The Pull Paradigm: Foundations of User-Centric Advanced Driver Assistance Systems Based on Bidirectional Car2X Communication

Dissertation
zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
der Naturwissenschaftlich-Technischen Fakultät I der Universität des Saarlandes

vorgelegt von
Sandro Castronovo

Saarbrücken, 25.11.2013

Datum des Kolloquiums:
20.12.2013

Dekan:
Prof. Dr. Mark Groves

Vorsitzender der Prüfungskommission:
Prof. Dr. Jörg Hoffmann

Gutachter:
Prof. Dr. Dr. h.c. mult. Wolfgang Wahlster
Prof. Dr. Philipp Slusallek

Akademischer Beisitzer:
Dr.-Ing. Christian Müller

Danksagung

Diese Arbeit wäre ohne eine Reihe von Personen nicht möglich gewesen. Ich möchte mich deshalb an dieser Stelle herzlich für die jeweilige Unterstützung bedanken bei:

- Herrn Wahlster, der ungeachtet seines eigenen Arbeitspensums Freizeit und Wochenenden für mich geopfert hat. Das ist bei weitem nicht selbstverständlich. Er gab mir die Möglichkeit in den vergangenen Jahren an einem spannenden und herausfordernden Thema zu forschen.
- Christian, der es wie kein anderer verstanden hat Synergien zwischen Projektarbeit und den Zielen der Dissertation zu schaffen. Die Unterstützung, die zahlreichen Diskussionen und Ratschläge in den vergangenen Jahren sind immer wertvoll gewesen.
- meinen Kollegen am DFKI, insbesondere den Mitgliedern des Automotive-Teams mit denen die Arbeit viel Freude macht.
- den Kollegen aus dem Projekt sim^{TD}, mit denen ich alle Jahreszeiten auf dem Testgelände in Friedberg erlebt habe um das System fit für den Feldtest zu machen.
- meinen Bachelor bzw. Master Studenten, deren Arbeiten in diese Dissertation mit eingeflossen sind: Björn Kunz, Tobias Del Fabro und Nils Schnabel.
- Monika Mitrevska, die im Rahmen meines Software Campus Projekts eine unverzichtbare Hilfe ist und immer für den nötigen Vitaminnachschub im Büro sorgt.
- meiner Frau Angela, die trotz ihrer eigenen Dissertation immer ein offenes Ohr für mich hatte.
- meiner Familie, die immer für mich da war. Ich freue mich darauf, dass wir uns nun öfter sehen als in der letzten Zeit.
- Margaret De Lap, die als Muttersprachlerin meinem Englisch den nötigen Feinschliff gegeben hat.

Short Abstract

This thesis develops applications for vehicular ad-hoc networks that go far beyond the currently established areas of driving safety and traffic efficiency. The ad-hoc network is regarded as a dynamic information resource which is available to any vehicle at any time. In contrast to current state-of-the-art research, the proposed Pull Paradigm starts at the user's vehicle rather than at an information source somewhere in the network, e.g. a braking car. To access information from highly dynamic ad-hoc networks, bidirectional communication and information discovery and retrieval play a vital role. Therefore, in the course of the work, the applicability of the Pull Paradigm to established vehicular ad-hoc networks is thoroughly examined and missing aspects are identified. It turns out that a number of enhancements to almost all layers of the network stack are necessary in order to apply the Pull Paradigm using existing technology. The central elements here are two novel algorithms for managing information flow and dissemination in ad-hoc networks, which are at first formulated from the abstract perspective of graph theory. Using the knowledge gained leads to the development of PADE, a platform that supports development of vehicular ad-hoc network applications. The designed algorithms are then implemented as a routing scheme, integrated and evaluated in large, simulated city scenarios. Furthermore, PADE combines "real" and simulated communication technologies and abstracts from them, so that applications can be transferred from the lab into a test vehicle with minimal effort. In order to achieve this ambitious goal, PADE builds on a number of existing simulation and communication technologies. The practical applicability of the Pull approach is shown in two demonstrators that are integrated into a BMW 5 series test vehicle. The presentation module of the PADE platform was tested in the currently largest field operational test for vehicular ad-hoc communication. Over 400 drivers in 120 vehicles experienced the system on a daily basis.

Kurzzusammenfassung

In dieser Doktorarbeit werden Anwendungen für Fahrzeug Ad-hoc Netzwerke erarbeitet, die weit über die derzeit etablierten Bereiche der Fahrsicherheit und Verkehrseffizienz hinausgehen. Das Ad-hoc Netzwerk wird dabei als dynamische Informationsressource angesehen, die jedem Fahrzeug zu jedem Zeitpunkt zur Verfügung steht. Im Gegensatz zum derzeitigen Stand der Forschung geht das vorgestellte Pull Paradigma vom Fahrzeug des Benutzers und nicht von der Informationsquelle aus, z.B. einem bremsenden Fahrzeug. Für den Zugriff auf Informationen aus hochdynamischen Ad-hoc Netzen, spielen bidirektionale Kommunikation, Informationssuche und -rücktransport eine entscheidende Rolle. Im Verlauf der Arbeit wird deshalb die Anwendbarkeit des Pull Paradigmas auf etablierte Fahrzeug Ad-hoc Netze untersucht und fehlende Aspekte identifiziert. Es zeigt sich, dass eine Reihe an Erweiterungen auf fast allen Ebenen des Netzwerkstapels nötig sind damit die bestehende Technologie um das Pull Paradigma erweitert werden kann. Zentraler Punkt hierbei sind zwei neuartige Algorithmen zur Informationsverwaltung und -verbreitung in Ad-hoc Netzwerken die zunächst abstrakt aus Sicht der Graphentheorie formuliert werden. Mit Hilfe der gewonnenen Erkenntnisse wird PADE, eine Plattform zur Entwicklung von Anwendungen für Fahrzeug Ad-hoc Netze, entwickelt. Die entworfenen Algorithmen werden dann als Routingverfahren im Netzwerkstapel realisiert, in diesen integriert und auf großflächigen Stadtszenarien im Simulator evaluiert. Des Weiteren vereint PADE "echte" und simulierte Kommunikationstechnologien und abstrahiert von diesen, sodass Anwendungen mit minimalem Aufwand vom Labor in ein Testfahrzeug überführt werden können. Um dieses ambitionierte Ziel zu erreichen, wird auf einer Reihe bereits bestehender Simulations- und Kommunikationstechnologien aufgebaut. Die praktische Anwendbarkeit des Pull Paradigmas wird anschließend in zwei Demonstratoren implementiert und in ein BMW 5er Testfahrzeug integriert. Das Präsentationsmodul der PADE Plattform wurde im derzeit weltgrößten Feldversuch für Fahrzeug Ad-hoc Kommunikation von über 400 Fahrern in 120 Fahrzeugen im Alltag getestet.

Contents

List of Abbreviations

sim^{TD} Sichere Intelligente Mobilität Testfeld Deutschland

A-STAR Anchor-Based Street and Traffic Aware Routing

ABS Antilock Braking System

ACC Adaptive Cruise Control

ACW Average Circuit Weight

ADAS Active Driver Assistance Systems

AHP Analytic Hierarchical Process

AODV Ad-Hoc On-Demand Distance Vector Routing

AU Application Unit

AUML Automotive User Interface Markup Language

BSA Basic Set of Applications

C2C CC Car2Car Communication Consortium

C2X PADE Car2X Platform for Development and Evaluation

CAM Common Awareness Message

CAR Connectivity-Aware Routing

CDR Circuit Discovery Ratio

CR Consistency Ratio

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

DENM Decentralized Environmental Notification Message

DSR Dynamic Source Routing

DSRC Dedicated Short Range Communication

DSSSP Dynamic Single-Source Shortest Path

EC European Commission

eCall Emergency Call

ED Environmental Display

EGO vehicle The car of the driver who is using a particular ADAS

ESP Electronic Stability Control

ETSI European Telecommunications Standards Institute

FFRDV Fastest-Ferry Routing in DTN-enabled Vehicular Ad-hoc Networks

FOT Field Operational Test

GPCR Geographic Routing in City Scenarios

GPSR-MA Global Perimeter Stateless Routing with Movement Awareness

GSR Geographic Source Routing

GyTAR Greedy Traffic Aware Routing Protocol

HLA High Level Architecture

HMI Human-Machine Interface

HSDPA High-Speed Downlink Packet Access

HUD Head-Up Display

IEEE Institute of Electrical and Electronics Engineers

iTetris Integrated Wireless and Traffic Platform for Real-Time Road Traffic Management Solutions

ITSSG Intelligent Transport Systems Steering Group

IVIS In-Vehicle Information Systems

JiST Java in Simulation Time

LDM Local Dynamic Map

LGHS Linked-Graphs Heuristic Search

LGHS-CD Linked-Graphs Heuristic Search with Circuit Discovery

LTE Long Term Evolution

MANET Mobile Ad-Hoc Network

MORA Movement-Based Routing Algorithm for Vehicle Ad Hoc Networks

MSIECV Multiple Simulator Interlinking Environment

NVV Number of Visited Vertices

OBU On-board Unit

OEM Original Equipment Manufacturer

OLSR Optimized Link State Routing Protocol

OSI Open Systems Interconnection

PCI Protocol Control Information

PDR Path Discovery Ratio

PDT Path Discovery Time

PDU Packet Data Unit

POI Point Of Interest

REACT Routing Protocol for Emergency Applications in Car-to-Car Networks using Trajectories

REM Requested Environmental Message

RSS Road Side Station

RSU Road Side Unit

SDU Service Data Unit

simITS Simulation for Intelligent Transportation Systems

SML Scenario Markup Language

SPaT Signal Phase and Time

SRC Short Range Communication

SSSP Single-Source Shortest Path

SWANS Scalable Ad-Hoc Network Simulator

TCP Transfer Control Protocol

TraNS Traffic and Network Simulation Environment

UDP User Datagram Protocol

UMTS Universal Mobile Telecommunications System

VANET Vehicular Ad-Hoc Network

VASE Vanet Simulation Environment

VRU Vulnerable Road User

VSimRTI Vehicle Simulation Runtime Infrastructure

W3C World Wide Web Consortium

WiMAX Worldwide Interoperability for Microwave Access

WLAN Wireless Local Area Network

List of Figures

List of Tables

1. Introduction

1.1. Motivation

The digital networking of the world has become ubiquitous and has affected more and more areas of our daily life in the first decade of the 21st century. People are able to check real-time bus schedules, to locate the closest and best-rated Italian restaurant, or to instantly share pictures of their latest mountain-bike trip with their family. However, the reason for this is not the wired Internet that was already widespread in the 90s, but rather the availability of wireless communication technologies to the masses. With the introduction of UMTS and WLAN for the expansion of wired local area networks, the Internet has become mobile and no longer bound to the home or to the working environment. Today's smart phones connect people everywhere, instantly, and at transfer rates that are comparable to wired technologies. This is achieved by the fourth generation of cellular radio: Long Term Evolution (LTE), which is currently rolled out. Furthermore, the trend in mobile communication is towards real-time Internet with the upcoming 5G standard.

A similar trend can also be observed in the context of the automobile, even if it is slightly offset. Today, a modern car is strongly networked internally and integrates hundreds of sensors that form the basis for increased safety, efficiency and comfort: climate sensors, radar, infrared- and daylight-cameras, just to name a few, enable a great variety of Advanced Driver Assistance Systems (ADAS) and In-Vehicle Information Systems (IVIS). Among others, the lane keeping assistant, automatic cruise control (ACC), parking assistance, and navigation are already offered in series-production vehicles. The widespread industry standard for connecting these sensors with the appropriate actuators in the car is currently being established via wired communication technologies.

When considering the success of wireless communication technologies in other areas such as smart cities, smart homes, or smart phones as mentioned above, the next logical step is to expand this technology to the domain of smart mobility. Individual transport in particular plays a major role in this context. OEMs recently introduced Internet in their cars allowing drivers to access emails, connect to their social networks, or retrieve information about parking spaces. While this is just a transfer of existing technology to the car, another great leap in the evolution of the automobile is now, after decades of research, just around the corner: Car2Car Communication is seen as the future of the smart car, since intermediary access points are dispensable. With the start of sim^{TD} in 2008, major German car companies, research institutes, and universities teamed up to bring the results of previous related projects onto the streets in the largest field operational test for cooperative vehicles. Even then, the technical foundations were already considered stable enough to test and evaluate use cases of the past in real traffic scenarios.

Direct communication between vehicles offers a variety of applications. Today, mostly safety-related applications are tested: cross-traffic assistance, emergency vehicle warnings, or congestion warnings just to name a few. From a technological point of view, this can be considered as extending existing sensors in a car with unidirectional wireless capabilities in order to share their information with other vehicles. These applications have one thing in common: Information is primarily broadcast from one sender to multiple receivers in order to disseminate messages and warnings over the ad-hoc network as quickly as possible, hence this communication strategy is referred to as *Push Paradigm*. Based on sensor data, dangerous situations that require notification of other vehicles are detected and sent to vehicles in range automatically by the application logic.

While the wireless extension of in-vehicle sensors will definitely contribute to the concept of smart cars, the technology itself constitutes enormous potential for another kind of use case beyond today's approach: By making sensors *accessible* to the community of cooperative vehicles, smart cars will become part of *smart vehicle networks*. Here, the ad-hoc network is considered as a huge information resource where data provided by sensors is available *on-demand* to anyone at any time.

Smart vehicle networks greatly enhance application areas of cooperative vehicles. The vision of this thesis is to introduce *interactivity* into the current communication paradigm of the connected car rather than just extending sensors by wireless capabilities and pushing their information to other cars. The application areas proposed in this thesis reach far beyond safety and traffic efficiency: They enable the driver to interact with the ad-hoc neighborhood.

