subsequent frames for interaction with an environment (via ART, attention responsive technology [67]). The image stabilization sensor (such as an accelerometer or inertial measurement unit) and head pose estimator start to be integrated as part of an eye tracker.



**Figure 81: VOG technique principle used in University Hospital of Munich eyetracking**

VOG techniques for eye detection can be classified in: shape-based, appearance-based or hybrid techniques (see Figure 82).



**Figure 82: Eye detection methods**

*Shape-based approaches* (fixed or deformable shape model) rely on features or contour information. These approaches extract simple elliptical (iris, dark/bright pupil, limbus) or complex (eyelids, eye corners, eyebrows, cornea reflections) eye characteristics using an eye model.

The most popular algorithms for characteristics recognition are of four classes:

- Voting methods (Hough transform [68], isophote curvature [69]) ;
- Model fitting methods (RANSAC [70], snakes-like for deformable models [71], [72]);
- Gradient of intensity techniques (guided Gabor filters[73], neural-networks [74]; evolutionary algorithms [75] or variance projection orientation and scale invariant [76]);
- Filter responses (Gabor wavelets [77], circular filter [78]).

*Appearance-based approaches* use image photometric (such as colour distribution) characteristics for an eye direct detection in a spatial or a transform domain. Transform domain representations provide invariance to scene illumination by removing subbands that are highly sensitive to illumination. Two main subclasses of these methods are: (1) template-based and (2) holistic methods.

1. Template-based methods use both, spatial and intensity information. An image-template is created and a matching is done directly between the image and the image-template, These methods are not invariant to the scale and rotations (such as changes of head position) neither to interpersonal variations.
2. Holistic methods use the intensity distribution and do not consider pixel spatial information. Based on the intensity distribution, an object efficient descriptor in a latent space is created and a matching between modeled (prototype) and extracted descriptor is performed using a similarity measure. This method allows handling variations in appearance.



For appearance-based approaches, large training databases are usually necessary in order to obtain good recognition results under several poses and illumination conditions (classifier or regression models are usually built).

Correlation based methods [79], support vector machines [80], [81], HMM (Hidden Markov Models) [82] and Eigen (PCA, Principal Component) analysis [83] are the most popular approaches exploiting the image intensity. Filter responses such as radial basis classifiers [84], and Haar transforms (with Adaboost [85]) are most frequently implemented; a pose independent eyetracking with Haar wavelets has been also proposed [86].

*Hybrid approaches* try to combine the benefits from previous described algorithms in order to overcome their shortcomings. These methods can be divided in two classes: (1) shape and intensity and (2) shape and color methods.

1. Part-based models (shape model of several subcomponents) [87] and Active Appearance Models [88], [89] are the most popular. However, these methods do not support large head movements.
2. Shape and color methods have not been largely investigated. [90] uses a color model to roughly track the eye and a gray-scale active appearance model for precise localization.

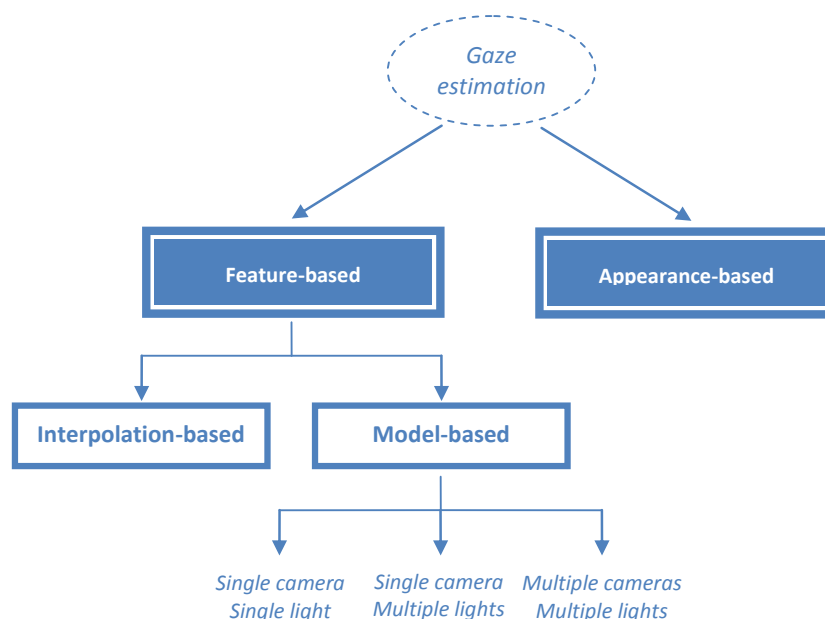
Advantages and drawbacks of VOG eye tracking are described in Table 11:

VOG eye tracking advantages	VOG eye tracking disadvantages
The VOG approach is a non-invasive allo-centred method which works with reasonable performance (time, precision) in indoor and outdoor environments	The implemented method requires frequently mirrors which makes the system cumbersome and heavy for a long usage
If used in a wearable ego-centred system, a continuous alignment of the camera's optical axis with user's gaze without restrictions of eye, head, leg or arm mobility is possible	The methods are frequently of a high temporal complexity and the convergence is strongly influenced by initialization
The system does not obstruct the end-user's field of view (or slightly in case of ego-centred)	The price of these systems is high as they require a camera of high quality, frequently with variable lenses. Low cost cameras can be used but they will induce constraints on end-user movements and illumination
The system is of high spatial precision	
Almost all methods can be applied to detect and track any part of the body or any object as the majority of methods model an eye via an ellipse	

**Table 11: Advantages and disadvantages of VOG eye tracking**

### 4.3.3 Techniques for computer vision based gaze estimation

The gaze estimation technique aims to determine gaze, ie. either the 2D-position of the gaze (known as the point of regard, PoR) or the 3D-direction of the gaze (known as the line of sight, LoS) [66], [91]. The most popular gaze estimation methods can be spread in two main classes (see Figure 83): feature-based and appearance based.



**Figure 83: Gaze estimation methods**

#### 4.3.3.1 Feature-based gaze estimation

Pupil and glint (first corneal reflection of the light) features are used by interpolation-based and by model-based gaze estimation methods [92].

The *interpolation (regression) methods* try to establish a mapping between image and gaze (2D or 3D) frame of reference in one of two ways: parametric ways (via a polynomial approximation [93]) or non parametric ways such as neural networks [94]. They avoid an explicit calculation of the intersection point between gaze direction and gazed object.

A very first real-time monocular approach to gaze estimation has been proposed for an IR source with head estimation (with mirrors and galvanometers) in 1974 [97]. For small pupil-glint vectors, a linear mapping allows estimating the point of regard (PoR). A monocular approach with a polynomial regression for PoR estimation and first-order linear regression for error in gaze estimation due to lateral head movement compensation improves the initial method [98]. The second IR source and spatial symmetries exploitation allow defining a simple calibration method via head movements.

A generalized regression method based on neural-networks using different geometric parameters of perception (pupil parameters, orientation and ratio of major and minor axes of the pupil, pupil-glint displacement) discards the calibration procedure (after the initial training) and offers an accuracy of 5° [94].

The control of tracking quality via coherency checks between calibration and system calculated data has been proposed [90].

*Model-based (or geometric) gaze estimation* computes the intersection point between gazed object and gaze direction; the latter is estimated through a geometric model of the eye [95], [96].

They model geometrically the human eye and 3D gaze direction vector using the physiology of image formation on the eye. A 3D scene point is estimated by the intersection of the line supporting gaze direction vector and the nearest object of the scene. Different methods exist

and depend on the different hardware configurations of the eyetracker: (1) single camera and single light (single glint), (2) single camera and multiple lights, and (3) multiple cameras and multiple lights.

1. In single camera and single light systems, the optical axis is estimated from a single glint and pupil centre [99]; however, this method is not head pose invariant. Villaneuva et al. [102] have shown that if the system is fully calibrated, the ellipse shape of the pupil and one light source are sufficient to obtain the head pose invariance.
2. For single camera and multiple lights systems, it was proven, by Shih [99] and Guestrin & Eizenman [101], that the gaze direction can be estimated from several light sources in fully calibrating system knowing cornea curvature (geometry of light source should remain unchanged). Yoo et al. [102] propose a method for PoR detection without calibration of two cameras. It is based on positions of the four light sources and its cross-ratio, an invariant of the projective geometry. Coutinho et al. [103] have improved the accuracy of Yoo's method, but the method remains not robust to depth-scale changes neither to interpersonal eye variability. The multiple light systems suffer from the risk that one of the multiple glints can disappear, what can significantly deteriorate the system's performance. A stereo vision for PoR estimation has been recently proposed [104], where a PoR is considered as result of the intersection of gaze vectors from two eyes.
3. Multiple cameras and multiple lights systems use cameras of different fields of view: a wide-angle lens camera increase the field of view of the system and allow free head movements (system "peripheral vision"), while narrow-angle lens cameras allow the precise gaze estimation (system's "fovea"). Through the system's pan and tilt capabilities (active vision), it tries to simulate the human eye behavior when a 3D world is explored [95]. Only recently, stereo vision techniques have been used [99]. It is an efficient approach to head mounted devices (with at least one camera oriented towards a user, and the other towards a scene). However, the stereo technique brings its own limits and additional processing volume (image matching, occlusion, etc).

#### 4.3.3.2 Appearance based gaze estimation

These methods [105], [106] do not detect the pupil and glint as this process can introduce errors. They use a raw image and map it directly to screen coordinates in order to avoid the knowledge of scene geometry and the camera calibration (these are implicitly modeled). No method is head pose invariant, a very hard problem as the eye may appear the same under different poses.

#### 4.3.4 State-of-art in eye trackers

In this section, eye trackers as assistive devices for people with severe disabilities are investigated (both, university prototypes and commercial devices). Eye trackers can increase the user's computer interaction capabilities by emulating a mouse, i.e. allowing object selection on the computer screen. A recent application is the eye (tracker) environment control via attention responsive technology (ART). In any case, the dwelling time or brain signals are today the only object selection (pointing) mean which could be associated with eye tracker (wearable and autonomous) technology.

#### 4.3.4.1 ISCAN ETL-500 remote eye-gaze tracker

Figure 84 shows the transportable remote eye-gaze tracker (EGT), built by Florida International University [107]. The system is US patented and built around the ISCAN® ETL-500 technology. This system implements the VOG principle for eye position measurement in an allo-centered (external) frame of reference. It works in illumination controlled (indoor) environments.

Two computing units working in parallel acquire image of the IR LED and estimate the user's gaze position and translate it (least-square algorithm) into the mouse current position (absolute or relative frame of reference). A camera should be connected to a host system via a USB port or wireless.



**Figure 84: ISCAN ETL500 eyetracker built by University of Florida.**

Advantages and drawbacks of ISCAN ETL500 eye-gaze tracker are described in Table 12:

ISCAN ETL500 advantages	ISCAN ETL500 disadvantages
The system reduces the mouse pointer jitter (up to 53%) making its trajectory smoother, and consequently allowing the user to have better control of the eye-based pointing device	The system is big and only poorly transportable
The system responds instantly to broad displacements of user gaze on a screen computer. Therefore, standard icons selection and clicking, and instantaneous responses are possible allowing basic interactions via internet (web navigation and email)	It suffers from parasitic reflections due to optical glasses and their metallic branches

**Table 12: Advantages and disadvantages of the ISCAN ETL500 eye-gaze tracker**

#### 4.3.4.2 Eyegaze Edge



**Figure 85: Eyegaze Edge and Eyegaze Analysis <sup>144</sup>**

<sup>144</sup> <http://www.interactive-minds.com/en/eye-tracker>

The EyeGaze Edge<sup>145</sup> (see Figure 85) consists of an IR camera and illumination unit, which can be mounted below the monitor and connected to the host computer (a binocular version is also available). It determines the relative distance between the pupil center and the corneal reflection to estimate where the person is looking on the screen. It predicts the point-of-regard (PoR) with an average accuracy of a quarter inch or better. The calibration procedure takes 15 seconds.

The system provides different functionalities such as mouse and keyboard emulation, generating speech, typing, internet browsing, environmental control (lights and appliances).

Advantages and drawbacks of EyeGaze Edge tracker are described in Table 13:

EyeGaze Edge advantages	EyeGaze Edge disadvantages
The system is of high accuracy (0.45 degree)	Parasitic IR sources can affect the accuracy
It works with most eyeglasses and contact lens	The system is not mobile (it is attached to a PC)

**Table 13: Advantages and disadvantages of the EyeGaze Edge tracker**

#### 4.3.4.3 MyTobii P10



**Figure 86: MyTobii P10**

MyTobii P10<sup>146</sup> (see Figure 86) is built as an all-in-one computer. It integrates an eye control system, a 15" computer screen and a computer. The system is fully automatic after a calibration procedure of around 10 seconds and that has to be done once. The eye control device is a binocular remote system and can be mounted on a desk, a bed or a wheelchair.

It provides the following functions: mouse and keyboard substitute, internet browsing, games, speech synthesis and environmental control.

Advantages and drawbacks of MyTobii P10 tracker are described in Table 14:

MyTobii P10 advantages	MyTobii P10 disadvantages
The system is of high accuracy (0.5 degree)	The system is not mobile
The system is easy to use and fully automatic	The system is expensive
It is very tolerant to uncontrolled head movements and varying lighting conditions	

**Table 14: Advantages and disadvantages of the MyTobii P10 tracker**

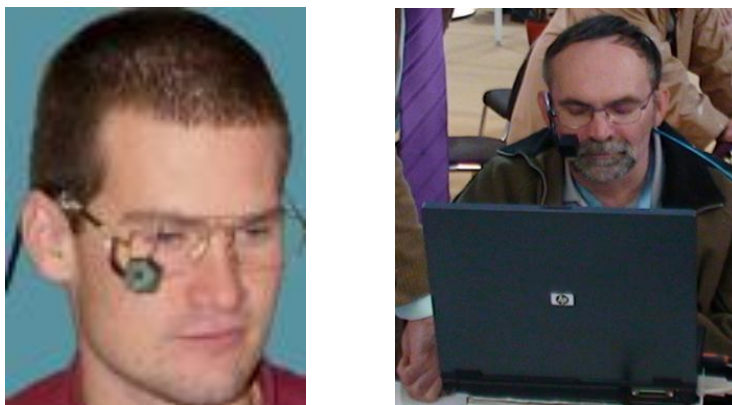
<sup>145</sup> <http://www.eyegaze.com/content/assistive-technology>

<sup>146</sup> [http://www.tobii.com/assistive\\_technology/products/mytobii\\_p10.aspx](http://www.tobii.com/assistive_technology/products/mytobii_p10.aspx)

#### 4.3.4.4 I4Control System

The I4Control System was developed by the Dept. of Cybernetics, Czech Technical University, Prague & Medicton Group Ltd<sup>147</sup> (partially supported by the FP6 COGAIN NoE) and implements the VOG principle for eye position as the eye movement is estimated on the deviation from balanced position of the infrared reflections [108]. The measurement through image processing (histogram thresholding and cubic filter) is done in both, allo-centred (external) and ego-centred frames of reference. It targets the screen explorations, 2D fix surface with relative pointing, in illumination controlled (constant, indoor) environments.

The system is built around a mini black & white camera and infrared diodes attached to a spectacle frame (see Figure 87). It can be connected to the PC via the USB port and emulates the PC mouse.



**Figure 87: I4Control System, Czech Technical University (MEMREC project and EU FP6 IST COGAIN NoE).**

Advantages and drawbacks of I4Control tracker are described in Table 15:

I4Control advantages	I4Control disadvantages
The system does not require a fixed head position; there is no restriction to head movement as long as it is in the camera's field of view) and the user can move "freely" without any restrictions	The system assists an object selection displayed on the surface localized at a predefined depth
Dwelling time reduces the influence of spontaneous eye blink	It suffers from parasitic reflections due to optical glasses and their metallic branches
No specific software for access to the PC is required (as well to the internet and email manager). The system is of low cost (built with off-the-shelf components)	The system is not autonomous (it is attached to a PC)
The training of cursor control is short	

**Table 15: Advantages and disadvantages of the I4Control tracker**

#### 4.3.4.5 OpenEyes

The OpenEyes project by Iowa State University, USA<sup>148</sup>, aims to provide an open-source open-hardware toolkit for low-cost real-time eye tracker building. Their eye tracker operates in infrared spectrum, but it is also possible to modify the configuration and remove the IR LED (see Figure 88). Consequently, two algorithms have been proposed: one operating in

<sup>147</sup> <http://nit.felk.cvut.cz/>

<sup>148</sup> <http://thirtysixthspan.com/openEyes/>

the infrared spectrum and the other operating in visible spectrum (VO). The algorithm applied in infrared spectrum is known under the name *Starburst* [109].



**Figure 88: OpenEyes eyetracker, Iowa State University**

The VOG iris position is detected with a feature-based (ellipse fitting) algorithm optimized via the RANSAC method. The system consists of two cameras mounted on a pair of glasses (after having removed the lens from the glasses) and attached to a PC (laptop/desktop). An eye camera allows tracking of eye-movements and a scene camera placed over the tracked eye captures images of the environment.

Advantages and drawbacks of OpenEyes tracker are described in Table 16:

OpenEyes advantages	OpenEyes disadvantages
The system is very accurate (one degree of visual angle)	This is a mini-obtrusive system
The system is of low cost (built with off-the-shelf components, 300€) and open-source	The system might have difficulties to process (especially outdoor) images in real-time as it uses standard PC platform and iterative algorithms and no illumination invariant image representation
It is wearable (backpack configuration)	The processing is done offline (the videos are recorded thanks to a Digital Video Recorder)

**Table 16: Advantages and disadvantages of the OpenEyes tracker**

#### 4.3.4.6 EyeWriter

The EyeWriter project<sup>149</sup> (Parson School of Design in cooperation with Alhambra Hospital from Los Angeles, USA, see Figure 89) is an ongoing collaborative research effort to empower people who are suffering from ALS (lateral sclerosis). It implements a Hirschberg principle for pupil position detection and tracking from an incoming camera or video image.

<sup>149</sup> <http://www.eyewriter.org/>



**Figure 89: EyeWriter, Alhambra Hospital, USA**

The system uses a calibration sequence to map the tracked eye/pupil coordinates to positions on a computer screen or projection. The calibration part of the software displays a sequence of points on the screen and records the position of the pupil at each point. It is designed so that a person wearing the glasses should focus on each point as it is displayed. When the sequence is finished, the two sets of data are used to interpolate where subsequent eye positions are located in relation to the screen. The system has a dedicated language, *Graffiti Makeup language*, which allows drawing (only).

Advantages and drawbacks of EyeWriter tracker are described in Table 17:

EyeWriter advantages	EyeWriter disadvantages
The system is open-source	It is obtrusive
It allows drawing	It needs to be attached to a PC (not wearable)

**Table 17: Advantages and disadvantages of the EyeWriter tracker**

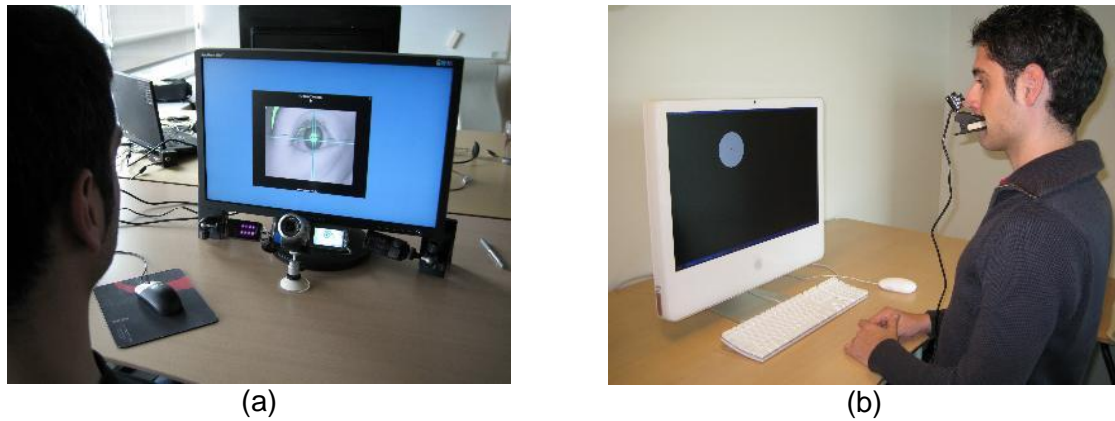
#### 4.3.4.7 ITU Gaze Tracker

The ITU Gaze Tracker by the Information Technology University of Copenhagen<sup>150</sup> (see Figure 90) is an open-source gaze tracking application that aims to provide a low-cost gaze tracking supported by the Communication by Gaze Interaction Association (COGAIN, NoE of the FP6 IST). The eye tracking software is video-based, and any camera equipped with infrared nightvision can be used, such as a video camera or a webcam. ITU Gaze Tracker has two implementations: remote desktop and mouth-mounted eye tracking (to allow free head movements) with no specific hardware modifications.

The system requires a webcam with built-in infrared LEDs and a piece of balsa wood. Although ITU encourages the design of a more convenient mounting than the present balsa bite bar, ITU hopes that a low-cost system like this may stimulate the development of new and more efficient gaze-based applications.

<sup>150</sup> <http://www.gazegroup.org/downloads/23-gazetracker>





**Figure 90: ITU Gaze Tracker (COGAIN NoE, FP6, IST): remote (a) and mouth-mounted (b) setup**

Advantages and drawbacks of ITU Gaze tracker are described in Table 18.

ITU Gaze Tracker advantages	ITU Gaze Tracker disadvantages
The system is of low cost (built with off-the-shelf components) and open-source	If the mouth-mounted configuration is used, the system is cumbersome
Two versions are available: remote and mouth-mounted configuration	It needs to be attached to a PC (not wearable)

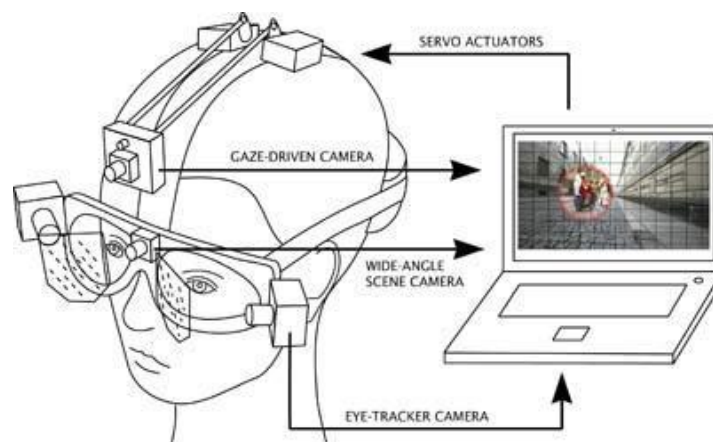
**Table 18: Advantages and disadvantages of the ITU Gaze Tracker**

#### 4.3.4.8 EyeSeeCam

The EyeSeeCam (see Figure 91) by the University of Munich Hospital<sup>151</sup> is a head-mounted gaze-driven camera that is controlled by eye movements. It implements the VOG technique. The video files contain an approximation of the retinal content and are recorded to the hard disk of a portable laptop computer.

EyeSeeCam is a particular eye tracking system in the sense that it combines:

- A head-mounted eye-tracker (for eye movements analysis),
- A camera motion control unit with servo motors (for a head-mounted camera steering to the target of gaze).



**Figure 91: EyeSeeCam**

Advantages and drawbacks of EyeSeeCam tracker are described in Table 19:

<sup>151</sup> <http://eyesecam.com/>

EyeSeeCam advantages	EyeSeeCam disadvantages
The system is of high accuracy, and offer total motor freedom user	The system uses a complex and of high cost technology; its weight is prohibited for a long usage
	The system is not mobile

**Table 19: Advantages and disadvantages of the EyeSeeCam tracker**

#### 4.3.4.9 MobileEye

The MobileEye (see Figure 92) by ASL Applied Science Technology tracks eye movements and points of gazed information during the performance of natural tasks. The system consists in head mounted optics and a color scene camera. A modified Digital Video Recorder (DVR) allows recording images from the eye and scene camera which are then processed on a laptop (not real-time system).



**Figure 92: ASL Mobile Eye**

Advantages and drawbacks of MobileEye tracker are described in Table 20:

MobileEye advantages	MobileEye disadvantages
The system offers unconstrained eye, head and hand movements	The system partly obstructs user field of view
It works under variable lighting conditions	The processing is done offline (the videos are recorded thanks to a Digital Video Recorder)
It is wearable (using a Digital Video Recorder)	

**Table 20: Advantages and disadvantages of the MobileEye tracker**

#### 4.3.4.10 Intelligaze IG-30

Alea technologies<sup>152,153</sup> (Germany) demonstrated the Intelligaze IG-30 gaze-interaction for gaze-controlled computer input.

The Intelligaze IG-30 system hardware uses a modular approach enabling gaze interaction on most of existing computers. The camera and illumination unit are integrated in one system, connected to the host computer via a USB port. There are different possible configurations: eye-or head-tracking, binocular or monocular tracking; all these configurations rely on the use of infrared illumination (Hirschberg principle).

Advantages and drawbacks of VOG eye tracking are described in Table 21:

<sup>152</sup> <http://www.cogain.org/wiki/Gazetalk>

<sup>153</sup> <http://asleyetracking.com/site/>

IntelliGaze IG-30 advantages	IntelliGaze IG-30 disadvantages
The system performs a 3D tracking	It is non-embedded, stationary system

**Table 21: Advantages and disadvantages of the IntelliGaze IG-30 tracker**

### 4.3.5 Summary and comparison of eye trackers

Table 22 provides a summary of some existing academic open-source eye tracker prototypes, while Table 23 lists the most important commercially available trackers all using a PC in order to process the eye movements and determine the gaze of the user.

All academic prototypes are PC based solutions, offering always head-mounted configuration frequently with dedicated software for PC mouse simulation with allo-centered (seldom allo- and ego-centered) referential for eye position estimation. They make larger and larger usage of image and vision processing research results. Almost all, academic and commercially available systems, allow interactions with a plan static surface located in a fixed distance from the end user and constant indoor illumination.

#### OpenSource eye trackers:

Name	University	Configuration	Number of cameras	Illumination	Processing
OpenEyes	Iowa State University	Head-mounted	2	Visible spectrum or infrared	Offline (Digital Video Recorder)
EyeWriter	Parson School of Design Alhambra Hospital	Head-mounted	1 or 2	Infrared	Online (PC)
ITU Gaze Tracker	IT University of Copenhagen	Remote or head-mounted	1	Nightvision and infrared	Online (PC)
EyeSeeCam	University of Munich Hospital	Head-mounted	3 (2009 version) or 4 (2008 version)	Infrared	Online (PC)
I4Control	Czech Technical University, Prague	Head-mounted	1	Visible spectrum	Online (PC)

**Table 22: Open-source eye trackers**

#### Commercial eye trackers:

Name	Company	Configuration	Number of cameras	Illumination
Intelligaze IG-30	Alea Technologies GmbH	Remote or head-mounted	1 or 2	Infrared
Mobile Eye	ASL Applied Science Laboratory	Head-mounted	1 or 2	
EyeMax System	DynaVox Technologies	Remote	1	Infrared
Erica	Eye Response Technologies	Remote	1	Visible spectrum
Eyeteck	EyeTech Digital Systems	Remote	1	Infrared
EyeCan	H.K. EyeCan	Head-mounted	1	Infrared
SeeTech	HumanElektronik GmbH	X	X	Infrared

Name	Company	Configuration	Number of cameras	Illumination
Eyegaze Com System	LC Technologies	Remote	1 or 2	Infrared
Visioboard	Metrovision	Remote	1	Visible spectrum
EcoPoint	PRC	Remote	2	Visible spectrum
Tobii C8, C12, CEye, MyTobii P10, D10	Tobii	Remote	2	Visible spectrum

**Table 23: Commercial eye trackers summary (COGAIN<sup>154</sup>).**

Table 24 briefly summarises characteristics of remote and head mounted eye trackers, and tries to give an outlook to future head mounted wearable autonomous systems for interaction assistance in a 3D environment (ART, attention responsive technology).

Criteria	Present eye tracker technology		Future eye trackers technology
	Remote eye tracker	Head-mounted eye tracker	Head-mounted eye tracker
<b>Final goal of the interaction</b>	independent communication with others via computer screen	Independent communication with others via computer screen	<ol style="list-style-type: none"> <li>1. Communicate</li> <li>2. Control the environment via screen (LCD?)</li> <li>3. Control the environment via ART (attention responsive technology)</li> </ol>
<b>Interaction with</b>	2D image (single static surface, fixed distance to objects)		3D scene (varying distance to objects)
<b>Detected facial features</b>	Face and/or eyes	Eye	Eye
<b>Estimated feature</b>	Eye's line of gaze with respect to a single stationary surface (desktop)	Eye's line of gaze with respect to the head (headband/helmet)	<ol style="list-style-type: none"> <li>1. Eye's line of gaze with respect to fixed markers</li> <li>2. Eye's line of gaze with respect to image interest points attached to gaze selected objects</li> </ol>
<b>Head position/camera</b>	Fixed, small or large movements (depends on the number of IR LEDs used)	Free movements	Free movements
<b>Architecture</b>	1-2 optical camera, infrared LED(s)	2-3 optical cameras, Infrared LED(s)	2 optical cameras, IMU
<b>Object detection</b>	Via markers/RFID	Online processing (PC): computer screen Offline processing (PC): multiple objects	Via computer vision and SLAM (Online processing on embedded system)
<b>Calibration</b>	At least 4 points	At least 4 points	No calibration (Seems impossible for instance given the current eye and gaze models)
<b>Intrusiveness to the user</b>	No	Yes (partially wearable device)	Yes (fully wearable device)
<b>Power autonomy</b>	No	No	Yes

**Table 24: Present and future eye trackers technology main characteristics.**

<sup>154</sup> [http://www.cogain.org/wiki/Eye\\_Trackers](http://www.cogain.org/wiki/Eye_Trackers)

## 4.4 Software support for Computer Vision

A plethora of software tools for Computer Vision exists, as each physical target platform has usually its own tools for system specification and implementation. The most popular generic and simple tools in image processing and computer vision are:

- MATLAB for functional specification and performance prediction
- *Image processing and Computer Vision libraries* which provide functions for image transformations and video capture (usually written in C/C++)
- *Numerical libraries* which provide mathematical tools supporting the above image and vision processing (such as vectors/matrices operations, linear algebra operations etc.)

### 4.4.1 MATLAB related Computer Vision libraries

There are lots of MATLAB related Computer Vision libraries. Here, only the main ones are briefly presented:

*Camera Calibration Toolbox for MATLAB* (proposed by Jean-Yves Bouguet, California Institute of Technology) [173] - This toolbox consists of several MATLAB functions that allow cameras calibration. The toolbox allows an estimation of camera intrinsic and extrinsic parameters. The toolbox supports and provides interfaces for:

- Intrinsic and extrinsic parameters estimation,
- Radial and tangential distortion correction (according to different models),
- Multiple frames captured from different viewpoints,
- Coplanar and non-coplanar targets,
- Stereo calibration and error analysis

*MATLAB Functions for Multiple View Geometry* (proposed by Andrew Zisserman, Oxford University) [174]. This library provides a set of MATLAB functions for computing:

- Geometric relations between multiple views of scenes
- Parameter estimation
- Scene and camera properties estimation/calculation from real world images using state-of-the-art algorithms.
- Single View: Camera model, Calibration, Single View Geometry
- Two Views: Epipolar Geometry, 3D reconstruction, Fundamental Matrix, Computing structure, Plane and homographies
- Three Views: Trifocal Tensor
- More Views: N-Linearities, Multiple view reconstruction, Bundle adjustment, Auto-calibration, Dynamic SfM, Cheirality, Duality

*MATLAB and Octave Functions for Computer Vision and Image Processing* (proposed by Peter Kovesi, Western Australia University) [175]. Kovesi's functions are a set of MATLAB functions (running also under Octave in some cases) that are intended for general image processing and computer vision tasks. Examples of these functions are: Model Fitting and robust estimation, spatial feature detection, feature matching, integral image, image denoising.

## 4.4.2 Image processing and Computer Vision Libraries

### 4.4.2.1 OpenCV

OpenCV means Intel® Open Source Computer Vision Library. It is a library of programming functions that implement some popular algorithms of Image Processing and Computer Vision. From an implementation point of view, OpenCV is a cross-platform middle-to-high level API that consists of a few hundred C/C++ functions. OpenCV is free for both non-commercial and commercial uses.

The most important characteristics of the Open CV for a new software design are:

- Open source computer vision library in C/C++,
- Code optimized for real-time applications,
- OS/hardware/window-manager independent,
- Generic image/video loading, saving, and acquisition,
- Low and high level API,
- Interface to Intel's Integrated Performance Primitives (IPP) for Intel's processor specific optimizations.

The most important Open CV characteristics for vision system specification implementations are:

- Image data structures management (allocation, release, copying, setting, conversion),
- Image and video data format for I/O (file, camera, image/video file I/O),
- Various dynamic data structures (lists, queues, sets, graphs),
- Linear algebra standard operations (Matrix and vector operations, solvers, eigenvalues, SVD),
- Basic image processing (filtering, edge detection, interest point detection, sampling and interpolation, color conversion, morphological operations, histograms, image pyramids),
- Structural analysis (connected components, contour processing, distance transform, moments, matching, Hough transform, polygonal approximation, plane geometric figure fitting, Delaunay triangulation),
- Motion analysis (optical flow, motion segmentation, tracking),
- Camera calibration,
- Detection and tracking of patterns,
- 2D and 3D transforms estimation (homography, stereo matchings),
- Object recognition (eigen-methods, HMM),
- Image labeling (line, conic, polygon, text drawing),
- Basic GUI (display image/video, keyboard and mouse handling, scroll-bars).

