Cooperative Intelligent Transportation Systems

Building a demonstrator for the CVIS-project on the Informatieve Weg

ing. Bobby J.W. Nijssen

February 12, 2008

Master's Thesis
Chair of Information Systems
Department of Computer Science
University of Twente

This thesis has been supervised by:

dr. ir. Maurice van Keulen, University of Twente
dr. ir. Nirvana Meratnia, University of Twente
drs. Marcel Konijn, LogicaCMG Nederland B.V.
ing. Ilske Verburg, LogicaCMG Nederland B.V.

Abstract

Within the next decades Intelligent Transportation Systems (ITS) will become more *cooperative* in the sense that vehicles and road-side infrastructure will work together in their task to create a safer, more efficient and more comfortable environment for everybody.

The Cooperative Vehicle-Infrastructure Systems (CVIS) project is one of those initiatives that is making this transition possible. For CVIS it can be (economically) beneficial to re-use existing road-side information systems and put those systems in a cooperative context. The Informatieve Weg (IW) or 'Informative Road' is an example of a *non*-cooperative ITS that has the chance to become truly cooperative.

This thesis reports on the process to realize a demonstration of a cooperative ITS concept that could be integrated with the IW's systems, all within the context of the CVIS project.

A taxonomy of cooperative ITS concepts is presented, which is based on communication-paths. It differentiates between different levels of cooperation, e.g. the vehicle and infrastructure communicate only one-way or truly work together. The concepts with the highest level of cooperation, i.e. *both* parties share information, are referred to as the 'vehicle infrastructure integration (VII) concepts'.

The CVIS project focuses on these VII concepts: it will provide solutions where the vehicle and infrastructure are tightly cooperating. The taxonomy is used to find a match between several discovered VII concepts and the already existing road-side information systems of the IW. This resulted in the selection of a cooperative ITS concept that was suitable to demonstrate: *cooperative weighing*. This concept is about informing a vehicle of its weight, measured by a road sensor.

A *generic VII framework* has been implemented and provides a two-way communication-layer between vehicles and infrastructure. The framework is suitable to demonstrate VII concepts in general and it can evolve further to form the basis of a real-life VII solution. The cooperative weighing concept uses the facilities of the framework and therewith exercises the framework's architecture.

Due to organizational lingering it was impossible to actually integrate the framework with the 'weighing module' of the IW. A simulation setup was used instead. This thesis provides the next steps to realize this integration.

The actual demonstration of the cooperative weighing concept took place on the IW. Its results were promising: the driver was informed of his vehicle's (simulated) weight within 2 seconds after his car passed the weigh point.

Preface

This thesis is the written outcome of my final project at the University of Twente, conducted externally at LogicaCMG Nederland B.V. in Rijswijk, during the period of September 2007 until February 2008. This project is my last step to attain the degree of Master of Science.

It has been a period where I committed myself to the Dutch public transport 1,5 working day each week, not wanting to give up my 'flat' in Enschede. Happily LogicaCMG prevented an extra 12 hours of weekly traveling by arranging another workplace for me, at the office in Arnhem. Thanks for that! The project has been a good mental and, because of the aforementioned, physical exercise too.

I did my final project in the area of cooperative Intelligent Transportation Systems (ITS) and operated in the graduation program of LogicaCMG, 'Working Tomorrow' (WT) [46]. I have been part of a team active in the field of mobility, within the Public Sector division of LogicaCMG.

I would like to thank my supervisors at the university: Maurice and Nirvana for their eye-opening feedback and support. Furthermore, my thanks go out to my colleagues at LogicaCMG: Marcel and Bart Bartels for enabling me to do this project in the first place, Ilske to keep me on-track and on-schedule, Raynni Jourdain for his facilities in Arnhem, Laurens Blankers for being my technical pal, Bart den Haak for his 'off-the-shelf piece of simulation software', Friso Westerhuis for providing his encouraging viewpoints and of course my WT-colleagues at Rijswijk and Arnhem for creating a sociable environment to work in. Last, but certainly not least: Lisette, thank you for all your love!

Rijswijk, February 12, 2008,

Bobby Nijssen

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

This section gives background information on Intelligent Transportation Systems (ITS), cooperative systems, the Cooperative Vehicle-Infrastructure Systems (CVIS) project and the role of LogicaCMG in this project.

ITS is the general term of applying information and communication technology to transport infrastructures and vehicles in order to improve road safety and efficiency. One of the challenges in the automotive industry and ITS is the use of **cooperative systems**. The CVIS project [21] is part of this.

1.1.1 Cooperative Intelligent Transportation Systems

The appearance of cooperative systems, together with the necessary communication technologies enable new kind of ITS concepts and applications. De Michelis et al. [50] define cooperative *information* systems as the following:

> *An information system is cooperative if it shares goals with other agents in its environment, such as other information systems, human agents and the organization itself, and contributes positively towards the fulfillment of these common goals.* [50]

An information system can share goals with its environment if it is able to exchange information, e.g. by use of a (wireless) network for communication. Michelis' definition can be put into the context of cooperative ITS:

> *Cooperative Intelligent Transportation Systems (ITS) are about vehicles and infrastructure sharing information to contribute positively towards their common goal of increasing road safety and efficiency.*

Cooperative ITS are based on (*see* Figure 1.1 on page 2):
- vehicle-to-vehicle (V2V),
- vehicle-to-infrastructure (V2I),
- infrastructure-to-vehicle (I2V)
- and infrastructure-to-infrastructure (I2I) communication.

The distinction between V2I and I2V is made because it is fundamental to classify ITS concepts. The author coins the collective name **vehicle infrastructure integration (VII)**[1] for applications that are primarily based on *both* V2I and I2V communication.

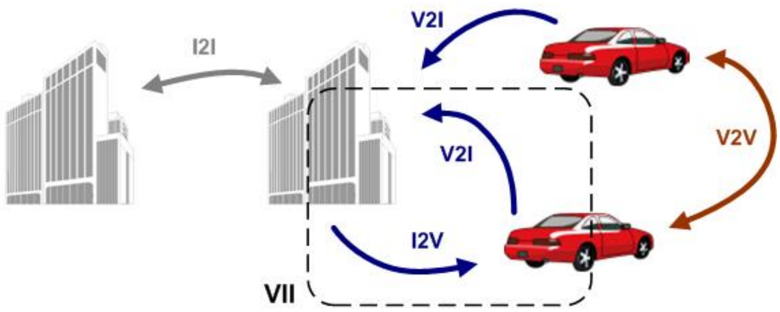

Figure 1.1: Communication-paths in cooperative ITS.

A driver communicates with infrastructure and other road users by use of an on-board unit (OBU). **ITS concepts** provide a unique service to drivers (through the OBU), road operators and other participants in road traffic. The four communication-paths are the means for this service provisioning.

For example; if a driver informs a traffic control centre (TCC) of his destination and a TCC combines this information with current road conditions, it can 'route' the individual vehicle to its destination more efficiently. This concept depends mostly on V2I and I2V communication and is therefore an example of VII.

Another concept would be that a vehicle gets involved in an accident and this vehicle alerts its following vehicles. This application could be entirely based on V2V communication.

1.1.2 CVIS and LogicaCMG's role

LogicaCMG [47] takes part in the CVIS project [21]. CVIS aims at developing an unified technology platform to support cooperative ITS. A consortium of around 60 partners participate in CVIS, among them are: technology companies, vehicle manufacturers, universities, research institutes and national road administrators. The project's goal is to improve road efficiency and safety throughout the European Union.

LogicaCMG leads a CVIS test site between Rotterdam and Antwerp, where a subset of CVIS' applications will be tested. Also the Dutch province of Noord-Brabant cooperates with LogicaCMG with regard to the province's Informatieve Weg (IW) [74] (Dutch for 'Informative Road') project. On this IW, several ITS concepts are already put into practice by use of intelligent road signs. The IW is a test site candidate for the CVIS project as well.

[1]The U.S. DOT runs an ITS initiative with the same name [77].

1.2 Motivation

CVIS' applications have to be tested at different test sites before 2010. LogicaCMG's main objective is to tackle test site deployment problems in advance, before the applications are demonstrated on the test site.

Besides this, all CVIS partners have their own opinion and 'personal' motivation to participate in the CVIS project. This creates an environment where practical results are sometimes overruled with decision-making and intensive negotiations.

Therefore LogicaCMG follows a more practical approach and wants to **demonstrate a *cooperative* ITS concept**[2] to the CVIS stakeholders in advance of the deployment of the CVIS applications. This is to show that CVIS' aims can be put into practice and that their applications can be deployed in an 'off-paper' environment. LogicaCMG is primarily involved with the VII aspect of CVIS, for that reason the project aims at demonstrating such a concept.

The IW is a preferable location to demonstrate this concept-prototype, definitely if this prototype utilizes the existing road-infrastructure there. LogicaCMG wants to gain experience in deploying CVIS-like concepts in real-world environments. A CVIS-like demo in cooperation with the Informative Road will give an image-boost to the province of Noord-Brabant.

1.3 Goal and objectives

LogicaCMG's **main goal** with this Master project is *to gain experience with the development and deployment of a VII application in the context of the CVIS project.* The corresponding objective of this project is to:

> Identify potential development and deployment issues of a vehicle infrastructure integration (VII) application by building and demonstrating a prototype.

In order to reach this main objective, the project is divided into three subobjectives:

1) Give an unbiased overview of Intelligent Transportation Systems (ITS), thereby concentrating on vehicle infrastructure integration (VII) and cooperative systems;

2) Specify and select one VII concept to be demonstrated with the IW;

3) Implement the specified VII concept in the form of a demonstrator and demonstrate it on the IW.

During the process of achieving these objectives, the encountered issues are identified and documented in this thesis.

[2]Example given in section 1.1.1 on page 2.

1.4 Approach

This section explains the approach for each individual objective and lists applicable research questions correspondingly.

1.4.1 Objective 1: give an overview of cooperative ITS

A literature survey is done by studying material on ITS, VII and cooperative systems. An analysis is made of the documentation made available by the CVIS-partners and of other ITS projects. This to position CVIS and other projects in the context of ITS and VII. Furthermore a glossary is created with relevant ITS terms. The following research questions are answered:

a) Where is the ITS market going to and which enabling technologies will have an impact on future ITS solutions?
b) How does the road-user experience ITS?
c) Which role plays the CVIS-project within the context of ITS?
d) On which enabling technologies is the CVIS-project based?
e) Which ITS concepts are out there worldwide?
f) What kind of concepts are applicable to VII?

1.4.2 Objective 2: select a concept for demonstration

Relevant VII concepts are listed and one is selected to be demonstrated on the IW. Its demonstration procedure is specified in detail. The following research questions are answered:

a) What do the physical- and software infrastructure of the IW look like?
b) Which concepts are relevant[3] and could be demonstrated on the IW?
c) What *single* concept is the most suitable to demonstrate?
d) How should this concept be demonstrated on the road[4]?

1.4.3 Objective 3: implement a VII demonstrator

A demonstrator is implemented to demonstrate the concept. In-house and commercial off-the-shelf (COTS) components are used whenever possible. Besides the development effort to build the demonstrator, the following research questions are answered:

a) Which software components make up the demonstrator?
b) How do these components interact and work internally to expose the desired behavior of the demonstrator?
c) In what way should the IW-infrastructure change to be able to demonstrate a concept, integrated with the IW's systems?
d) What are the criteria to evaluate the concept demonstration?
e) Did the demonstration meet its evaluation criteria?

[3]In terms of: feasibility, social relevance and relevance to the CVIS-stakeholders. Added to this, it is important that the concept exploits some enabling technologies of the CVIS project.

[4]Due to project time constraints and lingering organizational consensus it is impossible to integrate the demonstrator with the IW's systems, a simulation setup is used instead.

1.5 Thesis structure

The structure of the remainder of this thesis is presented in Figure 1.2. In the figure, the research questions covered by the chapters are indicated in *italics*. For example, Chapter 2 answers questions ‘a’ through ‘d’ of ‘Objective 1’.

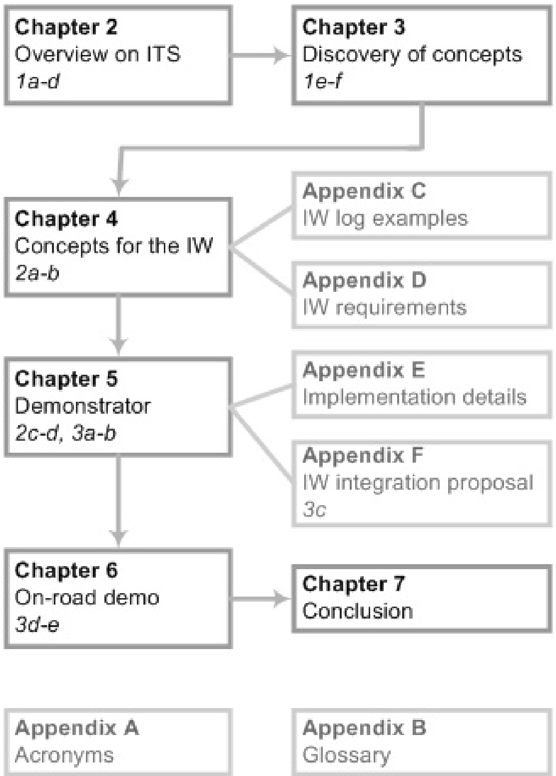

Figure 1.2: Outline of the remainder of this thesis.

Chapter 2 covers ITS in general, the future of ITS and explains the CVIS project in more detail. Chapter 3 gives a survey of existing cooperative ITS projects and provides a taxonomy for ITS concepts. Chapter 4 is the result of an (technical) analysis of the IW. Additionally, concepts that are suitable for demonstration on the IW are selected and described. Chapter 5 provides a technical overview of the demonstrator, which is built on top of a *generic VII framework*. Chapter 6 evaluates the execution of the demonstrator. Chapter 7 concludes the Thesis.

The more technical implementation details are deferred to Appendix C, D, E and F. Besides, a list of acronyms and a glossary of important terms are given in Appendix A and B respectively.

Chapter 2

Overview on ITS

This chapter gives a general overview of Intelligent Transportation Systems (ITS) and describes its future outlook. It also positions the road-user and the Cooperative Vehicle-Infrastructure Systems (CVIS) project in this field.

2.1 Reason for existence

ITS aim to improve the *road efficiency*, *road safety* and *driver convenience*. Systems that 'work together', called *cooperative systems*, play an appearing role in the field of ITS.

Worldwide national governments push the research and development effort in the field of ITS to reach their road safety and efficiency goals.

In 2005, Bishop [4] summarized the government *safety* targets as the following. It seems that their targets have not been changed since.

Europe gives attention to road safety within the Road Safety Action Programme (RSAP). The European Commission (EC) aims for a 50% reduction of road fatalities by 2010 [25]. The Dutch government in particular aims to reduce fatalities by 10% by 2010 and by 40% by 2020.

Australia's target is comparable, a 40% reduction of fatalities per 100.000 population by 2010 [4, 9].

In 2003, **Japan** targetted to half the number of fatalities in 10 years time. This means a reduction of 50% by 2013. The Japanese Ministry of Land, Infrastructure and Transport initiated a major ITS initiative called Smartway [40, 39], which will be described later.

In the **United States** there were 43.443 fatalities in 2005 [76]. The US Department of Transportation's goal is to reduce crashes per 100 million vehicle miles, from the current 1,51 to 1,0 by 2008 [76, 4].

According to the aforementioned targets, the national governments' first priority is that of using ITS to improve road *safety* and reducing the number of accidents. Additionally, traffic congestion costs the European Union (EU) around 100 billion euro each year [22]. To decrease these costs, ITS are implemented to improve road *efficiency*. Also the business sector contributes to ITS for efficiency significantly.

2.2 Future outlook

A while ago, vehicle electronics came with technology such as radio and ABS. The in-vehicle technology of today is maturing and moving towards cooperative in-vehicle systems. Around 30% of a car's total cost was spent to electronics in 2005. This percentage will rise another 5% for 2010 [49]. Control units in the vehicle share sensor data and provide functions more cooperatively. However, these in-car systems still lack cooperation with the 'outside' environment of a vehicle [49].

While these developments in the automotive industry continue to mature, innovations are ongoing in the fields of *information-*, *(wireless) communication-* and *broadcasting*[1] *technology.* Some examples of these innovations are summarized by Chen et al. [8] and are shown in Figure 2.1.

ITS Enabling Technologies	Infrastructure Side	Vehicle Side
Data acquisition	Traffic detectors	AVI
	Weather monitors	Weigh-in-motion
Data processing	Data fusion	GPS
	AID	Digital map
Data communications	Stationary communications	Mobile comm.
	Fiber optics	DSRC
Information distribution	VMS	HAR
	Internet	RDS/TMC
Information utilization	Ramp metering	Route guidance
	UTC	Crash avoidance

AID = automatic incident detection
AVI = automatic vehicle identification
DSRC = dedicated short-range communications
GPS = global positioning system
HAR = highway advisory radio
RDS/TMC = radio data system/traffic message channel
UTC = urban traffic control
VMS = variable message sign

Figure 2.1: ITS enabling technologies, in [8].

In the field of information technology, the Internet enables better and omnipresent distribution of information. In communication technology, fiber optics enable faster data communications. Besides this, ITS specific innovations like automatic vehicle identification (AVI) enable the exploration of ITS concepts at a higher functional level. An example of this is electronic toll collection (ETC), which depends on AVI.

It can be concluded that **the future trend in ITS is** that the automotive industry innovations blend-in with the three fields mentioned earlier [57]. The main enabler for this 'blend' is communication technology that provides ubiquitous ways of communication, e.g. Dedicated Short Range Communication (DSRC) and even more promising Communication Access for Land Mobiles (CALM) technology, which is described in more detail later. Figure 2.2 on page 9 shows innovations from different fields of practice blending into a future 'networked car'.

[1] Like the Global Positioning System (GPS) and the more accurate counterpart Galileo.

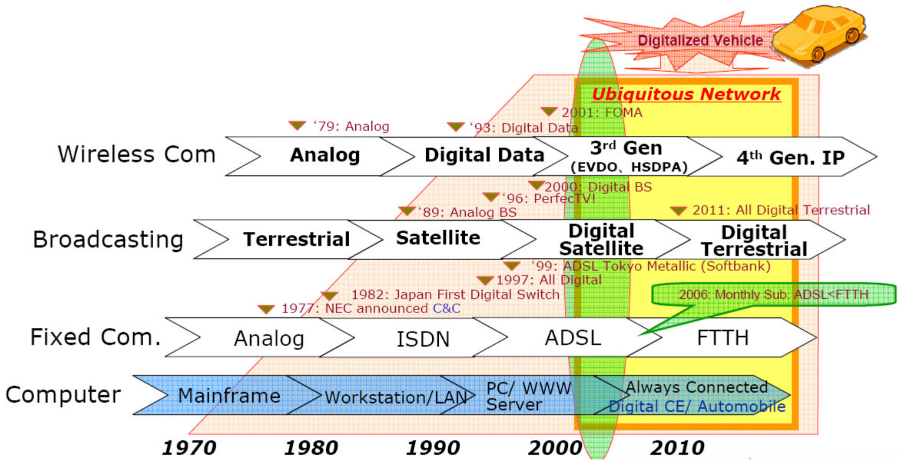

Figure 2.2: Four fields of practice blend into a future 'networked car', in [57].

The mentioned communication technologies make cooperation between vehicles and infrastructures possible. These cooperative systems lead to new kind of services, to be offered to the traveler in a tailored way. E.g. with DSRC there exists a means to send sensor information *from* a vehicle *to* the infrastructure [51]. V2I communication will play a pivot role in future ITS solutions. Concepts where vehicles provide information on road-conditions to the road-side infrastructure, while this infrastructure provides dynamic journey advisories to the vehicles, become possible.

The ultimate future goal is to enable vehicles to make decisions just as (or more) intelligent as human drivers [4]. The infrastructure should fully assist in achieving this goal. More intelligent vehicles and their cooperation with road infrastructure leads to safer roads, more efficient road-traffic and new commercial opportunities.

The future also beholds the possibility that vehicles could drive themselves, without human decision-making involved.

2.3 Road-user experience

This section illustrates a *model* of how drivers execute their task of driving and gives examples how ITS can influence the behavior of the driver in a negative way. It explains an ITS standard of *user services*. This gives a broad view of the kind of services provided to a road-user by ITS. Additionally, it highlights the importance of *driver-vehicle interfacing*.

2.3.1 Model of the driving task

Houtenbos et al. [31] give a model on how the driving task is actually performed by drivers. This model is shown in Figure 2.3 on page 10 and illustrates how drivers perceive their environment and make decisions accordingly.

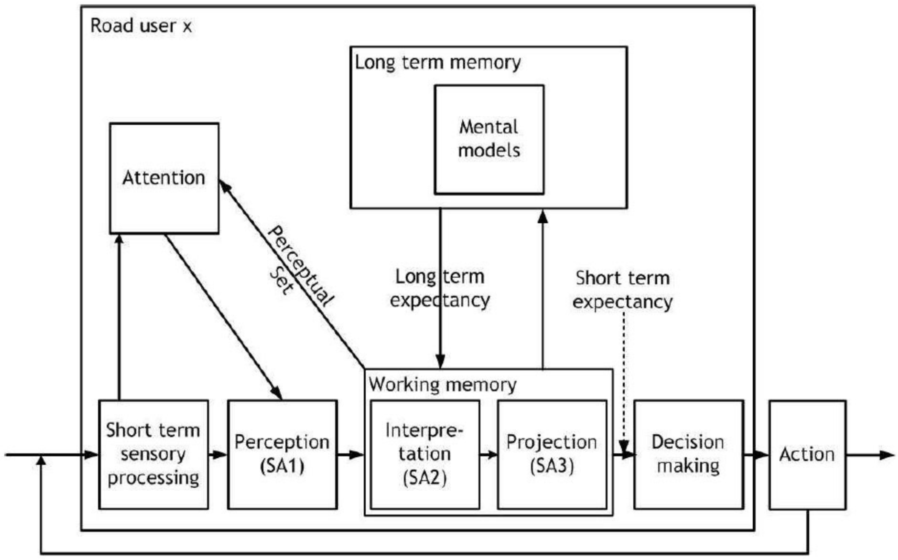

Figure 2.3: Information processing in driving, in [31].

Two kinds of expectancy are distinguished in their model: *long term* and *short term* expectancy. Long term expectancy is based on the driver's experience and education, while short term expectancy has to do with the current expected behavior of the environment (it is projected by the driver).

Drivers *interpret* the environment and *project* what would happen next, e.g. a car will cross the road in the next couple of seconds. This is an example of short term expectancy. A long term expectancy example would be a driver to expect all vehicles on a highway to go in the same direction. Therefore wrong-way drivers are not seen in some occasions.

The advanced driver assistance systems (ADAS) support the driver with information provisioning or even take over part of the driving task. Houtenbos et al. mention that the effects of ADAS change the behavior of a driver. Two examples:

- Take a system that analyzes the situation at intersections and decides on who has the right-of-way. The driver will possibly forget his rules of road traffic after a while.

- Drivers with an adaptive cruise control (ACC)[2] tended to overtake more and keep to the left lane more [29].