Imagine the following situation from the year 2020: Bob arrives at a large parking lot. It is not immediately clear to him where the next open spot is located. Before he wastes time searching, he asks his new parking-lot assistant to guide him. The application acknowledges his request and augments it with specifications of Bob's car and his personal preferences. Then the request is sent onto the vehicular ad-hoc network. Thanks to the always-on always-connected paradigm of communicating cars, his request is also forwarded by parked vehicles. Using on-board sensors, they can identify open spots next to them. Even more, with the help of the specifications that were sent along with the request, *appropriate spots* are identified. An answer is generated and immediately sent back to Bob using other cars as communication relays. Bob is happy about the instant response to his request and starts guidance to his parking spot.

The described scenario illustrates that a bidirectional communication in mobile ad-hoc networks has the potential to greatly enhance driver experience. However, when the driver or the system requests information available somewhere in the network, the technical realization gets significantly more challenging than in current state-of-the-art ad-hoc networks. Since today's protocols are highly optimized for fast dissemination, they lack the ability to perform information source discovery and -retrieval. For example, Bob's request for a parking spot that fits the preferences of his car cannot be implemented since searching for the parking spot and delivering the result back to him are currently not considered at all. Within the proposed *Pull Paradigm* of this thesis, questions on how to locate the relevant information in a highly dynamic ad-hoc network and how to route back information once it is found are addressed. Information caching and dissemination within the ad-hoc network is also of great significance. Therefore, central elements of this thesis are two novel algorithms that enhance existing technology by the ability to perform mandatory information source discovery and -retrieval.

Another important aspect is that there is hardly any focus on the "last mile": the presentation of the requested information to the user. Generally speaking, an HMI design concept has not yet been developed. Therefore, a comprehensive automotive HMI is implemented that coordinates applications accessing limited resources in parallel. Although input and analysis of Pull requests also play a major role in interactive vehicular ad-hoc networks, this thesis focuses on the application and network level, thus developing *enabler technologies*. Requests are resolved and decomposed on the application level without requiring the user to input complex data. For example, Bob only needs to trigger the search for a parking spot while the car adds specifications of his car and personal preferences automatically.

Large field tests for evaluating wireless communication technologies and their applications as in sim^{TD} are expensive and potentially safety-critical. This is why a lot of research in the past has focused on the simulation of the underlying network communication in order to test Car2X-based ADAS in the lab before they get integrated into real cars. To simulate local danger warnings, for example, more than one simulator is required: Network simulation, traffic simulation, and environment simulation have to be coupled in order to address realistic scenarios. Analyzing the results of a test run involves scanning log-files containing information on wireless transmissions and the integrated applications. In order to assess the effect on drivers, a driving simulator is necessary. Therefore, a driving simulator will be added in the practical part of this thesis.

In order to give the reader a more comprehensive understanding of the Pull Paradigm, it is described in the next section.

1.2. The Pull Paradigm

This thesis proposes a novel paradigm for Car2X Communication networks by introducing *interactivity* to the ad-hoc community. This Pull Paradigm is compared to the established Push Paradigm in Figure 1.1. The Push Paradigm is *unidirectional* and starts with an event somewhere at an *information source* within the ad-hoc network. This event, a car crash for example, is then broadcast as fast as possible. On receipt, cars perform a relevance check and, if positive, the event is presented to the user. Furthermore, cars forward this event even if it is not relevant in the local context. The network stack is tailored to these kinds of applications. This will be thoroughly discussed in Chapter 2.

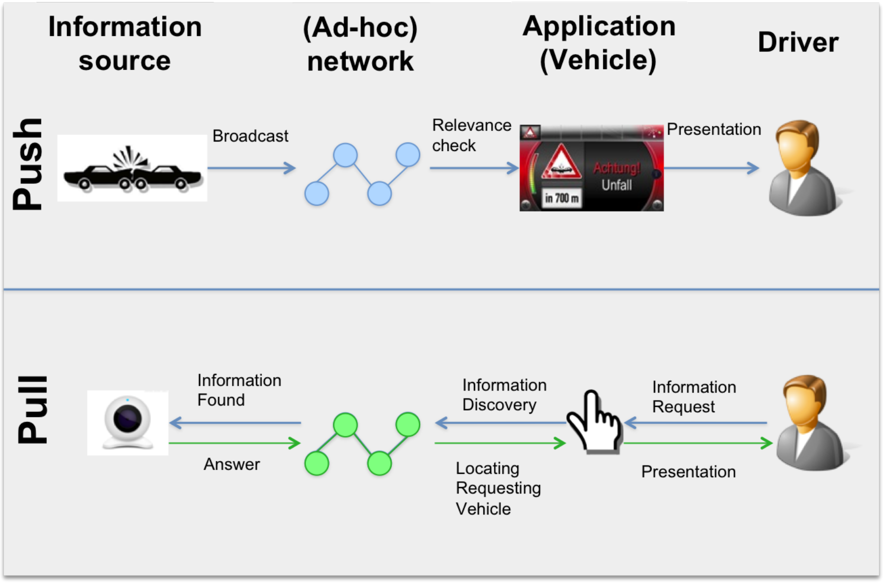


Figure 1.1.: The Pull Paradigm starts at the user who is actively requesting information from the ad-hoc network while the Push Paradigm starts at an information source, e.g. accident.

The Pull Paradigm, however, is *bidirectional* and starts at the user who is proactively requesting information from the ad-hoc network (Figure 1.1, bottom). In contrast to the Push Paradigm, an *discovery procedure* is mandatory in order to locate the information source, as it is unknown at the time when an information request is issued. Once the source is located, information needs to be transported back to the user's car. Most likely the topology of the ad-hoc network has since changed and therefore other cars are involved in the return path than in the forward run. Information discovery and information retrieval are key challenges for implementation of the Pull Paradigm and are addressed in Chapter 5. Here, advanced algorithms for information management in vehicular ad-hoc networks are developed.

The Pull Paradigm is defined as follows:

Definition 1 *The Pull Paradigm considers all entities of a vehicular ad-hoc network as potential information sources. In established vehicular ad-hoc networks every participating entity possesses only limited knowledge about other entities and information they hold. This limitation is defined by the communication range of an entity. The Pull Paradigm overcomes this limitation by making information sources globally accessible for every participating entity.*

1.3. Research Questions

This thesis addresses the following research questions:

- **Is the proposed Pull Paradigm described in Section 1.2 applicable to established Car2X Communication networks?** The established Car2X Push Paradigm is highly optimized to distribute information in vehicular ad-hoc networks as quickly as possible. This will become clear when looking at each layer of the Car2X network stack and its applications which are tailored to safety and traffic efficiency. However, within the Pull Paradigm vehicular ad-hoc networks are considered as a highly dynamic information resource. This thesis will show that an implementation of the Pull Paradigm requires profound changes to state-of-the-art Car2X networks.

- **What are possible application areas for the Pull Paradigm considering specific limitations of Car2X Communication networks?** As this thesis will show in Section 2.1, research in the Car2X domain has hitherto mainly considered safety applications especially when using vehicular ad-hoc communication. Introducing the Pull Paradigm widens the scope of possible application areas. Within the context of this research question the thesis will propose novel applications for the Pull Paradigm and also consider specific technical limitations of Car2X Communication networks.

- **Are existent simulation tools applicable for integrating Pull applications?** Realistic simulations of Car2X networks are an essential component in the toolchain for developing applications for cooperative vehicles. They enable cost-efficient evaluation of large scenarios or safe execution of dangerous situations. By looking at existing simulation tools this thesis will assess how they can be applied to the proposed Pull paradigm and what enhancements are necessary.

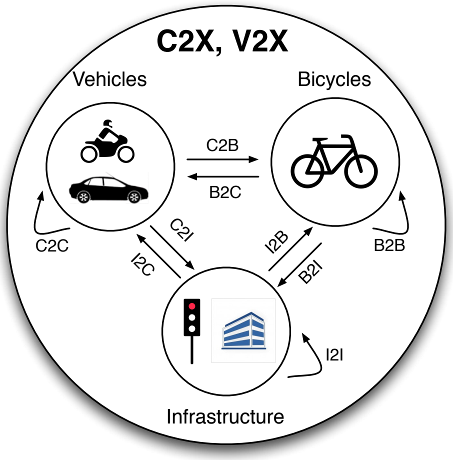
- **How can development of Pull applications be supported on the software level?** Existing software tools for developing Car2X based applications are tailored to safety and traffic efficiency. Within the scope of this question a framework that support development of Pull applications is introduced. The thesis will develop an end-to-end support chain from the protocol level, over the application level, to HMI level.

- **How can existent ADAS / IVIS and Pull applications be integrated in a common HMI framework?** The modern car integrates a range of active driver assistance systems (ADAS) and in-vehicle information systems (IVIS). Adding Pull applications to this already-rich variety of systems brings up the question of how these can be integrated. Here, an HMI framework for Car2X-based Push and Pull applications will be developed that coordinates parallel-running tasks which access limited HMI resources at the same time. The focus is on information presentation rather than the input of Pull requests.

- **What are qualified evaluation metrics that measure performance of the Pull Paradigm?** Up to this point, no metrics are known that express reliability and robustness of a Car2X network that support bidirectional communication. Within the scope of this research question, the thesis will define suitable metrics that are applied to the evaluation of Pull-enabled vehicular ad-hoc networks.

1.4. Terminology

Throughout this thesis several abbreviations are used. This section summarizes these terms and introduces their meaning.



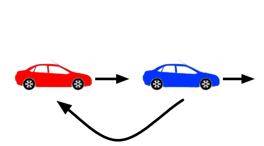
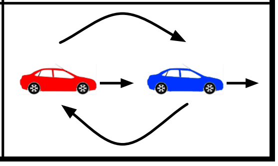
Push	Pull
unidirectional	bidirectional
asymmetric	symmetric
information-centric	user-centric
	

Figure 1.2.: Terminology used in this thesis. **Left:** Communication partners in the Car2X domain, **right:** Terminology for the Pull Paradigm.

In European research the term “Car2X (C2X)” has been established when referring to vehicular ad-hoc networks and will be used in this thesis. “X” denotes the communication partner but not the specific wireless communication technology used. Communication partners in the context of this document are other cars (C2C), bicycles (C2B), motorcycles, or infrastructure (C2I), which encompasses stationary objects on the roadside. In the literature, these include roadside units, traffic lights, electronic traffic signs, or traffic authorities. Furthermore, the garage door, home, and office can also be included.

Communication in the opposite direction is often indicated by interchanging the letters, e.g. I2C for infrastructure-to-vehicle communication (see also Figure 1.2, left). With the communication paradigm detailed in this thesis, novel communication partners are added, such as environmental displays, or charging stations for electric powered vehicles. Note that the term "Vehicle2X (C2X)" can often be found in the American literature but is not used in this document.

The terminology for the Push and the Pull Paradigm is shown in Figure 1.2, right. As introduced in Section 1.2, bidirectional communication is added to established Car2X networks but also the term *symmetric communication* can be used in contrast to *asymmetric communication* for the Push Paradigm. The Pull Paradigm is considered to be *user-centric* whereas the Push Paradigm is clearly focused on presenting information, hence it is said to be *information-centric*.

1.5. Outline of this Thesis

Chapter 2 gives an in-depth analysis of the established unidirectional Car2X networks. Early research in this domain and evolvements of communication technologies over time are presented. An important role in Car2X research is vendor-independent communication, which led to several efforts for standardization on an international level. The chapter continues by looking at today's wireless communication technologies and analyzes their applicability to the Pull Paradigm. Furthermore, tools for an all-embracing simulation of Car2X networks are presented.

Chapter 3 describes application areas of the Push Paradigm. It follows well-established categories of applications in this domain: safety, traffic efficiency, infotainment and other. Different sources are analyzed: conceptual use cases that were specified by leading standardization institutes as well as implementations in the largest field operation tests. Finally, an overview of already-available products in the context of connected cars are presented.

Chapter 4 draws conclusions from the results that were gained in Chapters 2 and 3. In order to implement the Pull Paradigm, Chapter 4 lists necessary enhancements to today's Car2X Communication networks.

Chapter 5 introduces advanced algorithms for information management in vehicular ad-hoc networks that add bidirectional communication. These algorithms are superior to the existing state of the art. They lay the foundations for implementation of the Pull Paradigm on the network level and add information discovery, advanced dissemination techniques, and information retrieval to Car2X networks as introduced in Section 1.1. For the sake purpose of generality, the problem is transferred to graph theory and solved within this domain.

Chapter 6 deals with the Pull Paradigm from the application level and introduces four novel categories with four applications each. Additionally, new ADAS are presented which incorporate vulnerable road users into the Car2X community. Requirements for each application with respect to the Pull paradigm are investigated. Furthermore, the multiple reality model is introduced, which describes changing perceivable characteristics of real objects in a way such that a *directed* information transfer to the intended addressee can be realized, e.g. no other road users are involved. Chapter 6 concludes by selecting some of the most promising Pull applications for implementation using a well-established selection process.

Chapter 7 implements the Pull Paradigm in software referred to as Car2X Platform for Application Development and Evaluation (C2X PADE). It consists of several modules, each addressing one requirement of the Pull Paradigm as identified in Chapter 4. C2X PADE provides tools for application development, messaging support, and also adds simulation tools for assessing large scenarios. A special abstraction layer enables transfer to a test vehicle with almost no code changes. Advanced algorithms developed in Chapter 5 are implemented as routing algorithms and integrated into the Car2X network stack. Furthermore, a presentation module supports development of graphical user interfaces.

Chapter 8 applies C2X PADE by developing Pull applications that were selected in Chapter 6. Additionally, Push applications implemented in sim^{TD} using PADE's presentation module are presented. The last section of this chapter implements a first version of the multiple reality model that was previously introduced in Chapter 6.

Chapter 9 evaluates advanced algorithms that were detailed in Chapter 5 on simulated city scenarios. The chapter introduces novel evaluation metrics that measure the algorithms' performance in the Pull Paradigm.

Chapter 10 presents PADE's practical application to several research systems while this thesis developed. In sim^{TD} for example, Volkswagen and Opel decided to replace the project-specific user interface with an OEM-designed version. PADE's presentation module controlled this user interface. This chapter describes the adaptions to PADE and the results. Furthermore, integration of PADE into DFKI's BMW test vehicle is presented. The last practical example describes a system for E-Bike fleet management that was demonstrated to the public at the Mobile World Congress 2013 (MWC) in Barcelona.

