

Cranfield University  
School of Engineering  
Applied Mathematics and Computing Group

PhD Thesis

Academic Year 2003-2004

Alexandre Dugarry

**Advanced Driver Assistance Systems  
Information Management and Presentation**

Supervisor: Dr. Shan Fu

June 2004

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

© Cranfield University 2004. All rights reserved. No part of this publication may be reproduced without the written permission of the copyright owner.

# Abstract

With the development of advanced driving assistance systems, in-vehicle communication and information systems, there are situations where the driver becomes overloaded by information, creating potentially dangerous conditions. In this Thesis a novel strategy is proposed, to prioritise and present information.

Firstly two main criteria are extracted, that allow the ability to rank messages: the risk associated with the non-presentation of the message, and its relevance to the environment. Fuzzy cognitive maps enable to represent expert knowledge and model these relationships.

Secondly, a strategy to present information is proposed. Using an importance index, calculated from the previous risk and relevance indices, but also information nature, time constraints and access frequency, a set of best interfaces is selected. Furthermore design a model of driver workload is designed, based on the multiple resources theory. By estimating in real time the workload of the driver, the system enables to choose an optimal interface, that should prevent overload.

This Thesis presents then the tools developed for the implementation and testing of the model. A video capture and data transfer program, based on the IEEE-1394 bus, enable in-vehicle real-time data capture and collection. Moreover, a software package for replay of the acquired data, analysis and simulation is developed. Finally, the implementation of the prioritisation and presentation strategy is outlined.

The last part of this work is dedicated to the experiments and results. Using an experimental vehicle, data in different driving conditions are collected. the experiment is completed by creating data to simulate potentially dangerous situations, where driver is overloaded with information. The re-

sults show that the information management and presentation system is able to prevent overload in most conditions. Its structure and design allow to incorporate expert knowledge to refine the classification.

# Acknowledgements

I would like to thank my supervisor Dr. Shan Fu and the Head of the Applied Mathematics and Computing Group, Prof. Chris Thompson, for the numerous and interesting opportunities they provided me with during these years. I must also express my gratitude to the Royal Academy of Engineering, that very kindly sponsored my attendance to the Electronic Imaging IV conference. Special thanks to Dr. J.D.B. Nelson, for his linguistic expertise, and for having the most comprehensible English accent around. And, by order of appearance, thanks to the AMAC staff and students: Peter, Rachael, Christian, Liong, Maryam, Homa, George, Julian, Wei, Pauline and Jerome.

I certainly would have not started and completed this Thesis without the constant support of my parents Dominique and Joël, my brother Camille and my sister Axelle. Finally, I would like to thank Aïda for being the best student I supervised, and for her infinite patience while I was ‘nearly finished’ with this work.

# Contents

<b>Table of Contents</b>	<b>5</b>
<b>List of Figures</b>	<b>8</b>
<b>List of Tables</b>	<b>10</b>
<b>Acronyms</b>	<b>11</b>
<b>1 Introduction</b>	<b>13</b>
1.1 Background . . . . .	13
1.2 Objectives . . . . .	14
1.3 Implementation . . . . .	15
1.4 Organisation of thesis . . . . .	15
1.5 Novelty of research work . . . . .	16
<b>2 Technology review</b>	<b>17</b>
2.1 Information flow . . . . .	17
2.2 Advanced Driver Assistance Systems . . . . .	20
2.2.1 Lateral control systems . . . . .	20
2.2.2 Longitudinal control systems . . . . .	21
2.2.3 Reversing/parking aids . . . . .	22
2.2.4 Vision enhancement systems . . . . .	22
2.2.5 Intelligent speed adaptation . . . . .	23
2.3 In-vehicle information system . . . . .	23
2.3.1 Instrumentation . . . . .	23
2.3.2 Communication . . . . .	23
2.3.3 Comfort and convenience . . . . .	24

2.4	Information management systems . . . . .	24
2.5	Data networks . . . . .	26
2.5.1	Automotive . . . . .	26
2.5.2	IEEE 1394 . . . . .	27
2.6	Driving environment recognition . . . . .	31
2.7	Human-vehicle interface and safety . . . . .	32
2.8	Driver workload . . . . .	33
2.9	Fuzzy cognitive maps . . . . .	36
<b>3</b>	<b>Information prioritisation</b>	<b>39</b>
3.1	Risk . . . . .	40
3.2	Relevance . . . . .	41
3.2.1	Normalisation . . . . .	45
3.3	Importance . . . . .	47
<b>4</b>	<b>Information presentation</b>	<b>48</b>
4.1	Optimal interface choice . . . . .	49
4.1.1	Nature . . . . .	49
4.1.2	Time constraints . . . . .	50
4.1.3	Access frequency . . . . .	50
4.2	Workload . . . . .	51
4.2.1	Definition . . . . .	51
4.2.2	Modelling . . . . .	52
4.3	Strategy . . . . .	55
<b>5</b>	<b>Implementation</b>	<b>59</b>
5.1	Real time data capture . . . . .	59
5.1.1	Background . . . . .	59
5.1.2	Time synchronisation . . . . .	60
5.1.3	Video capture . . . . .	61
5.1.4	Data transfer . . . . .	62
5.2	Simulation and replay . . . . .	66
5.3	Prioritisation . . . . .	68

<b>6</b>	<b>Case study</b>	<b>69</b>
6.1	Experiment design . . . . .	69
6.2	Strategy implementation . . . . .	74
6.2.1	Definitions . . . . .	74
6.2.2	Lane warning . . . . .	75
6.2.3	Navigation . . . . .	76
6.2.4	Phone . . . . .	77
6.2.5	Instrumentation . . . . .	78
6.2.6	Traffic Alerts . . . . .	79
6.2.7	Driving task . . . . .	79
6.2.8	Normalisation . . . . .	80
6.2.9	Interfaces . . . . .	81
6.3	Results . . . . .	82
<b>7</b>	<b>Conclusion and further works</b>	<b>87</b>
	<b>Bibliography</b>	<b>90</b>
	<b>Appendices</b>	<b>97</b>
<b>A</b>	<b>Facial expression presentation for real-time Internet communication</b>	<b>97</b>
A.1	Introduction . . . . .	99
A.2	Methodology . . . . .	100
A.2.1	Features extraction . . . . .	101
A.2.2	Raster to vector conversion . . . . .	103
A.2.3	Conversion to SVG . . . . .	105
A.3	Application . . . . .	106
A.3.1	Description . . . . .	106
A.3.2	Results . . . . .	107
A.4	Conclusion and further work . . . . .	110
<b>B</b>	<b>IEEE-1394 Data Transfer</b>	<b>111</b>
<b>C</b>	<b>Data replay, simulation and analysis</b>	<b>118</b>

# List of Figures

2.1	IEEE-1394 PC architecture . . . . .	28
2.2	IEEE-1394 tree architecture . . . . .	29
2.3	IEEE-1394 data frame . . . . .	30
2.4	IEEE-1394 network layers . . . . .	30
2.5	Multiple resources theory . . . . .	34
2.6	Relation between task demand, performance and workload [1] . . . . .	35
2.7	Fuzzy cognitive map . . . . .	37
3.1	Example of risk values . . . . .	42
3.2	Simple FCM for lane departure warning relevance . . . . .	44
3.3	Sigmoid function for $a = 2$ , $b = -1$ and $\alpha = 1$ . . . . .	45
3.4	Speed average membership function . . . . .	46
4.1	Presentation media . . . . .	49
4.2	Information message characterisation . . . . .	51
4.3	Example of preferred interfaces for navigation direction messages . . . . .	57
4.4	Flow chart for real-time message management . . . . .	58
5.1	JPEG image capture algorithm . . . . .	62
5.2	Single versus multiple threads data transfer. . . . .	64
5.3	Simplified flowchart for IEEE-1394 file transfer. . . . .	65
5.4	Events synchronisation . . . . .	67
6.1	Experimental vehicle views . . . . .	70
6.2	Experiment route . . . . .	72
6.3	Experiment route events . . . . .	73
6.4	FCM for lane warning system relevance . . . . .	75



6.5	FCM for navigation system . . . . .	76
6.6	FCM for phone . . . . .	77
6.7	FCM for instrumentation system . . . . .	78
6.8	FCM for traffic alerts . . . . .	79
6.9	Sigmoid function for phone . . . . .	81
6.10	Driver steering workload measurement. . . . .	83
6.11	Prioritisation result . . . . .	84
6.12	Prioritisation result . . . . .	85
6.13	Irrelevant lane warning message . . . . .	86
A.1	Median filtering . . . . .	101
A.2	Vectorisation algorithm . . . . .	105
A.3	Networking architecture . . . . .	107
A.4	Rendered SVG with 160x120 source images . . . . .	108
A.5	Rendered SVG with 320x240 source images . . . . .	108
A.6	Rendered SVG with 640x480 source images . . . . .	109
A.7	Performance measurements . . . . .	109
C.1	UML view of player class . . . . .	119
C.2	UML view of file class . . . . .	120
C.3	Player view . . . . .	121
C.4	Simulation view . . . . .	122
C.5	Experimental data prioritisation - lane warning . . . . .	123
C.6	Experimental data prioritisation - traffic alert . . . . .	124

# List of Tables

2.1	Percentages of drivers for different types of distractions [2]	32
3.1	Risk groups and information sources	41
3.2	ADAS and relevance to the environment	43
6.1	Risks groups and values	74
6.2	Systems and interfaces for simulation	82

# Acronyms

**ADAS** Advanced Driver Assistance System

**ARM** Asynchronous Resources Manager

**AGP** Accelerated Graphic Port

**CAN** Control Area Network

**CSR** Control and Status Register

**DAB** Digital Audio Broadcasting

**DVD** Digital Versatile Disk

**FCM** Fuzzy Cognitive Map

**GPS** Global Positioning System

**HEUDIASYC** Heuristique et Diagnostic des Systèmes Complexes

**HMI** Human Machine Interface

**HVI** Human Vehicle Interface

**IEEE** Institute of Electrical and Electronics Engineers

**IIDC** Instrumentation and Industrial Digital Cameras

**IDB** ITS Data Bus

**ITS** Intelligent Transport Systems

**JPEG** Joint Picture Experts Group

**LASMEA** Laboratoire des Sciences et Matériaux pour l'Electronique et  
d'Automatique

**MOST** Media Oriented Systems Transfer

**NRZ** Non Return to Zero

**LIDAR** Light Detection And Ranging

**RADAR** Radio Detection And Ranging

**ROADSENSE** Road Awareness for Driving via a Strategy that Evaluates  
Numerous Systems

**SWT** Standard Widget Toolkit

**UBP** Université Blaise Pascal

**UTC** Université de Technologie de Compiègne

# Chapter 1

## Introduction

### 1.1 Background

The following work was developed in the context of the European Union Framework V project Road Awareness for Driving via a Strategy that Evaluates Numerous Systems (ROADSENSE). The main objective of Roadsense was to ‘develop an industry standard evaluation framework for new Human Vehicle Interactions with a specific focus on Human Machine Interfaces’ [3]. This involved collaborative work with the car manufacturers Jaguar, PSA Peugeot Citroen, Renault, Fiat and Porsche, the European universities Université Blaise Pascal (UBP), Université de Technologie de Compiègne (UTC), and Cranfield University, and the research centres TNO, in Nederland, Laboratoire des Sciences et Matériaux pour l’Electronique et d’Automatique (LASMEA) and Heuristique et Diagnostic des Systèmes Complexes (HEUDIASYC), in France.

In the year 2002, a total of 49,718 people were killed on European roads [4, 5, 6], and approximately 1,700,000 were injured. Between 1992 and 2002, more than half a million people (619,885) died in car accidents. Much effort has been applied to decrease these figures, both political, via legislation, and through the design of novel safety systems. Many driver support systems have been developed in the last decade to improve road safety in Europe. These have taken the form of active systems such as, for example, antilock brake system, electronic stability program, airbags, collision detection or intelligent

cruise control. In parallel, many systems, such as navigation systems, have been designed to make the driving task easier.

Until now, most of the research and development efforts have been dedicated to individual technologies; colour display, vocal interfaces, visibility improvement using far-infrared, dynamic navigation, monitoring of driver behaviour and lane keeping. However, very few look take a holistic approach to the driver support system.

This research was carried out to solve a new problem, arising with the development of new in-car information systems: too many of them may surpass the driver's abilities, and therefore create potentially dangerous situations.

## 1.2 Objectives

In this thesis is proposed an enhanced design strategy to filter and present information to the driver. Of particular importance is the alleviation of lowering driver workload via provision of selected information with optimal media. Moreover, it aims at designing a system that the automotive industry can use and expand, by incorporating their own knowledge of the driver behaviour and its relationship to the environment.

This work can be regarded in terms of three main research objectives.

Firstly, the information and the in-car information system must be characterised in order to highlight classification criteria.

Secondly, it is necessary to define the driver's workload. Thus a model must be designed to represent the capabilities of the driver and their ability to process supplementary information. Workload measurement techniques can be implemented to correct the model and make it more accurate in real-time.

The third and last aspect of the research is to select a presentation medium, based on the importance of the information, according to the classification criteria and the driver's workload estimation and measurement.

## 1.3 Implementation

To evaluate the performance of the prioritisation strategy, data representing the Human Vehicle Interface (HVI) and the driving environment are required. However in-car data logging systems are usually expensive and limited in terms of number of data streams they can acquire simultaneously.

To tackle these difficulties, a real-time distributed recording system must be developed. Integrated into the Roadsense project, it must be based on the IEEE-1394 bus and produce compatible files.

Furthermore, to complete the evaluation, some data must be simulated.

Since a system is required that affords car manufacturers the opportunity to adapt and extend to their own needs, proprietary implementations must be avoided. Java is a programming language that does not depend on the underlying architecture. As presented in section 5.2, its performance is comparable to C++ in some areas, and its structure and robustness make it the language of choice for this work.

However, the data logging system needs to be readably available. It will be developed under Linux, an open-source operating system. IEEE-1394 is supported and libraries for video cameras control and raw communications are freely available, as discussed in Section 5.1.

## 1.4 Organisation of thesis

Chapter 2 presents the currently available technology and offers a review of the relevant literature. It begins with a description of the in-car information network, the driving assistance systems and the human-vehicle interface. The concepts of task demand and driver workload are discussed before the mathematical tools employed, such as fuzzy logic and fuzzy cognitive maps, are presented.

Chapter 3 is dedicated to the proposed prioritisation strategy. The characteristics of information sources are firstly studied, in order to extract classification criteria. Then a model is built, with respect to the representation or the driver workload. Finally a prioritisation strategy is proposed.

In Chapter 4, is discussed the choice of the presentation media. It depends

on the results of the previous chapter, but also on several other constraints, including the driver's ability to process information.

In Chapter 5, is presented a set of tools to implement this strategy, including novel real-time data logging, replay, simulation and analysis applications.

A case study is presented in Chapter 6. An experimental vehicle is used to collect data from real driving situation. Events generated by the in-car information system are simulated, and the prioritisation is implemented. In the last part of the chapter, several examples of information prioritisation and presentation are presented.

Chapter 7 concludes this thesis.

## **1.5 Novelty of research work**

This work presents several novel contributions.

First it takes an holistic approach by considering the whole information system, from sensors to interface. It proposes a prioritisation strategy valid for the whole in-vehicle information system.

Secondly, it offers a comprehensible set of criteria to characterise and sort information, and introduces the notion of relevance of information to the area of automotive human-vehicle interfaces. Fuzzy cognitive maps are applied for the first time to this domain. This makes the system predictable, easily maintainable and expandable.

Finally, it presents an implementation of a novel real-time architecture to capture and synchronise data produced by the in-car information system, as well as a set of tools to analyse and improve the prioritisation and presentation strategy.



# Chapter 2

## Technology review

This Chapter aims to give a description of the in-car information system, the Advanced Driver Assistance Systems (ADASs), and the literature that is relevant to information management and presentation.

It is first essential to study the information flow from the sensors that produce it, to the driver.

### 2.1 Information flow

The in-car information system comprises three inter-dependant parts: the information producers, the transport network, and the driver interface. Although previous systems included their own sensors, communication layer, and dedicated presentation interface, today they tend to be more integrated in the in-car information system. The information producers are the various sensors, communication devices, or more advanced systems that make information available to the user, namely

- The raw sensors that are part of the common instrumentation (speed, revolutions per minute, oil level, temperature, etc.);
- The driver assistance systems;
- The entertainment system (radio, video);
- The communication devices, such as phone and internet terminal;

- The comfort system (e.g. air conditioning).

The information moves from the information provider to the driver interface through a heterogeneous network that takes the form of two layers. The first layer is used for safety-critical information, i.e. information that, if not delivered in time, could result in harming the driver or its passengers. The Control Area Network (CAN) bus is generally chosen for this task. A second layer is dedicated to less critical information, such as communication and entertainment. The Media Oriented Systems Transfer (MOST) and Digital Audio Broadcasting (DAB) buses have been designed for this purpose.

The information must then be presented to the driver, through an appropriate interface. There are three ways to communicate with the driver, namely: visually, aurally, or haptically. Although the sense of smell has been proved to be a very efficient carrier of information [7], there is no system yet that exploits this media.

Currently, the main channel of nowadays human-vehicle interfaces is visual. A classic in-car interface is usually composed of a instrumentation panel, in front of the driver, behind the wheel and one or more LCD screen in the central console, dedicated to the display of the peripheral systems, such as navigation systems, radio information, TV, etc. Moreover, the driving task itself mainly requires the use of the visual channel: the driver needs to look at the road, the sides of the car and the mirrors.

The visual display can be sorted in several categories:

- Static displays, such as the instrumentation panel;
- Lights turned on or off: used lights, warning;
- Blinking lights: indicators;
- Text display — message is displayed on a screen;
- Image display: navigation system (pictogram);
- Video/image sequence: TV, DVD;

The presentation of an information reflect generally its importance. On current commercial vehicles, the information required to perform the driving

task, such as speed, RPMs, indicators or petrol level are generally displayed as near as possible to the line of sight; thus the driver needs minimum time to switch from the driving scene to the instruments.

Information of secondary importance such as radio, air conditioning status is usually displayed on the central console. They are further from the line of sight, but need to be accessed in a less frequent manner.

Three properties are used to visually render the criticality, or the importance of a message:

- its position on in-car interface. The driver pays more attention to information displayed in its direct field of vision; secondary information can be displayed on the central console.
- its size, relative to other items on the display.
- its colour: most interfaces use a red, orange or green display. Colours have established meaning within a population.

There are several categories of aural interfaces. The simplest is the ‘beep’ used for warning the driver. The in-vehicle speech synthesis systems appeared in the 1980’s and alerted the driver with more meaningful messages. It is also used nowadays for navigation systems and text-to-speech systems, such as e-mail reading programs. Finally there is the streamed information, such as radio, music, etc.

The sound can also be directional if the vehicle is equipped with multiple speakers systems. The driver hears the sound coming from a particular direction. This can be used to attract their attention to a particular location. This is used in some parking assistance systems to locate the unseen obstacles.

Furthermore, aural information do not require any scanning: they can be detected from any direction. However they are transient, and requires attention before they disappear. They are harder to focus, but this can be helped by setting different pitches, intensities (volume), or semantic content [8]. As the human earlid cannot be closed, sound is generally the best interface to present warnings.

The haptic interfaces are a recent development and not yet widely used. Generally these belong to two groups:

- vibrations, such as for example certain lane warning systems: the steering wheel vibrates when the vehicle crosses a line. It simulates the rumble strip present on certain roads.
- force-feedback, e.g. in certain intelligent speed-adaptation systems, where the throttle pedal resists slightly if the vehicle is about to go over the speed limit.

To conclude this section, it must be noted that, in case of cross-modal interactions, visual information is usually processed before aural information. Proprioceptive information tends to be processed last. As Wickens [8] points out, this is what happens when, seating in a train and looking at another departing, one can feel oneself moving; Proprioceptive information state that there is no movement, but this is discarded because there is visual information about movement.

## **2.2 Advanced Driver Assistance Systems**

Advanced Driver Assistance Systems (ADAS) have been studied for the last 15 years, and today are beginning to appear in common vehicles. They aim at providing assistance to the driver, by informing them about the car, the road, or any potential hazard, or by providing an active assistance, such as emergency braking. Two European projects, COMUNICAR [9] and STARDUST [10], both review the state-of-the-art. The ADAS can be grouped into five categories, lateral control systems, longitudinal control systems, reversing or parking aids, vision enhancements systems, and intelligent speed adaptation. An overview of these systems is presented hereafter.

### **2.2.1 Lateral control systems**

The lateral control systems monitor the sides of the vehicle, and possibly take action to prevent a collision. Three groups of systems are currently being developed:

- The lane keeping and warning systems help the vehicle to stay in lane, preventing drowsy or inattentive drivers to cross the road lines and hit an obstacle. Such systems have been demonstrated within several European projects. The lane *warning* systems alert the driver in case of lane departure; the lane *keeping* systems correct the trajectory of the vehicle. They generally use image analysis to detect the lines on the road. Such system has been demonstrated in the ARGO vehicle [11].
- The blind spot monitoring systems detect the presence of overtaking vehicles, and alert the driver. in the LACOS project [12], it is based on a camera and a RADAR.
- Side-obstacle warning systems use cameras or RADAR to detect obstacles at the sides of the vehicle. Such a system was demonstrated in the DaimlerChrysler VITA II vehicle, in the Prometheus project [13].

### 2.2.2 Longitudinal control systems

Generally, the longitudinal control systems will monitor the situation in the front and rear of the vehicle, and act upon the throttle and the brakes if necessary. Five groups of systems are distinguished:

- Adaptive Cruise Control (ACC) or distance keeping systems have been studied for a long time and were first introduced in the Prometheus project. A sensor, such as Radio Detection And Ranging (RADAR), Light Detection And Ranging (LIDAR), etc., measures the distance between the host vehicle and any obstacles ahead. Traditional cruise control systems maintain the speed set by the driver by acting upon the throttle, whereas ACCs also use the brake and other vehicle's dynamics parameters of the vehicle to decelerate if an obstacle is detected.
- The forward collision warning and avoidance system alerts the driver if an obstacle in front of the car is detected, allowing them to take evasive action. It might apply the brakes if the driver is not reacting. Generally they use a radar sometimes coupled with cameras.

- Intersection collision warning systems communicate with the infrastructure to detect vehicles crossing an intersection.
- So called ‘Stop and Go’ systems are designed for intense urban traffic. They allow the vehicle to stop and depart at low speed, without any intervention from the driver, by following the vehicle in front. They are built on the same premise as the ACC.
- Pedestrian detection systems alert the driver if a pedestrian or a vulnerable object enters the path of the vehicle. Different technologies are used, but the two main ones are laser and stereovision.

### **2.2.3 Reversing/parking aids**

The reversing and parking aids systems aim at providing help to the driver at low speed, for example for manoeuvring operations.

- The reversing aids are composed of a camera looking at the rear and a display mounted on a panel. They allow the driver to have a better view of what stands behind his vehicle.
- Parking aids give an estimation of the distance between the vehicle’s bumpers and the obstacles near them. Generally they use ultrasonic sensors.

### **2.2.4 Vision enhancement systems**

Light and weather conditions can badly impair the ability of the driver to detect potential hazards on the road. Automotive manufacturers and researchers, including Cranfield University [14], have developed night vision systems, generally based on infrared images.

Two techniques are commonly used. The first one is based on near infrared images and requires illuminating the objects on the road with a infrared light beam; after processing, the resulting image shows enlightened objects. The second technique is based on far infrared video images, which give a thermal map of the environment. It does not require any source of light. Pedestrians, animals, and running vehicle are hotter than the normal

environment and therefore are more visible in the image. Both techniques can be combined [15], and the resulting can be presented via to the driver on a screen or a head-up display.

### **2.2.5 Intelligent speed adaptation**

The intelligent speed adaptation system aims at maintaining the speed of the vehicle below the authorised limit. It relies on a navigation system or a communication system to provide the local speed limit. This limit is notified to the driver via screen, sound, or a harder throttle pedal when it is reached.

## **2.3 In-vehicle information system**

The HVI must communicate the information produced by the assistance systems to the driver, as well as information related to car status, communication, and comfort and convenience.

### **2.3.1 Instrumentation**

The basis of the vehicle-to-driver communication is the instrumentation panel. It provides information about the car dynamics, such as speed, engine RPM, gear, etc.; car status: lights, indicators, and defects or faults such as brake or temperature. Part of this data is purely informative, but some of it is critical.

### **2.3.2 Communication**

The second group of in-car information systems is the communication systems that allow the driver to send and receive information to and from the ‘outside’ world. It includes mobile phone, messaging (e-mail SMS, fax), traffic information, Internet, or telematic services like remote car diagnosis or assistance calling.

### 2.3.3 Comfort and convenience

Current dashboards present information related to both comfort and convenience, for example sound (radio, CD), video (TV, DVD), and climate control.

Last but not least, navigations systems are now available on most modern cars. They give directions and position using a map and a Global Positioning System (GPS) receiver. They can also be coupled with external sources such as traffic and map updates.