The aforementioned phenomenon is analogous to the nature of cooperative systems. De Michelis et al. [50] explain that new information provided to an agent[3] leads to a change of its behavior.

[2] Cruise control with the feature to keep a fixed following distance to a frontal vehicle.
[3] The driver and the in-vehicle system could be seen as a cooperative system.

2.3.2 ITS user services

Countries worldwide gained experience with their ITS programs and reached a consensus on the possible services that ITS could provide to its users. The users of ITS are not limited to road users. More stakeholders use ITS, like the *road network operators*, *transport services providers* and *fleet owners*. [8]

The International Organization for Standardization (ISO) provides with ISO-1997 a taxonomy of 32 ITS user services, *see* Figure 2.4.

Traffic management	1. Transportation planning support
	2. Traffic control
	3. Incident management
	4. Demand management
	5. Policing/enforcing traffic regulations
	6. Infrastructure maintenance management
Traveler information	7. Pretrip information
	8. On-trip driver information
	9. On-trip public transport information
	10. Personal information services
	11. Route guidance and navigation
Vehicle systems	12. Vision enhancement
	13. Automated vehicle operation
	14. Longitudinal collision avoidance
	15. Lateral collision avoidance
	16. Safety readiness
	17. Precrash restraint deployment
Commercial vehicle	18. Commercial vehicle preclearance
	19. Commercial vehicle administrative processes
	20. Automated roadside safety inspection
	21. Commercial vehicle on-board safety monitoring
	22. Commercial vehicle fleet management
Public transport	23. Public transport management
	24. Demand-responsive transport management
	25. Shared transport management
Emergency management	26. Emergency notification and personal security
	27. Emergency vehicle management
	28. Hazardous materials and incident notification
Electronic payment	29. Electronic financial transactions
Safety	30. Public travel security
	31. Safety enhancement for vulnerable road users
	32. Intelligent junctions

Figure 2.4: ITS user services, based on ISO 1997, in [8].

Services 7-17 are of *direct* benefit to a regular vehicle driver. Roughly, this subset can be split into services providing efficiency (7-11) and safety (12-17). The taxonomy can be used to decide what kind of ITS-concepts are applicable to a certain end-user environment in a more structured way.

2.3.3 Driver-vehicle interfacing

Information and services can be provided to a driver by an on-board unit (OBU) or a mobile phone. There are several ways to communicate with the driver [4]:

- Audible, like the voice of a navigation system;
- Visual, for example by a dashboard icon, a head-up display (HUD) or navigation system screen;
- or Haptic, e.g. by resistance in pedals or steering wheel, tightening of the seatbelt or seat vibration.

Key factors to successful driver-vehicle interfacing include: perceptability, comprehensibility, learnability, trust, misuse potential (i.e. drivers choosing higher risks because they trust the 'safety net' of the system) and error robustness. [4]

2.4 Cooperative Vehicle-Infrastructure Systems (CVIS)

The European Road Transport Telematics Implementation Coordination Organization (ERTICO)[4] [22] is a partnership of around 100 public and private partners involved in ITS.

ERTICO coordinates, among others, the European CVIS project [21], which aims for a technology platform to support *cooperative* ITS. It develops (reference) applications that focus on the *efficiency* improvement of road traffic [21, 75]. It is a partnership of 60 companies, each contributing with their specialties. Currently, about 300 people are employed in total.

2.4.1 Project structure

The size of the project and its goals demand a good division of tasks. CVIS is broken down into four blocks, as shown in Figure 2.5 on page 13.

The **top** block represents the *coordination-oriented* sub-projects. Within it, the Core Architecture Group (CAG) 'acts both as a technical project in producing the overall architecture common to all technology and application sub-projects, and also as an IP[5] coordination mechanism for integrating and synchronising the different activities into a coherent work programme' [21].

The **left** and **right** blocks represent the *technology-* and *application-oriented* sub-projects, respectively. The technology block supports the application block with transparent communication facilities, application management[6] and positioning/mapping services. In the application block, different ITS concepts are implemented by reference applications.

[4]Many countries have their own ITS body or association, like: ITS Netherlands, ITS Sweden, ITS Spain etcetera. Each of them initiating 'local' ITS-projects. In order to share knowledge better and improve communication, ERTICO hosts a Network of National ITS Associations [20] since 2004, where the local ITS-bodies are grouped together.

[5]IP stands for integrated project.

[6]Based on OSGi, a service-oriented environment for the Java platform [61].

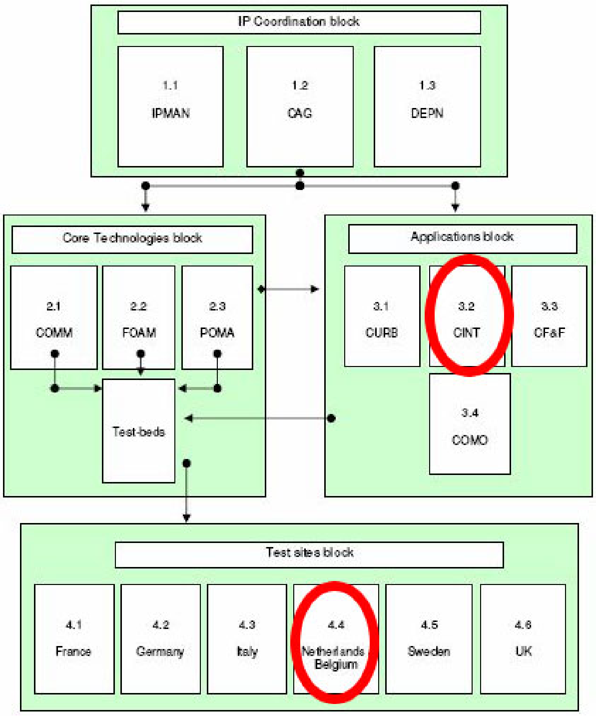

Figure 2.5: CVIS project structure, in [21].

The **bottom** block represents the different sites where the CVIS-applications will be tested.

The ellipses in Figure 2.5 indicate LogicaCMG's focus within the CVIS project. In the application block, LogicaCMG concentrates on Cooperative Inter-urban Applications (CINT). Marcel Konijn, one of the supervisors of this Thesis, leads the test-site between Rotterdam and Antwerp. CINT and other CVIS-applications will be tested there.

2.4.2 Cooperative Inter-urban Applications (CINT)

CINT 'will develop and validate cooperative services to improve the efficiency, safety and environmental friendliness of traffic on the inter-urban road network and offer a safe and comfortable journey to drivers and passengers' [21].

It employs 18 people from 15 partners. The CINT applications should be implemented early 2008 and will be tested at the end of 2008, according to the CINT project leader.

CINT provides two groups of applications [5, 80]:

1. **Enhanced Driver Awareness (EDA)** focuses on road safety and informs the driver about the traffic situation in front of his current position. Hereby information about e.g. the weather situation, speed regulations, a nearby accident and wrong-way drivers play an important role.

2. **Cooperative Travelers Assistance (CTA)** focuses on the assistance of the driver and improving the efficiency of the traffic. It assists the driver by use of two services: *pre-trip* and *on-trip* planning. The traveler plans his trip before he leaves using the pre-trip planning service. All journey information is fed to a TCC. Based on this and other knowledge (e.g. weather conditions, traffic jams, road works etc.) the TCC can provide the on-trip planning service: it assists drivers *during their journey* to navigate to their destinations most efficiently.

The model of Houtenbos et al. [31], explained in Section 2.3, can be put into the context of CINT. EDA and CTA have a closer relation to, respectively, the 'short term-' and 'long-term expectancy' of a driver.

Figure 2.6 illustrates the differences of the two with respect to quality of service (QoS) and allowed processing time. According to the figure, EDA should deliver a higher QoS in a more real-time manner than the CTA applications. EDA contributes to safety, so the driver should be alerted *in* time, e.g. if a wrong-way driver is approaching.

On the other hand, CTA can respond later in providing journey advices and is permitted a lower QoS.

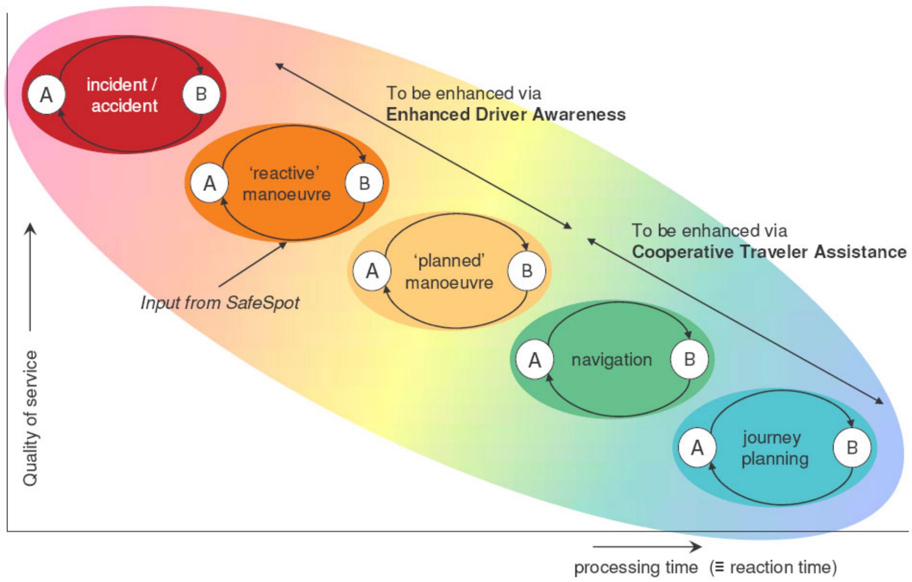

Figure 2.6: EDA and CTA regarding QoS and allowed processing time, in [80].

2.4.3 CALM as enabling technology

CVIS, like any other ITS initiative, is enabled by innovations from several fields of technology. An overview of those technologies have been presented in Figure 2.1 on page 8.

Additionally, the Communication Access for Land Mobiles (CALM) standard can be seen as the main enabler for CVIS. CALM lays an ubiquitous communications foundation for the CVIS project and is visualized in Figure 2.7. It comprises all seven layers of the ISO's OSI reference model [44, 7]. The standard is based on IPv6 and is currently being defined by the ISO TC 204 Working Group 16, who define CALM as the following:

> *The scope of CALM is to provide a Standardized set of air interface protocols and parameters for medium and long range, high speed ITS communication using one or more of several media, with multipoint and networking protocols within each media, and upper layer protocols to enable transfer between media.* [37]

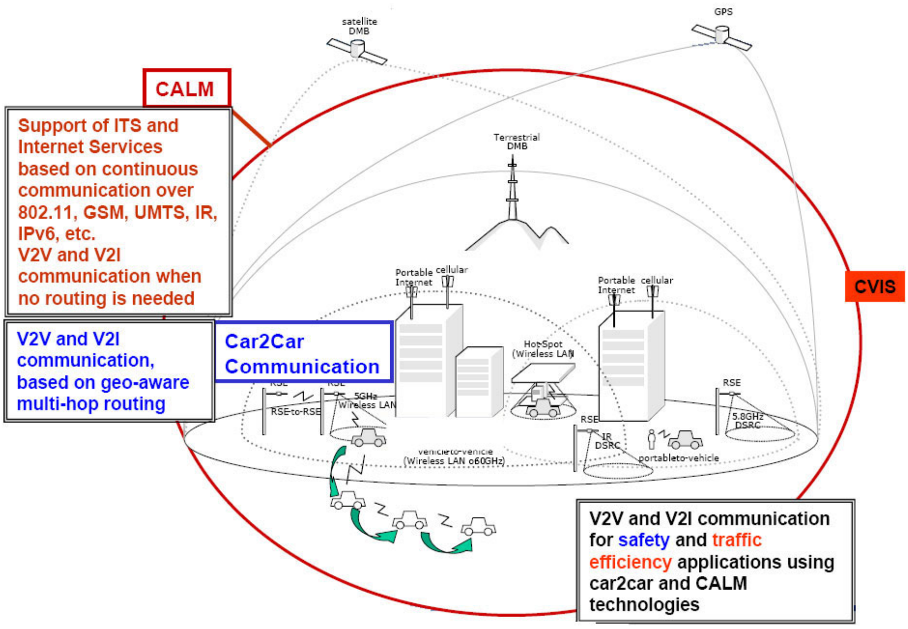

Figure 2.7: CALM will facilitate communication for CVIS' applications.

The CALM standard provides:

- **Universal communications** [21]. Meaning a ubiquitous way of communication between the operator, service provider and vehicle. Devices in each individual vehicle can be reached at the current location of the vehicle by their IPv6 address using any underlying communication medium connected to the traffic infrastructure or other vehicles.

- **Agnostic use of media**. In order to provide a continuous service to the driver and to the TCC, the seamless use of communication media is needed. This means that an TCC's IPv6 connection with a vehicle remains active when the vehicle is connected with the infrastructure by DSRC and is suddenly switched to Universal Mobile Telecommunications System (UMTS) for example.

2.4.4 Position in the ITS context

As suggested by its name, CVIS focuses on *cooperative* ITS. In comparison with other EC-funded projects [23, 66, 10] CVIS emphasizes on road efficiency and less on safety, *see* Figure 2.8.

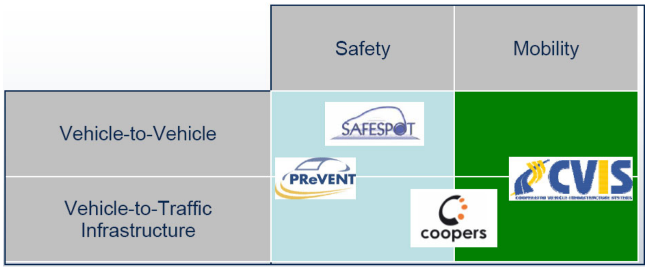

Figure 2.8: CVIS compared to other EC-funded projects, in [43].

Within the EU, the **PReVENT** project 'develops and demonstrates preventive safety applications' [23]. **Safespot**'s objective is 'to understand how intelligent vehicles and intelligent roads can cooperate to produce a breakthrough for road safety' [66]. **Co-operative Systems for Intelligent Road Safety (COOPERS)** 'focuses on the development of innovative telematics applications on the road infrastructure with the long term goal of a Co-operative Traffic Management between vehicle and infrastructure' [10].

CVIS' envisioned applications are of a truly cooperative nature [70] compared to some projects outside the EU [51, 39]. Most of the time, the concepts of those projects do not involve vehicles and infrastructure working together tightly to achieve a shared goal in the sense that *both* V2I and I2V are being exploited by the concept.

Chapter 3

Discovery of concepts

This chapter presents a short survey of ITS projects worldwide and lists their underlaying cooperative ITS concepts. Various additional concepts are listed. A taxonomy is introduced, which filters these concepts and identifies the vehicle infrastructure integration (VII) concepts that are relevant for demonstration.

3.1 Introduction

A wide variety of ITS concepts could be demonstrated on the IW. For this project, it is important that the demonstrated concept is cooperative. To be more specific, it should be cooperative in the sense that the IW-infrastructure works together with one or more vehicles. Therefore it is important to distinguish between *cooperative* and the more specific *VII* concepts, of which a definition is given before.

Bishop [4] lists a few examples of cooperative ITS applications, *see* Figure 3.1 on page 18. He calls them Cooperative Vehicle Highway Systems (CVHS). This term suggests that cooperative vehicle systems are implemented on highways only. This is not true as cooperative ITS can be useful in urban areas as well, e.g. on an unpaved farmers road. Therefore the name 'cooperative ITS' is a better choice.

In Figure 3.1 on page 18 cooperative ITS concepts are split into three categories[1]. The VII-concepts reside in the category indicated by the rectangle. The concepts of the other two categories aim more at improving safety and involve less tight cooperation between vehicles and infrastructure.

3.2 Project survey

This section summarizes a few ITS projects and initiatives worldwide, together with their underlaying *cooperative* ITS concepts. Several ITS books [4, 8] give a far more detailed and broader overview to worldwide ITS projects, their concepts and lessons learned.

ITS projects are usually initiated by *national governments* themselves and by *partnerships*, which are government funded most of the time. They are

[1]This figure differs from Figure 2.4 on page 11 in the sense it shows a classification based on *communication paths*, instead of a classification based on end-user environments.

Vehicle to/from infrastructure (for traveler information, convenience, productivity)	*Infrastructure -to-vehicle safety applications*	*Vehicle-to-vehicle safety applications*
Electronic toll collection	Curve speed warning	Emergency braking of forward vehicle
Wireless diagnostics	Optimal speed advisory	Precrash warning
Software "flashing," (i.e., updating vehicle software wirelessly)	Highway work zone warning	Driver advisories when making turns across traffic
Commercial driver log information	Highway/rail crossing collision avoidance	Lane change warning
Vehicles acting as data probes for traffic and road information	Traffic signal violation warning	Approaching emergency vehicle advisory
Real-time traffic updates	Low bridge warning	Stop sign movement assistance
Real-time map updates	Road condition warning	Road condition warning
Enhanced route planning and guidance		Cooperative adaptive cruise control
Drive-through payment (gas, parking, fast food)		Vehicle platooning
Wireless transfer of digital entertainment (games, music, video)		
Data exchange for border clearance		

VII

Figure 3.1: A sampling of cooperative ITS, in [4].

complicated in the sense that they cross multiple knowledge areas, have a large impact on the road-infrastructure, are costly and have a longterm return on investment (ROI).

3.2.1 INVENT in Germany

INVENT [36] stands for *intelligenter Verkehr und nutzergerechte Technik*, which is German for 'Intelligent traffic and user-oriented technology'. The project has been completed in 2005 and was a four-year research initiative supported by 24 companies, all aiming to improve traffic flow and safety.

It was divided into three sub-projects, which implemented vehicle and system demonstrators [35]:

- **Driver assistance/active safety**. Makes effective use of vehicle sensors to prevent accidents and improve traffic flow. Examples are: crossing bicycle detection, lane changing hazard warning and automatic stop-and-go of the vehicle. These concepts are not cooperating with the infrastructure, they are 'only' an intelligent way of using *in-vehicle* sensor data.

- **Traffic management 2010 (NIV)** [34]. Focuses entirely on flow improvement and is based on two cooperative concepts: *traffic performance assistance* and *network traffic equalizer*. The first uses V2V communication to minimize the length of a traffic jam and adapting the speed of vehicles in such a way to return faster to an uncongested flow. The second is an example of VII and provides individual route guidance to drivers based on the extended floating car data (XFCD)[2] method, *see* Figure 3.2 on page 20.

- **Traffic management in transports and logistics** [33]. Strongly based on VII as well. It provides effective utilization of transport routes and a monitoring service to track goods more precisely.

3.2.2 SRA in Sweden

The Swedish Road Administration (SRA) conducted an intelligent speed adaptation (ISA) trial during the period 1999-2002 [2, 4]. ISA aims to improve road safety by helping the driver not to exceed the speed limit.

Three ISA models have been tested:

- Warning ISA, where the driver received a warning signal when the speed limit was exceeded.

- Informative ISA, a warning and in addition the driver is informed with the existing speed limit on the road.

- Active accelerator, generates counter pressure in the accelerator pedal to inform the driver when the speed limit is exceeded.

[2] An extended version of the floating car data (FCD) method. Both terms are explained in the Glossary.

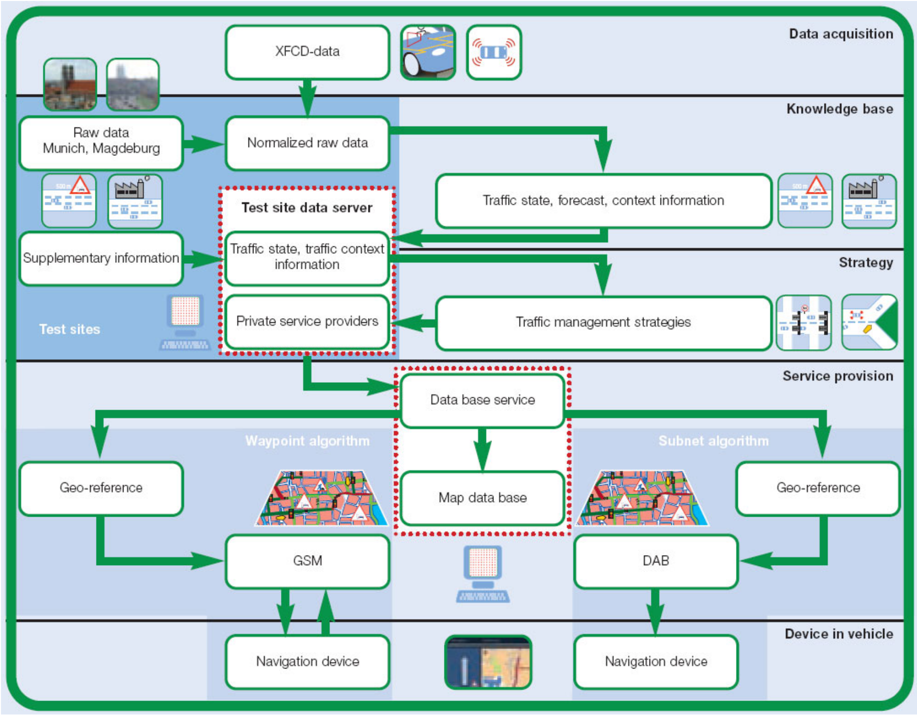

Figure 3.2: INVENT's provisioning of dynamic route guidance, in [36].

The models have been implemented in different ways at several trial sites. For positioning, GPS and static transponders were used. The infrastructure communicated with the vehicle by use of GSM and by transmitters on lamp-posts, these transmitters were not connected to a central system.

At the Borlange site, the vehicles were connected to a server by a GSM connection. This can be seen as a form of VII. The vehicle-clients were able to receive map updates and send log updates. However, the real-time 'decision making' was still executed *locally* inside the vehicle.

Another initiative of the SRA is the website TRAFIKEN.nu [78]. Here, road users can check the road conditions before their journey. It shows actual and planned road works, traffic jams and average speed of the traffic flow. Figure 3.3 on page 21 shows an interactive map of Stockholm.

3.2.3 PATH in California, USA

The Partners for Advanced Transit and Highways (PATH) have been active since 1988 [51]. They started implementing in-car control systems in 1991. After further research and development they were able to demonstrate an *automated platooning* concept in 1997. Here, cars can follow each-other in a controlled way with a fixed distance between them, *see* Figure 3.4 on page 21. This could be seen as an early cooperative ITS. It involves V2V communication only and is therefore not an example of VII.

The PATH moved to the development of systems that are more coopera-

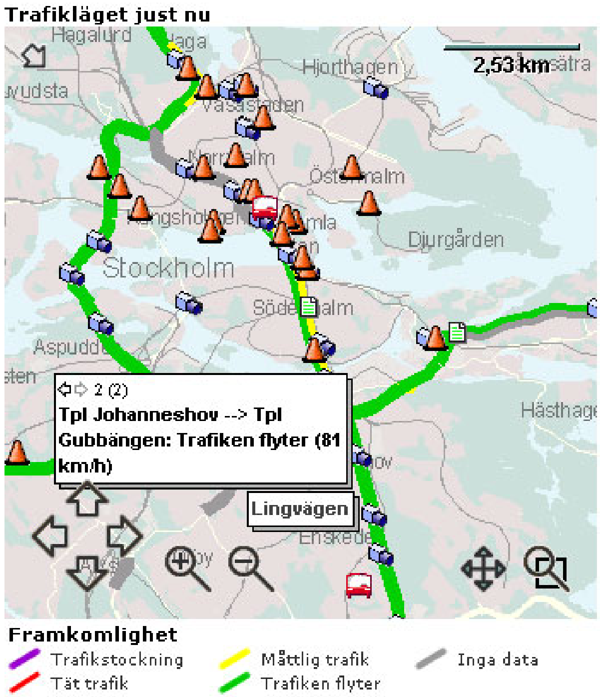

Figure 3.3: An interactive traffic map of Stockholm [78].