Finally, **Chapter 11** concludes with the findings of this thesis and critically discusses the applicability of the Pull Paradigm to first generation Car2X Communication networks.

2. Foundations of Car2X Communication

This chapter lays the foundations that enable the implementation of the Pull Paradigm in Car2X Communication networks. For this, it is essential to build up a strong understanding of the underlying communication technology. Only with this knowledge it is possible to assess possibilities and limitations with respect to the goals of this thesis.

The analysis of Car2X Communication networks starts with a historical overview of projects that were completed since the 1980s. The goal is to familiarize the reader with research in this domain. Car2X Communication has been thoroughly studied in the past across all continents. Therefore the presentation of projects cannot be complete, but covers important milestones. Besides this historical overview, latest advances in the domain of vehicular ad-hoc networks is discussed. This includes two field operational tests on national and international level which are considered as a last step before introduction to the market. Parts of this thesis contributed to one these projects, the currently largest field operational test for Car2X Communication, sim^{TD}.

In order to allow vendor-independent vehicular communication, activities have to be coordinated across OEMs and technology has to be standardized. These aspects are of great importance for this thesis, since the developed system should be compatible with current standards and/or augment them at key points. Established standards and standardization institutes relevant for Car2X Communication are therefore presented.

Vehicular ad-hoc networking and other wireless communication poses several challenges to the technology which are introduced this chapter. Furthermore, essential tools for simulating large-scale scenarios are analyzed. After selecting suitable tools, they will be used in Chapter 7 for building a software framework that fulfills the requirements of the Pull Paradigm. Novel applications can then be built on top of this framework.

2.1. The “Push” Paradigm: Today’s Car2X Communication Networks

2.1.1. Historical Overview

An increasing number of C2X-related research projects have been conducted since the beginning of the 1980s. Activity could be observed in Europe, the United States of America and Japan. This section describes the history of research in this domain but due to the large number and variety of projects, it makes no claim to be complete. Rather, important milestones and contributions are described. The focus is on activities in Europe, and especially applications, as well as their development over time.

PROMETHEUS

Literature in the C2X domain often cites the EUREKA-PROMETHEUS[1] [Braess, 1995] project as the first step towards cooperative vehicles. The project was carried out from 1987 – 1994 and funded by the European Union and pursued the ultimate goal to develop autonomous cars. Nowadays, ADAS like Automatic Cruise Control (ACC) can be found in series-production vehicles. The basis for this technology was developed within research activities in Prometheus: The VITA vehicle (Vision Information Technology Application) was able to steer, brake, and accelerate the car autonomously. By detecting cars and other obstacles using a computer vision approach, it was able to keep a certain distance from vehicles in front of it. Wireless communication technology was also researched in order to improve traffic safety. [Walid Dabbous, 1988] present first approaches to C2C communication on the basis of wireless ad-hoc networks. Even in this early stage, they distinguish a communication system and an expert system which uses artificial intelligence techniques for presenting action guidelines to the driver. Implementations today pursue a similar approach: Most frameworks use a communication unit for managing the C2X neighborhood and an application unit. On the application side, Dabbous et al. describe five scenarios:

- **Merging lanes (highway):** While entering the motorway, the vehicle is in another vehicle’s blind spot. Cooperation is needed between the two involved cars in order to resolve this potentially critical situation.
- **Passing (rural road):** Passing in rural scenarios is especially dangerous when large parts of the road are occluded. A head-on crash could be the consequence. In the scenario described in [Walid Dabbous, 1988], the car announces its intention to pass and awaits acknowledgement from the moving vehicle ahead.
- **Crossing vehicles (urban traffic):** Two vehicles intend to cross the same intersection at the same time. Vehicular communication prevents a crash by adjusting

[1] PROgraMme for a European Traffic of Highest Efficiency and Unprecedented Safety

their trajectories. This scenario is still part of C2X research today: Lateral cross-traffic assistance is described in Chapter 3. In contrast to this first approach, autonomous intervention is not desired.

- **Convoy driving (on a highway):** One vehicle is leading a convoy on the motorway.
- **Accident in fog (on a highway):** Stopping a car on a highway leads to automatic broadcast of an emergency signal. Approaching vehicles are warned accordingly. The content of this signal is not specified in the document. This scenario can be compared to today’s well-researched local danger warnings (cf. Chapter 3).

Some of the ideas in Prometheus can be found in today’s C2X networks. Although first steps in autonomous driving succeeded in Prometheus, implementation of the fully operational system including all ideas described above was not achieved due to the lack of suitable communication technology. However, the size and impact of the PROMETHEUS programme led to a high degree of popularity in the Car2Car research community.

Wolfsburger Welle

However, it should be noted here that earlier, less well-known research exists. For example, an initiative by Volkswagen and Siemens from 1981 to 1983 is worth mentioning. The project “Wolfsburger Welle” (Wave of Wolfsburg) [Zimdahl, 1983] can also be classified as work within the I2C domain. It pursued the goal of optimizing traffic flow in urban environments by suggesting the optimal speed for passing upcoming traffic lights during their green phase. In order to transmit the relevant information from the road infrastructure into vehicles, Volkswagen and Siemens used infrared light. Even in this early phase, they used an HMI in the form of a horizontal bar (see Figure 2.1). Depending on current vehicle speed, the bar was moving on top of a yellow and a green zone, indicating during which phase the car would pass the traffic light. Drivers could then accelerate or decelerate accordingly. The system was integrated together with “Autoscout” in a VW Transporter T3. The latter system can be seen as a first step towards today’s common navigation systems, which will not be presented in detail here. Although technology for I2C communication experienced a shift from infrared light to radio waves, a similar use case in recent research within the Push Paradigm will be discussed in Chapter 3.

Two interesting results of the project should also be mentioned here. First, drivers were often speeding when using the Wolfsburger Welle. They justified their behavior by pointing out that the system recommended this action. Second, due to the high number of necessary glances towards the HMI, it was found to be highly distracting from the primary task [Richter, 2005]. This indicates that even in this early stage, researchers were aware of driving distraction, but approaches to minimize it were not addressed within the scope of the project. The focus was clearly on technical questions.

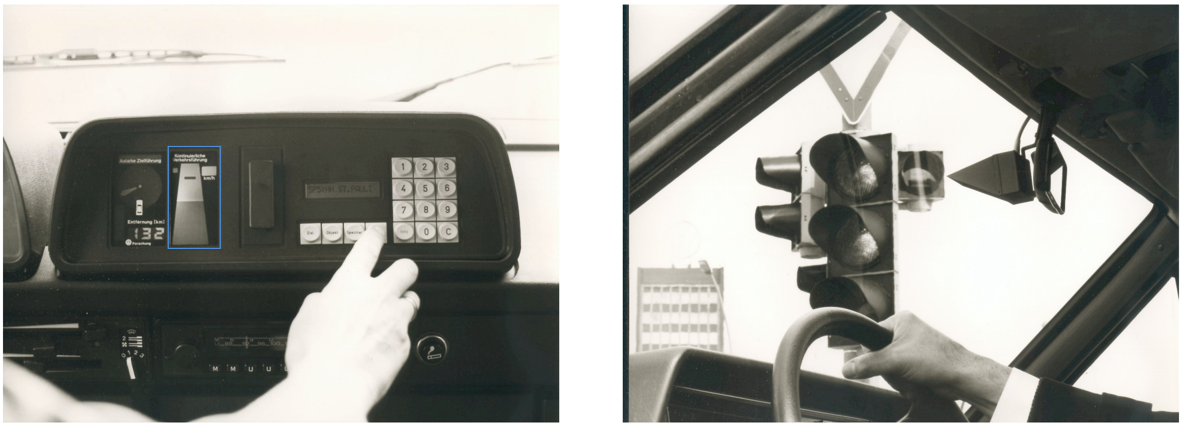


Figure 2.1.: "Wolfsburger Welle". On the left picture the HMI for this system is shown. The two zones marked by the blue box were yellow and green. Movement of the black bar on top was coupled to current vehicle speed and traffic light phases. By accelerating or decelerating, the driver could adapt her speed in order to pass the traffic light during its green phase. The right picture shows an infrared receiver behind the rear-view mirror. Pictures were taken from a joint press release of the Siemens AG and the Volkswagenwerk AG [Siemens, 1983].

With the increasing availability of suitable wireless communication technology, several research projects were carried out during the 1990s.

Chauffeur I / II

From 1996 to 1998, the EU-funded project Chauffeur I[Brandenburg et al., 2000] investigated the feasibility of what is called "road trains" today: A vehicle driving in front is leading autonomously steered followers. The first car of the train is controlled by an active driver. Chauffeur I targeted the safe increase of freight traffic on motorways using these road trains. The project involved professional drivers and freight forwarders. Together with experts, they evaluated the technical and ecological feasibility of three Chauffeur I applications:

- Tow-Bar: Two trucks are driving in vertical alignment where only the first truck needs a driver. The truck behind is maintaining speed and distance.
- Platooning: A series of more than two trucks are coupled electronically. Only the first truck needs a driver.
- Automated platooning: a feasibility study of fully automatic truck platooning.

Both involved parties, freight forwarders and professional drivers, clearly saw the need for such a technology. Professional drivers explicitly pointed out the importance of suitable HMIs. Passing strategies and keeping the driver aware of the running system were

Figure 2.2.: Trucks used in Chauffeur II for the platooning application. On the left picture, the active infrared pattern used for tracking is clearly visible.

also rated as essential. At this time, making such a system more generic was also discussed, allowing it to couple arbitrary trucks, not only Chauffeur-equipped vehicles.

Based on the findings of Chauffeur I, the follow-up project Chauffeur II[Brandenburg et al., 2000] then realized two applications: Chauffeur Assistant and Platooning. The first was able to follow an arbitrary truck driving in front. The second realized platooning on a test track involving three trucks. From a technical point of view, Chauffeur II combined vision-based approaches (infrared), radar and also 5.8 GHz radio communication. Figure 2.2 shows trucks used for this system during the final presentation.

Road trains can also be applied to the Pull Paradigm. Consider a situation of the future where this mobility concept on motorways is a common way to travel. Before "attaching" to a road train, the driver could start a request to other cars in order to identify the most suitable for them, e.g. the leading vehicle of the road train travels a route that closely matches her own.

CONCERT

The CONCERT[2] project was an EU-funded research initiative from 1996 – 1998. Eight European cities were working on the same goal: travel demand management by means of telematic tools [Appert et al., 1998a]. Among other things, it combined integrated payment with smart cards, pricing and access control for managing road use[3]. The project is mentioned here because one of the demonstrators, TRON2, is particularly interesting within the scope of this thesis. It was concerned with road pricing based on 5.8 GHz DSRC equipment [Appert et al., 1998b]. The goal was to charge cars entering the city

[2]CO-OPERATION FOR NOVEL CITY ELECTRONIC REGULATING TOOLS
[3]http://cordis.europa.eu/telematics/tap_transport/research/projects/concert.html

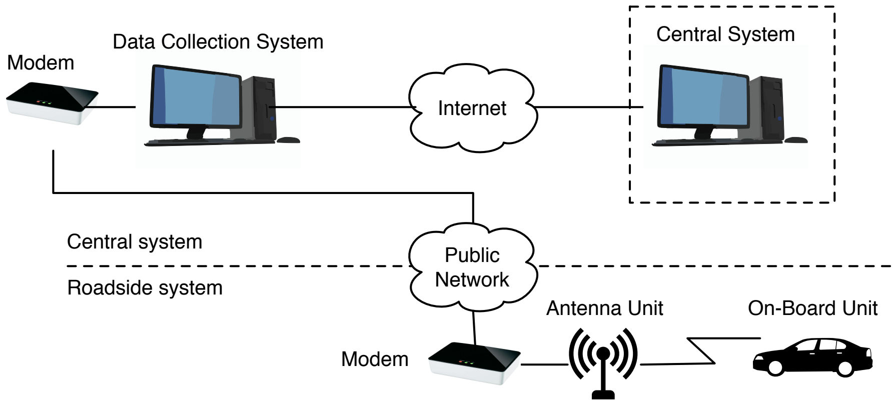


Figure 2.3.: System overview of the TRON2 demonstrator. An early example of C2I communication applied to toll payment. Adapted from [Appert et al., 1998b].

center of Trondheim by using C2I communication. A roadside station (RSS) covered all entries of the town center. On-board Units (OBUs) were placed in the cars, each carrying a unique identifier. Figure 2.3 shows a complete system overview of the TRON2 demonstrator. User acceptance was measured using questionnaires. For technical evaluation, drivers were instructed to follow a validation procedure when passing an RSS. No information could be found about evaluation results of TRON2. However, it is listed in this section in order to demonstrate early application of I2C communication to wireless payments.

COOPERS

The project Co-operative Systems for Intelligent Road Safety (COOPERS) [Bankosegger et al., 2010] was carried out from 2006 to 2010 within the 6th framework programme of the European Commission.

COOPERS aimed to fill the gap between C2C Communication and research in the area of autonomous vehicles. By creating a permanent link between cars and road infrastructure, COOPERS addressed overall traffic safety and cooperative traffic management. Interestingly, the project used direct communication between vehicles. The infrastructure communication relied on broadcast systems, infrared and mobile radio (see Section 2.2.1 for a detailed description of these technologies).

On the application level, COOPERS studied 12 traffic services [McDonald and Richter, 2008]:

- Accident/Incident/Wrong-Way Driver Warning: provide in-vehicle, dynamic information to warn other drivers of an accident/incident.
- Weather Condition Warning: provide in-vehicle, dynamic information to warn drivers of hazardous conditions ahead caused by adverse weather conditions.