*OpenCV* offers the following *modules* for applications development:

- *cv* – Main OpenCV functions,
- *cvaux* – Auxiliary (experimental) OpenCV functions,
- *cxcore* – Data structures and linear algebra support,
- *highgui* – GUI functions,
- *ml* – Machine learning functions.

#### 4.4.2.2 CVD Library

This C++ library is designed to be easy usable and portable and offers functions for efficient saving, loading and displaying of video. It supports OpenGL and X Windows. It is released under the LGPL License.

CVD main features

- Modern C++ design,
- Optimized assembly code,
- Extensive use of exceptions,
- OpenGL wrappers.
- Type-safe easy to use images,
- Flexible loading and saving from a variety of types (native: pnm, bmp ...; external libraries required: jpeg, tiff, png),
- Image grabbing from video sources,
- Colorspace conversions on images and video streams,
- Various image processing tools,
- Camera calibration supported methods: Linear, Cubic, Quintic and Harris.
- LibCVD will compile on any close to standard Unix environment, with a C++ compiler (e.g. gcc version at least of 3.1) and GNU Make. Additionally, libCVD supports both normal and cross compilers. It is known to work on: Linux, Windows, Mac OS.

#### 4.4.3 Numerical Libraries

##### 4.4.3.1 LAPCK/BLAS

Most existing numerical libraries (such as GSL and TooN briefly introduced hereafter) rely on LAPCK/BLAS routines. BLAS (Basic Linear Algebra Subprograms) is a library for vector and matrix operations and LAPACK (Linear Algebra PACKage) is a library of linear algebra routines initially dedicated to general purpose (universal) computer benchmarking. The routines are written in Fortran, but are also available in C (and referred as CBLAS and CLAPACK). LAPACK relies on BLAS and provides the following routines:

- Linear equations
- Linear least squares problems
- Eigenvalue problems
- Singular value decomposition

##### 4.4.3.2 GSL

The GNU Scientific Library (GSL) is a numerical library for C and C++ programmers. It is free software under the GNU General Public License. The library provides a wide range of mathematical routines such as random number generators, special functions and least-squares fitting. There are over 1000 functions in total with an extensive test suite.

Unlike the licenses of proprietary numerical libraries, the license of GSL does not restrict scientific cooperations. It allows sharing programs freely with any designer.

On GSL's website, more information is provided to know which third-party packages is compatible with GSL and how to install them in order to provide extensions to GSL. GSL also provides an interface to BLAS for vector and matrix operations.

#### 4.4.3.3 TooN Library

The Tom's object-oriented Numerics (TooN) library is a set of C++ header files which provide basic numerics facilities such as:

- Vectors, matrices data structure support
- Matrix decompositions
- Function optimization
- Parameterized matrices (e.g. transformations)
- Linear equations

It provides classes for statically- (known at compile time) and dynamically- (known at execution time) sized vectors and matrices and it delegates advanced functions (like SVD or multiplication of large matrices) to LAPACK and BLAS (this means it needs BLAS and LAPACK libraries).

The library makes substantial internal use of templates to achieve run-time speed efficiency whilst retaining a clear programming syntax. Furthermore, TooN library is designed to integrate very well with the CVD computer vision library.

#### 4.4.4 Computer Vision software for embedded platforms

Computer Vision Libraries written in C/C++ are often specific for a targeted vision system. It is usually not easy to port parts of such libraries to a new target platforms because of the high level of nested functions. Most of the available C/C++ libraries are intended for universal processing units (frequently designed as a CPU, central processing unit). At this level of abstraction, an image is considered as a 2D data structure and 2D and 3D mathematical (iterative and recursive) functions are implemented offering reasonable temporal and spatial performances (in the case of weak performance a dedicated massively parallel SIMD machine or an MIMD multiprocessors might be targeted).

For DSPs, very basic but fundamental libraries (BSL – Basic System Library) are provided by manufacturers; for example, Texas Instrument provides *DSPLIB* and *IMGLIB* for its DPS-based embedded systems. These libraries make a very extensive usage of the very fine architectural characteristics of DSPs at almost microprogramming architectural level (VLIW, multi-level data and instructions pipelining, MMU, multiscalar processing, etc.). Consequently, these libraries are not compatible with high level language libraries designed for universal processing units: one additional level of automatic translation is required. Therefore, if an external library should be ported to a DSP, which is often the case for new not yet marketed applications, most external library functions must be rewritten in order to be adapted to the DSP microprogramming architecture. This task is not easy, if the details of hardware fine architecture are not available (what is usually kept secret by DSP manufacturers); moreover, the resulting library usually leads to strongly deteriorated system performances (leading to lose the benefit from DSP usage).



## 4.5 Conclusion

This section has proposed an overview of image and vision technology used for human interaction assistance through different body parts. It seems that a VOG (video-oculography) approach will be the future of eyetrackers as it can work in indoor and outdoor environments (despite of its performance weaknesses: near frontal view angles, fully open eyes, constrained illumination).

Very few of the existing eye trackers have been adapted to motor impaired persons. The most popular adaptations consist of mouse simulation (different implementation of the mouse basic operation) with an object selection on one fixed planar surface (computer screen).

The object detection on surface or screen should be extended in order to offer an augmented interaction with a 3D world to motor impaired persons. An almost natural application for such eyetrackers would be Attention Responsive Technology (ART), useful e.g. for environmental control applications. Therefore, future eye trackers should avoid IR light as they should work in indoor and outdoor environments, and they should follow non-invasive; structure-from-motion approaches, solving the problem of precise and reliable gaze estimation without calibration.

Future eye trackers should be both, ego- and allo-centered, i.e. wearable and head-mounted, allowing hands free interaction with a 3D object and free head movements (motion tracking sensors) and mobility. New theoretical models are necessary for low cost cameras in order to obtain accurate gaze estimation in real scenes (with varying illumination). Furthermore, gaze estimation when a person is moving is an additional challenge in wearable tracking systems which requires more predictive methods still investigated in robotic navigation systems (for example EKF/UKF, visual or visual-inertial SLAM, etc.).

## 5 State of the Art in Hardware

In this section, available computing hardware for the AsTeRICS project will be explored and evaluated. This includes the hardware alternatives for the AsTeRICS embedded platform, the microcontrollers for sensor- and actuator modules, and the computing resources for the Smart Vision Module for computer vision processing.

### 5.1 Hardware for the AsTeRICS embedded platform

In the following, embedded computing solutions will be presented which could serve as core computer for the AsTeRICS personal platform. A detailed list of requirements for this embedded computing hardware will be defined in the document for Deliverable 2.1 (System and Architecture Specification).

#### 5.1.1 IGEPv2



Figure 93: IGEPv2 OMAP3 computing board

<b>Product</b>	IGEPv2 rev. C
<b>Supplier</b>	ISEE Spain <a href="http://www.igep-platform.com/">http://www.igep-platform.com/</a>
<b>Architecture/Chipset</b>	OMAP3530 @ 720 MHz and DSP TMS320DM64+ @ 430 MHz
<b>Operating System(s)</b>	Poky Linux, Ubuntu Linux, Android
<b>RAM</b>	512 MB DDR SDRAM 200 MHz, no expansion
<b>Flash/ROM/Other</b>	512 MB NAND, microSDHC up to 16 (32) GB
<b>Availability</b>	from above supplier
<b>Price (04/2010)</b>	145.0€
<b>Dev. kit Details / Price</b>	SDK under VMware Player for free, no special dev. kit needed
<b>Dimensions / Weight</b>	93 x 65 mm
<b>Power Consumption</b>	2-3.5 W @ 5V
<b>Temperature Range</b>	-40 to +80°C
<b>Onboard Interfaces / Expansion Slots</b>	Ethernet 10/100M BaseT, WiFi 802.11b/g, USB host 1x A 2.0 only, OTG 1x mini-AB 2.0 / 1.1, BlueTooth rev. 2.0, I2C 2x, SPI 3x (1 for TouchScreen) UART 3x, LCD 18/24-bit TFT with TouchScreen and DVI-D output, microSDHC 1x, GPIO ~46x
<b>Additional Onboard Peripherals</b>	RTC, 2x dual-color user LED, camera interface, analog audio Line-in and Line-out, optional 2x 3Mbps ADC, optional RS-485

Table 25: Features of the IGEPv2 OMAP3 computing board

## 5.1.2 ISIS XL



**Figure 94: ISIS XL Atom-based computing board**

<b>Product</b>	ISIS XL
<b>Supplier</b>	Eurotech Maryland, USA <a href="http://www.eurotech-inc.com/single-board-computer-atom-pc104-isis.asp">http://www.eurotech-inc.com/single-board-computer-atom-pc104-isis.asp</a>
<b>Architecture/Chipset</b>	Intel Atom up to 1.3 GHz
<b>Operating System(s)</b>	Windows XP, XP Embedded, Linux
<b>RAM</b>	up to 1 GB DDR2 SDRAM 400/533 MHz
<b>Flash/ROM/Other</b>	2 or 4 GB PATA Flash
<b>Availability</b>	from above supplier
<b>Price (04/2010)</b>	870 EUR
<b>Dev. kit Details / Price</b>	With 8" TouchScreen display with stylus, audio panel, power supply, SDK DVD for 1470 EUR
<b>Dimensions / Weight</b>	PC/104+ form factor 96 mm x 90 mm
<b>Power Consumption</b>	4-4.5 W @ 5V
<b>Temperature Range</b>	-40 to +85°C
<b>Onboard Interfaces / Expansion Slots</b>	Ethernet 10/100M BaseT, WiFi N/A, USB host 8x 2.0/1.1 (pin hdr.) OTG N/A, BlueTooth N/A, I2C 1x, SPI N/A, UART 3x LCD 18/24-bit LVDS and VGA output, TouchScreen only via additional ET TSC1 UART interface, SDIO 1x, GPIO 8x, PC/104+ 32-bit PCI 1x, PC/104 16-bit ISA 1x, PCI Express Mini Card 1x
<b>Additional Onboard Peripherals</b>	Analog/digital audio Mic In, Line-In and Line-out, RTC, GPS receiver, many additional expansion board with e.g. ADC/DAC, UART, RS422/485, CAN, optoisolated GPIO, relays, ... <i>Note: The expansion boards significantly affect the dimensions.</i>

**Table 26: Features of the ISIS XL Atom-based computing board.**

### 5.1.3 Gumstix Overo Fire



Figure 95: Gumstix Overo Computer-On-Module

<b>Product</b>	Gumstix Overo Fire
<b>Supplier</b>	Gumstix, inc. California, USA <a href="http://www.gumstix.com/">http://www.gumstix.com/</a>
<b>Architecture/Chipset</b>	OMAP3530 @ 600 MHz and DSP TMS320DM64+
<b>Operating System(s)</b>	OpenEmbedded Linux
<b>RAM</b>	256 MB DDR SDRAM 200 MHz, no expansion
<b>Flash/ROM/Other</b>	256 MB NAND, microSDHC up to 16 (32) GB
<b>Availability</b>	from above supplier
<b>Price (04/2010)</b>	\$219 - 1 CPU module
<b>Dev. kit Details / Price</b>	Overo EVM Pack \$442
<b>Dimensions / Weight</b>	CPU module 17 x 58 x 4.2 mm, 6 grams
<b>Power Consumption</b>	2-3.5 W @ 5V
<b>Temperature Range</b>	0 to +85°C
<b>Onboard Interfaces / Expansion Slots</b>	Ethernet 10/100M BaseT, WiFi 802.11b/g, USB host 1x A 2.0 only, OTG 1x mini-AB 2.0 / 1.1, BlueTooth rev. 2.0, I2C 2x, SPI 2x, UART, LCD 24-bit TFT with TouchScreen and DVI-D output, microSDHC 1x, GPIO >50x <i>Note: Not all interfaces are available at the same time!</i>
<b>Additional Onboard Peripherals</b>	12-bit camera interface, analog audio Line-in and Line-out, optional 6x 10-bit ADC, 1-wire, 10-bit address/16-bit data bus with CS/WR/RD control signals, 2x IRQ

Table 27: Features of the Gumstix Overo computer-On-Module

### 5.1.4 BugLabs “BUG” platform



Figure 96: BugLabs OMAP3-based BUG platform

<b>Product</b>	BUG 1.0
<b>Supplier</b>	Bug Labs, Inc. NY, USA <a href="http://www.buglabs.net">http://www.buglabs.net</a>
<b>Architecture/Chipset</b>	ARM 11 @ 532 MHz
<b>Operating System(s)</b>	OpenEmbedded Linux + OSGi Framework
<b>RAM</b>	128 MB, no expansion
<b>Flash/ROM/Other</b>	32 MB Flash, microSDHC up to 16 (32) GB
<b>Availability</b>	from above supplier
<b>Price (04/2010)</b>	\$449 (BUGbase WiFi)
<b>Dev. kit Details / Price</b>	\$749 (BUGbundle)
<b>Dimensions / Weight</b>	131 x 65 x 20 mm (BUGbase)
<b>Power Consumption</b>	N/A
<b>Temperature Range</b>	0 to +85°C
<b>Onboard Interfaces / Expansion Slots</b>	Ethernet 10/100M BaseT, WiFi 802.11b/g, USB host 2.0 (via expansion), OTG 1x 2.0 only, BlueTooth 2.0 + EDR, I2C 4x, SPI 4x, UART 4x, LCD serial link, microSDHC 1x, GPIO 4x <i>Note: Not all interfaces are available at the same time!</i>
<b>Additional Onboard Peripherals</b>	Small status LCD, joystick, 4x user button, internal Li-Ion 1100 mAh battery backup, 2x I2S, RTC, serial camera interface

Table 28: Features of the BugLabs OMAP3-based BUG platform

### 5.1.5 Bitsy-G5



Figure 97: Bitsy-G5 MX31-based computing platform

<b>Product</b>	Bitsy G5
<b>Supplier</b>	Eurotech Maryland, USA <a href="http://www.eurotech-inc.com/single-board-computer-imx31-com-bitsy-g5.asp">http://www.eurotech-inc.com/single-board-computer-imx31-com-bitsy-g5.asp</a>
<b>Architecture/Chipset</b>	Freescale i.MX31 @ 532MHz and vector FP co-processor
<b>Operating System(s)</b>	Windows CE 6.0 / Linux
<b>RAM</b>	128 MB DDR SDRAM, no expansion
<b>Flash/ROM/Other</b>	64 MB NOR Flash, SD card / PCMCIA-II expansion
<b>Availability</b>	from above supplier
<b>Price (04/2010)</b>	N/A
<b>Dev. kit Details / Price</b>	N/A
<b>Dimensions / Weight</b>	76 x 127 mm, 71 g
<b>Power Consumption</b>	1W
<b>Temperature Range</b>	-40 to +85°C
<b>Onboard Interfaces / Expansion Slots</b>	Ethernet 10/100M BaseT, WiFi N/A, USB 2.0 host 1, OTG N/A, BlueTooth N/A, I2C 1x, SPI 1x, UART 3x, 16-bit LCD and 4/5-wire analog-resistive touch panel, SD 1x, GPIO 9x
<b>Additional Onboard Peripherals</b>	1x PCMCIA Type II 3V3 or 5V0, optional 802.3af PoE, external interrupt inputs, expansion bus interface, 2x1W stereo output, 1x mic input

Table 29: Features of the Bitsy-G5 MX31-based computing platform

### 5.1.6 VIA EPIA-P710



Figure 98: VIA EPIA P710 computing platform

<b>Product</b>	VIA EPIA-P710
<b>Supplier</b>	VIA Technologies, Inc. Taiwan <a href="http://www.via.com.tw/en/products/embedded/ProductDetail.jsp?productLine=1&amp;id=730&amp;tabs=1">http://www.via.com.tw/en/products/embedded/ProductDetail.jsp?productLine=1&amp;id=730&amp;tabs=1</a>
<b>Architecture/Chipset</b>	VIA C7 @ 1000 MHz
<b>Operating System(s)</b>	Windows XP, Windows CE, Linux
<b>RAM</b>	Up to 2 GB DDR2-533/667
<b>Flash/ROM/Other</b>	1x UDMA133 PATA, 1x SATA
<b>Availability</b>	from above supplier
<b>Price</b>	\$311
<b>Dev. kit Details / Price (04/2010)</b>	Standard PC kits
<b>Dimensions / Weight</b>	100 x 72 mm
<b>Power Consumption</b>	10-15 W
<b>Temperature Range</b>	0 to +50°C
<b>Onboard Interfaces / Expansion Slots</b>	Ethernet 10/100/1000M, WiFi: N/A, SB 2.0 host 3x, OTG N/A, BlueTooth N/A, I2C N/A, SPI 1x (via exp. module), UART 1x (via expansion module) CRT and LVDS panel output, SD via expansion module, GPIO 4x (via expansion module)
<b>Additional Onboard Peripherals</b>	1x PATA, 1x SATA, Line In/Out, Mic In, other interfaces on expansion modules: PCIe x1, PCIe x4, ExpressCard

Table 30: Features of the VIA EPIA P710 computing platform

### 5.1.7 Asus EeePC T101MT Netvertible



Figure 99: Asus EeePC T101MT netvertible

<b>Product</b>	Asus EeePC T101MT Netbook with Tablet Function (Netvertible)
<b>Supplier</b>	Several manufacturers
<b>Architecture/Chipset</b>	Intel Atom N450 CPU 1.6Ghz, Intel GMA350 Graphics Chipset
<b>Operating System(s)</b>	Windows XP/Vista/7 , Linux
<b>RAM</b>	2 GB
<b>Flash/ROM/Other</b>	320GB hard drive (optional 64GB SSD)
<b>Availability</b>	First quarter 2010
<b>Price (04/2010)</b>	450EUR
<b>Akku / Time</b>	Lithium Polymer, 6,5 h
<b>Dimensions / Weight</b>	10", around 1,3 kg
<b>Power Consumption</b>	±10W
<b>Temperature Range</b>	0 to +60°C
<b>Onboard Interfaces / Expansion Slots</b>	Ethernet 10/100M, WiFi 802.11b/g/n, USB 2.0 host 3x, OTG N/A BlueTooth, I2C N/A, SPI N/A, UART N/A, LCD panel with touchscreen (1024 x 600) + multitouch, VGA output, SD/SDIO,GPIO N/A
<b>Additional Onboard Peripherals</b>	Line In/Out, Mic In, Webcam: 0,3 MP

Table 31: Features of the Asus EeePC T101MT netvertible



## 5.1.8 WePad Table PC

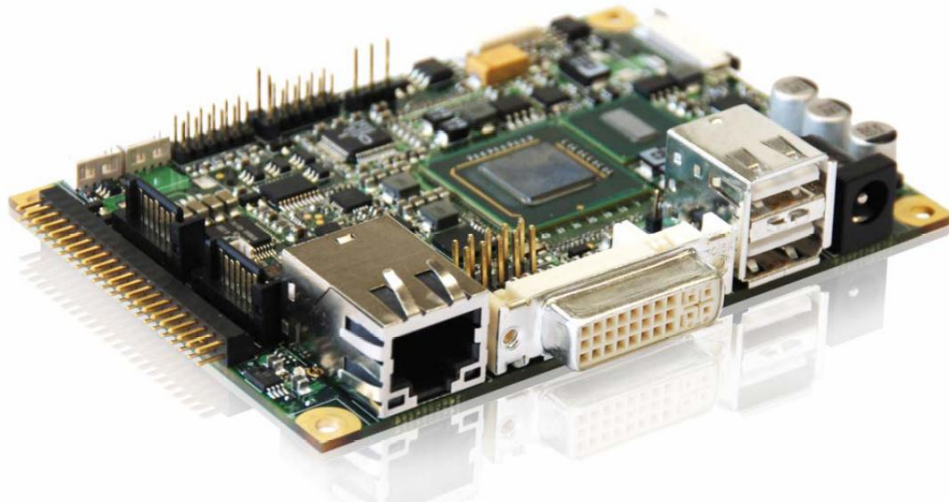


Figure 100: WePad tablet PC

<b>Product</b>	WePad Tablet PC
<b>Supplier</b>	WePad GmbH, Neofonie GmbH, <a href="http://wepad.mobi">http://wepad.mobi</a>
<b>Architecture/Chipset</b>	Intel Atom N450 CPU 1.6Ghz
<b>Operating System(s)</b>	Customised Linux, Android, Windows
<b>RAM</b>	1 GB
<b>Flash/ROM/Other</b>	16 / 32 GB
<b>Availability</b>	2010
<b>Price (04/2010)</b>	449EUR / 569EUR (3g, 32GB)
<b>Akku / Time</b>	Lithium Polymer, 6 h
<b>Dimensions / Weight</b>	11,6", around 0,8 kg
<b>Power Consumption</b>	±10W
<b>Temperature Range</b>	0 to +60°C
<b>Onboard Interfaces / Expansion Slots</b>	WiFi 802.11b/g/n, 3G (optional), GPS (optional), USB 2.0 host 2x, OTG N/A BlueTooth, I2C N/A, SPI N/A, UART N/A, LCD touchscreen (1366 x 768) HDMI output, SD/SDIO, GPIO N/A, Ambient light sensor, Accelerometer
<b>Additional Onboard Peripherals</b>	Line In/Out, Mic In, Webcam: 1,3 MP

Table 32: Features of the WePad tablet PC

### 5.1.9 Kontron pico ITX-SP



**Figure 101: Kontron Atom-based Single Board Computer**

<b>Product</b>	Kontron Pico ITX-SP
<b>Supplier</b>	Kontron Technology AG, Eching/München, Germany, <a href="http://www.kontron.com">http://www.kontron.com</a>
<b>Architecture/Chipset</b>	Intel Atom Z510/Z530 up to 1.6 GHz
<b>Operating System(s)</b>	Windows XP, XP Embedded, Windows 7, Linux
<b>RAM</b>	up to 2 GB DDR2 SDRAM 400/533 MHz
<b>Flash/ROM/Other</b>	2 or 4 GB PATA Flash, SATA HD / SSD
<b>Availability</b>	May 2010
<b>Price (04/2010)</b>	280 EUR
<b>Dev. kit Details / Price</b>	With power supply, cable set, 1 GB ram for 380 EUR
<b>Dimensions / Weight</b>	Pico ITX+ form factor 10,7cm x 7,2 mm
<b>Power Consumption</b>	ca. 5 W @ 5V
<b>Temperature Range</b>	0 to +60°C
<b>Onboard Interfaces / Expansion Slots</b>	Ethernet 10/100/1000M BaseT, WiFi N/A, USB host 6x 2.0/1.1 (pin hdr.), OTG N/A, BlueTooth N/A, I2C 1x, GPIO: 4 in / 4 out, SPI N/A, UART N/A, LCD 18/24-bit LVDS and DVI output, SDIO 1x, GPIO N/A
<b>Additional Onboard Peripherals</b>	Analog/digital audio Mic In, Line-In and Line-out, RTC

**Table 33: Features of the Kontron Atom-based Single-Board-Computer**

## 5.2 Microcontrollers for interfacing sensors and actuators

Microcontrollers provide many features for interfacing digital products with the real world like analog-to-digital converters (ADC), digital-to-analog converters (DAC), general input/output ports (GPIO) and standard busses as I2C, SPI, UART or USB.

Microcontrollers are thus highly interesting to extend the interfacing capabilities of the AsTeRICS Embedded Platform and to connect sensors or actuators to the platform or to the PC. For example the Universal HCI Actuator (which will be developed in course of the project) shall act as mouse, keyboard or joystick when plugged into the USB port of a PC. Some modern 8-bit microcontrollers provide a native USB interface and firmware examples for USB HID devices, which makes them an ideal choice for this task.

### Common types of microcontrollers

Today's microcontroller market is divided into three segments: 8-, 16-, and 32-bit cores. Traditionally, the 8-bit segment has been the largest. The 32-bit segment is the fastest growing segment and may already have surpassed the 8-bit segment in size. The 8- and 16-bit segments are still growing, but not at the same rate (with 8-bit segment still growing faster than 16-bit segment). On the long term, the 16-bit segment may be absorbed by the 8- and 32-bit segments. Today's 32-bit microcontroller devices are built in the latest technologies (when compared to the 8- and 16-bit cores) [34].

A typical 8-bit microcontroller has a performance between 10 and 30 MIPS while 16-bit microcontrollers have between 20 and 40 MIPS. The new 32-bit microcontrollers often have 80 to 100 MIPS and more.

Microcontroller manufacturers are able to take advantage of these latest technologies to allow 32-bit microcontrollers to keep getting smaller, and thus they continue to get cheaper. Because of the price and performance of the 32-bit microcontrollers, many analysts believe that design engineers will leapfrog the 16-bit segment.

On the other hand, a 32-bit microcontroller could be "technical overkill" for many types of applications, where dedicated functionality is needed and the desired task can be performed without a huge application stack or even without an operating system, which makes the functionality available immediately after power-up, bypassing boot times.

Today's 8-bit microcontrollers are highly specialised low-cost products which often feature on-chip self-programming flash memory, various clock- and energy saving modes, debugging support, various peripheral units and communications controllers. Wireless support via ZigBee or Bluetooth can easily be added.

When interfacing a sensor or controlling a simple actuator is desired, 8-bit microcontrollers are still the best choice in many cases, as firmware programming is straight-forward and the part costs are low. The next section will give an overview of interesting product alternatives in the sector of USB capable 8-bit controllers and outlines their main product features.

### 5.2.1 PIC18 series of microcontrollers

The PIC18F1XK50 series [35] of 8-bit microcontrollers feature a USB v2.0 compliant interface which supports both Low Speed and Full Speed operation. The supported transfer modes include control-, interrupt-, isochronous- and bulk transfers with up to 16 endpoints (or up to 8 bidirectional endpoints). Physical connection to a USB host is automatically detected. The key parameters are given in the table below.

Parameter name	Value
Program Memory Type	Flash
Program Memory (KB)	16
CPU Speed (MIPS)	12
RAM Bytes	768
Data EEPROM (bytes)	256
Digital Communication Peripherals	1-A/E/USART, 1-MSSP(SPI/I2C)
Capture/Compare/PWM Peripherals	1 CCP, 1 ECCP
Timers	1 x 8-bit, 3 x 16-bit
ADC	9 ch, 10-bit
Comparators	2
USB (ch, speed, compliance)	1, Full Speed, USB 2.0
Temperature Range (C)	-40 to 125
Operating Voltage Range (V)	1.8 to 5.5
Pin Count	20
XLP	Yes
Cap Touch Channels	4
PWM outputs	Up to 5
Price per unit	€ 2,20

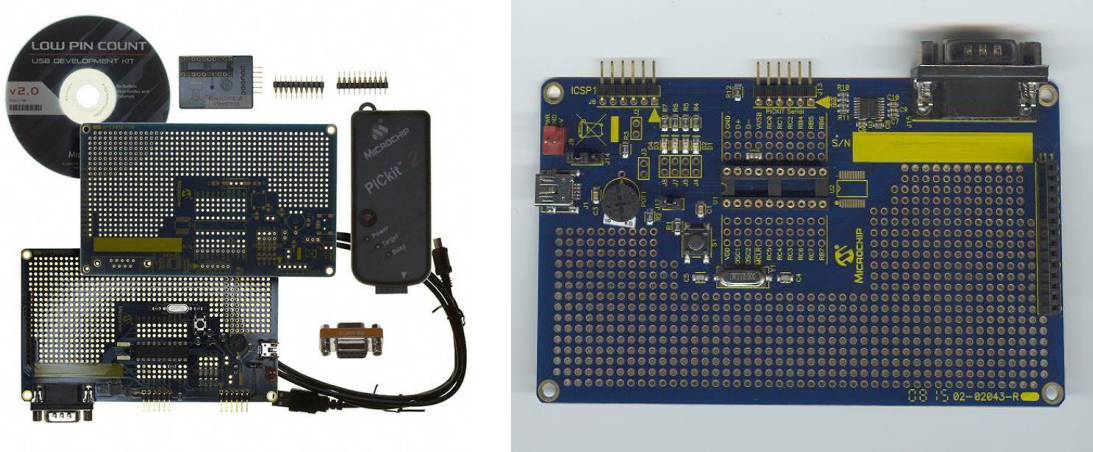
**Table 34: Features of the PIC18F1XK50 microcontrollers [35]**

Additionally to these key parameters, the PIC18F1XK50 series provides the following features:

- Self-programmable under software control
- Support for Interrupt Priority Levels
- 8x8 Single-Cycle Hardware Multiplier
- Can be internally clocked up to 48MHz
- Fully tunable Internal Oscillator - Supplies frequencies from 31KHz to 16MHz - Can be user-adjusted for frequency drift
- Master I2C/SPI Interface - Supports all modes of 3-wire SPI - I2C™Compatible Master and Slave modes with address masking
- Robust GPIO with 25mA sink/source current
- Programmable On-chip Voltage Reference; can be used with Comparators or ADC
- Operating Environment Versatility
- Four power-managed operating modes

The PIC18F1XK50 microcontrollers are available in various packages, including PDIP, SOIC, SSOP and QFN.

For the PIC18F1XK50 microcontrollers, a low cost development kit is available which optionally includes the PICkit2 debugger/programmer (see Figure 102). This development kit also includes software examples for USB compliant HID devices as mouse, keyboard or joystick [36]. The PIC USB development kit costs currently around €32, 00 at digikey.



**Figure 102: Low pin count USB development kit including PICkit2 debugger [36]**

Microchip provides USB software to support USB on 8-bit, 16-bit and 32-bit MCUs. This software is royalty free source code and also includes example projects. The 8-bit PIC18 family supports USB device mode. The 16-bit PIC24F and 32-bit PIC32 products with USB support device mode, as well as embedded host and On-The-Go. All product families support up to (FS-USB) full-speed operation (12 Mbps).

#### **Software Support for the PIC low pin count USB development kit:**

The software support includes USB firmware for the microcontroller as well as a USB device driver for the PC which allows the PC to treat the microcontroller as a USB device. Classes supported include HID, CDC, MSD and generic, with software examples for:

- Audio – USB device (Microphone, Midi)
- USB – Bootloaders
- CDC - Serial Emulator
- HID (digitizers, joystick, keyboard, mouse, custom HID)
- WinUSB - Generic Driver Demo

#### **5.2.2 Atmel AT90USB64/128 microcontrollers with USB support**

The AT90USB family of 8-bit microcontrollers by Atmel [37] supports a wide range of USB applications like keyboards, mice, phone accessories, toys or serial adapters. An on-chip bootloader allows firmware download from a PC, without using a parallel programmer or dedicated hardware. The table below shows features of the AT90USB64 and AT90USB128 microcontrollers, respectively.

Parameter name	Value
Program Memory Type	Flash
Program Memory (KB)	64/128
CPU Speed (MIPS)	16

Parameter name	Value
RAM Bytes	4096 / 8192
Data EEPROM (bytes)	2048 / 4096
Digital Communication Peripherals	1- USART, 1 SPI, 1 I2C
Capture/Compare/PWM Peripherals	2 CCP, 4 PWM
Timers	2 x 8-bit, 2 x 16-bit
ADC	8 ch, 10-bit
Comparators	2
USB (ch, speed, compliance)	1, Full Speed, USB 2.0
Temperature Range (C)	-40 to 85
Operating Voltage Range (V)	1.8 to 5.5
Pin Count	64
PWM outputs	4
Price per unit	€ 6,90

**Table 35: Features of the Atmel AVR90USB64/128 microcontrollers [37]**

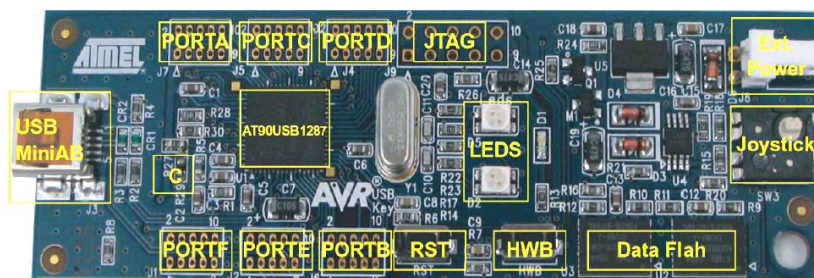
The AT90USBKey (see Figure 103) is a low-cost demonstration board based on the AT90USB1287 microcontroller. It allows the quick evaluation of AVR microcontrollers and the AT90USB family.

#### Software Support for the Atmel90USB family of microcontrollers:

Among others, demo applications for following USB applications are available:

- USB – mouse, keyboard and generic HID demonstrations
- USB – mass storage implementation
- CDC – host demonstration

Designers can either run demonstration programs or their own application. The AT90USBkey includes following additional onboard peripherals: 4+1-ways joystick, 2 Bi-Color LEDs, a temperature sensor and two serial dataflash memories. An external power supply can be used for USB host or stand-alone operation [38]. The price of the AT90USBKEY development board is currently € 25,05 at digikey.



**Figure 103: AT90USBkey development board [38]**

### 5.2.3 Cypress microcontrollers with USB support

Cypress Semiconductors [40] has a long tradition in USB controllers. The Cypress M8 one-time programmable (OTP) microcontroller was one of the first integrated controllers for low speed USB peripherals. The EZ-USB series offers USB high speed capability, IO-options

dedicated to data storage (e.g. ATA) or interfaces for biometric sensors. The encore series eliminates the need for an external crystal or resonator, pull up resistors, wakeup circuitry and 3.3V regulator to reduce the overall system cost. The CY7C64215, which is part of the encore-III series, is based on flexible Programmable System on Chip (PSoC) architecture and is a full featured, full speed (12 Mbps) USB part which features configurable analog, digital, and interconnect circuitry to enable a high level of flexibility.