Figure 3.4: An automated platoon on a highway.

tive, like forward collision warning and intersection collision avoidance. These two concepts involve V2V and I2V communication. Other concepts of PATH include:

- **Road condition monitoring**. A probe-vehicle senses road-condition and classifies it as a 'normal', 'slippery', 'very slippery' or 'rough' surface. This vehicle alerts following vehicles via a road-side unit (RSU).
- **Traffic probes**. Vehicles send their GPS location information to a RSU.
- **Vehicles providing weather data**[3]. The OBU sends current temperature, windscreen wiper- and air conditioner status to a RSU so it knows the weather conditions on the road.
- **In-vehicle signing**. Road-side sign information will be send to the OBU for display to the driver.

The communication paths in these four concepts are most of the time one-way. For this project, the PATH's concepts are not good examples of vehicle and infrastructure 'working together' in a VII setting. One form of cooperation between RSUs and vehicles is being implemented by the first aforementioned concept. Strictly, this is VII. However, the RSU is used as a 'gateway' to send information from a vehicle to other vehicles.

PATH's other ideas can be useful in the sense that they can be part of a 'bigger' concept, exploiting I2V *in combination with* V2I communication. For example, vehicles provide weather data to a TCC, to be used to advise other travelers.

3.2.4 Smartway in Japan

In May 2007, the Japan Ministry of Land, Infrastructure and Transport [38] started testing Smartway [40] on public roads. Smartway is in an advanced stadium with regard to demonstrating cooperative applications. The execution of Smartway's trial operation is planned halfway 2007 [39].

The project uses two kinds of OBUs: an 'independent' one, which provides information by audio only, and one that is integrated with a car navigation system, which provides information by both audio and visuals. The project mainly uses DSRC for V2I and I2V communication.

Smartway comprises the following cooperative application concepts [39], which are a mixture of VII, I2V and V2I examples:

- **Providing information on obstacles and conditions ahead**. RSUs sense obstacles on the road and inform approaching vehicles. OBUs display a warning by use of audio and visual. *See* Figure 3.5 on page 23.
- **Merging assistance**. A RSU detects approaching vehicles at a merge point[4]. Just before the merge point, OBUs alert drivers for the presence of merging vehicles.

[3] This is rather strange because RSUs could sense weather conditions more easily than a vehicle could do.

[4] Two roads pointing into the same direction coming together. They 'merge'.

- **Internet provisioning**. At parking areas and on road sections, a wireless IP data link is there to provide an Internet connection. DSRC and wireless LAN is used for this.

- **Smart parking facilities**. The driver can use a parking area by use of ETC. *See* Figure 3.6 on page 24.

- **Location support**. A RSU broadcasts location information to be picked up by OBUs at places where car navigation systems fail to determine the vehicle's location.

- **In-vehicle signing**. Information on road signs will be available in-car.

The corresponding systems of the aforementioned concepts are cooperative in the sense that RSUs communicate with OBUs. However, for most of these concepts this communication is one-way, except for the *Internet provisioning* and the *ETC* example.

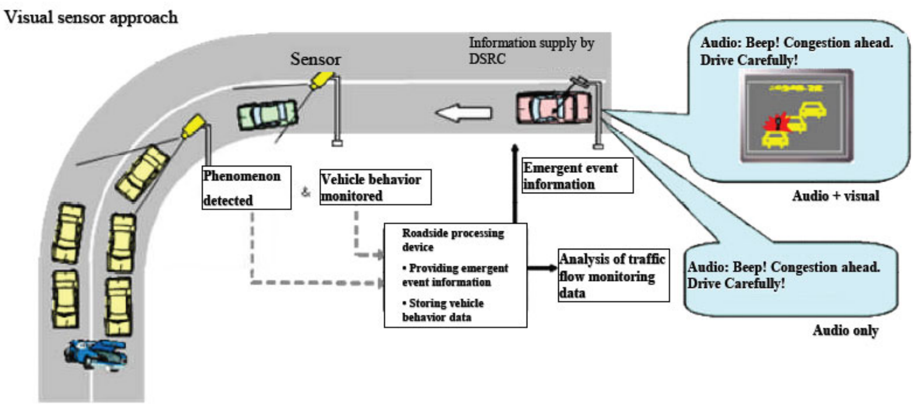

Figure 3.5: Smartway provides information on obstacles, in [39].

3.3 Additional concepts

During this project several cooperative ITS concepts came across the mind of the author. It might be the case that these concepts are already existent. These additional concepts are:

- **Incident management.** Road operators employ road workers[5]. Most of them drive along(side) the roads with their service vehicles. It would be beneficial to let these service vehicles communicate with road-side systems. For example, the traffic lights can report to the road worker that they are out-of-order, or the road worker can remotely 'write' a message on an electronic display in case of an accident.

[5]In Dutch they are called 'kantonniers'.

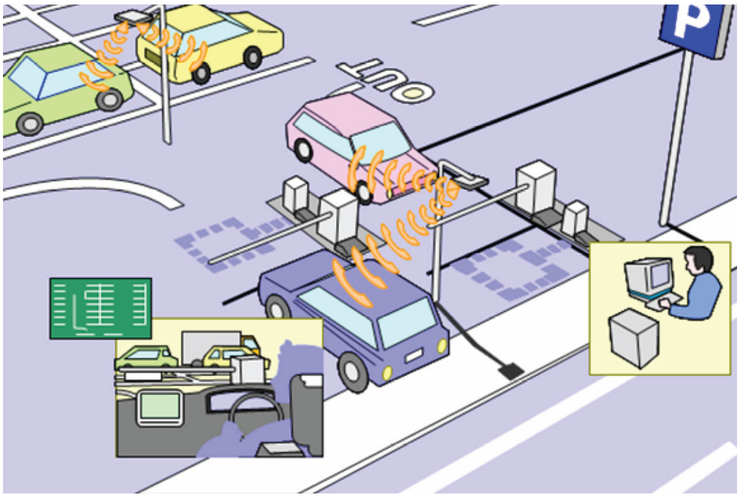

Figure 3.6: Smartway's smart parking facilities, an example of ETC, in [39].

- **Driver-interacted platooning.** Platooning (*see* Figure 3.4 on page 21) can decrease congestion because the vehicles can move closer each other. An OBU that provides an interactive service to the driver can take this platooning concept to a next level. For example, the driver uses an OBU to select nearby vehicles to 'team up' and form a platoon, e.g. when the driver enters a highway.

- **Jackknifing detection.** Caravans (and other equipment) that is towed by a vehicle can 'jackknife', i.e. the caravan starts to move sideways, becomes uncontrollable and folds. The towing hook could be fitted with a sensor that is connected to an OBU. The vehicle can inform the road-side infrastructure when jackknifing occurs to alert upstream vehicles or emergency services for example.

- **Logistics vehicle tracking.** Vehicles or trucks that drive for a logistics company can be tracked by their operational management using GPS and V2I communication. E.g. warnings can be sent to the management when the vehicle is off-route.

- **Busses working together with bus-stops.** (Public transport) busses can communicate with bus-stops, acting as infrastructure. For example, more accurate waiting times can be displayed at bus-stops when the current location of busses is known. Additionally, the bus traveler can let known his presence to the bus driver in advance, i.e. by pressing a button at the bus stop and the display of a warning-sign in the bus to the driver[6]. This reduces the occurance that the driver not sees a person waiting. Furthermore, the driver can be sure no passenger is waiting for him at a certain bus stop, so he could take a shorter route.

[6]The person could even select a *line* at a multi-line bus stop. This way a driver does not have to stop when the person is waiting for another line.

- **Closeby motorcycle alert.** Motorcycles are less notable than cars. To provide extra safety to the motorcycle (and to the car), a vehicle driver can be informed about a closeby (or approaching) motorcyclist.

- **Cooperative speed advice.** The road-side infrastructure informs vehicles to slow down or speed up (based on the current speed of the vehicle) to be able to catch the next green light.

- **Cooperative weighing.** A weighing-sensor built into the road can measure a vehicle's weight. This weight-information can be provided to the driver and to other stakeholders, e.g. a logistics company, government or trading party. A match between a measurement and a vehicle can be made according to the vehicle's location, which is provided to the infrastructure.

  The driver and the mentioned stakeholders could be interested in this information because heavy trucks cause road wear and could be unsafe. Furthermore, most countries apply rules for (maximum) axle weights and trucks with a high axle weight are taxed.

3.4 Taxonomy

The aforementioned ITS concepts can be used as inspiration for the selection of a concept to be demonstrated on the IW. The concepts are classified according to their *communication-paths*[7] to ease this selection process. This classification is presented in Table 3.1.

The CVIS sub-projects Enhanced Driver Awareness (EDA) and Enhanced Driver Awareness (EDA) (*see* Section 2.4.2) are listed in the taxonomy for completeness. However, they are not part of the selection process and are therefore not numbered in Table 3.1.

Initiator	#	**ITS concept**	Communication-path: V2V	V2I	I2V	I2I	**VII**
CVIS		Enhanced Driver Awareness (EDA)	•	•	•	-	•
		Cooperative Travelers Assistance (CTA)	-	•	•	-	•
INVENT	1	Driver assistance/active safety	-	-	-	-	-
	2	NIV: traffic perf. assistance	•	-	-	-	-
	3	NIV: network traffic equalizer	-	•	•	•	•
	4	Mgmt. in transports and logistics	-	•	•	•	•
SRA	5	ISA at Borlange site	-	•	•	•	•
	6	TRAFIKEN.nu	-	-	-	•	-
PATH	7	Road condition monitoring	○	•	•	-	○
	8	Traffic probes	-	•	-	-	-
	9	Vehicles providing weather data	-	•	-	-	-
	10	In-vehicle signing	-	-	•	-	-
Smartway	11	Providing information on obstacles	-	•	•	-	•
	12	Merging assistance	-	•	•	-	•
	13	Internet provisioning	-	•	•	-	•
	14	Smart parking facilities	-	•	•	•	•
	15	Location support	-	-	•	-	-
	16	In-vehicle signing	-	-	•	-	-
Additional	17	Incident management	-	•	•	-	•
	18	Driver-interacted platooning	•	-	-	-	-
	19	Jackknifing detection	○	•	•	•	•
	20	Logistics vehicle tracking	-	•	•	-	•
	21	Busses working together with bus-stops	-	•	•	-	•
	22	Closeby motorcycle alert	•	-	-	-	-
	23	Cooperative speed advice	-	•	•	-	•
	24	Cooperative weighing	-	•	•	•	•

Table 3.1: Classifying ITS concepts according to communication-paths.

A bullet (•) indicates that the concept is based on that specific communication path and shows if the concept is an example of VII (last column of Table 3.1). A circle (○) means that the specific communication path is used *indirectly*. For example; vehicles in the PATH/Road condition monitoring concept communicate with other vehicles via the infrastructure, thus it uses V2V communication in an indirect way. Strictly this concept is an example of VII, however the infrastructure is 'dumb' in the sense it acts as an information gateway only (therefore this is indicated by ○).

[7] *See* Figure 1.1 on page 2.

Chapter 4

Concepts for the IW

This chapter describes the process of selecting vehicle infrastructure integration (VII) concepts that are relevant for demonstration with the Informatieve Weg (IW) and are useful in the Cooperative Vehicle-Infrastructure Systems (CVIS) project context.

4.1 Methodology

An analysis of the *existing* IW infrastructure is done, preceding the selection of a suitable cooperative concept. Several sources [74, 17, 18, 19, 27] are used as input to the analysis. A *domain model* [81] is defined to be able to reason unambiguously about the IW domain. This creates a solid foundation for further functional and technical analysis and it structures discussions with the parties involved.

Project goals are related to high-level *properties* that should be held by the concept. The 'repository' of concepts, described by Chapter 3, is sought to find suitable concepts that comply with these concept properties. The analysis of the IW helps *1)* to decide what existing pieces of functionality could be reused by a VII concept and *2)* to assess the feasibility and relevance of the concept to be demonstrated.

4.2 Informatieve Weg (IW)

Informatieve Weg (IW) is Dutch for 'Informative Road', an ITS project initiated by the Dutch province of Noord-Brabant. It went live at the end of 2005. Several electronic signs at the IW provide real-time information to drivers. This information is referred to as 'the *services* provided by the IW' hereafter. E.g. when bicycles are crossing the road, the approaching vehicles are alerted by signs lighting up [74].

The functionality of the IW is spread out on three national roads [17]:

- N629 from Dongen to Oosterhout,
- N632 from Dongen to Tilburg,
- N282 from Breda to Tilburg.

An overview map of the complete IW area is shown in Figure 4.1. The locations of a *signaalgever (SG)*[1] and *verkeersregelinstallatie (VRI)*[2] are indicated on the map by corresponding icons. A SG is a multi-functional electronic sign that can display full-color textual information and icons.

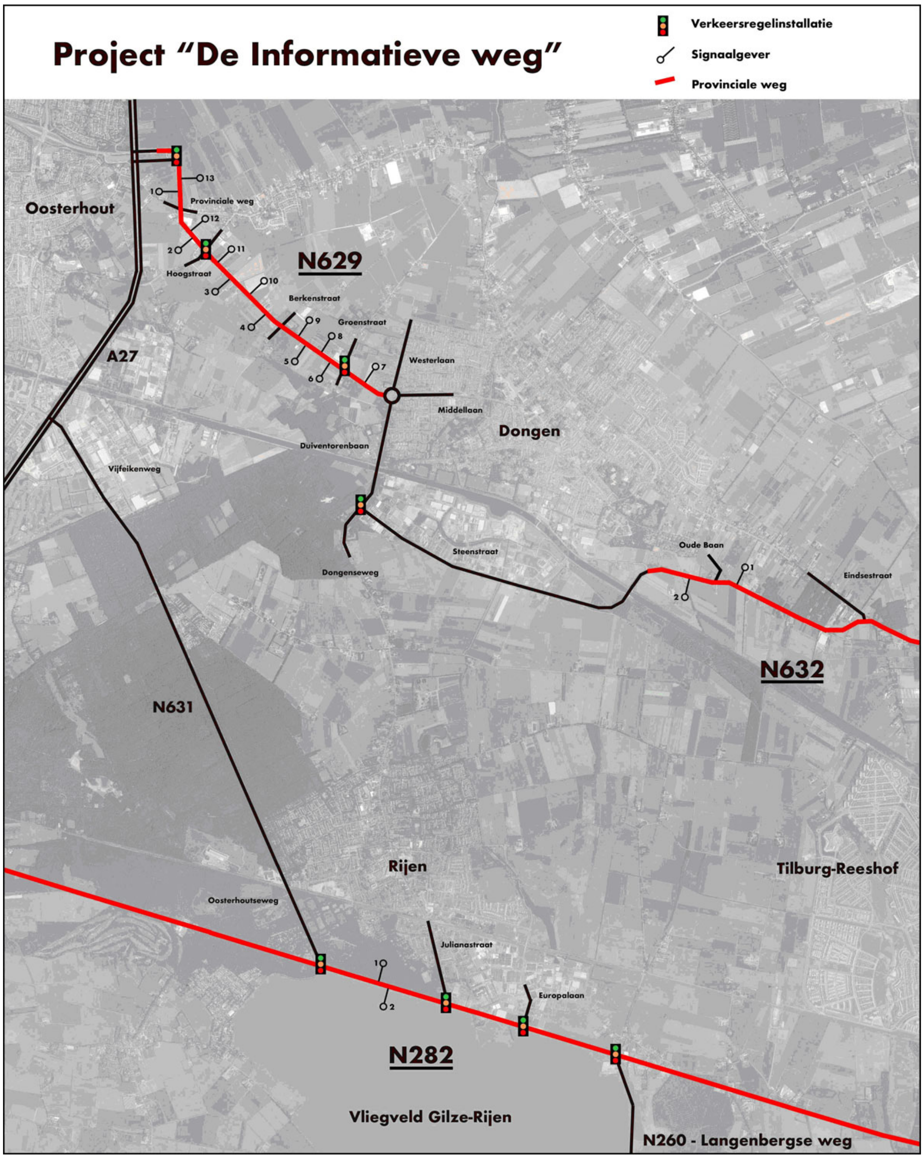

Figure 4.1: Overview map of the Informatieve Weg (IW), in [17].

[1]Dutch for 'signal provider'.
[2]Dutch for 'traffic lights controller'.

4.2.1 Domain model

The entity-relationship model (ERM) of Figure 4.2 defines an unambiguous vocabulary for the IW domain [81]. It identifies entities and their relationships and is used to come to a common ground of understanding together with the province of Noord-Brabant, LogicaCMG and other CVIS partners. The model is distilled from several documents of the province [17, 18, 19, 27].

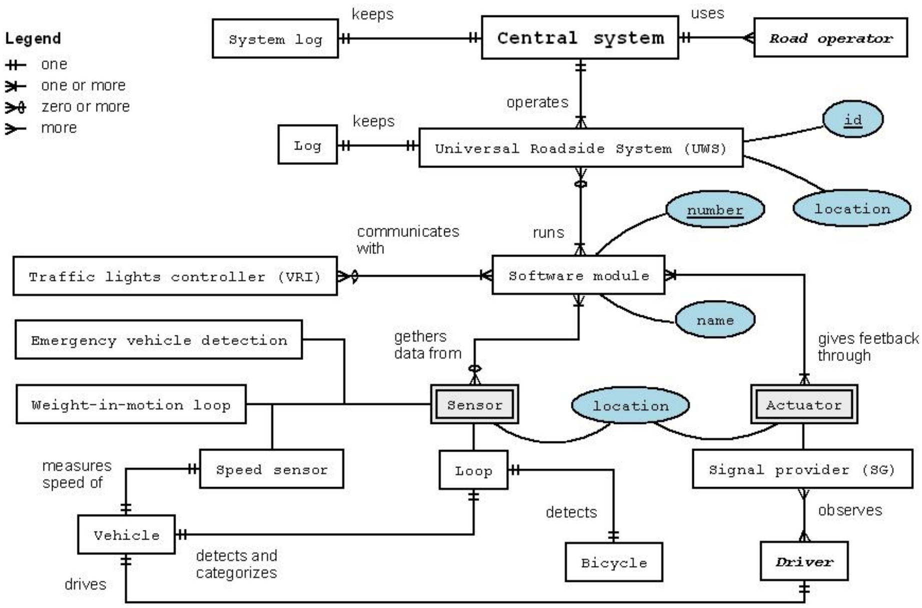

Figure 4.2: ERM describing the domain of the IW.

Conclusions can be drawn from the model, like 'the central system[3] keeps a different log than a UWS'.

A box represents a physical (like a loop to detect vehicles or a driver), conceptual (a system log) or an abstract entity (like a sensor, shown in gray). The lines between the boxes represent their relation to each-other. The *relative* cardinality of the relation is shown by four icons[4], a legend explains their meaning. For example, one central system operates *one or more* universeel wegkant systeem (UWS). Attributes are shown as ellipses that are connected to the entity they belong to.

It is important to note that the terms SG and UWS are used interchangeably at some places in the IW-documentation [17, 18, 19, 27], while they are indeed different entities in the real world.

4.2.2 Existing functionality

Several software modules provide services to drivers[5] and road operators. The individual modules are explained in more detail later. A Unified Modeling Language (UML) use case diagram shows the functionality of the IW. The

[3]Words referring to the model are underlined in this section.
[4]The 'crows foot' notation is used for conciseness.
[5]Also specific services for truck-drivers are provided.

number(s) of the software module(s) that implement the use case are shown between parentheses. The diagram is shown in Figure 4.3.

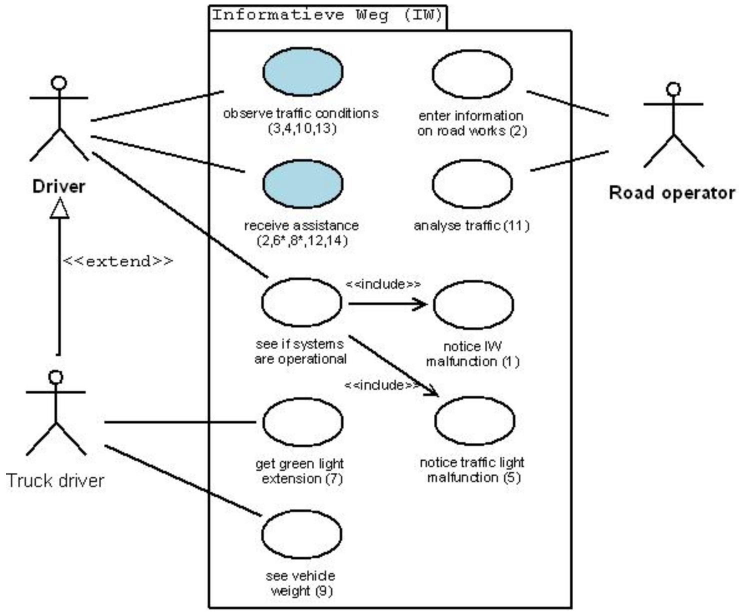

Figure 4.3: Generalized UML use case diagram of the IW.

Two important use cases of the driver are (shaded in the figure): *observe traffic conditions* and *receive assistance*. In fact, they are a generalization of several use cases. This generalization clarifies the service communality with the CINT application block of CVIS, explained in Section 2.4.2.

It is explained that CINT provides EDA and CTA applications. EDA enhances the awareness of the driver and is close to the 'observe traffic conditions' use case of the IW. CTA is close to the 'receive assistance' use case.

This distinction makes it easier to reason about the relevancy of the selected concepts, i.e. it shows a relation between CVIS and the IW. For example; the provided service of module 2 is more 'CTA-like', while for module 3 it is more 'EDA-like'.

Appendix D summarizes IW requirements that might be relavant to take into account for the future integration of the IW with CVIS applications.

4.2.3 Software modules

The existing functionality of the IW is decomposed into 14 software modules. Each individual module provides a unique service to the drivers that use the IW. Almost every module follows the same autonomous[6] process, as shown by the data flow diagram (DFD) [81] in Figure 4.4 on page 31. Exceptions are module 2 and 11, which involve the road operator actor, as shown in Figure 4.3.

The sensor readings (from detection loops, speed sensors etc.) are fed to the module. The module processes this information and logs the events, e.g. 'a

[6] The module does not depend on other modules.