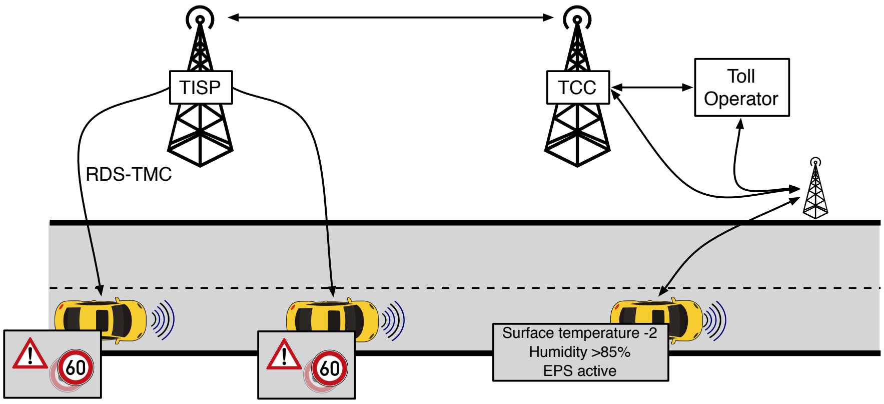


Figure 2.4.: The vision of the project “Co-operative Systems for Intelligent Road Safety” (COOPERS). Adapted from [Pfliegl, 2010].

- Roadwork Information: provide in-vehicle, real time information of roadwork to warn drivers on road conditions ahead caused by the roadwork.
- Lane Utilization Information: provide drivers with in-vehicle and real time information on lane control and lane utilization including lane banning, lane keeping (no passing), and auxiliary lane information (shoulder lane).
- Variable Speed Limit: provide drivers with in-vehicle information on the current speed limit of the road being traveled on.
- Traffic Congestion Warning Secondary services: provide drivers with in-vehicle and real-time traffic congestion information.
- Intelligent Speed Adaption with Infrastructure Link: provide drivers with in-vehicle and real-time information on the current speed limit of the road they are traveling on and provide warning if the speed limit is exceeded.
- International Service Handover: the objective of the service is to enable drivers to access road conditions and traffic information for a neighboring country and/or to find road operators before crossing a border/changing operator areas.
- Road Charging to Influence Demand: the objective of the service is to realize electronic fee collection which offers the possibility of charging road vehicles in a more flexible way
- Estimated Journey Time / Recommended Next Link / Map Information Check: to help drivers to plan their trips based on estimated journey time

While some use cases like road tolling were also researched in CONCERT [Appert et al., 1998a], COOPERS introduced a number of additional applications based on C2I communication using cellular networks. Most of these services are considered in the

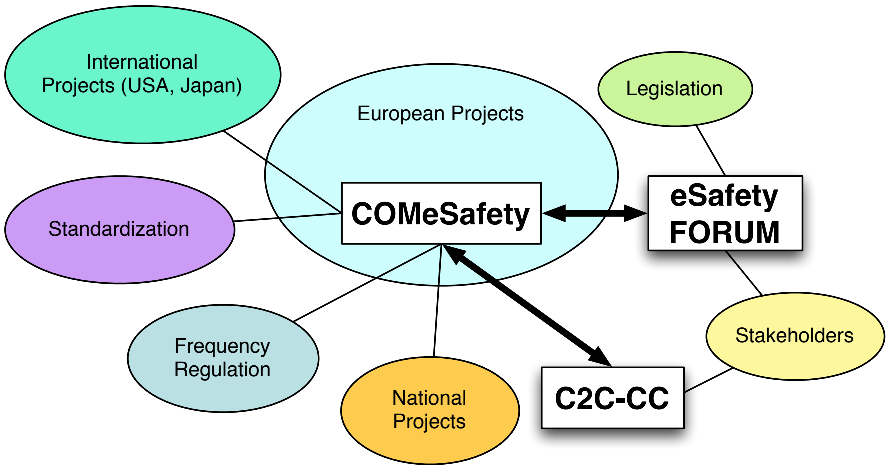


Figure 2.5.: COMeSafety overview diagram.

latest field operational tests for Car2X Communication networks, as Chapter 3 will show. However, the focus is not on cellular networks. Rather, communication technologies are combined in order to provide the most reliable services.

2.1.2. Linking Activities of Different Countries

By the beginning of the 21st century, several activities could be observed that aimed for harmonization and accelerating market introduction of IT-based technology for road safety. The most important ones are presented here: the eSafety Forum and its support actions COMeSafety 1 and 2.

eSaftey Forum

The eSafety forum[4] is an initiative of the European Commission starting in 2002. The initiative emerged from the idea that information and communication technology can contribute to traffic safety and significantly lower road fatalities. The eSafety forum therefore brings together all involved parties from European industry in order to accelerate development and market introduction. Furthermore, several research projects helped to advance the technology. In order to guarantee a smooth operation beyond borders of countries, it was important to embed the initiative on a European level.

Of course, the eSafety forum includes advances for safety-related applications based on vehicular communication, but it is not limited to those. The systems supported include both, cooperative and stand-alone systems. An important safety feature that was pushed by the forum and was already introduced in production vehicles is the popular electronic stability program (ESP). Other ADAS include lane departure warnings or lane changing assistance.

[4] eSafety website: http://ec.europa.eu/information_society/activities/esafety/index_en.htm

COMeSafety 1 and 2

COMeSafety[5] is an explicit support action to the eSafety forum initiated by the European Commission in 2006. The duration was projected to be 48 months and was followed by COMeSafety 2 afterwards. Figure 2.5 shows the complexity of COMeSafety and its relation to other projects and organizations. According to the EC, “COMeSafety supports the eSafety Forum with respect to all issues related to vehicle-to-vehicle communications as the basis for co-operative intelligent road transport systems”[6]. Among others, COMeSafety supported important cooperative vehicle projects like CVIS [Kompfner, 2010], SAFESPOT [Ehrlich et al., 2010], PReVENT [Schulze et al., 2008], and COOPERS [Bankosegger et al., 2010].

COMeSafety coordinates and consolidates research results of the supported projects. By collaborating with the Car2Car Communication Consortium and the European Telecommunications Standards Institute (ETSI), it is also supporting worldwide dissemination, harmonization and standardization of systems based on vehicular communication. Activities such as COMeSafety are particularly important to understand mutual influence of research projects, stakeholders, and standardization bodies.

2.1.3. Preparing Market Introduction: C2X Field Operational Tests

The Car2X projects presented in the previous section developed research prototypes. Their goal was to stabilize the underlying network technology as well as to prove the feasibility of Car2X. It should also be remarked that applications served as a component in these feasibility studies, but important questions that are connected with them were not addressed: although a number of different use cases were developed and standardized (see Chapter 3), the effects on drivers, modality choice and timing were not considered at all. Furthermore, all tests were conducted on closed test sites with no real traffic. Recently, when the first Field Operational Tests (FOTs) were started, more emphasis was put on these type of questions. By being part of one of these FOTs, the author of this thesis contributed to a large extent to these human-factor issues that have to be considered when introducing this technology.
This section presents two of the largest FOTs for Car2X Communication that were carried out recently. They are considered as a final step before market introduction.

Safe Intelligent Mobility Test Field Germany (sim^{TD})

Parts of this thesis were conducted in the project sim^{TD}[7] [Assenmacher, 2009]. It was one of the first and largest FOT for Car2X Communication and was carried out from 2008 – 2013. The project’s testing grounds were in Frankfurt / Main and included three scenarios: motorway, rural road and city (see Figure 2.6, left). sim^{TD} was a joint project of

[5]COMeSafety website: http://www.comesafety.org/
[6]http://ec.europa.eu/information_society/activities/esafety/doc/rtd_projects/fact_sheets_fp6/call_4/comesafety.pdf
[7]sim^{TD} website: http://www.simtd.de

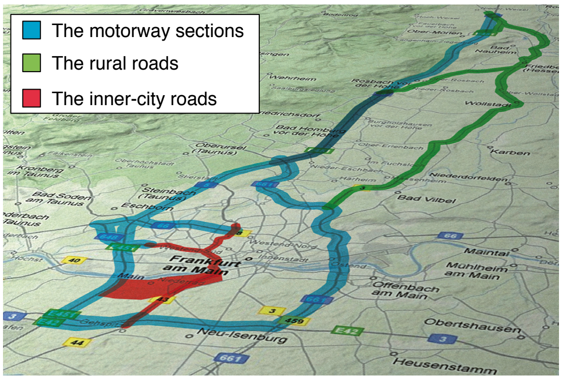


Figure 2.6.: **Left:** The entire sim$^{\text{TD}}$ Test Field Hesse, centered around the Hessian metropolis Frankfurt am Main, **Right:** Cars of the sim$^{\text{TD}}$ test fleet returning to base after a day in the field. Image source: Official sim$^{\text{TD}}$ press material available at http://www.simtd.de

leading German vehicle manufacturers, components suppliers, telecommunication companies and research institutions. Additionally, the Car2Car Communication Consortium (cf. Section 2.1.5) was supporting the project. As a German research initiative, sim$^{\text{TD}}$ was funded by the government with around 69 million Euro, and around 31 million Euro were contributed by the partners. Three federal ministries were involved: Education and Research, Economics and Technology, and Transport Building and Urban Development.

By consolidating research results of the prototypes developed in earlier projects, sim$^{\text{TD}}$ went one step further and evaluated not only the technology but also its impact on traffic and drivers by embedding the FOT in a metropolitan area. Therefore, the goals of sim$^{\text{TD}}$ were manifold. From an application point of view, over 20 Car2X-based use cases in the categories of safety, traffic efficiency and added-value services were selected in a well-defined process, documented and evaluated within the scope of sim$^{\text{TD}}$. The project was also a first approach to harmonize these different categories in one complete system. Furthermore, sim$^{\text{TD}}$ developed validation metrics for evaluating the results of the FOT. As another major goal, traffic safety should be increased using Car2X Communication. Furthermore, sim$^{\text{TD}}$ aimed to define and validate a roll-out scenario for the selected functions. This is especially important, since car manufacturers are planning to introduce the technology to the market by 2015 [Car 2 Car Communication Consortium, 2011]. sim$^{\text{TD}}$ made essential contributions to this goal by evaluating real-life scenarios in everyday traffic situations. To a large extent, the applications were also subject to standardization processes during the first decade of the 21st century (c.f. Section 2.1.5). sim$^{\text{TD}}$ represents the latest development in this field and shows the focus of today's Car2X Communication networks also with respect to application areas. Because of their importance to this thesis, Chapter 3 is dedicated to this topic and will discuss them in detail.

The German Research Center for Artificial Intelligence (DFKI) was a project partner and was responsible for the human-machine interface (HMI) solution. Within this scope, the presentation component of PADE introduced in Chapter 7 was developed. Due to the considerable demands in terms of stability and robustness of the FOT to the integrated components, PADE's presentation component represents a well-tested and efficient software module that received continuous improvements before it was rolled out on all 120 cars of the test fleet. Test drivers experienced the system on a daily basis and their feedback was integrated into updates. Due to its generic architecture it could be applied to the applications that were developed in this thesis with almost no modifications. Therefore, it represents a system which goes beyond the level of research prototypes.

In the context of sim^{TD} there is another interesting German research activity. Started in 2006, the initiative AKTIV[8] (Adaptive and Cooperative Technologies for Intelligent Traffic) [AKTIV Consortium, 2006] brought together 28 partners from automobile manufacturers and their suppliers, electronics and telecommunication companies, research institutes, as well as public road and traffic authorities. It is funded with 27 million Euro by the Federal Ministry of Economy and Technology.
The project addresses three different application areas: AKTIV-AS is concerned with active safety, therefore developing "visionary assistance systems". An example is given by the automatic braking system that is designed to prevent rear-end collisions by "active hazard braking". AKTIV-VM addresses traffic management and developed a cooperative information system in order to increase efficiency on the roads. The third application area is CoCar (Cooperative Cars). Here, the communication between vehicles and traffic information systems is researched. Ad-hoc and mobile communication were used to realize a system which provides "up-to-the-minute" information on road conditions to individual vehicles. As Chapter 3 shows, this is also a use case in sim^{TD}. Although AKTIV was not involved in the ongoing standardization process, its follow-up project CoCarX[9] (Cooperative Cars eXtended) can be defined as an extension to sim^{TD} [Kosch et al., 2012a]. Research done in CoCarX uses a combination of LTE, the next-generation standard for mobile communication, and wireless ad-hoc networking in order to provide on-time information for the connected car. However, LTE coverage is still fragmentary throughout Europe. CoCarX is nonetheless mentioned here because it advances technology used in sim^{TD}, although the use cases are the same. The focus is on technology and feasibility studies.

DRIVE C2X

The project DRIVE C2X[10] [Stahlmann et al., 2011] was the counterpart to sim^{TD} on a European level from 2011 – 2013. With total costs of 18.8 million Euro and 12.4 million Euro requested funding from the European Commission, it was smaller in terms of project size compared to sim^{TD}. The consortium was formed by 31 partners includ-

[8] http://www.aktiv-online.org
[9] http://www.aktiv-online.org/english/aktiv-cocar.html
[10] DRIVE C2X website: http://www.drive-c2x.eu

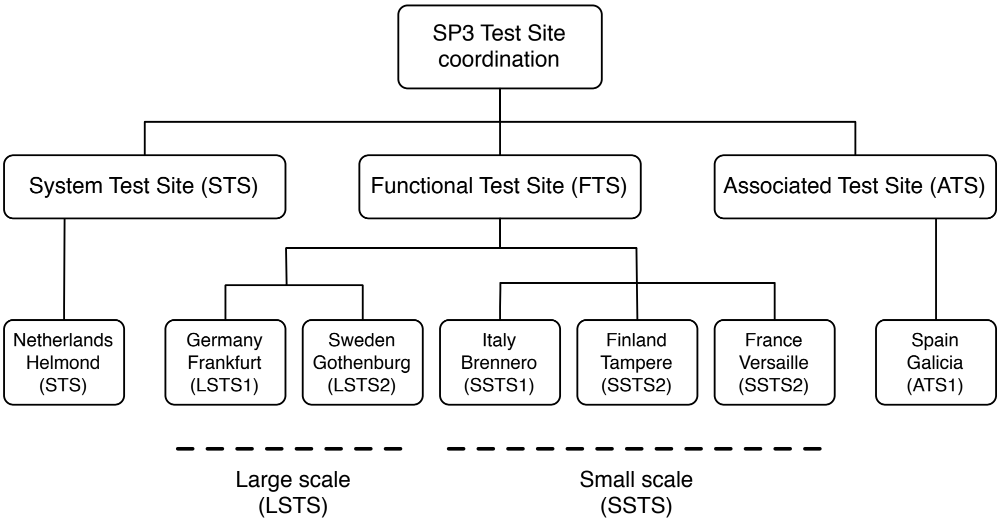


Figure 2.7.: The test sites of the DRIVE C2X project throughout Europe.

ing car manufacturers, the electronics and supplier industry, software developers, traffic engineers, research institutes as well as road operators. Among others, the European Telecommunications Standards Institute (ETSI, see Section 2.1.5) was one of the 15 support members of the project.