Parameter name	Value
Program Memory Type	Flash
Program Memory (KB)	16
CPU Speed (MIPS)	24
RAM Bytes	1024
Data EEPROM (bytes)	emulated in flash
Digital Communication Peripherals	1 SPI, 1 I2C, 1 USART
Capture/Compare/PWM Peripherals	Up to 4 CCP, PWMs
Timers	up to 4 8/16 bit
ADC	Up to 14
Comparators	1
USB (ch, speed, compliance)	1, Full Speed, USB 2.0
Temperature Range (C)	-40 – 85
Operating Voltage Range (V)	4 to 5.5
Pin Count	56
PWM outputs	Up to 4
Price per unit	€ 4,41

**Table 36: Features of the Cypress CY7C64215 [40]**

The Digital System of the encore-III series is composed of four blocks, where each block is an 8-bit resource that is used alone or combined with other blocks to form 8, 16, 24, and 32-bit peripherals. The digital blocks are connected to any GPIO through a series of global busses that can route any signal to any pin. The busses also enable signal multiplexing and performing logic operations. This configurability frees designs from the constraints of a fixed peripheral controller [39]. The table above shows features of the CY7C64215 encore-III Programmable System on Chip.

The enCoRe III development kit CY3664 (see Figure 104) supplies the user with an in-circuit emulator (ICE) that works in conjunction with actual silicon to provide an accurate and efficient development system. The PSoC Designer software consists of a graphical user interface, assembler, C compiler, linker and debugger for a highly integrated code development environment. Several example user modules are provided, including a compliant Full-Speed USB module and other peripheral user modules. The kit currently costs €1.022 at digikey.

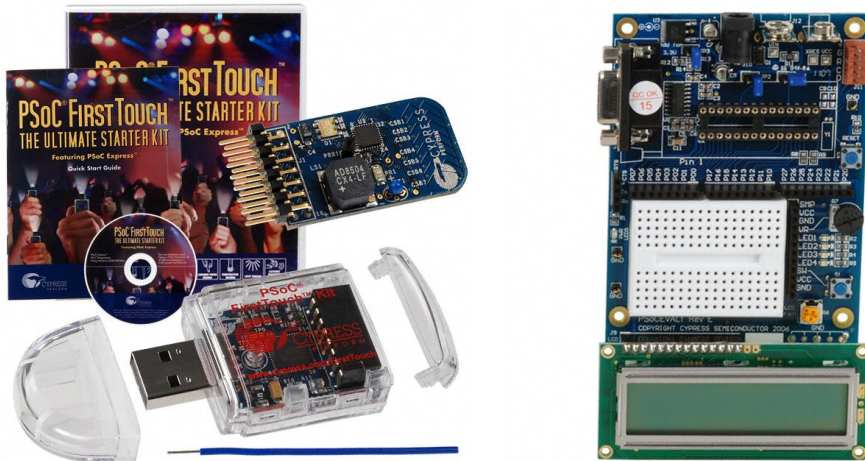


Figure 104: CY3664 encore3 development kits [41]

### Software Support for the Cypress encore-III series:

- Windows Mass Storage Driver
- GPIF designer (device descriptor editor)
- Linux Peripheral Driver
- USB Development tools for Visual Studio (SuiteUSB 3.4)

## 5.3 Wireless Interfaces

Wireless communication interfaces are interesting for the AsTeRICS project, because of various reasons:

- recent mobile device can mostly be accessed via WiFi or Bluetooth
- low-power wireless links like ZigBee are interesting for home automation.
- a remote update of AsTeRICS configurations using a wireless link is desired.

Figure 105 shows the IEEE 802 wireless space. The 2400–2483.5 MHz band is available for license-free operation in most countries.

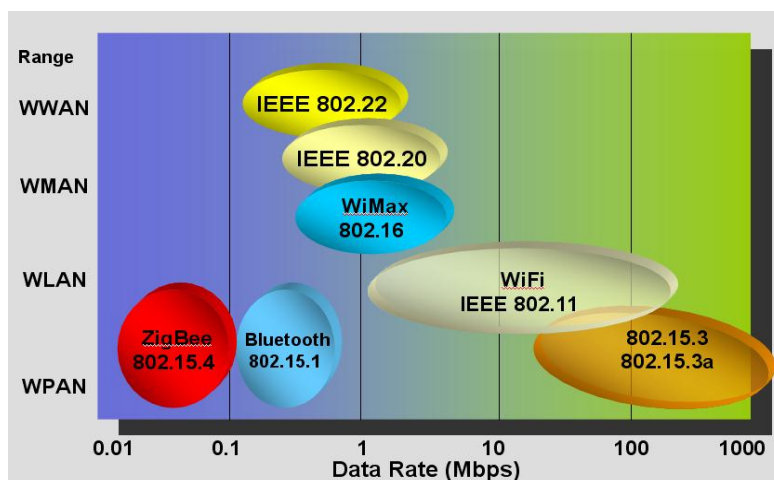


Figure 105: Diagram of the IEEE 802 wireless space



2.4 Ghz radio band advantages	2.4 Ghz radio band advantages
Same solution for all markets without SW/HW alterations	Shorter range than a sub 1 GHz solution (with the same current consumption)
Large bandwidth available, allows many separate channels and high data rates	Many possible interferers are present in the band
100% duty cycle is possible	
More compact antenna solution than below 1 GHz	

**Table 37: Advantages and disadvantages of the 2.4Gh radio band for wireless communication**

In the following chapters, the IEEE 802.15.4 (ZigBee), 802.15.1 (Bluetooth) and 802.11 (WiFi) will be discussed in more detail.

### 5.3.1 ZigBee and 802.15.4

The IEEE 802.15.4 standard is intended to offer the fundamental lower network layers of a type of wireless personal area network (WPAN) which focuses on low-cost, low-speed ubiquitous communication between devices (in contrast with other, more end user-oriented approaches, such as Wi-Fi<sup>155</sup>). The emphasis is on very low cost communication of nearby devices with little to no underlying infrastructure, intending to exploit this to lower power consumption even more.

#### ZigBee

ZigBee is a low-cost, low-power, wireless mesh networking proprietary standard. The low cost allows the technology to be widely deployed in wireless control and monitoring applications, the low power-usage allows longer life with smaller batteries, and the mesh networking provides high reliability and larger range. ZigBee operates in the industrial, scientific and medical (ISM) radio bands; 868 MHz in Europe, 915 MHz in the USA and Australia, and 2.4 GHz in most jurisdictions worldwide. The technology is intended to be simpler and less expensive than other WPANs such as Bluetooth. Because ZigBee can activate (go from sleep to active mode) in 15 ms or less, the latency can be very low and devices can be very responsive — particularly compared to Bluetooth wake-up delays, which are typically around three seconds. Because ZigBees can sleep most of the time, average power consumption can be very low, resulting in long battery life.

ZigBee protocols are intended for use in embedded applications requiring low data rates and low power consumption. ZigBee's current focus is to define a general-purpose, inexpensive, self-organizing mesh network that can be used for industrial control, embedded sensing, medical data collection, smoke and intruder warning, building automation, home automation, etc. The resulting network will use very small amounts of power – individual devices must have a battery life of at least two years to pass ZigBee certification.

#### TIMAC

TIMAC is standardized wireless protocol based on IEEE 802.15.4 suitable for battery-powered and/or mains powered nodes and applications with low data-rate requirements. It can create point-to-point or star network topology.

<sup>155</sup> <http://en.wikipedia.org/wiki/Wi-Fi>

### 5.3.2 BlueTooth

Bluetooth is a standard and a communications protocol primarily designed for low power consumption, with a short range (power-class-dependent: 100m, 10m and 1m, but ranges vary in practice) based on low-cost transceiver microchips in each device. Bluetooth makes it possible for these devices to communicate with each other when they are in range. In most cases the effective range of class 2 devices is extended if they connect to a class 1 transceiver, compared to a pure class 2 network. This is accomplished by the higher sensitivity and transmission power of Class 1 devices.

Bluetooth uses a radio technology called frequency-hopping spread spectrum, which chops up the data being sent and transmits chunks of it on up to 79 frequencies. Bluetooth provides a way to connect and exchange information between devices such as mobile phones, telephones, laptops, personal computers, printers, Global Positioning System (GPS) receivers, digital cameras, and video game consoles through a secure, globally unlicensed Industrial, Scientific and Medical (ISM) 2.4 GHz short-range radio frequency bandwidth. The Bluetooth specifications are developed and licensed by the Bluetooth Special Interest Group (SIG). The Bluetooth SIG consists of companies in the areas of telecommunication, computing, networking, and consumer electronics. Bluetooth exists in many products, such as telephones, modems and headsets. The technology is useful when transferring information between two or more devices that are near each other in low-bandwidth situations. Bluetooth is commonly used to transfer sound data with telephones (i.e., with a Bluetooth headset) or byte data with hand-held computers (transferring files).

### 5.3.3 WiFi

The WiFi Alliance was founded 1999 as WECA (Wireless Ethernet Compatibility Alliance), an organisation which had more than 300 industry members. The main goal of the WECA was to certify products of different vendors, using the IEEE802.11 standard as a basis. The term WiFi has become popular as a synonym for Wave-Lan (WLAN). Especially in the sector of Personal Computers, Netbooks, Personal Digital Assistants (PDAs) and recent smart phones, WiFi has become a very popular means of wireless communication, supporting the TCP/IP stack just as a wired ethernet connection. WiFi features different encryption technologies like WEP, WPA and WPA2, and different connection speeds from 11 Mbps (802.11b) up to 200 Mbps and more (802.11n).

### 5.3.4 Comparison of wireless technologies

The following tables show a comparison between the discussed wireless technologies in terms of power consumption and features:

Current consumption of chipsets for each protocol				
Standard	ZigBee	TIMAC	Wi-Fi	Bluetooth
Chipset	CC2430	CC2430	CX53111	BlueCore2
VDD [V]	3	3	3.3	1.8
TX [mA]	24.7	10	219	57
RX [mA]	27	10	215	47
Bit rate [Mb/s]	0.25	0.03	54	0.72

**Table 38: current consumption of wireless chipsets**

	<b>ZigBee</b>	<b>TIMAC</b>	<b>Wi-Fi</b>	<b>Bluetooth</b>
<b>Range</b>	10-50 meters	10-50 meters	50-100 meters	10 – 50 meters
<b>Networking Topology</b>	Ad-hoc, point to point, star, or mesh	point to point, star	Point to hub	Ad-hoc, very small networks
<b>Operating Frequency</b>	868 MHz (Europe) 900-928 MHz (NA), 2.4 GHz (worldwide)	868 MHz (Europe) 900-928 MHz (NA), 2.4 GHz (worldwide)	2.4 and 5 GHz	2.4 GHz
<b>Data Rate</b>	250 kbps @2.4 GHz, 40 kbps @ 915MHz and 20 kbps @868 MHz	250 kbps @2.4 GHz, 40 kbps @ 915MHz and 20 kbps @868 MHz	54Mbps	1Mb
<b>Complexity (Device and application impact)</b>	Low	Low	High	High
<b>Power Consumption (Battery option and life)</b>	Very low (low power is a design goal)	Very low (low power is a design goal)	High	Medium
<b>Security</b>	AES block cipher, (CTR, counter mode)		RC4 stream cipher, (WEP), AES block cipher	E0 stream cipher
<b>Authentication</b>	CBC-MAC (ext. of CCM)		WPA2 (802.11i)	Shared secret
<b>Typical Applications</b>	Industrial control and monitoring, sensor networks, building automation, home control and automation, toys, games	Sensors, battery powered, low data rate applications	Wireless LAN connectivity, broadband Internet access	Wireless connectivity between devices such as phones, PDA, laptops, headsets
<b>Abbreviations:</b> AES:advanced encryption standard WEP: wired equivalent privacy WPA: Wi-Fi protected access CBC-MAC: cipher block chaining message authentication code CCM: CTR with CBC-MAC CRC: cyclic redundancy				

Table 39: Feature-comparison of different wireless technologies

## 5.4 Summary and Proposal

There are several embedded hardware platforms which could be used for our project and this document cannot cover them all.

For highly demanding computational tasks or for use cases where a big touchscreen is desired, a netbook with tablet option (netvertible) would be a good solution, offering processing power for even complex computer vision algorithms or speech recognition / synthesis and Windows compatibility for an affordable price.

For use cases where low weight and low power consumption are essential, the small OMAP3-based application boards would be good candidates.

A very good consensus which meets the high computational requirements, the system modularity and the low power consumption is offered by the recent Single Board Computers with Pico ITX form factor. The Kontron pico-ITX SP described in the above section comes with LCD interface for the optional display module, 6 USB host adapters for sensor- and actuator modules, a high-performance Atom CPU with 1,6Ghz, 1GB RAM, SATA interface and a state-of-the art chipset which is optimised for low power consumption. The supported operating systems include Windows and Linux derivatives. This board is a good candidate for the embedded hardware for the first AsTeRICS Prototype.

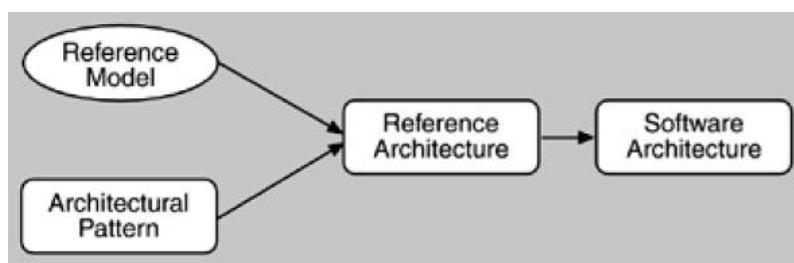
## 6 State of the art in Software Architectures

### 6.1 Introduction

The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them [112]. In this case “externally visible properties” refers to the assumptions that other elements can make of an element (e.g., provided/required services, performance characteristics, shared resource usage, etc.). From this definition we can also imply that software architectures embody information about how participating elements interact between each other and omit information related to elements that do not pertain to their interaction.

Three main concepts of software architecture, as defined in [112], are: a) the architectural patterns (or styles), b) the reference models and c) the reference architectures. Patterns impose useful constraints on the architecture and therefore define a set or family of architectures that satisfy them. Common examples of architectural patterns are Blackboard, Client-server, Peer-to-peer, Distributed computing, Component Based Software Engineering (CBSE) and Service-Oriented-Architecture (SOA).

A reference model represents the decomposition of functionality and data flow between elements into individual parts of the system, while reference architectures map the provided functionality onto software elements. Figure 106 represents how the aforementioned concepts can contribute in specifying comprehensive software architectures.



**Figure 106: The relationships of reference models, architectural patterns, reference architectures, and software architectures [112].**

### 6.2 Architectural Patterns

We begin our review on software architectures with a brief discussion on common software architectural patterns. Special emphasis will be given on the patterns considered most relevant to the project’s objectives.

Patterns are useful tools in developers’ hands since they provide a means to reuse the design knowledge gained by experienced practitioners, so that developers familiar with an adequate set of patterns can apply them immediately to design problems without having to rediscover them [113], [114]. Furthermore, patterns facilitate communication between stakeholders and can be used for tracking the rationale behind specific design decisions since they explicitly evaluate consequences and implementation trade-offs. For the rest of this subsection we discuss a small subset of existing patterns considered relevant to the requirements and vision of AsTeRICS.

## 6.2.1 Layers

Layering is one of the most common techniques in software engineering. It is mainly used for breaking apart complicated software systems in a stack of layers. A classic example comes from the field of networks where the FTP layer is on top of TCP, which is on top of IP. As a basic rule in this pattern higher layers use various services defined in lower layers, but lower layers are unaware of the services provided by the layers above them. Furthermore, layers tend to hide services of lower layers from the layers above.

On the positive side, layers can be developed and understood as independent subsystems, with minimum dependencies between other layers. Therefore, they can be maintain or even replaced by alternative implementations without knowing much about other layers. In addition layered architectures promote reusability since they encapsulate common functionality which can be utilized by layers built upon them. On the other hand, layers do not encapsulate everything that well. For example some GUI attributes that need to be stored in low level layers (such as a database) must be added in every layer in between. Finally, the most difficult part is to decide what layers to have and what are the responsibilities of each layer. Adding extra layers may harm the performance of the system [115].

There are basically two major kinds of modern, layered architectures: two-tier client/server and three-tier—also known as n-tier architectures. Client/Server architectures often operate over a network and most probably on separate hardware. The idea is to partition the work load between service providers (*servers*) and service requesters (*clients*). Servers are often powerful machines that run one or more programs and share their resources with clients. Clients on the other hand, are less powerful, do not share any of their resources and initiate communication sessions with servers which *listen* to incoming requests. Today *Client/Server* is used as a generic term to denote any application architecture that divides processing among two or more processes, often running on two or more machines. Most internet applications, such as email, Telnet, FTP, DNS are based on simple, two-tier architectures [116].

The clear separation of concerns between clients and servers enables easier maintenance operations such as updates, repairs or even server relocation that do not affect clients running on them. In addition, the centralized data storage on servers enhances security control on data access. However, as the complexity in the functional logic of software systems increases, two-tier architectures are deemed insufficient. The original solution was embedding extra functionality into the client layer but the resulting code quickly became hard to work with and maintain. With the proliferation of Java and its application in web frameworks, architectures with an extra layer for presentation became popular. These architectures are called three-tier, and according to Brown *et al* [117] they include the three following principal layers: *presentation*, *logic*, and *data* (see Figure 107).

The presentation layer is mainly responsible for the interaction between the user and the system. This can be realized either through service provision, command line invocations, http requests, etc. The logic or domain tier (also known as business layer) is the actual purpose of the system, the work that the system needs to do for a specific domain. Finally, the data layer is mainly used for storing and retrieving information in data storage means, such as databases. Having a separate layer for data management keeps data neutral and independent from application servers or business logic and improves scalability and performance.

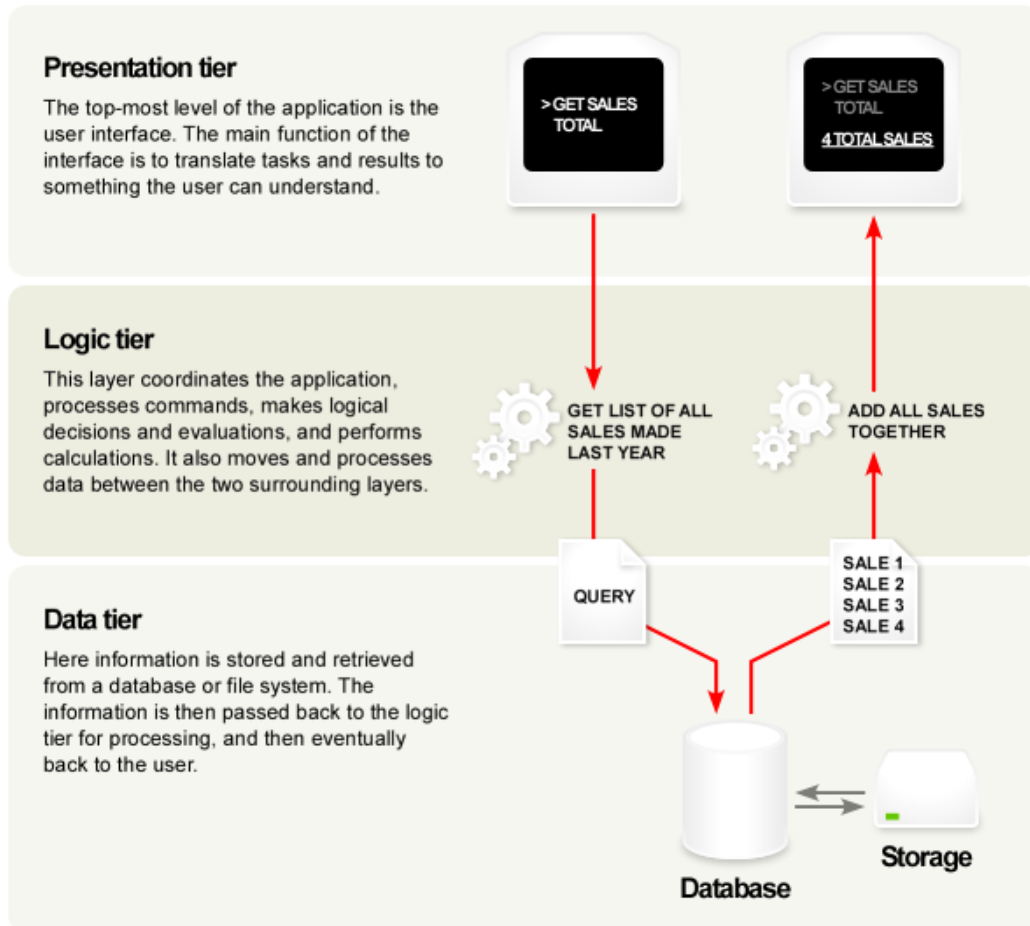


Figure 107: Visual overview of a Three-tiered application<sup>156</sup>.

## 6.2.2 Component-based software

The basic concept in component-based architectures is building software from prefabricated components. The idea was originally introduced by Mclroy in 1968 [118]. However, modern component based development was not in widespread use until years later with technologies like Visual Basic components.

The primary goal of component technology is the independent deployment and assembly of components to create software, partly or fully, from prefabricated parts. According to Szyperski [119], [120], “software components are executable units of independent production, acquisition, and deployment that can be composed into a functioning system. To enable composition, a software component adheres to a particular component model and targets a particular component platform.” The software unit is characterized as a component if it:

- Can be deployed independently - this means that components are well separated from their environments and other components and encapsulate their constituent features.
- Is composable with third-party components - therefore, components need to be self-contained and have clear specifications of what they require and provide (i.e., encapsulation and well defined interfaces).

<sup>156</sup> “Multitier architecture - Wikipedia, the free encyclopedia.” [http://en.wikipedia.org/wiki/Multitier\\_architecture](http://en.wikipedia.org/wiki/Multitier_architecture)

- Has no (externally) observable state - in other words, different implementations of the same components should not be distinguishable from their external behavior.

Software components should be accompanied by contracts that specify the dependencies and interfaces of the component. Moreover, component contracts specify how the component can be deployed, instantiated and how the instance is expected to behave through the advertised interfaces.

Fundamental in CBSE is the notion of frameworks. Component frameworks, as defined in [120], regulate how components plugged into the framework interact with each other and with their environment. They usually implement protocols for connecting participating components and enforce policies on how component instances perform specific tasks. A main objective of component frameworks is to reduce the development, deployment and maintenance costs of software systems. Examples of well-known component models are JavaBeans<sup>157</sup> the .NET framework [121], the CORBA Component Model<sup>158</sup>, the Spring Framework<sup>159</sup>, the Fractal [122] and OSGi<sup>160</sup>. In section 6.4 we discuss the most recent and interesting frameworks, according to the AsTeRICS objectives.

From an architectural point of view, components are the main building blocks of CBSE that specify and formalize architectures. Their roots come from the extensive work made on Architectural Description Languages (ADLs). In practice, most architectural descriptions are informal documents, centered on box-and-line diagrams, with additional text explaining the diagrams. Visual conventions are not standardized although the majority of designs tend to use the ubiquitous box-and-line diagrams. As a result, the architecture cannot be analyzed for consistency or completeness cannot be traced to an implementation, its non-functional properties cannot be enforced as it evolves and there is no tool support to assist software architects in their tasks. ADLs were born to address this kind of problems and have provided a conceptual framework where ideas are expressed in a precise notation and opening the possibility for validation and simulation. ADLs are very well suited for the task of defining component-based architectures.

The common terminology provided by ADLs and present component-based architectures include:

- Components: The building blocks of the description language. They correspond to the idea of a black box in a visual tool. ADLs like ACME [132] define components as the primary elements of computational and data stores of a system.
- Ports: In most ADLs, components are described to have one or more interaction points with the environment. Such interaction points are called ports or interfaces. In ADL's terminology, a port defines both the required services from the environment (required interfaces) and the provided services to the environment (provided interfaces).
- Connectors: Connectors abstract relationships between components. They are the "glue" that connects the black boxes. A connector can be as simple as a pipe or as

---

<sup>157</sup> "Java SE Desktop Technologies - Java Beans."  
<http://java.sun.com/javase/technologies/desktop/javabeans/index.jsp>

<sup>158</sup> "Object Management Group." <http://www.omg.org/>

<sup>159</sup> "SpringSource.org." <http://www.springsource.org>

<sup>160</sup> "OSGi Alliance | About / OSGi Technology." <http://www.osgi.org/Main/HomePage>

complex as a database connection or an ORB in a CORBA environment. Connectors may have interfaces representing the roles of the participants in the interaction.

- **Systems:** A collection of components related by connectors. A key feature is that a system may be hierarchical, that is, both components and connectors may have an internal structure described by more primitive components and connectors.
- **Non-Functional properties:** Annotations to components and connectors describing further semantics.
- **Constraints:** Represent architectural invariants that must hold as the design evolves over time. Constraints play a key role in dynamic architectures - where the components are replaced at run time – in the planning process for component selection, helping to discard invalid combinations.
- **Architectural styles:** An architectural style defines the vocabulary of design elements and the composition rules. Examples of styles are pipe and filter architectures, layered architectures, Chiron-2 (C2), and Weaves architectures.

### 6.2.3 Service-Oriented Architectures – SOA

When studying the SOA domain, we observe that there is no standard, universally accepted definition of Service-Oriented Architectures. According to IBM [124], service-oriented architecture presents an approach for building distributed systems that deliver application functionality as services to either end-user applications or other services. It comprises of elements that can be categorized into *functional* and *quality of service*. The basic functional elements are:

- **Transport:** the mechanism for moving service requests from the consumer to the provider and service responses from the provider to the consumer.
- **Service Communication Protocol:** an agreement between the provider and the consumer on what is provided and the expected quality of service
- **Service Description:** an agreed schema describing what the service is and what is needed for successful invocation of the service.
- **Service:** describes what the service actually provides.
- **Business Process:** a collection of services, invoked in a particular sequence with a particular set of rules, to meet a business requirement.
- **Service Registry:** a repository of services and their descriptions where providers register their services and consumers discover and retrieve available services.

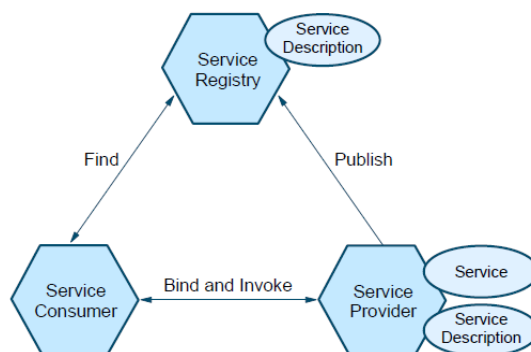
Main quality of service elements are:

- **Policy:** a set of conditions and rules under which the provider makes available a service to consumers.
- **Security:** a set of rules related to security control of service consumers invoking services.
- **Transaction:** the set of attributes applied to a group of services in order to deliver a consistent result.
- **Management:** the set of attributes related to management of services

The standard way that basic elements in service oriented architecture collaborate is as follows: The service provider publishes a service with its description on the service repository. Interested consumers search in the repository for available services that satisfy their requirements. If such a service exists the registry provides the consumer with the



interface contract and the endpoint address for the service. This interaction cycle is depicted in Figure 108.



**Figure 108: Collaborations in a SoA [124]**

Services in SOA are usually characterized as self-contained, modular, loosely coupled and location-transparent. They also support abstraction, reusability, composition and interoperability [124], [125].

SOA is considered by many as an evolution of component-based architectures. Allen and Frost [126] first related the notion of Services with components by defining a component as “an executable unit of code that provides physical black-box encapsulation of related services. Its service can only be accessed through a consistent, published interface that includes an interaction standard. A component must be capable of being connected to other components (through a communications interface) to a larger group”. In component-based software engineering, applications are assembled from components that can be used without any knowledge of either their implementation or their underlying platform. SOA goes a step further by introducing an abstract business model defining the concepts of functionality as a product or an enterprise resource, service provider, service consumer, and service contract. While the owner of a component-based application is responsible for the instantiation of components, the service provider is responsible for the creation and the management of services.

Although SOA was initially proposed to organize business software, service-orientation provides facilities that are applicable beyond that scope. For example, support has been developed for interface type and semantics descriptions, QoS descriptions, service discovery protocols, and binding factories. Nowadays, the SOA concepts are more and more exploited in a large set of producer/consumer systems.

## 6.3 Middleware

Many modern software systems define middleware-based architectures. Formally, middleware is defined as “reusable software that resides between the applications and the underlying operating systems, network protocol stacks, and hardware” [127]. Traditionally, the role of middleware was to hide resource distribution and heterogeneity from the business logic of applications. However, middleware technology has proved to be quite versatile, thus extending its role to a more general domain: “bridging the gap between application programs and the lower-level hardware and software infrastructure to coordinate how parts of applications are connected and how they interoperate” [128].

Schmidt decomposes middleware into four distinct layers according to the functionality they are designed to offer [129]:

- *Host-infrastructure middleware* encapsulates and enhances native operating system operations. The main task of this layer is eliminating accidental incompatibilities of individual operating systems. A good example is the Sun Java Virtual Machine (JVM). It provides a platform-independent way of executing code by abstracting the differences between various operating systems and CPU architectures.
- *Distribution middleware* is commonly referred as today’s middleware. Its main task is to provide higher-level distributed programming models with reusable mechanisms that automate and extend the native operating system operations encapsulated by the host infrastructure middleware. The main advantage of the distribution middleware is that it enables developers to program distributed applications much like standalone applications, simplifying the development process. Some well known implementations on this layer are CORBA, RMI, and DCOM.
- *Common middleware services* augment the distribution middleware by defining higher-level domain independent components that allow application developers to concentrate on programming application logic, without having to deal with distribution-specific implementation issues. Developers can reuse common middleware services for performing recurring distribution tasks such as, event notification, logging, persistence, real-time scheduling, fault tolerance, etc. Common examples of this layer are CORBA services, Microsoft Web Services and Enterprise Java Beans (EJB).
- Domain-specific middleware services are tailored to the requirements of specific domains, such as avionics, telecom, e-commerce, health care, etc. Examples in this category are Siemens MED Syngo, a common software platform for distributed electronic medical systems and Boeing Bold Stroke, a software platform for Boeing avionics mission computing systems.

## 6.4 Software Frameworks

In addition to patterns, *software frameworks* are usually used to build specific programs more quickly since they encapsulate functionality common to a class of applications. While patterns enable reuse of *abstract* design and architecture knowledge, it is through the use of software frameworks that software developers can implement common design patterns and decide on common implementation roles [130]. Furthermore, component frameworks facilitate the reuse of actual, general-purpose code which can be easily configured and deployed.

Frameworks gained much attention with the spread of Object Oriented languages [113], where software frameworks are understood as a set of cooperating classes, some of which may be abstract. The latter make up a reusable design for a specific class of software. In general we can define software frameworks as a set of code or libraries which provide functionality common to a specific domain or concept. Frameworks encapsulate commonly used functionality and facilitate code reuse.

### 6.4.1 Enterprise JavaBeans

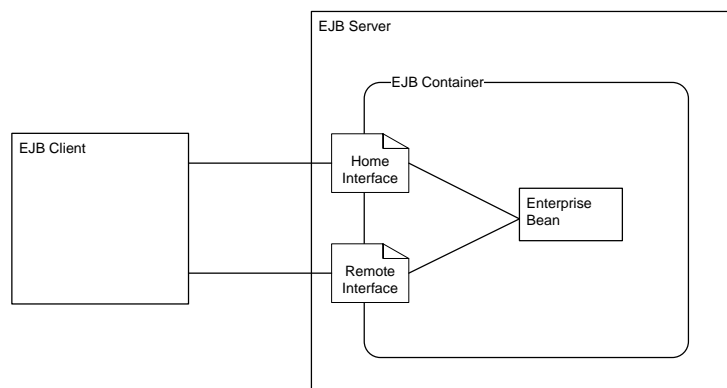
The Enterprise JavaBeans (EJB) specification was originally introduced by IBM in 1997 and later adopted by Sun Microsystems in 1999. EJB is a component architecture for the development and deployment of component-based, distributed business applications. It is what Sun proposed as a solution to the portability and complexity problems of CORBA. EJB introduced a much simpler programming model, allowing developers to create portable distributed components called enterprise beans. Sun claims that “applications written using the EJB architecture are scalable, transactional, and multi-user secure. These applications may be written once and then deployed on any server platform that supports the Enterprise JavaBeans specification”. Therefore, EJB provides a model for building server-side components that implement specific business processes. These components can be then assembled to a complete application that satisfies the requirements of the targeted business. The individual components are distributed and may reside on different servers [131].

The main building block of EJB is the enterprise bean that can only run and function inside a special environment called EJB container. The latter manages every aspect of the bean at run time such as lifecycle control, remote access to the bean, security, concurrency, etc. In this way, the programmer of a java bean can focus on the business logic while the java bean container takes care of generic, cross-cutting aspects, such as security, persistence and interaction with other enterprise beans. An enterprise bean can interact with its container using any of the following mechanisms:

- **Callback Methods:** with this approach every bean must implement a subtype of the *EnterpriseBean* interface which defines a set of callback methods. A varying callback method will be invoked from the container to alert the bean for a change in its lifecycle. Events that might trigger a callback method invocation are: end of a transaction, bean removed from memory, etc.
- **EJBContext:** an enterprise bean is allowed to obtain an EJBContext object which is a direct reference to its container. The EJBContext interface provides methods to the bean for accessing information about its environment (such as the status of a transaction or a client's id), or obtain remote reference to itself.
- **Java Naming and Directory Interface (JNDI):** JNDI is a Java API that provides naming and directory functionality to applications written in the Java programming language. Using JNDI, applications based on Java technology can store and retrieve named Java objects of any type. Every bean by default has access to a special naming system called the Environment Naming Context (ENC). JNDI ENC allows a bean to access its own properties and resources like JDBC connections to other enterprise beans.