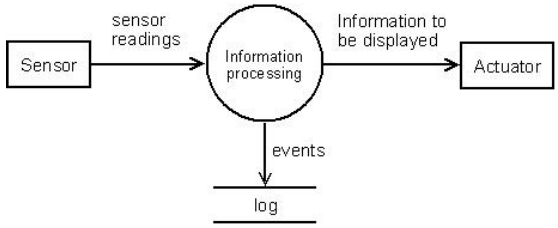

Figure 4.4: DFD for all IW software modules, except modules 2 and 11.

bicycle is detected' or 'a car passed that traveled at 80 km/h'[7]. The result of this processing is information to be provided to actuators. Within the context of the IW the actuators are called SGs. Each module logs activity in the log of its UWS. The individual logs of the UWSs are appended to the system log of the central system *eventually* and not in real-time.

A module is identified by a unique number, ranging from 1 to 14. The functionality of each module is listed in Table 4.1 on page 32. Module 6 and 8 are not operational. Module 14 is irrelevant for CVIS [27], and thus for the demonstrator, because it is essentially a closed system. The response time (last column of Table 4.1) is the maximum allowed time between sensing data and showing the information to the driver through an actuator. It is assumed that the modules with a higher allowed response time are more suitable in a VII demonstration setting as they put less requirements on the timeliness of the information that these modules can provide to other infrastructures or vehicles.

Figure 4.5 gives a graph showing the maximum allowed response time for each IW module.

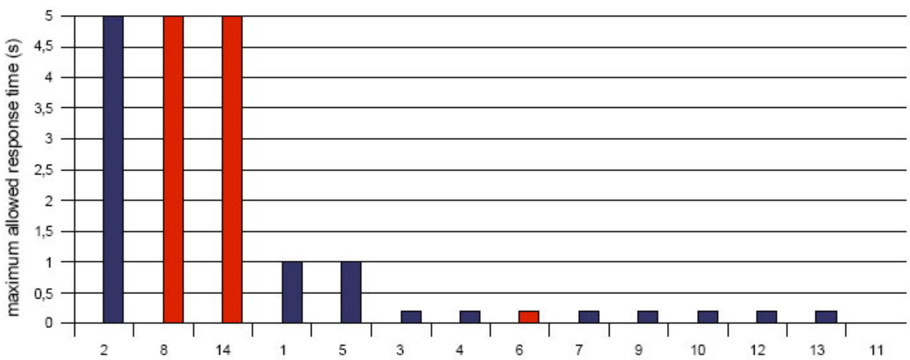

Figure 4.5: Response time for IW software modules, based on [18].

[7] *See* Appendix C for log examples.

#	Name	Function and workings	Resp. time (s)
1	Out-of-order	Drivers are informed when the UWS is out-of-order, the SG displays ‘Buiten bedrijf’.	1
2	Manual intervention road operator	Road operators can display a temporal message on a SG, e.g. in case of road works, detours etc.	5
3	Actual speed	The actual speed of a vehicle is measured within a distance 20 to 300 meters. The actual speed is shown to the vehicle by the SG when the vehicle speed is greater than a specified threshold.	0,2
4	Crossing bicycles	Approaching vehicles are alerted by a SG when bicycles are about to cross over.	0,2
5	Traffic lights out-of-order	Drivers are informed when traffic lights are out-of-order. A VRI sends out a signal when it is out-of-order and a SG shows a message accordingly.	1
6	(Speed advice)	Drivers are advised to drive at a certain speed to be able to be in time at a specific junction (controlled by a single VRI) to catch a green light. This advisory speed is displayed by a SG. Also referred to as LODYSA.	0,2
7	Goods-traffic green light extension	The traffic lights will stay green (are extended) when e.g. a truck is approaching. A SG shows to the driver when the green light extension is active.	0,2
8	(Travel time)	The estimated travel time will be displayed by a SG.	5
9	Vehicle weight	The weight of a vehicle is displayed by a SG.	0,2
10	Close tailing	The amount of seconds to a following vehicle is displayed to the tailing vehicle by a SG.	0,2
11	Traffic analysis	The speed and length of passing vehicles are logged for traffic analysis. The vehicle is categorized based on its length.	-
12	Chained speed advice	A speed advice is given to catch a chain of green lights at downstream junctions. This module communicates with multiple VRIs, while module 6 communicates with one VRI only to provide a speed advice. Also referred to as ODYSA.	0,2
13	Emergency services	A SG alerts the drivers when an emergency vehicle is approaching.	0,2
14	(Road guidance)	Intelligent lights are activated in the road-surface to guide traffic in low-lighting conditions.	5

Table 4.1: Existing software modules of the IW, based on [18].

4.2.4 Functionality located on the provincial road N629

The provincial road N629 provides the most diverse functionality and can be seen as the most comprehensive part of the IW. Figure 4.6 gives a close-up view of the N629 and shows the location of 13 SGs, positioned alongside the road.

A module can run on several UWSs, thus at multiple physical locations. The functionality of the 14 modules is implemented at specific SG-locations, *see* Table 4.2. The location numbers refer to the SG-numbers shown in Figure 4.6.

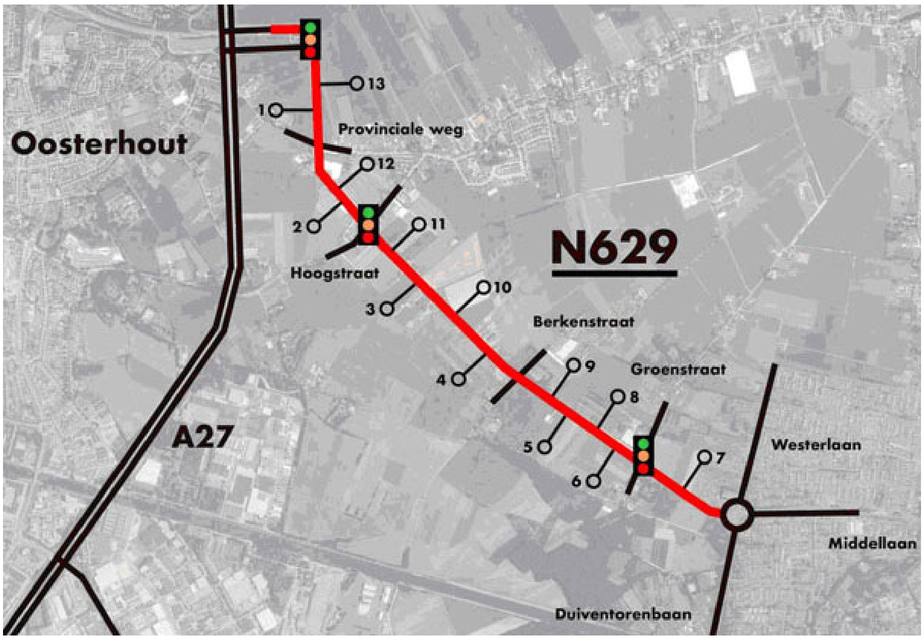

Figure 4.6: Road N629 of the IW, in [17].

Software module / Location	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
1. Out-of-order	•	•	•	•	•	•	•	•	•	•	•	•	•
2. Manual intervention	•	•	•	•	•	•	•	•	•	•	•	•	•
3. Actual speed	•	•	•	•	•	•	•	•	•	•	•	•	•
4. Crossing bicycles	•			•		•	•		•			•	
5. Traffic lights out-of-order		•									•		•
6. (Speed advice)													
7. Green light extension		•									•		•
8. (Travel time)													
9. Vehicle weight					•			•					
10. Close tailing	•	•	•	•	•	•	•	•	•	•	•	•	•
11. Traffic analysis	•	•	•	•	•	•	•	•	•	•	•	•	•
12. Chained speed advice		•	•	•	•	•	•	•	•	•	•	•	
13. Emergency services	•	•	•	•	•	•	•	•	•	•	•	•	•
14. (Road guidance)	•	•	•	•	•	•	•	•	•	•	•	•	•

Table 4.2: Position of the software modules on the N629, in [17].

4.3 Concept selection criteria

This and the following section elaborate on which concrete ITS concepts are relevant for deployment on the IW to form a proof-of-concept for the CVIS project.

In order to select ITS concepts that are relevant for demonstration, the following question should be answered: *What properties should an ITS concept have to be relevant for LogicaCMG, the CVIS project and the province of Noord-Brabant?* Table 4.3 relates the project outcome with corresponding properties of the concept.

Project outcome	**Concept property**
A concept demonstration that sets an example for VII and cooperative ITS.	**1)** It should involve vehicles and infrastructure co-operating in achieving a shared goal, like providing information on IW road works to the driver.
Identification of potential development and deployment issues, which are valuable to the CVIS project.	**2)** It should have some relation with a concept - or a group of concepts - being developed within CVIS. For example, it should contribute to the awareness of the driver (EDA) or assist the driver during his journey (CTA).
Provide an image-boost to the IW project and the province of Noord-Brabant.	**3)** It should integrate with existing (physical) infrastructure of the IW. Or the IW-infrastructure behavior should be simulated when on-road integration is not feasible in this project.

Table 4.3: Relation between project outcome and concept properties.

4.4 Demonstration candidates

This section will present the ITS concepts that are suitable for demonstration on the IW. It shows the match between the ITS concepts described in Chapter 3 and the *selection criteria* presented above. The demonstration *candidates* are derived from this match and will be described more concretely.

An ITS concept is classified according to:

- Its communication-paths (V2V, V2I, I2V and I2I) used;
- If it is an example of VII and thus uses V2I *and* I2V communication (i.e. **property 1** holds);
- Its relation with certain CVIS sub projects, among them are CTA and EDA (i.e. **property 2** holds);
- Its relation with specific IW-modules (i.e. **property 3** holds).

The complete classification of concepts is given in Table 4.4. The concepts were already preselected on their VII aspect (*property 1*) by the taxonomy introduced in the previous chapter, i.e. the taxonomy discovered the VII concepts. *See* Table 3.1 on page 26 for the foundation of this preselection.

Regarding *property 2* and *3*; a concept is related with a CVIS sub project when the concept's provided service aims for the same goal as the sub project of CVIS. A concept is related with an IW-module if the module can contribute to the service provision that is envisioned by the concept.

#	ITS concept	VII	Related to: CVIS	IW-module
1	Driver assistance/active safety*	-	EDA	4,10
2	NIV: traffic perf. assistance	-	-	-
3	NIV: network traffic equalizer*	•	CTA	(8),11
4	Mgmt. in transports and logistics*	•	CFF	7,9
5	ISA at Borlange site*	•	-	3,(6),12
6	TRAFIKEN.nu*	-	CTA	2,5,(8),11
7	Road condition monitoring	◦	CURB	-
8	Traffic probes	-	CURB	-
9	Vehicles providing weather data	-	CURB	-
10	In-vehicle signing	-	-	1-10,12,13
11	Providing information on obstacles*	•	EDA	4
12	Merging assistance	•	EDA	-
13	Internet provisioning	•	-	-
14	Smart parking facilities	•	-	-
15	Location support	-	-	-
16	In-vehicle signing	-	-	1-10,12,13
17	Incident management	•	COMO	-
18	Driver-interacted platooning	-	CTA	-
19	Jackknifing detection	•	CURB	-
20	Logistics vehicle tracking	•	CFF	-
21	Busses working together with bus-stops	•	-	-
22	Closeby motorcycle alert	-	EDA	-
23	Cooperative speed advice*	•	-	(6)
24	Cooperative weighing*	•	CFF	9

Table 4.4: Relating ITS concepts to CVIS and the IW.

The CVIS sub-projects below are not explained before and are referred to by the table:

- *Cooperative Fleet and Freight Applications (CFF)* aim to develop cooperative ITS for the commercial and logistics sector; [70]
- *Cooperative Urban Applications (CURB)* use FCD in combination with infrastructure detections to provide improved traffic information; [70]
- *Cooperative Monitoring (COMO)* aims to develop specifications and prototypes for the collection, integration and delivery of extended real-time information on individual and collective vehicle movements and on the state of the road network. [21]

The demonstration candidates for the IW are derived from Table 4.4 and refer to the concepts indicated with an asterisk (*) in this table. They are described more concretely below.

1. **Active safety.** The driver is alerted through an OBU when bicycles are crossing the road, when his vehicle is too close to its follower and when an emergency vehicle is approaching. Concepts #1 and #11 inform the driver about the traffic situation in front of him. The first concept only uses 'local' vehicle sensors, hence it is not an example of VII. However, its ideas could be useful. Both concepts are related to EDA, described in Section 2.4.2. The IW-modules *crossing bicycles* (4), *close tailing* (10) and *emergency services* (13) could provide useful event data. The information should be provided to the driver synchronously, within a second or less (*see* Figure 2.6 and 4.5).

2. **Informative intelligent speed adaptation (ISA).** The OBU shows a warning when the driver exceeds the speed limit, together with the existing speed limit. It could provide speed advices to catch green lights as well. It is based on concept #5; the informative ISA model [2] from the Swedish Road Administration (SRA), as described in Section 3.2.2 and concept #23. Both concepts have no strong relation with a CVIS sub project. Therefore *property 2* does not hold, but *property 1* and *3* do hold strongly. The IW-modules *actual speed* (3) and *chained speed advice* (12) could provide useful event data. At the time of writing module *speed advice* (6) is out of order. The information should be provided synchronously and in real-time.

3. **Journey assistance.** Related to CTA, described in Section 2.4.2. Two variants are distinguished, both providing real-time information to a driver:

    - On-trip information, the driver receives information on road-works, out-of-order traffic lights and upcoming traffic jams based on his current location. It is inspired by concept #6, the TRAFIKEN.nu website [78], *see* Section 3.2.2. The VII aspect of this concept is achieved by using the event data of the IW modules *manual intervention road operator* (2), *traffic lights out-of-order* (5) and *traffic analysis* (11) for in-vehicle information. Estimated travel time could have been provided as well, but the module *travel time* (8) is out-of-order.
    - Traffic routing, the driver receives on-trip navigation advisories to improve traffic efficiency based on pre-trip data, provided by the driver. The solution should contain a routing engine, which makes this variant relatively complicated to realize. It is based on concept #3 and is 'pure' CTA. Statistical information from IW module *traffic analysis* (11) could be used. However it is costly to provide data of module 11 to third parties according to the Province of Noord-Brabant. This makes 'traffic routing' possibly a less feasible option. Module 8 provides useful data as well, but is unavailable.

4. **Cooperative weighing.** A vehicle is weighed on the IW and its weight is communicated to the driver and can be sent to transport- or logistics stakeholders. It has the closest relation with concept #4 and the CFF sub

project of CVIS. It uses module *vehicle weight* (9). The information provisioning to the aforementioned stakeholders could occur *a*synchronously, i.e. it does not need to happen real-time.

The concepts that are *not* chosen to be a demonstration candidate include those that do not hold one or more of the three aforementioned properties strongly.

For example, the 'incident management' (#17) concept was not chosen because it has no relation with specific IW-modules, i.e. it does not hold *property 3*. The concept 'in-vehicle signing' (#10) has a strong relation with IW-modules, but is not a VII concept and is therefore not a candidate, i.e. it does not hold *property 1*.

Chapter 5

Demonstrator

This chapter elaborates on the developed software that demonstrates cooperative weighing on the Informatieve Weg (IW). A vehicle infrastructure integration (VII) framework has been implemented, which can be used to demonstrate VII concepts in general. Cooperative weighing is built on top of this framework and exercises the framework's architecture.

5.1 Selected concept: cooperative weighing

Cooperative weighing was introduced in Section 3.3 and provides the vehicle's weight, which is measured by the infrastructure, to the driver and to other stakeholders.

Heavy trucks cause road wear and could be unsafe. Most countries apply rules for (maximum) axle weights and trucks with a high axle weight are taxed. For these reasons the truck driver and third-parties (e.g. logistic company, government or trading partner) are interested to know the vehicle's weight, also during driving. In case of international transport the need for this weight-information could be even higher because different regulations can apply.

The concept of **cooperative weighing** is picked from the demonstration candidates listed in Section 4.4 on page 36, for the reasons below.

- It is a new and innovative concept compared to the listed candidates 1, 2 and 3.
- It uses an IW-module that has a higher chance of being available for a demonstration. E.g. candidate 3 uses IW-module 11, which has a low chance of being available for third-parties.
- It fits into the cooperative concepts of the CVIS project and can be part of the CTA application. It could also be used in the CFF sub project.
- Its implementation is considered to be feasible within a short time frame and it is the least time-critical of the four candidates.

5.1.1 Functional details

Cooperative weighing is based on a weigh-in-motion (WIM) sensor, which is built into the road-surface of the IW. This sensor measures the weight of a vehicle while it moves over it. The measured weight is provided by the 'vehicle weight' module (no. 9) of the IW. An essential problem is how to determine *which* vehicle is weighed by the WIM sensor. This can not be determined directly with the current infrastructure/sensor on the IW, as it can not identify a vehicle[1].

For this project it is infeasible to change the existing infrastructure. Therefore the vehicle's identity is deduced from location information of the vehicle and weigher, *see* Figure 5.1.

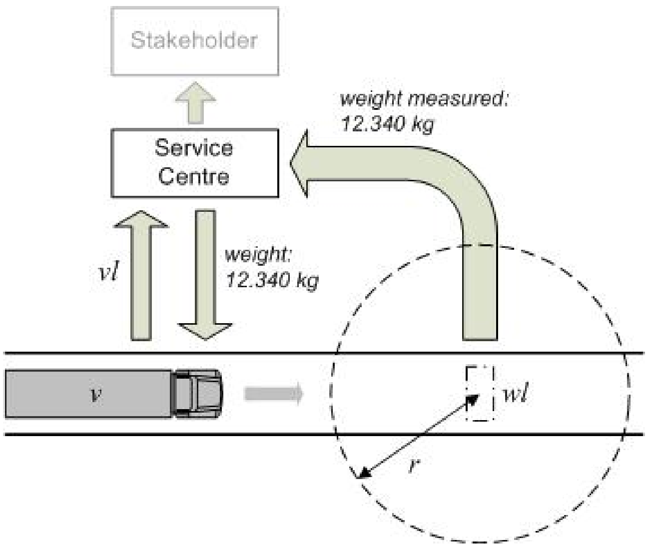

Figure 5.1: Cooperative weighing, an example of VII.

The fixed position of the WIM sensor is indicated by `wl`. A truck `v` is connected with a Service Centre and sends its (changing) location `vl` periodically. The weight-measurements coming from the road-side infrastructure are sent to the Service Centre. When truck `v` is in radius `r` of weigher location `wl`, the Service Centre 'listens' for measurements and sends the weight to the vehicle[2].

This method is not fail-proof, e.g. the Service Centre is unable to identify vehicles when two of them are moving over the sensor in a too short period of time.

The concept is an example of VII, because it both uses:

- V2I communication, as `v` sends its location to the Service Centre;
- I2V communication, as the Centre sends the measured weight to `v`.

[1]License plate recognition or other automatic vehicle identification (AVI) techniques are necessary to identify a vehicle.

[2]For the range, the *direct distance* between the vehicle and the weigher is taken and was sufficient for this demonstration setting, e.g. it would have been overkill to take the road distance or the like.

5.1.2 Demonstration script

This script defines the chronological order of events that are executed during the demonstration of the cooperative weighing concept, taking into account the limitations posed by the infrastructure.

Preparation

1. The weigher location is chosen. This is done 'virtually', i.e. the latitude/longitude coordinates of a position somewhere on the Heistraat (N629 road of the IW) is obtained using Google Earth [26]. The weigher location `wl` is set to: $51.631977, 4.909472$ (decimal latitude and longitude degrees). These coordinates can be changed easily to the 'real' weigher location later.
2. A (web)server is installed and connected to the Internet.
3. A car will be equipped with an OBU, which runs the demonstrator application.
4. A simulator application is initialized to mimic the IW-infrastructure, i.e. it sends arbitrary and timestamped weigher-events to the server, like: 'weight measured: 1.239 kg' or 'weight measured: 3.392 kg'.
5. The OBU is configured to connect with the Internet through General Packet Radio Service (GPRS) or UMTS.
6. A GPS unit is connected to the OBU via Bluetooth.

Execution

The numbers of the execution steps below match the numbers in Figure 5.2 on page 42.

1. The car is positioned at the junction Rijensestraatweg/Westerlaan and the driver instructs the OBU to connect to the server.
2. The car moves north onto the Westerlaan. The OBU displays the linear distance[3] in meters to the weigher location `wl` periodically.
3. The car turns left onto the Middellaan.
4. The car passes the weigher location `wl`.
5. The OBU displays a simulated/arbitrary weight (provided by the server) to the driver within a reasonable amount of time.

[3]The linear distance is the length of a direct line between two map points.

Figure 5.2: Map explaining the cooperative weighing demo.

5.2 Generic VII framework

All VII concepts have a 'communication commonality', i.e. facilities have to be in place to communicate with OBUs and road-side infrastructure.

Sivaharan et al. [69] concluded that there is a real need for middleware in the area of cooperative vehicles to ease the burden on the application developer. Therefore, a generic VII framework has been implemented, which can act as a communications/middleware platform to demonstrate VII concepts in general.

The author's vision of this framework is presented in Figure 5.3.

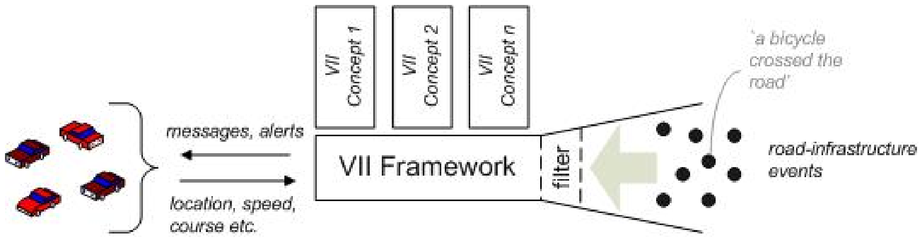

Figure 5.3: Author's vision of a VII framework acting as a communication-platform.

It envisions vehicles that can be provided with messages coming from the 'infrastructure'[4] and can provide information (like location and speed) to this infrastructure. The framework, being part of the infrastructure, can filter events coming from the road-side infrastructure (like the IW). VII concepts obtain vehicle and event data from the framework in order to provide messages to *individual* vehicles.

The rest of this chapter shows the work that has been done to work towards this vision.

5.2.1 Requirements and constraints

The following requirements are met by the framework:

1. It foresees in a two-way message-based communication between an OBU and a server. This way the framework can mimic V2I and I2V communication.

2. It keeps a model of the vehicles' state (like location and speed), which is used by the VII concepts.

3. The 'generic' aspect of the framework translates into a means to plug-in an arbitrary amount of VII concepts, as the ones listed on page 36.

4. The amount and variety of the events coming from the road-side infrastructure can be massive. Therefore, the framework filters these events for the plugged-in concepts.

[4]A differentiation is made between road-side infrastructure (e.g. the systems alongside the IW) and the infrastructure in general, which *also* includes the framework.

5. It mimics the CALM[5]-'behavior' in the way it is able to send messages to an *individual* (and identified) OBU, residing in a vehicle.

For the project it was a constraint to use an OBU that could communicate wirelessly by a 'publicly available' Global System for Mobile communications (GSM)/GPRS network. E.g. the use of road-side transponders or DSRC portals is an alternative, but was infeasible for this project.

It was decided to use a Nokia E61 smartphone [59] as an OBU. The phone has broad communication capabilities, like; WiFi, Bluetooth, GPRS and a direct cable connection (for testing purposes). Additionally, the phone can provide information to the vehicle-driver through a relatively large screen. *See* Figure 5.4.