## 2.4 Information management systems

Information overload is a problem that the scientific community has been increasingly aware of, over the past few years. Abernethy [16] and Hoekstra [17] outline the potential safety risks linked to information overload. For the last two decades more information has been made available to the driver; today it is possible to have a ‘virtual office’ in a vehicle, in addition to the traditional ‘entertainment system’. Abernethy cites studies showing that using a mobile while driving could increase the risk of having an accident up to 300%. These two articles stress that the limited capabilities of the driver must be taken into account.

The military industry is very active in developing information management systems that avoid user overload, especially for command and control centres [18] or fighter aircrafts [19, 20]. These two last papers give recommendations for pilot-vehicle interface design. Information presentation must be pertinent, timely, flexible, responsive, predictable and intuitive. It must be founded on ‘a coherent model of human capabilities and limitations’.

More recently, information management was applied to the automotive domain. Cha et al. [21] propose a survey-based study to prioritise information to be presented to the driver. In [22], Spelt et al. describe an in-vehicle information management system test platform, and enumerate the tasks that such a system must perform: only present information needed or wanted by the driver; merge information from different sources; use presentation media that take into account the ambient conditions; and display information in a



way that gives the driver enough time to interact with it. However, their paper does not describe any implementation of these recommendations. The IN-ARTE project [23], however, aims at developing an ‘open architecture for the integration of multiple on-board support systems’. IN-ARTE is a three years project funded by the European Union. The paper focus on the presentation of five IN-ARTE functions: front obstacle approach, curve approach, lane or road departure risk, lane change, speed violation limit and traffic sign approach. These messages are statically associated to a mode of presentation, depending on their criticality. The information is ranked and using a criteria called the activation distance, based on vehicle and obstacles parameters, such as speed, acceleration, and distance.

To conclude this section, the very recent CoDrive project [24] must be mentioned. CoDrive was developed by the TNO research institute. It started in January 2001 and finished in January 2004, as the present thesis was being written up. CoDrive aims, as this work does, to prioritise and present information to the driver. An Human Machine Interface (HMI) framework was developed, comprising a prioritising mechanism, a workload estimator, a set of design rules, and a basic display layout.

In the presented implementation, the allowable workload is estimated from the velocity and the road type, given by the comparison of the GPS position and a digital map. For example, if the speed is below 30kph and the road is a motorway, then the allowable workload is estimated as ‘low’.

The prioritisation mechanism relies on services and an auctioneer. Services provide information messages associated with a priority index and a workload index and submit them to the auctioneer. The priority index reflects the urgency of the information, or its criticality. The workload index depends on the complexity of the interface. If the workload is higher than the allowable workload, the message is rejected; the service can select a presentation media that is less demanding and send the message again. The messages are then sorted and presented according to their priority index.

Both the work presented in this thesis and the Codrive project study the problem of in-car information prioritisation and presentation, and propose to prioritise information according to the criticality of the messages and the ability of the driver to process them. Both rely on a central system to filter

and decide of the presentation of information.

However there are many differences. In CoDrive, information prioritisation is only based on message urgency; in this work it is preferred to consider the risk of not presenting the messages, and the *relevance* of the message to the environment, which is a novel concept. To efficiently present information, it is proposed here to characterise messages using the level of abstraction, the time constraints, the access frequency and the induced tasks; a workload model, based on Wickens' multiple resources theory, is designed. In CoDrive interfaces and messages are given static workload and demand values.

Moreover, CoDrive focus on the presentation of navigation information, whereas in this work, both theory and implementation are build for a heterogeneous in-car information system.

## 2.5 Data networks

### 2.5.1 Automotive

Two buses have been designed for automotive industry: CAN, and MOST.

CAN was designed for automotive industry by Bosch in the 1980's. It's a high-speed (2 Mbit/s) serial data network designed to exist in industrial environments. Its key features are:

- Robustness: includes extensive error checking mechanisms, and is designed to be able to continue communication even if one of the wires is broken, shorted to power or shorted to ground .
- Use Non Return to Zero (NRZ) bit encoding: 0 and 1 are represented by non-neutral values.
- Data messages do not contain any address but a unique content identifier (e.g. revolutions per minutes, oil temperature). This identifier is also a priority index.

CAN is used to transport safety-related data by the automotive industry.

MOST [25] was designed by a consortium including Audi, BMW, Daimler-Chrysler, Harman-Becker and Oasis Silicon Systems to provide a common

high-speed bus and simplify the process of integrating multimedia devices in vehicles. MOST bus is implemented as a 25 Mbit/s fibre-optic network. It is slower than IEEE-1394 and with the development of an automotive-oriented standard (IDB-1394) based on IEEE-1394b and supported by the powerful ITS Data Bus (IDB) group, MOST's chances to be implemented are quite poor. Furthermore, IEEE-1394 has the advantage that it is already known and used by the consumer electronics industry.

### **2.5.2 IEEE 1394**

The development of Firewire began in the mid 1980s by Apple Computer. Other manufacturers became interested in the project and formed a working committee to create a standard on this architecture. The resulting specifications were submitted to IEEE (Institute of Electric and Electronic Engineers) and the IEEE 1394-1995 was adopted [26].

The IEEE 1394 bus is a serial bus with the following main benefits:

- Inexpensive compared to parallel buses
- Easy to use: hot Plug And Plug support, up to 63 nodes on the same bus
- No need of host processor
- High speed and scalable performance
- Support for isochronous applications
- Support for both backplane and cable implementation.

The IEEE-1394 specification was released in 1995, and included the basic 1394 functionalities and communications capability up to 400 Mbps.

The IEEE-1394a (2000) was a revised version, which provided, among others, asynchronous streaming.

The IEEE-1394b brings asynchronous streams, additional media interfaces (UPT5, Plastic Optical Fibre), longer distances (up to 100 m) and higher data speed (up to 3200 Mbps).

## PC Architecture

IEEE-1394 was designed to be integrated into the existing PC architecture. As shown in Figure 2.1, PC Architecture is basically composed of a CPU (Central Processing Unit), the 'brain' of the computer, a host bridge which links the processor to the rest of the system, memory, a graphic card linked to the processor by an Accelerated Graphic Port (AGP) bus, and other devices like keyboard, mouse, network interface card, hard disk drives connected to the processor via the PCI (Peripheral Component Interconnect) bus.

A 1394 bus can be connected to the PCI bus via a PCI/1394 bridge.

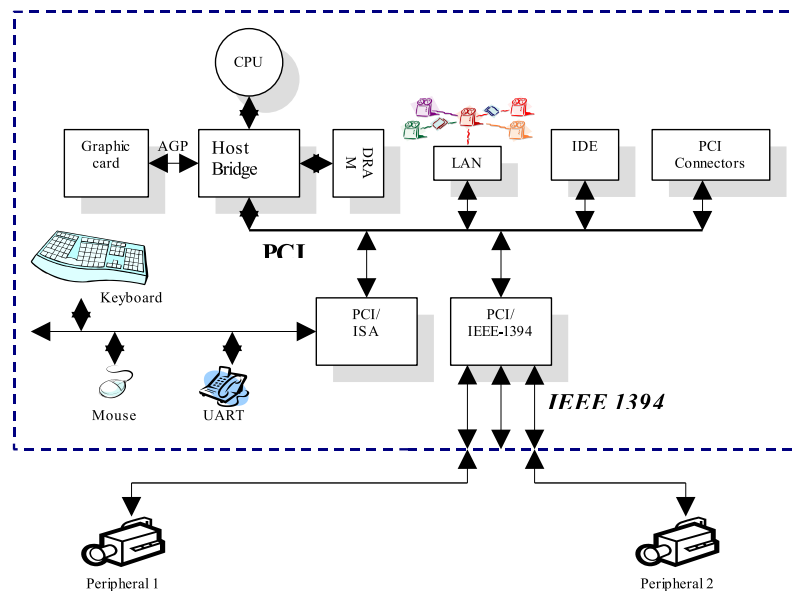


Figure 2.1: IEEE-1394 PC architecture

## Topology

IEEE 1394 uses a tree architecture, as shown in Figure 2.2. An interface node (a 1394 device such as a PC, a camera, a speaker) can be composed of multiple ports.

A maximum of 63 nodes can share the same bus, whereas 1023 buses can exist within the same system. IEEE-1394 uses point-to-point-transmission:

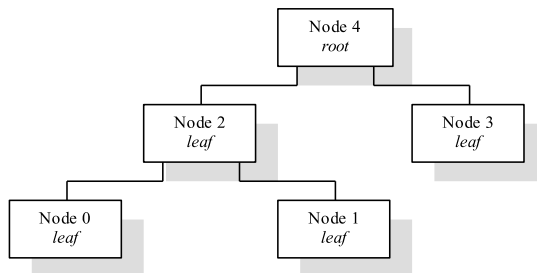


Figure 2.2: IEEE-1394 tree architecture

when a node receives a packet to one port, it retransmits the packet to the other ports.

### Asynchronous and isochronous communications

A synchronous data transfer is when the data is sent at regular intervals with a constant data length. This mode is used when a constant transfer rate is needed, for example for video or sound, which require real-time processing.

At the opposite, a data transfer is called asynchronous when no constant rate is needed. This mode will be used to transfer TCP/IP packets, for example.

A benefit of the IEEE-1394 bus is its support both asynchronous and isochronous transfers modes. In the example, shown in Figure 2.3, nodes  $N_1$  and  $N_2$  need to isochronous communication. They will first contact the isochronous transaction manager (IRM) node to ask for bandwidth allocation. A new channel will be allocated if sufficient bandwidth exists. At each bus cycle ( $125 \mu s$ ), isochronous data is sent, and the remaining space is used for the asynchronous data. Isochronous will use a maximum of 80% of the bus cycle ( $100 \mu s$  of  $125 \mu s$ ).

Asynchronous transfers have a variable length; they occupy the free bandwidth left after isochronous communications. An acknowledgement is sent by the receiver for every packet, and therefore guarantees that the packet has been received.

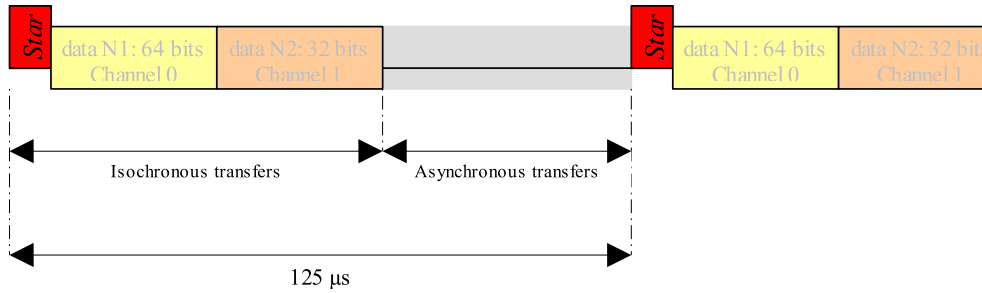


Figure 2.3: IEEE-1394 data frame

## Protocol

The IEEE-1394 standard is based on 3 layers, as illustrated in Figure 2.4 below.

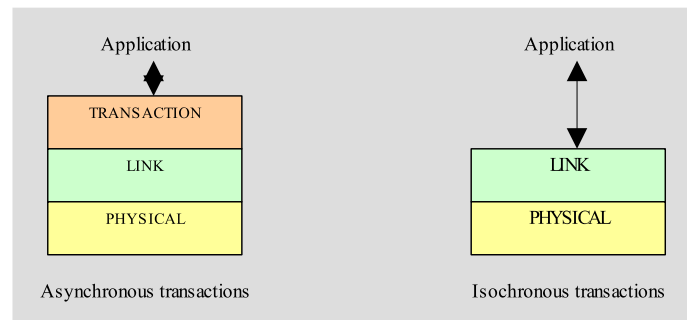


Figure 2.4: IEEE-1394 network layers

The transaction layer implements the request-response protocol required to conform to the ISO/IEC 13213:1994 [27] standard Control and Status Register (CSR) architecture for microcomputer buses (read, write and lock).

The link layer supplies an acknowledged datagram, a one-way data transfer with request confirmation, to the transaction layer. The link layer handles all packet transmission and reception responsibilities, plus the provision of cycle control for isochronous channels.

The physical layer provides the initialisation and arbitration services necessary to assure that only one node at a time is sending data and to translate the serial bus data stream and signal levels to those required by the link layer.

## 2.6 Driving environment recognition

Since the driver's behaviour is heavily influenced by the external conditions, it is necessary to characterise the driving environment. In a urban situation, with cars and pedestrians around, one will not drive as on a motorway or on a country road.

Studies sort the driving environment according to the road and traffic type. Engström and Victor [28] worked on real-time recognition of real-time driving patterns. They focus on large scale driving patterns, rather than small scale events, such as overtaking manoeuvres or lane exceedence, and applied their work to the detection of four different driving environments (highway, main road, suburban and city driving), with a accuracy of 90.6%. A statistical pattern recognition framework is employed, implemented by means of feedforward neural networks.

Löfsted and Svensson [29] detect the current driving situation. In the proposed implementation, five situation are identified: parking, traffic jam, city traffic, suburban traffic, main road and motorway. The classification is performed in real time by a a fuzzy expert system, taking four CAN bus channels in input.

Hauptman et al. [30] took a neuro-fuzzy approach to classify the driving environment. From the data produced by seven sensors (wheel speed, brake, motor revolution speed, lateral acceleration, torque, gear and driving style), the system computes the current driving environment: Autobahn, highway, road, city, and stop. The manually optimised system achieves 74% of correct identifications.

Qiao et al. [31] also describe a neuro-fuzzy system for recognition of driving environment (town, congestion, ascent, descent, highway). It uses a self-supervised algorithm. Firstly, six sensors out of the thirteen available were selected, based on pre-processing results. Seven extraction variables were then constructed from the sensors measurements.

In [32], Yamada et al. focus on detecting water and snow on the road, by analysing images of the road. Five kinds of road conditions are discriminated: dry, wet, slushy, icy and snowy. For each pixel of the image, the ratio of horizontal to vertical polarisation image intensity gives features related to

water, whereas snow-related features are extracted by texture analysis. The authors report an accuracy rate between 83 and 100%. The advantage of this method is that it provides spatial detection, as opposed to conventional methods using optical or ultrasonic sensors, that only give information at one point.

## 2.7 Human-vehicle interface and safety

Detecting the driving environment is necessary to make an in-car information system adapt to its surrounding and perform some tasks for the driver. For example, the rain or light sensor can turn on the windscreen wipers or headlights automatically.

These new systems have one goal: help the driver drive, by lowering the number of tasks they have to perform. Indeed, in the past years numerous studies have described the problem of driver overload and driver distraction. In [2], Stutts et al. explain that 12.9% of police-reported accidents in the US involve some form of driver inattention. Table 2.1 shows the specific sources of distractions among these drivers.

Outside person, object or event	29.4%
Adjusting radio, cassette, CD	11.4%
Other occupant in the vehicle	10.9%
Moving object in the vehicle	4.3%
Other device/object brought into vehicle	2.9%
Adjusting vehicle/climate control	2.8%
Eating or drinking	1.7 %
Using/dialling cell phone	1.5%
Smoking related	0.9 %
Other distraction	25.6%
Unknown distraction	8.6%

Table 2.1: Percentages of drivers for different types of distractions [2]

These results demonstrate that the human-vehicle interface is a cause of accidents. It is important to note that this report was published in 2001,



with data gathered between 1995 and 1999; at this time in-car interfaces were embedding fewer and less complex information systems.

Not only manual operation is a cause of danger in vehicle. Harbluck [33] demonstrates that under increased cognitive load, drivers spend less time checking instruments and rear view mirror, and tend to react slower to incidents, reducing driving safety.

Information overload can therefore make driving more dangerous. Driver workload has been studied and can be modeled, as presented in the next section.

## 2.8 Driver workload

As De Waard [1] points out, the notion of mental workload was first introduced by Broadbent [34]. He and his colleagues built the theory that people have limited processing capabilities, and referred to as *resources*. Wickens [35] defines the resources as ‘the mental effort supplied to improve efficiency’, whereas the *capacity* is the upper limit of the processing capabilities. Resources are ‘scarce’ and ‘can be deployed under voluntary control’. For Kahneman [36], the capacity is not fixed, and can increase under load.

Wickens introduced in [37] a multiple resources theory, in which he explains that operator limitations depend on the mode of presentation of the information, its encoding and the stage of processing of the information.

The mode can be visual, aural or haptic. Aural and haptic information tend to be processed before the visual information: a loud noise will attract the attention of a person, as it indicates a major change of the environment. Aural and visual information are very different. Aural information is ubiquitous, and is generally of limited length; visual information requires the person to direct their gaze towards the source of the information, and the message can generally be observed again.

The code of the message is either verbal or spatial. Most of the information we receive is spatial. Looking at the road gives an idea of the distance, the speed and the direction of the objects. Similarly, we perceive the analog displays (speedometer, oil level) of a vehicle as spatial information. Conversely, the verbal information is expressed using language. Speech, numeric

displays are verbal. Studies have shown that these two modes involve different physical resources. The right hemisphere of the brain is associated with the perception of spatial information, whereas the left hemisphere is thought to process the verbal information.

Three stages of information processing are generally considered: encoding, central processing and responding. It has been proved that the two former stages do not use the same resources as the latter one [37]. For example, tracking the vehicle in front (perception) does not prevent the driver to brake or accelerate (response).

Wickens proposes therefore a multiple resource theory in which he states that people can perform concurrent tasks better if they do not require the same resources. Resources are organised in a three dimensional space, as shown on Figure 2.5. The dimensions are the processing stages (encoding and central processing, responding), the perceptual modalities (visual, aural), and the processing codes (spatial or verbal). If the two tasks use different resources, time sharing between them will be more efficient, and changes in the difficulty of one task will be less likely to influence the performance of the other.

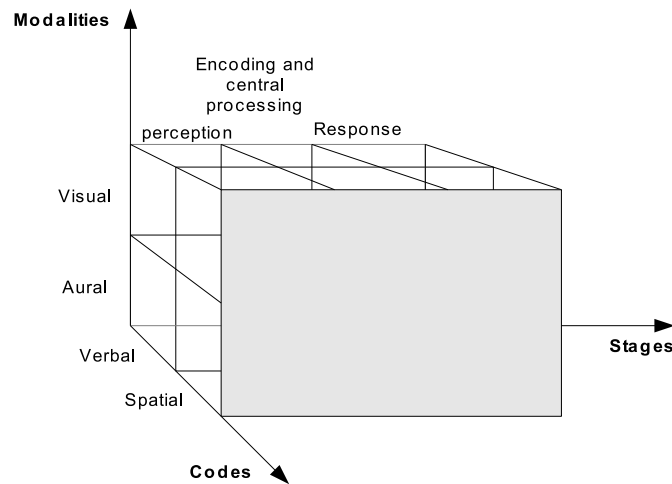


Figure 2.5: Multiple resources theory

This model is further extended in [8]. In [38] Wickens adds the dichotomy

between ambient and foveal vision, and proposes a computational model.

It is also worthwhile mentioning other theories such as computational processes and energetical mechanisms. However in this work, the workload model will be mainly build on Wickens' multiple resources theory.

De Waard [1] makes the point that it is important to distinguish the terms of workload, demand, and effort. The task demand is determined by goals that have to be reached by performance. Workload is the reaction to this demand. The effort is the voluntary mobilisation of resources, and is actually composed of task-related effort, which is exerted in case of controlled information processing, and a state-related effort, which is exerted to maintain an optimal state for task performance. An illustration of these notions will be given later in this section.

De Waard merges and extends theories describing the relation between task performance, workload and effort. Figure 2.6 shows how he separates the demand into four regions. The first region (D) is the *deactivation* region. As the demands starts to increase, the operator has to work hard to enhance the task performance with low resources. The second region (A1) shows the state related effort: the operator mobilises enough resources to respond optimally to the task, as shown in region A2. As the task demand continues to increase, the operator has to work harder to maintain performance; that is the task-related effort region (A3). In region B it no longer possible to maintain performance, as the operator capacity has been reached, until he gets finally overloaded, in region C.

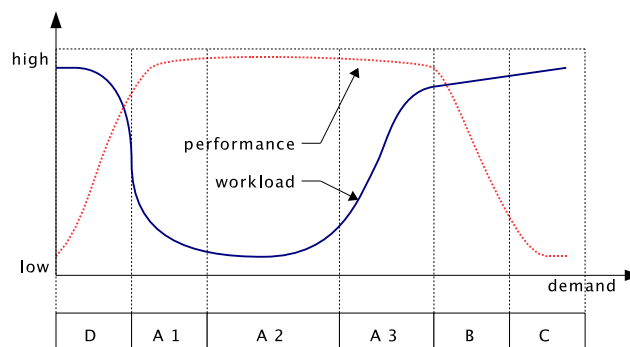


Figure 2.6: Relation between task demand, performance and workload [1]

The methods for measuring workload are numerous. Since the purpose of this study is to consider the driver workload as another prioritisation criteria, the intrusive measurement methods will be discarded. It is necessary to have an in-field and real-time indicator of driver workload. Among the non-intrusive measurement methods is steering entropy [39].

## 2.9 Fuzzy cognitive maps

Fuzzy Cognitive Maps are an extension of cognitive maps and were first introduced by Kosko [40]. They allow representation of causal relationships between facts. These relations can be either positive or negative, causing increase or decrease of value. Both relation and facts can have fuzzy values.

Fuzzy logic was first formalised in 1965 by Zadeh [41]. In classic binary logic, facts are either true or false; in fuzzy logic, they can be true and false simultaneously, to a certain extent. Fuzzy logic admits the ‘grayness’ between black and white.

In fuzzy cognitive maps, causal relationships between facts are weighted, whereas facts can be represented by degrees of truth. For example, consider the statement ‘when it rains, the road is slippery’. In binary logic, both facts, rain and road adherence, would be represented by binary values, 0 or 1, and the relationship between the two would have a value of  $-1$ .

However if it only rains a little, the adherence of the road will still be good; if it rains a lot, the road adherence will be much worse. Similarly, some roads are better drained than others; the rain will have less effect on road adherence. Binary values cannot render these relationships properly: either the road will be slippery, or it will not. Using fuzzy values, the level of truth of the facts and the strength of the causal relationships enable to create a more realistic model.

FCMs are composed of concepts linked by weighted relations. As Figure 2.7 illustrates, every relation  $R_{ij}$  between two concepts  $C_i$  and  $C_j$  has a weight  $w_{ij}$ , and every concept  $C_i$  has a threshold function  $f_i$ . The value  $x_i$  of a concept  $C_i$  is calculated with the following formula:

$$x_i = f_i \left( \sum_j w_{ji} x_j \right). \quad (2.1)$$

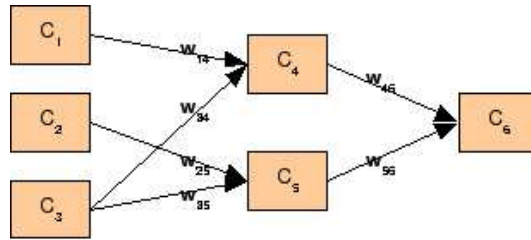


Figure 2.7: Fuzzy cognitive map

A weight value of  $+1$  defines a positive relationship ( $x_i$  increases), whereas a value a  $-1$  depicts a negative relationship ( $x_i$  decreases). The value  $0$  means that no relation exists between the two concepts.

FCMs have been applied to many domains, such as the modelling and control of dynamic systems [42, 43] and modelling of virtual worlds [44]. They have advantages over other techniques, such as artificial neural networks.

Artificial neural networks mimic the structure and the behaviour of ‘biological’ neural networks, as explained by Callan [45]. They are composed of units, or neurons, connected by weights. The inputs of one unit are generally combined by summing their values multiplied by the weights, and the output is calculated with an activation function, that can return a real number or a discrete value such as  $\{0, 1\}$ .

This is similar, in structure, to the fuzzy cognitive maps. However, in the neural networks, there is no meaning associated with units and weights, whereas in a fuzzy cognitive map, the values associated with the concepts represent their degree of truth, and the weights represent the strength of a causal relationship.

Moreover, in an artificial neural network, the values of the weights are not set manually. Instead, the network is trained with several data sets. A program adjust the weights automatically. The longer the training is, the more accurate the results are.

Therefore, artificial neural networks are sometimes compared to black boxes. The designer of the system does not have full control of the behaviour

of the system, which might be unwelcome in some areas, such as commercial safety systems.

Fuzzy cognitive maps, compared to other techniques, have several benefits. They represent knowledge in human-readable form, as they are an exact image of the causal relationships between concepts. Their simple structure make them computationally inexpensive, compared to, for example, expert systems. It is also possible to combine knowledge from different experts by merging the associated FCMs. These features make them more robust, easier to build, maintain, and usable in real-time environments.

# Chapter 3

## Information prioritisation

Information management systems consist of sequences of decision-making procedures to present selected information, in chosen means, at the right moment. As discussed in the previous chapter, in-car information systems are either essential to safety, or provide a desirable service to the driver. However, due to the limits of an individual capabilities, providing information without careful selection can often cause ‘overload’. Information prioritisation consist ranks information by order of importance, according to the dynamic characteristics of information messages.