Each enterprise bean contains at least three classes: the *home* and *remote interface* implementation classes and the bean's implementation class. The home and remote interfaces enable communications between a client and an enterprise bean. The home

interface performs life cycle operations for *creating*, *finding*, and *removing* the enterprise bean. The remote interface duplicates all the business methods that are to be available to a client. This is needed, because the client cannot access the bean directly.



**Figure 109: Basic EJB Architecture**

There are two main types of enterprise beans: the *Session Beans* and the *Message Driven Beans (MDBs)*. Session beans can be any of *Stateful*, *Stateless* or *Singleton*. Stateful session beans keep track of the clients they are dealing with in each session and only one client at a time can access the bean. The state of the stateful session beans might be persisted automatically by the container to free up memory if client interaction was inactive for some time. Stateless session beans are collection of related services, each represented by a method. Access to a single bean instance is still limited to one client per session, which makes stateless session beans thread-safe. Stateless beans maintain no state from one method invocation to the next. Finally, singleton session beans are suitable for situations where a single enterprise bean is shared across and concurrently accessed by more than one client. They offer similar functionality to stateless session beans but there is only one bean per application. Singleton beans are instantiated once per application, and exist for the lifetime of the application. Finally, they maintain their state between method invocations but are not expected to maintain their state across server crashes or downtimes.

Message driven beans allow applications to process messages asynchronously. They often handle operations that do not require an immediate support (e.g., subscription to a mailing list). In this case, the clients do not access the message beans through the home or remote interface but through the Java Message Service (JMS). Therefore, MDBs instead of providing a client view (through interfaces) simply listens for incoming messages and processes them automatically.

The main advantages of EJB are portability and the simplicity of the EJB programming model. The former comes from the fact that beans developed for one container can be also deployed to another, if it offers more features or better performance. The latter comes from the fact that the container takes care of most of the common and complex tasks such as security, transactions, concurrency, resource management and persistence. Therefore, developers are free to focus on the business logic of their beans. However, as argued in [132] there is a lack of appropriate mechanisms for easily reconfiguring (or extending with new services) EJB containers. This limitation implies some important drawbacks for dynamic software systems. For example it prevents introducing extra functional aspects for components and does not allow application programmers and developers to make important trade-offs such as degree of configurability versus performance and space consumption.

Therefore, these limitations can make the use of these frameworks and languages difficult in different environments, including embedded systems.

## 6.4.2 OSGi

The Open Service Gateway initiative (OSGi) Alliance, proposed in 1998 the OSGi framework and programming model in order to address limitations of the standard modular system employed in Java. In particular, it extends the information hiding principles (already available at the class level) to the level of packages and JAR files. The introduced dynamic modular system enables monitoring and modifying Java applications' compositions at run-time. Overall, OSGi provides an implementation framework for developing component-based and service-oriented applications and a run-time infrastructure for controlling the life cycle of the deployed software models, or bundles as they are called in OSGi terminology [133].

OSGi does not replace the Java class and resource packaging standard but rather adds simple metadata to a JAR, to transform it into a bundle. This allows the creation of "real" modules that can interact at runtime with the aid of the OSGi framework. These metadata are placed inside the JAR file in a special file referred to as "MANIFEST.MF" that consists of:

- The bundle's name that defines its unique identity and optionally a symbolic name.
- The version of the bundle.
- The list of packages imported and exported by the bundle.
- Information about the minimum JRE runtime version that a bundle requires to run (an optional property).
- Additional information such as vendor of the bundle, copyright statement, custom properties, etc.

Because bundles are JAR files enriched with metadata, the developer may choose to use them either inside or outside the OSGi runtime. In case the developer uses the OSGi system it benefits from a very specific and well-defined set of rules that allow defining explicit dependencies amongst bundles and controlling how classes are shared across bundles, using the mechanism of package imports and exports.

Consequently, the dependency relationship defined is not a hierarchical one, as in the case of Java, but a graph of service providers and consumers is formed. This describes that a bundle might require the service (i.e. functionality) provided by another bundle but that same bundle might provide its service to another bundle. For instance, bundle B illustrated in Figure 110 implements and exports a package named "org.asterics.gui" by declaring it in its metadata file. Hence, bundle A might choose to import package "org.asterics.gui" by declaring it in the imports section of its metadata file.

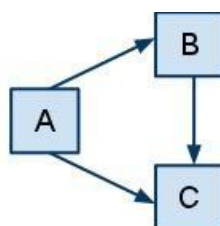
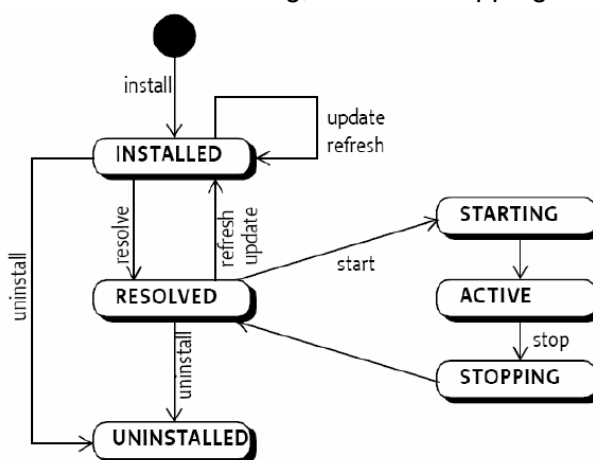


Figure 110: OSGi Class Loader Graph

The OSGi System is responsible to perform the complex resolution process, by finding and matching the dependencies through simple metadata declaration statements. The resolution process is more efficient than Java class loading (i.e. where one global classpath and class loader exists), since the OSGi runtime provides class loaders for each bundle. Therefore, each class loader is aware of the classes and resources included in its JAR. Class loading requests are delegated as a result from one class loader to another in accordance to the dependencies described, avoiding searching through a global and long list of classes; defined as the Java classpath.

Figure 111 shows the different states which the OSGi framework defines for the bundles. Before a bundle is started, all dependencies, such as export and import of packages and required bundles, have to be resolved. While the bundle is in the installed or resolved states, the bundle can be updated with new versions, or uninstalled. Services can be registered and unregistered when the bundle is in the starting, active or stopping states.



**Figure 111: The OSGi Bundle lifecycle**

As it is stated in [134] "OSGi is not just the module system for Java. It is rather the dynamic module system for Java". At its early stages the OSGi System was rather static since the bundle that implemented the service also registered it. The intention was to design a system that offers a mechanism that is implicitly dynamic, like OSGi services (i.e. bundles) but loaded "on demand", like Eclipse extensions [135]. This was accomplished via the Declarative Services (DS) framework [135], [136] included with core OSGi since Release 4. As before, each service is implemented as a bundle but the Service Component Runtime (SCR) of DS is now responsible to register the service on its behalf. The SCR was designed to obtain the necessary information from an XML file called the Service Descriptor in order to register the service. This XML file is also packaged inside the JAR file of the bundle and a pointer to the XML file is defined in the metadata file as a property.

The Declarative Services framework was designed to refine the OSGi services model rather than to replace it [135]. In particular, DS support lazy registration via the SCR component, which allows registering a delayed service in the service registry by creating a proxy object that acts as a placeholder for the service. Although the service is available, the SCR component only requests to load the implementation when the consumer asks for the service; i.e. OSGi with load "on demand" mechanism. Finally, another interesting and powerful feature of DS is the ease provided for composing services and maintaining control of the composition by providing three dimensions for dependencies: (1) Optional vs. Mandatory, (2) Unary vs. Multiple and (3) Static vs. Dynamic. For instance, a dependency

across bundles is dynamic if a bundle B that depends on a terminated instance of bundle A can be substituted with a new instance of A.

Today several OSGi framework implementations exist. Most of them have been developed independently and many of them are open source. We briefly discuss the four main open source implementation of OSGi below:

- **Equinox:** the most widely used OSGi implementation since it is deployed in the core runtime of Eclipse and in other packaged products, such as IBM's WebSphere Application Server. Equinox offers an implementation of the OSGi R4 core framework specification and it is licensed under the Eclipse Public License (EPL).
- **Knopflerfish:** another popular and mature open source implementation of both OSGi releases (3 and 4). Knopflerfish is developed and maintained by Makewave AB and is available under a BSD style license. Makewave also offers a commercial implementation named Knopflerfish Pro.
- **Apache Felix:** the most lightweight implementation of the OSGi Release 4 (it is the smallest size software at Apache Software Foundation). It is a community implementation and licensed under the Apache License version 2.0.
- **Concierge:** an optimized OSGi R3 framework implementation particularly suited to resource-constrained platforms such as mobile and embedded devices. Concierge claims significantly better performance in resource-constrained environments. It is licensed under a BSD-style license.

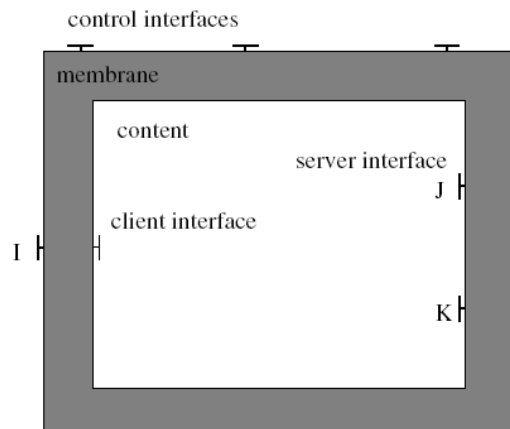
### 6.4.3 FRACTAL

FRACTAL is a component-based, reflective framework that enables introspecting the internal structure and execution state of its components through well defined interfaces. FRACTAL can be used for implementing, deploying and reconfiguring various software systems (mainly operating systems and middleware, but also graphical user interfaces). Implementations exist in various programming languages. Besides the reference implementation in Java, called Julia, there are implementations in C++ (Plasma), SmallTalk (FracTalk), .NET (FractNet) and other languages.

As a component-based framework, components in FRACTAL are the main building blocks and are characterized by the main features of components in Component Based Software Engineering, as defined in [137]. FRACTAL components can be formed as a composition of other sub-components, their resources and state can be shared with other components, they have introspection capabilities and can be reconfigured easily. The transparency degree of the components is configurable according to the requirements of each application and can vary from full-fledged introspection and intercession capabilities to black boxes. At runtime components are encapsulated, have a distinct identity and support one or more interfaces. *Server* interfaces correspond to access points for incoming operations and *Client* interfaces to access points for outgoing operations. Figure 112 represents the basic structure of a FRACTAL component. Main parts of it are the *membrane*, which enables introspection and reconfiguration of the component's internal properties and the *content* which is a composition of a finite set of sub-components.

Inter-component communication between FRACTAL components is achieved through the binding mechanism. Bindings are normal FRACTAL components that mediate communication between other components. We distinguish between Primitive and

Composite bindings. The former implement connections between client and server interfaces and the latter form a communication path between an arbitrary numbers of component interfaces.



**Figure 112: An example of a FRACTAL component [122]**

The most straightforward way to initialize and start a FRACTAL component is by creating a bootstrap component that manages the life cycle of the application components. In other words, a FRACTAL application typically contains a 'root' component whose single job is to manage the life cycle of the components of the application. The FRACTAL API provides a method to request such a bootstrap component. In order to facilitate life-cycle support for its components, FRACTAL also provides a life-cycle controller interface. Through this controller interface, operations can be addressed to start and to stop the component as well as to retrieve its internal state.

#### 6.4.4 Microsoft .NET Framework

Microsoft provides the .NET framework that supports building and running windows applications and XML Web services. The main objectives of the .NET framework, as described by Microsoft<sup>161</sup> are:

- To provide a consistent object-oriented programming environment that allows storing and executing source code either locally or remotely.
- To provide a code execution environment that minimizes deployment and versioning conflicts, enhances security and eliminates performance problems of scripted or interpreted environments.
- To ensure that code based on the .NET Framework can integrate with any other code.

Programmes written for the .NET framework are executed in a runtime environment known as Common Language Runtime (CLR). CLR includes a virtual machine and provides common services such as resource management, exception handling and security aspects. The .NET platform is language independent and supports several implementation languages. Some of them are C#, J#, C++/CLI and Visual Basic. To enable different programming language integration, Microsoft defined the Common Type System (CTS) specification and the Common Language Infrastructure (CLI) specification. The former defines all the possible

<sup>161</sup> Overview of the .NET Framework. [http://msdn.microsoft.com/en-us/library/zw4w595w\(VS.71\).aspx](http://msdn.microsoft.com/en-us/library/zw4w595w(VS.71).aspx) last accessed 15 Apr. 2010



data types and programming constructs supported by the CLR and how they may or may not interact with each other conforming to the CLI specification. The latter describes the executable code and runtime environment that form the core of framework and the free and open source implementations Mono and Portable.NET. The specification defines an environment that allows multiple high-level languages to be used on different computer platforms without being rewritten for specific architectures.

Special compilers that support the .NET platform translate programs to a platform-independent intermediate language known as Common Intermediate Language (CIL). Once the program or library is loaded by the CLR, the CIL is compiled to native code (just-in-time compilation, JIT) and executed by the CLR. Since the language compilers produce platform-independent code in CIL it can be executed on any platform for which a compatible CLR exists.

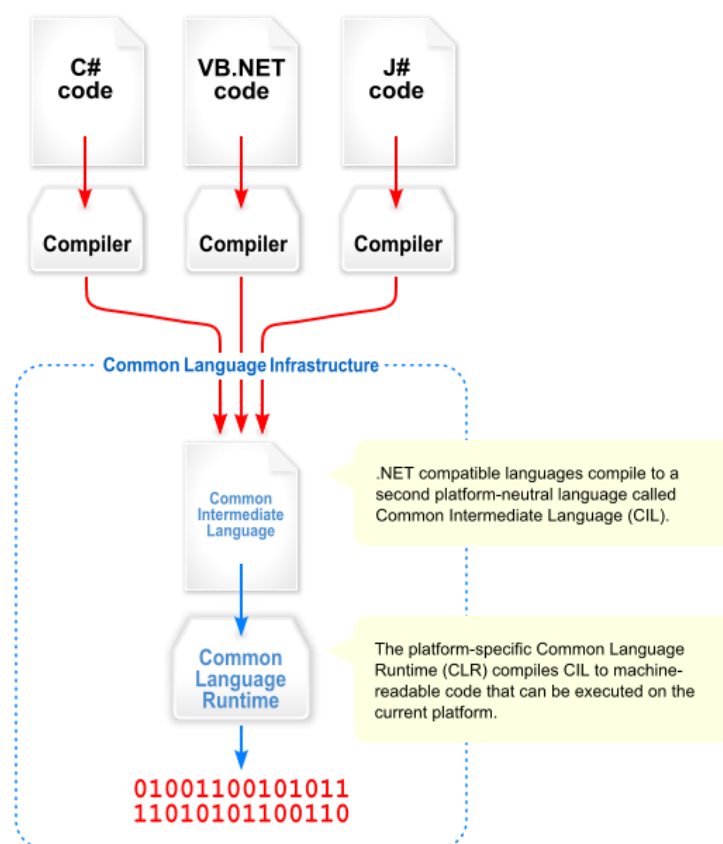


Figure 113: Overview of the Common Language Infrastructure<sup>161</sup>

The main building blocks in .NET are called *Assemblies*. They are self-contained modules compiled to CIL code with one or multiple classes. .NET distinguishes two major assembly types: *executables*, which represent the program, and *libraries*. Executables bear .exe extensions, while libraries end with .dll. This clearly identifies Microsoft Windows as the root of .NET. Assemblies consists of one or more files and one of them, called Manifest, holds metadata related to the Assembly. CLR checks the manifest file to ensure that the correct method is called. Optionally, an assembly may be assigned a so-called Strong Name (SN). It consists of the assembly's filename, its version, and a cryptographic signature. The SN uniquely identifies an assembly which may then be installed to a library pool, the so-called Global Assembly Cache (GAC). The GAC holds and manages different versions of shared assemblies.

The .NET component model is based on the aforementioned assembly approach. A component comprises an interface which may consist of properties (simplified getter and setter methods), methods, and events. Interfaces are directly defined in the particular languages and transformed to CIL by the compilers. This way, no dedicated description language for interfaces is required. A priori unknown components are accessed using reflection. Components are in-process modules. They are loaded either by the CLR or by the application itself and are executed in the calling process' memory space. Services in the .NET framework represent the traditional service approach. They are out-of process entities which may be implemented as standalone processes or web services. Services are not restricted to the web service approach, but might be implemented as executables.

Finally, the .NET framework includes a complete class library with pre-coded solutions for common programming requirements. The Base Class Library, as it is called, provides a comprehensive, object-oriented collection of reusable components that can be used for aiding the application development. Offered reusable types are used for data accessing, user interfacing, database connectivity, cryptography, web application development, etc.

## 6.5 Serialisation

Serialisation of objects is essential for the communication and exchange of information between systems. This might be necessary for the communication between devices, but also between two systems based on different technologies, running on the same device. The focus of this section is on lightweight and high performance, availability in several programming languages/platforms and extendable solutions. The functionality of remote procedure calls (RPC) is also required.

### 6.5.1 XML-RPC

XML-RPC<sup>162</sup> is a specification for remote procedure calls over the internet. The underlying transport technology is HTTP, for the encoding XML will be used. The last changes in the protocol definitions have been made in 2003, so the protocol is quite established. Therefore, APIs are available in many languages, for some languages also several implementations are available. The usage of XML generates overhead, so the exchanged messages are longer than in other technologies and the decoding of XML consumes runtime.

### 6.5.2 Protocol Buffers

Protocol Buffers<sup>163</sup> is a serialization format with an interface description language developed by Google. The company has opened their serialisation technology, so it is publicly available. The data for serialisation have to be defined, compiled and the outcome source files can be included in the own implementation. The compiler supports output files for the languages Java, C++ and Python. Third party companies and projects offer implementations for several additional programming languages. Protocol Buffers is prepared for RPC, but does not include RPC mechanisms. Therefore, RPC implementations are available from third party companies and projects.

---

<sup>162</sup> <http://www.xmlrpc.com/>

<sup>163</sup> <http://code.google.com/intl/de-DE/apis/protocolbuffers/>

### 6.5.3 Thrift

Thrift<sup>164</sup> was developed by the company Facebook for their serialisation and RPC requirements. It became part of the Apache Incubator project<sup>165</sup> and is now publicly available. Quite similar to Protocol Buffers, the messages will be defined and the definition will be compiled. Thrift supports several programming languages and also includes RPC functionality.

### 6.5.4 Comparison of Technologies

Performance is a very important fact for many applications, using serialisation and RPC. So, people start to compare technologies. The above listed three technologies are just a small selection of many available technologies. This three were picked because of performance, availability, standards and usability. Thrift-protobuf-compare<sup>166</sup> is a project, comparing the performance of different serialisation technologies, with some meaningful graphics<sup>167</sup>

---

<sup>164</sup> <http://incubator.apache.org/thrift/>

<sup>165</sup> Apache Software Foundation. Apache Incubator. <http://incubator.apache.org/>

<sup>166</sup> Thrift-protobuf-compare. <http://code.google.com/p/thrift-protobuf-compare/wiki/Benchmarking>

<sup>167</sup> Wondering around. <http://www.eishay.com/search/label/Thrift>

## 7 State of the Art in Relevant Areas of Software Engineering

Large parts of the AsTeRICS project consist of developing drivers and applications for an embedded platform and creating a configuration suite which communicates with the platform. In this part of the document relevant areas of software engineering are discussed. In Section 7.1, different operating systems for embedded systems are evaluated. After this, frameworks which allow building and integrating such operating systems into a binary (fitting to a certain platform) are presented. Section 7.2 introduces Java support and Java Virtual Machines with a special focus on embedded use. Finally, Section 7.3 presents existing graphical modelling software for dynamic systems and signal processing and describes features of such tools which are important for the AsTeRICS architecture and Configuration Suite.

### 7.1 Operating Systems for Embedded Platforms

Embedded systems with complex functionality are usually structured as multiple programs for concurrency. The system may need to perform several operations at the same time, necessitating some amount of internal concurrency [138]. An operating system not only provides the necessary infrastructure to manage concurrent processes, but also provides an abstraction layer to the hardware, interfaces to input- and output devices, and a standardized software stack for networking (for example TCP/IP).

For the AsTeRICS project both, the decision for the target hardware platform and the decision for a suitable operating system are essential. The following sections will shortly introduce the principal concepts of operating systems and present an overview of the state-of-the-art in operating systems for embedded computing platforms which could be interesting for the AsTeRICS project. Please refer to section 5 for information on the supported operating systems for a particular hardware.

#### 7.1.1 Real-time Operating Systems and tiny RT-Kernels

The notion of “real-time” refers to the capability of a system to produce results within a timely deadline, even in a worst-case scenario. Hard real-time systems are characterized by the fact that severe consequences can result if logical as well as timing correctness properties of the system are not met (e.g. the control electronics for an airbag in the car). On the other hand, a soft real-time system has less stringent requirements for the timing, for example user interaction with the mouse.

##### RTOS, monolithic kernels and microkernels

A Real-Time Operating System (RTOS) is software that manages the time of a microprocessor or microcontroller. A real-time multitasking kernel (also called a real-time operating system, RTOS) is software which ensures that time critical events are processed as efficiently as possible. The use of a RTOS generally simplifies the design process of a system by allowing the application to be divided into multiple independent elements called tasks [139]. In a microkernel operating system, large parts of the system run as tiny, separate processes, which are protected by the memory management unit. In a monolithic kernel on the contrary, the whole operating system is a large program with special rights and

capabilities. Microkernels allow a strong customisation of the operating system where unneeded parts can be omitted to save system resources.

An RTOS allows real-time applications to be easily designed and expanded. New functions could be added without requiring major changes to the software. The RTOS can ensure that all time critical events are handled as efficiently as possible and allow a deterministic calculation of the systems execution- and response times. An important issue in real-time systems is the time required to respond to an interrupt and to actually start executing user code to handle the interrupt. Every RTOS disables interrupts when manipulating critical sections of code. The longer an RTOS disables interrupts, the higher the interrupt latency.

### **Processes and Multitasking**

The process is the basic unit of software concurrency. A process is an execution of a program, having its own state, which is often known as its context. A context switch changes control of the CPU from one process to another. A real-time operating system (RTOS) is often used to manage concurrent processes running on one or multiple CPUs. RTOSs are designed to meet the demands of real-time processing: they provide efficient context switching and mechanisms for meeting deadlines.

Multitasking is the process of scheduling and switching the CPU between several tasks; a single CPU switches its attention between several sequential tasks. Multitasking provides the ability to structure your application into a set of small, dedicated tasks that share the processor. One of the most important aspects of multitasking is that it allows the application programmer to manage complexity which is inherent in real-time applications. The RTOS is responsible for scheduling processes. However, the scheduling policy is determined in advance by the system architect. Processes in embedded systems often use some form of priority-based scheduling policy. Each process is assigned a priority. The highest priority process that is ready to run gains control of the CPU for the next time quantum. In a priority based kernel, control of the CPU will always be given to the highest priority task ready to run. When the highest priority task gets the CPU, however, depends on the type of scheduler used. There are two types of schedulers: non-preemptive and preemptive.

### **Preemptive vs. non-preemptive scheduling**

Non-preemptive schedulers require that each task does something to explicitly give up control of the CPU. To maintain the illusion of concurrency, this process must be done frequently. Non-preemptive scheduling is also called cooperative multitasking. In a preemptive scheduler, when an event makes a higher priority task ready to run, the current task is immediately suspended and the higher priority task is given control of the CPU (see Figure 114).

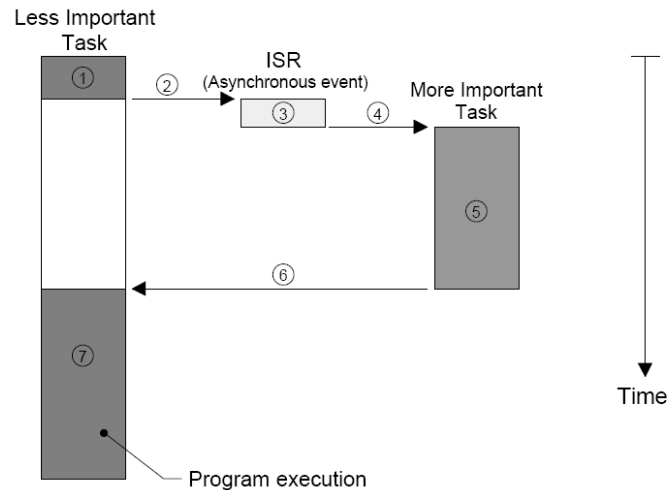


Figure 114: Illustration of preemptive scheduling [139]

### Kernel services

Real-time operating systems provide synchronisation services like semaphores or interprocess-communication to your application. A semaphore is a protocol mechanism used to control access to a shared resource (mutual exclusion), signal the occurrence of an event or allow two tasks to synchronize their activities. If the semaphore is already in use, the requesting task is suspended until the semaphore is released by its current owner. A suspended task consumes little or no CPU time [139], [140].

### RTOS variants

There are hundreds of RTOS vendors and products available for 8, 16 and 32 bit microprocessors. Some of these packages are complete operating systems and include a real-time kernel, an input/output manager, windowing systems, a file system, networking, language interface libraries, debuggers and cross-platform compilers [141]. A comprehensive list is beyond the focus of this document, but some example RTOS implementations and their characteristics are given in the Table 40.

Product	Short Description / special features	Licences / Price	Supported architectures
QNX, QNX Neutrino	POSIX compliant, Linux oriented, microkernel, GUI functions, common networking protocols and file systems, synchronisation, fault containment; many device drivers and board support packages <sup>168</sup>	Proprietary, source code available for non-commercial use	x86, PowerPC, ARM (including OMAP 3530 and L-138 platforms), MIPS and SH-4
FreeRTOS	Semaphores, preemptive and cooperative scheduling, IP-stack. very tiny (3 C-files)	Open source (LGPL alike)	ARM7, ARM9, ARM CortexM3, PowerPc, 8-bit uCs (PIC, AVR, MSP430)
VxWorks	Scalable footprint, Kernel configurator, USB and IP stacks, simulator and on-chip debugging support, multiprocessor support	Proprietary (WindRiver,Inc)	X86, PowerPC, ARM, MIPS, ColdFire

<sup>168</sup> QNX real time Operating System. <http://www.qnx.com/>

Product	Short Description / special features	Licences / Price	Supported architectures
RTAI Linux	Realtime extension for linux via kernel modules, adds interrupt handling between linux kernel and hardware, linux runs as low priority RT-task. user space realtime via LXRT extension <sup>169, 170</sup>	Open source (GPL for kernel space, LGPL for user space)	x86, Via, ARM9
RTLlinux	Runs entire linux system as preemptive process, POSIX API, RT core and linux tasks communication via message queues, compatible to 2.4 and 2.6 kernel versions [142]; complete distribution + development tools available with commercial license only	2 license types: OpenRTLlinux: (GPL); WindRiver RealtimeCore: (commercial)	X86, PowerPC, Dec Alpha; ARM (commercial license)
Windows Embedded Compact	Discussed separately, see Section 7.1.2.1		
PalmOS	Discussed separately, see Section 7.1.3		

**Table 40: Comparison of real time operating systems and kernels**

## 7.1.2 Windows Derivatives

Microsoft has taken to entering the mobile market as soon as 1996. The current product line is called Windows Embedded and consists of several OS versions for non-PC devices.

### 7.1.2.1 Windows Embedded Compact

Windows Embedded Compact (or Windows Embedded CE) is a commercial, component-based, real time operating system provided by Microsoft. It is intended for platforms with limited memory and processing resources such as minimal PCs or embedded systems. It provides a small memory footprint of less than 350kB [143]. Windows CE supports several processor architectures (x86, ARM, MIPS). Windows Embedded CE is not based on existing Windows versions and stripped down but is rather a separately developed kernel for small devices which can run in memory smaller than 1Megabyte. It provides real time support by providing deterministic interrupt latency and supports concurrent processes with dedicated virtual memory [143]. A specialised version of Windows Embedded Compact is Windows Mobile which is intended for use in mobile phones and PDAs. Licensees of Windows Embedded Compact have to agree to the Microsoft End User License Agreement (MS EULA) and receive binary packages precompiled for the target platform.

There exist several development tools for Windows Embedded Compact. Recent version of Microsoft's Visual Studio offer support for programming Windows Embedded applications and provide an emulator to test these programs on the host PC. Windows Embedded Compact uses the .NET Compact Framework (.NET CF) which allows programming in C# and VB.Net. The .NET CF provides a subset of classes from the full .NET Framework with some additional classes developed specially for mobile applications.

Microsoft also provides the Windows Embedded Platform Builder [144] which enables the developer to configure what features of the system should be enabled or disabled when building the OS for a certain platform (e.g. use of GUI components or not if no LCD display exists on the target platform).

<sup>169</sup> RTAI linux overview: [http://wuzwuz.nuigalway.ie/~pcor/lectures%20\(2005\)/embedded\\_systems\\_apps/Linux\\_Lecture\\_03\\_RTAI\\_Overview.pdf](http://wuzwuz.nuigalway.ie/~pcor/lectures%20(2005)/embedded_systems_apps/Linux_Lecture_03_RTAI_Overview.pdf)

<sup>170</sup> RTAI: Real-Time Application Interface: <http://www.linuxjournal.com/article/3838>

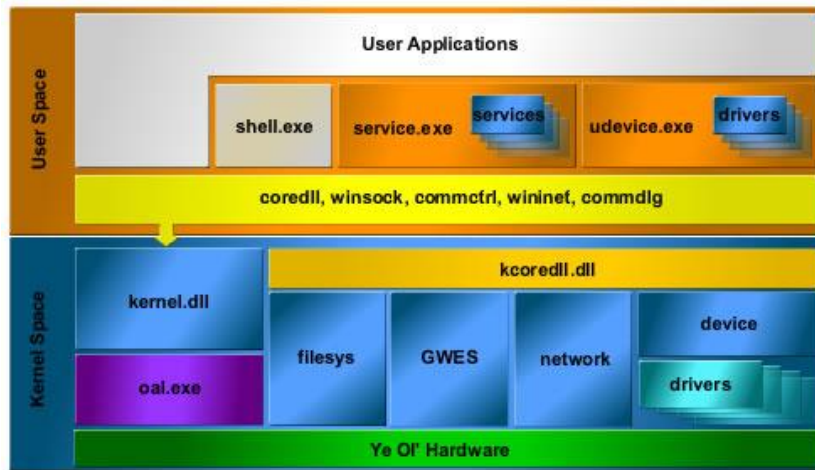


Figure 115: Windows Embedded Compact Kernel Architecture [145]

Figure 115 shows the kernel architecture of Windows CE 6. The kernel consists of its core (kernel.dll and oal.exe), a file system manager, a device manager which handles the device drivers, a graphical windowing and event system (GWES) handling GUI mechanics and a network manager [145].

Figure 116 depicts the OEM abstraction layer (OAL) of Windows Embedded Compact. The OAL abstracts together with a set of device drivers the underlying hardware that an original equipment manufacturer provides for a platform running Windows Embedded Compact. The OAL handles hardware units on the board such as real time clocks, timers, caches, interrupts and peripheral communication units [146].

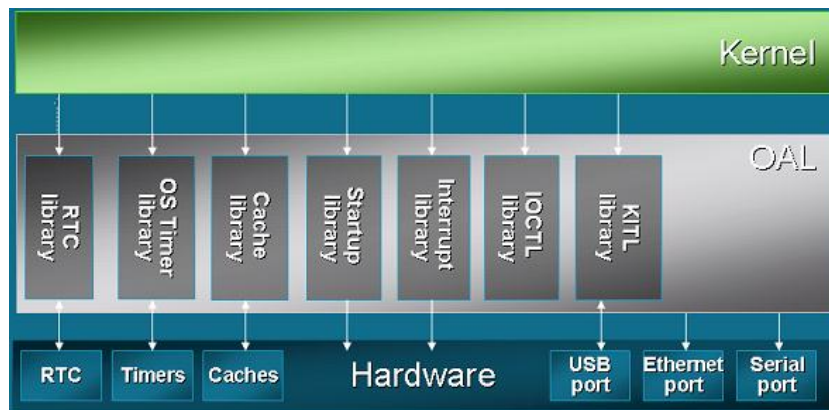


Figure 116: OAL Architecture [146]

The drawback of using Windows Embedded is the fact, that the operating system is a closed source system, shutting the developer out of the internals of the kernel and disallowing to introduce patches if necessary. The grant agreement of the AsTeRICS project states that the core of the system will be provided under the LGPL or GPL, therefore the Microsoft embedded operating systems cannot conform to this plan.



### 7.1.2.2 Other Windows Embedded Versions

As opposed to Windows Embedded Compact, Windows Embedded Standard is a Windows NT based version of an operating system for embedded systems. Windows Embedded Standard supports the x86 and the x64 processor architecture. It provides access to the complete Win32 API; therefore it can run all applications and device drivers written for the desktop version of Windows. Windows Embedded Standard and its predecessor Windows XP Embedded are used in such devices as Automated Teller Machines (ATMs), cash registers, set-top boxes and network attached storage (NAS).

### 7.1.3 Palm-OS

Palm OS is an operating system intended for the use in PDA devices and has recently been implemented in devices such as smart phones, gaming consoles, barcode readers and GPS devices. The OS was first written for Motorola's DragonBall processors but is based on ARM architecture since Palm OS 5.0.

Today two companies develop operating systems inheriting from the Palm OS: Palm Inc. which provides Palm webOs and the company ACCESS which develops the ACCESS Linux Platform as well as Garnet OS.