Figure 5.4: A Nokia E61, used as an OBU.

The framework was implemented in the Java programming language. However, the ideas explained in this section can be implemented in other programming languages as well.

5.2.2 Platform limitations

A Java environment involving communication with mobile phones poses the following limitations:

- *The phone has to initiate* the communication dialogue. It is difficult to let the server initiate and push content directly to the phone. The IP-address of a mobile phone is assigned by the provider in a dynamic way, i.e. the same IP address is not bound to a specific phone.

  Pushing mechanisms on mobile phones include SMS and Multimedia Messaging Service (MMS), which need an infrastructure-side server sending these messages. This could not be realized in the project.

- The Java mobile phone application[6] supports *HTTP connections only.* Socket or datagram connections are <u>optionally</u> supported [48, 67].

[5] *See* Section 2.4.3 on page 14.
[6] Implementing the Mobile Information Device Profile (MIDP) 2.0.

5.2.3 Design decisions

The following design decisions were made, which are based on the aforementioned requirements, constraints and limitations of the development platform:

1. A *client/server topology* [6] was chosen, where the OBUs are the clients and the framework is the server. A peer-to-peer (P2P)[7] setup [6] would have been ideal because both the OBU and framework have information that might be interesting to the other party. However, this is impossible because the development platform poses the limitation that the phone has to initiate the communication dialogue.

2. It is crucial that information can be 'pushed' from the framework to the OBU, e.g. an event sensed by the road-infrastructure should be shown to the driver by an OBU.

   A *poll-method* was used as a workaround for the limitation that the phone has to initiate communication. For this method, the OBU polls the server periodically to check for new information. This method approaches the behavior of a server that pushes messages to an OBU.

   Of course this method causes a lot of network traffic and could lead to server resource contention and congestion [12]. However, these problems are not an issue for this low-scale demonstrator setting, i.e. only one vehicle is equipped with an OBU and is driven during demonstration.

3. It was decided to keep the size of the information to be exchanged between the OBU and the framework to a bare minimum to keep the GPRS communication latency as low as possible. This to be prepared for performance issues.

4. The connection mechanism at the client side is limited to HTTP, which is a request-response protocol. Therefore, the framework/server should 'talk' HTTP. The use of Servlets[8] is within the Java context a logical choice to realize this.

   It was decided to use a *middleman architecture* [48]. Here a Java Servlet acts as a 'middleman'. This middleman can be seen as a gateway for (mobile) clients to a server back-end and eases the connection of a HTTP-restricted client.

   Seedorf et al. [67] also used this architecture in a mobile context. Their 'topic map query tool' enables mobile workers to retrieve information from a server, using a mobile device.

[7] Where either party initiates a connection.

[8] *'Servlets are protocol- and platform-independent server side components, written in Java, which dynamically extend Java enabled servers. They provide a general framework for services built using the request-response paradigm.'* [71]

5.2.4 Deriving the architecture

The architecture of the framework is presented in Figure 5.5. The logic of one or more VII concepts resides in the client and server application, as indicated in the figure by the dotted rectangles[9]. This logic determines a VII concept demonstration.

The architecture crosses four execution environments: the *OBU*, *Web Container*, *Server* and *Road-side Systems*. The last three together can be considered as the 'infrastructure' in a vehicle infrastructure integration (VII) setting. The behavior of each individual environment is covered next.

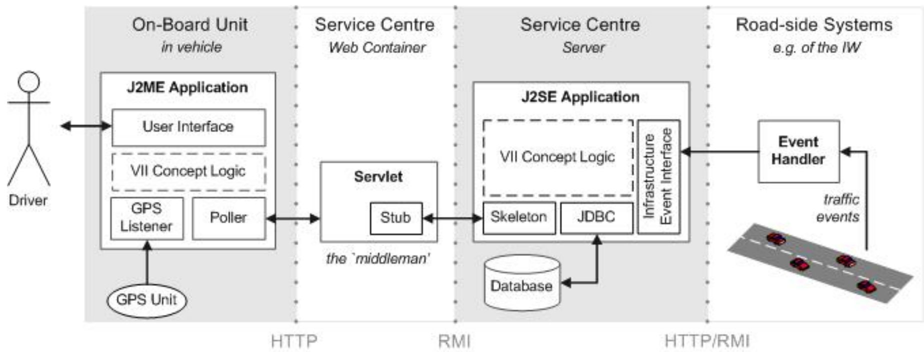

Figure 5.5: Architecture of the generic VII framework.

1. *OBU*. A Nokia E61 smartphone acts as an OBU, which runs a Java Platform Micro Edition (ME) application. This application provides information and services to the driver. It receives location, heading and speed information from a GPS unit and calls a Java Servlet *periodically* to exchange information.

2. *Web Container*[10] that runs a Java Servlet. The Servlet provides an effective means to handle request messages from the OBU and to send back a response, e.g. 'a bicycle is crossing the road'. The response is calculated by the Server.

3. *Server*. A Java application acts as a server and can be connected to a database, if necessary. It receives messages from OBUs (through the Servlet) and events from the road infrastructure. Based on these two it calculates response-messages to be sent to specific OBUs.

4. *Road-side Systems*. The road 'generates' events (e.g. a bicycle is crossing the IW) and notifies the Service Centre. For this project a simulator was used instead of the road-side systems, *see* Section 5.2.5.

The four environments of the framework are connected using middleware, indicated by the three dotted vertical lines in Figure 5.5. These three middleware 'bridges'[11] are explained next.

[9] This visualizes the third requirement of the framework.
[10] Apache Tomcat 5.0 [73] is used.
[11] A bridge in the sense that a transparant connection is made using middleware.

OBU and Service Centre

This bridge is the most sophisticated of the three and is therefore described in detail. Britton et al. [6] identify eight middleware elements, of which three are considered important to clarify this bridge:

- *Communications link*. The physical connection is provided by GPRS and a cable connection for testing purposes. The *IP-passthrough* [58] function of the wired connection is ideal for testing purposes, because no GPRS traffic costs have to be made. The cable connection can be changed easily to a wireless connection by selecting a different communication profile on the phone. The communication bearer is thus transparent to the application. HTTP over TCP/IP is used on top of this physical connection.

- *Middleware protocol*. A client/server setup is used, with the OBU as client and the Service Centre as server. The OBU initiates the connection and periodically contacts the server to exchange information.

  A UML state machine diagram in Figure 5.6 shows the behavior of the OBU and the Service Centre. It uses the global variables `p` and `r` to model the communication between the OBU and the Centre. The protocol is message-based: the OBU polls the server by use of an HTTP POST message `p` and the Service Centre replies with an HTTP response `r`.

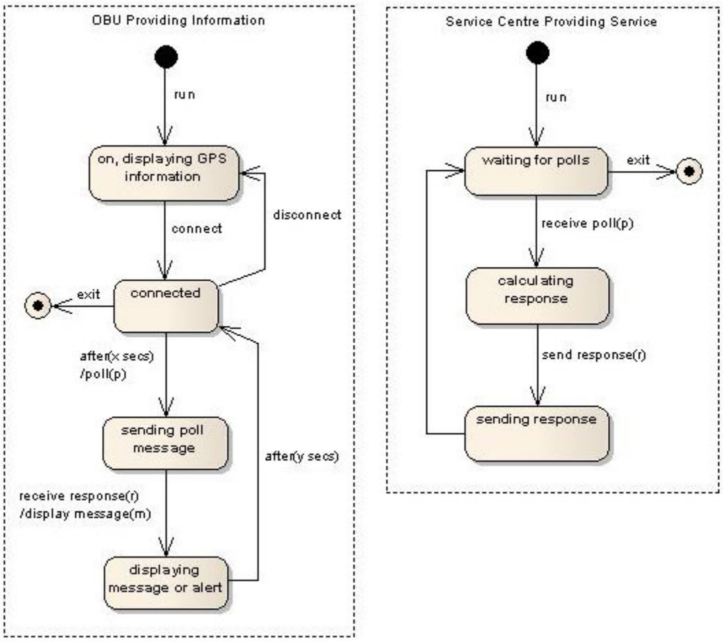

Figure 5.6: UML state machine diagram describing the interaction between the OBU and Service Centre.

- *Data presentation.* The OBU sends its location-information[12] to the Service Centre. The Service Centre has to send information based on road-traffic events to the OBU.

  This information is exchanged in the JavaScript Object Notification (JSON) [11]. A JSON string is a serialization of structured data, e.g. a set of (nested) name/value pairs. A JSON message is shown below, which is sent by the OBU periodically.


```
{
  "obuID":1,
  "location":{
    "longitude":51.978959,
    "latitude":5.925751,
    "speed":123
    "course":60
  }
}
```


  It is important to note that the `obuID` has to be sent with each OBU-request. Only then the Service Centre is able to map its local state to the (new) information provided by the OBU.

  JSON is used in this framework because of its concise notation, which was considered important to keep communication-latency as low as possible (third design decision). Additionally, JSON libraries were available for the mobile- and server platform [41].

  The Extensible Markup Language (XML) runs over HTTP as well, but leads to a higher communication latency because more data has to be transmitted. It is also more complex to implement, i.e. more advanced parsers are needed in a mobile environment to weigh up against the restricted memory and processing power available.

The use of XML web services [6] is a more appropriate choice if the latency of the communication media is low enough. Furthermore, it was not easy to implement web services using the Nokia smartphone because it does not support the Java Specification Request (JSR) 172 [60]. This JSR provides web services middleware on the mobile Java platform.

Servlet and Server (within the Service Centre)

The Servlet and Server communicate (and are decoupled) using Java's Remote Method Invocation (RMI)[13] [68]. This way, the architecture becomes scalable. The Web Container and Server can become redundant and OBU requests can be load-balanced.

Service Centre and Road-side Systems

This bridge could not be realized in this project because of organizational lingering. A simulator setup was used instead and is described in the next section.

[12] Additionally, the speed and course of a vehicle is provided.
[13] Other middleware techniques are possible as well [6].

Appendix F provides the details on how to connect the road-side systems of the IW to the Service Centre. The proposed solutions could be used as a starting point to integrate the framework with road-infrastructure other than the IW as well.

5.2.5 Simulating traffic events

VII involves, by definition, a connection with some kind of road infrastructure. For this project it was infeasible to realize a connection with the systems of the IW[14]. Therefore a simulator was needed to mimic the IW.

As explained before, the road infrastructure could 'generate' events, e.g. 'an emergency vehicle is approaching' or 'the traffic lights are out-of-order'. The VII framework should pick-up these events to achieve cooperation between vehicles and infrastructure.

LogicaCMG has developed an application that simulates events originating from *OBUs* already: the **OBU Simulator**. This application is used in this project to simulate *road-events* instead. A screenshot of the OBU Simulator is shown in Figure 5.7.

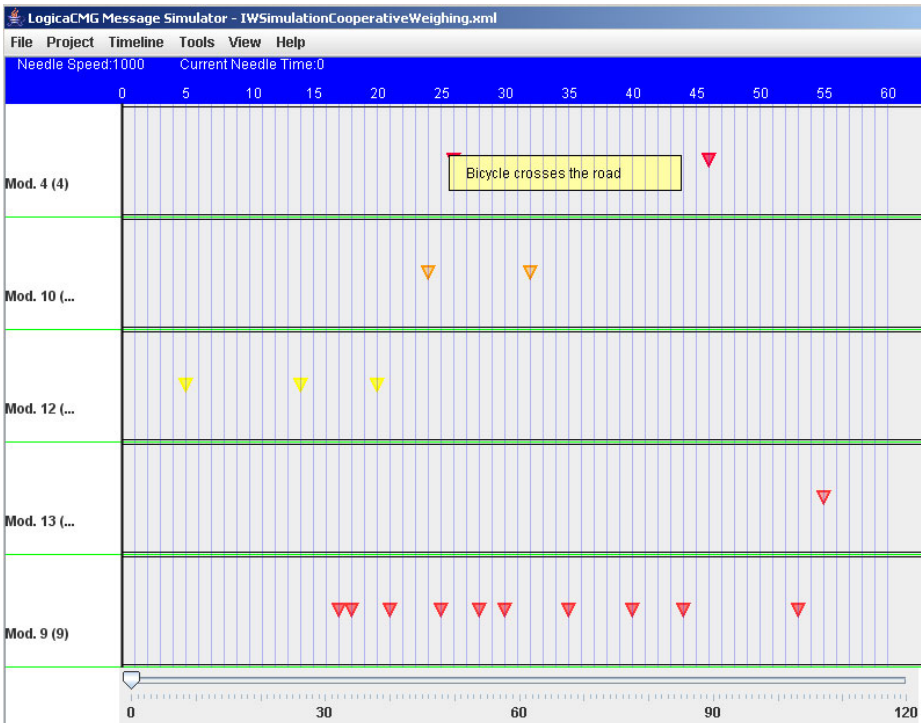

Figure 5.7: LogicaCMG's OBU Simulator simulating IW-events.

The application displays timelines, which can be 'played'. A time-needle shows the current time and moves from the left to the right of the screen. An event, indicated by a triangle (▽), is triggered when the needle passes it. Additionally, a notification is sent to a server indicating that the event occurred.

[14]However, several alternatives are proposed in Appendix F to realize this.

Figure 5.7 shows the simulator with five timelines, each representing a certain IW software module. For example, IW module 4 (crossing bicycles) generates the event 'Bicycle crosses the road' at $time = 26$.