In current in-vehicle information systems, information is usually statically associated to an interface and a priority rank. For example, a navigation direction will always be presented on the central console screen and will always prevail on the radio. The problem is that, with an increasing number of embedded systems, messages must share common interfaces, and designing the prioritisation rules becomes much more complex. What happens if the phone rings, with the radio on, while the driver changes lane, following a direction from the navigation system?

The literature review showed different approaches to this problem. They all agree that the safety criticality of information must be taken into account. IN-ARTE, for example, links the priority of the information to the temporal distance of the danger. However in this thesis the whole in-car system is considered; therefore classification criteria that are valid for every information must be extracted, not only for safety-related information. It is proposed in

this chapter to group the information sources into risk classes.

Furthermore, it is acknowledged that overload often occurs because non-essential information are presented. This chapter discusses two major characteristics of information, namely risk and relevance. These two measure allow us to compute an importance index, that is used for the prioritisation. Then a model of the environment and its relationship with the information is built.

### 3.1 Risk

Information has meaning for the driver. Knowing the source of a message, they are able to decide how much attention they must give: if a warning is issued from a crash-avoidance system, it is quite sensible to consider it more important than direction from a navigation system. It is propose to sort the sources in four categories, depending on the risk they represent:

- Systems related to safety: they aim at alerting the driver of potential hazards. Included in this group, are the lane warning system, the blind spot monitoring system, the collision warning system and the pedestrian detection system;
- Information related to the driving task. These help the driver to perform its primary task — driving. This group comprises the active cruise control, stop and go, vision enhancement and navigation systems;
- Communication systems: phone, e-mail;
- Entertainment and comfort: radio, music, video, air conditioning.

Every information message carries an intrinsic risk, which is related to its non-presentation. This value must be taken into account in the prioritisation strategy. For example, if information produced by the radio is not presented to the driver, it might have no consequences. Conversely, if information is related to an obstacle in the path, and if the driver is not aware of it, it can result in a crash.



It is possible to sort the information sources  $S_i$  into static *risk groups*. Four groups are proposed: safety-related information, driving task information, communication, confort and entertainment. Table 3.1 gives an example of such classification.

<b>Group</b>	<b>Information system</b>
Safety	Lane warning, Blind spot monitoring, Collision warning and avoidance
Driving	Stop and go, reversing and parking aid, vision enhancement, navigation, instrumentation
Communication	Phone, Internet
Confort and entertainment	Air conditioning interface, radio, music

Table 3.1: Risk groups and information sources

Every group  $G$  is associated to a risk interval  $[\frac{k}{n}, \frac{k+1}{n}]$  where  $k, n \in \mathbb{N}^*$ ,  $n$  being the total number of groups and  $k$  the index of the group in the list sorted by the order of risk.

Within each risk group, information sources are also sorted. The value  $r(S_i) \in [0, 1]$  describe their membership of the group. For example, presenting the information from ‘instrumentation’ source is more critical than presenting navigation directions. The success of the driving task depends more on the former, and less on the latter. However, both sources belong to the ‘driving’ risk group. The group membership  $r$  allows us to differentiate them within the risk group:

$$r(\text{instrumentation}) > r(\text{navigation})$$

The risk classification is illustrated in Figure 3.1 hereafter.

## 3.2 Relevance

In modern in-car information system, as in the modern world in general, there is more information available to the driver than he is actually able to

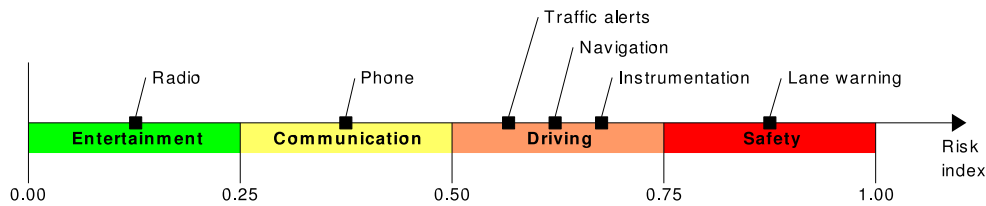


Figure 3.1: Example of risk values

process. Some decades ago, getting information required action and minimal effort: people had to go to the store to receive information about products, buy newspapers, and queue in front on the phonebooth to be able to call.

Today, the flow is reversed: people do not fetch information anymore, information comes to them. Information systems tends to be more and more ‘talkative’. Many times at conferences, speakers had to interrupt their presentations to close an operating system message window, declining to ‘install the update now’ or refusing ‘help on this feature’. These messages were irrelevant to the presentation.

Information systems should only present essential information, especially when the addressee has limited processing capabilities. In [46], this problem was addressed for transmission of emotions over the Internet. There are several means of communicating in real-time over Internet. For example the ‘chat’ programs allow to send and receives text messages. Emotions are usually rendered as ‘smileys’, such as :- ) for a smile, or ‘emoticons’, small images representing faces. A short survey showed that people found that they did not convey emotions accurately. It is also possible to use video; however people want often to preserve their visual anonymity. In this paper, it is proposed to transmit only the essential part of the image, the features that compose the face expression. Using image processing techniques, the most meaningful facial features are extracted, and converted to a vector format. The result is sent over the Internet and rendered on the receiver screen. The full text of this article is available in Appendix A.

Similarly, in a vehicle, information presentation must be consistent with the driver needs. The driving environment makes some messages more relevant in particular situations. For example, traffic alerts are more relevant

when they are related to the route the vehicle is following; lane departure warnings are more relevant when the manoeuvre is unintentional, caused by driver's fatigue.

A key point of the work presented in this thesis is to consider the relevance of the information and use it as a classification criteria. Depending on the current driving situation, the relevance of an ADAS varies; in some environment, an assistance system can even become useless. For example, a lane departure warning system alerts the driver if the vehicle crosses the limits of the lane. It is very useful when the manoeuvre is not intentional, and caused by drowsiness or distraction. In this case the warning must be strong and immediate. However, there are situations where these alerts are not needed, and sometimes even not wanted, such as overtaking manoeuvres or moving to an exit lane.

Before presenting a message to the driver, the information management system must ensure that it is relevant to the environment. Table 3.2 shows a list of system and associate situations where they are of particular interest.

<b>Information system</b>	<b>Relevant when...</b>
Lane warning	Motorway and roads where driver is likely to get drowsy; NOT relevant in city, too many lane changes
Blind spot monitoring	When car is going to change lane (indicator, lateral displacement)
Collision warning and avoidance	NOT when parking
Stop and go	When stop and go are frequent (city, car jam)
Reversing/parking aid	Reversing / parking
Vision enhancement	Night, low visibility
Traffic information	Congestion on route

Table 3.2: ADAS and relevance to the environment

Relevance is a vague concept. It depends on multiple environmental factors, that may be difficult to model and measure. For example, the relevance

of the lane departure warning system is higher if the manoeuvre is unintentional. The manoeuvre is more likely to be unintentional if the driver is tired, or is distracted. In turn, night and high driving time may favour driver tiredness.

It is proposed to use the Fuzzy Cognitive Maps to model the effects of environmental constraints on the relevance of information. Concepts represent statements, such as ‘received traffic alert is related to route’. The value of the concepts represents in which extent this statement is true: 1 for totally true,  $-1$  if its opposite is true. A value of 0 is the neutral point, where the statement is as much true as it is untrue.

In this study, the in-car information system is regarded as composed of information sources, concepts and sensors. The sensors provide raw ‘meaningless’ information, such as yaw rate or steering wheel angle. The information sources produce information that can be presented to the driver. ADAS are information sources. The concepts are intermediary values that represent how true is a statement; they are used to compute the relevance  $\rho(S_i)$  of an information source  $S_i$ :

$$\rho(S_i) \in [-1, 1]. \quad (3.1)$$

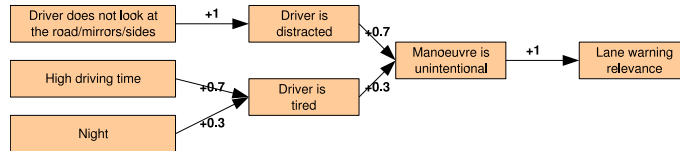


Figure 3.2: Simple FCM for lane departure warning relevance

Figure 3.2 shows a simple FCM representing environment influence over lane departure warnings relevance. ‘Night’, ‘High driving time’, and ‘Driver does not look at the road/mirror/sides’ are concepts that can be measured. They have positive influence on the concepts ‘Driver is distracted’ and ‘Driver is tired’: the longest the driving time is, the more tired the driver gets. The ‘+’ sign is associated with a weight, that gives the extent of the effect.

For example, if the driver seems quite distracted (0.8), is driving for a reasonable amount of time (0.5), at night (0.7), the relevance of the lane departure warnings is:

$$\begin{aligned}\rho(\text{lw}) &= 1.0 \times (0.7 \times (1.0 \times 0.8) + 0.3 \times (0.7 \times 0.5 + 0.3 \times 0.7)) \\ &\approx 0.728\end{aligned}$$

where the threshold function for every concept is taken as:  $f(x) = x$ .

In this situation, the lane warning system is therefore clearly relevant. It must be presented to the driver in a way that reflects this relevance, as explained in the next chapter.

### 3.2.1 Normalisation

To process data with the proposed FCM, it is first necessary to convert the sensor values to ‘truth’ values, included in the interval  $[-1, 1]$ .

For raw sensors, it can be done using a normalisation function, such as a step function or a parameterised sigmoid function, defined in Equation (3.2) and plotted in Figure 3.3.

$$T(t) = \frac{a}{1 + e^{-\alpha t}} + b, \quad a, b, \alpha \in \mathbb{R} \quad (3.2)$$

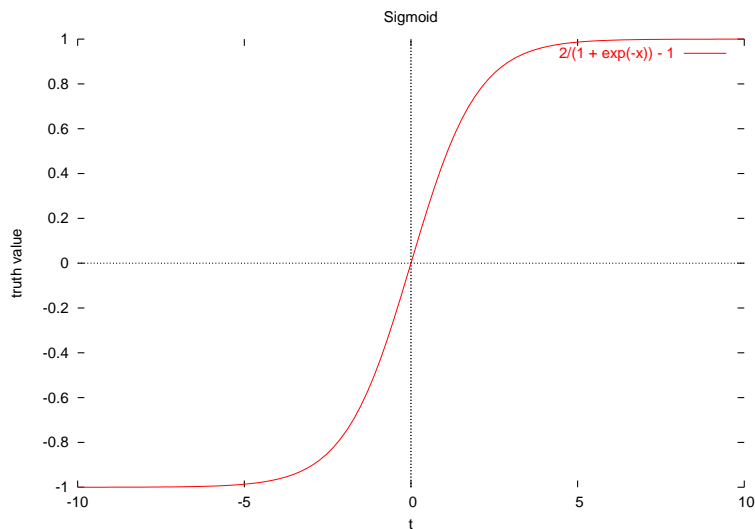


Figure 3.3: Sigmoid function for  $a = 2$ ,  $b = -1$  and  $\alpha = 1$

For example, the value of the ‘High driving time’ concept, mentioned earlier, can be computed from the current time, and the journey start time.

If it assumed that less than 60 minutes driving is not a long driving time, and above 60 minutes is,  $a$ ,  $b$  and  $\alpha$  must be chosen such as  $T(60) = 0$ , and  $T(t) > 0$  when  $t > 0$ .

In some cases, there is no sensor representing the environment parameters that influences information such as traffic, road type, etc. It is proposed the use of fuzzy rules, because they allow easy representation of such relations.

For example, inner-city traffic can be viewed as a situation where the number of stops is frequent and the speed is quite low [29]. In term of fuzzy rules, this would be described as:

if (frequency of stops) is high AND (average speed) is low THEN (relevance of stop and go system) is high

The definition of the linguistic variables, such as ‘high’ and ‘low’, can be extracted from experts, or by experiments. Figure 3.4 shows a possible membership function of the speed average.

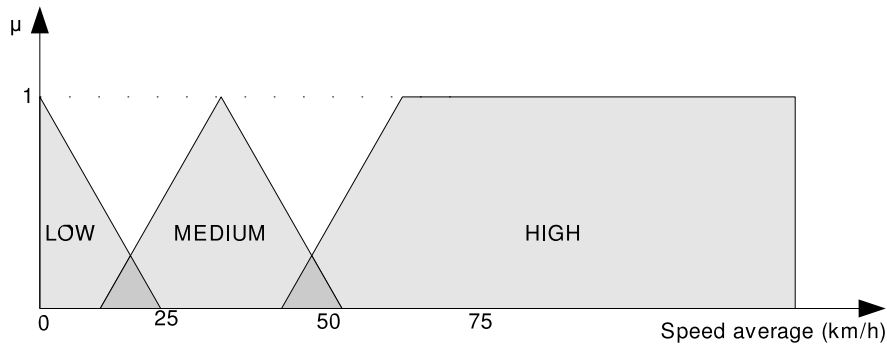


Figure 3.4: Speed average membership function

Fuzzy rules can be combined and a crisp relevance value is obtained by defuzzification; several methods exist, but the most common one is the centroid. It consists in computing the centre of gravity  $(x, y)$  of the combined fuzzy values.

$$x = \frac{1}{A} \iint x\sigma(x, y)dA \quad y = \frac{1}{A} \iint y\sigma(x, y)dA$$

where  $\sigma$  the density of the area and  $A$  is its surface:

$$A = \iint dx dy.$$

### 3.3 Importance

From the risk and relevance values, an importance value  $\pi(S_i)$  is calculated. It represents the overall importance of the information produced by the source  $S_i$ :

$$\pi(S_i) = \frac{1}{n} \left[ G(S_i) + \left( \frac{\alpha r(S_i) + \beta \rho(S_i)}{\alpha + \beta} \right) \right], \quad (3.3)$$

where  $\alpha, \beta \in \mathbb{R}^+$  are parameters that allow the balance between risk and relevance to be tuned,  $n$  is the total number of risk groups,  $G(S_i)$  the risk value of the group to which to source  $S_i$  belongs,  $r(S_i)$  its risk membership value, and  $\rho$  its relevance.

This importance index is used to prioritise information in the strategy discussed in the next chapter.

# Chapter 4

## Information presentation

There are numerous means of presenting information to the driver. It has been often debated, however, what information is needed and how it should be presented to the driver. The emphasis for this chapter is not placed on technological issues of information presentation, but on a strategy to propose and present information.

The driver is surrounded by information sources, such as warning systems, instrumentation, radio, etc. These sources produce information messages, that are communicated to the driver via interfaces. In the previous chapter, a set criteria was established to sort information, namely the relevance to the environment, and the risk associated to the non-presentation of the messages.

In current HVIs, a message is usually associated to a presentation media. For example, the navigation map is displayed on the central console screen, an incoming call is rendered as a ring tone, and the ‘empty tank’ warning is shown as an icon on the instrumentation panel. This presentation is efficient for in-car information systems where the number of devices, or information sources, is small. However, if the number increases, different sources must share common interfaces, and must compete for driver attention.

In the previous chapter, a technique was developed to rank information dynamically. In this chapter, a strategy is designed to decide how, and how many, information messages should be presented. Firstly are outlined the main rules that make a media more able to render a message than another. Secondly, these messages are studied in terms of workload, and model is built



to propose a strategy to avoid driver information overload.

## 4.1 Optimal interface choice

In chapter 2, different manners of presenting information are reviewed. They are summarised and illustrated in Figure 4.1.

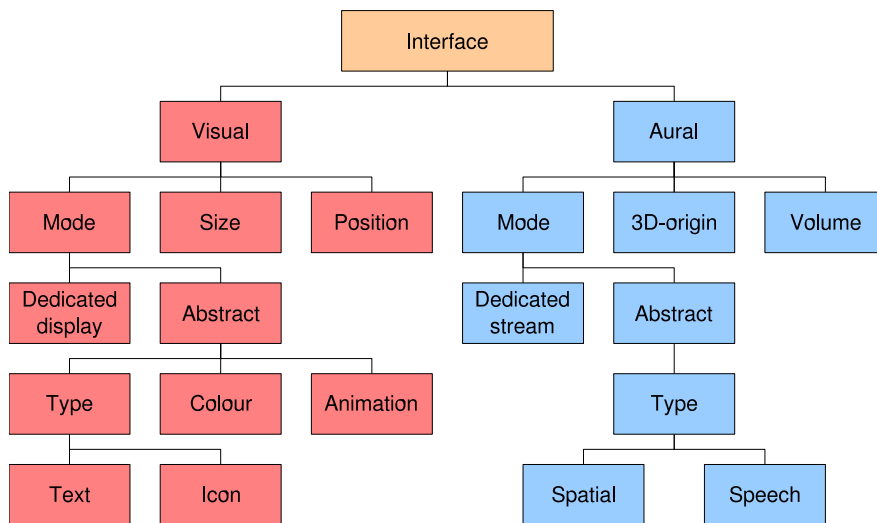


Figure 4.1: Presentation media

The previous chapter provides us with two important characteristics of information: its relevance to the environment, and its associated risk. The messages can be sorted according to their importance. Using this classification, an optimal interface must be allocated to each message, i.e. an interface that best conveys it, taking into account the constraints of the environment.

To decide of a presentation media, three others parameters must be considered: the nature of information, the time constraints, and the frequency information is going to be accessed.

### 4.1.1 Nature

Messages can be abstract or non-abstract. The raw data, such as radio or video, are non-abstract. In this case the information management system has very little control over them: they must have a dedicated interface.

Abstract information include the warning messages, navigation directions, etc. The system has full control of the presentation mode; it must communicate the meaning of the message to the driver. For example, a direction produced by a navigation system can be presented as text displayed on the central console panel, as speech, or as an icon, such as an arrow, displayed on the navigation screen.

The nature of information limits the number of compatible interfaces.

### **4.1.2 Time constraints**

An information message is generally only valid for a certain amount of time, and/or for a specific space location. If it is not delivered within this interval, it loses value. For example, in the case of a navigation system, a direction, such as ‘Turn right in 300 meters’ must be given before the related junction, within the right distance. For streamed, non-abstract information, such as radio or music, it is important that messages are delivered in a timely way. Otherwise the stream gets interrupted, and the information becomes meaningless.

The information management system must therefore take into account the timeliness of information, including duration and deadline. Urgent information, i.e. information that must be delivered in a very short time, must be presented in a way that attracts driver’s attention.

### **4.1.3 Access frequency**

The frequency of access to information is an essential parameter for selecting presentation media. Displays presenting frequently accessed information are usually positioned near the road scene, so the time required to switch from the display to the road is minimal. Aural interfaces are reserved to information that does not need to be re-accessed. Indeed, aural messages are transient by nature, and are usually delivered only once. Considering the access frequency of information constrains the modality and the position if the interface is visual.

Furthermore, the access frequency is not constant and depends on the environment. For example, the navigation map will be looked at more often

when the vehicle is approaching a junction, and its presentation must facilitate its access. Like the relevance, this can be modeled using Fuzzy Cognitive Maps; chapter 6 gives an example.

To conclude and summarise this section, the selection of a interface must take into account several parameters, namely its nature, its timeliness, and its access frequency. The selected interface must be compatible with the message; if the information is urgent the interface should attract driver’s attention; finally if it must be accessed frequently the interface should be positioned close to the road scene.

Figure 4.2 illustrates the characteristics of information that influence the choice of an interface. Note that the last item, the induced task, is related to the workload, discussed in the next section.

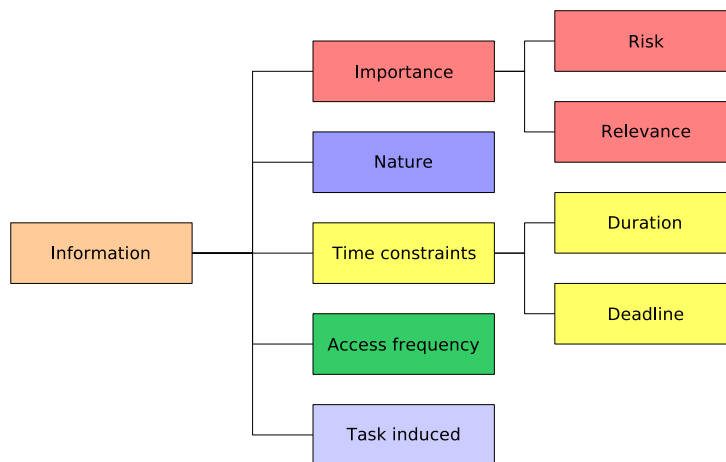


Figure 4.2: Information message characterisation

## 4.2 Workload

### 4.2.1 Definition

As seen in chapter 2, the driver cannot perform an unlimited number of tasks simultaneously. By estimating their workload it is possible to ensure that they do not exceed their capabilities, and maintain good overall performance.

The workload is not only created by information the driver receives, but also by the tasks they perform. Driving, for example, is a demanding task. The attention must be focused on the road, the instrumentation and the environment. The response is manual but complex, because it involves upper and lower body.

The driving environment influences the complexity of the task. An overtaking manoeuvre, for example, requires more effort than not overtaking. The perceptive demand is heavier, as the driver must estimate the speed and distance of the other vehicles; the response demand is also higher, as they must quickly increase the speed of the vehicle, whilst changing direction.

When the driver receives an information message, they have to process it and perform the relevant action. This action may only be perceptual, e.g. in the case of music. But in other cases, more driver resources are involved. For example, When a navigation information such as ‘Turn right in 300 meters’ is emitted, the demand is as follows:

- perceptual: read or hear the message
- cognitive: understand the message, link the information to the current driving environment and prepare manoeuvre.

This induced demand must be taken into account in the design prioritisation strategy: Only considering the perceptual load of the human-vehicle interface may lead to driver overload.

### **4.2.2 Modelling**

In this Section, a task is defined as the communication, including response, between information system and driver. According to Wickens’ multiple resources theory, every task performed by the driver requires the use of limited separate resources. Tasks can be executed simultaneously in a more efficient way if they use different resources.

As seen earlier, the resources can be viewed as a matrix of order 3, the dimensions of which are the code (verbal or spatial), the stage (perception, encoding or response), and the mode (auditory or visual).

The load rate of one resource is defined as  $l_{c,s,m} \in [0, 1]$  with  $c, m \in \{1, 2\}$ ,  $s \in \{1, 2, 3\}$ . A value of  $l_{c,s,m} = 0$  means that the resource is not used, whereas a value of  $l_{c,s,m} = 1$  indicates that this resource is fully allocated; more work would result in driver overload and decrease of task performance.  $\mathbf{L}$  is the load matrix.

Similarly, each task  $T$  can be associated with a demand matrix  $\mathbf{D}$ , of elements  $d_{c,s,m}$ , that represent which, and how much, resource is required for task success and good performance. For example, the task ‘driving on a quiet road’ requires mainly visual spatial perception, and manual response. The demand matrix would be of the form:

$$\mathbf{D}_{visual} = \begin{pmatrix} 0.5 & 0.1 & 0.5 \\ 0 & 0 & 0 \end{pmatrix} \quad \mathbf{D}_{auditory} = \begin{pmatrix} 0 & 0.1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Note that, as there is no distinction of mode in the cognitive stage, the values of auditory and visual cognitive demands are equal.

The values of the load matrix can be influenced by environmental factors. For the driving task, perception will be made more difficult when the weather is bad and the visibility is low; similarly, if the road is icy, the load rate of the response will increase. These relations can be properly described by fuzzy cognitive maps.

To prevent tasks interfering with each other, not only must the resources be kept under the maximum load limit, but also the model must take into account inter-resources interference. As Wickens [37, 8] explains, the more dimensions the resources share, the more they tend to conflict. In [38] he defines a conflict matrix, whose elements are determined arbitrarily. A different approach is presented here, based on the position of the resources in the 3D space.

Mathematically, the conflict value  $c$  between two single-resource tasks  $T_1$  and  $T_2$  is related to the interference  $\chi$  between the involved resource  $R_{ijk}$  and  $R_{mnp}$ . This can be computed from the normalised distance between two points of coordinates  $(ijk)$  and  $(mnp)$  in the 3D space:

$$c(T_1, T_2) = \chi(R_{ijk}, R_{mnp}),$$

and

$$\chi(R_{ijk}, R_{mnp}) = 1 - \frac{1}{\sqrt{6}} \|\overrightarrow{R_{ijk}R_{mnp}}\|. \quad (4.1)$$

For example, the conflict value between the two tasks ‘reading a book’ (verbal, perceptual, visual) and ‘listening to the news’ (verbal, perceptual, auditory) is given by computing the distance between the points of coordinates (1, 1, 1) and (1, 1, 2) :

$$\begin{aligned} c(T_1, T_2) &= 1 - \frac{1}{\sqrt{6}} \|\overrightarrow{R_{111}R_{112}}\| \\ &= 1 - \frac{1}{\sqrt{6}} \sqrt{(1-1)^2 + (1-1)^2 + (2-1)^2} \\ &\approx 0.592. \end{aligned}$$

Similarly, the tasks ‘reading a book’ (1, 1, 1) and ‘listening to music’ (1, 2, 2) have a conflict value of approximately 0.423.