#### 7.1.3.1 Access Linux Platform + Garnet OS

Garnet OS is the renamed version of Palm OS that the current PalmOS developer company Access released after Palm Inc. bought back the rights of the trademark Palm. Garnet OS is the latest OS running on devices such as the Palm Tungsten T. It is tailored for ARM CPUs providing an emulator for older applications written for the former OS versions. The OS itself is a single tasked OS supporting several display sizes and input methods. It provides basic applications for personal data management.

The Access Linux Platform (ALP) started its releases in 2007. It was marketed as the next generation of Palm OS and was scheduled to be implemented in devices such as the "First Else" starting with 2009. The ALP is an operating system for mobile devices based on open software. It is as the name implies based on a Linux Kernel and on other open source components. It provides means to execute legacy Palm OS code, Java code and Linux applications based on GTK+. The use of components from different open-source projects makes ALP an open operating system on the software level. Furthermore Access' Hiker Application Framework has been released to the open source community. However the business model of Access works with hardware licenses [147]. Legacy Palm OS code is run in an emulated Garnet OS running in the ALP [148]. Currently only two smart phones use or have been announced to use the ALP. The Linux kernel sublayer can be accessed making it theoretically possible to develop drivers for other devices such as the sensors needed for this project.

#### 7.1.3.2 Palm webOs

Palm webOs is the Linux based newest generation of Palm OS provided by Palm Inc. The operating system is integrated into the Palm mobile devices Palm Pre and Palm Pixi.

The operating system is released under an end user licence agreement by Palm. Similar to the ALP webOs is based on a Linux kernel and uses several open-source software projects as components of the platform, all released under GPL or LGPL. Applications for webOs are

all web-based and developed using simple technologies such as HTML, CSS and Javascript [150]. However a Services API provides access to the hardware platform and webOs allows applications to run in the background or have multiple concurrent screens. Applications can generate alarms and events on the device thus allowing interaction with the hardware. The operating system features an application framework called Mojo and supports multithreading.

Palm webOs also includes a Plugin Development Kit [151], which enables the developer to create plugins written in C/C++ which complement the existing features of the Web applications. These plugins allow webOs applications to share a screen or to perform two way communication between application and plugin. Palm webOs is targeted towards the development of applications for use in mobile devices. The SDK allows to access the basic Linux kernel and to run the ipkg packet manager which makes it possible to create Linux device drivers thus making the operating system usable in terms of development for the AsTeRICS project. However the operating system is not completely open as some parts are distributed in binary form and protected by the Palm EULA.

Figure 117 shows the architecture of Palm webOs. The system is based on a variant of a Linux 2.6 kernel with an appropriate bootloader. This kernel provides the driver access to the hardware through the standard udev interface, access to the file system and to the network. On top of this kernel OS services, the media server and the communication stack are accessible by the upper layers. Applications are running on top of the user interface system manager which manages the current state and visibility of concurrently running applications.

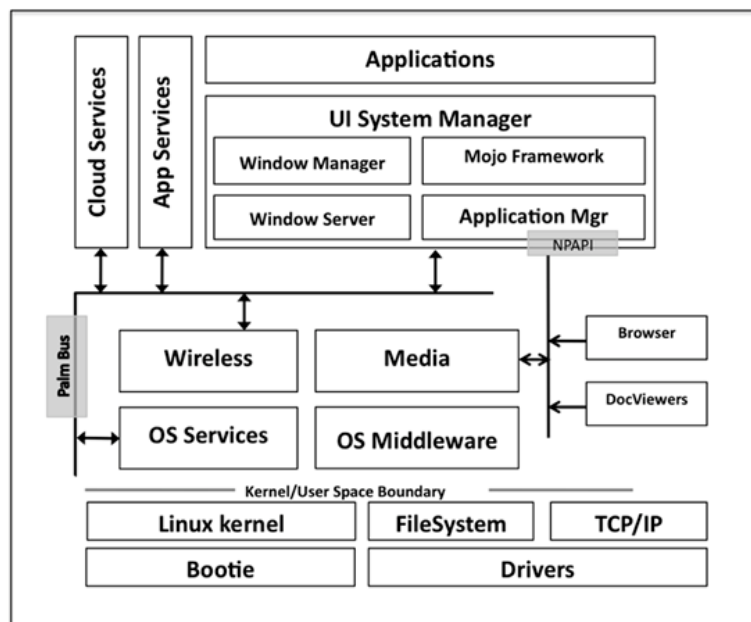


Figure 117: Architecture of Palm webOs [150]

#### 7.1.4 Symbian-OS

Symbian OS is an operating system intended for phones and mobile devices. It is based around the EKA2 (EPOC Kernel Architecture 2) kernel. As of Symbian OS 9.1, the operating system was released under the Eclipse Public License (EPL).

The kernel has a modular architecture to allow fast porting of it for different phone hardware architectures. It provides pre-emptive multithreading and full memory protection. It consists of the following parts [152]:

- A nanokernel handling the operating system basics such as scheduling, task synchronization and timing
- The Symbian OS kernel attaches to the services of the nanokernel and offers the functionality needed by the Symbian OS API. The kernel provides OS methods for working with threads, processes and to perform inter task communication.
- The memory model organizes the address space for the processes running in the OS and abstracts the access to the MMU from the nanokernel and the Symbian OS kernel.
- The Board Support Package (BSP) provides the abstraction of the hardware dependent components. It provides drivers and extensions to add API calls to access additional hardware the OS encounters on a specific platform.

The kernel is intended for use in mobile which can be observed in its ability to run a full signalling stack for telephony along with the operating system handling the user applications. Normally these stacks need to run in a separate CPU on the phone. The EKA2 provides means to emulate the CPU these stacks are written for, as porting them to Symbian OS itself is often not feasible.

Due to its strong tendency to being an operating system for mobile phones, Symbian OS does not seem to be a perfect candidate for the AsTeRICS project.

### 7.1.5 Linux Derivatives

In recent times many Linux based operating systems for embedded systems have been developed. The most important Linux based distributions are presented in the next sections.

#### 7.1.5.1 Android

Android<sup>171</sup> is a Linux based operating system for mobile and embedded devices. The operating system is developed by the company Google and its source code is released under an Apache license allowing other developers to create closed source changes to the code base without having to return them to the open source community.

Figure 118 shows the architecture of the Android operation system. The system is based on a Linux 2.6 kernel which is used as the hardware abstraction layer. Linux is used due to its proven driver model, a wide range of existing drivers for communication with peripheral components and the fact that it provides memory and process management and other core components of an OS. The kernel development team added a few drivers to the main branch of the Linux kernel source tree at first but later created a full branch from the code base creating a split Android kernel from the original Linux kernel. So although Android is Linux based not each new Linux kernel will work with the platforms Android runs on and only kernel updates from the Android developers can be used with the OS<sup>172</sup>.

---

<sup>171</sup> Android Website, <http://www.android.com>

<sup>172</sup> Androidology – Architecture Overview, <http://www.youtube.com/watch?v=QBGfUs9mQYY>

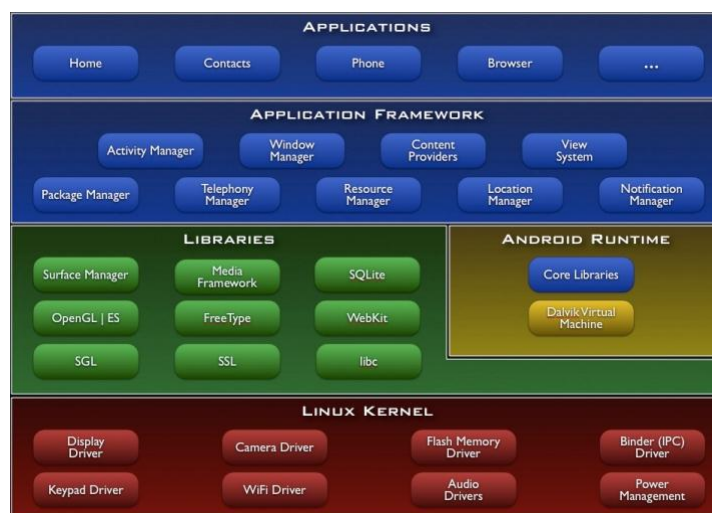


Figure 118: Android OS Architecture<sup>172</sup>

On top of the kernel a layer of native libraries written in C or C++ and compiled for the platform the OS is running on (currently mostly ARM CPUs) is placed. The libraries contain graphic handling in 2D and 3D, database connections, a browser engine, font rendering and a C library.

Android applications are written in Java and the applications are executed by the Android Runtime. This consists of a Java Virtual Machine, the Dalvik VM. Dalvik is a non standard Java Virtual Machine (JVM) which does not adhere to the Java SE or ME specifications (no support of J2ME classes, AWT or Swing). Dalvik is a register based JVM as opposed to the usual implementation of stack based JVMs. This approach allows Dalvik applications to be smaller in code size due to avoidance of opcodes for instruction dispatch and unnecessary memory accesses. However this requires byte code for the Dalvik VM to be converted from standard Java bytecode to an alternative instruction set which is understood by the Dalvik VM. This code is compiled into a file format called `.dex`. Uncompressed `.dex` files are commonly a few percent smaller than compressed `.jar` files. Furthermore memory usage is reduced by using shared constants of the classes of an application in the `.dex` file. The Dalvik VM is bundled with a set of core components written in the Java language (depicted in blue) in the runtime core libraries<sup>173</sup>.

On top of the runtime the application framework provides the means to handle the lifecycle and requests of any Android application. It is designed to make reuse of components easy. Each application can make use of certain capabilities of other applications (such as the possibility to select files from a folder) removing the need to re-implement such requirements. Apart from application lifecycle, the framework consists of the telephony management, window management and notification system among others. The application layer is where applications are running and can access the provided features from the application framework. Above the Android runtime, all code is written in Java. Each Android application is running in its own process and is executed by a separate instance of the Dalvik VM in the process<sup>174</sup>.

<sup>173</sup> Google I/O 2008 – Dalvik Virtual Machine Internals, <http://www.youtube.com/watch?v=ptjedOZEXPM>

<sup>174</sup> What is Android?, <http://developer.android.com/guide/basics/what-is-android.html>

Android provides an SDK with emulator and IDE integration which allows to develop Android applications in Java using the Android API, the newest API version is 2.1 (Codename Éclair). Development of native parts of code for applications is possible using the Android Native Development Kit (NDK). The NDK allows coding in C and C++ and compiling this code with GCC for the platform of the embedded system. Android applications can access this native code using the Java Native Interface (JNI).

Custom implementations of Android and modifications to port Android to new devices are done using the Android Platform Development Kit (PDK). Device driver development is similar to driver development under Linux and modules can be added to the kernel or loaded as modules. Android in its officially released form does not offer access to the Linux subsystem however using the PDK one can modify the system to provide access to the full Linux system.

The open software base of Android makes the system a viable option for the AsTeRICS project. There are however a few non standard components in the operating system. The use of a Java Class library different to the common specifications as well as the use of a non standard C library make development and especially porting of code to Android complicated.

#### 7.1.5.2 Maemo

Maemo<sup>175</sup> is an operating system and development framework for mobile devices of the Internet Tablet class. It has strong ties to the mobile phone manufacturer Nokia and is used in Nokia's N devices from N770 upwards. Maemo is based on a Debian Linux Distribution and heavily uses open source projects. GUI handling, most frameworks and libraries hail from the GNOME project.

Unlike most other operating systems for mobile devices, Maemo provide easy root access to the Linux base. The user can enter a command "sudo gainroot" in the shell.

Maemo uses a Linux 2.6 kernel bundled with a Debian version stripped down for mobile devices. BusyBox is used as a small footprint version of the Linux core utilities. The graphical components are using the X server system and the Matchbox window manager. Matchbox is a simplified window manager for mobile devices which only allows showing one window at a time. The graphical user interface is in the recent versions based on either GTK+ or on Qt. Applications use the Hildon UI framework which was developed to provide a finger friendly user interface for use with touch screen devices and provides methods to manage the desktop and updates<sup>176</sup>. Interprocess communication (IPC) in Maemo is done using the D-Bus which is the standard way of IPC in GNOME and Qt. It offers methods to allow two applications to communicate with each other by having them register to wait for certain events on the bus. These events can either be messages from other applications or events from the kernel.

Like Android, Maemo seems like a possible operating system for the AsTeRICS platform, providing a fully fledged Linux platform for embedded systems. The core provides lots of functionality which is not needed in the context of this project.

#### 7.1.6 Build Frameworks for Embedded Linux and Cross Compilation

---

<sup>175</sup> Maemo Platform Website, <http://www.maemo.org/>

<sup>176</sup> Maemo 5 Software Architecture, [http://wiki.maemo.org/Documentation/Maemo\\_5\\_Developer\\_Guide/Architecture/Top\\_Level\\_Architecture](http://wiki.maemo.org/Documentation/Maemo_5_Developer_Guide/Architecture/Top_Level_Architecture)

The process of compiling an operating system for an embedded system consists of several steps. One needs to compile a kernel and the libc C library, generate a root file system and provide several shell programs. This process needs to be mostly done as a cross compilation, running the compile on a different computer (and often processor architecture) than the targeted embedded system. In other words cross compilation is the method of compiling code for the instruction set of one processor family (target) on a system running on a different instruction set (host). The resulting binaries cannot run natively on the host system (unless an emulator for the instruction set exists) but can be transferred to the target which can handle them.

There are several frameworks which allow automating the build process of such an OS. Relevant frameworks are presented in the following section.

### 7.1.6.1 BuildRoot

BuildRoot<sup>177</sup> is a set of scripts which automate the process of compiling an OS for an embedded system. It is based on the make utility provided in Linux and provides a menuconfig method (see Figure 119) to configure the correct architecture, kernel and packages used for the OS on the embedded system. After configuration a call to make will start the build process, automatically resolving dependencies, download, configure and building needed packages.

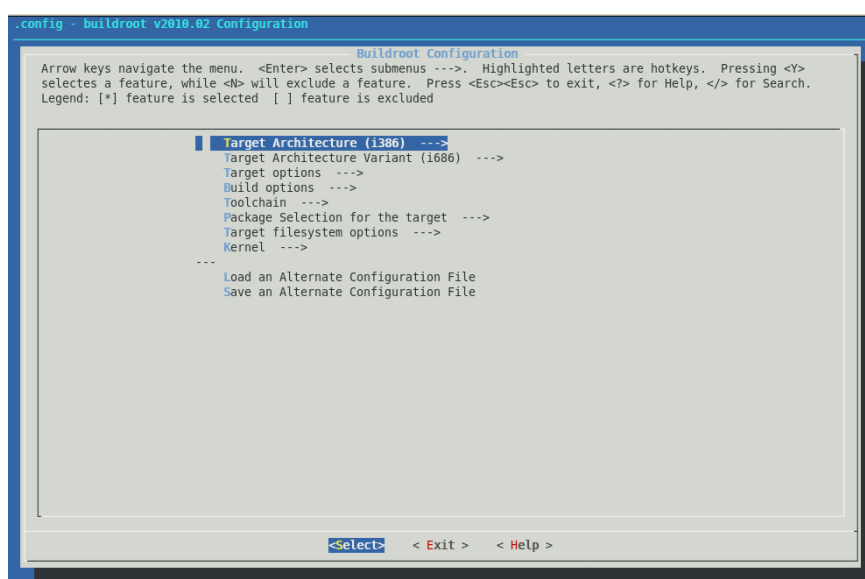


Figure 119: Screenshot from BuildRoot configuration menu

Buildroot supports a large number of target architectures like x86, ARM, AVR32, MIPS and Sparc.

### 7.1.6.2 Chroot / Portage

A change root (chroot) - environment is commonly used for development- and cross-development related work in the Linux-OS, because it allows installation and usage of build-related software without affecting the normal operating environment of the Linux computer. As some packages are not very cross-compilation friendly it is never a good idea to do cross-development directly in the main Linux instance. For example some packages might end up

<sup>177</sup> Buildroot Project Website, <http://buildroot.uclibc.org/>

overwriting the host's libraries with cross-compiled ones, potentially rendering the host system unusable. Performing development work in a chroot-environment encapsulates the installation of development packages (whose main purpose is for building packages) and other build-related infrastructure, because the chroot'ed process and its children are restricted to a subfolder in the file system. This subfolder features the same structure as the root directory of a Linux system.

For building a Linux system for a different (embedded) target device, chroot can be used together with source-based package management utilities like the Gentoo portage application. Gentoo Linux is a source-based distribution, which means instead of a (binary) package management (like rpm) it uses portage which will download sources and then configure, compile and install the package, also called "emerging". This has various advantages, like being able to completely customize dependencies and use of libraries as well as features of packages. As such, Gentoo Linux does not have a "version" per se, as it is always possible to update the portage tree which has so-called "ebuilds" which are comparable to OpenEmbedded's "recipes" and contain instructions for portage on how-to emerge a specific package. Thus it becomes rather easy to use portage for cross-compiling, as the only difference - from an abstract view - in the whole process of "emerging" will be the use of a different tool-chain.

As mentioned previously Gentoo Linux uses "emerge" (part of "portage") as its "package manager". Using emerge it is easy to compile almost any existing Linux software, that is in the portage tree. A cross-development platform is represented by the corresponding emerge version, for example "emerge-armv7l-glibc-linux-gnueabi" for ARMv7 devices. Using just "emerge" will compile for x86 and install in the chroot environment. This can be used to add a package to the host chroot environment itself. It is very easy to write your own ebuilds and use them to handle your own projects. This is not required though, as all standard development tools are available for "traditional" compilation.

Find further information via the linux man pages "man emerge" or see <http://linuxreviews.org/man/emerge/> for the complete man-page.

### 7.1.6.3 Scratchbox

Scratchbox<sup>178</sup> is a sandbox environment providing tools to cross compile software for a certain platform. Scratchbox was developed as the cross compilation tool for the Maemo OS. Just like chroot Scratchbox provides a chroot jail, which moves changes the work environment in such a way that the root directory does no longer point to the host system's root but to another destination. Thus the cross compilation tools can access their libraries without interfering with the libraries of the native operating system. Furthermore isolation is achieved through library call diversion and wrapping of compiler executables.

Scratchbox

- Is a software package to implement development sandboxes (for isolation)
- Contains easy-to-use tools to assist cross-compilation
- Supports multiple developers using the same development system
- Supports multiple configurations for each developer

---

<sup>178</sup> Scratchbox Project Website, <http://www.scratchbox.org/>

- Supports executing target executables on the hardware target, via a mechanism called sbrsh
- Supports running non-native binaries on the host system via instruction set emulators (like Qemu).

Furthermore it is possible to create software packages which after some configuration can be exported and afterwards downloaded via the internet on other platforms using the standard Debian package manager. The generated packages can be used in the Scratchbox emulations or on the actual device.

The Scratchbox was originally started by a company called Movial and has been sponsored by Nokia. It is provided under the GNU General Public License (GPL) and was initially intended to work with Debian Linux distributions although other distributions are now also including Scratchbox. Since it was developed for the Maemo environment, the Scratchbox toolkit naturally supports the ARM architecture of the Nokia N Devices. It also supports the x86 architecture and is moving towards support for PowerPC and MIPS.

#### 7.1.6.4 OpenEmbedded

OpenEmbedded<sup>179</sup> represents a build framework for embedded Linux which allows the creation of complete Linux Distributions for embedded systems. OpenEmbedded uses so-called “recipes” which contain the dependencies of a specific package and are processed by the “bitbake” tool, which calls common build tools like compiler, cross compiler and linker. Bitbake works with a directory structure containing folders for e.g. sources code, staging (runtime files and libraries which are needed to build target packages), and deploy (resulting compiled packages that can be installed using a package management system like opkg, or complete disk images which include bootloader and linux kernel or the complete root file system with installed packages)

OpenEmbedded cross-compile thousands of packages including GTK+, Qt, the X Windows system or Java. Even the target tool chain (gcc for the desired target platform) is compiled during the build process of a package, if necessary.

OpenEmbedded offers support for many hardware architectures, and multiple releases for those architectures. One of the biggest advantages of OpenEmbedded is that the build system can be easily customized for a wide range of target architectures and can be integrated with minor efforts by using appropriate patches to the build recipes. Building a Linux distribution for a different machine is a matter of changing the machine type in the bitbake config file (given that the architecture is already supported in OpenEmbedded). The configuration can be specified by setting the respective environment variables. This makes OpenEmbedded a reasonable choice for bringing a Linux operating system to a state-of-the-art embedded hardware platform.

Among the primary features of OpenEmbedded are:

- Handling of cross-compilation
- Handling of inter-package dependencies
- Ability to emit packages (tar, rpm, deb, ipk)
- Ability to create images and feeds from packages

---

<sup>179</sup> OpenEmbedded Project Website, [http://wiki.openembedded.net/index.php/Main\\_Page](http://wiki.openembedded.net/index.php/Main_Page)



- High configurability to support many machines, distribution and architectures.

Among the disadvantages of OpenEmbedded are the huge size of the source trees (including all the patches for different machines) and the great amount of hard disk space which is needed to build distributions or packages. For a complete build of a root file system including major packages like mplayer and Firefox and a graphical user interface (e.g. gnome desktop) around 100 GB of disk space on the host computer are needed and more than 1,4 million files are created in the OpenEmbedded folder.

#### 7.1.6.5 Angström Linux

Angström Linux is a scalable Linux distribution with high portability and a reconfigurable core. Angström relies heavily on OpenEmbedded and is its de facto standard distribution. It scales from devices with only 4MB of flash to devices with several Terabytes of storage capabilities<sup>180</sup>.

Angström Linux should run with every device OpenEmbedded supports that can run a Linux 2.6 kernel and it especially focuses on embedded devices such as PDAs, mobile phones and routers. However it can also run on a standard desktop PC. Angström provides many of the features of conventional desktop Linux distributions such as a package management system, networked installation of pre-built packages and automatic dependency resolution. [153] provides a list of the supported architectures.

Several pre-built images of Angström Linux are available in different flavours from a base image with nearly no features up to a desktop version with a prebuilt office suite. The image is placed in the flash memory or another type of storage such as an SD card and upon reset the machine specific bootloader loads the image. After this the kernel mounts the root file system and runs through the init procedure as in every other Linux distribution.

On the website of Angström Linux an online image builder called Narcissus is provided. This tool provides a web interface on which the developer can select the platform, the user environment and which packages should be included in the image. After selection the image is built on a remote server and appears as an FTP link on the web interface.

#### 7.1.6.6 Poky Linux

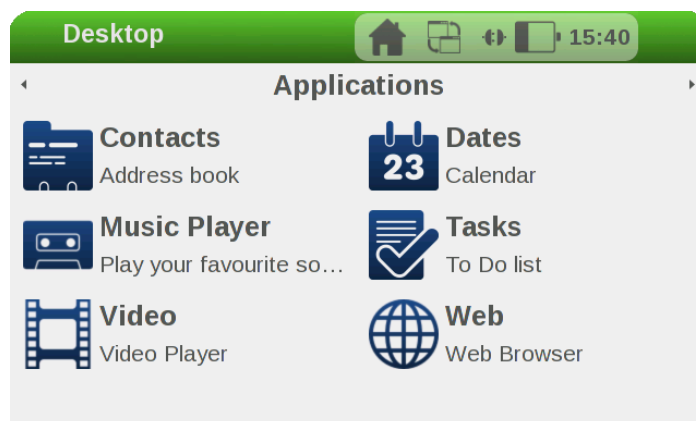
Poky [154] declares itself an open source platform build tool providing a software development environment for creating images for platforms running a Linux operating system. It is based on OpenEmbedded but has been customized to use only a stable and focused subset of OpenEmbedded containing open source components such as Linux, X11, Matchbox, GTK+, Pimlico and Clutter. Its further intent is to support a large number of platforms based on x86 or ARM architecture.

Poky uses the KDrive X server, which is a thin implementation of the X server popular in embedded devices, the Matchbox window manager and the D-Bus message bus system. Poky fully works with QEMU, an emulator for embedded CPUs, which makes Poky a useful development and testing platform. Poky comes with Sato, a user interface environment based on Gnome Mobile, which provides an efficient and fast user interface designed for smartphone and PDA screens. Sato uses the Matchbox window manager which, as

---

<sup>180</sup> The Angström Distribution, <http://www.angstrom-distribution.org/>

mentioned above, is developed for embedded platforms and provides only one window at a time (see Figure 120).



**Figure 120: Sato Desktop managed by Matchbox window manager**

Poky builds images using the bitbake tool with a customized set of recipes. It comes with a set of more than 500 pre-configured software packages. Furthermore most of OpenEmbedded 5000 software packages can be easily included into Poky.

## 7.2 Java Support and Java Virtual Machines

Java is a programming language developed to create platform independent code which can easily be ported to different CPU architectures. The source code of Java applications is compiled into Java byte code which is in turn executed by the Java Virtual Machine (JVM). This virtual machine can interpret the byte code which naturally needs to translate a byte code instruction into a series of native processor instructions to perform the action the byte code instruction represents and is therefore slower than native code. In newer versions (at least of the official Sun JVM) the Java Runtime performs Just-In-Time (JIT) compilation of the code which translates the byte code into native code of the CPU architecture the JVM is running on. Thus JIT compilation allows the Java code to run as fast as native code after a short period of reading and translating the byte code, revoking arguments of Java opponents about the reduced speed of Java due to its being interpretive. Another option of byte code conversion to native code to improve speed is the use of Ahead-Of-Time compilation which performs the conversion to native code during compile time or at least before executing the program. An advantage of this method is the fact that the execution time of the instructions making up a certain portion of code do not change during runtime which makes the program's execution more deterministic, a fact that is very important when operating a real-time system.

As opposed to C and C++ Java Virtual Machines come with a convenient memory management system called garbage collection (GC). This means that the developer does not have to release allocated objects when they are no longer needed in order to free memory. The garbage collector tracks objects which are no longer referenced by the program, releases them and frees the memory occupied by them. Several garbage collection algorithms:

- **Stop-the-world mark and sweep** methods are the simplest version of GC. The JVM halts all threads, frees the unused memory and restarts the threads. This method can lead to memory fragmentation.
- **Stop-the-world mark and copy** methods work like mark and sweep but moves the still referenced objects to avoid memory fragmentation. The method takes a little longer as references have to be updated to point to new locations
- **Generational methods** use certain assumptions which objects are often released quickly (such as local variables in a function) and only scan those objects most of the time
- **Incremental methods** do not stop the world for a full scan but only perform short incremental steps of GC without interfering too much with the program execution

The main drawback of GC has already been mentioned in the method descriptions. The GC collection needs extra time and introduces non-deterministic pauses into the execution of a Java program. Incremental GC helps with this issue but cannot completely remove it.

Although Java is advertised as a platform independent language this is only true for the byte code itself. However a virtual machine which interprets the byte code and can run on the processor architecture the Java application shall run on is needed. Different implementations of a JVM are normally released for several operating systems running on certain processors. In the following a few open source Java Virtual Machine implementations which could be used in the AsTeRICS project are presented.

### 7.2.1 Java SE for Embedded Use

Sun Microsystems provides a Java Runtime Environment compatible to the Java Standard Edition specification for embedded systems<sup>181</sup>. The runtime is implemented for x86, ARM, PowerPC and MIPS architectures using the basic processor instruction set to allow the broadest range of processors to work with the runtime. The runtime is recommended to be run with at least 32MB of RAM and 32MB of ROM or other storage. Java SE for Embedded Use supports the full SE specification adding configurations to optimize memory usage and reduce the footprint of the runtime environment. Depending on the implementation the JRE supports either Java 1.4.2 (MIPS), Java 5 (PowerPC, x86) or Java 6 (ARM)<sup>182</sup>. Small footprint runtime environment can reach a ROM footprint of as low as 24MB on Java 1.4.2 and 30MB on Java 5<sup>181</sup>.

Java SE for Embedded is available for free for development but needs to be commercially licensed when deployed in an embedded system which does not comply with the constraints the grant agreement makes on open software.

---

<sup>181</sup> Java SE for Embedded Use, <http://java.sun.com/javase/embedded/>

<sup>182</sup> Java SE for Embedded Use – System Requirements, [http://java.sun.com/javase/embedded/system\\_requirements.jsp](http://java.sun.com/javase/embedded/system_requirements.jsp)

## 7.2.2 OpenJdk

OpenJDK<sup>183</sup> is the open source variant of the JRE and class library provided by Sun Microsystems. OpenJDK is released under GPL with a classpath exception. This exception declares that the use of an unmodified class library when creating a new Java compiler or a Java application, the GPL has no effect on the licensing of the created application allowing the developers to keep their sources closed even when building additional layers onto the existing Java runtime which would normally require a release of the full source code according to the GPL. OpenJDK first implemented the specification of Java 7 but has later been backported to support the Java 6 language specification.

OpenJDK is made available mostly as open source but some parts of it were at first only distributed as binary closed source packages. Sun Microsystems however promised to gradually replace the commercial packages by open sourced free versions of them. As of May 2008 only one per cent of the code was still proprietary code. A project called IcedTea was created to allow building of OpenJDK without any proprietary code pieces in an effort to add OpenJDK to Linux distributions which do not allow any proprietary software. The fact, that all these free software parts can be built from source theoretically allows IcedTea to compile OpenJDK using any GCC implementation available thus making it possible to build the JDK for the platform used in the AsTeRICS project<sup>184</sup>.

## 7.2.3 JamVm

JamVM<sup>185</sup> is an open source Java Virtual Machine which conforms to Java Specification 5. The JVM is built for having a very small memory footprint of little more than 200 kilobytes. It supports the full Java SE specification and provides support for the Java Native Interface and the Reflection API. The interpreter supports JIT compilation using super-instructions which as the project website claims has the same performance as other simple JIT compiler while still staying portable. On Linux JamVM supports the following architectures: PowerPC, i386, ARM, AMD64 and MIPS. Ports to other architectures should be simple as most of the JamVM code is written in C.

JamVM has been developed with improvement of performance and reduction of memory footprint in mind. This is supported by several features such as<sup>185</sup>:

- Use of Posix threads with a full thread implementation including Thread.interrupt() and reduced suspend latency
- Object references implemented as direct pointers without going through object handles
- Improved locking mechanisms reduce locking overhead
- Two word object header to minimise heap overhead
- Garbage collector can run synchronously or asynchronously within its own thread
- Use of a lightweight native interface for internal native methods removing the overhead of going through JNI

---

<sup>183</sup> OpenJDK project website, <http://openjdk.java.net/>

<sup>184</sup> Free and Open Source Java, <http://www.sun.com/software/opensource/java/faq.jsp>

<sup>185</sup> JamVm Website, <http://jamvm.sourceforge.net/>

Garbage collection is done using a stop-the-world mark and sweep garbage collector which introduces a certain non deterministic latency into the running system and might lead to heavily fragmented memory in long running applications.

The amount of supported hardware architecture makes the use of JamVm a possible option in the AsTeRICS project, if a JVM is needed.

### 7.2.4 Cacao

Cacao<sup>186</sup> is a JVM developed by the Institute of Computer Languages at the Technical University of Vienna. It is released under the GNU General Public License (GPL) while containing some third-party components which are however compatible to the GPL. Other licensing methods can be negotiated with the commercial partner Theobroma Systems. Cacao comes with a JIT compiler which translates Java bytecode into native code. Cacao implements most of the Java Standard Edition 5 specification which provides higher efficiency than older versions of Java.

Implementations of the Cacao VM exist for i386, x86\_64, ARM, PowerPC, MIPS64 and SPARC 64 architectures. Garbage collection in Cacao currently uses the BoehmGC, a conservative garbage collector also used available for C and C++. However the project website announces improvements in garbage collection including a generational GC which allows to only inspect certain objects at certain points in the code helping to improve performance<sup>187</sup>. There is no indication whether the new GC will be incremental or stop the world; however the BoehmGC claims that it does incremental GC on platforms that support this. Therefore it can be assumed that the new GC will also be incremental reducing the performance loss when the garbage collection is initiated.

Cacao can be built with different configuration in which some features are disabled reducing the memory footprint of the virtual machine. The VM works either with GNU Classpath as class library or can use the SUN class library implementations from OpenJDK or PhoneME.

### 7.2.5 PhoneME

PhoneME<sup>188</sup> is a Java ME implementation of a JVM by Sun Microsystems. The PhoneME project is released as open source under the GPL. Java ME is aimed at small devices like mobile phones, PDAs or set top boxes. Java ME comes with two configurations:

- The Connected Limited Device Configuration (CLDC)
- The Connected Device Configuration (CDC)

While the CLDC provides a much reduced class library and represents the minimum a virtual machine needs to work with a program, the CDC is a subset of the Java Standard Edition specification. It contains most of the class libraries except for the GUI implementations. On top of these configurations, a profile is used to bind the class library of the configuration to the device. Such profiles are e.g. the Mobile Information Device Profile (MIDP), the Foundation Profile and the Personal Profile.

The current project page of PhoneME offers implementations of both configurations. The PhoneME Feature software is a project targeted towards “feature phones”, phones typically

---

<sup>186</sup> CacaoVM project website, <http://www.cacaovm.org/>

<sup>187</sup> The new exact Cacao Garbage Collector, <http://c1.complang.tuwien.ac.at/cacao/wiki/GarbageCollector>

with high-resolution screens, multiple forms of messaging (SMS, MMS, IM, Email), basic 2D and 3D gaming, a camera, music player, Internet browser, etc. This software implements the CLDC and the MIDP which offers GUI features, data storage and basic 2D API. The PhoneME Advanced software implements the CDC Foundation Profile which was developed for devices which are in need of a full implementation of the Java Standard Edition<sup>188</sup>.