DRIVE C2X was carried out on seven test sites (see Figure 2.7) throughout Europe in which the benefits of the technology in terms of traffic safety and efficiency as well as added-value services were identified. The project also came up with a cost estimate for introducing cooperative systems to the market in relation to the benefits. According to the project description, DRIVE C2X provided decision-makers at car manufacturers with valuable information regarding costs of system implementation in relation to benefits and revenue generated by Car2X Communication. Therefore, DRIVE C2X was more concerned with socio-economic and business-economic questions in connection with cooperative systems. Another important aspect mentioned is the cooperation with ETSI, as standards are also a key factor in this domain.

Only 9 functions were selected for full evaluation in DRIVE C2X on all test sites. As Chapter 3 will show, a complete overlap exists between the functions tested in DRIVE C2X and sim^{TD} where the latter introduces additional use cases.

At this point, it should be noted that the FOT conducted within DRIVE was prepared by an earlier project called PRE-DRIVE C2X (preparation for driving implementation and evaluation of C2X communication technology) [Schulze et al., 2010]. It was carried out in context of the 7th framework programme funded by the European Commission from 2008 to 2010. The total cost of the project was 8.4 million Euro (5 million Euro funding). Together with DRIVE, this European-wide FOT is still smaller than sim^{TD}.

Based on the developed scenarios and applications in COMeSafety, a specification and prototype of a common European Car2X Communication system should be realized and necessary tools for conducting a FOT on an European level based on Car2X technology should be developed. From an application point of view, PRE-DRIVE C2X came up with a description of 53 use cases. Out these, 23 are safety related, 13 address traffic efficiency, and 19 are described as "infotainment, business and deployment use cases". According to a previously defined selection method, 16 were chosen for prototypical implementation (6 safety, 6 traffic efficiency and 4 infotainment, business and deployment use cases). However, it is remarked in [Enkelmann et al., 2008] that a use case must be supported by at least one partner of the consortium in order to implement it. In the context of PRE-DRIVE C2X, no information on this could be found. However, in DRIVE C2X only 9 use cases were selected for full implementation and assessment within the project, e.g. on all test sites of the project (cf. Chapter 3).

2.1.4. Recent Developments

UR:BAN

Started in 2012 and continuing until 2015, the German research project UR:BAN[11] develops ADAS and traffic management systems for complex city environments. In contrast to sim^{TD} and DRIVE, pedestrians and vulnerable road users such as bicyclists are also considered. However, they are not equipped with Car2X technology. Cars use internal sensors for detecting VRUs. In the context of additional Push applications in Chapter 6, VRUs are also integrated into the Car2X community.

The project shows that recent developments in the research area of cooperative systems are more focused on the users rather than on technical questions only. This becomes clear when looking at the three sub projects of UR:BAN: cognitive assistance, networked traffic system, and human factors. Within cognitive assistance, safety of complex city scenarios is improved. The sub project models vehicle surroundings in order to protect vulnerable road users, develops accidence avoiding maneuvers, and offers lateral and longitudinal vehicle control support. However, all application areas in this first sub project do not involve Car2X Communication. This is considered in the second sub project, networked traffic system, that develops methods for improving traffic efficiency and lowering emissions in city environments. The vision of the project is to deploy intelligent infrastructure and to connect it to intelligent vehicles which also takes into account "drive trains". Required information about the environment, e.g. traffic forecasts, is acquired using C2I communication. The sub project human factors in traffic targets drivers of future ADAS and IVIS. UR:BAN aims to develop an "individualized design" in order to reduce stress and to achieve safe and efficient driving in the city. Therefore, UR:BAN is continuing the work on human factors in the context of automotive HMIs for ADAS and IVIS that started in sim^{TD}.

[11] http://www.urban-online.org/

CONVERGE

Converge[12] (COmmunication Network VEhicle Road Global Extension) is a German research project that started in 2012 and continues until 2015. The project considers the Car2X systems network as a whole and converges its partners and their roles into one communication-, services- and organization architecture. This is especially important for future cooperative systems since vendor-independent communication is not only essential for direct communication between vehicles but also when traffic authorities are involved. CONVERGE provides a common service architecture that is open to vehicle manufacturers, traffic centers, and mobile communications providers. In practice, this architecture allows cars to communicate over different cellular networks using the same architecture. Technically, the project adapts message formats that are exchanged between cars to the requirements of cellular networks. Finally, a "virtual marketplace" offers Car2X services to drivers, vehicle manufacturers, and traffic centers. The concept of this virtual marketplace is particularly interesting for this thesis since new applications can be deployed faster without complex standardization procedures. The use cases proposed in Chapter 6 would benefit from such a harmonized architecture of a Car2X Systems Network.

DFKI is also involved in CONVERGE and continues the work that was done in sim$^{\mathrm{TD}}$. The presentation module of PADE (cf. Chapter 7) serves as a basis and is further developed and adapted to the project.

2.1.5. Standards and Standardization

ADAS recently introduced in cars on the market like the Automatic Cruise Control (ACC), solely rely on active sensors in the vehicle that are used to determine the situation of the outside environment. No complementary sensors of other cars are needed for proper function. In the Car2X domain, however, other vehicles also need to be equipped with communication technology and use common protocols in order to realize vendor-independent functionality. Data exchange at high velocities in an ad-hoc manner requires dedicated protocols and hardware. This is a major reason why standardization processes are considered as a mandatory part of this field of research. Not only were technical standards issued since the beginning of this century, but also a common set of applications and architectures have been defined. However, standardization processes are still ongoing and are far from being completed. This section presents the organizations that are mainly responsible for pushing forward common standards that ensure interoperability in heterogeneous C2X communication networks. Furthermore, the most relevant standards that are currently established or under development are presented.

Standardization Institutes in the Context of Car2X Communication

European Telecommunications Standards Institute (ETSI) According to their own description, ETSI[13] "produces globally-applicable standards for Information and Com-

[12] http://www.converge-online.de/
[13] ETSI website: http://www.etsi.org

munications Technologies (ICT), including fixed, mobile, radio, converged, broadcast and internet technologies”. The organization is listed as non-profit and recognized by the European Union as a standardization institute. ETSI brings together 700 member organizations from 62 countries around the world. Standards can be accessed free of charge. As in most organizations of this type, work is carried out in committees. Each committee forms one or more working groups that consist of technical experts from member organizations. They maintain a work program which leads to one of the deliverables of the types European Standard (EN), ETSI Standard (ES), ETSI Guide (EG), Special Report (SR), ETSI Technical Specification (TS), ETSI Technical Report (TR) or ETSI Group Specification (GS)[14].
The ETSI Technical Committee on Intelligent Transportation Systems (TC ITS) is responsible for developing communication standards for the Car2X domain and is divided into five working groups:

- **WG1** develops “ETSI deliverables on the application requirements and services”. In particular, this working group came up with a “Basic Set of Applications” (BSA) based on C2C and C2I communication. These are especially important for this thesis and will be presented in detail in Chapter 3.
- **WG2** “shall develop the overall architecture and address cross (OSI) layer issues”. The reference architecture will be presented in the next section.
- **WG3** is concerned with information transfer in the Car2X domain. Communication in an ad-hoc manner is especially interesting for this thesis. Therefore, Section 2.2.2 provides a wider view on this topic, not limited to ETSI standards.
- **WG4** is concerned with low-level questions like frequency allocations.
- **WG5** elaborates on security issues related to Car2X communication.

It is noted that ETSI TC ITS is meant to establish partnerships with other standardization institutes that address similar topics.

Car2Car Communication Consortium (C2C CC) The Car 2 Car Communication Consortium[15] is a non-profit institution equipped and supported by major European car companies, equipment suppliers and research institutes. 12 vehicle manufacturers, 22 associate members and 28 development members are part of this network. The activities of the C2C CC are focused on the contribution of C2C- and C2I-based technologies and applications. Therefore, they explicitly support ETSI TC ITS. The C2C CC also describes worldwide activities, i.e. harmonization of activities in the United States of America and Japan where notable effort is also put into establishing standards in the Car2X domain. One of the main goals is to establish the technology on the market by

[14] ETSI deliverable types: http://www.etsi.org/standards/standards-creation-process/etsi-deliverable-types
[15] Car2Car Communication Consortium website: http://www.car-to-car.org

supporting major car manufacturers. Most of the C2C CC documents are not available to the public, but they provide a "Manifesto" which describes architectures and scenarios in the domain. Both will be presented within this chapter, their architecture in the next section, and use cases and scenarios in Chapter 3.

Institute of Electrical and Electronics Engineers (IEEE) A more general institution active on a global level is IEEE[16]. According to their own description, "IEEE is the world's largest professional association dedicated to advancing technological innovation and excellence for the benefit of humanity". IEEE currently has over 400,000 members in more than 160 countries. Within IEEE, the "IEEE Standards Association (IEEE-SA) is a leading consensus building organization that nurtures, develops and advances global technologies, through IEEE".

IEEE developed a number of global communication standards and is mainly mentioned here because of the fact that IEEE 802.11 has become an integral part of wireless networking. This standard has a number of derivates which are named by letters a,b,g or p. Consumer devices use, for example, 802.11 b/g/n for wireless communication. With 802.11a and especially the amendment 802.11p, tailored to wireless vehicular ad-hoc networking, IEEE is indispensable to the success of vehicular ad-hoc networking. These standards will be described in Section 2.2.1.

International Organization for Standardization (ISO) Like IEEE, ISO is working on a global level but, in contrast their standards are not limited to technology but also include business and processes. ISO is a network of national standards bodies and currently counts 164 members in countries all over the world. In contrast to IEEE, most ISO institutes are either under the control of the government or a governmental institution in their respective country. Most ISO standards are not freely available but rather have to be bought from their website.
Technical committee TC 204 is responsible for standardization of all aspects in the ITS domain. Its WG 16 is working on wide-area communications/protocols and interfaces. Here, 8 sub-working groups are developing different aspects of "Communications Access for Land Mobiles" (CALM[17]) in the C2C, C2I and C2Internet domain. They work in close collaboration with ETSI, and therefore the architectures presented in the next section can be transferred to the ones developed by ISO CALM.

World Wide Web Consortium (W3C) The W3C[18] is developing protocols and guidelines for the World Wide Web. Well-known standards issued by the W3C are for example HTML, XML, OWL, or SVG. The work of the consortium is guided by two design principles: *web for all*, e.g. results from the W3C should be available to anyone regardless of "hardware, software, network infrastructure, native language, culture, geographical location, or physical or mental ability"[19] and *web on everything*, e.g. mobile phones,

[16]IEEE website: http://www.ieee.org
[17]ISO CALM website: http://calm.its-standards.info/
[18]http://www.w3.org/
[19]http://www.w3.org/Consortium/mission.html

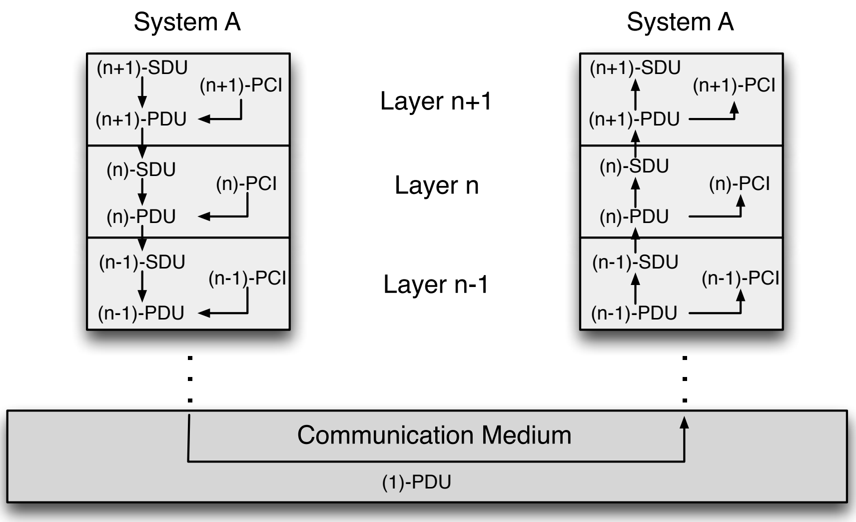


Figure 2.8.: Communication paradigm within the Open Systems Interconnection (OSI) model [ISO/IEC, 1994].

interactive television systems, or kiosks. Therefore, the work of the W3C is global and cultural independent.

The W3C is mentioned here because results of a workshop on “Web and Automotive” led to the launch of the *Automotive and Web Platform Business Group*. Since the Web became mobile during the last years, the W3C puts efforts in transferring this process also to the connected car. The focus of this business group is to determine and to expose vehicle data using Web API(s)[20]. This process is particularly relevant for this thesis. Applications of the Pull Paradigm would benefit from such an open vehicle API since access to this kind of data is currently restricted.

Standardized Architectures

Numerous activities in the first decade of the 21st century have been aimed at standardization of C2X technology and architectures. This section presents the most relevant architectures for this thesis and describes similarities and differences. While it provides an overview of complete C2X network architectures, it focuses especially on network stacks used in vehicles for communicating with the environment.