A simulation can be stored in an XML project file. An example fragment of the used project file is listed below.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project>
3   <configuration>
4     <protocol name="HTTP" ipaddress="127.0.0.1" port="80" url="/CVISServlet/CVISServlet"/>
5     <proxy url="" port="0"/>
6   </configuration>
7   <timelines>
8     <timeline id="4" name="Mod. 4" lenght="1" sourceLocation="">
9       <fragment name="Bicycle crosses the road" messageType="" startTime="26"
10        responseTime="0" async="true" color="-65485" sourceLocation=""
11        source="{&quot;uwsOrigin&quot;:3,&quot;moduleID&quot;:3,&quot;eventType&quot;:2336}"
12        expectedsourceLocation="" expectedsource="" compareMethod="NONE"/>
13      <fragment name="Bicycle crosses the road" messageType="" startTime="46"
14        responseTime="0" async="true" color="-65485" sourceLocation=""
15        source="{&quot;uwsOrigin&quot;:3,&quot;moduleID&quot;:3,&quot;eventType&quot;:2336}"
16        expectedsourceLocation="" expectedsource="" compareMethod="NONE"/>
17    </timeline>
18    <timeline id="13" name="Mod. 13" lenght="1" sourceLocation="">
19      <fragment name="Emergency vehicle detected"
20        messageType="" startTime="55" responseTime="0" async="true"
21        color="-52429" sourceLocation=""
22        source="{&quot;uwsOrigin&quot;:1,&quot;moduleID&quot;:13,&quot;eventType&quot;:2342}"
23        expectedsourceLocation="" expectedsource="" compareMethod="NONE"/>
24    </timeline>
25    <timeline id="9" name="Mod. 9" lenght="1" sourceLocation="">
26      <fragment name="Vehicle weighed" messageType="" startTime="17"
27        responseTime="0" async="true" color="-52429" sourceLocation=""
28        source="{&quot;uwsOrigin&quot;:1,&quot;moduleID&quot;:9,&quot;eventType&quot;:2342,
29         &quot;eventText&quot;:&quot;1.034 kg&quot;}" expectedsourceLocation=""
30         expectedsource="" compareMethod="NONE"/>
31    </timeline>
32  </timelines>
33 </project>
```

The server, which receives the event-notifications, is specified in the `configuration`-tag (lines 3-6). Furthermore, `timeline`s can be defined (lines 8-31), which contain `fragments` (events). A `fragment` contains a `source` (e.g. lines 15 and 22), which defines a JSON[15] notification. This notification is sent to the VII framework for further processing at the moment the event occurs.

The `fragment` shown in lines 26-30 is an example of a weigher-event used to demonstrate the cooperative weighing concept, which is introduced at the beginning of this chapter.

[15] *See* Section 5.2.4.

5.3 Plug-in of cooperative weighing

A VII concept can be plugged into the VII framework by 'filling-in' the dotted rectangles shown in Figure 5.5 on page 46. In case of multiple plugged-in concepts, the framework executes these concepts in sequential order when the framework receives a poll-message[16] from an OBU. The plug-in of a concept is achieved technically by creating a Java-class that implements the `VIIConcept`-interface of the framework, *see* Appendix E.

The framework handles the communication to and from the OBU(s) and filters events that occurred at the road-side infrastructure. Generally, a VII concept *consumes information* coming from OBUs and the infrastructure. Based on these two 'streams of information', the concept *calculates messages* to be sent to individual OBUs.

5.3.1 Algorithm

The concept of **cooperative weighing** was chosen for demonstration and was explained in Section 5.1. The vehicle's weight is measured on the IW and is sent to the OBU, to be displayed to the driver. The measurements coming from the infrastructure/IW are simulated. The algorithm of the cooperative weighing concept is listed in pseudo-code below.

Require: location wl of the weigher, i.e. latitude and longitude
Require: detection radius r in meters
1: **repeat**
2: get location vl of vehicle v (measured at time t) from framework
3: **if** (distance between vl and wl) $<= r$ **then**
4: get weigher-event e that occurred closest to t from framework
5: **if** e is not offered to a vehicle before **then**
6: compose message m based on e
7: offer m for vehicle v to framework
8: // weight is sent to vehicle
9: // weight is displayed by the on-board unit
10: **end if**
11: **end if**
12: **until** on-board unit application disconnects

Using the framework, the concept *consumes* the location of a vehicle and the weigher-event (coming from the IW-simulator) in lines 2 and 4 respectively.

The location of the weigher, a latitude/longitude coordinate, is constant and known beforehand. The concept *calculates* the distance between `wl` and `vl` using a database with a spatial extension in line 3.

When the vehicle is 'in-range' (i.e. within radius `r`) of the weigher, a message containing the weight of the vehicle is sent through the framework in lines 5-10.

More implementation details are covered by Section E.2 in Appendix E.

[16] *See* in Section 5.2.4 explanation of the *middleware protocol.*

5.3.2 Deployment of the demo

A 'CVIS Service Centre' server[17] is arranged in order to execute the demonstration. This server of LogicaCMG is physically located at the D-CIS Lab in Delft [14]. It is wired to the Internet and runs the following:

- A database, which stores the location information of the vehicles and does some geographic calculations (*see* Section E.2 for more details);
- The Service Centre part of the VII framework with the plugged-in cooperative weighing concept (as explained by this Chapter);
- The simulator application, which simulates the road-events coming from the IW (as explained in Section 5.2.5).

The server works together with the Nokia smartphone (the OBU). It runs the OBU part of the VII framework and is connected to a regular GPS unit[18]. The OBU will be traveling in-car accross the N629 of the IW.

The situation is visualized in Figure 5.8. The solid arrows represent the information flow.

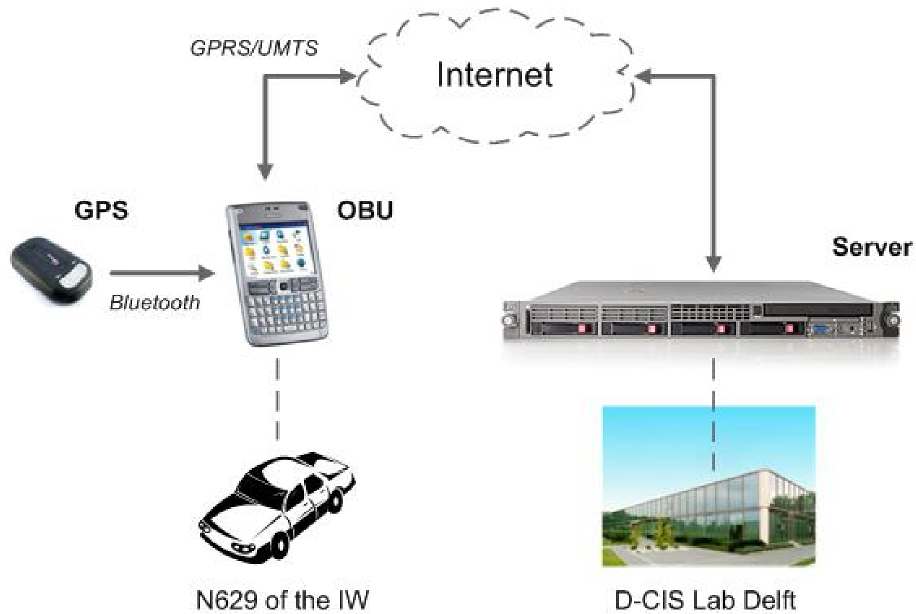

Figure 5.8: Deployment of the cooperative weighing demo.

[17] An HP ProLiant DL360 with an Intel Xeon 3.0 GHz processor and 2GB of memory [28] is used as a server.

[18] An Adapt-Mobile AD-900 Bluetooth GPS receiver [1] is used. This is a unit in the user segment.

Chapter 6

On-road demo

This chapter evaluates the on-road demonstration of the cooperative weighing concept. The traffic events coming from the Informatieve Weg (IW) were simulated for this demonstration.

6.1 Evaluation criteria

The demonstration is evaluated according to the criteria below, which refer to Figure 5.1 on page 40.

1. The driver of car `t` is informed of the car's weight within 20 seconds[1] after he passed the location of the weigher `wl`.
2. The OBU displays the weight message only once, i.e. the alert-window on the OBU is not displayed repetitively because this could annoy the driver.
3. The OBU shows the distance to the weigher 'live' during driving.

6.2 Execution

The demonstration was executed on January 18, 2008 and according to a predefined script, which has been described in Section 5.1.2. The demo is recorded on video [55], wherein the display of the OBU is shown at suitable moments.

During the *preparation* phase, the server was initialized remotely, i.e.:

1. the Service Centre application was started
2. and the demo XML project file[2] was loaded into and 'played' with the simulator application.

The OBU was connected to the Service Centre using a GPRS connection. The faster UMTS connection was not used because a UMTS-enabled SIM card, to be used in the OBU, was not available.

[1] This response time was set before the demonstration was executed and seemed to be a good enough challenge for the framework. It could have been even lower (higher amount of seconds) for this specific non-time-critical concept.

[2] For an example, *see* Section 5.2.5.

Step one of the *execution* phase posed a first problem. The starting point (Rijensestraatweg) goes under the Westerlaan, i.e. the planned route did not take into account road-leveling. A different starting point had to be chosen: a nearby industrial zone at the junction Rijenseweg/Steenstraat. The remaining four execution steps could be followed as planned. A few tasks were executed before driving:

1. the configuration of the OBU was verified (*see* Figure 6.1),
2. the OBU was connected with the GPS unit
3. and the connection to the Service Centre was established after a GPS-fix[3] was obtained.

Figure 6.1: Settings screen of the CVIS OBU.

Then the car drove on the IW towards the weigher location. The main screen of the OBU showed the 'distance to weigher' and real-time GPS information during driving, *see* Figure 6.2 and the video [55].

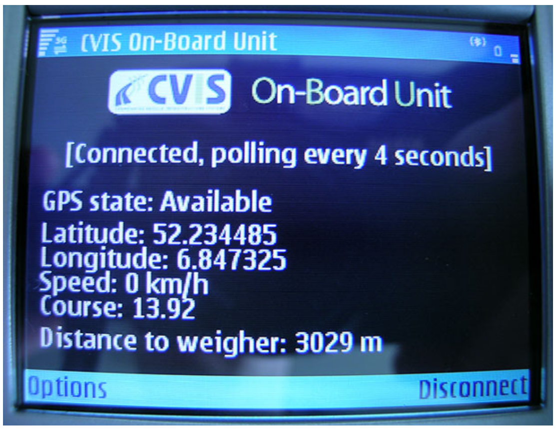

Figure 6.2: Main screen of the CVIS OBU showing information.

[3]A GPS unit receives a GPS-fix when it 'knows' its geographic coordinates/position.

When the car approached the weigher location closely, the OBU displayed the measured weight (coming from the Service Centre) almost instantly, *see* Figure 6.3.

Figure 6.3: The CVIS OBU displaying a weight alert.

When the car drove passed the weigher location, the 'distance to weigher' (as displayed by the OBU) increased continually as expected. Then the car turned around on the Heistraat and drove towards the weigher location again. While doing this, the 'distance to weigher' decreased again. When the car drove passed the weigher location for the second time, the alert box displaying the weight was shown again on the OBU.

During the demonstration, a log-file was kept by the Service Centre. Additionally, the car's route was stored in the database. This route can be shown in Google Earth [26]. Figure 6.4 on page 56 shows the route traveled by the car during the demonstration.

Figure 6.4: Google Earth displays the route traveled by the demo car.

6.3 Results

The demonstration was successful and did result into the following, referring to the aforementioned evaluation criteria.

1. The driver was informed of the (simulated) weight almost instantly (i.e. within 2 seconds) when the car passed the weigher location. It turned out to be practically impossible to measure this exactly without sophisticated measuring equipment because the car drove around 60 km/h.

2. The alert box showing the weight was not displayed repetitively. However, when the car drove passed the weigher location for the second time (i.e. it drove a few hundred meters passed the weigher location, it turned around and approached the weigher location again) the alert box was shown again. This is considered as desired behavior.

3. The OBU application showed the approximate distance to the weigher in a 'live' manner, as shown by the video [55]. When the car approached the weigher, the displayed distance decreased. When the car drove away from the weigher, the displayed distance increased.

Chapter 7

Conclusion

This chapter concludes this thesis. Three sub-objectives led to the realization of the project's main objective: *identify potential development and deployment issues of a vehicle infrastructure integration (VII) application by building and demonstrating a prototype*. This chapter discusses the three sub-objectives first. Recommendations are given thereafter, which embody the outcome of the main objective. The chapter concludes with future work.

7.1 Discussion

The main objective has been achieved by realizing a demonstrator application that showed a cooperative ITS concept on the Informatieve Weg (IW). The total project work was split into three sub-objectives, which are:

1) Give an unbiased overview of Intelligent Transportation Systems (ITS), thereby concentrating on vehicle infrastructure integration (VII) and cooperative systems;

2) Specify and select one VII concept to be demonstrated with the IW;

3) Implement the specified VII concept in the form of a demonstrator and demonstrate it on the IW.

Section 1.4 defined research questions for each particular sub-objective. The answers to these questions are drawn next.

7.1.1 Providing the overview of cooperative ITS

In Chapter 2 the thesis establishes the current and future state of cooperative ITS and positions the CVIS project herein. This information is used as a foundation for the rest of this thesis and is useful for LogicaCMG to come to a common ground of understanding in the field of cooperative ITS.

The chapter describes the shift from non-cooperative towards cooperative ITS, which become more appearant. This shift is triggered by ongoing innovations in the field of vehicle electronics, information and (wireless) communication technology and broadcasting. Examples of these innovations are: adaptive cruise control (ACC), the Internet, data fusion, fiber optics, Communication

Access for Land Mobiles (CALM), the Global Positioning System (GPS) and Galileo. These fields 'blend' and make cooperative ITS possible, together with new concepts and applications.

The CVIS project is, among others, part of this shift and is based on CALM technology. CALM creates an omnipresent communication environment where the vehicles and infrastructure can exchange information in a transparent way.

ITS concepts are discovered in Chapter 3 in order to find a suitable concept to demonstrate on the Informatieve Weg (IW). The chapter gives a summary of finished and ongoing ITS projects in Germany, Sweden, USA and Japan. The projects' underlaying cooperative ITS concepts are revealed and described. Besides these already existing concepts, several (new) concepts are listed that crossed the mind of the author.

A taxonomy based on communication-paths is explained, which is used to 'filter' these discovered ITS concepts to identify only those concepts that depend on both vehicle-to-infrastructure (V2I) and infrastructure-to-vehicle (I2V) communication. These concepts are being called the VII concepts, which are *fundamentally* interesting to demonstrate on the IW.

7.1.2 Selecting the VII concept for demonstration

A subset of the discovered VII concepts (Chapter 3) is selected to provide a list of concepts that are suitable for demonstration with the IW. This selection process is described in Chapter 4 and results into *demonstration candidates*, which are presented on page 36.

These demonstration candidates are applicable in the context of both CVIS and the IW. To found this selection, a technical analysis of the IW has been executed and is described in Section 4.2, Appendix C and D. The existing functionality of the IW is related to the discovered VII concepts. Additionally, the described details of the CVIS project (Section 2.4) were used to relate the VII concepts with CVIS sub-projects.

This selection process and the taxonomy mentioned earlier enable LogicaCMG and others to structurally select ITS concepts that are suitable to be demonstrated/deployed in a specific infrastructure environment. The process relates goals of the demonstration with properties that should be held by the concept to be demonstrated.

The concept of *cooperative weighing* was chosen for demonstration, mainly because of its innovativity and that it used functionality of the IW that had the most chance of being available for integration in a feasible amount of time. Cooperative weighing provides a truck's weight to its driver and to third-parties, e.g. a logistics company. The execution of this concept is described stepwise in the form of a demonstration script. This all is covered by Section 5.1.

7.1.3 Realizing the demonstration

The foundation has been laid for the cooperation between LogicaCMG and the Province of Noord-Brabant in order to realize a cooperative ITS with the IW. This is brought about with meetings and the exchange of (technical) information, e.g. the technical analysis of the IW (Section 4.2) is based on information provided by the Province and a communication proposal document (Appendix F) was sent to the Province in order to come to a common ground of understanding.

Despite of this effort, it was impossible to *technically* integrate IW's systems with a demonstrator within the project's time. The main reason for this is that the Province partly subcontracted the technical fulfillment of the IW project. This led to a lingering technical focus, of all involved parties, on the realization of an *integrated* demonstrator.

Appendix F proposes several middleware solutions to achieve this technical cooperation. The solutions are useful as a starting point for LogicaCMG and the Province to make an integration of their systems happen.

Simulating traffic events

Because of the aforementioned reasons, it was decided to use a simulation setup to realize the demonstration. This setup makes use of a simulator 'stub', which was available in-house. It mimics traffic events, e.g. the events that could occur on the IW. Traffic events occur at a certain point in time and give information about the (changed) state of the road, e.g. 'a bicycle is detected' or 'a car is moving x km/h at location y'. The stub is described in Section 5.2.5.

Implementing the demonstrator

Chapter 5 and the first section of Appendix E provide in-depth technical detail of the implemented demonstrator.

A vision for a generic communication/middleware platform for VII is given in the beginning of Chapter 5, Figure 5.3. The work towards this vision is done in the form of a *generic* VII framework, which provides the necessary (communication) facilities to realize VII demonstrations. This is done with the intention to provide a reusable and adaptable platform that can be used by LogicaCMG to experiment with VII. For example; the cooperative weighing concept was used for this demonstration, but LogicaCMG could decide to demonstrate other VII concepts using this same framework.

The framework is implemented in the Java programming language and uses a Nokia smartphone as an OBU. It was a project constraint to use a mobile phone connected through a 'public' telco-network. This forced the use of a client/server topology, wherein the OBU acts as a client and a Service Centre as the server.

The used topology and used OBU are a suboptimal choice in an *out-of-lab* environment (but are sufficient for demonstration purposes) because:

- Client/server is suboptimal because both vehicle and infrastructure want to provide information to the other party in a VII setting. A peer-to-peer (P2P) topology is ultimately a better choice.

- The used OBU is suboptimal because it is not physically integrated in a vehicle and can not provide e.g. a haptic interface to the driver.

Section 5.3 and the last section of Appendix E describe the functional details of cooperative weighing and how the concept uses the facilities provided by the VII framework. This information is useful to LogicaCMG as an instruction to realize demonstrations of *other* VII concepts using the framework.

Performing the demonstration

Chapter 6 and a video [55] give a detailed report of the on-road demonstration of the cooperative weighing concept. The demo was executed on the N629 of the IW, according to a predefined script (Section 5.1.2). Furthermore, evaluation criteria (Section 6.1) were defined before the demonstration.

A server, running the Service Centre application, was set-up in a lab in Delft. The wireless communication between the OBU and the server (V2I and I2V) was realized using GPRS.

A 'weigher location' was chosen on the IW. For the demo, a car drove towards this imaginary weigh point, passed it, turned around and drove back to pass it again. When the car drove passed the weigh point, the OBU displayed an alert with the (simulated) weight of the car. The Service Centre calculated the distance of the car to this weigh point. This distance was shown 'live' by the OBU to the driver.

The results of demonstration (Section 6.3) were positive according to the set evaluation criteria: the driver was informed of his vehicle's (simulated) weight within 2 seconds after his car drove passed the weigh point.

The news of this demo was received by the Province of Noord-Brabant enthusiastically and would possibly contribute to further developments.

7.2 Recommendations

The recommendations below embody the outcome of the main objective: *identify potential development and deployment issues of a vehicle infrastructure integration (VII) application by building and demonstrating a prototype.*

- **Commitment at an organizational level should be achieved.** Technology is only one dimension in the realization of cooperative systems. The organizational dimension plays a key role as well [50]. During this project the author experienced that cooperation with the Province of Noord-Brabant at an organizational level is needed indeed.

- **Privacy and security issues should be anticipated for.** When a driver (an end-user of CVIS) does not trust the system, he will not use it. Swahn and Udin [72] have looked into these two issues regarding the CVIS project. Biesecker et al. [3] present the results from an information security analysis that was based on the U.S. national ITS architecture. These findings might be interested in the CVIS context as well.

- **(Software) change should be anticipated for.** Cooperation means the exchange of information between parties, or agents. Agents cooperate in order to learn. When an agent learns, its behavior will *change* to better fit its environment. This is analogous to ITS. The human behind an OBU will act differently when he is provided with all the new information provided by the CVIS systems. Therefore it is recommended to invest in an application design which can evolve easily over time. In [50]: *'...the problem is how to build information systems which continue to share goals with their organizational environment, human users, and other existing systems as they all evolve.'*

- **Look beyond standard UML for modeling [45, 16].** For example, Kusek et al. [45] present four approaches to extend UML Sequence diagrams to capture the mobility of agents. These ideas can be adopted into the ITS context, i.e. an agent can be seen as a vehicle.

- **Reverse engineering can clarify existing road infrastructure.** Anticipate on the non-availability or poor technological quality of third-parties involved in the integration of CVIS with existing road infrastructure. Reverse engineering, down to a certain level, should play a key role in making the workings of existing road side systems clear.

- **Data exchange format should be chosen wisely.** The use of a standard meta-model is advised, especially when the data crosses organizational boundaries. Several meta-models for ITS exist [53], like DATEX II [15]. It is straightforward to use XML as the markup language, because an XML message can be validated against these meta-models.

  However, the use of XML as the markup language to exchange information between the vehicle and infrastructure could lead to performance problems if these problems are not anticipated for in the software design:

  - Regarding *computability*; the XML message must be parsed by the agent before it can be used.
  - Regarding *data size/throughput*; XML messages are generally large in size, e.g. because the language is verbose by nature.

  These problems are more likely to occur at the vehicle-side because OBUs generally have less computing power, since they have to meet portability requirements. A solution would be using an efficient parsing method and XML compression [54].

  A 'light-weight' markup language, like JSON [41], is chosen for this project because of its simplicity. It is a workaround for these mentioned problems, but JSON can not be validated against a meta-model by default.

7.3 Future work

- Instead of using a simulator that 'generates' the road-events, the IW can be integrated with the VII framework and the cooperative weighing concept. Appendix F can form the foundation for this integration process where 'live' road-events from the IW can be processed by the framework.

- The VII framework can be put into CVIS' execution environment. This environment consists of more advanced CVIS-specific OBUs, CALM communication facilities and the Java-based OSGi [61] service platform. This

CVIS environment was being developed concurrently at the time this project was undertaken and is therefore not taken into account.

A redesign of the VII framework, together with the corresponding programming effort, is necessary to make the framework fit into the eventual CVIS environment:

  – The CVIS OBUs in combination with CALM enable a P2P network topology, which is a recommended choice. Currently, the framework and its architecture use a client/server topology because of the reasons explained in Section 5.2.3.
  – The use of CALM commits to the use of IPv6, while the current VII framework/demonstrator relies on IPv4. Therefore, the framework should be adapted to use IPv6, or its IPv4 network traffic should be tunneled through an IPv6 network.
  – The framework's implementation should be made suitable for the OSGi environment.

- The scalability and robustness of the VII framework can be improved using a *distributed* publish/subscribe architecture [12, 32]. A good mobile publish/subscribe system can deal more gracefully with vehicles and infrastructure 'going offline' [32]. Additionally, the current non-distributed framework is a single point of failure. The Java Event-based Distributed Infrastructure (JEDI) system [12, 13] could be used as inspiration or a basis for such a distributed architecture.

- Ideas and research in the field of mobile ad-hoc networks (MANETs) [69] and vehicular ad-hoc networks (VANETs) [65] can be used in combination with the information presented by this thesis. For example, Rybicki [65] explains the Internet as an enabler for creating a P2P network of cars sharing traffic information.

- It is yet unknown how the drivers and other users of CVIS-applications will respond to its introduction.

  A recent study [79] executed a user need survey, which gives an indication to what extend drivers would like to be assisted by their cars when driving.

  Several studies [31, 29] indicate that the behavior of drivers changes if they use ADAS, like CTA. This change could lead to negative side effects, e.g. drivers forgetting their rules of road traffic. More research could be done to put this into the context of CVIS.

Bibliography

[1] Adapt-Mobile. Adapt-Mobile AD-900 Speedcambuster Bluetooth GPS receiver. `http://www.adapt-mobile.com/mobile/detail.php?p_id=15`, 2008.

[2] T. Biding and G. Lind. Intelligent Speed Adaptation (ISA), Results of large-scale trials in Borlnge, Lidkping, Lund and Ume during the period 1999-2002, 2002.

[3] A. Biesecker, E. Foreman, K. Jones, and B. Staples. Intelligent Transportation Systems (ITS) Information Security Analysis. Technical report, U.S. Department of Transportation, Federal Highway Administration, 1997.

[4] R. Bishop. *Intelligent Vehicle Technology and Trends*, pages 8–23, 177–223, 315–319. Artech House, 2005.

[5] R. Bossom, A. Taddei, H. Zakizadeh, S. Coutel, T. Boyer, F. Tuijnman, S. Tsegay, M. Baggen, and G. Granieri. D.CINT.3.1 Architecture and System Specifications. Technical report, Cooperative Vehicle-Infrastructure Systems, 2007.

[6] C. Britton and P. Bye. *IT Architectures and Middleware, second edition*, pages 25–26, 59–76, 77–86, 99–106, 116–117. Addison-Wesley, 2004.

[7] G. J. Brookshear. *Computer Science, an overview, sixth edition*, pages 150–158. Addison Wesley, 2000.

[8] K. Chen and J. Miles. *ITS Handbook 2000*, pages xviii, 41, 201–253, 255–359. Artech House, 1999.

[9] Commonwealth of Australia. National Road Safety Strategy 2001 - 2010. `http://www.atcouncil.gov.au/documents/atcnrss.aspx`, 2006.

[10] COOPERS. Project website. `http://www.coopers-ip.eu/`, 2007.

[11] D. Crockford. RFC 4627: The application/json Media Type for JavaScript Object Notation (JSON). Technical report, 2006.

[12] G. Cugola and H.-A. Jacobsen. Using Publish/Subscribe Middleware for Mobile Systems. In *Mobile Computing and Communications Review, Volume 6, Number 4*, pages 25–33, 2002.

[13] G. Cugola, E. D. Nitto, and A. Fuggetta. The JEDI event-based infrastructure and its application to the development of the OPSS WFMS. In *IEEE Trans. on Software Engineering, Volume 27*, pages 827–850, 2001.

[14] D-CIS Lab. Main website. http://www.decis.nl/index.php, 2008.

[15] DATEX Organization. Official DATEX Website. http://www.datex2.eu/, 2007.

[16] S. A. DeLoach and J. L. Valenzuela. An Agent-Environment Interaction Model. In *Lecture Notes in Computer Science, Agent-Oriented Software Engineering 2006*, pages 1–18, 2007.

[17] Directie Economie en Mobiliteit. Evaluatie Informatieve weg, Achtergrond-document (in Dutch). Technical report, The Dutch Province of Noord-Brabant, 2005.

[18] Directie Economie en Mobiliteit. Functionele beschrijving verschillende modules (in Dutch). Technical report, The Dutch Province of Noord-Brabant, 2005.

[19] Directie Economie en Mobiliteit. Vraagspecificatie De Informatieve weg, Provinciale weg N629 Dongen - Oosterhout (in Dutch). Technical report, The Dutch Province of Noord-Brabant, 2005.

[20] ERTICO. Network of National ITS Associations website. http://www.itsnetwork.org/, 2004.

[21] ERTICO. Cooperative Vehicle-Infrastructure Systems (CVIS) project website. http://www.cvisproject.org/, 2007.

[22] ERTICO. Main website. http://www.ertico.com/, 2007.

[23] ERTICO. PReVENT project website. http://www.prevent-ip.org/, 2007.

[24] K. L. et al. JSR 179: Location API for J2ME. http://jcp.org/en/jsr/detail?id=179, 2007.

[25] European Commission. Road Safety Action Programme website. http://ec.europa.eu/transport/roadsafety/road_safety_observatory/rsap_en.htm, 2007.

[26] Google. Google Earth Website. http://earth.google.com/, 2007.

[27] H. Harms. Notulen overleg CVIS testsite Informatieve Weg op 11 september 2007 (in Dutch). Technical report, The Dutch Province of Noord-Brabant, 2007.

[28] Hewlett-Packard Development Company, L.P. HP ProLiant DL360 G5 Server series. http://h10010.www1.hp.com/wwpc/us/en/en/WF05a/15351-15351-3328412-241644-241475-1121486.html, 2008.

[29] M. Hoedemaeker. Driving with intelligent vehicles. Driving behavior with ACC and the acceptance by individual drivers. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 506–509, 2000.

[30] S. Hoogendoorn. *Pragmatisch modelleren met UML 2.0 (in Dutch)*. Addison Wesley, 2004.

[31] M. Houtenbos, H. Jagtman, M. Hagenzieker, P. Wieringa, and A. Hale. Understanding road users expectations: an essential step for ADAS development. In *European Journal of Transport and Infrastructure Research, 5, no. 4*, pages 253–266, 2005.

[32] Y. Huang and H. Garcia-Molina. Publish/Subscribe in a Mobile Environment. In *Wireless Networks 10*, pages 643–652, 2004.

[33] INVENT. A brief supplement to the INVENT brochure describing the component project Traffic Management in Transport and Logistics VMTL. `http://www.invent-online.de/downloads/VMTL-handout-E.pdf`, 2003.

[34] INVENT. A brief supplement to the INVENT brochure describing the component project Traffic Network Equalization NIV. `http://www.invent-online.de/downloads/NIV-handout-E.pdf`, 2003.

[35] INVENT. Press Release, INVENT Final Presentation in Munich, April 28. `http://www.invent-online.de/en/downloads/INVENT-Presserelease-05-04-28.pdf`, 2005.

[36] INVENT. Project website. `http://www.invent-online.de/`, 2007.

[37] ISO TC 204 Working Group 16. Main website. `http://www.calm.hu/`, 2007.

[38] Japanese Ministry of Land, Infrastructure and Transport. Homepage of the Japanese Government ITS program. `http://www.mlit.go.jp/road/ITS/`, 2007.

[39] Japanese Ministry of Land, Infrastructure and Transport. Introduction of Japanese Smartway 2007. `http://www.its.go.jp/ITS/Smartway/SW2007/SMARTWAY2007PublicRoadTest.pdf`, 2007.

[40] Japanese Ministry of Land, Infrastructure and Transport. Smartway 2007 Public Road Test. `http://www.its.go.jp/ITS/index/indexSW2007.html`, 2007.

[41] JSON.org. Introducing JSON. `http://www.json.org/`, 2007.

[42] J. Knudsen. Networking, user experience, and threads. `http://developers.sun.com/mobility/midp/articles/threading/`, 2002.

[43] T. Kosch. Towards a Common European Communication Architecture for Cooperative Systems. In *Presentation series of the The Fully Networked Car Workshop, http://www.itu.int/ITU-T/worksem/ict-auto/200703/programme.html/*, 2007.

[44] J. F. Kurose and K. W. Ross. *Computer Networking, A Top-Down Approach Featuring the Internet*, pages 44–53. Addison Wesley, 2001.

[45] M. Kusek and G. Jezic. Extending UML Sequence Diagrams to Model Agent Mobility. In *Lecture Notes in Computer Science, Agent-Oriented Software Engineering VII*, pages 51–63, 2007.

[46] LogicaCMG Nederland B.V. Working Tomorrow website, in Dutch. http://www.workingtomorrow.nl/, 2007.

[47] LogicaCMG plc. Main company website. http://www.logicacmg.com/, 2007.

[48] Q. Mahmoud. Advanced MIDP Networking, Accessing Using Sockets and RMI from MIDP-enabled Devices. http://developers.sun.com/mobility/midp/articles/socketRMI/index.html, 2002.

[49] P. Malaterre. Interconnexion between Electronic OEM car network and outside infrastructure. In *Presentation series of the The Fully Networked Car Workshop, http://www.itu.int/ITU-T/worksem/ict-auto/200703/programme.html/*, 2007.

[50] G. D. Michelis, E. Dubois, M. Jarke, F. Matthes, J. Mylopoulos, M. Papazoglou, K. Poh, J. Schmidt, C. Woo, and E. Yu. Cooperative Information Systems: A Manifesto, 1997.

[51] J. Misener and S. Shladover. PATH Investigations in Vehicle-Roadside Cooperation and Safety: A Foundation for Safety and Vehicle-Infrastructure Integration Research. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 9–16, 2006.

[52] H. Mortiz. Geodetic Reference System 1980 (GRS80). In *Bulletin Geodesique, volume 54*, 1980.

[53] Mott MacDonald Limited. ITS Metadata Registry, Recommended Models. http://www.itsregistry.org.uk/recommended.html, 2007.

[54] W. Ng, W.-Y. Lam, and J. Cheng. Comparative Analysis of XML Compression Technologies. In *World Wide Web, Volume 9, Number 1*, pages 5–33, 2006.

[55] B. Nijssen and M. Konijn. Video recording of the cooperative weighing demonstration on the Informatieve Weg. http://nl.youtube.com/watch?v=sFNZyv0kcQY, 2008.

[56] B. Nijssen and M. Stam. Proposal for a communication architecture for the Informatieve Weg (IW). Technical report, LogicaCMG, 2007.

[57] T. Nobe. Architectural change of Vehicle IT and possibility to offer WEB2.0 type of services for networked car. In *Presentation series of the The Fully Networked Car Workshop, http://www.itu.int/ITU-T/worksem/ict-auto/200703/programme.html/*, 2007.

[58] Nokia. IP Passthrough tutorial - use your PC Internet connection from your phone. http://wiki.forum.nokia.com/index.php/IP_Passthrough_tutorial_-_use_your_PC_Internet_connection_from_your_phone, 2007.

[59] Nokia. Nokia E61 mobile device. http://europe.nokia.com/A4142101, 2007.

[60] C. E. Ortiz. Introduction to J2ME Web Services. http://developers.sun.com/mobility/apis/articles/wsa/, 2004.

[61] OSGi Alliance. Main website. http://www.osgi.org/, 2007.

[62] PostgreSQL Global Development Group. PostgreSQL main website. http://www.postgresql.org, 2007.

[63] Refractions Research. PostGIS website. http://postgis.refractions.net/, 2007.

[64] Refractions Research. PostgreSQL PostGIS Manual, Chapter 6: PostGIS Reference. http://postgis.refractions.net/docs/ch06.html#id2759577, 2007.

[65] J. Rybicki, B. Scheuermann, W. Kiess, C. Lochert, P. Fallahi, and M. Mauve. Challenge: peers on wheels - a road to new traffic information systems. In *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 215–221, New York, NY, USA, 2007. ACM.

[66] Safespot. Project website. http://www.safespot-eu.org/, 2007.

[67] S. Seedorf, A. Korthaus, and M. Aleksy. Creating a Topic Map Query Tool for Mobile Devices using J2ME and XML. In *4th International Symposium on Information and Communication Technologies*, pages 111–116, 2005.

[68] G. Seshadri. jGuru: Remote Method Invocation: Introduction. http://java.sun.com/developer/onlineTraining/rmi/, 2000.

[69] T. Sivaharan, G. Blair, A. Friday, M. Wu, H. Duran-Limon, P. Okanda, and C.-F. Sorensen. Cooperating Sentient Vehicles for Next Generation Automobiles. In *1st International Workshop on Applications of Mobile Embedded Systems (WAMES'04)*, 2004.

[70] A. Solberg, A. Schmid, M. Baggen, F. Alesiani, H. Stratil, R. Bossom, S. Forkert, M. Schlingelhof, A. Burkert, S. Haverkamp, and P. Mathias. D.CVIS.3.3 Architecture and System Specifications. Technical report, Cooperative Vehicle-Infrastructure Systems, 2007.

[71] Sun Microsystems. The Java Servlet API White Paper. http://java.sun.com/products/servlet/whitepaper.html, 2007.

[72] J. Swahn and C. Udin. Cooperative Vehicle-Infrastructure Systems: Identification, Privacy and Security. Master's thesis, Linkopings universitet, 2007.

[73] The Apache Software Foundation. Apache Tomcat website. http://tomcat.apache.org/, 2007.

[74] The Dutch Province of Noord-Brabant. Informatieve weg (Dutch for Informative Road) website. http://www.informatieveweg.nl/, 2003.

[75] S. Tsugawa. Trends and Issues in Safe Driver Assistance Systems. In *IATSS Research, Vol. 30, No. 2*, pages 6–18, 2006.

[76] United States Department of Transportation. Federal Highway Administration website. `http://safety.fhwa.dot.gov/`, 2007.

[77] U.S. Department of Transportation, Research and Innovative Technology Administration. The Vehicle Infrastructure Integration (VII) initiative. `http://www.its.dot.gov/vii`, 2007.

[78] Vagverket. TRAFIKEN.nu. `http://trafiken.nu/`, 2007.

[79] C. van Driel. *Driver Support in Congestion, An Assessment of User Needs and Impacts on Driver and Traffic Flow*. PhD thesis, University of Twente, 2007.

[80] P. van Koningsbruggen, M. Konijn, H. Za-kizadeh, A. Taddei, T. Boyer, S. Coutel, A. Penning, A. Egemalm, and I. Fraser. D.CINT.2.1 Use Cases and System Requirements. Technical report, Cooperative Vehicle-Infrastructure Systems, 2006.

[81] R. Wieringa. *Design Methods for Reactive Systems*, pages 49–57, 70–107, 133–164, 201–211, 225–232. Morgan Kaufmann, 2003.

Appendix A

Acronyms

ACC	adaptive cruise control
ADAS	advanced driver assistance systems
AVI	automatic vehicle identification
CAG	Core Architecture Group
CALM	Communication Access for Land Mobiles
CINT	Cooperative Inter-urban Applications
COTS	commercial off-the-shelf
COOPERS	Co-operative Systems for Intelligent Road Safety
CFF	Cooperative Fleet and Freight Applications
COMO	Cooperative Monitoring
CSV	comma-separated values
CTA	Cooperative Travelers Assistance
CURB	Cooperative Urban Applications
CVHS	Cooperative Vehicle Highway Systems
CVIS	Cooperative Vehicle-Infrastructure Systems
DFD	data flow diagram
DOT	Department of Transportation
DSRC	Dedicated Short Range Communication
EC	European Commission
EU	European Union
EDA	Enhanced Driver Awareness

ETC electronic toll collection

EFCD enhanced floating car data

ERM entity-relationship model

ERTICO European Road Transport Telematics Implementation Coordination Organization

FCD floating car data

GPS Global Positioning System

GPRS General Packet Radio Service

GRS Geodetic Reference System

GSM Global System for Mobile communications

GST Global Systems for Telematics

HUD head-up display

I2I infrastructure-to-infrastructure

I2V infrastructure-to-vehicle

ISA intelligent speed adaptation

ISO International Organization for Standardization

ITS Intelligent Transportation Systems

IVE in-vehicle equipment

J2SE Java Platform Standard Edition

JAD Java Application Descriptor

JDBC Java Database Connectivity

JEDI Java Event-based Distributed Infrastructure

JSON JavaScript Object Notification

JSR Java Specification Request

MANETs mobile ad-hoc networks

ME Micro Edition

MIDP Mobile Information Device Profile

MMS Multimedia Messaging Service

MTS Mobile Traffic Services

OBE on-board equipment

OBU on-board unit

OSI Open Systems Interconnection

P2P peer-to-peer

PATH Partners for Advanced Transit and Highways

QoS quality of service

RMI Remote Method Invocation

ROI return on investment

RSU road-side unit

RSAP Road Safety Action Programme

SIM Subscriber Identity Module

SMS Short Message Service

SOAP Simple Object Access Protocol (initial acronym, now simply SOAP)

SRA Swedish Road Administration

TCC traffic control centre

TMC traffic management centre

TICS Transport Information and Control Systems

UML Unified Modeling Language

UMTS Universal Mobile Telecommunications System

VANETs vehicular ad-hoc networks

V2I vehicle-to-infrastructure

V2V vehicle-to-vehicle

VII vehicle infrastructure integration

WIM weigh-in-motion

WSDL Web Services Description Language

XFCD extended floating car data

XML Extensible Markup Language

Dutch acronyms

BPS Beschrijvende Plaatsbepaling Systematiek

DSI dynamische snelheids informatie

DVM dynamisch verkeersmanagement

HWN hoofdwegennet

IW Informatieve Weg

RVM regionale verkeersmanagers

SG signaalgever

VRI verkeersregelinstallatie

UWS universeel wegkant systeem

Appendix B

Glossary

Communication Access for Land Mobiles (CALM)

The scope of CALM is to provide a Standardized set of air interface protocols and parameters for medium and long range, high speed ITS communication using one or more of several media, with multipoint and networking protocols within each media, and upper layer protocols to enable transfer between media. [37]

cooperative information systems

An information system is cooperative if it shares goals with other agents in its environment, such as other information systems, human agents and the organization itself, and contributes positively towards the fulfillment of these common goals. [50]

cooperative Intelligent Transportation Systems (ITS)

ITS where information is shared between vehicles and with the road-infrastructure to contribute positively towards the realization of common goals that lead to the improvement of road safety, road efficiency and/or driver convenience.

extended floating car data (XFCD)

Automatic vehicle-based event and traffic state detection supporting detailed, up-to-the-minute data for traffic, environment, and weather information. [36]

floating car data (FCD)

Also known as 'vehicle probing'. *A method for mobile acquisition of current traffic data using individual probe vehicles as sensors. When a vehicle enters jammed or congested traffic, it can send an automatic report (including its position and the traffic situation encountered) by mobile communication to an evaluation center.* [36]

in-vehicle signing

The provision of road-sign information (like speed limits, road closures etc.) to a driver.

Intelligent Transportation Systems (ITS)

Synonymous with Transport Information and Control Systems (TICS), also known as Intelligent *Transport* Systems. *ITS comprise a wide range of novel tools for managing transport networks, as well as services for travelers. Also called 'Transport Telematics,' ITS tools are based on three core features: information, communications, and integration. The collection, processing, integration and supply of information are the heart of ITS. Whether offering real-time information about currect traffic conditions for a network, or on-line information for journey planning, ITS tools enable authorities, operators, and individual travelers to make better informed, more coordinated, and more intelligent decisions.* [8]

on-board unit (OBU)

Synonymous with on-board equipment (OBE) and in-vehicle equipment (IVE). *ITS equipment situated in a vehicle.* [8]

platooning

A group of vehicles operating under closely coupled vehicle follower longitudinal control. The close coupling is achieved by communication among the member vehicles regarding information about vehicle movements. [8]

vehicle infrastructure integration (VII)

The integration of vehicle- and infrastructure systems by means of wireless communications. In this Thesis, VII refers to ITS concepts that are based on both V2I and I2V communication. Some research papers [51] use the term for ITS concepts based on V2I only.

Appendix C

IW log examples

This appendix shows two log fragments of the Informatieve Weg (IW). These logs were provided by the Province of Noord-Brabant and stored in comma-separated values (CSV)-files. One file for each SG. The CSV-columns have the following meaning:

1. *UWS_Id*, the identifier of the UWS;
2. *UWS_Name*, the name of the UWS together with its location;
3. *Log_Time*, local time when the event occurred;
4. *Log_Id*, unknown;
5. *Log_Text*, description of the event;
6. *Value_1*, *Value_2* and *Value_3* are customizable values;

C.1 Signal provider SG-N629 4.454 L