Concerning the built of the JVM from sources to work with the processor architecture it should run on, OpenEmbedded Bitbake recipes exist for both the Feature and the advanced implementation<sup>189</sup>.

Since PhoneME is a Java ME implementation it provides less extensive APIs however the memory footprint of this JVM fits more severe memory constraints. Based on the choice of working platform for the AsTeRICS project this might or might not be an issue in the selection of a JVM if one is needed.

## 7.2.6 Summary of Java Virtual Machines

The following table shows a comparison of the presented Java Virtual Machines based on the most important features.

	OpenJDK	JamVm	Cacao	PhoneME Advanced	Java SE for Embedded Use
Implemented Java version	SE 6	SE 5	SE 5	SE 1.4.2	See Section 7.2.1
Just-in-Time compilation on program execution	Yes	Yes	Yes	Yes	Yes
Garbage collection	Garbage-First [172]	Stop the world - Mark and sweep	Stop the world - Mark and sweep	Generational	Not stated
Licensing	GPL	GPL	GPL	GPL	Royalty when deployed

**Table 41: Comparison of different JVMs**

## 7.2.7 Real-time Issues in Java

The desire to use Java/OSGi for the AsTeRICS application framework makes the timing situation even worse: A Java application running on a standard JAVA Virtual Machine without special modifications does not guarantee predictable timing behaviour. The garbage collection mechanism can occupy the CPU at any time for hundreds of milliseconds. There are JVM implementations which, to a certain degree, introduce real-time capabilities in JAVA which will be presented in the following. These frameworks are mostly bound to commercial licenses and often do not support all standard JAVA class implementations.

<sup>188</sup> PhoneME project website, <https://phoneme.dev.java.net/>

<sup>189</sup> OpenEmbedded Wiki – Java, <http://wiki.openembedded.net/index.php/Java>

## Timing evaluations for the AsTeRICS runtime environment

To keep jitter and latency times low is a major challenge for the implementation of the AsTeRICS runtime environment, as unpredictable interruptions would be annoying to the user, especially when running mouse-cursor control configurations.

Several system parameters, process properties and implementation strategies can be used to tune the responsiveness of the runtime environment even in a standard Linux operating system and JVM:

- The priority of Linux processes can be changed via the nice level and the scheduling policy (linux *sched\_tool* utility)
- A lightweight window manager occupies less system resources than a standard one
- Background services can be reduced to the minimum
- the JAVA garbage collection mechanism can be performed incrementally, minimising undesired latencies

In the first phase of WP4 (software development), performance tests with different operating system and JVM configurations will be implemented to evaluate the feasibility of a standard Linux distribution with a JVM for the intended system functionalities. As defined in Risk R1 of the Risk analysis and Contingency Plan, appropriate actions will be taken if these tests show that the responsiveness or overall performance of the system on the desired hardware platform would not be sufficient (see AsTeRICS Description of Work<sup>190</sup>, Risk management).

In order to enhance the Java runtime to work in a more deterministic manner needed for real-time applications, the Real-Time Specification for Java (RTSJ)<sup>191</sup> was created. The RTSJ introduces several concepts to the Java language which had previously been unaddressed. These features are:

- The possibility to schedule threads
- Scoped and immortal memory to improve memory management and reduce problems during garbage collection
- Extended semantics for task synchronization of real-time threads with each other as well as with non real-time threads
- Asynchronous event handling, context switches and termination of threads
- Direct memory access

In the following a few real time Java frameworks, commercial or free, will be presented in short.

### 7.2.7.1 JamaicaVM

JamaicaVM<sup>192</sup> by the company aicas is a real-time enabled JVM. It guarantees hard real-time execution of all primitive Java commands even in the memory allocation related areas. Garbage collection is done using a hard real-time garbage collector which works incrementally and only uses a few machine instruction cycle per increment. Compilation is done statically and analyzed and optimized to such an extent, that the code size is minimized

---

<sup>190</sup> Description of Work, technical annex 1 of the AsTeRICS Grant agreement

<sup>191</sup> Real-Time Specification for Java Project Website, <http://www.rtsj.org>

<sup>192</sup> JamaicaVM – Java Technology for Realtime, <http://www.aicas.com/jamaica.html>

as much as possible. JamaicaVM is available for development on Windows, Linux and Solaris operating systems, works with many embedded operating systems such as Linux, Linux/RT, VxWorks and Windows CE and many CPU architectures such as x86, ARM, PowerPC and MIPS among many others.

### 7.2.7.2 Aonix PERC

Aonix<sup>193</sup> is a company specialising in the field of real-time systems and safety critical applications. The product line PERC provides Java virtual machines which meet real-time requirements. The JVM was developed from scratch for embedded systems and provides different configuration ranging up to the full support of the Java SE 5 specification. It is tailored for fast response times, achieving reactions to an event in less than 1ms. It supports interpreting execution as well as JIT compilation and ahead of time (AOT) compilation. Garbage collection is handled in a deterministic way reducing jitter of thread execution times almost completely.

### 7.2.7.3 WebSphere Real Time

WebSphere Real Time<sup>194</sup> is a real-time enabled JVM developed by IBM. It supports the Java Standard Edition 6 Specification. Code compilation uses AOT compiling and saving of byte code to provide a deterministic task execution length, a JIT compile runs as a low priority thread not interfering with the higher prioritized code execution. The garbage collection is done in real-time and provides determinism in the duration of garbage collection cycles. The runtime environment allows the developer to use the standard Java environment in a deterministic fashion providing response times of less than 1ms on Red Hat and Novell real time operating system and below 3ms on other OS. If faster reaction times are needed the developer can directly access the extensions the RTSJ specification provides.

WebSphere Real Time is intended for use on PCs and servers running a RedHat or SUSE Enterprise Linux distribution on an x86 or ia64 architecture and is therefore not ideally suited for use in the AsTeRICS project.

### 7.2.7.4 JAVA SE Real-Time Systems

Sun Microsystems provides a real-time enabled implementation of RTSJ called Sun Java Real-Time System<sup>195</sup>. The runtime environment is available for free for evaluation (90days) and academic use but has to be licensed when commercially used. The system requirements state the need for a dual core system and one of the following real time operating systems: Solaris 10, SUSE Linux Enterprise Real Time 10 or Red Hat Enterprise MRG.

### 7.2.7.5 Javolution

Javolution<sup>196</sup> is an open source implementation of the RTSJ. It is released under the BSD License. The intent of the Javolution project is to improve the performance and predictability of Java applications. It provides new real-time compliant implementations of the standard libraries Java (such as util, lang, text and others). The replaced classes remove a list of non-deterministic features from the normal implementation like class initialisation at the first time

---

<sup>193</sup> Aonix PERC Products, <http://www.aonix.com/perc.html>

<sup>194</sup> IBM WebSphere Real Time, <http://www-01.ibm.com/software/webservers/realtime/description.html>

<sup>195</sup> Sun Java Real-Time System, <http://java.sun.com/javase/technologies/realtime/index.jsp>

<sup>196</sup> Javolution Project Website, <http://javolution.org/>



of use instead of when instantiated and resizing of arrays. Context programming is used to achieve separation of concerns and Javolution supports parallel computing by providing a `ConcurrentContext` class. Interfacing with native applications is supported by base classes for programming with structs and unions. Concerning deterministic behaviour of the classes in the Javolution packages is achieved in the range of milliseconds.

Javolution is written entirely in Java, which makes it run on any platform which can run a JVM. It is moreover working with every Java specification from Java ME to the Java Enterprise Edition (EE) and has a code memory footprint of about 300kB. As of April 2010 Javolution is available as an OSGi bundle.

### 7.3 Graphical Modelling of Dynamic Systems and Signal Processing

Visual modeling of signal processing configurations allows building virtually any type of signal processing chain without the need to know a particular programming language. A design is created and adjusted visually by connecting graphical design elements via channels and editing properties of those elements. The design flexibility is only limited by the available processing elements and the supported input- and output modalities (e.g. hardware support for measurement equipment, sensor interfaces, interfaces to control hardware or software support to display data on the computer screen in an appealing way).

Especially in the area of scientific computing, visual design has become very successful and is a de facto standard for mathematical modeling in engineering and design of control systems today. Feature-rich application packages like Matlab/Simulink or LabView have contributed significantly to the success of visual design for signal processing, as their use can reduce prototyping times for new scientific approaches and even time-to-market of new product solutions tremendously.

The idea of visual design for dynamic modeling is quite simple: Many different problems (ranging from simple simulations to complex systems) can be composed out of atomic functional blocks. Drawing a block diagram or a flow chart diagram is a common method in engineering to reduce the complexity of a system or to illustrate its structure. By providing functional blocks via graphical design software, creating even large models becomes possible in an intuitive way, without having to know about implementation details of a programming language or operating system. The block diagram or flow chart not only illustrates the structure of a system, it also defines its functional properties and implementation. Furthermore, the reuse of well-tested functional units decreases development time and increases robustness of the resulting model.

A visually designed signal processing chain is composed out of following fundamental types of design elements:

- Signal sources, having one or more output ports. A signal source could be a signal generator, a file reader, a measuring device like a volt-meter, a thermal sensor etc.
- Mathematical operators, having at least one input and one output port, e.g. digital filter, correlation / coherence, average, threshold-level, integrator, FFT-transformation
- Signal sinks (targets) with one or more input ports. Possible sinks include sound-output, graphical output, device control etc.

Although visual design is common in various areas of research, it is a new strategy for prototyping Assistive Technologies. The following sections will give an overview of existing solutions for graphical modeling, with special focus on features which are important for the AsTeRICS project such as user interaction capabilities or support of embedded runtime environments.

### 7.3.1 Simulink (MathWorks)

Simulink is a software application that allows graphical modelling, simulation and analysis of dynamic systems like electrical circuits, mechanical setups or thermodynamic systems. Simulink is part of the Matlab software package by The MathWorks, Inc., and integrates seamlessly with software modules written in the dedicated Matlab programming language. Matlab/Simulink features a rich set of mathematical and computational functions and block-sets for specific areas of engineering and research and has therefore become a de facto standard for modelling and simulation in the scientific community. Commonly used computational blocks are: integration and derivation, filters and transfer functions, matrix operations, trigonometric functions, logic and discrete operations or polynomial functions [155].

To create a simulation of a dynamic system using Simulink, the user first draws a block diagram using the Simulink model editor. This block diagram graphically depicts the time-dependent mathematical relationship among the system's inputs, processors and outputs (see Figure 121). After specifying the desired start and end times and starting the model simulation, results can be shown in an oscilloscope and can be exported to a file or to the Matlab workspace [156].

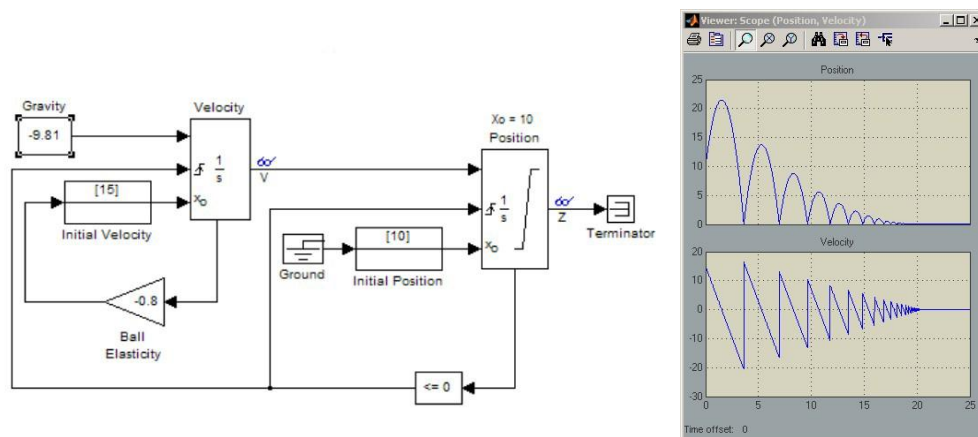


Figure 121: Simulink graphical editor and oscilloscope window [155]

The individual functional blocks are chosen from a list of block-sets. Connections between blocks are drawn with the mouse. Each block has a list of properties which can be set by double-clicking the block. A number of blocks can be combined to a single block to lower the complexity of the model view.

The *Signal Routing* block-set features some useful elements to interconnect functional blocks and to handle the data flow among them. Some of the most interesting routing features are:

- busses, which contain numerous individual signals
- data stores to temporarily buffer values
- multiplexers to group scalar signals into vectors signals

- demultiplexers to split vectors into smaller vectors or scalars
- selectors to pick signals from a bus or vector
- signal switches to select one of multiple inputs

The *Signal Processing* block-set features functional blocks for sampling continuous signals and processing of linear, time-invariant systems (LTI systems). Among those functionalities are:

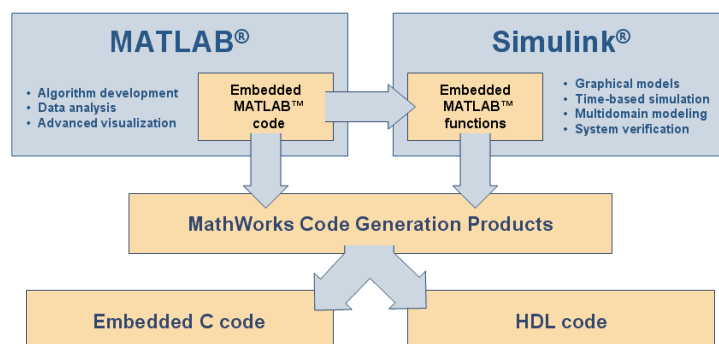
- quantizers, Up- and Downsampling, Interpolation
- adaptive and multirate filters, convolution
- FFT and wavelet transformation
- windowing functions
- peak finder, repeat and delay
- correlation and autocorrelation
- statistical blocks like min, max, median, RMS, variance, standard deviation, detrend

Such functions are of course essential also for the AsTeRICS project, where features have to be extracted from raw live data streams acquired from a biosignal amplification device (e.g. Enobio) or analogue sensors (accelerometer, bend sensor etc.)

### Simulink and embedded platforms

Matlab/Simulink is available for Microsoft Windows, Mac OS and Linux operating systems, but needs the processing power of a recent X86 architecture. A Simulink model could be exported to a smaller embedded platform only with major performance decline.

However, Matlab provides a system extension called “Embedded Matlab” which allows transferring a reduced set of Matlab operations directly into C-Code which could be cross-compiled for an embedded platform. Using the Matlab “RealTime Workshop”, and the Matlab Code Generation Products, C-code can be generated from Simulink models containing Matlab Code. Furthermore, the Simulink “HDL-Coder” can generate synthesisable Verilog and VHDL code from Simulink models for FPGA prototyping or ASIC hardware implementations (see Figure 122):



**Figure 122: Interfacing Matlab/Simulink with embedded systems [157]**

### 7.3.2 LabVIEW (National Instruments)

The National Instruments LabVIEW suite is one of the market leaders in software for test, measurement, and control applications. The focus of LabVIEW is rapid application development and cost-efficiency in interfacing with measurement and control hardware. LabVIEW is well suited for measurement and control in industrial environments, development of autonomous vehicles, medical data acquisition and generally for systems which include physical measurements, analysis and signal processing, decision making, data logging and report generation [158].

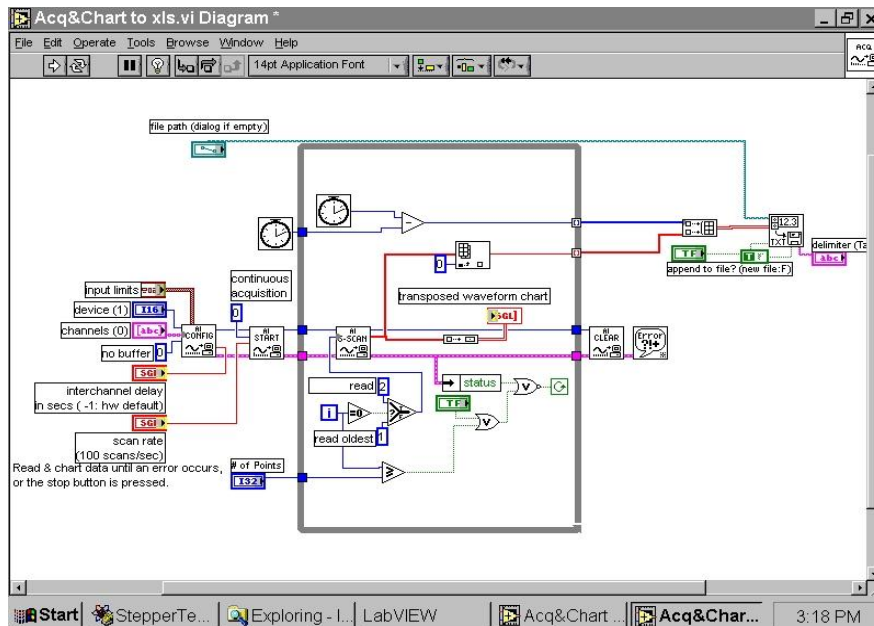


Figure 123: LabVIEW VI editor for system design [159]

For defining application and signal processing functionalities, LabVIEW uses a graphical programming language to define flowcharts in the so-called VI editor (see Figure 123). Palettes contain the functional blocks which can be inserted into the VI block diagram. These include code blocks like while loops, conditional blocks, mathematical blocks and user interface blocks like a customisable oscilloscope chart, switches, sliders, buttons and knobs for user input, boolean button displays, meters and many more. Parameters and characteristics of the functional block can be set in configuration dialogs [158].

Compared to Matlab/Simulink, LabVIEW is less focussed on mathematical analysis and scientific mathematics, but features a rich set of hardware support and user interfacing capabilities.

#### LabView and integration of hardware for measurement and control

Thousands of measurement and control devices can be interfaced with LabVIEW, including the numerous data acquisition hardware devices produced by National Instruments. One example is the USB based NI CompactDAQ module featuring numerous analogue and digital IOs and integrated sensors, e.g. for temperature measurement. Furthermore, there are software modules to interface custom embedded hardware, and modules for camera integration and image processing, report generation and database integration. Supported operating systems include Windows, Mac OS-X, Linux and embedded real-time OS.

## LabView and Embedded platforms

National Instruments provides a complete development solution for the graphical system design of embedded applications targeted to a variety of architectures ranging from off-the-shelf real-time and field-programmable gate array (FPGA)-based systems to custom microprocessor and microcontroller devices.

The LabView modules for embedded design include:

- The FPGA Module to configure FPGAs on NI hardware
- The Microprocessor SDK to integrate native LabVIEW analysis functions with pre-existing C code and I/O drivers on 32-bit microprocessors
- The DSP Module to use graphical design tools based on LabVIEW to create embedded applications that run on select digital signal processors (DSPs)
- The NI LabVIEW Embedded Module for ADI Blackfin Processors to integrate LabVIEW Embedded technology and ADI VisualDSP++
- The Embedded Module for 32-bit RISC ARM microcontrollers, which are widely used across many embedded designs.

The following Figure 124 shows the two available ARM32- based hardware platforms for the LabVIEW Embedded Module [161]:

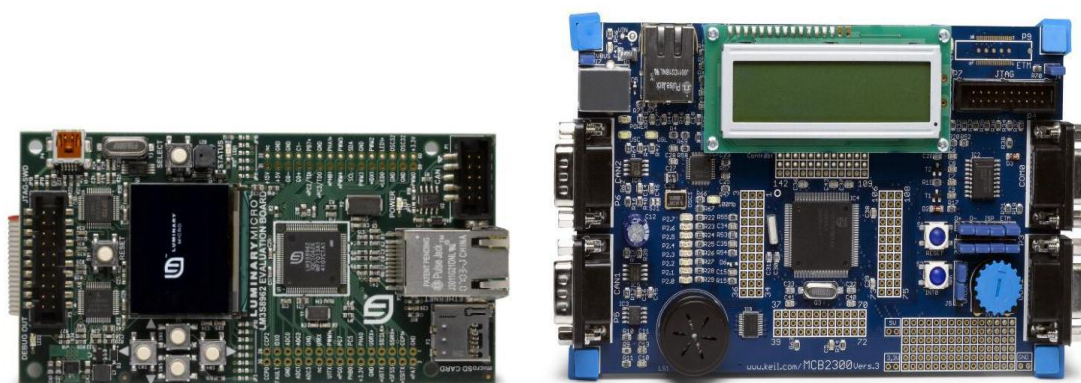


Figure 124: LabVIEW Arm Cortex-M3 (Luminary Micro) and Arm7 (NXP) prototyping boards

### 7.3.3 BioExplorer (CyberEvolution)

BioExplorer by CyberEvolution is a software application for real-time biophysical data acquisition, processing, and display which runs on x86 PCs with Windows operating systems (Win2000 or above). BioExplorer is one of the few professional software applications for bio- and neurofeedback for personal use, entertainment, education, and experimentation. Its main focus is opto-acoustic feedback of online EEG and ECG measurements. Key features are interfacing to various biosignal acquisition hardware and low-cost EEG devices, online signal processing and monitoring of signal features, recording and playback of archive files and offline processing of biosignal data [162].

BioExplorer offers a small number of feature-rich signal processing elements, mostly dedicated to EEG and ECG analysis. The following table lists those elements and gives a short description of their purpose [163]:

Bioexplorer design element	Purpose
AND, OR, NOT	The output is the logical AND of two or more signals
Averager	Calculates the average value of a signal over a specified time period.
Bandpass Filter	Filters a signal, allowing frequency components that are between the low band edge and the high band edge to pass through, and attenuating components outside this range.
Bar Graph	Graphically indicates the current value of a signal.
Button	manually set the state of a signal.
Coherence	Calculates the coherence between two signals.
Correlation	Calculates the correlation between two signals.
Counter	Counts properties of a signal (rising edges, falling edges, time high or time low).
Difference	Outputs difference between the current sample and the previous sample
Dominant Frequency	Calculates the dominant frequency (the frequency component with the largest amplitude) of a signal.
Expression Evaluator	Uses a user-defined expression to calculate the output based on one or more signals, e.g. '( ln2 - ln1 ) / ( ln1 + ln2 )'.
Max, Min	Outputs the maximum or minimum value of a signal.
Mean Frequency bandwidth.	Calculates the mean frequency in a given
Meter	Indicates the current value of a signal.
Oscilloscope	Displays one or more signals over time.
Power Ratio	Calculates the ratio of the power in two frequency bands, or the percent of the total power contained in one frequency band of a signal.
Pulse Detector	Given an ECG signal, determines the current pulse rate.
Source	This object represents a channel on a device (e.g. BrainMaster channel 1, etc). Source objects are used to get data from a device into the Signal Diagram.
Spectrum Analyzer	Displays the spectrum of a signal.
Standard Deviation	Calculates the Standard Deviation and Mean of a signal
Threshold	This object compares a signal to a threshold.
Trend	Displays the mean value of one or more signals over a user specified time period (e.g. 10 minutes).

Table 42: Available instruments and processing elements for Bioexplorer designs

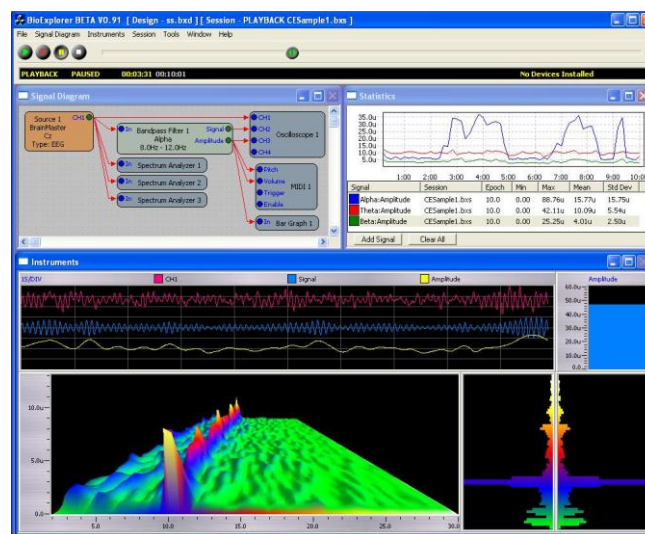


Figure 125: Bioexplorer design and monitoring windows

A Bioexplorer design consists of a signal diagram and a set of instruments. The design is graphically created by adding objects to the signal diagram window and wiring them together (see Figure 125).

Design objects can be configured by double clicking their icon in the signal diagram, which opens a dialog window with element properties. Additionally to the processing elements, Bioexplorer provides various elements targeted to realtime opto-acoustic feedback such as audio- and MIDI player elements, players for audio CDs, DVD movies or different movie files.

BioExplorer supports various biosignal acquisition hardware [162]:

Manufacturere	Biosignal Device
BrainMaster Technologies	BrainMaster
Brainquiry	PET – EEG, QPET
EEG Info	NeuroAmp
J&J	I-330-C2 Mini, I-330-C2, I-330-C2+
Jeff Carmen	Carmen HEG
Minder Labs	Pendant HEG, Pendant EEG, Pocket A1, Pocket A2, Pocket A3
MindTuner	X1-10
Mind Peak	WaveRider CEO, WaveRider jr, WaveRider Pro
Neurobit Systems	Neurobit Lite
OpenEEG	ModularEEG, MonolithEEG
QDS Systems Biosignal Research Tools	QDS NFK Access 2 EEG CH QDS NFK Excalibur II 4 EEG CH QDS NFK Quantum 8 EEG CH
Roshi Corporation	Roshi I, Roshi II, Roshi II+
TeleDiagnostic Systems	Alpha200, Alpha400

**Table 43: Bioexplorer supported hardware**

### 7.3.4 BioEra (Proatech LLC)

BioEra by Proatech LLC is a software application for signal analysis and presentation with focus on biofeedback. BioEra started as an open source project but has been commercialised after two years of development. The open source version (BioEra-Lite) is still available at the sourceforge project page [164]. A BioEra design contains a signal flow model from input to output. The signal flow is passed through visually edited objects (elements). Usually, BioEra is used together with a biofeedback device which sends digitised bioelectric signals (EEG, ECG or other biosignals) to the host computer via USB, serial port, wireless or other media. Supported hardware includes most of the devices listed in Table 43, and some recent low-cost EEG amplifiers as the OCZ NIA 2-channel EEG with dry electrodes [164].

BioEra is written nearly entirely in JAVA and theoretically works on any computing platform which can run a Java Virtual Machine (JVM) with JAVA versions higher than 1.4. BioEra-Lite has been shown to work on a PDA (Personal Digital Assistant) with limited resources. However, only Windows XP and Vista are officially supported in the commercial version of BioEra.

BioEra provides a designer window to edit the data-flow diagram and one or more runtime windows to present runtime results to the user. A runtime window can be in View or Edit

mode. The Edit mode allows chart layout edition, the View mode presents charts and interaction items at their final positions to the user.

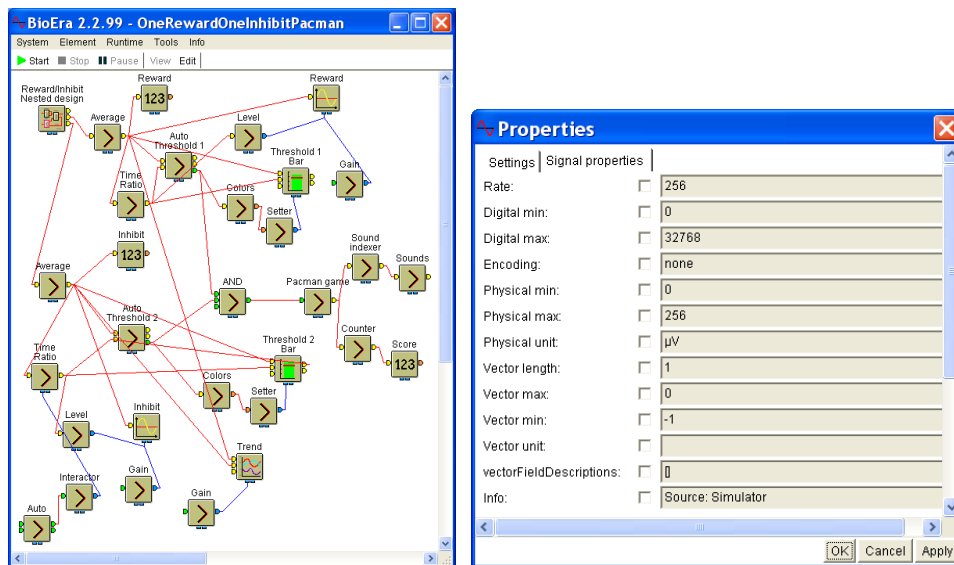


Figure 126: BioEra design window and element properties

## Design Window

Designs are created and edited using mouse and keyboard. Element properties can be adjusted via a pop-up menu. A design is stored in a single .bpd file (BioEra process design). An interesting feature is the “Nested Design”: Sub-Designs can be included into the current design using the NestedDesign element. This is very useful to keep the complexity of the current design within the desired level, and allows reuse of element clusters. Two dedicated elements (NestedInput and NestedOutput) define input- and output-connections of a nested design. These connections can be accessed at the nested design’s graphical icon (see Figure 126).

## Stream pipes and event pipes

The end points of connections between elements are called *pipes* in BioEra. There are many different pipe types for streamed signals (logical, scalar, float, complex, vector, matrix and object), where each input pipe has internal buffer. Object streams contain objects (e.g. texts) which are exchanged by reference (not by value like in other streams).

Each element in BioEra has special input- and output connectors for receiving or sending special events (see the blue connection in the design window, Figure 126). Events can be used to propagate single state changes or to set individual properties of elements which are not accessible through stream connectors [165].

## Signal parameters on element level

Information about signal parameters is provided on element level. Examples include sampling rate, digital or physical minimum and maximum. An element can retrieve these values from other elements connected to one of its inputs. An element can also modify its own values if necessarily. For example if an FFT transform is being done 4 times per second, then it means that the output rate is 4, and this value is returned when rate value is requested (see Figure 126).



BioEra can operate in frequency domain and in time domain. Several scales are available to designate the signals [165]:

- amplitude scales: microvolts ( $\mu\text{V}$ ) for EEG signals, millivolts (mV) for EMG, digital values (raw integer values from sampling device), balanced digital, percent
- time scales: milliseconds, seconds, minutes, hours, and days of relative (since the recording start) or absolute time; number of samples
- frequency domain scales: Hertz

### Processing order of elements and initialization order

Elements are initialised and processed in a defined order: elements that are closer to inputs (signal sources) are initialized and processed first. When new elements are inserted into the chain in the signal flow diagram, a sort is performed to maintain this processing order.

### User controls and system interactions

BioEra contains user control elements which let the user modify current processing settings (e.g. sliders, buttons, combo-boxes, menu-lists). The graphical widgets of those elements can be positioned freely in the runtime window.

*Interactor* elements can be used to send events to the operating system or to manage the running design configuration, for example by

- receiving or generating keyboard activity
- receiving or generating mouse activity
- receiving or generating system activities (start/stop design, pause)
- sending events to single elements (activate, deactivate, process)
- loading or saving the current design
- executing external applications

## 7.3.5 BrainBay

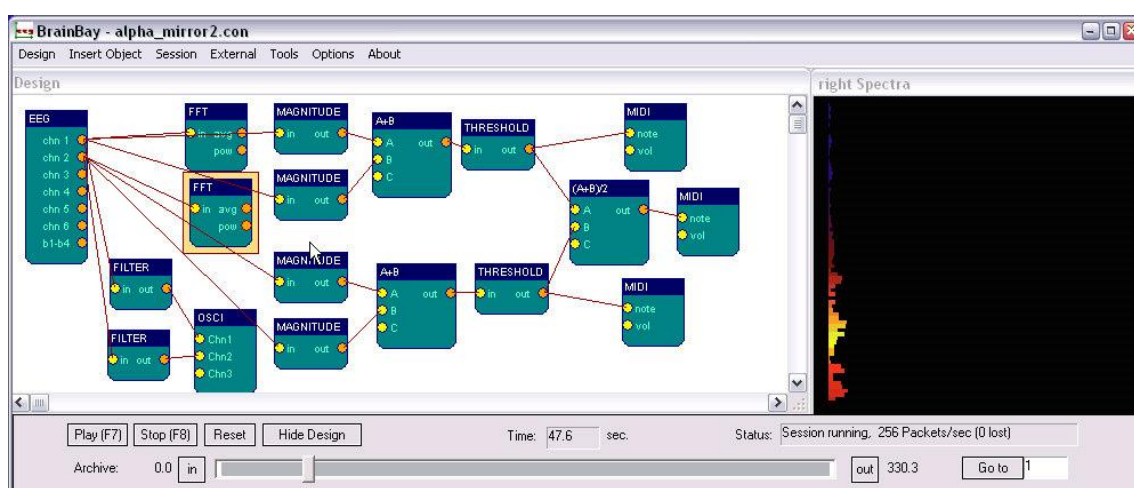


Figure 127: BrainBay design and runtime windows

BrainBay [166] is a Windows-based software application for graphically supported design of biosignal processing configurations, targeted to online biofeedback and to the development of alternative human computer interfaces (HCI). BrainBay provides several design elements

dedicated to assistive functionalities like face detection from live camera images, mouse-control or sending keyboard input to other running applications. BrainBay is written in C++, using the Win32 API, the source code is provided under a GPLv3 license. A designer manual describes the integration of new processing elements, making the system open to future extensions like novel output elements for device control or interfaces to other assistive technologies (see Figure 127).

BrainBay allows adjustment of element-parameters via respective dialog windows during system operation, the effects of parameter changes can be evaluated in the running configuration. This is very useful compared to other approaches, where the running design has to be stopped before changes can be applied.