The OSI model In the following a short introduction into the Open Systems Interconnection (OSI) model is given. Detailed information on this topic can be found in state-of-the-art literature [ISO/IEC, 1994], [Day and Zimmermann, 1983]. However, since the presented standardized C2X architectures are based on this model, it is introduced

[20]http://www.w3.org/2012/08/web-and-automotive/Overview.html

7. Application	Hosts applications and provides them access to the OSI
6. Presentation	Provides transparent access to different data representation schemes
5. Session	Provides mechanisms for organizing interactions between application processes
4. Transport	Ensures data integrity and reliability between two endpoints
3. Network	Provides means for connecting, maintain and terminate connections between two endpoints
2. Data link	Transfers data between entities and possibly correct errors on the physical layer
1. Physical	Provides access to the physical medium

Figure 2.9.: The seven abstraction layers of the OSI model [ISO/IEC, 1994].

here in order to give the reader a comprehensive and complete understanding of the topic.

The OSI model describes a standardized architecture of a communication system by introducing the concept of seven abstraction layers. Each layer groups similar functionalities and communicates with the layers immediately above and below it. Usually, the lowest layer (1) is close to the physical communication medium of the system while the highest layer (7) serves and hosts applications. Often, this layer model is referred to as "stack" because in order to send data from the application to another entity within the network, a packet has to traverse all layers from seven to one on the sender side and from one to seven on the receiver side. This is modeled as follows: the Service Data Unit (SDU) of layer n-1 is passed to layer n, on which the data is then encapsulated with the layer's Protocol Control Information (PCI), e.g. header and footer are added. This process transforms the SDU into a Packet Data Unit (PDU) of layer n. A SDU effectively represents the payload of layer n-1. Upon reaching layer 1, the packet is sent over the communication medium. The process of receiving a packet works in the opposite direction: layer n+1 removes its specific headers and footers, processes the data and forwards the SDU (effectively a PDU of layer n) to the layer above. Figure 2.9 shows the seven layers of the OSI model.

Layer 7 – Application Layer 7 is at the top of the OSI reference model and provides no services to other layers. Here, applications and their protocols like http, ftp or telnet reside. The layer allows applications to access the OSI in a well-defined manner, transparent to the respective transport, network and physical characteristics of the communication technology.

Layer 6: Presentation The presentation layer provides ready-to-use access to different data representation formats for the application layer. The idea is to relieve layer 7 from dealing with details of numerous syntax forms, e.g. it unpacks data and converts it to a format, that is commonly understood on application level. A concrete example would be converting between different character encodings. Usually, encryption and decryption are also done on this level. The presentation layer still treats data in a structured way. All layers below split data into packets and do not possess an understanding of the content.

Layer 5: Session The session layer provides mechanism for communication between two entities within a network. An application process may talk to several other entities at the same time. Layer 5 ensures that the correct entities are able to start, stop and maintain their communication session with each other while others are running simultaneously. An example would be a chat session: Layer 5 ensures that the messages are not confused between different sessions.

Layer 4: Transport Layer 4 is responsible for ensuring correct data transfer between two nodes in a communication network. It also provides functionality for data integrity, flow control, multiplexing and possibly addressing of nodes. The standard reads, that the transport layer relieves “the upper layers from any concern with providing reliable and cost-effective data transfer”. A popular example is the Transfer Control Protocol (TCP) which detects defective and dropped packets and ensures their re-sending. Layer 4 constitutes the basis for routing protocols, which are present on layer 3.

Layer 3 – Network The network layer is responsible for transferring data packets from a sender to a receiver, where in most cases more than zero nodes are between the two communicating entities. This packet forwarding mechanism is called routing and is usually done between concatenated networks. It abstracts from the physical communication medium. Layer 3 also provides host addressing and ensures quality of service according to the parameters requested by layer 3. The popular Internet Protocol (IP) resides at this layer and uses a connectionless communication, e.g. the sender does not acknowledge receipt of a packet. This is done on layer 4 if needed.
In wired networks, routing tables are maintained in order to determine the next hop in each node. This maintenance obviously produces a communication overhead. In wireless networks, where bandwidth is a limited resource, this overhead has to be kept at a minimum, e.g. determining a route before the packet is sent, as is done in wired networks, is not feasible. Also, in vehicular ad-hoc networks, the routing task becomes significantly more challenging, since only the topology formed by the cars in transmission range is known. These restrictions lead to the fact that routing is of utmost importance for applications in the context of the Pull Paradigm. Information needs to be transferred in a stable and reliable way between two nodes in a highly mobile environment. Section 2.2.2 therefore analyzes existing approaches for vehicular ad-hoc networks while Chapter 5 presents advanced algorithms to cope with these challenges.

Layer 2: Data Link The data link layer provides strategies for accessing the physical communication medium, detecting and possibly correcting errors occurring in the physical layer. In wired as well as in wireless networks, only one node is allowed to access the medium at a time. This makes coordination between the nodes necessary. However, it is still possible that two communicating entities access the medium in parallel. Dealing with these collisions is crucial for layer 2. For wireless networks, state-of-the-art algorithms are available; especially in 802.11 the popular *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) algorithm is used [Gast, 2005].

Layer 1: Physical This layer provides the physical means for accessing the communication medium, e.g. electrical, optical, etc. Procedures for establishing and stopping (physical) transmission of signals are also handled. Layer one operates on single bits and manipulates the signal accordingly in order to transmit the bitstream over the communication medium.

Reference Architecture of the C2C CC The members of the Car2Car Communication Consortium (C2C CC) describe in their manifesto [Baldessari et al., 2007] a reference architecture for C2X networks as depicted in Figure 2.10. It distinguishes between three different domains: in-vehicle, ad-hoc and infrastructure domain. The *in-vehicle domain* refers to two logical components, the application unit (AU) and the on-board unit (OBU). The on-board unit is responsible for managing C2C and C2I communications and providing these services to applications in a well-defined manner. In the in-vehicle domain, OBUs form a mobile ad-hoc network (MANET) which is characterized by direct communication of OBUs without a coordinating infrastructure. On the communication level, an OBU fulfills low-level tasks like forwarding data on behalf of other OBUs, congestion control, ad-hoc routing, and so on. According to C2C CC, an OBU should at least be equipped with one short-range wireless communication technology. They explicitly mention 802.11p in this context. Other technologies like consumer WLAN are described as optional. Interestingly, they mention ad-hoc communication using 802.11p only in conjunction with safety-related applications. This concept is also reflected by the communication architecture of an AU. This second component of the in-vehicle domain hosts all applications which can access the communication services of an OBU. Typically realized as a dedicated device, it is connected by a wired network with an OBU. Furthermore, nomadic devices can also serve as an AU, e.g. smartphones. Connection to the OBU is then dynamically established via Bluetooth, WLAN or even wired network. It should be pointed out that the concept of AU and OBU is logical, which means that both can even reside in the same physical device.

The *ad-hoc domain* is formed by vehicles carrying an OBU and roadside units (RSUs). In contrast to an AU, the location of an RSU is fixed. It is also equipped with at least one short-range communication technology (802.11p). The C2C CC sees the primary task of an RSU as being in the domain of safety-related applications. As it can be seen in Figure 2.10, an RSU is also connected to other networks and can provide

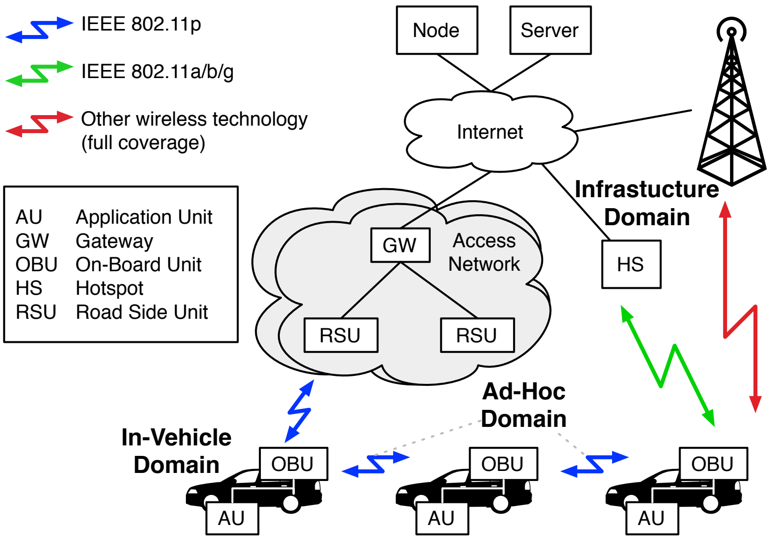


Figure 2.10.: Car2X Reference Architecture as standardized by the Car2Car Communication Consortium. Adapted from [Baldessari et al., 2007].

these services to vehicles. Other functionalities are given by the C2C CC: An RSU is to extend the range of the ad-hoc network by forwarding messages to other OBUs and distributing information to traffic management centers. It can run safety-related applications and use C2I communication in order to notify drivers about critical road situations and communicate with other OBUs either using ad-hoc or infrastructure communication.

Public authorities and operators of public/private hotspots form the *infrastructure domain*. The C2C CC describes mainly non-safety applications in this domain. For example, applications can use mobile radio, RSUs or hotspots to communicate with these entities.

The reference architecture of the C2C CC describes two communication principles. First, it should provide spatial and timely dissemination of information within the ad-hoc network. Second, it should make it possible for mobile nodes to communicate with wired networks. The C2C CC points out that information dissemination is primarily used for safety-related applications, but is not limited to these. However, they provide no further information on other applications of the first communication principle.
In an ad-hoc network, they distinguish between *receiver-centric* and *sender-centric* information dissemination. Common communication principles are in most cases sender-centric, which means it is the sender's decision to determine the target of a message. In terms of vehicular ad-hoc networks, the sender communicates a geographical area for which the information is valid. A receiving node then checks it against its own position and either forwards the message to the target area or broadcasts it within this area. When working according to the receiver-centric principle, the receiver determines if the

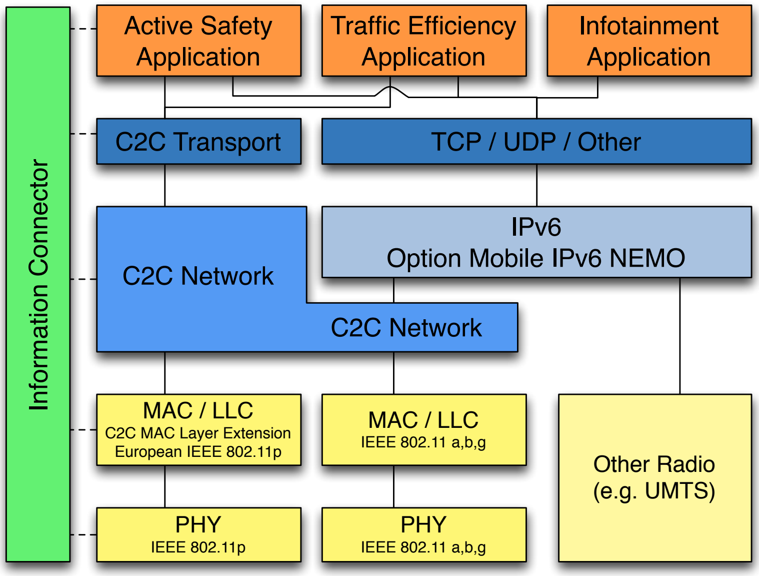


Figure 2.11.: Protocol architecture of an On-board Unit (OBU) as standardized by the Car2Car Communication Consortium [Baldessari et al., 2007].

message should be further broadcast by merging the received information with its own state.

Particularly interesting for this thesis is the detailed structure of an OBU, as it helps to understand how communication for applications is structured within their architecture. It is shown in Figure 2.11. The layer scheme is derived from the well known OSI model as described in detail in [ISO/IEC, 1994]. On the physical layer, three different communication schemes are to be provided: consumer WLAN, 802.11p and other cell-based radio, e.g. UMTS or LTE. C2C CC explicitly mentions the importance of 802.11p for active safety applications (cf. Chapter 3).

The MAC layer is not finally approved at the point this thesis was submitted. However, the basis is IEEE 802.11 MAC with some simplifications and enhancements. These enhancements, for example, include estimation about the current channel load and strategies to prevent medium congestion. All this information is also provided to the upper layers. Simplifications are that authentication, de-authentication and privacy are omitted in the C2C CC specification. However, the standard MAC algorithm of 802.11 is used.

In the standardized architecture of the C2C CC, the network layer is responsible for disseminating information in the vehicular ad-hoc network. It defines three different data delivery schemes:

- *Geographical broadcast*: Information is only valid within a certain geographical area which is specified in the packet itself. The broadcast algorithm tries to keep information within this area even when vehicles are entering and leaving very quickly. If the vehicle sending this information is not within the area of validity, the algorithm tries to deliver it to its target first and then floods the network in order to disseminate it.

- *Event driven single hop broadcast*: a data packet is disseminated to all OBUs and RSUs within transmission range of the sending vehicle.
- *Beacon packets* are a special case of event-driven single-hop broadcasts and are periodically sent by an OBU to the neighborhood. Typically, these packets contain additional application data like the speed and direction of the sending vehicle.

A very vague hint is given regarding unicast communication within the network layer. The C2C CC manifesto describes the possibility to exploit vehicles’ movements and positions in order to realize a multi-hop unicast communication and therefore deal with the fast topology changes within the ad-hoc network. However, their manifesto is intended to give a high-level overview rather than specifying algorithms for information dissemination in vehicular ad-hoc networks.

Also in the definition of the transport layer, the C2C CC manifesto briefly describes the concept of unicast-based connection-oriented communication over the ad-hoc network in connection with requirements for safety applications. However, here again no description of actual implementations of this concept is given. The transportation layer is mainly responsible for providing services like multiplexing and de-multiplexing and combining data from different applications. The C2C CC manifesto only mentions safety applications as clients for these services.