```
"1";"SG-N629 4.454 L";"28-06-2007 11:57:50";"2342";"Hulpverleningsvoertuig gedetecteerd";"";"E1F1FM13SG01";""
"1";"SG-N629 4.454 L";"28-06-2007 11:57:50";"2212";"Beeld aanvraag door Module";"13";"10";""
"1";"SG-N629 4.454 L";"28-06-2007 11:57:50";"2213";"Beeld op Signaalgever";"13";"10";","
"1";"SG-N629 4.454 L";"28-06-2007 11:57:56";"2299";"melding 2299";"Input failed; ioret = 0";"...";"..."
"1";"SG-N629 4.454 L";"28-06-2007 11:57:59";"2212";"Beeld aanvraag door Module";"13";"0";""
"1";"SG-N629 4.454 L";"28-06-2007 11:57:59";"2213";"Beeld op Signaalgever";"0";"0";","
"1";"SG-N629 4.454 L";"28-06-2007 11:57:59";"2212";"Beeld aanvraag door Module";"Requestguidance";"13";"False"
"1";"SG-N629 4.454 L";"28-06-2007 15:23:24";"2206";"Niet aanwezig, RFU";"Communicatie, level 1";"1";"8"
"1";"SG-N629 4.454 L";"28-06-2007 16:53:38";"2206";"Niet aanwezig, RFU";"Communicatie, level 1";"1";"8"
```

C.2 Signal provider SG-N629 3.746 L

```
"3";"SG-N629 3.746 L";"25-06-2007 10:23:29";"2343";"Cyclus-detectie Overflow";"";"E2F1FM04SG03";""
"3";"SG-N629 3.746 L";"25-06-2007 10:23:29";"2336";"Fiets gedetecteerd";"86310914";"E2F1FM04SG03";""
"3";"SG-N629 3.746 L";"25-06-2007 10:23:31";"2335";"Snelheidsadvies";"";"E2F1FM12SG03";"55"
"3";"SG-N629 3.746 L";"25-06-2007 10:23:39";"2212";"Beeld aanvraag door Module";"4";"0";""
"3";"SG-N629 3.746 L";"25-06-2007 10:23:42";"2335";"Snelheidsadvies";"";"E2F1FM12SG03";"60"
"3";"SG-N629 3.746 L";"25-06-2007 10:23:49";"2212";"Beeld aanvraag door Module";"12";"12";""
"3";"SG-N629 3.746 L";"25-06-2007 10:23:49";"2213";"Beeld op Signaalgever";"12";"12";","
```

Appendix D

IW requirements

This appendix summarizes several initial requirements of the Informatieve Weg (IW) that might be relevant for its future integration with CVIS applications.

The province of Noord-Brabant defined requirements used as input for their (technical) contractors [19]. The assumption is made that these requirements are implemented in the current IW.

A subset of these requirements indicate relevant facts that should be taken into account when realizing a CVIS-application on the IW. These are repeated below, together with their corresponding requirement IDs, as defined in [19].

1.10.15 When the power of a SG is shut off, it can display an 'out of order' message for max. 30 minutes.
1.12.04 The loops can detect the amount of vehicles, the vehicle-speed and can categorize the vehicle into five categories. (applicable to **module 11**)
1.14.01 Emergency vehicles can be detected by use of audio recognition or by an OBU sending a special signal to the infrastructure. (**module 13**)
1.15.05/06 A road side cabinet (containing an UWS) should have an extra 10 'height units' (19 inch wide) physical space and 16A of extra current available.
1.15.19 A nameplate is mounted at the outside of a cabinet / UWS. The name has the form 'UWS N629 hm L/R'. hm is a hectometer indication with three decimals behind the comma, like 102.000 or 9.275. L/R is an indication of left or right according to the Dutch Beschrijvende Plaatsbepaling Systematiek (BPS) standard.
1.16.02/03 A standard headless[1] PC is used for the UWS and it runs a 'traditional' operating system.
1.16.04 GPRS or equal is used for wireless communication. E.g. between UWS and the central system.
1.16.08/09 A system-log is kept and can be accessed by GPRS. *See* Appendix C for log examples.
1.16.15 An administrator can check the actual status of an UWS through the network connection. Thus by means of GPRS, see 1.16.04.
1.17.01 A central system collects all the logs and provides the ability to control

[1]Without keyboard and mouse.

all connected UWSs.

1.17.05 The central system is connected to the UWSs by use of wireless communication.

Appendix E

Implementation details

This appendix elaborates on the implementation details of the demonstrator in the Java programming language. The UML [30] notation is used to clarify certain technical aspects.

E.1 VII framework

This section provides the implementation details of the generic VII framework that was described in Section 5.2.

Sun's Java Platform was used to implement the framework. The Micro Edition (ME)[1] of this Platform implements the Mobile Information Device Profile (MIDP) 2.0 and was used on the OBU, i.e. a smartphone. The Java Platform Standard Edition (J2SE) was used at the server side.

E.1.1 Component decomposition

The framework consists of several components. The component diagram of Figure E.1 gives an overview. It is a formalization of Figure 5.5 on page 46 and shows the interfaces that hookup the components more explicitly.

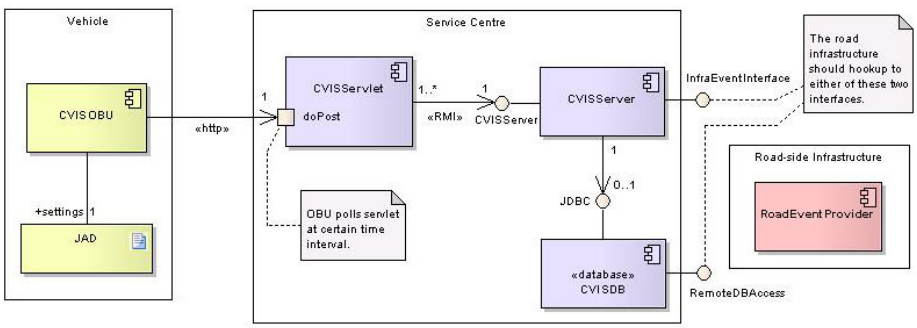

Figure E.1: UML component diagram of the generic VII framework.

[1]Formally known as J2ME.

The CVISOBU polls a CVISServlet at the Service Centre periodically and information is exchanged. The CVISServlet in turn talks to the CVISServer interface with RMI [68].

The road-side infrastructure - including the Informatieve Weg (IW) - can provide road event data to either the EventInterface or the RemoteDBAccess interface, provided by the CVISServer or CVISDB respectively. More details on the IW with the framework are explained in Section F.

E.1.2 On-Board Unit: CVISOBU

The CVISOBU application is a 'MIDlet'[2] running on a Nokia E61 smartphone [59]. MIDlets are described by a Java Application Descriptor (JAD)-file. The JAD of the CVISOBU is shown below.