Comparing these two values confirms that it is easier to listen to music than listen to the news while reading a book. This is consistent with the multiple resource model.

The load matrix for each task must take into account how much resources interfere. Each of its element depends on the values of the demand matrix:

$$l_{csm} = d_{csm} + \gamma_\chi \sum_{(ijk) \neq (csm)} \chi(R_{csm}, R_{ijk}) d_{i,j,k}, \quad (4.2)$$

where  $d_{csm}$  represents the task demand for the resource with code  $c$ , modality  $m$  and stage  $s$ , and  $\chi(R_{csm}, R_{ijk})$  represents the interference between the two resources, as defined in Equation (4.1). The parameter  $\gamma_\chi$  allows to adjust the effect of the interferences on the demand.

When different tasks  $T_i$  require common resources, the total demand is calculated by adding the demand matrices:

$$\mathbf{D}_{total} = \sum_i \mathbf{D}(T_i).$$

When estimating in real-time the driver’s workload, it is essential to distinguish aural from visual information. As discussed in chapter 2, people cannot filter out auditory information, as we are not equipped, sadly, with earlids. When a message is presented, it automatically creates a perceptual workload for the driver. Conversely, people can focus visual information. If it is not looked at, a visual interface does not create any workload. One must not forget, however, that people still can perceive information that is

not directly focused (ambient vision). If the interface presents information in a salient way, by the use of colours or animation, it may be distracting and may create a workload even if the driver does not look directly at it.

### 4.3 Strategy

Part of the problem studied in this work is to present information by optimal means. It has been demonstrated that information can be characterised with notions such as relevance to the current environment and risk related to the potential consequences of not presenting the information. These two notions can be combined to create an overall importance index.

Moreover, when an information message is presented, it generally induces a task for the driver to perform. For example, when the phone rings, it is likely that the driver is going to pick up and start a conversation. If the driver is not able to have a phone conversation at this moment, the message should not be presented, and the caller should be put on hold. To evaluate the capacity of the driver to process the message, the prioritisation system must take into account the current availability of the driver's resources defined by Wickens' model, the perceptual demand and cognitive demand of the message itself. It should also consider the demand of the task that the driver is likely to perform after receiving the message.

For example, the lane departure warning system alerts the driver when the vehicle crosses road lines. This warning can be presented by several means, such as sound, a voice message, or an icon displayed on the dashboard. These media have different associated demands. The sound is mainly auditory, perceptual and spatial, the voice message is auditory, perceptual and verbal and also has a cognitive part, whereas the icon is visual, perceptual and spatial. However the induced task remains the same and does not depend on the presentation mode. If the lane departure was unintentional, the driver will try to correct the trajectory immediately. This is a highly demanding task, with a demand matrix of the form:

$$\mathbf{D}_{visual} = \begin{pmatrix} H & H & VH \\ H & H & VH \end{pmatrix} \quad \mathbf{D}_{auditory} = \begin{pmatrix} M & H & L \\ M & H & L \end{pmatrix}$$

where  $L, M, H, VH$  are linguistic variables meaning respectively low, medium, high, and very high load. The human-vehicle interface manager must ensure that the driver is able to not only receive the information message, but also to perform the task induced by this message.

As seen in chapter 2, the presentation media reflects the importance of the information. The driver expects critical information to be presented in a more salient manner, such as, for example, a spatial auditory signal to present a warning. Note that critical information is, by nature, urgent. The mode of presentation depends also on the message itself, and its compatibility with certain driver interfaces. For example, direction information, produced by a navigation system, can be presented as a verbal message, an icon, or as text displayed on a screen. However, the GPS position of the vehicle can only be presented in a visual manner; a verbal interface would be awkward.

Moreover, as discussed earlier, the presentation mode must also take into account the rate at which the information will be accessed. On most HVI, the speedometer is positioned as near as possible from the road scene. Thus the time required to switch from the road to the speedometer, visual scanning and focus time included, is minimal. Information that needs to be accessed more rarely can be placed further from the road scene, or even require user action to be displayed.

This thesis aims to reduce driver information overload by filtering the messages issued by the system. According to the model, described in the previous section, overload occurs when the sum of the tasks demands exceeds the acceptable load in one or more resources of the mental model. Therefore, to prevent this from happening, the system must lighten the load on individual resources.

One must note that, in most cases, the overload does not occur for only one resource. If overloaded resources are part of the perception stage, selecting another interface will help; otherwise, one could choose to either interrupt other sources or not present the message.

In an attempt to reduce driver overload, the following strategy is proposed. A model is built to continually monitor the driver's current workload, as well as a list of the tasks they are performing and a list of the messages they are currently receiving.



To each message is associated a list of possible interfaces, ranked by order of preference. These interfaces are selected to satisfy the constraints presented in Section 4.1. For example, the preferred presentation mode for a navigation direction message might be speech and an icon displayed on the instrumentation panel. If this is not available, the next preferred mode might be speech with an icon displayed on the central console screen. This is illustrated in Figure 4.3.

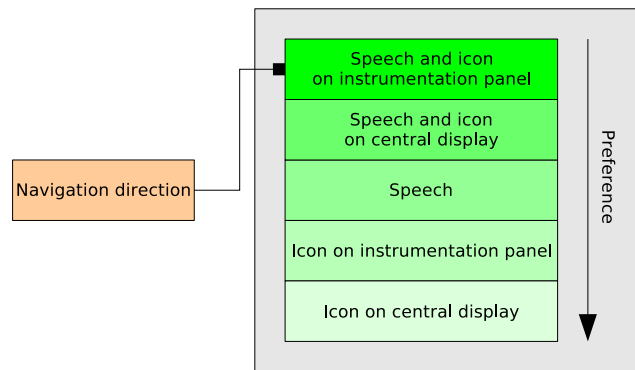


Figure 4.3: Example of preferred interfaces for navigation direction messages

When a new message is produced by an information source, the information management system computes its importance index and updates the ranked list of messages. Starting by the most important one, it estimates the demand of the task induced by the message, and checks if it is within the driver’s capabilities. It then evaluates the load created by the presentation of the message, using the first ‘preferred’ interface. If the load surpasses the driver’s capabilities, an alternate media is considered, until no more presentation mode is available. In this case the information message is not presented.

The procedure is outlined in Figure 4.4.

The system must also continuously recompute, in real-time, the FCMs that depict environment influence over tasks difficulty and information relevance. This strategy will be implemented and tested in a case study, in chapter 6.

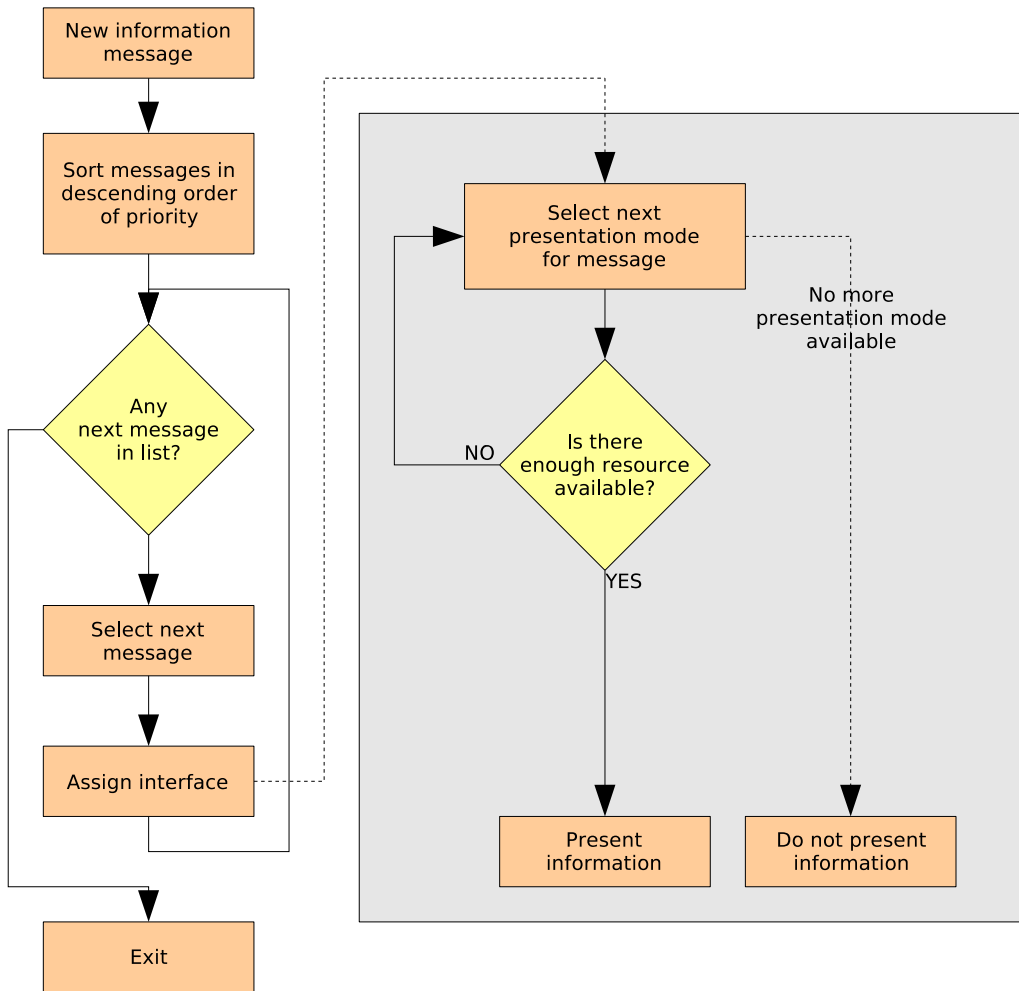


Figure 4.4: Flow chart for real-time message management

# Chapter 5

## Implementation

In the previous chapters, techniques for information management and prioritisation were developed. It is clear that a large quantity of data need to be processed, and it is important to ensure that these techniques can be implemented. In this chapter is described the development of the tools required to test the strategy outlined in chapter 4. These tools must allow us to acquire data in real-time and with accurate synchronisation of the streams; they must provide the capability of transporting and storing large quantities of data for further analysis; they must perform, in real-time, measurements and calculation on the data; and finally, they must enable us to simulate events generated by sensors and driver assistance systems.

### 5.1 Real time data capture

#### 5.1.1 Background

In order to validate the prioritisation and presentation model presented here, it is important to test it in the most realistic possible conditions. RoadSense, briefly mentioned in chapter 1, is a 36 months European project comprising several universities and car manufacturers. It aims at providing a platform for validating Advanced Driving Assistance Systems, and is composed of two main parts. Firstly a set of valid driver metrics is selected and validated by a group of Human Factors experts. Secondly, a system, called D-BITE, is designed to capture data and compute metrics in real time. Like the in-car

information system, the data are heterogeneous (video, sound, CAN, etc.) Because of the large amount of data to be captured, the system must be distributed.

As part of the project specification, it has been proposed to use the IEEE-1394 bus as the backbone of the D-BITE system. As seen in chapter 2, this bus has many big advantages, such as speed and support of both synchronous and asynchronous communications.

The work presented here was developed within the Roadsense project. D-BITE functionalities and structure [47] were specified by the universities participating in the project, namely Cranfield University, Université de Technologie de Compiègne, and Université Blaise Pascal. However, the group concentrated a Microsoft Windows implementation, whereas, for the sake the openness and evolutivity, the system described hereafter is based on the open-source operating system Linux.

Linux was created by Linus Torvalds at the University of Helsinki, in Finland [48]. It was released under a particular license to enable anybody to modify, distribute and sell any part of it. Countless volunteers are participating to the development of its kernel and other applications, fixing and adding features. This makes Linux extremely stable and adaptable to any requirements. For example, it is possible to add support for specific functions, such as advanced IEEE-1394 communication, by modifying parts of the operating system, called modules. This process of kernel recompilation is lengthy and quite complex, but was used several times in this project, to benefit, for example, from the Asynchronous Resources Manager (ARM) functions, as explained in Section 5.1.4.

### **5.1.2 Time synchronisation**

The human factors studies require a time resolution of 1 millisecond. If this might be achievable on a single computer, it becomes more complex when a greater number of computers are involved. Indeed, the clocks of the computers must be set at the exact same time. This is difficult to operationalise: numerous delays can occur, such as, for example, during the transmission of the command over the network linking the computers.

Moreover, different computers usually have slightly different clock speeds. In order to decide on a synchronisation strategy, an experiment was designed and conducted to verify if clocks from different computers were incrementing synchronously. Two computers running Linux are linked via an IEEE-1394 network. The IEEE-1394 bus specifications require each node of the network to store a common time value [26]. This field is 32 bits long, and has a resolution of 40 nanoseconds, which is far beyond the requirements. However, its maximum value is 128 seconds.

A 3-steps manoeuvre is repeated:

1. Computer1 records the system time  $t_{s1}$ , in microseconds, and the IEEE-1394 time  $t_{i1}$
2. These two values are sent in asynchronous mode to computer 2;
3. Computer2 computes the transmission time from comparing the received IEEE-1394 time with the current value  $t_{i2}$ , and calculates the time difference between the two computers  $t_{s2} - t_{s1} + t_{i2} - t_{i1}$ .

Repeated numerous times, this experiment shows that the smallest time shift is  $30\mu\text{s}$  every 1s, which means that for at the end of a 3 hours experiment, the clocks will have shifted of 324 milliseconds. This is not acceptable.

It is proposed to record the IEEE-1394 field to synchronise the nodes times [49], along the system time value, at fixed frequency, in a *synchronisation file*. The data are timestamped with the system time value; when the experiment is over, the files coming from different computers are gathered and the time-stamps are corrected according to the comparison of the synchronisation files.

### 5.1.3 Video capture

The Linux IEEE-1394 sub-system is an open-source implementation [50]. As the development takes place on versions 2.4.x of the Linux kernel, two C libraries are used: *libraw1394*, for raw IEEE-1394 communications, and *libdc1394*, to control video cameras compliant with the Instrumentation and Industrial Digital Cameras (IIDC) standard.

This program takes input from a video stream using a camera linked to the recording computer by the IEEE-1394 bus, and produces a sequence of compressed Joint Picture Experts Group (JPEG) images. This allows the efficient storage of many hours of video, and permits, when necessary, the analysis of single images.

The program algorithm is structured in three stages, as shown in Figure 5.1. Firstly the resources (network, memory, and variables) are initialised; secondly, every time a new pixel array is received, the data are compressed and a new file is written.

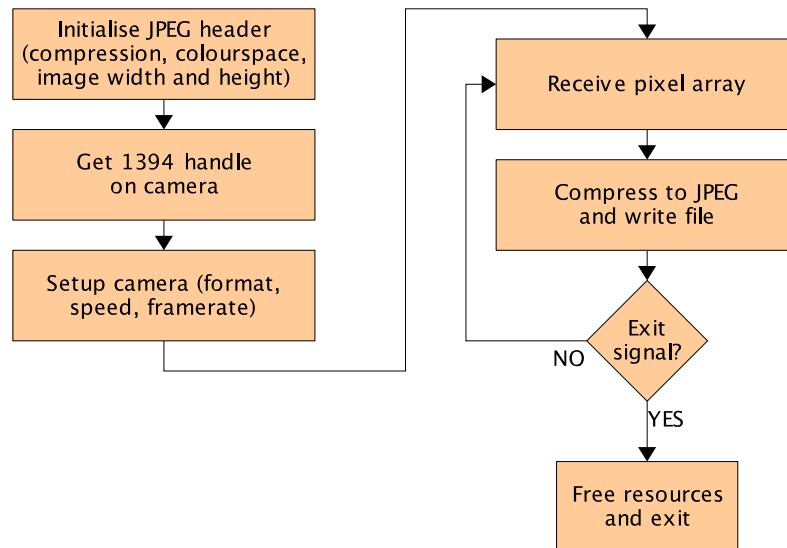


Figure 5.1: JPEG image capture algorithm

#### 5.1.4 Data transfer

Since data are captured on different computers, it is necessary to transfer the resulting files to a single computer, where they can be synchronised and recorded to a removable media, such as a Digital Versatile Disk (DVD). The existing IEEE-1394 network can also be used for this purpose, as it allows asynchronous transfers. Moreover it spares the installation of a secondary data network. The detailed description of the program might not be essential to the reader, and will not be given here. However, the data transfer

mechanism is an important element of the data acquisition, as it enables the collection of data captured on different computers, or nodes, of the network. The data transfer program makes use of the ARM libraries, that enable the sending and receiving of packets through the IEEE-1394, in asynchronous mode; it is not possible to use the isochronous mode, because it does not guarantee that every packet is successfully delivered.

ARM functions enable the developer to reserve a range of memory addresses on the IEEE-1394 board. This range is actually mapped to a memory zone of the computer itself. When data are sent to this zone, a so-called ‘callback’ function is triggered, and the data can be processed. For the data transfer implementation, the address `FFFF F000 3000h` is used, with a buffer size of 2 kilobytes, which is the maximum size of asynchronous packets at 400 Mbit/s.

The program is designed to receive data from several nodes simultaneously. It must therefore be implemented as a multithreaded application. Threads are sequences of instructions that can be executed simultaneously within the same program. In the present case, each thread of execution is associated to one single data transfer. This way, a delayed packet from one node does not affect all currently active data transfers from other nodes. Figure 5.2 illustrates the benefits of multi-threads over single thread data transfer. In both cases, three transfers are made; each block represents a data unit transferred. During the first transfer, the delivery of the fifth block is delayed. While the two other transfers are delayed with a single threaded program, they can be performed successfully in the case of a multi-threaded program, and the overall transfer time is improved.

After initialisation, the program creates a folder for each node on the network, where the received files will be stored. Then, every time a packet is received, the callback function checks if the transaction number associated with the packet is known. If it is, the data are passed to another thread to be written. Otherwise, a new thread is created to process the data. The program main algorithm is described in Figure 5.3.

In Appendix B the C header file of the main program can be found. Data structure, packet format and functions are described in more details.

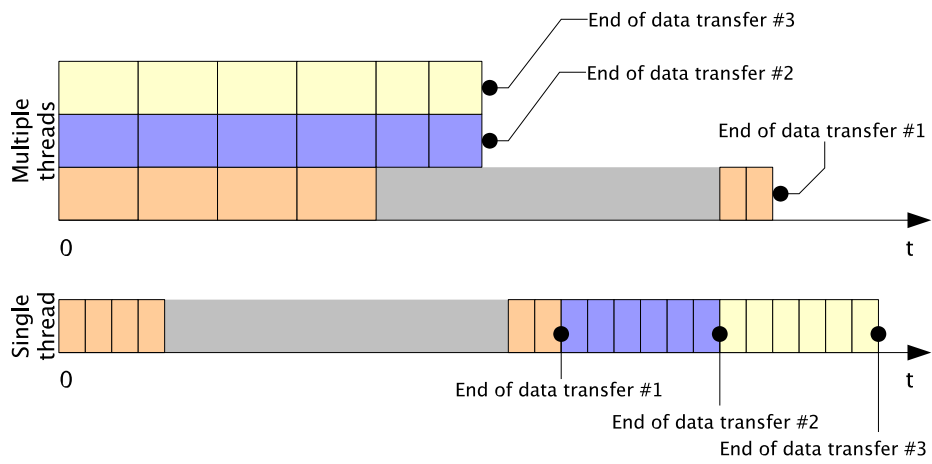


Figure 5.2: Single versus multiple threads data transfer.



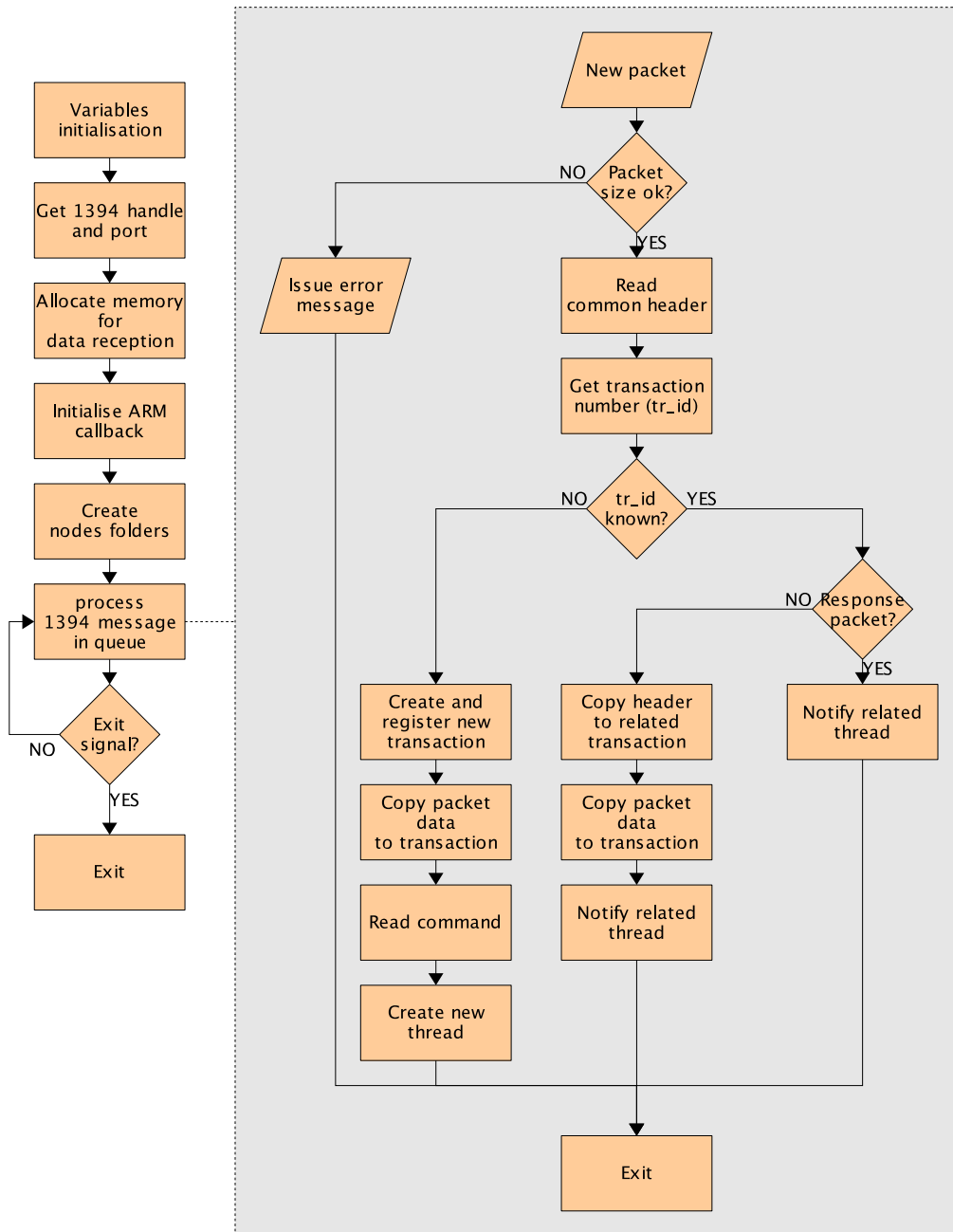


Figure 5.3: Simplified flowchart for IEEE-1394 file transfer.

## 5.2 Simulation and replay

The purpose of the simulation program is to test the prioritisation strategy on part-real situations. The data must be replayed in a manner that takes into account the moment of their recording and the delays between events.

This program was developed in the context of the ROADSENSE project, and uses the data format defined by the consortium partners. A Java implementation was chosen.

Java is a programming language developed by Sun, first born as ‘Oak’, in 1991. Although it is usually thought to be slow, recent performance benchmarks show that in some areas its performance is comparable to C or C++ [51, 52, 53]. It is widely used in application server environments, embedded systems and is also used for development of real-time applications [54]. For applications requiring a graphical interface, alternatives to the default libraries have appeared, such as the Standard Widget Toolkit (SWT) library [55]. It uses native components of the operating system display, and allows a Java application to feel and look like a standard application. Moreover, Java robustness and strong language structure reduce the length of the traditional design-coding-debug loop. Java is therefore adapted to implementations with strong time constraints, as is the case in this project.

The first part of the application enables replay of the files acquired with the D-BITE system. It is composed of an interface that provides common replay functionalities, such as play, pause, fast forward, or step-by-step; a set of tools to add and remove files; a ‘synchroniser’, that organises the timely replay of events; and players for every data type, associated to a specific graphic representation. The D-BITE player can replay CAN data, sequences of bitmap or JPEG images, stereovision and gaze tracker data.

The design of the program makes extensive use of Java oriented-object features, such as inheritance and interfaces. Two fundamental classes, `DBTFile` and `DBTPlayer`, implement the D-BITE file structure and replay mechanism. These classes are extended for each data type, as shown in the UML class diagram in Appendix C.