The C2C CC application layer of an OBU as specified in Figure 2.11 provides common services to applications, namely the maintenance of the local database, sending and receiving procedures, message processing as well as access to local sensor data of the vehicle. Typically, well-defined access to the CAN-bus messages is provided.

The concept of an *Information Connector* provides cross-layer communication between the different layers in an “efficient and well-structured manner”. Here too, no further details are given how this well-structured manner is achieved. It should be noted, that the C2C CC architecture of an OBU does not specify a security concept.

Looking closer at the architecture concept in Figure 2.11 reveals that every application can send packets using either consumer WLAN (IEEE 802.11 a,b,g) or the mobile radio connection. However, it is not intended that infotainment applications use C2C transport, C2C network and 802.11p in order to communicate with other vehicles directly. This is an important point in connection to this thesis: According to the C2C CC, it is not possible for infotainment applications to access the vehicular ad-hoc network in order to provide services. Furthermore, active safety or traffic efficiency applications can use several wireless communication technologies in parallel. This might, for example, be

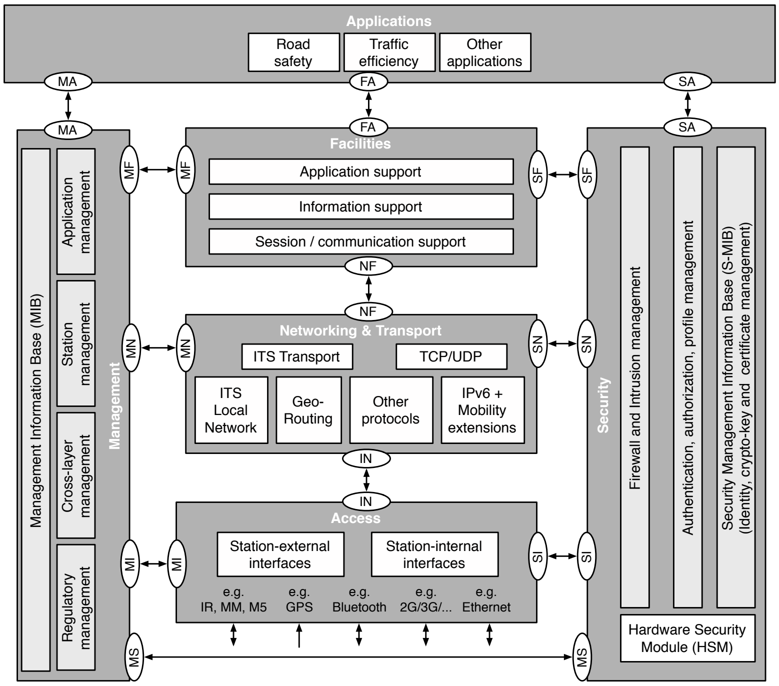


Figure 2.12.: ETSI ITS station architecture. Adapted from [European Telecommunications Standards Institute (ETSI), 2010a].

interesting for notifying other vehicles of a dangerous road situation using 802.11p while sending this information to traffic management centers using the mobile radio connection.

ETSI EN 302 665 ETSI documented in their European standard EN 302 665 [European Telecommunications Standards Institute (ETSI), 2010a] the communications architecture of a C2X system. Compared to the architecture of C2C CC, it is kept more abstract. However, a large number of research projects comply with their ITS station reference architecture which is depicted in Figure 2.12. Its layer structure is derived from the OSI model [ISO/IEC, 1994] but was extended in several ways for ITS applications. Taking a closer look at Figure 2.12, it becomes clear that functionalities of the OSI layer model have been combined into the three blocks in the middle:

- Facilities contain functionality from OSI layers 5 (session) and 6 (presentation)
- Networking and transport contain functionality from OSI layers 3 (network) and 4 (transportation)
- Access contains functionality from OSI layers 1 (physical) and 2 (data link)

The application layer on top represents OSI layer 7. Information flow between two layers/components is symbolized with bidirectional arrows. Connections' names in ovals are just the first letters of the two involved components, e.g. *MA* means information exchange between management component and application layer. Similar to the architecture of an OBU of the C2C CC, a management entity is present for cross-layer communication. ETSI also considers a security module which is available to all layers. Each entity participating in a C2X network contains an implementation of the ITS reference architecture (and/or parts of it) and possibly other functionalities specific to the communication entity. Within this scope, ETSI distinguishes four sub-systems as follows:

- Personal ITS sub-system
- Vehicle ITS sub-system
- Central ITS sub-system
- Roadside ITS sub-system

These four sub-systems are symbolized in Figure 2.13 and are connected to the C2X community using *ITS peer-to-peer communications*. ETSI EN 302 665 provides no details on the physical communication technologies used. A *personal ITS sub-system* contains a complete implementation of the ITS station reference architecture and offers communication services within a *hand-held device*. This could be a smartphone, tablet or even a laptop. Furthermore, it can serve as HMI. The ITS station connects to the HMI via the internal network. Note that in this sub-system the ITS reference architecture is implemented in a single device.

Compared to the personal ITS sub-system, the composition of a *vehicle ITS sub-system* is different. It contains a unit for communication (ITS stations router) which implements layers access and networking & transport. The ITS station host serves as a runtime environment for applications and implements layers facilities, networking & transport and access. Besides that, the vehicle ITS station gateway connects the ITS station to the car’s internal network. This architecture can be easily mapped to an AU and OBU of the C2C CC. Here, the OBU also contains functionalities to connect to the car’s network.

The *Central ITS sub-system* also contains an ITS station host comprising all elements of the reference architecture. An additional gateway connects the sub-system to the other systems on the facility layer. A third component, the border router, connects to entities outside the scope of the central ITS sub-system on the network & transport layer. ETSI points out that this approach may avoid the ITS security layer. Besides that, the functionality is similar to the ITS gateway. However, no indication as to the reasons for circumventing security is given in the document. One reason might be that the ITS gateway connects to related ITS applications, and is therefore bound to the security model, while the border router communicates with non-ITS entities.

The fourth component in ETSI’s model contains the most complex combination of their ITS reference architecture. In the ITS roadside sub-system a mandatory ITS station host is present. An ITS gateway provides the functionality to connect the roadside

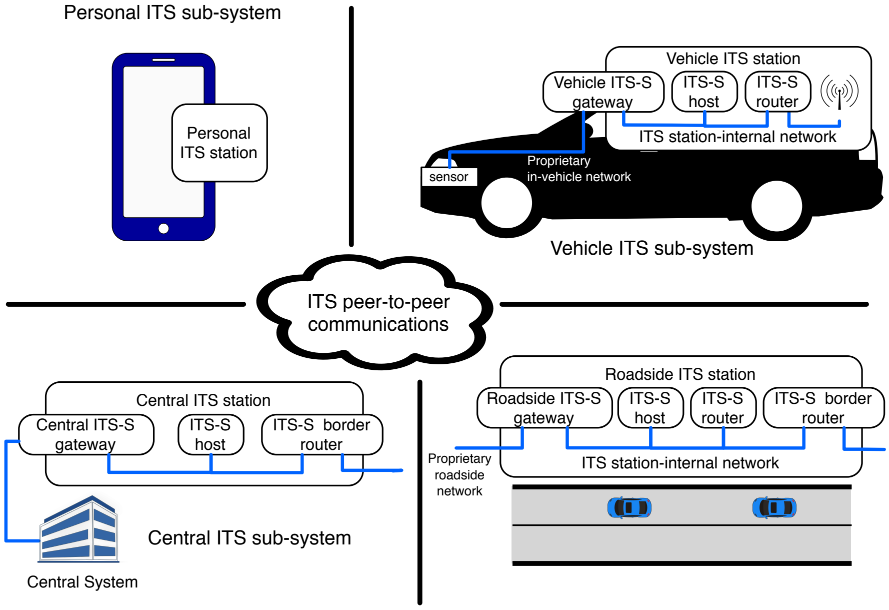


Figure 2.13.: ETSI ITS sub-systems (adapted from [European Telecommunications Standards Institute (ETSI), 2010a]).

system, e.g. inductive loops or variable message signs (VMS). An additional ITS station router is used for communication between the sub-system and roadside units. Another border router provides access to other entities, again with possible avoidance of the security layer. All components are connected by the station's internal network.

Up to this point, the architecture of ETSI introduced four functional components which are all derived from their general ITS station reference architecture. An ITS station host is a complete implementation of this reference architecture and contains all applications. It is comparable to the application unit of the C2C CC. An ITS station gateway features functionality up to the facility layer and connects other OSI protocol stacks at layer five to seven. In the sub-systems presented, this component is used to communicate with proprietary networks like the vehicle bus. However, this is not mandatory for complying with the standard. An ITS station router and border router provide similar functionality. They connect other entities of the C2X network at OSI layer 3 and are also associated with the ITS station's internal network in order to speak to other functional components. However, the ITS station border router might not follow the security and management principles of ITS, probably to communicate with non-ITS entities.

AutoNet Architecture [Kosch et al., 2012b] describe a generic reference architecture for C2X networks. In their terminology they call this architecture "AutoNets". Similar to the other two presented architectures, they distinguish several domains within a Car2X

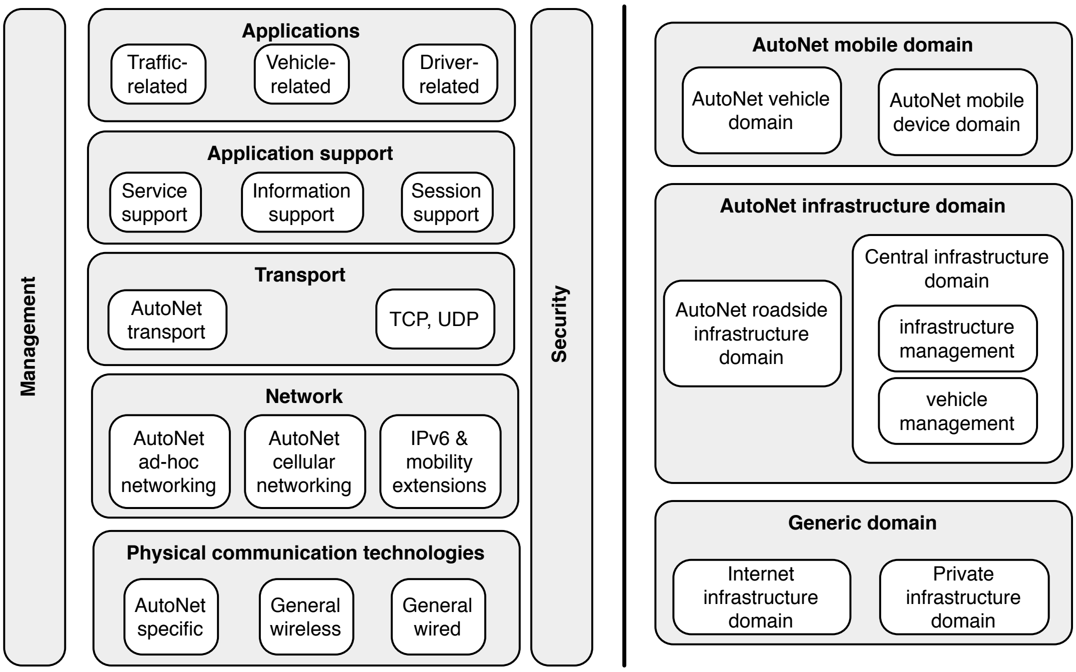


Figure 2.14.: **Left:** AutoNet architecture, **right:** AutoNet domain view. Picture adapted from [Kosch et al., 2012b].

network and also show how their domain model can be mapped to others. Figure 2.14, right shows the domains “AutoNet mobile”, “AutoNet infrastructure” and “Generic”.
Within the mobile domain [Kosch et al., 2012b] distinguish between the vehicle and the mobile device domain. The first subsumes all vehicles like cars, trucks or motorcycles while the latter includes nomadic devices like portable navigation systems, smartphones, etc. The infrastructure domain provides AutoNet specific technology in order to communicate with the mobile domain. The roadside infrastructure sub-domain comprises stationary communication devices within the field, like roadside units or C2X-enhanced traffic lights and signs. The central infrastructure sub-domain provides centrally-managed services and applications. In an implementation, this would be traffic management and vehicle management centers.
The third, “Generic” domain describes services and applications which do not depend on C2X-specific communication technology. The two sub-domains describe communication with entities using common internet technologies and access to private networks, like service networks of a specific OEM. The authors describe their architecture as being inspired by ETSI, which becomes clear when looking at the described domains. However, Kosch et al. point out that in contrast to ETSI they distinguish two different types of entities within the central infrastructure domain. Infrastructure and vehicle management are two different tasks. This fact is not considered in the Central ITS Station.

The similarity to ETSI [European Telecommunications Standards Institute (ETSI), 2010a] becomes even more clear when looking at the “AutoNet Generic Reference Protocol Stack", which is presented in Figure 2.14, left. They adopt to a substantial extent the layer model with cross-layers for management and security. The layer architecture follows the OSI model presented in Figure 2.8 but, similar to ETSI, combines several layers into one.

The **application layer** holds all applications. It should be noted at this point that Kosch et al. introduce a different classification of applications. This is addressed in Chapter 3. The **application support layer** provides the session management in compliance to the OSI session layer. Service support serves applications for awareness of service announcements from other nodes in the network. Information support provides applications with means for assessing the status of the network. The **transport layer** is described according to OSI as providing end-to-end communication between nodes but tailored to the needs of vehicular networks. The **networking layer** is also compliant with OSI but features more than one transport protocol. The architecture comprises several communication technologies on the **physical layer**.

Up to this point, several generic architectures for Car2X networks have been presented. Besides individual details, it can be concluded that all comply with the 7-layer OSI model presented in this chapter. However, all presented architectures subsume several layers of OSI into one, e.g. the application support layer of Kosch et al. comprises session and presentation layers of OSI. Also, all architectures combine more than one communication technology in their layer model. It should also be noted here that the presented architectures describe concepts, not implementations. The definition and standardization of these concepts can be seen as the continuous effort to harmonize individual activities carried out in several independent research projects.
The next section provides an analysis of wireless communication technologies. The complete Car2X network stack which was introduced by the architectures above is discussed in detail.