```
1  MIDlet-1: CVIS OBU, /cvis_icon.png, com.logicacmg.CVIS.demonstrator.obu.CVISOBU
2  MIDlet-Jar-URL: CVISOBU.jar
3  MIDlet-Icon: /cvis_icon.png
4  MicroEdition-Configuration: CLDC-1.1
5  MIDlet-Version: 1.0.0
6  MIDlet-Name: CVIS OBU
7  MIDlet-Vendor: LogicaCMG
8  MIDlet-Permissions: javax.microedition.location.Location,
9      javax.microedition.location.Orientation,
10     javax.microedition.io.Connector.http, javax.microedition.io.PushRegistry
11 MicroEdition-Profile: MIDP-2.0
12 obuID: 1
13 tccURL: http://localhost:8080/CVISServlet/CVISServlet
14 pollInterval: 2000
15 gpsUpdateInterval: 1
```

It is important to set proper MIDlet-Permissions (lines 8-10) to be able to use a certain Java Specification Request (JSR), like no. 179 [24], which provides GPS location information. Lines 12-15 are property values that are used as settings by the CVISOBU. For example, the tccURL (line 13) describes the location of the CVISServlet.

Figure E.2 on page 83 shows the class diagram of the CVISOBU. The main-class CVISOBU kicks off the application and performs the following:

1. It registers a GPSLocationListener, which updates its GPS-data periodically[3] (as set by the gpsUpdateInterval property). This data, together with the obuID is put in a PollMessage. This message is serialized using the org.json.me package [41], which is specifically designed to run on the Java ME platform.

2. It schedules a Poller class, which runs a PollerThread periodically (as set by pollInterval property).

3. It delegates the MainCanvas to display the GPS data and shows pop-up alerts to the driver.

[2]A Java Platform ME application using the MIDP implementation is called a MIDlet.
[3]This is provided by JSR 179 [24].

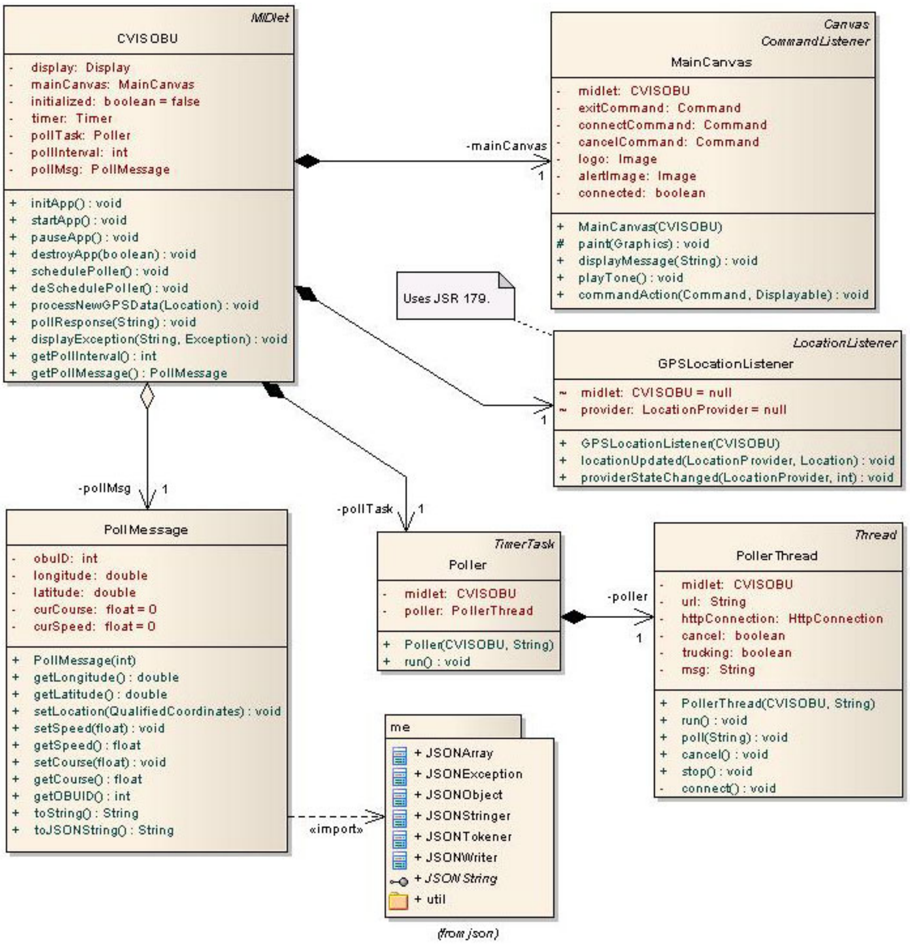

Figure E.2: UML class diagram of the `CVISOBU` MIDlet.

The initial `CVISOBU` implementation instantiated a `PollerThread` over and over for each poll (*see* left hand side of Figure E.3, the two vertical thin lines). This led to the tasks not being performed at all on the Nokia E61.

Knudsen [42] showed a clean way of implementing threads in MIDlets. This led to a new implementation of the `PollerThread` (right hand side of Figure E.3). This implementation instantiates the `PollerThread` only *once* (thin vertical line), it polls, waits for a fixed period and polls again (thicker two vertical lines). This solved the problem on the Nokia E61.

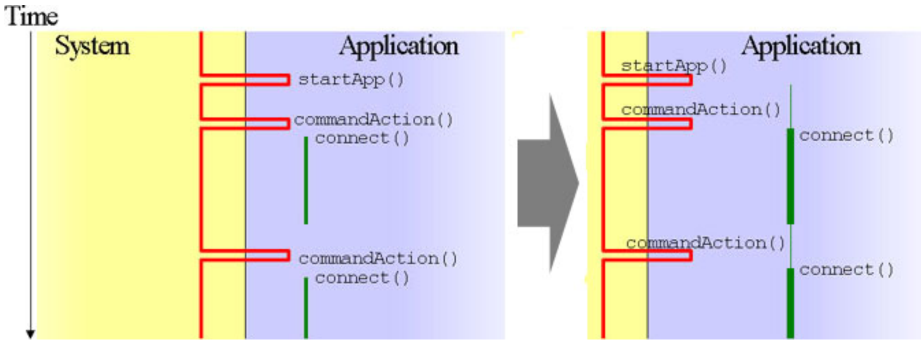

Figure E.3: A workaround for a threading problem for the Nokia E61, in [42].

E.1.3 Service Centre: `CVISServlet` and `CVISServer`

The `CVISServlet` is a simple Java Servlet that waits for a HTTP POST and calls corresponding methods of the `CVISServer` interface. It serves as a 'middleman' (explained in Section 5.2.3 on page 45).

The class diagram of the `CVISServer` is shown in Figure E.4 on page 85. The interfaces `CVISServer` and `InfraEventInterface` are shown at the top of the figure and connect the `CVISServer` with the `CVISServlet` and the road-side infrastructure, respectively.

The current implementation[4] handles the events from the IW-simulator with the Servlet, through the `CVISServer` interface. Ideally, these events should pass the `InfraEventInterface` instead to achieve a clear separation of concerns. This interface was not implemented because it depends on which middleware technique will be used to connect with the road-side infrastructure (*see* Appendix F).

The main-class `CVISServerImpl` starts the application. It uses the `org.json` package [41] to decode the poll-messages, sent by the OBU. A `VehicleManager`[5] keeps track of (connected) `Vehicles`, their location, speed and course.

An infrastructure event, like 'a bicycle crossing the road' is stored in an `InfraEvent`, which is managed by a `InfraEventFilter`. This 'filter' provides search-facilities, e.g. the `getEventClosestTo()` method returns the `InfraEvent` originating from a certain infrastructure 'software module' and that occurred

[4]The current implementation connects with an IW-simulator by use of a HTTP POST (explained in Section F.3.2). More about the simulator setup can be found in Section 5.2.5.

[5]Implements the second requirement listed in Section 5.2 on page 43.

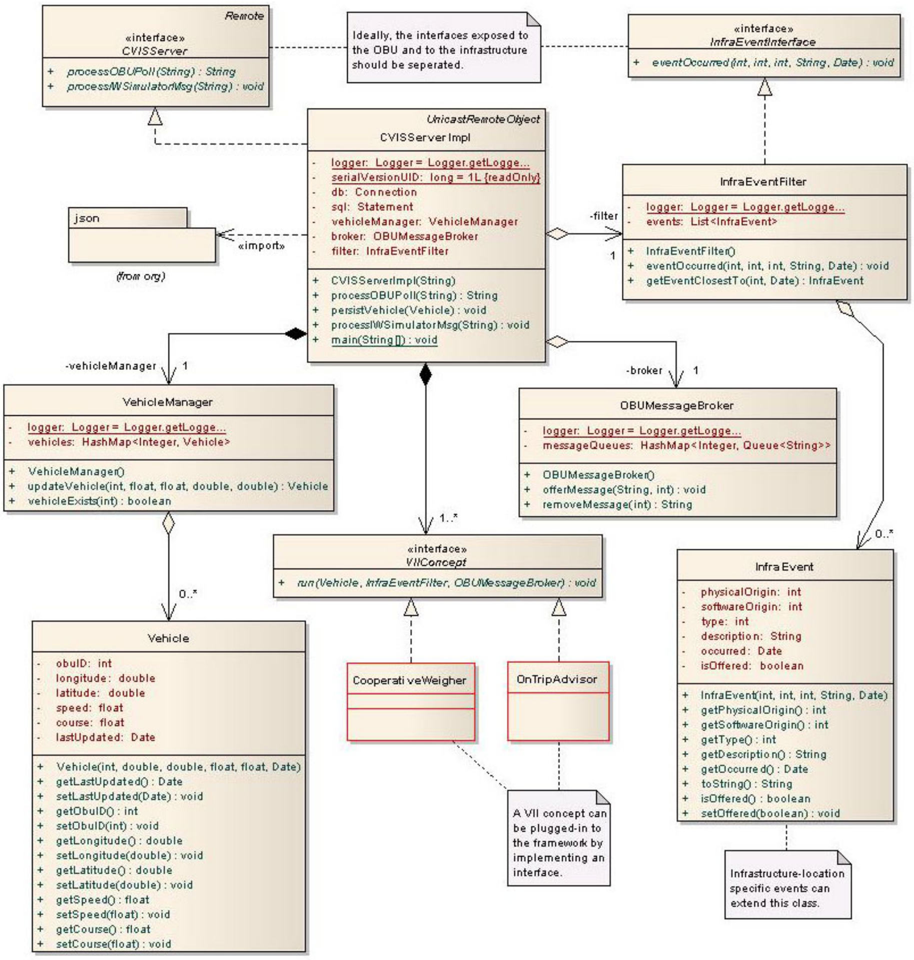

Figure E.4: UML class diagram of the `CVISServer`.

closest to a certain point in time. This event-information can than be used by a VII concept.

A VII concept, like the ones listed on page 36, can be plugged into the `CVISServer` component relatively easily. The `VIIConcept` interface forms the boundary[6] between the framework and the implemented concept.

The `CooperativeWeigher` and `OnTripAdvisor` implement the `VIIConcept` interface and examples of implemented VII concepts. The `CVISServerImpl` runs a `VIIConcept` when a OBU-poll is received.

The `VIIConcept` receives the `Vehicle` object that corresponds with the OBU sending the poll. Additionally, it receives references to the `InfraEventFilter` and `OBUMessageBroker`. The `InfraEventFilter` is used by the `VIIConcept` to lookup `InfraEvents` that match with the `Vehicle` object-state. The `OBUMessageBroker` is called to send the calculated response message back to the OBU.

E.1.4 Message provisioning behavior

Figure E.5 shows the behavior of the framework, i.e. the sequence of events:

1. The framework receives road-events,
2. poll-messages are sent by the OBU and received by the server,
3. the server calculates a poll-message response (by 'running' VII concepts),
4. the response is returned to the OBU and displayed to the driver.

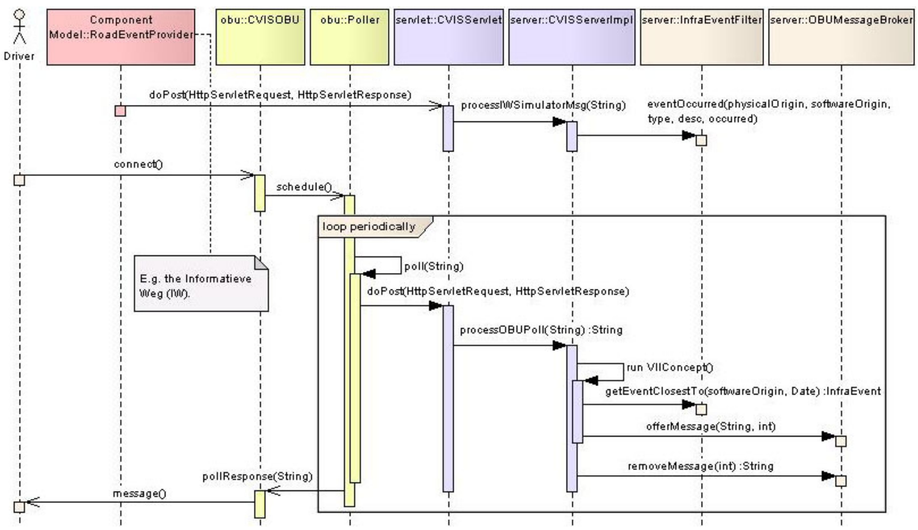

Figure E.5: UML sequence diagram showing the provision of messages to OBUs.

[6]This boundary is indicated by the dotted rectangles in Figure 5.5 on page 46.

E.2 Cooperative weighing

The the cooperative weighing concept is embodied in the `CooperativeWeigher` class, which implements the `VIIConcept` interface. It uses several facilities of the VII framework (*see* Figure E.6):

- A `Vehicle` object contains the state-information of the 'calling vehicle';
- The `InfraEventFilter` is used to retrieve `InfraEvents`, of which the weight-measurements are extracted;
- The `OBUMessageBroker` is used to send the weight-message to the vehicle.

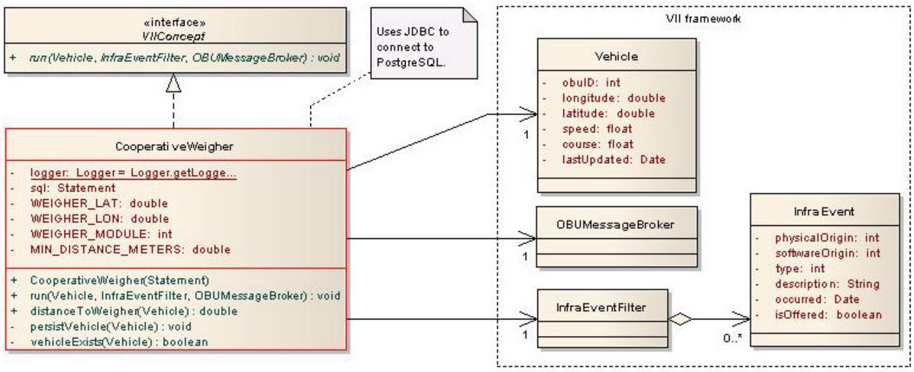

Figure E.6: UML class diagram of the `CooperativeWeigher`.

The fixed location of the weigher is set by the `WEIGHER_LAT` and `WEIGHER_LON` constant variables. The vehicle detection radius is set by `MIN_DISTANCE_METERS`. The `WEIGHER_MODULE`[7] is used to only filter those `InfraEvent` objects that are applicable for the concept (a match is found with the `softwareOrigin` of `InfraEvents`). The `CooperativeWeigher` uses JDBC to connect to a PostgreSQL [62] database with PostGIS [63] extensions.

E.2.1 Data model

The database only keeps track of the latest position, speed and course of OBUs. Every time the `CooperativeWeigher` is ran, this information is updated. The data model is fairly simple and consists of one `obu` table, which is created with the following SQL-statement:

```
CREATE TABLE obu(
  id int8 NOT NULL,
  "location" geometry,
  speed float4,
  course float4,
  changed timestamp,
  latitude float8,
  longitude float8
) WITHOUT OIDS;
```

[7]The ID of the *vehicle weight* IW module (9).

The `geometry` column contains GIS-encoded latitude/longitude points. The `CooperativeWeigher` stores the `Vehicle` object in the `obu` table using the `persistVehicle()` method. SQL `INSERT` statements, like the following, are executed by `persistVehicle()`:

```
INSERT INTO obu (id, location, latitude, longitude, speed, course, changed)
  VALUES(
    1, GeomFromText('POINT(51.626595 51.626595)'),
    51.626595, 51.626595, 0, 0, CURRENT_TIMESTAMP
);
```

The example dataset of Figure E.7 shows the state-information of two vehicles.

id int8	location geometry	speed float4	course float4	changed timestamp	latitude float8	longitude float8
1	0101000000276BD44334D04940FD14C78157AB1340	0	0	2007-11-23 11:54:21.709	51.626595	51.626595
3	0101000000CDCB1DAA72D14940915E746FD9991340	0	0	2007-11-28 11:56:22.343	51.6363117833604	51.6363117833604

Figure E.7: Example of state-information, stored in the `obu` table.

E.2.2 Calculating geographical distance

The `CooperativeWeigher` calculates the distance between the current location of the `Vehicle` and the fixed weigher location. This to determine if a vehicle is 'in range'[8] and the measured weight has to be send. The actual calculation is performed by the database.

The PostGIS extension of PostgreSQL provides two functions to calculate a linear distance between two latitude/longitude points [64]. The `distance_sphere()` function calculates the distance on a sphere, which is symmetrical. The `distance_spheroid()` provides a more accurate calculation because it more resembles the Earth's shape. Figure E.8 shows the difference between the two shapes.

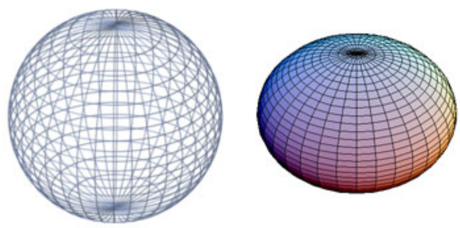

Figure E.8: A sphere and a speroid.

This SQL-statement calculates the distance of vehicle 1 to the weigher:

```
1 SELECT distance_spheroid(
2   (SELECT location FROM obu WHERE id = 1),
3   GeomFromText('POINT(51.631977 4.909472)'),
4   'SPHEROID["GRS_1980", 6378137, 298.257222101]'
5 ) as distance
```

Line 4 defines the shape of the Earth according to the Geodetic Reference System (GRS)-80 [52], which is a standard in the field of geodesy.

[8]Set by the `MIN_DISTANCE_METERS` constant.

Appendix F

IW integration proposal

This appendix gives suggestions on how to integrate the systems of the Informatieve Weg (IW) with the Service Centre of the framework in order to realize vehicle infrastructure integration (VII).

F.1 Introduction

Cooperation with the Province of Noord-Brabant has been achieved on an organizational level to some extend during this project. The Province is willing to look into the possibilities to integrate the IW with a Service Centre for CVIS. This cooperation is necessary to be able to deploy a cooperative ITS concept on the IW.

The author defined a **communication architecture proposal** document [56] to gain the technical focus of the Province. A summary of this document is given in this section. Herein, three middleware alternatives are described to get IW-log and event data that is to be processed by a CVIS Service Centre at LogicaCMG.

First of all the document is used as a starting point to intensify cooperation between LogicaCMG and the Province of Noord-Brabant within the context of CVIS. Furthermore, it explains LogicaCMG's test-site developments to other CVIS partners, therefore some IW-details (comparable with Section 4.2) are explained in the document as well.

F.2 Elements of the communication-link

A communication-link has to be realized in order to demonstrate a concept that is integrated with the IW. The big arrow in Figure F.1 on page 90 represents the *one-way* communication flow between the IW and the Service Centre.

The IW consists of a Central System and several UWSs, as explained before in Section 4.2. The UWSs keep a local log file, which contains time stamped event data, e.g. 'a bicycle is detected' or 'a traffic light went out of order'. *See* Appendix C for examples. Each UWS sends log updates to the Central System periodically. This is a *deferrable* task [6], which means that the system log will

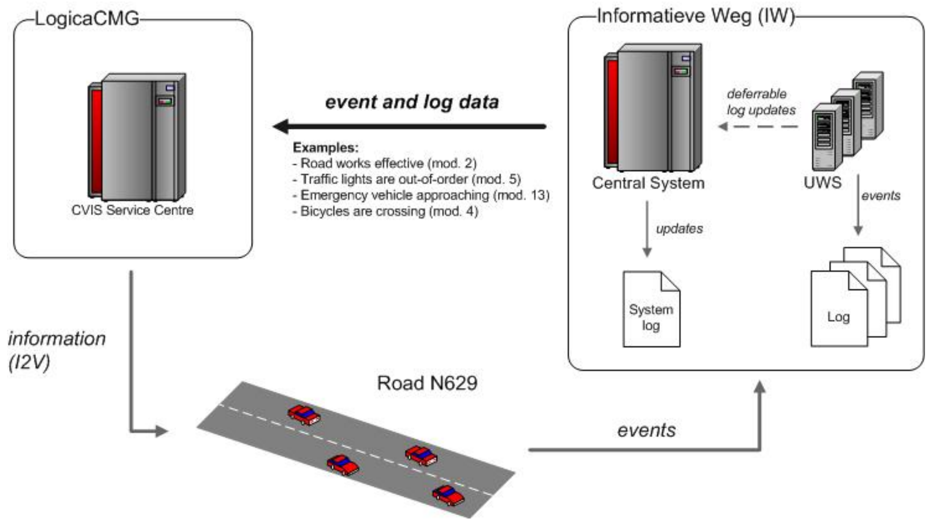

Figure F.1: Information-flow between the IW and LogicaCMG, in [56].

contain the logged events of the 'local' UWSs eventually and not in real-time.

Three considerations for achieving the communication link are:

- A **choice has to be made** to communicate with the Central System or the UWSs individually. The CVIS Service Centre can be provided with more real-time data when the communication is established with the UWSs, because of the 'deferrable log-task' described before. However, any data aggregation done by the Central System can not be utilized and should occur at the Service Centre.

- The **Internet will play an obvious role** in the architecture because it is the most easily accessible communication bearer for both parties. Security- and availability requirements have to be considered, especially for Internet communications. However, these requirements are less important for a pure demonstration.

- Communication can be either initiated by the IW or the CVIS Service Centre. Event-data has to be sent *from* the IW *to* the Centre. Therefore it would be better if **the IW acts as the initiator** of communication. Less ideal would be that the Centre has to periodically pull (and thus act as initiator) new event information from the IW. This 'pull-based' approach would lead to resource contention at the IW-side, network overload and congestion [12].

- A standardized **data format should be agreed upon**. Several XML meta-data models exist [53] that are suitable in an ITS context. For example, DATEX II [15] is suitable especially for traffic data exchange across organizational boundaries [53].

F.3 Middleware alternatives

Three middleware solutions are proposed to the Province to establish the communication link, as shown above in Figure F.1.

F.3.1 Remote database access

Either party will provide **remote database access** [6] to the other party. Security can be restricted to the other party at different levels of granularity. I.e. a party does not have to 'open' the whole database for the other party, it can also grant permission to show a specific set of database tables to the other.

It is preferred that the IW initiates the connection, here the CVIS Service Centre provides database access to the IW systems. The IW has to send `SQL INSERT` or `UPDATE` statements (containing timestamped events) to a database at the Centre. *See* Figure F.2.

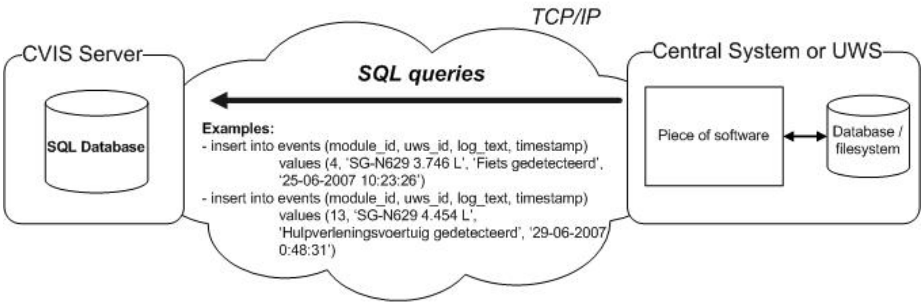

Figure F.2: Remote database access setup with the IW as initiator, in [56].

When the Centre is the initiator, it has to pull data periodically from the IW systems to check if 'something new happened' (with `SQL SELECT` statements). This will cause a lot of network overhead. However, this overhead can be minimized if the IW-database contains rows that are timestamped, then the Centre can query only the 'new' rows/events.

F.3.2 Simple push architecture

This alternative is most useful when the IW initiates. It represents a **simple push architecture**, which is ideal to broadcast changes of state [6]. I.e. the IW's 'software state' changes when a bicycle is detected by a detection loop.

A piece of software, residing on the Central System or the UWSs, will send an `HTTP POST` message to the CVIS Service Centre. *See* Figure F.3 on page 92.

This alternative is relatively easy to implement, but is by default less secure. For example, a 'man-in-the-middle' can easily modify or reroute the message.

F.3.3 Publish/subscribe topology using web services

A more ideal situation is that the IW is loosely coupled to the Service Centre where the IW does not have to know the existence of a Service Centre. Here, the Service Centre subscribes to receive event updates from specific IW software modules.

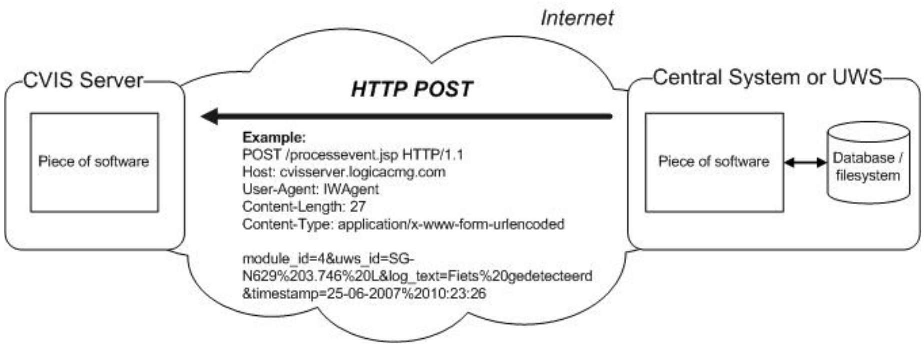

Figure F.3: Push of an `HTTP POST` message, in [56].

Figure F.4 shows the server subscribing for IW module 2. As soon as an event happens at the side of the IW, this event information is send to the CVIS Service Centre. The publish/subscribe topology can be easily extended. That is, other parties (besides CVIS) can subscribe to get information on IW events as well.

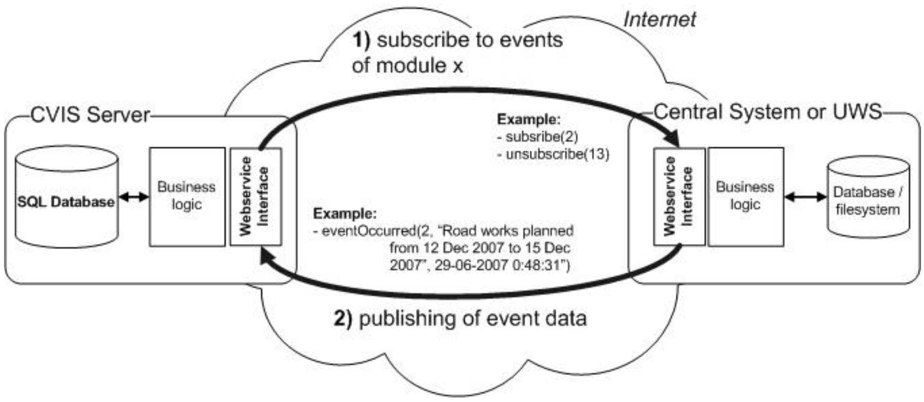

Figure F.4: A publish/subscribe topology using web services, in [56].

Web services [6] are used to establish the communication. For this purpose, an interface[1] has to be defined at the side of the Centre and at the side of the IW. This is shown by the figure.

[1]Described by a WSDL-file [6].

F.4 Comparison

Table F.1 gives a rough comparison of the three middleware alternatives and shows the responsibilities of the Province and LogicaCMG to make the communication link (Figure F.1) possible.

	Remote DB access	**HTTP POST**	**Web services**
Ease of implementation	++	+	-
De-coupling	–	+	++
Level of security	medium	low	medium-high
Network / CPU load	IW initiates: low, LCMG initiates: average-high	low	average
IW-responsibility	Provide the ability to access their databases or certain tables through TCP/IP or create a piece of software that sends `SQL INSERT / UPDATE` queries when an event occurs.	Create a piece of software that sends an `HTTP POST` message to the Centre when an event occurs.	Create a software module that exposes an interface providing a publish/subscribe concept.
Centre-responsibility	Idem + realize I2V communication.	Create software to receive the message and do further processing + realize I2V communication.	Create software that subscribes to certain IW event-publications and handles those events accordingly + realize I2V communication.

Table F.1: Comparison of middleware alternatives, in [56].

It can be concluded that HTTP POST is well-to-do, while the web services alternative is more difficult to implement. This extra difficulty pays off in terms of more security and results in a better decoupled architecture. I.e. the architectural components are more independent of each-other.

Technical implementation effort has to be undertaken at either side for all cases to achieve cooperation.