The application is designed as multithreaded. The displays share one common thread. One thread is dedicated to the synchronisation of events;

this enables the synchronisation to be more accurate. Furthermore, one thread is created for each data type replayed. The synchronisation mechanism is illustrated in Figure 5.4.

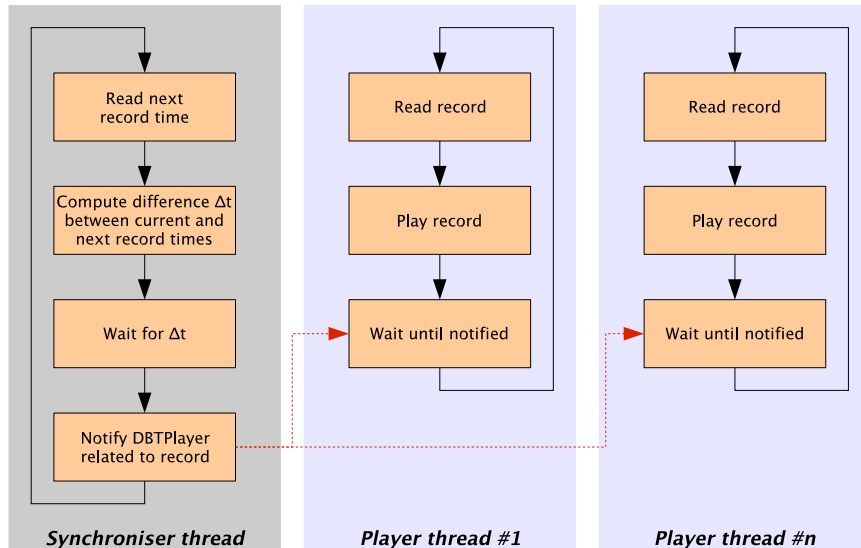


Figure 5.4: Events synchronisation

The second part of the program permits simulation of events. For example, users might want to test the behaviour of the prioritisation system when the phone is ringing, and during an overtaking manoeuvre. It is obviously not feasible to create the real conditions, and difficult to implement on a driving simulator platform. Indeed this would allow the testing of the reaction of the driver to the system, but not the strategy itself.

The simulation program creates files during the replay of an experiment. A control panel allows the user to record events, such as, for example, indicators, phone or navigation system status. These events are timestamped using the experiment time, provided by the `getTime()` function of the synchroniser. This function returns the sum of the time elapsed since the last replayed event and its timestamp.

## 5.3 Prioritisation

The implementation of the prioritisation strategy has been designed to be integrated into the D-BITE player, but also, with minor modifications, into an experimental vehicle. It has been developed in Java, and is composed of five main packages, FCM, fuzzy logic, information system model, control and visualisation.

The FCM package exploits the object structure of the language. Each concept is represented by a class. Computation of the FCM is recursive — the computation of each element triggers the computation of its inputs. A concept can be linked to a sensor via the `DataProvider` interface. In this case, requesting the value of the concepts triggers the evaluation of the sensor. The visualisation package provides a graphical user interface for the FCM, as shown in Figure C.4.

In Appendix C, Figure C.5 and C.5 presents the data replay and analysis tool.

# Chapter 6

## Case study

### 6.1 Experiment design

This study has the purpose of evaluating the prioritisation strategy discussed in the previous chapters.

In order to realistically simulate the in-car information system, the following sensors are selected: lane position, steering wheel position, speed, light sensor, and gaze direction. Most of these sensors are commonly found on current vehicles, apart from the lane position and the gaze direction. They are commercially available, but this study will use Cranfield University implementations [56], for three main reasons: they are freely available, they can be directly integrated into the main application, and they offer good performance.

Moreover, the simulation must also include driving assistance systems that are currently available or will be in a near future. The following systems are selected:

- instrumentation;
- traffic alerts;
- navigation;
- radio;
- lane warning;
- phone.

The data is acquired using the available ROADSENSE vehicle, a Jaguar X-Type, equipped with a computer running D-BITE and recording, at the time of the experiment, CAN data (speed, steering wheel angle, throttle

pedal, brake), forward looking video and driver video. Figure 6.1 shows pictures of the experimental vehicle.



(a) Forward-looking and driver video cameras. The cameras are highlighted.



(b) Steering wheel and instrumentation.



(c) D-BITE hardware setup.



(d) Experimental vehicle.

Figure 6.1: Experimental vehicle views

The driver video is post-processed to compute the gaze directions, calibrated for six gaze areas: left mirror, rear mirror, right mirror, road, and central console. Because the CAN data does not include the indicators status, it must also be generated in post-process.

Moreover, the radio, traffic alerts, lane warning, navigation direction and phone are simulated.

To validate the prioritisation strategy, it is required to simulate situations where the driver is overloaded with information. Note that, at this stage,

the purpose is not to make a human factors study. Indeed, the prioritisation strategy is designed to embed expert knowledge, in FCMs, risks, tasks, and interfaces definition; however in this study the aim is to validate its structure, and prove that it has the potential of improving in-vehicle information management. The experiment will therefore be conducted in four different stages:

1. Acquire data from real driving situation;
2. Simulate, in post-process, environmental conditions;
3. Validate the resulting ranking order of information messages;
4. Validate the proposed presentation modes.

Two data streams from the vehicle CAN bus are extracted: the speed, and the steering wheel angle. Video images from the road and the driver also also recorded.

To simulate information, the software package presented in chapter 5 is used. During the replay of the data, it allows us to reproduce the main events generated by phone, navigation system, traffic alerts systems, instrumentation and lane departure warning. This program permits the modification of the truth value of concepts that are not computed in real-time, such as, for example, night and driving time.

The experiment is conducted on local roads, in normal traffic conditions, around Cranfield University. This route is roughly 20 miles long and is composed of 30% of motorway, 15% of A-road, 50% of B-road and 5% of ‘urban’ road, as shown in Figure 6.2. Figure 6.3 details the the road types and route events.

The B-road section is quite curvy and narrow; it provides a good example of demanding driving task, where the driver must concentrate on the handling of the vehicle. On the motorway section, four lane changes are performed, with and without preparation.

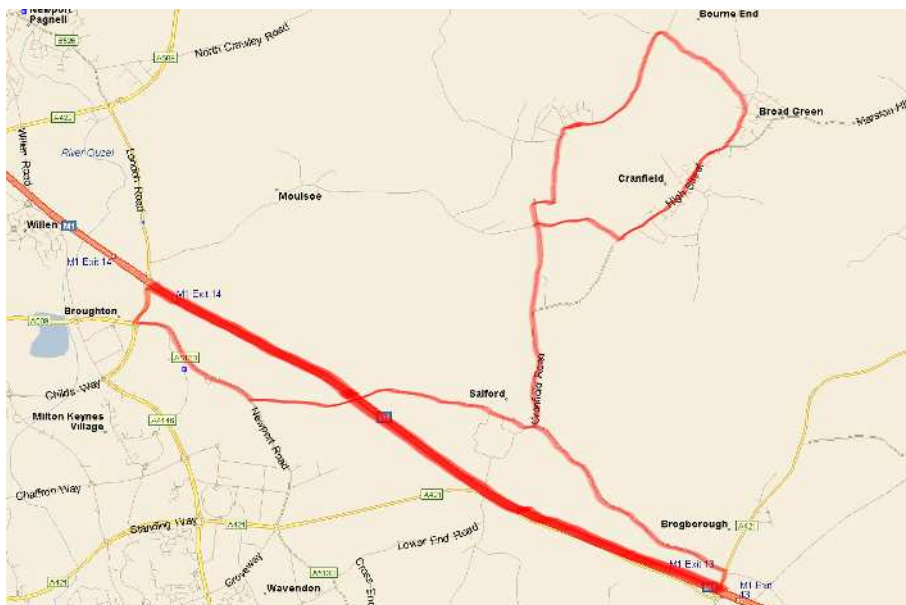


Figure 6.2: Experiment route



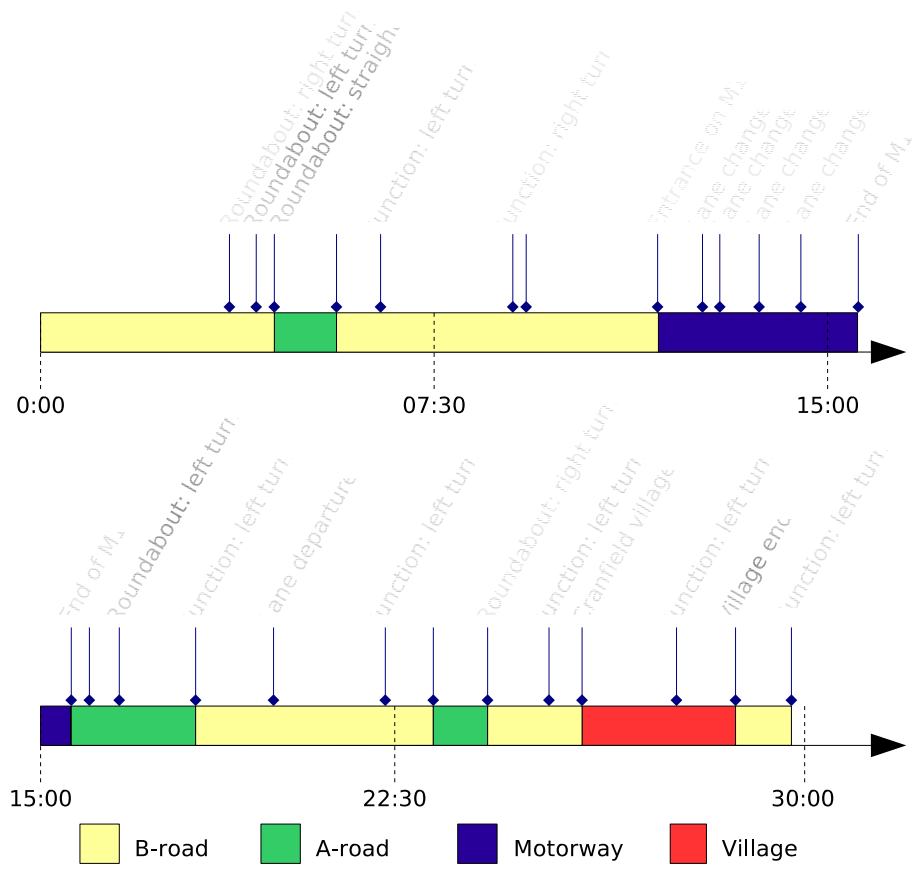


Figure 6.3: Experiment route events

## 6.2 Strategy implementation

Part of the implementation of the prioritisation and presentation strategy is to characterise the information system. In this section, each information message is associated to a risk group, a risk membership and a an induced task. For this study, the demand matrices of the tasks is determined empirically; for a real-time implementation it would be necessary to base these values on measurements and expert knowledge.

### 6.2.1 Definitions

As it was proposed in chapter 3, the information sources are sorted in four risk groups: safety, driving, communication and entertainment. Each information message is given a risk and a membership value, as explained in Section 3.1. Table 6.1 shows the values used for this experiment.

System	Message	Group	Membership
Navigation	Map with position	Driving	0.2
	Direction	Driving	0.2
Phone	Incoming call signal	Communication	0.5
	call data	Communication	0.9
Instrumentation	Driving instruments	Driving	0.8
	Alert	Driving	0.9
Radio	News	Entertainment	0.6
	Music	Entertainment	0.2
Traffic alerts	alert	Driving	0.7
Lane warning	Crossing line	Safety	0.6

Table 6.1: Risks groups and values

Moreover, six linguistic variables are  $Z, VL, L, M, H, VH$  defined, representing respectively null, very low, low, medium, high, and very high values.

## 6.2.2 Lane warning

The lane warning systems detects when the vehicle crosses a line and alerts the driver with an aural, visual or haptic signal. It prevents the vehicle from going off its lane and potentially cause an accident. It is relevant when the driver crosses the lines unintentionally. This can happen for various reasons: he might have fallen asleep, or become distracted.

### Relevance

In this case study, it is assumed that a gaze tracker is available. This apparatus gives indication of where the driver is looking, and will help detect when the driver is not focusing their attention on the road. The drowsiness of the driver, however, is difficult to detect without the use of intrusive sensors. But what can be measured is the driver activity (pedals, gear, steering wheel, etc.). Moreover, external conditions such as day or night, time of the day, and time of driving can make drowsiness more likely.

However, if the driver intends to change lane, the relevance of the lane warning system must be less. The driver manifests this intention by turning the indicator but also by looking in the mirrors.

Figure 6.4 illustrates the fuzzy cognitive map representing the relations between the environment and the relevance of the lane warning system.

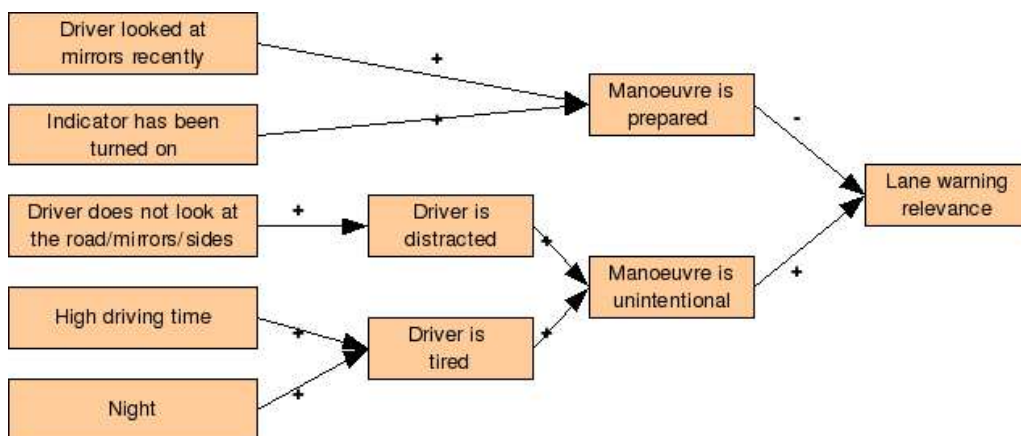


Figure 6.4: FCM for lane warning system relevance

Four elementary piece of information must therefore be computed in real-time and given as input to the FCM.

### Induced tasks

When the system issues a message, it means that the vehicle is about to cross a line. If the driver did not intend this manoeuvre, he must very quickly check his environment and take corrective action. The demand matrices can be therefore expressed as:

$$\mathbf{D}_{visual} = \begin{pmatrix} Z & Z & Z \\ VH & M & VH \end{pmatrix} \quad \mathbf{D}_{auditory} = \begin{pmatrix} Z & Z & Z \\ Z & M & Z \end{pmatrix} .$$

### 6.2.3 Navigation

Generally, navigation systems have two main functions. Firstly they give the position of the vehicle on a map, displayed on a screen integrated into the dashboard. Secondly they give directions to a pre-programmed destination.

#### Relevance

The relevance of a navigation system therefore depends on the interest the driver has in it. If they programmed a route and activated the system, they is certainly interested in the information. On the other hand, they could very well decide not the follow the directions given by the system. In this case its relevance should be decreased.

The related FCM can be build, as shown below (figure 6.5).

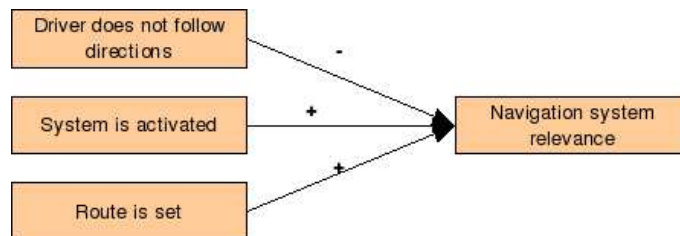


Figure 6.5: FCM for navigation system

## Induced tasks

The navigation system aims at making the driving task easier, and is therefore a member of the the ‘driving’ group. It issues two major types of information: the position of the vehicle on a map and directions. The former does not actually induce any direct task, however the latter make the driver relate the information to their environment (signs, junctions, cars) and possibly prepare a manoeuvre. This translates into the following demand matrices:

$$Thei\mathbf{D}_{visual} = \begin{pmatrix} L & L & Z \\ VL & M & Z \end{pmatrix} \quad \mathbf{D}_{auditory} = \begin{pmatrix} Z & Z & Z \\ Z & M & Z \end{pmatrix}.$$

### 6.2.4 Phone

In-vehicle communication systems have been the focus of many recent studies. Although it is now forbidden in several European countries to hold a phone in one’s hand whilst driving, people are free to use integrated systems. These are becoming more and more popular, as the daily time spent on commuting increases [57].

#### Relevance

The relevance of the phone in the vehicle firstly depends, as for the navigation system, on its activation. Different drivers will attach different importance to the phone. A learning system may measure that by looking at the reactivity of the driver: does he usually pick up quickly? Does he interrupt other tasks to answer?

Given this, a simple FCM can be drawn (figure 6.6).

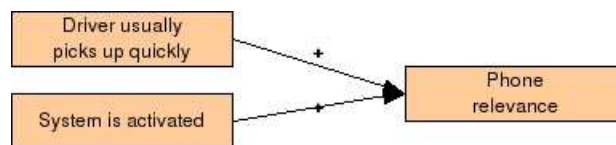


Figure 6.6: FCM for phone

## Induced tasks

When a incoming call is signalled, the driver is likely to pick up and start a conversation. On recent cars, equipped with integrated communication systems, ‘picking up’ is not a demanding task; however, the conversation itself demands perceptual and cognitive resources, and can be described by:

$$\mathbf{D}_{visual} = \begin{pmatrix} Z & H & Z \\ Z & Z & Z \end{pmatrix} \quad \mathbf{D}_{auditory} = \begin{pmatrix} H & H & L \\ Z & Z & Z \end{pmatrix}.$$

### 6.2.5 Instrumentation

The instrumentation panel displays every information related to status of the car, such as speed, engine RPMs, indicators, lights, petrol, oil temperature, etc.

#### Relevance

This information is related to the driving task, and therefore might not be desired when the vehicle is not actually serving its main purpose: everybody can remember waiting for some time in their vehicle, listening to the radio. As it is generally positioned where it is best viewed, the instrumentation panel would then be the best location to display channels information, showing a movie or any other kind of entertainment. On most cars, this can be detected by the position of the key: the second position powers only the radio, whereas the third one starts the instrumentation. Otherwise, but less accurately, the status of the engine (on or off) can be a hint that the driver only wishes to power the entertainment system. This is represented in figure 6.7.

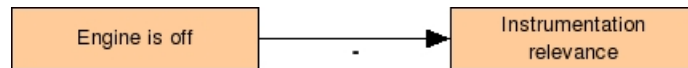


Figure 6.7: FCM for instrumentation system

## Induced tasks

The instrumentation itself does not induce any task; however, if a warning is issued, the driver must diagnose the problem and consider stopping the

vehicle. This can be translated into the following demand matrix:

$$\mathbf{D}_{visual} = \begin{pmatrix} Z & M & L \\ Z & Z & Z \end{pmatrix} \quad \mathbf{D}_{auditory} = \begin{pmatrix} Z & M & L \\ Z & Z & Z \end{pmatrix}.$$

## 6.2.6 Traffic Alerts

### Relevance

The traffic alerts are relevant only if they are related to the driver route, or roads nearby. This information can be computed from the navigation system, if the driver programmed their route earlier. Alternatively, the route can be estimated from the position and the direction. Moreover, if the traffic-flow slows down, the driver might be interested in traffic alerts. The following FCM (figure 6.8) illustrate these relations.

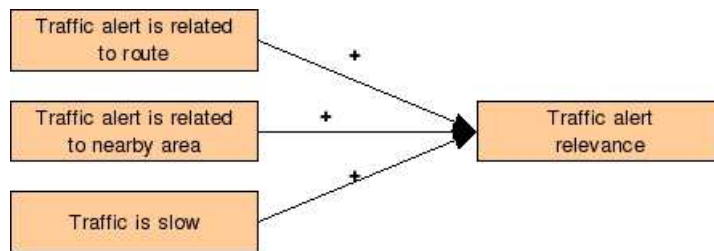


Figure 6.8: FCM for traffic alerts

### Induced tasks

Receiving a relevant traffic alert makes the driver think of his route and possible alternatives. This is a purely cognitive, low-demand task:

$$\mathbf{D}_{visual} = \begin{pmatrix} Z & L & Z \\ Z & L & Z \end{pmatrix} \quad \mathbf{D}_{auditory} = \begin{pmatrix} Z & L & Z \\ Z & L & Z \end{pmatrix}.$$

## 6.2.7 Driving task

The information system must also take into account the tasks the driver performs. Every time the driver interacts with the vehicle, it creates a demand that must be taken account into the total workload calculation. In this experiment this will be limited to an estimation of the driving task demand.

The driving task demand depends on environmental factors. On small country roads, driving requires more effort than on long, straight roads. It is proposed to use the variations of steering wheel angle as an indicator of the driving task demand. ‘Big’ variations of the angle around the 0 position are recorded. This enables to filter out the small corrections of trajectories that are part of normal driving. Furthermore, changes of signs of the angle value denotes more driver’s activity.

In this implementation, frequency of events is measured where :

$$\begin{cases} \alpha_{i+1} - \alpha_i > 40 \\ \alpha_{i+1} \times \alpha_i < 0 \end{cases}$$

with  $\alpha_i$  the steering wheel angle at record  $i$ . The frequency is computed in real time on the last 30 seconds which represents, in average and for this experiment, 750 records.

### 6.2.8 Normalisation

The sensors output must be converted into a *truth* value. This value must be 1 if the sensor output is totally true,  $-1$  if its contrary is true. Simple thresholding functions can be used, but in this study a parameterised sigmoid function will be mostly considered, as shown below.

$$T(t) = \frac{a}{1 + e^{-\alpha t}} + b \quad (6.1)$$

For example, the parameters of the truth function for the proposition ‘driver usually picks phone up quickly’ can be calculated as follows. This proposition is fully true for  $t = 0$ , but starts to be untrue when  $t > 5$ ,  $t$  being the average ‘picking up’ time. The parameters must therefore satisfy three conditions:

$$\begin{cases} T(t) = 1 & \text{if } t = 0 \\ T(t) = 0 & \text{if } t = 5 \\ \lim_{t \rightarrow \infty} T(t) = -1 \end{cases}$$

From the first and third condition the values of  $a$  and  $b$  are deduced:

$$\begin{cases} b = 1 - \frac{a}{2} \\ a + b = -1 \end{cases} \text{ then } \begin{cases} a = -4 \\ b = 3 \end{cases}$$



The second condition gives the value of  $\alpha$ :

$$\begin{aligned} 0 &= \frac{-4}{1 + e^{-5\alpha}} + 3 \\ e^{-5\alpha} &= \frac{1}{3} \\ \alpha &= \frac{1}{5} \ln 3 \end{aligned}$$

The truth function for the proposition ‘Driver usually picks up quickly’ is therefore:

$$T(t) = \frac{-4}{1 + e^{\frac{1}{5} \ln 3 t}} + 3.$$

This function is represented in Picture 6.9.

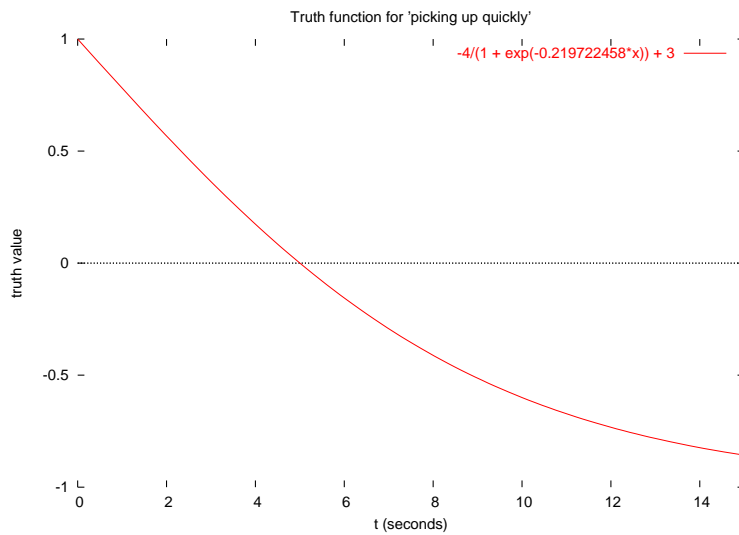


Figure 6.9: Sigmoid function for phone

## 6.2.9 Interfaces

The case study includes the simulation of the human-vehicle interface. It is chosen to simulate only the aural and visual interfaces, the haptic interface requiring specific hardware.

The visual interface is composed of two displays: the instrumentation panel  $P_i$ , in front of the driver, and the ‘navigation’ panel  $P_c$ , situated on the

central console. As seen previously, the importance of the information will be represented by its location, its size, colour and level of abstraction (text, icon).

In this case study, the aural interface will be a standard sound system. Information importance will be represented by the sound volume and its modality.

In Table 6.2 a list of interface is associated to each information message, sorted by order of preference.