On the other hand, signal properties and signal propagation of signal values are only rudimentarily implemented in BrainBay: In a BrainBay configuration, the sampling rate is defined globally and all channels between output- and input ports exchange floating point numbers at this sampling frequency. A system functionality to transfer single events is missing, as the possibility to access individual properties of elements aside from the input ports.

### Webcam interface with feature detection

The BrainBay camera element (see Figure 128) allows analysis of live webcam images using Intel's OpenCV - library for Computer Vision [167]. The camera element provides four output ports that deliver biometric data of a user's face detected in the camera's field of view: the X/Y – coordinates of the the nose and the X/Y coordinates of the chin. This functionality is based upon two algorithms:

- A Haar-Classifer Cascade [167] that was trained with frontal faces is used for face-detection. When a face could be detected, the positions of the nose and of the chin are estimated (regions-of-interest, ROIs).
- Motion-Tracking of the ROIs: the locations of nose and chin are tracked using a Lukas-Kanade Optical Flow algorithm [166]. This algorithm is a fast method for calculating the movement of points in two consecutive image frames. Re-initialisation of the face position is performed if confidence levels are exceeded. The relative movement of nose and chin are presented to the camera element's output ports and can be processed in the signal path as any other signal.



Figure 128: BrainBay camera element properties, live camera view with feature detection

## Camera-Mouse configuration

If the camera-element and the mouse-control element are connected via appropriate signal transformations, a functional camera-mouse system can be configured. A possible wiring for a camera-mouse setup (see Figure 129) with support for left and right-clicking is described in [168].

The movement of the nose can be directly mapped to x/y movement of the mouse cursor. If tremor reduction is desired, appropriate averaging elements are added to the signal path. If absolute positioning is preferred over a relative positioning of the mouse (“joystick-mode”), a signal integrator can be added to the signal path.

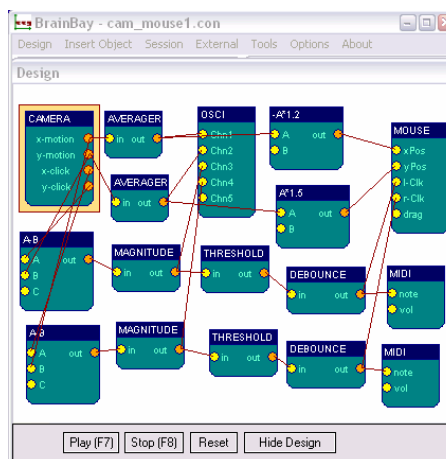


Figure 129: BrainBay camera mouse design

For extracting voluntary mouse clicks, rapid movement of the chin is used. The chin-position is subtracted from nose-position to get the relative chin-movement. Then a high-pass filter is applied on the x and y axis to detect a rapid movement. The magnitude of these signals is utilised as control parameter for the left and right-clicking operations: as soon as a threshold is reached, a voluntary movement is assumed and a mouse click is performed.

## Custom User Interaction widgets

The adjustment of essential configuration parameters like speed or sensitivity to motion by the user is of particular importance for AT-configurations. BrainBay provides a special dialog-element for customized user interaction widgets: a widget is defined by specifying active areas of a custom bitmap-image, which can be mapped to button or slider functions. For every button or slider, the dialog element gets a new output port where the current value (on/off, slider position) is available for other elements. At runtime, the widget is presented as skinned dialog on the desktop where it can be positioned freely. Interaction with the widget changes the respective values on the dialog element’s output ports.

Figure 130 shows a running camera-mouse configuration, which allows adjustment of mouse acceleration and selection of click-type and dwelling mode for mouse clicks via a customized user interaction widget. Here, the user operates the “Autonomy”- application by FORTEC for environmental control via the camera mouse [169].

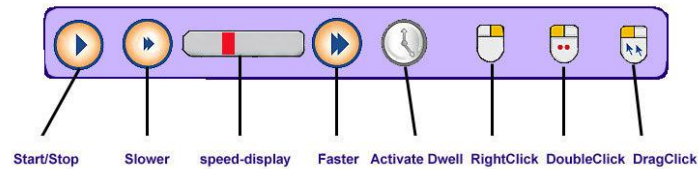


Figure 130: Using the Autonomy environmental control system via BrainBay camera mouse

### Keyboard interaction and word prediction

To enable keyboard interaction for disabled users, the *keystrike*- and *speller* elements have been added to BrainBay. The *keystrike* element can send a number of selectable key codes to the application which currently has the input focus. A list of keycodes can be selected and will be generated when the element receives a trigger-signal on its input port. Using eight *keystrike*-elements, a camera-element, and appropriate elements for signal conditioning, a camera-driven input system for cursor key-control can be implemented in just a few minutes. Such a setup can be used to control a computer game via head movements (up/down/left/right mapped to the four cursor keys)

To provide a more convenient way to write letters, words or sentences, the *speller*-element features different selection modalities and word prediction. The *speller* element has a runtime window where the current options and selection (highlighted) are presented. The inputs of the *speller* element allow direct selection (“trackball” or “joystick”-mode) and also switching to the next selection by a triggered stimulus (threshold of an input signal) or by time (automatic switching). A dictionary of commonly used words and phrases can be imported from a text file. Figure 131 shows the *speller* element’s properties and the corresponding runtime windows (the letters are ordered according to their probability for faster selection.)

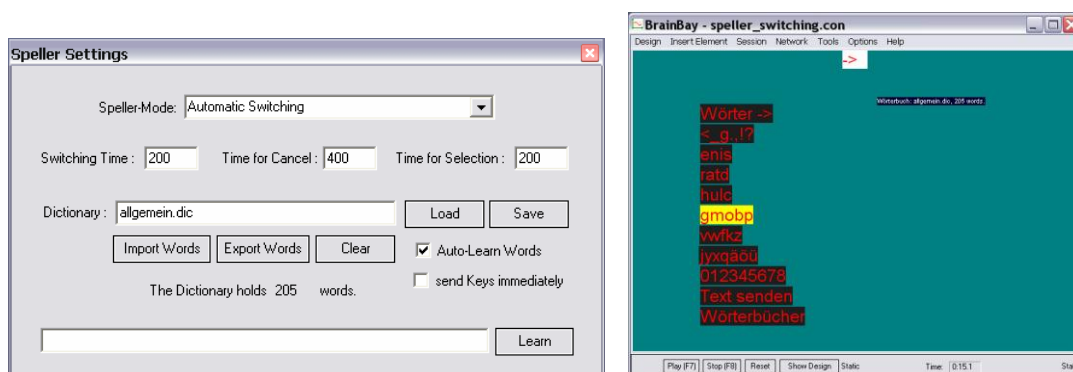


Figure 131: BrainBay speller and word prediction elements

### 7.3.6 OpenVibe

OpenViBE is a free and open source visual design software for Brain Computer Interface applications (BCIs). The whole software has been developed in C++ and released under the terms of the LGPL-v2+ license. OpenVibe consists of a set of software modules and box algorithms that can be integrated easily using the graphical design editor which allows to design BCI applications such as for Virtual Reality (VR) interaction [170].

#### Key features of OpenVibe are: [196]

- Software modules (“box-algorithms”) dedicated to the acquisition, pre-processing, processing and visualization of cerebral data, as well as to the interaction with VR displays can be combined via a user-friendly graphical language
- Designed for different types of users: BCI- and VR-developers, clinicians, researchers, programmers and end-users
- OpenVibe operates independently from the different software and hardware targets, includes an abstraction allowing to run with various acquisition machines and is based on free and portable software (e.g., GTK+, IT++, GSL, VRPN, GCC)
- Signals of different types (stream, event, electrode localisation)
- 2D- and 3D-topographical mapping (see Figure 132)
- Box-algorithms for existing BCI-paradigms (P300 speller, movement imagination)

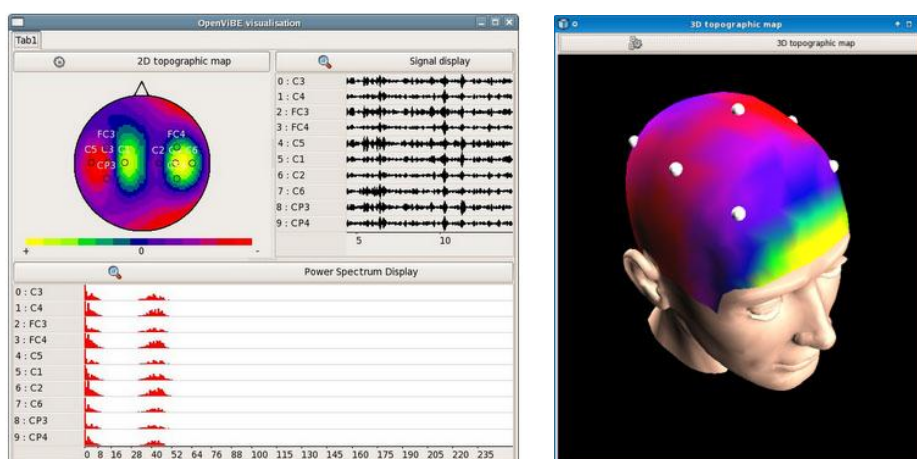


Figure 132: 2D- and 3D-topographical mapping of EEG data using OpenVibe [171]

#### OpenViBE functionalities [170]:

- File reading and writing: Generic files (r/w), GDF files (r/w), BCI Competition IIIb (r)
- Pre-processing: Time based epoching, Stimulation based epoching, Moving average, Windowing, Channel selection, Simple DSP, Temporal filter, Downsampling
- Feature extraction: Spectral analysis, Spatial filter
- Classification: Linear Discriminant Analysis
- Stimulation presentation: Graz-like motor imagery presentation
- Visualisation: Simple levels, Raw signal, Spectrum, Time/Frequency analysis, 2D/3D topography (see Figure 132)
- Connection to external software: VRPN integration (Virtual Reality Peripheral Network), TurboFieldTrip integration (r/w, in progress)

### Preconfigured scenarios and existing BCI-applications [170]:

- BCI based on motor imagery
- Real-time visualization of brain activity in 2D or 3D, Neurofeedback
- P300 speller (see Figure 133)

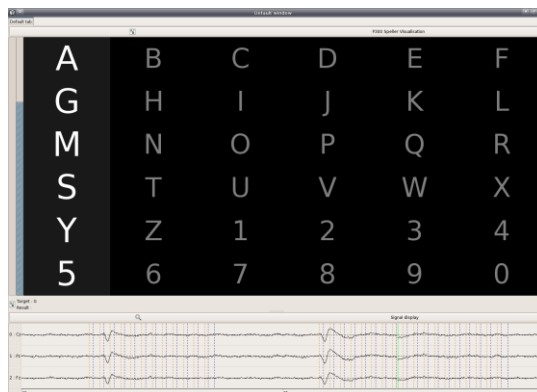


Figure 133: P300 BCI application designed with OpenVibe [170]

### 7.3.7 Summary and Comparison

Table 44 shows a comparison of different tools for system modelling and signal processing.

Product	Short Description / main features	Licences / Price	Embedded support	AT support
Matlab/Simulink	Modelling of control systems, signal analysis	€ 6000 (ML+SL+ signal processing toolbox)	C-code and VHDL generation, filter implementaiton for FPGAs	None
LabView	Measurement and control, interfacing data acquisition hardware	US\$ 4700 (DevSuite incl. signal-processing Toolkit)	Dedicated evaluation platforms, support of Arm-boards	None
BioExplorer	Biofeedback (EEG, EMG, ECG, ..)	US\$ 350	None	None
BioEra	Biofeedback (EEG, EMG, ECG, ..)	US\$ 250	PDA (not officially supported)	Keyboard-Interactor, Mouse Interactor
BrainBay	Biofeedback and Human Computer Interaction	GPLv3 (sourcecode available)	None	Webcam-Interface, Mouse-control, Keyboard Interactor, speller with word prediction
OpenVibe	Visual design framework for BCI applications and VR integration	LGPLv2+ (sourcecode available)	None	P300 speller, mu-rhythm detection, classifier training

Table 44: Comparison of visual tools for system modelling and signal processing

## 8 Relevant European Projects

In this section we identify relevant European projects in the fields of Assistive Technology and Brain/Neuronal Computer Interaction (BNCI). Our goal is to minimize overlapping and re-use common knowledge and technology, if and whenever is possible. Since there are several EU projects in the area of BNCI and Assistive Technology we have identified the latest and most relevant European projects.

### 8.1 TOBI: Tools for Brain Computer Interaction<sup>197</sup>

TOBI is a large European Integrated Project that started in November 2008. Main objective of the project is to develop practical technology in the field of BNCI and improve the quality of life of disabled people. TOBI focuses mainly on the design of non-invasive BNCI prototypes that combine existing assistive technologies and rehabilitation protocols. By integrating different BNCI assistive technologies, the consortium aims to improve the quality of life of people with motor disabilities in four application areas:

a) Communication & Control

Much like AsTeRICS, the goal in this area is to improve people's communication potential and opportunities by providing access to a variety of devices such as virtual keyboards, internet, email, telephony, fax, SMS and environmental control.

b) Motor Substitution

Restore of lost motor functions is a priority for people with physical disabilities. TOBI works on ways for improving grasping and reaching functionality of patients.

c) Entertainment

The goal is to give patients better access to ambient living devices, gadgets and social networking applications or BNCI-games.

d) Motor Recovery

BNCI technology can play a potential rehabilitation role in enhancing motor function recovery after a cerebrovascular accident. TOBI technology aims to support the mental practice of motor actions via BNCI training which might boost the clinical rehabilitation and possibly lead to a better functional outcome.

We believe that AsTeRICS and TOBI share similar goals, especially on the BNCI part of our project. We aim to achieve brain computer interaction through the Enobio biosignal unit, supplied by our partner STARLAB. Enobio is a wearable, modular and wireless electro-physiology sensor system for the recording of electroencephalography, electrocardiography and electrooculography signals. Our goal is to coordinate our work in BNCI with prototypes and technology advances developed within the TOBI project.

---

<sup>197</sup> TOBI: Tools for Brain Computer Interaction, <http://www.tobi-project.org>

## 8.2 BRAIN: Brain-computer interfaces with Rapid Automated Interfaces for Nonexperts<sup>198</sup>

Main objective of BRAIN is to enhance intercommunication and interaction skills of disabled people through the development and integration of Brain-Computer Interfaces into practical assistive tools. The goal is to improve user interaction with other people, home appliances, assistive devices, personal computers, internet technologies, etc. BRAIN is specialised on BNCI based user interaction and aims to augment its reliability, flexibility, usability and accessibility by improving the main components of BNCI systems: signal acquisition, operating protocol, signal translation and application.

Among others, the main contributions in the area of BNCI targeted by the BRAIN project are:

- a) A user-friendly, easy to use BNCI wizard that will walk each user through a series of tests to determine optimal parameters. Tests series will evaluate crucial parameters across three different BNCI approaches and a system expert based on the tests results will determine the best BNCI approach and parameters for each user.
- b) A strong user focus: high priority in BRAIN is given to the user needs and therefore different user groups have been identified for requirements elicitation and tests. The goal is to make real world BNCI systems more flexible, usable, reliable and accessible.
- c) An intuitive graphical user interface (IGUI) that allows users to visually interact with the services provided by the BRAIN application. IGUI visually displays to the user available actions at any given time.
- d) A universal interface (UAI), which aims to enable end-users to interact with different applications through the IGUI. Due to the limited interaction that the BNCI allows, all applications have to be designed as a collection of simple commands, represented as icons in the IGUI.
- e) The use of Steady State Visual Evoked Potential (SSVEP)-based BNCIs, which offer higher information throughput and require shorter calibration periods when compared to other BNCI modalities.
- f) A new type of electrodes based on water in combination with a high performance amplifier front end, all of which perform as good as currently used electrode technologies but without their disadvantages.

We identified common objectives between AsTeRICS and BRAIN. The user friendly and easy to use Assistive Technology as well as the strong user focus and integration of several assistive devices are only a few. Although BRAIN is mainly focused on BNCI technologies we aim to a close collaboration between the two projects to share knowledge in BNCI and other relevant aspects.

---

<sup>198</sup> BRAIN, <https://www.brain-project.org/>



### 8.3 BrainAble: Autonomy and social inclusion through mixed reality Brain-Computer Interfaces: Connecting the disabled to their physical and social world<sup>199</sup>

BrainAble is a European collaborative project started in January, 2010. Its main objective is to improve the quality of life of people with disabilities by overcoming exclusion from home and social activities. They aim to provide an ICT-based human computer interface (HCI) composed of BNCI sensors combined with affective computing and virtual environments.

Human – Computer interaction will be enhanced with new tools that enable people to better control their environment and gain functional independence and autonomy during their daily life activities. For this purpose, electroencephalography, electrooculography, electromyography, and electrocardiography will be used. Furthermore, BNCI information will be used for indirect user interaction, such as by changing interface or overall system parameters based on measures of boredom, confusion, frustration, or information overload. Such self-adaptive tools will increase effective communication bandwidth (in a BNCI sense) since users will be able to use several sensors to control their environment and adaptation will reduce errors. All of these help provide the user with the desired control capability.

Finally, the developed HCI will be complemented by an intelligent user interface based on virtual reality that will help disabled people to move, interact with various devices, create self-expression assets using music, pictures and text, communicate with other people both online and offline, play games to counteract cognitive decline, and train new skills and task realization.

Much like AsTeRICS, BrainAble aims to produce a set of technologies suitable for assisting people with physical disabilities regardless of cause. Being specialized on human-machine interfaces for the disabled, BrainAble can provide valuable information on interfacing with ambient devices in living environments as well as outer virtual environments, such as social network services. We aim to investigate the possibility of reusing HCI knowledge developed during the BrainAble project, whenever possible..

### 8.4 DECODER<sup>200</sup>

DECODER is a European collaborative project (started in February, 2010) that aims to use brain-computer-interfaces for detecting consciousness in non-responsive patients. This project is specialized on patients that are only little or not at all able to interact with their environment. The goal is to provide access to modern information and communication technology, such as internet, personal computer or home appliances when only a single response of a person is available.

This highly specialised project aims to integrate state of the art BNCIs and assistive technology applications and adapt them to the needs of non-responsive patients. In particular, existing AT applications will be adapted to single-switch devices to be interfaced through BNCI techniques with non-responsive patients.

---

<sup>199</sup> BrainAble, <http://www.brainable.org/>

<sup>200</sup> DECODER, <http://www.decoderproject.eu/>

Although DECODER targets a very specific domain of BNCI for non-responsive patients, research results from DECODER can be very useful when developing the integration of AT devices to the AsTeRICS platform as well as the brain-computer interfacing through the ENOBIO device.

## 8.5 MUNDUS: Multimodal Neuroprosthesis for Daily Upper limb Support<sup>201</sup>

MUNDUS is an EU small or medium scale focused research project started on March, 2010. The project aims to provide an assistive framework for recovering direct interaction capability of severely motor impaired people based on arm reaching and hand function. The goal is to integrate sensors, actuators and control solutions, adaptable to the level of severity or progression of the disease. The framework is intended for people affected by neurodegenerative disease such as Amyotrophic Lateral Sclerosis (ALS), Friedreich Ataxia (FRDA), Multiple Sclerosis (MS) and high level Spinal Cord Injury (SCI). The main parts of the MUNDUS framework are:

- a) The adaptable and modular facilitator which enables fast adjustment to new situations according to the progression of the disease.
- b) The MUNDUS controller which integrates several types of information collected by electromyography, bioimpedance, head/eye tracking and eventually brain computer interface commands.
- c) The actuators which modularly combine a lightweight, non-cumbersome exoskeleton (for compensating arm weight), a biomimetic wearable neuroprosthesis (for arm motion) and small and lightweight mechanisms to assist the grasp of collaborative functional objects identified by radio frequency identification.

Although MUNDUS targets a single medical / rehabilitation device for recovering arm reaching and hand function the idea of assistive technology, tailored to the specific capabilities of the single user matches the requirement of AsTeRICS for developing a construction set adaptable to the motor abilities of end-users, therefore, we aim to investigate the possibility of MUNDUS and AsTeRICS platforms integration.

## 8.6 Future BNCI<sup>202</sup>

Future BNCI is a European project of the seventh framework, started in January, 2010. The goal is to help BNCI research community and explore future directions in the emerging class of the so called BNCI systems. Through a series of dissemination activities, Future BNCI aims to provide information for the BNCI community and explore the best future research opportunities in different aspects of BNCI research. The expected results are detailed reports on promising technologies related to sensors and signals, signal processing, applications and devices and interfaces and operating environments. The project also aims to coordinate the many new research efforts (academic and industrial) and clarify terms and definitions which in some cases are confusing, out-dated, non-existent, and/or only sporadically

---

<sup>201</sup> MUNDUS, <http://www.mundus-project.eu/>

<sup>202</sup> Future BNCI, <http://future-bnci.org>

accepted by groups in the area. Finally, Future BNCI aims to identify possible collaborations between different players in the area of BNCI.

AsTeRICS aims to keep constant communication with Future BNCI especially through our common partner, Starlab.

## 8.7 COGAIN: Communication by Gaze Interaction<sup>203</sup>

COGAIN is a “Networks of Excellence” European project under the sixth framework programme, completed in August, 2009. COGAIN continues its work in the form of an association. The goal of the project is to give access to seriously disabled people to affordable tools that allow them to communicate using their eye movements. COGAIN gathers Europe’s expertise in eye tracking integration with computers from the perspective of assistive technologies for citizens with motor impairments. The main objectives of the network are to develop new technologies and systems, improve existing gaze-based interaction techniques, and facilitate the implementation of systems for everyday communication. The researches aim to provide the most disabled members of society eye-gaze based communication and command tools so they can control computers, devices, and wheelchairs or access the internet. Since the project kick off some promising eye-tracking technologies have emerged. Users are now able to create texts by eye-typing, write and send emails or instant messages, play games and follow news and sports.

Two of the technologies pioneered under the COGAIN umbrella are *Dasher* and *GazeTalk*. The former is an information-efficient communications system driven by continuous pointing gestures while *GazeTalk* is a multilingual, predictive text entry system.

The knowledge pool maintained by COGAIN is a very useful source for the Smart Vision Module (SVM) we plan to build in AsTeRICS. We aim to take into consideration technologies and advances developed during this and other eye-tracking related projects while we design and develop the AsTeRICS smart vision module.

## 8.8 I2HOME: Intuitive interaction for everyone with home appliances based on industry standards<sup>204</sup>

I2HOME is an EU funded project under the sixth framework programme, completed in December, 2009. The objective of the project was to make devices and appliances in the home environment more accessible to persons with mild cognitive disabilities and older persons, using a new mainstream user interface standard, called Universal Remote Console (URC) standard. The project team also focused on making standardised and developed access strategies applicable to areas outside the home.

In the universal control hub architecture proposed, a hub component provides a connection point between end user devices and home appliances. This enables anybody to write user interfaces (according to the user needs) for the control of networked home appliances from different vendors.

Similar to the approach we plan to follow in AsTeRICS, I2HOME partners used a middleware-based solution for enabling various gadgets and appliances, including air

---

<sup>203</sup> COGAIN, <http://www.cogain.org>

<sup>204</sup> I2HOME, <http://www.i2home.org/>

conditioning and heating, to be controlled by the interface a user has selected. It was estimated that by the beginning of 2010 more than 100 organizations and companies in Europe were going to use or work with I2HOME technology.

In AsTeRICS we foresee a possible collaboration and knowledge exchange between us and the I2HOME project team. We have already made initial contacts with members of the I2HOME team and we will investigate the possibility of a PC running an URC remote control together with AsTeRICS platform in a later stage.

## 9 Conclusions

Throughout this document, an extensive overview of the state-of-the art in technology relevant for the AsTeRICS project has been given.

In the first section, AT input devices, software solutions and techniques with special importance for our project have been shown, which have been selected from a huge variety of assistive technologies on hand today. Many different sensors and actuators are available, but some of them are immoderately expensive due to a small market niche of very specialised AT-products, which makes a high margin for the manufacturers a necessity – not for the benefit of the end user.

Concerning interoperability, AT-device connectors or transfer protocols used by AT devices are often kept proprietary, which makes concurrent use of different devices hard or impossible. Open Source products in the AT scene are still rare and differ in quality. Up to now, the AT-landscape does not offer an open standard for interfacing and combining different sensors and actuators, which would allow flexible integration of AT hardware and software for an efficient design of user-driven custom solutions.

New technological approaches like computer vision, Brain and Neural Computer Interfaces and smart consumer- or entertainment devices have interesting capabilities for people with severe motor disabilities. The creative use of these techniques and devices and their combination with existing AT products could make user interfaces for many people with disabilities more appealing and more efficient.

The market for embedded computing platforms is also a very heterogeneous one, with a large amount of different devices, prototyping boards and design solutions. Product prices in this sector differ significantly, depending mostly on the product volumes. A commercial off-the-shelf computing platform as for example a netbook is available at lower cost as a highly specialised embedded board with comparatively less memory or computing power. For the AsTeRICS hardware platform, low cost and high performance computing platforms with small size and low power consumption have been evaluated. The recent Atom based embedded computing boards with picoITX form factor are interesting candidates for the AsTeRICS embedded platform, offering good performance at low power consumption and small size. For the first prototype, the development partner of the consortium will start to evaluate these boards in course of WP3 and WP4

Recent products in the sector of convertible netbooks and tablet-PCs offer interesting assistive functions and will come into the project focus especially for computer vision tasks, speech processing or gesture-based interaction via a big touchscreen.

Browsing today's manifold solutions for operating systems, we find a palette ranging from tiny, highly specialised real time kernels to fully-featured multi-purpose operating systems including ample device support and graphical desktops. Microsoft Windows (XP and Windows 7 respectively) can be considered as the most relevant operating system for Assistive software applications. Using an embedded board with Atom CPU / X86 architecture offers the possibility to run a standard Windows OS on the platform, featuring latest JAVA technology and good support for USB device integration.

The state of the art in software architecture provides service oriented and component oriented design patterns. These architectures allow a very flexible and structured development of even complex software systems. Well developed and tested frameworks implementing these techniques reduce the development times and the risk of design failures. JAVA/OSGI offers a platform independent solution to deploy service plugins during runtime, which can ease the task of implementing the toolkit for a flexible AT construction set according to an open specification.

Furthermore, we could approve the planned approach to create a graphical construction and configuration tool for AT functionalities based upon interconnectable plugins, considering established software solutions for signal processing as Matlab/Simulink, LabView or BioExplorer.

This state of the art analysis gives a profound basis for the design- and implementation decisions which have to be taken by the consortium in course of WP2, WP3, WP4 and WP6. It further amplifies the arguments for a flexible construction set for rapid integration of user-tailored AT solutions, and strengthens our belief that we will be able to contribute an open specification and a concrete implementation thereof in the sector of AT integration, which can be beneficial for both, AT users and AT developers.

In the next steps, this document will, together with the outcome of the user requirement analysis (D1.1) lay the basis for the consortium to define the System Specification and Architecture, which will be exposed in the D2.1 deliverable.

## References

- 1 Pfurtscheller, G., Müller, G., Neuper, C., Korisek, G.: Control of a hand orthosis by EEG vs. EMG in a patient with lesion at level C5/C6. *European Journal of Trauma - E-Supplement* 1/2002.
- 2 Lucas, L., DiCicco, M., Matsuoka, Y.: An EMG-Controlled Hand Exoskeleton for Natural Pinching. *Journal of Robotics and Mechatronics* Vol.16 No.5, 2004
- 3 Nussbaum G., Veigl C., Miesenberger K. EMG Signal Controlled Pneumatic Gripper for Mouthsticks. 2009. In *Proceedings of Assistive Technology from Adapted Equipment to Inclusive Environments - AAATE 2009*
- 4 Gareth Evans, Paul Blenkhorn. A head operated joystick - experience with use.
- 5 Eom, G.-M., Kim, K.-S., Kim, C.-S., Lee, J., Chung, S.-C., Lee, B., Higa, H., Furuse, N., Futami, R., and Watanabe, T. Gyro-mouse for the disabled: 'click' and 'position' control of the mouse cursor. 2007. *International Journal of Control, Automation, and Systems*, 5(2). pages 147–154.
- 6 Capacitive facial movement detection for human–computer interaction to click by frowning and lifting eyebrows. VilleRantanen, Pekka-Henrik Niemenlehto , Jarmo Verho and Jukka Lekkala. *Medical and Biological Engineering and Computing*. Springer Berlin / Heidelberg. Volume 48, Number 1
- 7 Barreto AB, Scargle SD, Adjouadi M (2000) A practical EMG-based human–computer interface for users with motor disabilities. *J Rehabil Res Dev* 37(1):53–64
- 8 Partala T, Aula A, Surakka V (2001) Combined voluntary gaze direction and facial muscle activity as a new pointing technique. In: *Proceedings of IFIP INTERACT'01*
- 9 Chin CA, Barreto A, Cremades JG, Adjouadi M (2008) Integrated electromyogram and eye-gaze tracking cursor control system for computer users with motor disabilities. *J Rehabil Res Dev* 45(1):161–174
- 10 Surakka, V., Illi, M. & Isokoski, P. (2004) Gazing and frowning as a new human-computer interaction technique. *ACM Transactions on Applied Perception*, 1, 1, 40-56.
- 11 Neural Network Screening of Electromyographic Signals as the First Phase to Design Novel Human-Computer Interaction. Pekka-Henrik Niemenlehto, Martti Juhola, Veikko Surakka. *Studies in Health Technology and Informatics* Volume 116, 2005
- 12 Effects of positive and negative affect on electromyographic activity over zygomaticus major and corrugator supercilii. Jeff T. Larsen, Catherine J. Norris, John T. Cacioppo *Psychophysiology*, Vol. 40, No. 5. (September 2003), pp. 776-785.
- 13 The Corneo-Retinal Potential Difference as the Basis of the Galvanometric Method of Recording Eye Movements. O.H. Mowrer, R.C. Ruch, and N.E. Miller. *American Journal of Physiology*. 114,423. 1936.
- 14 Survey of Eyemovement Recording Methods. L.R. Young and D. Sheena. *Behavioral Research Methods and Instrumentation* 7( 5): 397-429. 1975.
- 15 Biosignal Processing in Virtual Reality. Lusted, H.S., Knapp, R.B., and Lloyd, A. *Third Annual Virtual Reality Conference*, San Jose, CA. 1992.
- 16 EOG-Controller for Rehabilitation Technology. E. Unger, M. Bijak, W. Mayr, C. Schmutterer and G. Schnetz. *The Fifth International Conference on Computers Helping People with Special Needs (ICCHP '96)*, Linz, Austria, July 1996. In J. Klaus; E. Auff; W. Kremser; W. Zagler (eds.) *Interdisciplinary Aspects on Computers Helping People with Special Needs..* Vienna: R. Oldenbourg. 401 – 408.
- 17 On building intelligence into eagleeyes. In: *Assistive technology and artificial intelligence*. Springer, Berlin
- 18 EOG-based Human–Computer Interface system development Lawrence Y. Deng, Chun-Liang Hsu, Tzu-Ching Lin, Jui-Sen Tuan and Shih-Ming Chang. (2009) *Expert Systems with Applications* Volume 37, Issue 4, April 2010, Pages 3337-3343
- 19 “Electrooculogram based System for Computer Control using A Multiple Feature Classification Model”, Armen R Kherlopian, Joseph P Gerrein, Minerva Yue, and Kristina E Kim. *Proceedings of the 28th IEEE EMBS Annual International Conference*, New York City, USA, August 30 – September 03, 2006, pp 1295-1298.
- 20 “A Human-Computer Interface (HCI) based on Electrooculogram (EOG) for Handicapped”, Batu Akam, and Ali Ozgur Argunsah, *IEEE 15th International Conference on Signal Processing and Communication Applications (SIU 2007)*, 11- 13 June, 2007, Eskisehir, pp 1-3.
- 21 EOG and EMG Based Virtual Keyboard: A Brain-Computer Interface Hari Singh Dhillon, Rajesh Singla, Navleen Singh Rekhi, Rameshwar Jha. *Computer Science and Information Technology, International Conference on In 2009 2nd IEEE International Conference on Computer Science and Information Technology*, Vol. 0 (August 2009), pp. 259-262.