<b>System</b>	<b>Message</b>	<b>Preferred interface</b>
Navigation	Map with position Direction	Image on $P_i$ , image on $P_c$ Icon on $P_i$ + speech, icon on $P_c$ + speech, speech, icon on $P_i$ , icon on $P_c$
Phone	Incoming call In-call	Ring tone, icon on $P_c$ Audio stream
Instrumentation	Driving instruments Alert	Image on $P_i$ Speech + icon on $P_i$ , Sound + icon on $P_i$ , icon on $P_i$
Radio	News Music	Audio stream Audio stream
Traffic alerts	alert	Speech, text on $P_i$ , text on $P_c$
Lane warning	warning	Sound + icon on $P_i$ , sound, icon on $P_i$

Table 6.2: Systems and interfaces for simulation

## 6.3 Results

As explained in the introduction of this section, this experiment aims to validate the strategy described in the chapters 3 and 4. In the first part, the

validity of the prioritisation strategy is studied. The second part shows some results of interface allocation.

Figure 6.10 shows the calculation of frequency of the major changes of the steering wheel angle . The upper part of the graph represents the steering wheel angle variations; the major angle changes are highlighted. The lower part show the frequency of changes, computed on the last 30 seconds. Note that the calculation for the first 5,000 records of the experiment (Cranfield University and car park) are not represented.

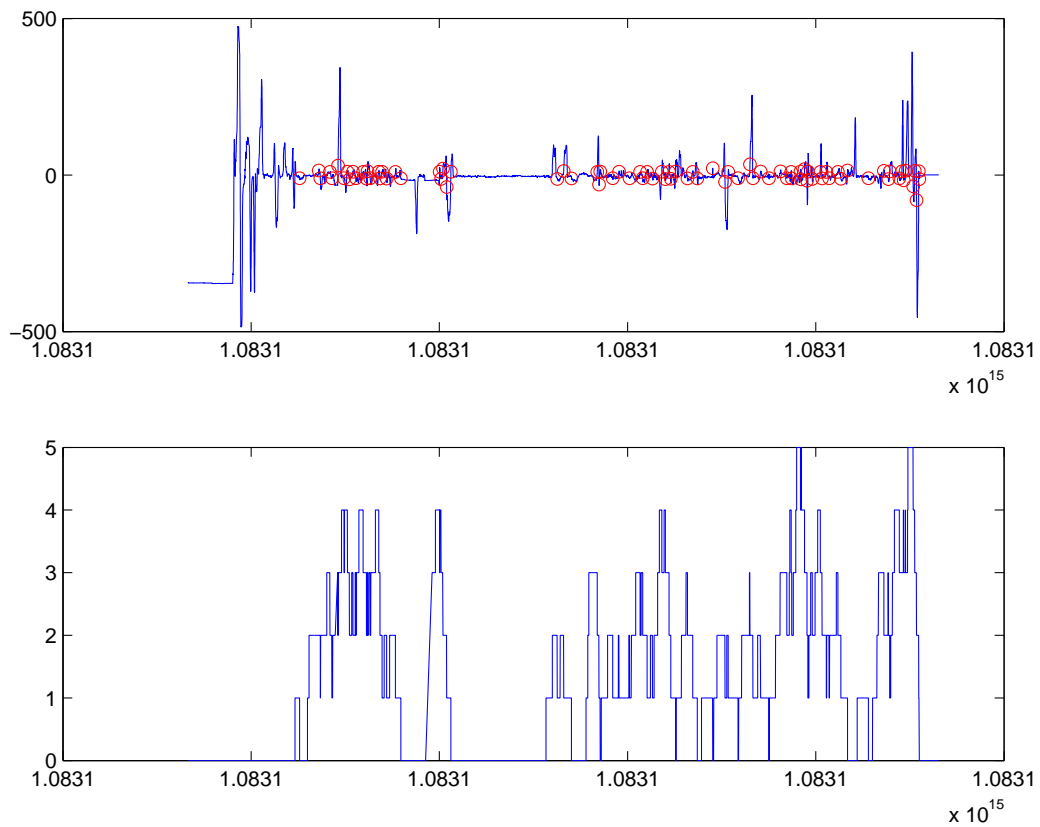


Figure 6.10: Driver steering workload measurement.

This figure shows that frequency of steering deviations is higher on the B-road sections, and in the village. This is consistent with an intuitive estimation of the driver ability to performs other tasks.

Figure 6.11 and 6.12 present two result of prioritisation. The load and demand matrices are shown respectively in the upper left and upper right

corner. The load is colour-coded, from blue, for a free resource, to red, for an overloaded resource. The lower half of the interface shows the messages currently presented and their allocated interfaces, as well as the tasks performed.

In these two situations, the prioritisation mechanism classifies information correctly. In Figure 6.11, the lane warning message is presented through a spatial auditory interface, and is given the highest priority index.

In Figure 6.12, instrumentation and navigation are the most important information. However as they both use visual resources, the driver is still able to process auditory messages. A relevant traffic alert can therefore be presented through the speech interface.

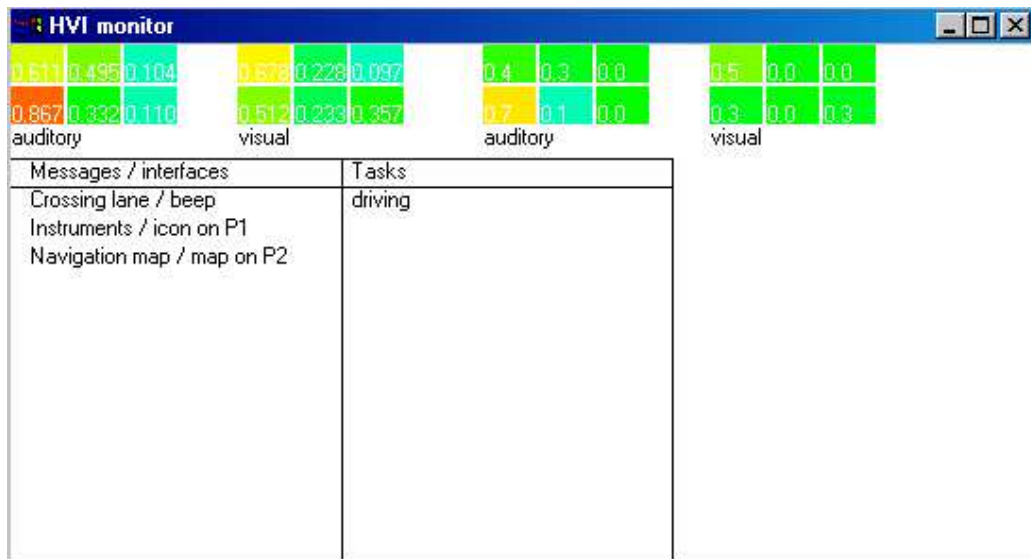


Figure 6.11: Prioritisation result

The prioritisation and presentation strategy also filters out successfully irrelevant information. Figure 6.13 illustrates this. The vehicle is about to enter the motorway, and therefore change lane. In normal conditions, the lane warning should alert the driver, which, in this situation, would be very disturbing. In the present case, the prioritisation strategy takes into account the relevance of the alert: as the indicator is on, and the driver looked at the mirrors recently, the relevance of the lane departure is low. The system therefore automatically dismisses the warning.

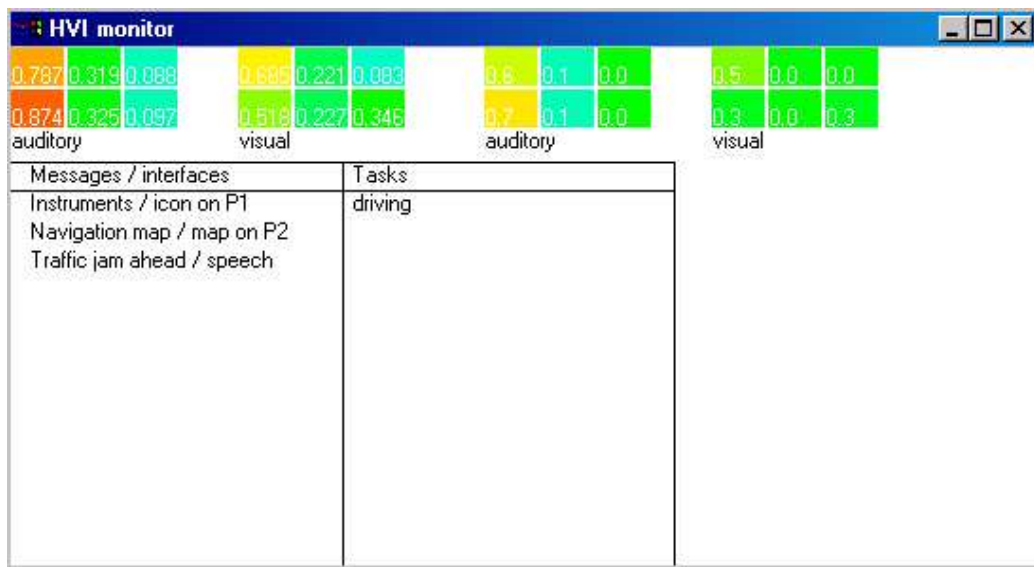


Figure 6.12: Prioritisation result

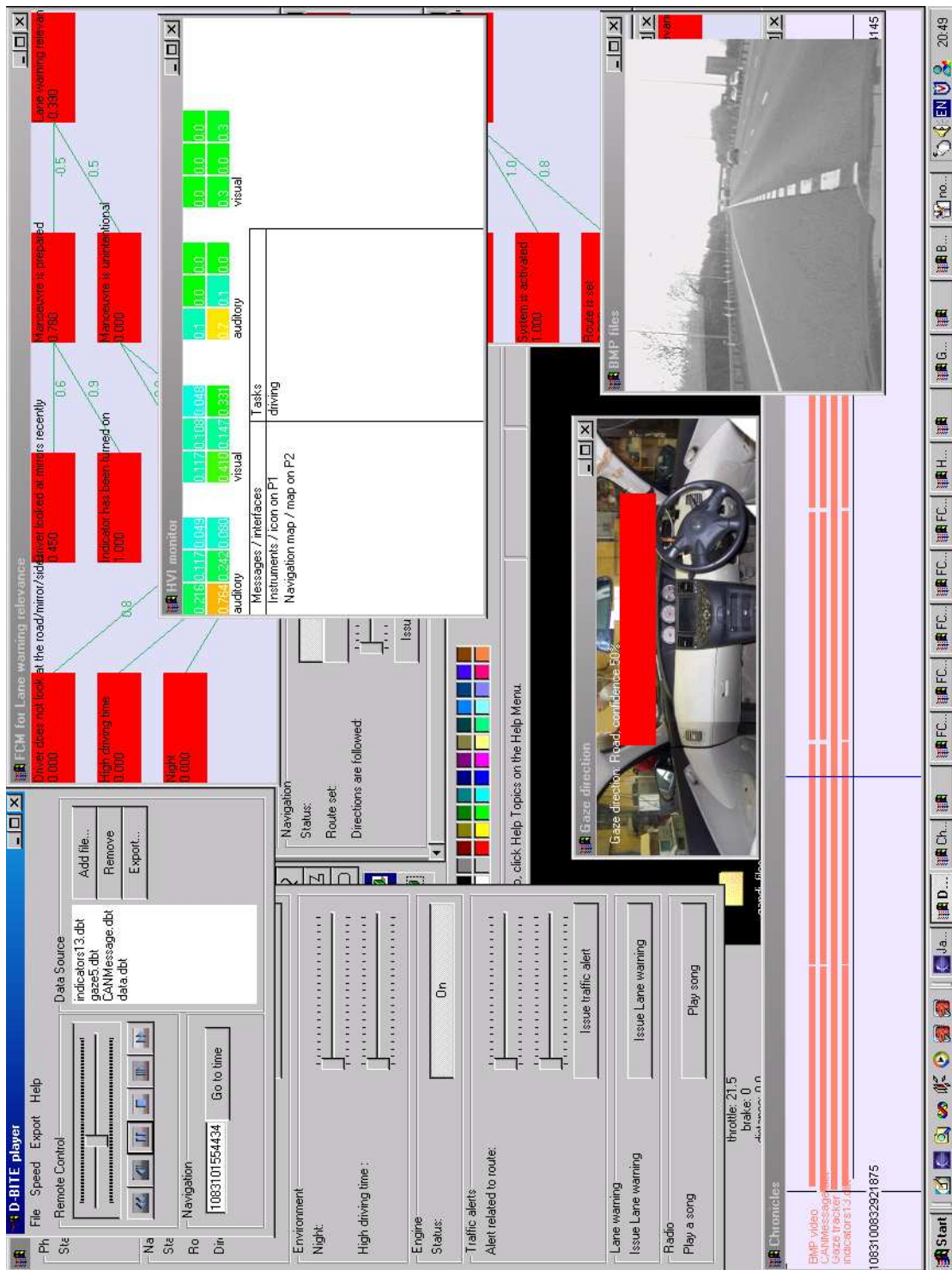


Figure 6.13: Irrelevant lane warning message

# Chapter 7

## Conclusion and further works

In this work was considered the problem of driver information overload. A novel strategy was discussed to prioritise and present ADAS information. Most of the currently implemented information management system associate information and interfaces statically; hard-coded rules attempt to prevent conflicts. For safety-related information only, it was also proposed to use prioritisation criteria based on dynamic characteristics, such as speed, location and acceleration of the hazards.

In this study a holistic approach to information prioritisation and presentation is proposed. The whole in-car information system was considered, from entertainment devices to safety-related systems.

It is proposed, in a first part, to classify information according to two main criteria: risk and relevance. The risk is related to the consequences of not presenting a particular piece of information. Information sources can be organised into risk groups; further hierarchisation is achieved by attributing membership values to information messages. Furthermore, relevance of an information message to the environment is a novel concept in this domain. It is proposed to model it using fuzzy cognitive maps: they are composed of causal relationships between concepts, and therefore allow to incorporate expert knowledge into the model.

To present information in an optimal way, it was first required to identify the factors that make an interface better than another. They are the nature of the information, its time constraints, and the frequency at which it must

be accessed. It was also proposed to consider the demand created by the interface, and by the task induced by the message. A model of the workload was built, based on Wickens' multiple theory, taking into account the interferences between resources. Finally, an overall information prioritisation and presentation strategy was presented.

The second part of this work is dedicated to the implementation and validation of this strategy. A set of novel tools was designed to acquire data and transfer data, in real-time, from an heterogeneous in-vehicle information system. These applications are based on the IEEE-1394 bus, and allow fast and reliable transfer of big amounts of data. An analysis and simulation application was also developed, as well as a concrete implementation of the prioritisation and presentation strategy. Based on one set of real data, this enables the simulation of a wider range of situations, and therefore better validates the system.

Indeed, it was possible to simulate different situations where the driver should have been overloaded with information. The data were acquired using a Jaguar X-Type as experimental vehicle, following a short route around Cranfield University. Using the simulation application, events were generated, representing the different information sources, such as, for example, the lane warning system or the navigation system. Several aspects of the proposed prioritisation and presentation strategy were demonstrated and validated: firstly it enables to model easily the influence of the environment on information relevance; secondly it outputs a list of messages ranked by importance; and thirdly it proposes interfaces adapted to the message, taking into account the driver limitations.

There are, of course, several improvements and additions that could be made to this work. As it was explained in the last chapter, the aim was not to develop a complete in-car information management system, but only set the foundations for it. The case study demonstrated the usability of the method. Validating an implementation of it would require Human Factors experts input on different aspects: the definition of the parameters influencing a message relevance, and the strength of these relationships; the driver's workload measurement method; or the definition of the demand matrices.



Moreover, it would be necessary to extend Wickens' model to include the haptic modality. The maximum workload, i.e. the driver's capabilities, is time-dependent, and driver-dependent. This is an important aspect that should be part of further developments of the model.

This work may have various applications. Its main objective is to propose a strategy to design a human-vehicle interface that is aware of the driver current needs and capabilities. Applied to the design of in-car information system, this could lead to the development of safer vehicles. Avoiding dangerous situations, where the driver is overloaded with information, would not be the only benefit. It would also help to reduce stress, making the journey more comfortable, and help building the trust relationship between driver and vehicle.

Information relevance is a concept that can be applied to other areas, such as personal information systems. Nowadays there is so much information available that people have to spent time choosing, skipping and skimming. Systems could be developed to filter out some, taking into account the environment, the context, and the driver capabilities.

# Bibliography

- [1] De Waard D. *The measurement of drivers' mental workload*. PhD thesis, University of Gronigen, Haren, The Netherlands, 1996.
- [2] Stutts J.C., Reinfurt D.W., and Staplin L. The role of driver distraction in traffic crashes. Technical report, AAA foundation for traffic safety, 1440 New York Avenue, N.W. Suite 201, Washington DC 20005, USA, 2001.
- [3] Directorate-General Research European Commission. Surface transport under the growth programme (1998-2002). Rue de la Loi/Wetstraat 200, B-1049 Brussels, 2002.
- [4] Papí J., Halleman B., and Figaredo A. *European Road Statistics 2002*. European Union Road Federation, Brussels, Belgium, 2002.
- [5] EUROPEAN COMMISSION DIRECTORATE GENERAL FOR ENERGY and TRANSPORT. Care - community road accident database. <http://europa.eu.int/comm/transport/care>, 2004.
- [6] EUROPEAN COMMISSION DIRECTORATE GENERAL FOR ENERGY and TRANSPORT. Road safety quick indicator 2003. [http://europa.eu.int/comm/transport/road/figures/accidents/quickindicat%or/documents/qi\\_2002.pdf](http://europa.eu.int/comm/transport/road/figures/accidents/quickindicat%or/documents/qi_2002.pdf), 2003.
- [7] Arroyo E., Selker T., and Stouffs A. Interruptions as multimodal outputs: which are the less disruptive? In *Proceedings of the Fourth IEEE International Conference on Multimodal Interfaces*. MIT Media Lab, Cambridge, MA, USA, 2002.

- [8] Wickens C. D. and Hollands J. G. *Engineering psychology and human performance*. Prentice Hall, third edition, 1999.
- [9] Belloti F. and De Gloria A. State of the art of driving support systems and on-vehicle multimedia hmi. Technical report, COMUNICAR, 2000.
- [10] McDonald M., Parent M., and Miller M. Stardust deliverable 1, critical analysis of adas/avg options to 2010, selection of options to be investigated. Technical report, European Commission, DG Research, Unit 1.5, 2002.
- [11] Vlavic L., Parent M., and Harashima F. *Intelligent Vehicle Technologies*, chapter 14, pages 446–457. Butterworth-Heinemann, 2001.
- [12] Directorate General Information Society Unit B5. Telematics applications for transport - project final reports 2000. Technical report, European Commission, Avenue de Beaulieu 31, 5th floor, 1160 Brussels, Belgium, September 2000.
- [13] B. Ulmer. Vita-an autonomous road vehicle (arv) for collision avoidance in traffic. In *Proceedings of the Intelligent Vehicles '92 Symposium*, pages 36–41, June 1992.
- [14] Nelson J.D.B. Applications of enhanced vision to the automotive industry. In *Proceedings of the SMi Conference on Enhanced Vision*, London, 2003.
- [15] Jahard F., Fish D.A., Rio A., and Thompson C.P. The fusion of near- and far-infra red images for optimal night-time automotive vision. In *6th Int. Conf. on Image Processing and Applications*. IEE IPA97, 1997.
- [16] Abernethy B. and Yang D. Stunned into action. *Traffic Technology International*, pages 60–67, Oct/Nov 2000.
- [17] Hoekstra E. Information overload? in-car internet on the intersate. *Traffic Technology International*, pages 25–28, April/May 2000.
- [18] Detsis G., Dritsas L., and Kostaras J. Information filtering and control for filtering the information overload problem. In *People in control: an*

*international conference on human interfaces in control rooms, cockpits and command centres*, number 81, June 2001.

- [19] Mulgund S.S. and Zacharias G.L. A situation-driven adaptive pilot/vehicle interface. In *Human Interaction with Complex Systems, 1996. HICS '96. Proceedings., Third Annual Symposium on*, pages 193–198. Charles Rivers Analytics, Cambridge, MA, USA, 1996.
- [20] Small R. L. and Howard C. W. A real-time approach to information management in a pilot associate. In *Digital Avionics Systems Conference Proceedings*, pages 440 –445. IEEE/AIAA, 1991.
- [21] Cha D. and Park P. User required information modality and structure of in-vehicle navigation system focused on the urban commuter. *Computers ind. Engng*, 33(3-4):517–520, 1997.
- [22] P.F. Spelt and D.R. Tufano. An in-vehicle information system for its information management. In *Proceedings of the 17th Digital Avionics Systems Conference*, volume 2, pages I31/1–I31/7, 1998.
- [23] Wevers K., Burns P., Portouli V., Saroldi A., and Widlroither H. Driver assistance and information systems — the in-arte way to integration. In *Proceedings of the 33rd ISATA 2000 Automotive & Transportation Technology Symposium -Safety/Interiors Track*, pages 305–314, 2000.
- [24] Zoutendijk A., Hoedemaeker D.M., Vonk T., Schuring O., Willemsen D., Nelisse M., and Van Katwijk R. Implementing multiple intelligent services in an intelligent vehicle with a safe workload aware hmi. In *Proceedings of the 10th World Congress and Exhibition on Intelligent Transport Systems and Services*, Madrid, Spain, 2003.
- [25] MOST cooperation, editor. *Media Oriented Systems Transport Specification Rev 2.2*. P.O Box 4327, D-76028 Karlsruhe, Germany, November 2002.
- [26] Anderson D. *Firewire System Architecture: IEEE 1394a*. PC system architecture. Addison-Wesley, Reading, Massachusetts, 2nd edition, 1998.

- [27] Institute of Electrical and Electronics Engineers, editors. *ANSI/IEEE Std 1212*. 1994.
- [28] Engström J. and Victor T. Real-time recognition of large scale driving pattern. In *Proceedings of the 2001 IEEE Intelligent Transportation Systems Conference*, Oakland, CA, USA, 2001.
- [29] Löfstedt J. and Svensson M. Baltazar, a fuzzy expert system for driving situation detection. Technical report, Department of Information Technology, Lund Institute of Technology, Lund, Sweden, 2000.
- [30] Hauptmann W., Graf F., and Heesche K. Driving environment recognition for adaptive automotive systems. In *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems*, volume 1, pages 387–393, 1996.
- [31] Qiao L., Sato M., Abe K., and Takeda H. Self-supervised learning algorithm of environment recognition in driving vehicle. In *IEEE Transactions on systems, man, and cybernetics*, volume 26, 1996.
- [32] Yamada M., Ueda K., Horiba I., and Sugie N. Discrimination of the road condition toward understanding of vehicle driving environment. In *IEEE Transactions on Intelligent Transportation systems*, volume 2, 2001.
- [33] Harbluck J.L. and Ian Noy Y. The impact of cognitive distraction on driver visual behaviour and vehicle control. Technical report, Ergonomic division, Road Safety Directorate and Motor Vehicle Regulation Dir., Transport Canada, Transport Canada, 330 Sparks Street, Ottawa, ON K1A 0N5, CANADA, 2002.
- [34] Broadbent D.E. *Perception and communication*. Pergamon, London, 1958.
- [35] Wickens C.D. *Varieties of attention*, chapter Processing resources in attention, pages 63–102. London Academic Press, 1984.
- [36] Kahneman D. *Attention and effort*. Prentice Hall, New Jersey, U.S.A., 1973.

- [37] Wickens C. D. *Engineering psychology and human performance*, chapter 9. HarperCollins Publishers, second edition, 1992.
- [38] Wickens C. D. Multiple resources and performance prediction. *Theoretical Issues in Ergonomics Science*, 3(2):159–177, 2002.
- [39] Nakayama O., Boer E.R., Futami T., and Nakamura T. Development of a steering entropy method for evaluating driver workload. In *International Congress and Exposition, Human Factors in Driving*, number 1999-01-0892, Detroit, MI, USA, March 1999.
- [40] Kosko B. Fuzzy cognitive maps. *International Journal of Man Machine Studies*, pages 65–75, 1986.
- [41] Zadeh L.A. Fuzzy sets. *Information and Control*, 8, 1965.
- [42] Koulouriotis D.E., Diakoulakis I.E., Emiris D.M., Antonidakis E.N., and Kaliakatsos I.A. Efficiently modeling and controlling complex dynamic systems using evolutionary fuzzy cognitives maps. *International Journal of Computational Cognition*, 1(2):41–65, 2003.
- [43] Stylios C.D. and Groumpos P.P. Fuzzy cognitives maps: a model for intelligent supervisory control systems. *Computers in Industry*, 39:229–238, 1999.
- [44] Dickerson J.A. and Kosko B. Virtual worlds as fuzzy cognitives maps. In *IEEE Virtual Reality Annual International Symposium*, pages 471–477, 1993.
- [45] Callan R. *The essence of neural networks*. The essence of computing. Prentice Hall, first edition, 1998.
- [46] Dugarry A., Berrada A., and Fu S. Facial expression presentation for real-time internet communication. In *Internet Imaging IV, Proceedings of SPIE/IS&T*, volume 5018, pages 89–98, Santa Clara, CA, January 2003.
- [47] Dugarry A., Fu S., Nelson J., and Wang W. *D-BITE specifications*. Cranfield University, Cranfield, UK, April 2002.

- [48] Siever E., Spainhour S., Figgins S., and Heckman J.P. *Linux in a Nutshell*. O'Reilly & Associates, Inc., third edition, 2000.
- [49] Dugarry A. Data synchronisation over 1394. Technical report, Road-sense, Paris, 2002.
- [50] Linux 1394 workgroup. Ieee-1394 for linux. <http://www.linux1394.org>.
- [51] Nikishkov G.P., Nikishkov Yu.G., and Savchenko V.V. Comparison of c and java performance in finite element computations. *Computers and Structures*, (81):2401–2408, 2003.
- [52] Walker W., Lamere P., and Kwok P. Freetts - a performance case study. <http://research.sun.com/techrep/2002/abstract-114.html>, 2002.
- [53] Cowell-Shah C.W. Nine language performance round-up: Benchmarking math and file i/o. [http://osnews.com/story.php?news\\_id=5602](http://osnews.com/story.php?news_id=5602), 2004.
- [54] Dibble P.C. *Real-Time Java Platform Programming*. Prentice Hall, 1st edition, March 2002.
- [55] Northover S. Swt: The standard widget toolkit. <http://www.eclipse.org/articles/Article-SWT-Design-1/SWT-Design-1.html>, 2001.
- [56] Nelson J.D.B. Gaze tracking with subspace selection of kernel eigenfeatures. *Pattern Recognition*, in review.
- [57] Broadley S. and Hird D. Focus on personal travel. National Statistics publication, Departement for Transport, Local Government and the Regions, TSPT2, Zone 2/15, Great Minister House, 76 Marsham Street, London SW1 P4RD, United Kingdom, 2001.
- [58] Dugarry A. and Berrada A. Internet survey: Internet emotion communication. <http://amac.soe.cranfield.ac.uk/facexp/results.php>. last accessed 28 October 2002.

- [59] Petit V. Face to face web communication. Master's thesis, Cranfield University, 2002.
- [60] Darwin C. *The Expression of the Emotions in Man and Animals*. Appleton & Company, 1872.
- [61] Ekman P. Facial expression and emotion. *American Psychologist*, 48(4):384–392, 1993.
- [62] World Wide Web Consortium. Scalable vector graphics (svg) specification w3c recommendation 04 september 2001. <http://www.w3.org/TR/SVG/>. last accessed 28 October 2002.
- [63] Pavlidis T. *Algorithms for Graphics and Image Processing*. Computer Science Press, Rockville, Maryland, 1982.



# Appendix A

## Facial expression presentation for real-time Internet communication

This paper was presented at the Electronic Imaging 2003 Symposium, in Santa Clara, USA, in January 2003.

# Facial expression presentation for real-time Internet communication

Alexandre Dugarry, Aïda Berrada and Shan Fu

Centre for Computer Vision and Image Processing, School of Engineering,  
Cranfield University,  
Cranfield, Bedfordshire, United Kingdom, MK43 0AL.

## Abstract

Text, voice and video images are the most common forms of media content for instant communication on the Internet. Studies have shown that facial expressions convey much richer information than text and voice during a face-to-face conversation. The currently available real time means of communication (instant text messages, chat programs and videoconferencing), however, have major drawbacks in terms of exchanging facial expression. The first two means do not involve the image transmission, whilst video conferencing requires a large bandwidth that is not always available, and the transmitted image sequence is neither smooth nor without delay. The objective of the work presented here is to develop a technique that overcomes these limitations, by extracting the facial expression of speakers and to realise real-time communication. In order to get the facial expressions, the main characteristics of the image are emphasized. Interpolation is performed on edge points previously detected to create geometric shapes such as arcs,

lines, etc. The regional dominant colours of the pictures are also extracted and the combined results are subsequently converted into Scalable Vector Graphics (SVG) format. The application based on the proposed technique aims at being used simultaneously with chat programs and being able to run on any platform.

## A.1 Introduction

Nowadays the most widely used real-time communication means are instant messaging (?chat? programs, ICQ), audio and video conferencing programs. These applications are usually easy to use and free, which makes them very popular with private users. However, a quick survey [58] carried out among thirty local Internet users confirms that these applications are not entirely satisfying.

Firstly, around 72% of the users think that the presently available means of real-time Internet communication do not convey emotions accurately. Chat users are often limited to a range of smileys (short combination of characters, e.g. :-) for a smile) or emoticons (small pictures symbolizing their feelings) they have to insert into their text messages. This does not show the spontaneous reactions that can be read on a face.

Secondly, the video conferencing applications transmit an image, the quality of which heavily relies on the bandwidth available. One could argue that this problem will disappear, as broadband Internet access will be made cheaper and more widely accessible. However, another drawback is that people do not necessarily wish to be seen as they talk. They might want to keep their anonymity, or prefer to be seen when they are ‘at their best’, which might not be the case at the time of the communication. The survey shows that this is true for 62% of the persons who answered the questionnaire.

The work presented in this paper aims to transmit a real time comprehensible representation of the facial expressions, whilst preserving the visual anonymity of the user.

An earlier approach to the problem was to automatically create a symbolic representation of the face expression [59]. In real life, the human emotions can be read from his face; these expressions can be very different from one

person to another, but the same person will produce similar expressions for similar emotions. This first approach read the mood of a user from a picture of his face, and computed the symbol to send. For every user, the first necessary stage was to establish a library of expressions. The user was asked to smile, laugh, express sadness or surprise as pictures were taken, processed and stored. But this method presented two drawbacks. The first one was the speed of the current template matching mechanism, which did not meet the real-time requirements. The second and certainly major one, was inherent to the method itself: only predefined expressions could be matched. It was assumed that the user would be able to simulate accurately any kind of emotion at the application set-up time, but this was not an easy task. Furthermore the combination of the facial features (lips, eyebrows, etc.) can express a great number of emotions, for which it was very difficult to define and to associate a representative symbol.

The new approach, presented here, uses an abstract representation, by converting the most meaningful features of the image into a vector format, and transmitting these data over the Internet. It is first necessary to remove the noise from the image and to simplify it whilst preserving the most meaningful features of the face. This step includes reducing the number of colours and detecting the edges. The image is then vectorised. To be fully portable, the application has been developed in Java, using the Java Media Framework (JMF) and the Java Advanced Imaging (JAI) API. The data produced are formatted using the Scalable Vector Graphics (SVG) language.

## **A.2 Methodology**

A way to overcome the limitations of the current Internet real-time communication is to transmit, in real-time, an abstract representation of the captured image. In order to get the facial expression, the main characteristics of the picture are emphasized. Edge-detection is applied and the features are extracted. Then, interpolation is performed on the detected elements to create vectors describing geometric shapes such as arcs, lines, etc.



Figure A.1: Median filtering

### A.2.1 Features extraction

The first difficulty to tackle when dealing with facial expressions is to define how they can be characterized and how the human brain understands them. Since Darwin[60], the central preoccupation of researchers interested in the face is to correlate movements of the face with emotional states. They agree with the fact that emotions are central in explaining facial movements[61].

It is interesting to note that most of the information is communicated by the eyebrow's relative orientation and the shape of the mouth. For example, raising or lowering eyebrows can be used as a question mark, whereas a facial shrug, meaning 'I don't know', can be expressed by the corners of the mouth being pulled up or down. By transmitting only the shapes of these facial features, emotions could be communicated.

#### De-noising

Some major difficulties encountered in feature detection are due to noise and variation in luminance. Since the source images are captured from a standard webcam, their quality is quite poor and the amount of noise is more significant with low-resolution images. Noise in images is generally a high frequency effect and corrupts data where the energy spectrum is low; most image-enhancing techniques aim at eliminating features that are random and uncorrelated.

To mitigate these affects, a median filter is applied, which consists in replacing each pixel value with the median value of its neighbours and itself. This attenuates any sudden jumps, often caused by noise, but preserves step edges without too much blurring. The process is illustrated in Figure A.1.

## Edge detection

Geometrically speaking, facial features can be considered as edges, i.e. boundaries between two dissimilar regions in the image. Edges are detected in areas where the intensity level fluctuates sharply, the quicker the intensity change the stronger the edge. A good edge detection stage makes the formation of extended boundaries and object recognition easier. Although edges provide clues that aid the analysis, they can be affected by the noise present in an image. Therefore, detected edges can occur in places where the transition between regions is not abrupt enough or else edges can be detected in regions of the image where the texture is uniform. For this reason, noise filtering is first applied so that the information is not occluded by any other uncorrelated data. The most common type of edge detection process uses a gradient operator, of which there have been several variations. Mathematically, for an image  $f(x, y)$ , the gradient magnitude,  $g(x, y)$  is computed as:

$$g(x, y) \cong (\Delta x^2 + \Delta y^2)^{\frac{1}{2}}$$

where  $\Delta x = f(x + 1, y) - f(x - 1, y)$  and  $\Delta y = f(x, y + 1) - f(x, y - 1)$

For example, the simplest implementation of this would be to convolve the mask  $[-1, 0, 1]$  with the image data, aligning the mask with the  $x$  and  $y$  axes to compute the values of  $\Delta x$  and  $\Delta y$ .

Variations on this theme have included the Roberts, Prewitt and Sobel operators. In practice, the Sobel operator is by far the most extensively used. In this case the masks are extended to a 3 by 3 neighbourhood, rather than the 3 by 1 neighbourhood given above. The  $x$  and  $y$  masks given below are first convolved with the image to compute the values of  $\Delta x$  and  $\Delta y$ . Then the magnitude of the edges is computed from these values.

Sobel Operator convolution kernels:

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

x – direction                      y – direction

## Thresholding

A drawback of the Sobel operator is that a response to a step edge is present over either two or three pixels width. This necessitates the use of post-processing like thresholding in order to reduce the complexity of further processing. The Threshold operation consists of segmenting an image into regions of similarity by grouping pixels within a common range of grey level into a predetermined set. This process enables one to convert the multi-banded images (colour images) into single banded images by assigning value 255 to each pixel  $(x, y)$  above a given threshold and assigning value 0 to others.

### A.2.2 Raster to vector conversion

The pictures that need to be processed contain many different features and the edges are of no particular kind. For example the mouth's shape varies a lot depending on the expression on the persons. Some thinning based algorithms have been tested, but they proved to give very poor results, removing too many pixels. This is due to the set of rules used for deleting points.

Therefore, an algorithm has been written to extract free-form paths from the binary images previously generated. It extracts sets of continuous pixels and creates lists that describe the pixels as paths or strokes.

Consider on a raster image an arbitrary region consisting of white pixels on a black pixel background. This region may be of arbitrary complexity. The algorithm gives a method by which the pixels in the contour of the region are returned in a list, such that it represents a series of connected pixels that indicate the contour or path. One of its special features is that it handles the path following without any sorting of the data.

It is assumed that the incoming raster image is single-banded (but the algorithm can be easily adapted to multi-banded images) and reduced to edges only. The width of the edges does not matter. The algorithm is made up of two phases: The first one consists of extracting the different regions of colour in the image. The second phase segments each extracted region into several lists of connected pixels that constitute the different paths in the image.

## First phase

Most of the facial features (eyebrows, eyes, mouth) are more horizontally inclined. Therefore the algorithm begins with scanning the raster image vertically from upper left to lower right.

Each time a white pixel is encountered a path following procedure starts. The procedure stores the pixel if there is a black pixel within a 3x3 neighbourhood. It then iterates to the next pixel in the vertical scan until there is no pixel of the same colour left. The whole process repeats until there is no unvisited pixel left.

At the end of this phase of the algorithm, there are as many generated vectors as there are colours in the image (one in this case).

To mark the visited points, a binary mask of the same size of the image is used. Whenever a point is visited, the corresponding point in the mask is set to true. The purpose of this is to avoid detecting the same point several times.

## Second phase

The second phase calculates a set of coherent paths from the different extracted regions.

For each pixel of each vector previously created, each of the eight surrounding positions is searched within the vector itself to determine where the consecutive pixel is. If no point is found within the 8-pixel neighbourhood, the algorithm is extended to one more pixel neighbourhood. If no point is found within this additional area, then the path is considered to be found and the procedure moves to the next unvisited pixel. This allows compensating small breaks in the input image.

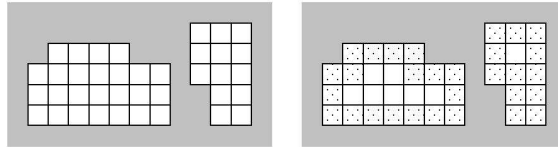
The process stops when all the points in the vector have been visited.

To avoid searching the whole vector for neighbours at every iteration, a special array, called limits, is created and filled with the cumulated number of elements found in each column of the raster image during the first phase. For example if 50 pixels have been detected in the 10th column of the image and if the total number of pixels detected in the 9 previous columns is 250, then  $\text{limits}[10] = 300$ . The purpose of this array is to give the range within which

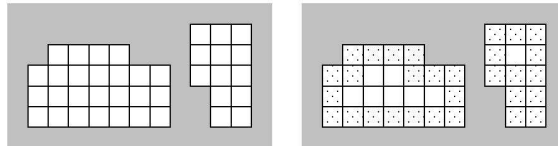


the vector generated in the first phase has to be searched for neighbours.

Figure A.2 illustrate the two phases of the algorithm.



(a) First phase



(b) Second phase

Figure A.2: Vectorisation algorithm

The complexity of this algorithm is of the order  $O(N)$  where  $N$  is the total number of pixels, but in practice, it is much reduced by the use of lookup tables, the binary masks and the limits array.

### A.2.3 Conversion to SVG

SVG stands for Scalable Vector Graphics and is a language used to describe graphics using XML syntax. At the time of writing, graphics on the Internet are usually bitmap files (JPEG, GIF, PNG): every pixel of the image has to be described, even if compression techniques are applied afterwards. This means that a bigger, higher quality image will necessarily have to use more memory.

Conversely, a vector graphics format describes shapes (lines, curves, circles, etc.) and not pixels. This gives a major advantage over standard image formats: the memory occupancy of a vector image does not depend on its size, or quality.

SVG has been a World Wide Web Consortium (W3C) recommendation since September 2001 [62]. Before that, the only vector graphics formats available and designed for Internet were proprietary, such as Macromedia Flash or Apple QuickTime. Being an open standard and in text form, SVG can be supported by a broad range of applications.

It was described earlier how a set of vectors was obtained, by expressing free form paths as a sequence of pixels. However, the conversion into SVG format requires a mathematical expression. Therefore it is necessary to find curves that pass through (interpolation) or near (approximation) the set of points previously computed.

Curve fitting refers to both interpolation and approximation. From a mathematical point of view, interpolation problems are easier to solve but approximation is more realistic in many applications where the exact values of the data are subject to noise corruption.

B-splines are a popular mathematical tool for curve fitting. In particular, SVG uses a special case of B-splines, namely Bezier polynomials, to define the paths within an image. Pavlidis [63] gives an algorithm to compute a Bezier polynomial from a set of points, which has been adapted and used here. The SVG conversion is then a straightforward procedure.

## **A.3 Application**

### **A.3.1 Description**

The application has been designed to transmit the face expressions of two or more users over the Internet in real-time, whilst preserving their anonymity.

It is composed of two parts, namely a client and a server:

- The server part grabs the images from the camera, processes them and creates the SVG files
- The client part receives the data and renders the SVG file to the screen.

Every computer running the application has to connect to a server in order to distribute the produced SVG graphics to other connected clients. In

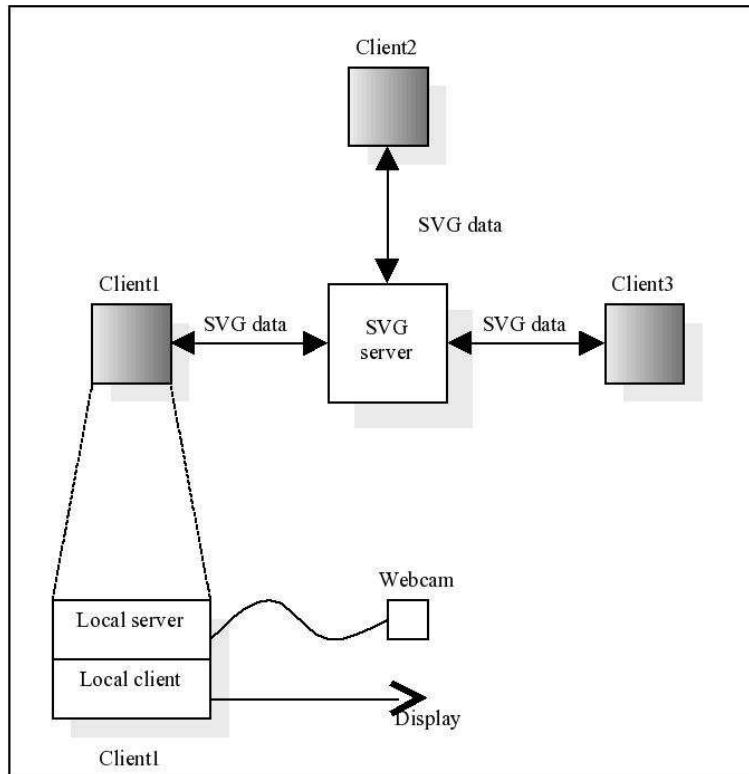


Figure A.3: Networking architecture

this implementation, the server's only role is to send every packet it received to all the registered clients, as shown in Figure A.3.

Because the application is aimed at running on different environments, Java is the language of choice. The Java Media Framework is used to capture images from a standard web cam, whereas the Java Advanced Imaging API provides improved image processing functions. Java also makes easier the implementation of the network communications (via sockets) and the creation of multiple threads handling the several tasks (grabbing, processing, compression, communication, rendering).

### A.3.2 Results

Originally developed on a Linux environment, the application has been tested on a computer running Microsoft Windows. This ensures that the program is portable. The webcam is a Philips USB Vesta 680K. Three image capture

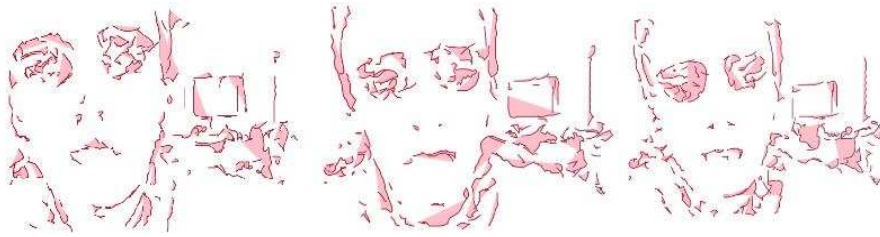


Figure A.4: Rendered SVG with 160x120 source images

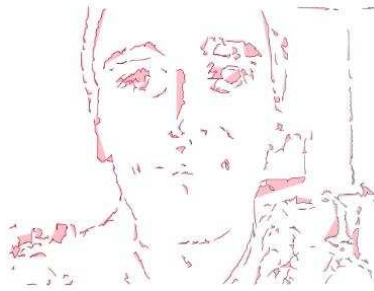


Figure A.5: Rendered SVG with 320x240 source images

sizes have been used: 640x480, 320x240 and 160x180. The bigger the image, the more detailed the vectorization will be, but it will also take more time to compute, send, and render.

The figures A.4, A.5, and A.6 show the image displayed to the clients, with different source image sizes. It is important to note that the application displays the SVG files with the same size, independently of the source image; in this paper the 160x120 images are shown smaller, as they are less meaningful.

Some performance tests have been conducted. Every time a frame is displayed, the current time is recorded. Figure A.7 shows the results. The values measure the total processing time, from image capture to SVG rendering.

The computer used for these tests is Pentium III based, running at 1GHz, with 512MB of RAM.

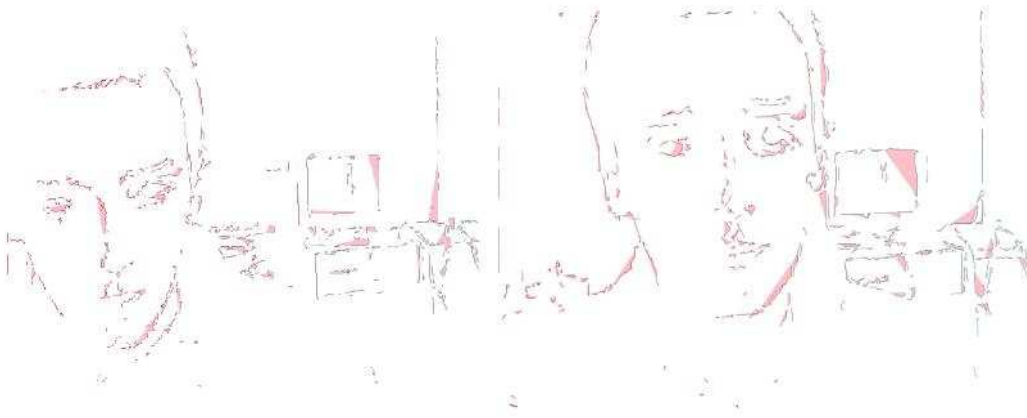


Figure A.6: Rendered SVG with 640x480 source images

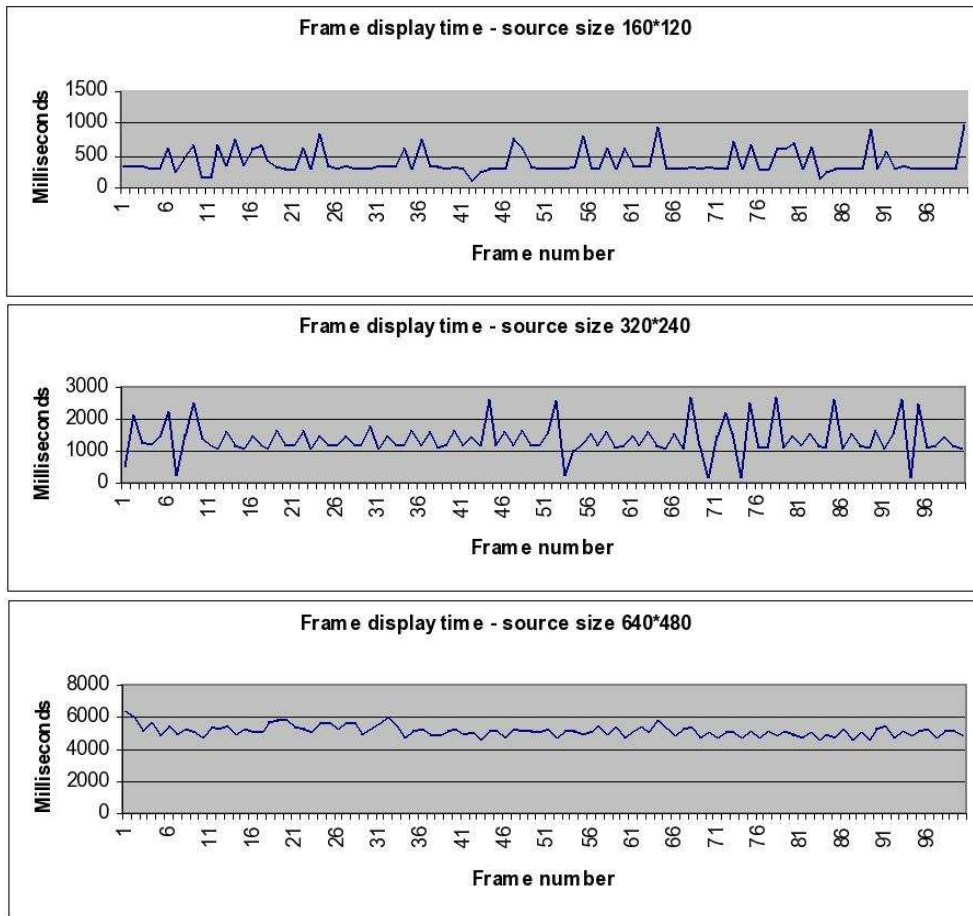


Figure A.7: Performance measurements

## A.4 Conclusion and further work

This paper presented an approach to communicate emotions by transmitting the facial expressions in real time over the Internet, whilst preserving the anonymity of the users. An application has been created, that takes images from a web camera, sends a Scalable Vector Format (SVG) over the Internet, and displays the received data. Because it aims to run on multiple environments, it has been developed in Java. The use of an open standard, SVG, allows the program to be easily linked with other Internet applications. The results show that with a medium sized source image, the emotions represented by the facial expressions are comprehensible; a bigger image gives a more accurate vectorization but needs much more time to be displayed, and does not bring more information about the emotions of the user.

Further work will focus on improving the vectorization procedure, by using other segmentation techniques, and post-processing the images. An additional concern will be the optimisation of the algorithms, to gain performance and increase the display frequency.