- 22 Classification of electro-oculogram signals using artificial neural network Guven A., Kara S. (2006) Expert Systems with Applications, 31 (1), pp. 199-205.
- 23 Huckauf A, Goettel T, Heinbockel M, Urbina M (2005) What you don't look at is what you get: anti-saccades can reduce the midas touch-problem. In: Proceedings of the 2nd symposium on applied perception in graphics and visualization. p 170
- 24 Huckauf A, Urbina MH (2008) On object selection in gaze controlled environments. J Eye Mov Res 2(4):1-7
- 25 "An EEG-based brain-computer interface for cursor control". Jonathan R. Wolpaw, Dennis J. McFarland, Gregory W. Neat and Catherine A. Forneris Electroencephalography and Clinical Neurophysiology. Volume 78, Issue 3, March 1991, Pages 252-259
- 26 Design and operation of an EEG-based brain-computer interface with digital signal processing technology. McFarland, D.J., Todd Lefkowitz, A., Wolpaw, J.R. Behavior Research Methods, Instruments, and Computers Volume 29, Issue 3, 1997, Pages 337-345
- 27 Spatial filter selection for EEG-based communication. Dennis J. McFarland, Lynn M. McCane, Stephen V. David, Jonathan R. Wolpaw. electroencephalography and clinical Neurophysiology 103 (1997) 386-394
- 28 Sensorimotor Rhythm-Based Brain-Computer Interface (BCI): Feature Selection by Regression Improves Performance. Dennis J. McFarland and Jonathan R. Wolpaw. IEEE Transactions on neural systems and rehabilitation engineering, vol. 13 n. 3 , sept. 2005
- 29 "Service Robot System Based on Brain-computer Interface Technology" Li Zhao, Chuo Li, Shigang, Cui Tianjin.
- 30 Paul Nisbet, Patrick Poon, Special Access Technology, [http://callcentre.education.ed.ac.uk/About\\_CALL/Publications\\_CAA/Books\\_CAB/SAT\\_CAC/sat\\_cac.html](http://callcentre.education.ed.ac.uk/About_CALL/Publications_CAA/Books_CAB/SAT_CAC/sat_cac.html)
- 31 Sreedharan, S., Zurita, E. S., and Plimmer, B. 3D input for 3D worlds. 2007. In OZCHI '07: Proceedings of the 19th Australasian conference on Computer-Human Interaction, pages 227-230, Adelaide, Australia
- 32 Pensyl, W. R., Jernigan, D. K., Qui, T. C. T., Fang, H. P., and Ping, L. S. Large area robust hybrid tracking with life-size avatar in mixed reality environment: for cultural and historical installation. 2008. In VRCAI '08: Proceedings of The 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry, pages 1-5, Singapore.
- 33 Bernardes, J. a., Nakamura, R., Calife, D., Tokunaga, D., and Tori, R. Integrating the wii controller with engine: 3d interfaces extending the frontiers of a didactic game engine. 2009. Comput. Entertain., 7(1). pages 1-19.
- 34 Robert Jania:: Make the transition from 8- to 32-bit MCU design as painless as possible <http://www.embedded.com> (10/06/09)
- 35 Microchip, Inc.: Data Sheet of the PIC18F1XK50/PIC18LF1XK50 microcontrollers <http://ww1.microchip.com/downloads/en/DeviceDoc/41350C.pdf>
- 36 Microchip, Inc.: Low Pin Count USB Development Kit User's Guide <http://ww1.microchip.com/downloads/en/DeviceDoc/41356B.pdf>
- 37 Atmel AVR AT90USB64/128 Data sheet [http://www.atmel.com/dyn/resources/prod\\_documents/7593S.pdf](http://www.atmel.com/dyn/resources/prod_documents/7593S.pdf)
- 38 Atmel AVR AT90USBKey Hardware User Guide [http://www.atmel.com/dyn/resources/prod\\_documents/doc7627.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc7627.pdf)
- 39 Cypress Semiconductors: controllers for Full Speed USB peripherals <http://www.cypress.com/?id=168>
- 40 Cypress Semiconductors, CY7C63X data sheet <http://www.cypress.com/?docID=20512>
- 41 Digikey, CY3664 development kit: <http://search.digikey.com/scripts/DkSearch/dksus.dll?Cat=2621773&k=cy3664>
- 42 Gibson, J.J. "The Ecological Approach to Visual Perception", Lawrence Erlbaum Associates, 1979
- 43 Berthoz, A. "Le sens du mouvement", Odile Jacob, 1997.
- 44 Gorodnichy, D., Roth, G., Nouse "use your nose as a mouse" perceptual vision technology for handfree games and interfaces, Image and Vision Computing, 22(12), Oct. 2004, pp. 931-942.
- 45 Bülthoff, H. H., Ernst M. O. "Merging the senses into a robust percept", Trends in Cognitive Sciences, 8, 162-169, 2004.
- 46 Lee, M.W., Cohen, I. "Proposal maps driven MCMC for estimating human body pose in static images", CVPR, 2004



- 47 Felzenszwalb, P.F., Huttenlocher, D.P. "Pictorial structure for object recognition", *Int. J. of Computer Vision*, 61(1), pp. 55-79, 2005.
- 48 Mori, G., Ren, X., Efros, A.A., Malik, J. "Recovering human body configurations : Combining Segmentation and Recognition", *CVPR*, 2004.
- 49 Lee, M.W., Cohen, I. "Human upper body pose estimation in static images", *ECCV*, pp. II: 136-138, 2004.
- 50 Navaratnam, R., Thayananthan, A., Torr, P.H.S., Cipollan R. "Hierarchical Part-Based Human Body Pose Estimation", *ECCV*, 2004.
- 51 Siddiqui, M. Medioni, G. "Robust Real-time Upper Body Limb Detection and Tracking", 4th ACM Int. workshop on Video Surveillance and Sensor Networks, Santa Barbara, USA, pp.53-60, 2006.
- 52 Comaniciu, D., Ramesh, V., Meer, P. "Kernel-based Object Tracking", *IEEE PAMI*, 25(5), pp. 564-577, 2003.
- 53 Broekhuijsen, J., Poppe, R., Peol, M. "Estimating 2D Upper Body Poses from Monocular Images", *Int. J. of Computer Vision*, 54(1-3), pp.181-207, 2006.
- 54 Sidenbladh, H., Black, M. "Learning the statistics of people in images and video", *Int. J. of Computer Vision*, 54(1-3):181-207, 2003.
- 55 Gorodnichy, D., Roth, G., Nouse "use your nose as a mouse" perceptual vision technology for handfree games and interfaces, *Image and Vision Computing*, 22(12), Oct. 2004, pp. 931-942.
- 56 Dalka, P. & Czyzewski, A. "LipMouse: novel multimodal human-computer interaction interface", *International Conference on Computer Graphics (SIGGRAPH '09)*, No. 48, 2009.
- 57 Choi, W., Lee D. & Na, J. "Applicability of No-hands Computer Input Devices for the Certificates for Microsoft Office Software", 11th Int. Conf. on Computers Helping People with Special Needs (ICHP), pp. 1176-1183, 2008.
- 58 Mohamed, Wahdi, Abdul, Koggalage, R. "Control of Mouse Movements using Human Facial Features", 3<sup>rd</sup> Int. Conf. on Information and Automation for Sustainability (ICAISP'2007), Melbourne, Australia, pp.13-18, 2007.
- 59 Lyons, M., J., Chan Chi-Ho, Tetsutani, N. "Mouth Type : Text entre by Hand and Mouth", *Proc. CHI'04*, Vienna, Austria, pp. 1383-1386, 2004.
- 60 Marg, E. "Development of Electro-Oculagraphy : Standing Potential of the eye in Registration of Eye Movement", *American Medical Association*, 1951
- 61 Kaczmarek, R., Couty, J-P., Malvache, N. "EOG Signals Processing used in an Ocular Command System Dedicated to Disabled People", 2<sup>nd</sup> IFMBE-IMIA Int. Workshop on BioSignal Interpretation (BSI'96), Sept. 23-28, Kanagawa, Japan, 1996.
- 62 Kate J.H., Van der Meer, P.,M. "An Electro-ocular switch for communication of the speechless", *Medical Prog. Technologies*, 10 (3), pp. 135-141, 1984.
- 63 Hori, J., Sakano, K., Miyakawa, M. Saitoh, Y. "Eye movement communication control system based on EOG and voluntary eye blink", *ICHP 2006*, Linz, Austria, *Proc./LNCS* vol. 4061, pp. 950-953, 2006.
- 64 Torok, N., Guilleman, V., Barnothy, J.M. "Photoelectric nystagmography", *Annals of Octology, Rhinology and Laryngology*, 60:917, 1951.
- 65 Hirschberg, J. "Über die Messung des Schieldgrades und die Dosierung des Schieloperation, *Centralblatt fur Praktisch Augenheilkunde*", (8) 325-327, 1885.
- 66 Hansen, D.W., Qiang, J. "In the Eye of the Beholder: A Survey of Models for Eyes and Gaze", *IEEE PAMI*, 32(3), pp. 478-500, 2010.
- 67 Bonino, D., Castellina, E., Corno, F., Gale, A., Garbo, A., Purdy, K., Shi, F. "A Blueprint for Integrated Eye-controlled Environements", *Univ. Access Inf. Soc.*, vol. 8, pp. 311-321, 2009.
- 68 Young, D., Tunley, H., Samuels, R. "Specialised Hough Transform and Active Contour Methods for Real-Time Eye tracking", *Univ. of Sussex, School of Cognitive and Computing Sciences*, Tech. Rep. n° 386, 1995.
- 69 Valenti, R., Gevers, T. "Accurate Eye center Location and Tracking using Isophote Curvature", *IEEE CVPR* ,23-28, pp.1-8, 2008.
- 70 Hansen, D.W., Pece, A.E.C. "Eye Tracking in the Wild", *Computer Vision and Image Understanding (CVIU)*, 98(1), pp. 182-210, 2005.
- 71 Cootes, T.F., Taylor, C.J., *Active Shape Models – Smart Snakes*, *Proc. British Machine Vision*, pp. 266-275, 1992.
- 72 Lam, K., Yan, H. "Locating and Extracting the Eye in Human Face Images", *Pattern Recognition*, vol. 29, pp. 771-779, 1996.

- 73 Herpers, R., et al. "Edge and Keypoint Detection in Facial Regions", 2<sup>nd</sup> Int. Conf. on Automatic Face and Gesture Recognition, 14-16 Oct., pp.212-217, 1996.
- 74 Waite, J., Vincent, J.M. "A Probabilistic Framework for Neural Network Facial Feature Location", British Telecom Technology J., 10(3), pp. 20-29, 1992.
- 75 Bala, J., et al. Visual Routine for Eye Detection using Hybrid Genetic Architectures, Proc. 13<sup>th</sup> ICPR, Vienna, Aug. 23-26, 1996
- 76 Feng, G.C., Yuen, P.C. "Multi-Cue Eye Detection on Grey Intensity Image", Pattern Recognition, vol. 34, pp. 1033-1046, 2001.
- 77 Sirhey, S., Rosenfeld, A., Duric, Z. "A Method of Detecting and Tracking Irises and Eyelids in Video", Pattern Recognition, 35(6), pp. 1367-1391, 2002.
- 78 d'Orazio, T., Leo, M., Cicirelli, G., Distanto, A. "An Algorithm for Real Time Eye Detection in Face Images", 17th ICPR, 3(0), pp. 278-282, 2004.
- 79 Grauman, K. et al. "Communication via Eye Blinks : Detection and Duration Analysis in Real-time", IEEE CVPR, vol. 1, pp. 1010-1017, 2001.
- 80 Huang, J., Li, D., Shao, X., Wechsler, H. "Pose discrimination and eye Detection using Support Vector Machines", NATO-ASI Conf. on Face Recognition: from Theory to Applications, pp. 528-536, 1998.
- 81 Zhu, Z., Fujimura, K., Ji, Q. "Real-time Eye Detection and Tracking under Various Light Conditions", Proc. Eye Tracking Research and Applications Symp., 2002
- 82 Samaria, F., Young, S. "HMM-Based Architecture for Face Identification", Image and Vision Computing, 12(8), pp. 537-543, 1994.
- 83 Hillman, P.M., Hannah, J.M., Grant, P.M. "Global Fitting of a Facial Model to Facial Features for Model-Based Video Coding", Proc. 3rd Int. Symp. Image and Signal Processing and Analysis, vol. 1, pp. 359-364, 2003.
- 84 Huang, J., Wechsler, H. "Eye Detection Using Optimal Wavelet Packets and Radial Basis Functions", Int. J. Pattern Recognition and Artificial Intelligence, 13(7), 1999.
- 85 Viola, P. Jones, M. "Robust Real-Time Face Detection", Proc. ICCV 2001, vol. II, p. 747, 2001.
- 86 Wang, P., Ji, Q. "Learning Discriminant Features for Multi-View Face and Eye Detection", IEEE CVPR 2005, vol. 1, pp. 373-379, 2005.
- 87 Xie, X., Sudhakar, R., Zhuang, H. "A Cascaded Scheme for Eye Tracking and Head Movement Compensation", IEEE Trans. Systems, Man, and Cybernetics, 28(4), pp. 487-490, 1998.
- 88 Cootes, T.F., Edwards, G.J. , Taylor, C.J. "Active Appearance Models", Proc. ECCV (European Conf. Computer Vision), vol. 2, pp. 484-498, 1998.
- 89 Ishikawa, T., Baker, S., Matthews, I., Kanade, T. "Passive Driver Gaze Tracking with Active Appearance Models", 11th World Congress Intelligent Transportation Systems, 2004.
- 90 Hansen, D.W., Hansen, J.P., Nielsen, M., Johansen, A.S., Stegmann, M.B. "Eye Typing Using Markov and Active Appearance Models", Proc. IEEE Workshop Applications on Computer Vision, pp. 132-136, 2003.
- 91 Wang, J-G., Sung, E. "Study of Eye Gaze Estimation", IEEE Trans. on Systems, Man and Cybernetics, part B: Cybernetics, 32(3), 2002, pp. 332- 350, 2002.
- 92 Duchowski, A. "Eye Tracking Methodology: Theory and Practice". Springer-Verlag, 2003.
- 93 Morimoto, C.H., Mimica, M.R.M. "Eye Gaze Tracking Techniques for Interactive Applications", Computer Vision and Image Understanding, vol. 98, no. 1, pp. 4-24, Apr. 2005.
- 94 Ji, Q. , Yang, X. "Real-Time Eye, Gaze, and Face Pose Tracking for Monitoring Driver Vigilance", Real-Time Imaging, vol. 8, no. 5, pp. 357-377, 2002.
- 95 Mukawa, N. "A Free-Head, Simple Calibration, Gaze Tracking System That Enables Gaze-Based Interaction", Proc. *Eye Tracking Research and Applications Symp. 2004*, pp. 115-122, 2004.
- 96 Wang, J.G., Sung, E., Venkateswarlu, R. "Estimating the Eye Gaze from One Eye", CVIU (Computer Vision and Image Understanding), 98(1), pp. 83-103, 2005.
- 97 Merchant, J., Morrisette, R. , Porterfield, J. "Remote Measurements of Eye Direction Allowing Subject Motion over One Cubic Foot of Space", IEEE Trans. Biomedical Eng., 21(4), pp. 309-317, 1974.
- 98 Morimoto, C.H. , Koons, D. , Amir, A. , Flickner, M. "Pupil Detection and Tracking Using Multiple Light Sources", Image and Vision Computing, 18(4), pp. 331-335, 2000.
- 99 Shih, S.-W., Wu, Y.-T., Liu, J. "A Calibration-Free Gaze Tracking Technique", Proc. 15th Int. Conf. Pattern Recognition (ICPR'2000), pp. 201-204, 2000.

- 100 Villanueva, A., Cabeza, R. "Models for Gaze Tracking Systems", *J. Image and Video Processing*, (3), 2007.
- 101 Guestrin, E.D., Eizenman, M. "General Theory of Remote Gaze Estimation Using the Pupil Center and Corneal Reflections", *IEEE Trans. Biomedical Eng.*, 53(6), pp. 1124-1133, 2006.
- 102 Yoo, D.H., Chung, M.J. "A Novel Non-Intrusive Eye Gaze Estimation Using Cross-Ratio under Large Head Motion", *Computer Vision and Image Understanding*, 98(1), pp. 25-51, 2005.
- 103 Coutinho, F.L., Morimoto, C.H. "Free Head Motion Eye Gaze Tracking Using a Single Camera and Multiple Light Sources", *Proc. ACM SIGGRAPH*, 2006.
- 104 Hennesy, C., Lawrence, P. "3D Point-of-Gaze Estimation on a Volumetric Display", *Proc. 2008 Symp. Eye Tracking Research and Applications*, 2008.
- 105 Williams, O.M.C., Blake, A., Cipolla, R. "Sparse and Semi-Supervised Visual Mapping with the  $s^3p$ ", *Proc. IEEE CVPR (Computer Vision and Pattern Recognition)*, pp. 230-237, 2006.
- 106 Tan, K.-H., Kriegman, D.J., Ahuja, N. "Appearance-Based Eye Gaze Estimation", *Proc. Sixth IEEE Workshop Applications of Computer Vision '02*, pp. 191-195, 2002.
- 107 Sesin, A., Adjouadi, M., Cabrerizo, M., Ayla, M. & Barreto, M. "An Eye Gaze Tracking System using Customized User Profiles to Help with Motor Challenges Access Computers". *Proceedings of the 11th international conference on Computers Helping People with Special Needs (ICCHP)*, pp. 1168-1175, 2008.
- 108 Fejtova M., Fejt, J., Lhotska, L. "Controlling a PC by Eye Movements : The MEMREC Project", *ICCHP'2004, LNCS 3118*, pp. 770-773, 2004.
- 109 Li, D., Winfield, D. & Parkhurst, D.J. "Starburst: A Hybrid Algorithm for Video-Based Eye Tracking Combining Feature-Based and Model-Based Approaches". *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) – Workshop*, p. 1-8, 2005.
- 110 Babcock, J. & Pelz, J. "Building a lightweight eyetracking headgear", *Proc. of the 2004 Symp. on Eye Tracking Research and Applications Conference (ETRA)*, pp. 109-114, 2004.
- 111 Li, D., Babcock, J. & Parkhurst, D. J. "openEyes: a low-cost head mounted eye-tracking solution". *Proceedings of the 2006 Symposium on Eye Tracking Research and Applications Conference (ETRA)*, pp. 95-100, 2006.
- 112 L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice (2nd Edition)*, Addison-Wesley Professional, 2003.
- 113 Gamma, *Design Patterns: Elements of Reusable Object-Oriented Software*, Pearson Education, .
- 114 F. Buschmann and K.H.&D. Schimdt, *Pattern-Oriented Software Architecture Volume 5*, Wiley India Pvt. Ltd., 2007.
- 115 M. Fowler, *Patterns of enterprise application architecture*, Addison-Wesley, 2003.
- 116 G. Reese, *Database programming with JDBC and Java*, O'Reilly Media, Inc., 2000.
- 117 K. Brown, G. Craig, G. Hester, and D. Pitt, *Enterprise Java programming with IBM WebSphere*, Addison-Wesley, 2003.
- 118 D. Mcilroy, "Mass-Produced Software Components," *Proceedings of the 1st International Conference on Software Engineering*, Garmisch Pattenkirchen, Germany, 1968, pp. 98, 88
- 119 C. Szyperski, *Component software: beyond object-oriented programming*, ACM Press/Addison-Wesley Publishing Co., 1998.
- 120 C. Szyperski, D. Gruntz, and S. Murer, *Component software*, Addison-Wesley, 2002.
- 121 Microsoft Visual C#(TM) .NET Language Reference, Microsoft Press, 2002.
- 122 E. Bruneton, T. Coupaye, M. Leclercq, V. Quéma, and J. Stefani, "The FRACTAL component model and its support in Java: Experiences with Auto-adaptive and Reconfigurable Systems," *Softw. Pract. Exper.*, vol. 36, 2006, pp. 1257-1284.
- 123 D. Garlan, R.T. Monroe, and D. Wile, "Acme: architectural description of component-based systems," *Foundations of component-based systems*, Cambridge University Press, 2000, pp. 47-67.
- 124 "IBM Redbooks | Patterns: Service-Oriented Architecture and Web Services," Jul. 2004.
- 125 T. Erl, *Service-oriented architecture*, Prentice Hall Professional Technical Reference, 2005.
- 126 P.R. Allen and S. Frost, *Component-based development for enterprise systems*, Cambridge University Press, 1998.
- 127 R.E. Schantz and D.C. Schmidt, "Middleware for distributed systems - evolving the common structure for network-centric applications," *Encyclopedia of Software Engineering*, 2001.

- 128 A. Gokhale, D.C. Schmidt, B. Natarajan, and N. Wang, "Applying model-integrated computing to component middleware and enterprise applications," *Communications of the ACM*, vol. 45, 2002, pp. 65–70.
- 129 D.C. Schmidt, "Middleware for real-time and embedded systems," *Communications of the ACM*, vol. 45, 2002, pp. 43–48.
- 130 D.C. Schmidt, "Applying Patterns and Frameworks to Develop Object-Oriented Communication Software," *OF HANDBOOK OF PROGRAMMING LANGUAGES*. MACMILLAN COMPUTER PUBLISHING, vol. 1, 1997.
- 131 Monson-Haefel, B. Burke, and S. Labourey, *Enterprise JavaBeans*, O'Reilly, 2004.
- 132 R. Pichler, K. Ostermann, and M. Mezini, "On aspectualizing component models," *Software: Practice and Experience*, vol. 33, 2003, pp. 957-974.
- 133 A.L. Tavares and M.T. Valente, "A gentle introduction to OSGi," *SIGSOFT Software Engineering Notes*, vol. 33, 2008, pp. 1–5.
- 134 N. Bartlett, *OSGi In Practice*, 2009.
- 135 N. Bartlett, "A Comparison of Eclipse Extensions and OSGi Services," Feb. 2007.
- 136 H. Cervantes and R.S. Hall, "Automating service dependency management in a service-oriented component model," *Proceedings of the 6th Workshop on Component-Based Software Engineering (CBSE) in conjunction with the 25th International Conference on Software Engineering (ICSE)*, Portland, Oregon, USA: 2003.
- 137 C. Szyperski, *Component software: beyond object-oriented programming*, Addison-Wesley Professional, 1997.
- 138 Wayne Wolf, "What Every Embedded Systems Designer Should Know", *Embedded System Conference*, San Francisco 2001
- 139 Jean J. Labrosse, „Designing with Real-Time Kernels“, *ESC*, San Francisco 2002
- 140 David Kalinsky, „A Survey of Task Schedulers“, *ESC*, Stuttgart 2001
- 141 Jean J. Labrosse, „The 10-Minute Guide to RTOS“, *Application Note AN-1003*, Micrium, Inc.
- 142 Dougan, Cort (2004) "Precision and predictability for Linux and RTLinuxPro", *Dr. Dobbs Journal*, February 01, 2004.
- 143 Intel Solution Brief for Industrial Control Devices, [http://download.microsoft.com/download/9/0/B/90B0F26E-F67F-4B2A-AA63-9E415AC1C1E5/SB\\_Microsoft\\_CE.PDF](http://download.microsoft.com/download/9/0/B/90B0F26E-F67F-4B2A-AA63-9E415AC1C1E5/SB_Microsoft_CE.PDF)
- 144 Windows Embedded Platform Builder User's Guide, <http://msdn.microsoft.com/en-us/library/aa913745.aspx>
- 145 Windows CE 6 Kernel Architecture, <http://www.addlogic.se/articles/articles/windows-ce-6-kernel-architecture.html>
- 146 Windows CE 6 OAL – What you need to know, [http://blogs.msdn.com/ce\\_base/archive/2006/11/21/ce6-oal-what-you-need-to-know.aspx](http://blogs.msdn.com/ce_base/archive/2006/11/21/ce6-oal-what-you-need-to-know.aspx)
- 147 Access and Open Source FAQ, Access Company Website, <http://www.access-company.com/about/opensource/faq.html>
- 148 Access Linux Platform Overview, <http://www.accessdevnet.com/index.php/ACCESS-Linux-Platform/ACCESS-Linux-Platform.html>
- 149 Access Linux Platform Architecture, <http://www.accessdevnet.com/index.php/ACCESS-Linux-Platform/Architecture.html>
- 150 Palm webOs Fundamentals, [http://developer.palm.com/index.php?option=com\\_content&view=article&id=1845&Itemid=27](http://developer.palm.com/index.php?option=com_content&view=article&id=1845&Itemid=27)
- 151 Palm webOs Plugin Development, [http://developer.palm.com/index.php?option=com\\_content&view=article&id=2000&Itemid=33](http://developer.palm.com/index.php?option=com_content&view=article&id=2000&Itemid=33)
- 152 Kernel Architecture 2 Architecture, [http://developer.symbian.org/main/documentation/reference/s%5E3/doc\\_source/guide/KernelandHardwareServices/kernelarch/Concepts/Architecture.html](http://developer.symbian.org/main/documentation/reference/s%5E3/doc_source/guide/KernelandHardwareServices/kernelarch/Concepts/Architecture.html)
- 153 Huckauf A, Goettel T, Heinbockel M, Urbina M (2005) What you don't look at is what you get: anti-saccades can reduce the midas touch-problem. In: *Proceedings of the 2nd symposium on applied perception in graphics and visualization*. p 170
- 154 Poky Platform Builder Project Website, <http://www.pokylinux.org/>

- 155 The MathWorks: Simulink graphical design of dynamic system simulations, <http://www.mathworks.com/products/simulink>
- 156 Edward B. Magrab, Shapour Azarm, Balakumar Balachandran, and James Duncan: An Engineers Guide to MATLAB, Prentice Hall, 3rd Edition, Jan. 17 2010, ISBN: 0131991108
- 157 The MathWorks: Deploying Matlab software on Embedded Systems
- 158 National Instruments: LabView : <http://www.ni.com/labview/>
- 159 Example of a LabView Signal Diagram: [http://www.webs1.uidaho.edu/ME451/Lab%20Results/Labview\\_%20Diagram.JPG](http://www.webs1.uidaho.edu/ME451/Lab%20Results/Labview_%20Diagram.JPG)
- 160 Example of a LabView graphical User Interface: [http://datataker.com/products/Pics/LabVIEW\\_weathervi\\_H.jpg](http://datataker.com/products/Pics/LabVIEW_weathervi_H.jpg)
- 161 Arm Prototyping Board for Labview: <http://www.ni.com/arm/>
- 162 Cyberevolution BioExplorer: Software application for Biofeedback, <http://www.cyberevolution.com>
- 163 Bernardes, J. a., Nakamura, R., Calife, D., Tokunaga, D., and Tori, R. Integrating the wii controller with engine: 3d interfaces extending the frontiers of a didactic game engine. 2009. Comput. Entertain., 7(1). pages 1–19.
- 164 BioEra Lite Sourceforge project page, <http://sourceforge.net/projects/bioera/>
- 165 Proatech LLC: BioEra Pro Developer's Manual, <http://www.bioera.net/manual.html>
- 166 Brainbay Project Website, <http://brainbay.io-res.org/>
- 167 The Intel Open Computer Vision Library: <http://www.intel.com/technology/computing/opencv>
- 168 C. Veigl: BrainBay: an Open-Source System for Biosignal- and Camera-Mouse Applications, Young Researchers Consortium, ICCHP 2006
- 169 Flachberger, C.; Panek, P; Zagler, W.L.: Compose Autonomy!- An Adaptable User Interface for Assistive Technology, Proceedings of the 2nd TIDE Congress(The European Context for Assistive Technology), IOS Press, Paris, 1995, p.413-416.
- 170 OpenViBE Software for Brain Computer Interfaces and Real Time Neurosciences: <http://openvibe.inria.fr/>
- 171 OpenVibe Screenshot: 2D topographical map
- 172 Detlefs D., Flood C., Heller S. Printezis T. Garbage-first Garbage Collection 2004. In Proceedings of the 4th International Symposium on Memory Management p.37-48, Vancouver, Canada
- 173 Bouquet, J., 2007. Camera Calibration Toolbox for Matlab. <http://www.vision.caltech.edu/bouquetj/calibdoc/>
- 174 Visual Geometry Group, 2005. MATLAB functions for multiple view geometry. Department of Engineering Science, University of Oxford.
- 175 Kovsi, P. D., 2007. MATLAB and Octave functions for computer vision and image processing. School of Computer Science & Software Engineering, The University of Western Australia.
- 176 P. Blenkhorn, "Using a Touch Sensitive Screen," *British Journal of Special Education*, vol. 13, 1986, pp. 161-161.
- 177 Rakotomamonjy, A., Guigue, V. "BCI Competition III: Dataset II- Ensemble of SVMs for BCI P300 Speller" *IEEE Trans Biomed Eng.* 2008 Mar;55(3):1147-54.
- 178 Steven Lemm, Christin Schäfer, Gabriel Curio. "Aggregating Classification Accuracy across Time: Application to Single Trial EEG" Campus Benjamin Franklin <http://ida.first.fhg.de/publications/LemSchCur07.pdf>
- 179 Yijun Wang, Bo Hong, Xiaorong Gao, and Shangkai Gao. "Phase Synchrony Measurement in Motor Cortex for Classifying Single-trial EEG during Motor Imagery". Proceedings of the 28th IEEE EMBS Annual International Conference New York City, USA, Aug 30-Sept 3, 2006
- 180 Dan Zhang, YijunWang, Xiaorong Gao, Bo Hong, and Shangkai Gao. "An algorithm for idle-state detection in motor-imagery-based brain-computer interface" *Computational Intelligence and Neuroscience* Volume 2007, Article ID 39714
- 181 Ferran Galán , Francesc Oliva and Joan Guàrdia. "Using mental tasks transitions detection to improve spontaneous mental activity classification" *Med Bio Eng Comput* (2007) 45:603–609 DOI 10.1007/s11517-007-0197-7
- 182 S. Bashashati, A. Fatourehchi, M. Navarro, K.Birch, "A comprehensive survey of brain interface technology designs," *Annals of Biomedical Engineering*, vol. 35, no. 2, pp. 137–169, February 2007. [Online]. Available: <http://dx.doi.org/10.1007/s10439-006-9170-0>.

- 183 T. Furuhashi, "Fusion of fuzzy/neuro/evolutionary computing for knowledge acquisition," Proceedings of the IEEE, vol. 89, no. 9, pp. 1266-1274, Sept 2001. [Online]. Available: <http://dx.doi.org/10.1109/5.949484>.
- 184 R. Duda, P. Hart, and D. Stork, Pattern Classification, second ed. John Wiley & Sons, 2001.
- 185 J. Qin, Y. Li, and W. Sun, "A semisupervised support vector machines algorithm for BCI systems." Computational intelligence and neuroscience, 2007. [Online]. Available: <http://dx.doi.org/10.1155/2007/94397>.
- 186 J. C. Sanchez, J. C. Principe, T. Nishida, R. Bashirullah, J. G. Harris, and J. A. B. Fortes, "Technology and signal processing for brain-machine interfaces," Signal Processing Magazine, IEEE, vol. 25, no. 1, pp. 29–40, 2008. [Online]. Available: <http://dx.doi.org/10.1109/MSP.2008.4408440>.
- 187 P. Herman, G. Prasad, T. M. M. McGinnity, and D. Coyle, "Comparative analysis of spectral approaches to feature extraction for EEG-based motor imagery classification." IEEE Trans. neural systems and rehabilitation engineering, vol. 16, no. 4, pp. 317-326, August 2008. [Online]. Available: <http://dx.doi.org/10.1109/TNSRE.2008.926694>.
- 188 G. Bin, X. Gao, Y. Wang, B. Hong, and S. Gao, "Vep-based brain-computer interfaces: time, frequency, and code modulations [research frontier," IEEE Computational Intelligence Magazine, vol. 4, no. 4, pp. 22–26, November 2009. [Online]. Available: <http://dx.doi.org/10.1109/MCI.2009.934562>.
- 189 F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, "A review of classification algorithms for eeg-based brain-computer interfaces." Journal of neural engineering, vol. 4, no. 2, June 2007. <http://dx.doi.org/10.1088/1741-2560/4/2/R01>.
- 190 Kachenoura, L. Albera, L. Senhadji, and P. Comon, "ICA: a potential tool for BCI systems," Signal Processing Magazine, IEEE, vol. 25, no. 1, pp. 57–68, 2008. [Online]. Available: <http://dx.doi.org/10.1109/MSP.2008.4408442>-
- 191 C.-T. Lin, L.-W. Ko, and T.-K. Shen, "Computational intelligent brain computer interaction and its applications on driving cognition," IEEE Computational Intelligence Magazine, vol. 4, no. 4, pp. 32–46, November 2009. [Online]. Available: <http://dx.doi.org/10.1109/MCI.2009.934559>.
- 192 Wei Wu, Xiaorong Gao, and Shangkai Gao, "One-Versus-the-Rest (OVR) algorithm: an extension of Common Spatial Patterns (CSP) algorithm to multi-class case," Proc. IEEE EMBS Conf., Shanghai, China, 2005.
- 193 I. Cester, J. Llobera, A. Soria-Frisch, "BCI classification, OVR approach", Starlab Technical Report, TN00146, 2009.
- 194 P. S. Hammon, S. Makeig, H. Poizner, E. Todorov, and V. R. de Sa, "Predicting reaching targets from human eeg," Signal Processing Magazine, IEEE, vol. 25, no. 1, pp. 69–77, 2008. [Online]. Available: <http://dx.doi.org/10.1109/MSP.2008.4408443>.
- 195 B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K.-r. Müller, "Optimizing spatial filters for robust EEG single-trial analysis," in IEEE Signal Proc. Magazine, vol. 25, 2008, pp. 581–607.
- 196 D. Coyle, G. Prasad, and T. M. M. McGinnity, "A time-series prediction approach for feature extraction in a brain-computer interface." IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society, vol. 13, no. 4, pp. 461–467, December 2005. [Online]. Available: <http://dx.doi.org/10.1109/TNSRE.2005.857690>.
- 197 D. Coyle, "Neural network based auto association and time-series prediction for biosignal processing in brain-computer interfaces," IEEE Computational Intelligence Magazine, vol. 4, no. 4, pp. 47–59, November 2009. [Online]. Available: <http://dx.doi.org/10.1109/MCI.2009.934560>.
- 198 L. Citi, R. Poli, C. Cinel, and F. Sepulveda, "P300-based bci mouse with genetically-optimized analogue control," Neural Systems and Rehabilitation Engineering, IEEE Transactions on, vol. 16, no. 1, pp. 51–61, 2008. [Online]. Available: <http://dx.doi.org/10.1109/TNSRE.2007.913184>.
- 199 R. Poli, L. Citi, F. Sepulveda, and C. Cinel, "Analogue evolutionary brain computer interfaces [application notes," IEEE Computational Intelligence Magazine, vol. 4, no. 4, pp. 27–31, November 2009. [Online]. Available: <http://dx.doi.org/10.1109/MCI.2009.934563>.
- 200 S. Fazli, F. Popescu, M. Danóczy, B. Blankertz, K.-R. R. Müller, and C. Grozea, "Subject-independent mental state classification in single trials." Neural Networks, vol. 22, no. 9, pp. 1305–1312, November 2009. [Online]. Available: <http://dx.doi.org/10.1016/j.neunet.2009.06.003>.