# Appendix B

## IEEE-1394 Data Transfer

### ctmu.h

```
#include <libraw1394/raw1394.h>
#include <stdlib.h>
#include <stdio.h>
#include <signal.h>
#include <sys/stat.h>
#include <libraw1394/csr.h>
#include <pthread.h>

#define FILENAMELENGTH 255
#define FILESIZELENGTH 4
#define BUFFERSIZE 2048

#define BUFFERADDRESS (CSR_REGISTER_BASE + 0x3000)

#define MINIMUM_PACKET_SIZE 8
#define COMMAND_GET_FILE 0x0005
#define COMMAND_GET_FILE_LIST 0x0006
#define COMMAND_SEND_FILE 0x0007
#define COMMAND_SEND_FILE_LIST 0x0008
#define COMMAND_RESPONSE 0x0009
```

```

#define APP_LAYER_PROTOCOL_ID 0xA

#define RESPONSE_CODE_OK      0x1

/**
 * These two structures describe the data format.
 * commonPacketHeader is sent at the beginning of
 * EACH packet, whereas firstPacketHeader is sent
 * only with the first packet
 */

typedef struct commonPacketHeader {
    unsigned short  pid;
    unsigned short  transaction_uid;
    unsigned int    packets_count;
} commonPacketHeader;

typedef struct firstPacketHeader {
    unsigned int    packets_number;
    unsigned int    command;
} firstPacketHeader;

/* response packet handler */
typedef struct response_context {
    struct response_data *responseData;
    raw1394handle_t     h;
} response_context;

/* This structure represents a response data */
typedef struct response_data {
    struct commonPacketHeader commonHeader;
    struct firstPacketHeader  firstHeader;
    unsigned int              response_code;
    unsigned int              packets_count;
} response_data;

```



```

/** Handles on transactions
 * Allow to store the variables required
 * during the full length of a transaction
 */
typedef struct transaction {
    struct commonPacketHeader *commonHeader;
    struct firstPacketHeader *firstHeader;
    unsigned int firstPacketHeader_read;
    unsigned int isHeader;
    char *data;
    char *databuffer;
    int dataOffset;
    unsigned int node;
    int data_length;
    pthread_t thread;
    pthread_cond_t condition;
    pthread_mutex_t condition_mutex;
    pthread_mutex_t transaction_mutex;
    pthread_mutex_t *getfile_condition_mutex;
    pthread_cond_t *getfile_condition;
    raw1394handle_t h;
    struct response_context *response;
    void *specific;
} transaction;

typedef struct transaction_sendFile {
    unsigned int filename_size;
    char *filename;
    unsigned char *ptr_filename;
    unsigned int filenameSize_read;
    unsigned int filename_read;
    int remaining ;
    FILE *out;
} transaction_sendFile;

```

```

typedef struct transaction_getFileList {
    unsigned int    request_sent;
    unsigned short  transaction_uid;
    char            *localpath;
    char            *filelistname;
} transaction_getFileList;

```

```

typedef struct thread_starter {
    char            *filelistname;
    int             node;
    raw1394handle_t h;
} thread_starter;

```

```

/**
 * File listing
 */
char *datapath;
char *default_filelistname;

```

```

/*****
 * Registry and transaction management
 *****/
#define REGISTRY_SIZE 64

```

```

pthread_mutex_t registry_mutex;
struct transaction **registry;

```

```

/* Return current handle*/
raw1394handle_t *getHandle();

```

```

/* Create new transaction */

```

```

struct transaction * createTransaction(
    struct commonPacketHeader *);

/* Return transaction corresponding to transactionID */
struct transaction * getTransaction(
    unsigned short transactionID);

/* Add transaction to registry */
int addToRegistry(
    struct transaction *tr);

/* Remove transaction from registry*/
int removeFromRegistry(
    struct transaction *tr);

/* Initialise registry */
void initRegistry();

/* Destroy registry and free all related resources*/
void destroyRegistry();

/*****
 * Directories-related functions
 *****/
char **nodesIDs;

/* Browse the network and
 * fill the array nodesIDs*/
void getNodesIDs(raw1394handle_t handle);

/* Build a folder for each node on the network */
void buildNodesFolders(raw1394handle_t handle);

/* Check if given path is a directory */
int isDirectory(char *path);

```

```

/* Create directory */
void buildDirectory(char *path);

char * prepareDirectory(
    char *filepath ,
    unsigned int node);

/*****
* Send and Receive commands
* These functions are called when a new
* command is received.
* Four different commands can be sent:
* " SendFile" , " GetFile" , " SendFileList ,
* and " GetFileList".
* The function starting with 'th' are
* designed to be run in their own thread.
*****/
void *thReceive_getFileList(void *ptr);
void *thReceive_getFile(void *ptr);
void *thReceive_SendFile(void *ptr);
void *thReceive_SendFileList(void *ptr);
void *thsendCommand_SendFile(void *ptr);
void *thsendResponse(void *ptr);

int ReceiveCommand_getFileList(
    raw1394handle_t h,
    struct transaction *tr ,
    struct raw1394_arm_request *arm_req ,
    unsigned int offset ,
    char *filelistname ,
    char *localpath);

void ReceiveCommand_SendFileList(
    raw1394handle_t h,
    struct transaction *tr ,

```

```

    struct raw1394_arm_request *arm_req ,
    unsigned int offset );

int ReceiveCommand_getFile(
    raw1394handle_t h ,
    struct transaction *tr ,
    struct raw1394_arm_request *arm_req ,
    unsigned int offset );

void getFilesFromList(char *filelist , unsigned int node);

int sendCommand_SendFile(
    raw1394handle_t h ,
    unsigned int dist_node ,
    char *filename ,
    unsigned short transaction_uid ,
    unsigned short command);

int sendCommand_getFileList(
    raw1394handle_t h ,
    unsigned int dist_node );

```

# Appendix C

## Data replay, simulation and analysis

Figure C.3 shows the D-BITE player interface replaying data acquired with an experimental vehicle, including two video streams, CAN data and gaze tracker data. The red rectangle shows the areas being looked at. In the bottom window, each red bar corresponds to one recorded event.

Figure C.4 is a view of the simulation application. In the top left corner lies the control panel windows; The FCMs windows are on the right-hand side.

The UML class diagrams hereafter illustrates the structure of the D-BITE replay application. Note that, for readability purposes, the diagrams are not fully developed.

At last, Figure C.5 and C.5 presents the data analysis and simulation in action.

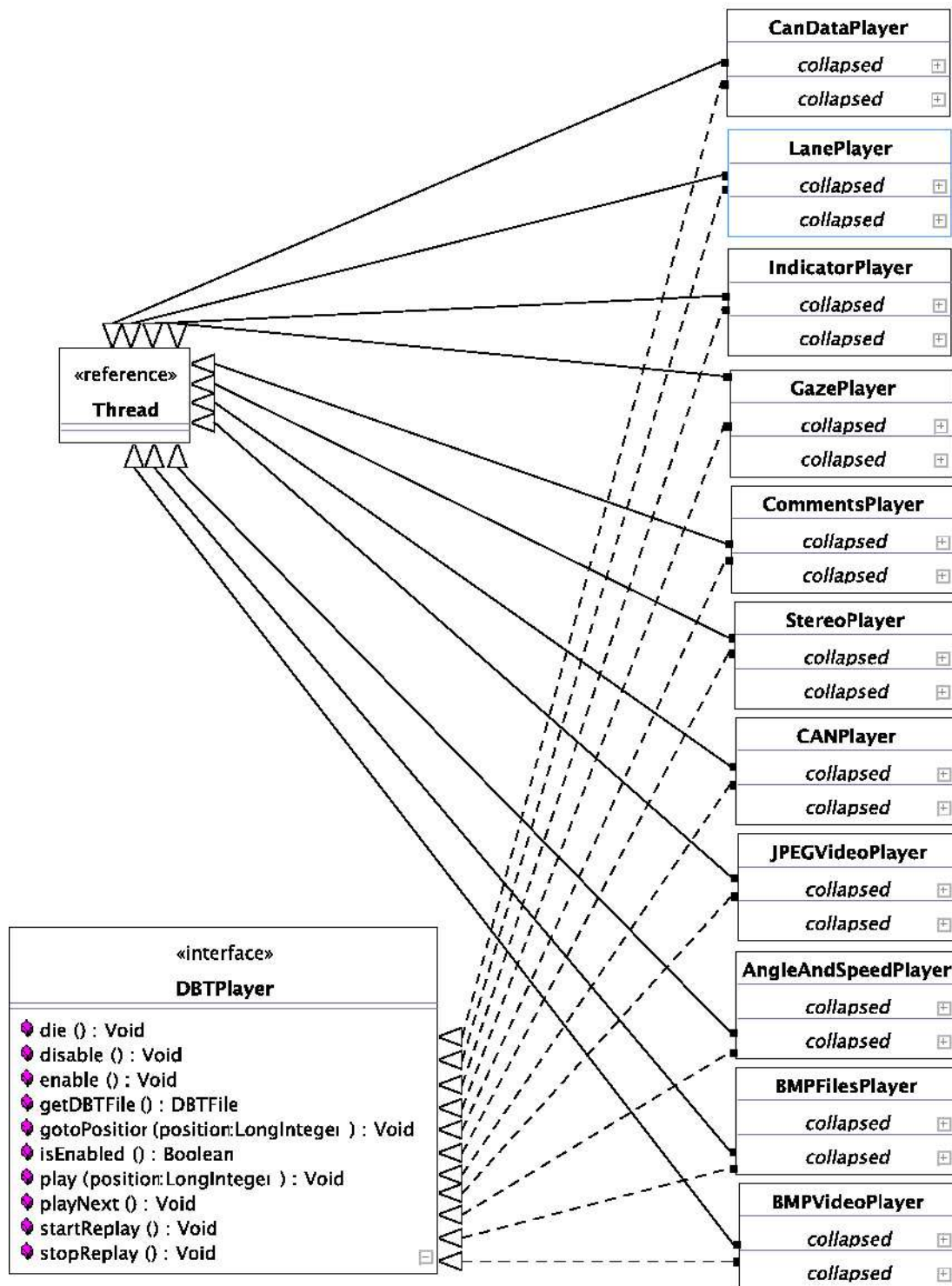


Figure C.1: UML view of player class

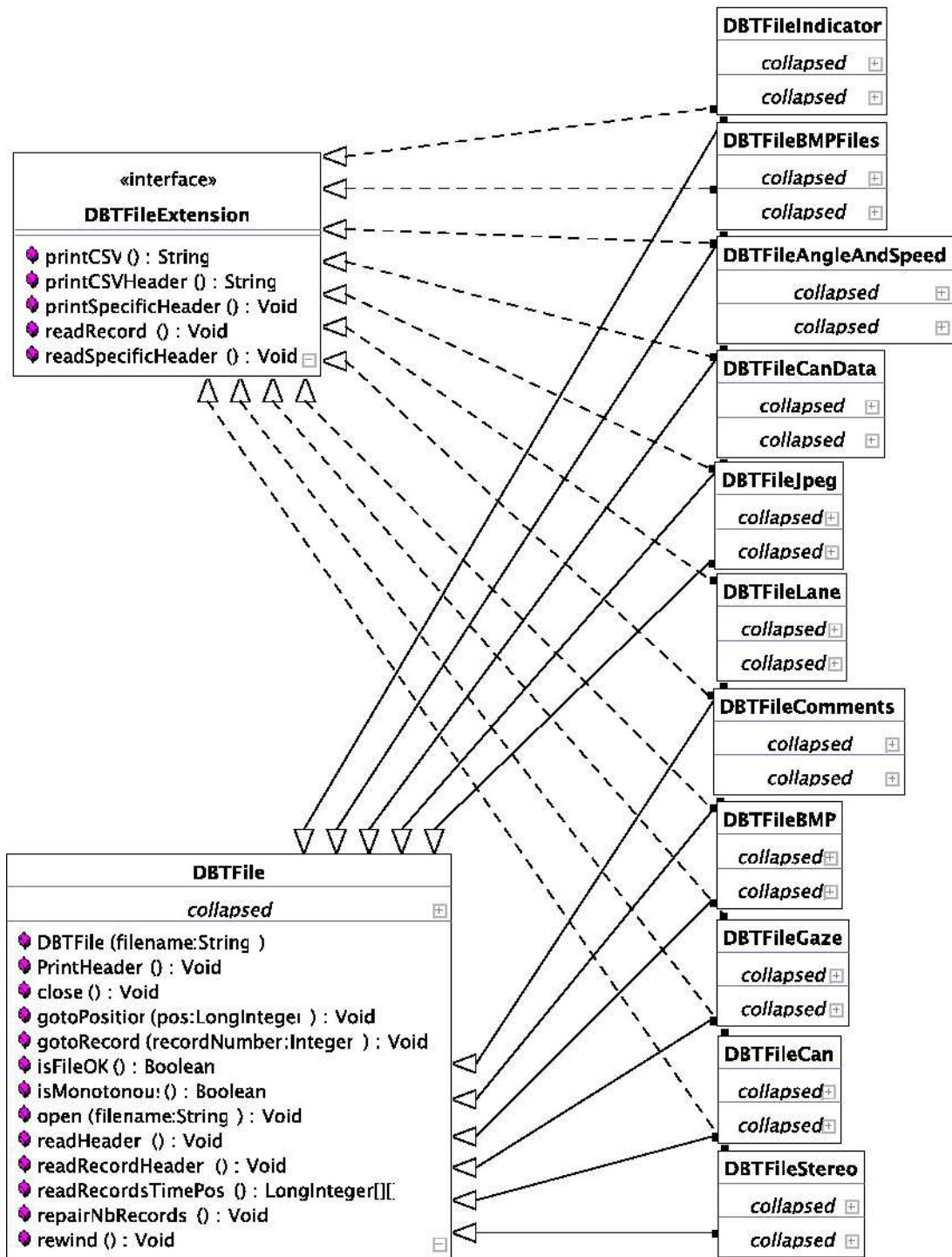


Figure C.2: UML view of file class





Figure C.3: Player view

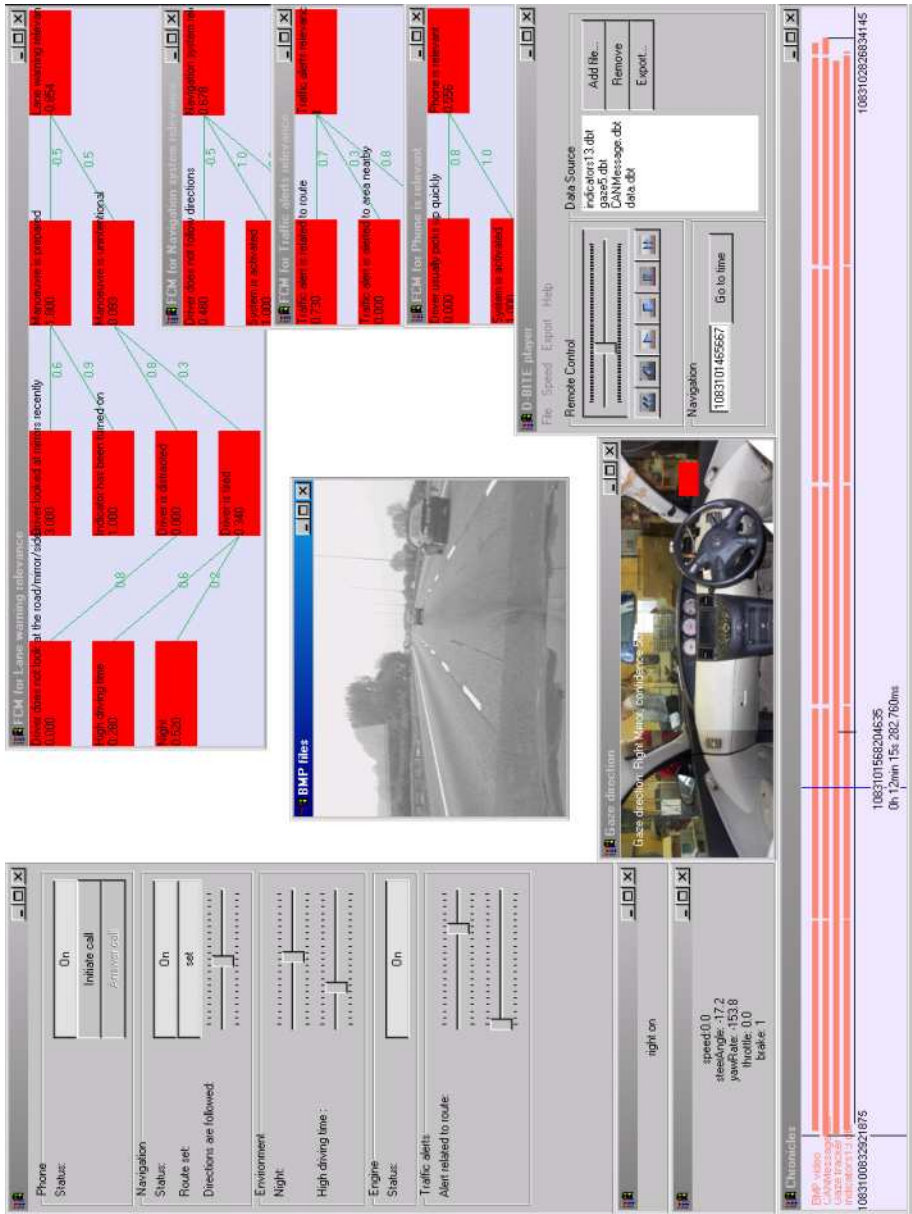


Figure C.4: Simulation view

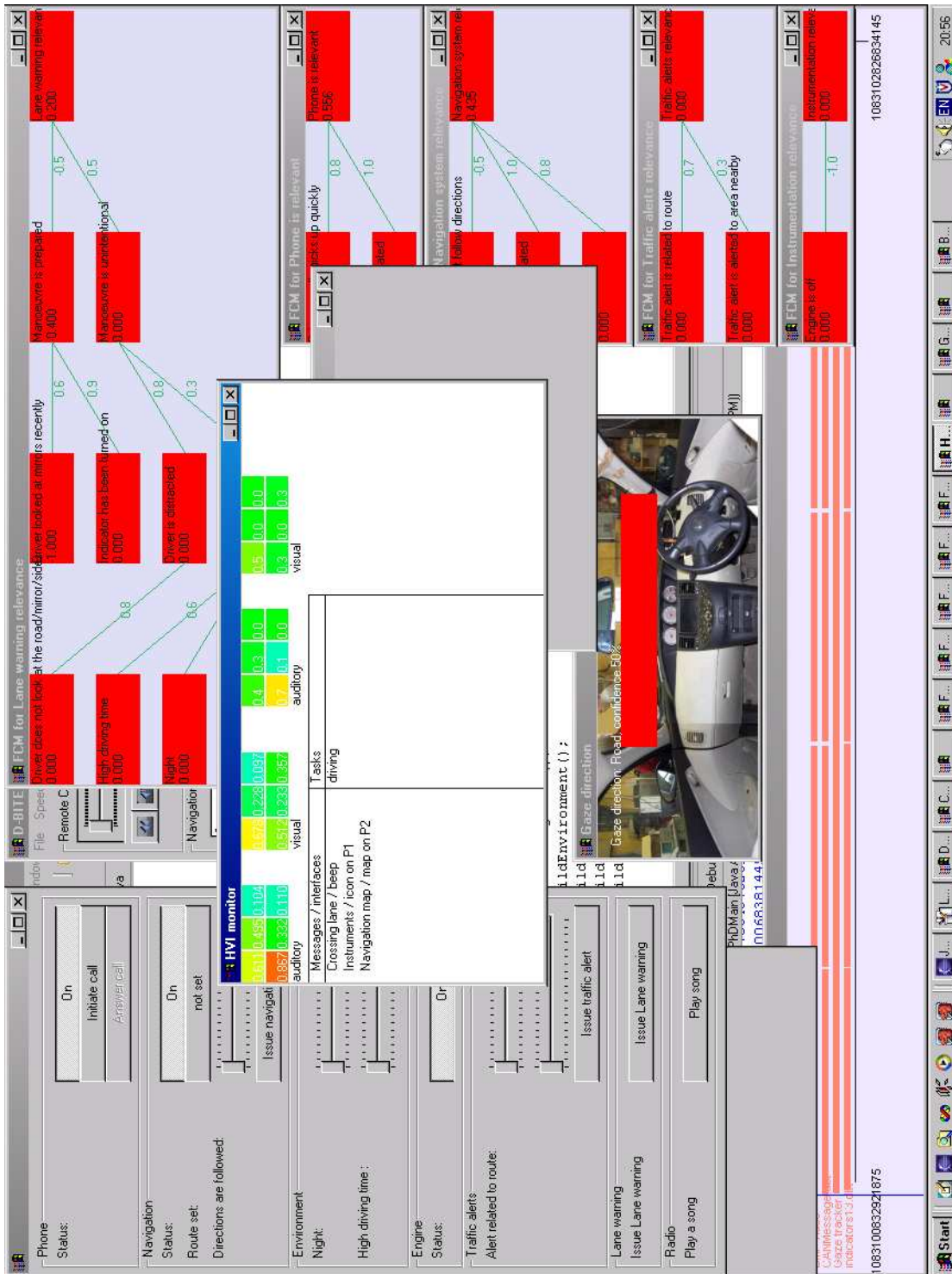


Figure C.5: Experimental data prioritisation - lane warning

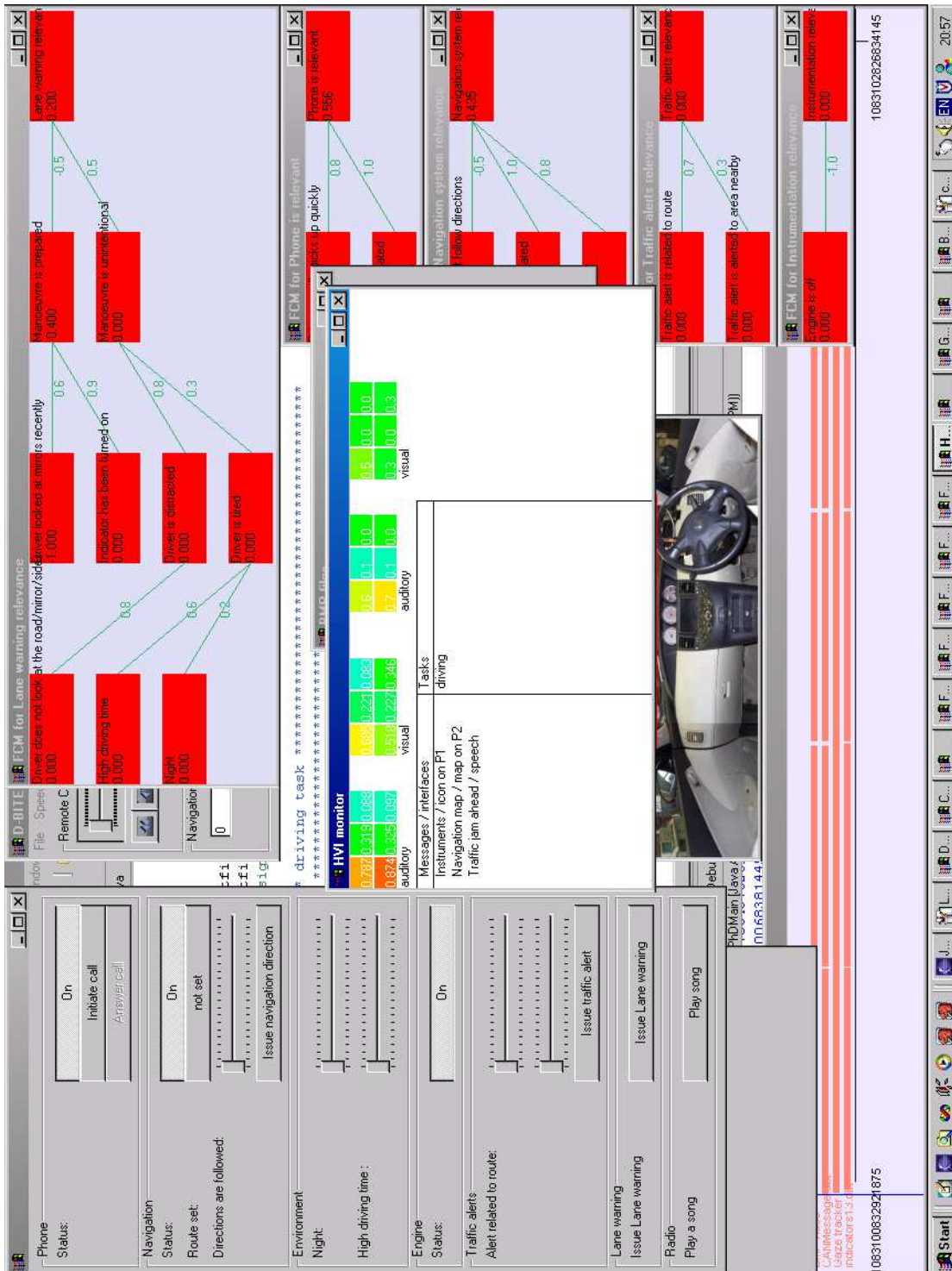


Figure C.6: Experimental data prioritisation - traffic alert