2.2. Car2X Networking: Technology and Simulation Environments

2.2.1. Wireless Communication Technologies

This section introduces available wireless communication technologies commonly applied to the C2X domain. The focus is on a high-level description as well as insights on advantages and disadvantages with respect to application scenarios within the Push and Pull Paradigms. A more technical description can be found in the state-of-the-art literature. Within the C2X domain, wireless communication technologies provide the basis for a wide range of use cases (cf. 3). Established distinction criteria are long- and short-range communication. Furthermore, broadcast services are currently available.

	Wireless Communication Technologies								
	Short Range		WLAN (802.x)			WiMAX	Mobile Radio		Broadcast
	RFID	Blue-tooth	.11a	.11b/g	.11p	802.16	UMTS/ HSDPA	LTE	DVB-S/T
Transfer Rate (Mbit/s)									
Transmission Range (m)									
Frequenzy (125GHz, 5GHz, 2.4GHz, 30kHz)									
Ad-hoc mode	✗	✗	✔	✔	✔	✔	✗	✗	✗
Optimized for low-latency	✗	✗	✗	✗	✔	✔	✗	✔	✗
Uplink Channel	✔	✔	✔	✔	✔	✔	✔	✔	✗
Downlink Channel	✔	✔	✔	✔	✔	✔	✔	✔	✔
Considered in Pull Paradigm	✗	✔	✔	✔	✔	✔	✔	✔	✗
Suitability for Pull Paradigm									

Figure 2.15.: Overview and assessment of available wireless communication technologies with respect to the Pull Paradigm. Note, that the table gives no exact values for transfer rate and transmission range since they are subject to fluctuation. It is only intended to compare the technologies. Legend: ✓: feature is present, ✗: feature is not present.

Mobile Radio Communication

Mobile radio communication takes an important position in vehicular ad-hoc networks. This communication technology enables long-range and robust data communication, typically providing internet access to cars.

A breakthrough for mobile phones was reached in the 1990s when analog services were replaced by the fully-digital GSM[21]. This second generation (2G) of cellular mobile telephony is the most widespread standard for mobile telephony and data services today. Initially only providing phone and short message service, GSM was augmented by GPRS[22] and EDGE[23] in order to provide packet-oriented communication. Today, the third generation (3G) UMTS[24] is available at least in densely populated areas and enables download rates up to 384 kbit/s. 2G has reached almost area-wide coverage even in rural areas. All presented standards constitute one of the building blocks of Car2X Communication.

[21] Global System for Mobile Communications
[22] General Packet Radio Service
[23] Enhanced Data Rates for GSM Evolution
[24] Universal Mobile Telecommunications System

For long-range communication in the automotive domain, the state of the art is the latest standard for cellular communications, "Long Term Evolution" (LTE) or the 4th generation of mobile networks. Although not yet widespread, LTE theoretically provides download rates up to 300 Megabits per second and lower latency times in comparison to UMTS. Cutting-edge Car2X research is currently connecting this service in order to provide location-based and added-value services to the driver. Applications in focus are finding parking spaces, community events, weather, and traffic information. Information is typically updated in periodic intervals using a point-to-point communication between the vehicle and the individual data providers. However, it is not common to download information on demand or to stream content, let alone to have real-time capability. It is expected that this will be supported by the 5th generation of mobile radio communication (5G) beyond 2020.

The very nature of today's cellular systems decreases bandwidth with an increasing number of participants associated with the same access point: All members of a cell share the available bandwidth. Furthermore, compared to C2C ad-hoc communication, direct information exchange between two nodes (cars) is not feasible: Packets are always routed via the access point involved. This causes an increase of latency as time for handshaking and routing has to be taken into account. Thus, latency is significantly higher compared to ad-hoc network communication introduced in the next section.

Wireless Local Area Networks

Wireless Local Area Networks (WLANs) have become widely accepted today in home and business environments for connecting computers and laptops to the Internet. They provide high-speed data transfers, typically with comparatively low latency times. This short-range communication technology is based on IEEE standard 802.11 and, similar to mobile radio communications, uses the concept of cells: An access point is utilized as a central hub for information exchange between the nodes and remote networks. These cells are typically smaller than they are for mobile radio communication ($<$100 meters) but latency also is usually lower. Because of the cell architecture, the same restrictions as for mobile phone networks apply. Latency increases and transfer speed decreases with increasing number of nodes. Since coverage of access points is fragmentary or almost non-existent outside cities, and cell size relatively small compared to mobile radio, an application to vehicular environments is not feasible and was therefore not considered in the past. However, there exists a so-called "ad-hoc mode" for all IEEE 802.11 derivatives. In contrast to the described "infrastructure mode", it allows data exchange between two nodes directly in a self-organized manner without a detour through an access point. Thus, ad-hoc mode does not need an area-wide coverage of infrastructure in the form of access points. This made IEEE 802.11 a promising communication technology for a series of scenarios in C2X research.

However, these obvious benefits are accompanied by the fact that the ad-hoc mode of widespread 802.11 derivates for home networks is not optimized for highly mobile nodes.

802.11 b/g and others feature comparatively-high latency times until a communication channel is established. Furthermore, the retention period on a channel is limited by the number of active sending parties. A node is only able to access the medium when no other participant is sending. In order to overcome these limitations, a working group was founded by IEEE. A new derivate to the 802.11 set of standards was developed: IEEE 802.11p. It specifies physical and medium access control layers (MAC) with a minimum overhead for establishing communication channels. So far, 802.11p has not been finally approved by IEEE and is officially a “draft” standard. Because of its characteristics specifically tailored to the needs of vehicular ad-hoc communication, it is believed to be the most promising technology for a series of scenarios in C2X research where near-real-time capabilities are crucial. It provides the basis for warning drivers about obstacles on the road, traffic jams ahead, and broken-down vehicles. Furthermore, ad-hoc infrastructure-to-vehicle communication is used to provide traffic efficiency services, like green light optimal speed advisories. A more detailed discussion on these scenarios follows in Chapter 3.

In contrast to consumer WLAN 802.11 b/g working on the 2.4 GHz frequency band, 802.11p operates between 5.850 and 5.925 GHz. Channels are 10 MHz wide. This also represents a difference from consumer WLAN where 20 MHz channels are established. 802.11p emanated from the activities of workgroup 802.11a. Both standards are very similar to each other. The main differences are frequency allocations of 5.180 to 5.825 GHz (11a) vs. 5.850 to 5.925 GHz (11p), and channel width (20 MHz for 11a vs. 10 MHz 11p).

Because of its technical characteristics described in this section, 802.11p is another suitable communication technology for Pull applications. However, the lack of organizing infrastructure makes it difficult to establish information exchange between two nodes that are not in each other’s transmission range. Routing data packets between these two entities might also be impossible due to low traffic density. Stable and low latency communication between multiple cars is a key requirement for applications within the Pull Paradigm. Therefore, this problem is addressed in more detail in Chapter 5 where beyond state-of-the-art algorithms for information flow in C2X networks are elaborated. It should also be noted that even though 802.11p provides near-real-time capabilities and low latency, applications that require on-demand access to information are not considered.

WiMAX

WiMAX (Worldwide Interoperability for Microwave Access) is an IEEE standard (802.16) and is operating at comparably high frequency bands, 10 to 66 GHz and 2 to 11 GHz for 802.16a. Up to 1 Gbit/s can be transferred using WiMAX. The intended application fields of this wireless communication technology is to provide high-speed Internet access in areas where wired technologies, for example, DSL are not available. The high transfer rates and cost-efficient availability makes it also interesting for an application to the car. A study by [Msadaa et al., 2010] compares IEEE 802.11p and 802.16e in a simulated environment. In a nutshell, WiMAX offers higher data rates and very low delays. However, 802.11p provides very short latency times and is more suited for low traffic loads. Depending on the application, this wireless communication technology might be interesting for the Pull Paradigm. It is particularly useful for communicating with the outside environment.

Broadcasting

Broadcast technologies have been around for a long period of time. Analog techniques for broadcasting radio programs (AM[25], FM)[26] are still in use today. However, digital broadcast systems are the state of the art. Digital Audio and Video Broadcast (DAB, DVB) are widespread technologies to provide traffic information as well as entertainment to drivers. In contrast to WLAN and mobile radio communication, the very nature of broadcast systems features a unidirectional information transfer from one source to multiple receivers. One of the key advantages of this technology is full availability in all parts of Europe. While data rates are high in such scenarios, latency times and the lack of an uplink channel are considered key drawbacks of broadcast systems. Therefore, broadcasting technology is not of particular interest for Pull applications.

Short-Range Communication (SRC)

Further widely accepted short-range communication technologies are Bluetooth, RFID and even communication between keys and vehicles. Bluetooth in particular an established technology for connecting smart phones to in-vehicle platforms in order to access sensors and data. Compared to long-range communication technologies, SRC typically provides lower data rates and high latencies. Even with these limitations, SRC could be a valuable communication technology for the Pull Paradigm. Since today's common smartphones are normally equipped with Bluetooth, personalized information and the phone's sensors are a valuable source for a number of applications.

Conclusions on Available Communication Technologies

Figure 2.15 summarizes the various wireless communication technologies presented in the previous section. It can be concluded that a variety of available technologies exist, but not all can be applied to the Pull Paradigm. At this point, broadcast systems like DVB-T and DVB-S are excluded from further consideration due to the lack of an uplink channel and comparatively high latency times.
Furthermore, RFID of the SRC category will not be considered, since it provides low transfer rates and is not optimized for low latency. However, Bluetooth might be interesting depending on the individual application since it provides access to already-existing data and sensors on smartphones. Therefore, Bluetooth will be considered for the purposes of this thesis.

WLAN constitutes a promising technology for the Pull Paradigm. In particular, 802.11p is optimized for data transfer at high speed and low latency and will be one

[25]amplitude modification
[26]frequency modulation

of the key technologies for Pull applications. Depending on the individual use case, 802.11a/b/g can also be considered. The same applies for WiMAX which offers high transfer rates and can particularly be used for C2I communication.

Mobile Radio provides stable internet connections even in mobile environments. Only LTE is optimized for low latency, but depending on the intended applications both technologies may be valuable. Note that the presented technologies all have their individual advantages and drawbacks. Therefore, it might also be interesting to consider a technology combination at some point in order to always use the optimal communication technology for an application.

2.2.2. Communicating on the Fly: The Ad-hoc Challenge

So far, a high-level overview of wireless communication technologies and standardized architectures of complete C2X communication systems has been given. The following will now focus on wireless ad-hoc networking described in the previous chapter and explain in more detail the approaches that cope with the special characteristics of vehicular ad-hoc networks. The analysis is structured in a layered approach according to the architectures presented in Section 2.1.5. Here, the focus is on the network and presentation/session layers. As this section will show, there is a need for augmenting these layers in order to work in the context of the Pull Paradigm.

Networking

In a highly mobile network where the topology is constantly changing and nodes move out of transmission range all the time, it becomes challenging to deliver data reliably from a source node to a destination node that is not in direct communication range. As described in Section 2.1.5, this routing procedure is one of the key functionalities of the network layer. In traditional wired networks, routing tables contain information about the networks to which a node is connected. These tables are stored and maintained in every node in order to determine the correct forwarding node for an incoming packet. State-of-the-art protocols take care of this task very reliably. For example, the popular Internet Protocol (IP) has been established worldwide. However, routing algorithms are still in the focus of research in vehicular ad-hoc networks. This section presents various routing schemes and analyzes their applicability to vehicular ad-hoc networks.

One common method of classifying routing algorithms is the point in time when a route is calculated [Füssler et al., 2002]:

- *Proactive routing*
- *Reactive routing*
- *(Greedy) Position-based routing*

Most algorithms for wired networks belong to the first category of *proactive routing*. Here, routes are constantly maintained and stored in tables even if they are not requested. At the time a packet has to be forwarded, the appropriate route can be looked up in the table. Popular routing schemes use either distance vector [Hedrick, 1988] or link state algorithms [Mcquillan et al., 1980]. Both will not be discussed here in detail as they play only a minor role for ad-hoc networks. The interested reader is pointed to the related standards which contain these algorithms: RFC 2328 [Moy, 1998] and RFC 5340 [Coltun et al., 2008]. Each of them is integrated in IPv4 and IPv6 respectively, two major protocols on the network layer used for point-to-point communication over the internet.
Proactive routing approaches generate a high routing overhead (payload in relation to transmitted data for route discovery and maintenance). Therefore, they can mostly be found in wired networks, where bandwidth is not considered as the limiting factor as it is for wireless networks. Because of the special characteristics of vehicular ad-hoc networks, the overhead of maintaining a routing table can easily block the communication channels. This is one of the reasons why there are not many proactive routing protocols that have been applied to vehicular ad-hoc networks. Two notable examples are RBVT [Nzouonta et al., 2009] which has an option for proactive routing (RBVT-P) and the Optimized Link State Routing Protocol (OLSR) [Clausen and Jacquet, 2003], which is an adapted version of the link state approach.

Reactive routing computes and maintains only requested routes. Before the packet is sent, algorithms of this class initiate a route discovery process which basically floods the network with routing request packages. If successful, the packet is then sent along the route while the source node usually maintains this route until the destination node becomes unavailable [Royer and Toh, 1999]. Popular representatives of this class are for example

- Ad-hoc On-Demand Distance Vector Routing (AODV) [Charles E. Perkins, 1997]
- Connectivity-Aware Routing (CAR) [Naumov and Gross, 2007]
- Dynamic Source Routing (DSR) [Johnson and Maltz, 1996]
- GVGrid [Sun et al., 2006]
- ROMSGP [Taleb et al., 2007]
- Geographic Source Routing (GSR) [Lochert et al., 2003]
- Dynamic Source Routing (DSR) [Johnson and Maltz, 1996]
- ROVER [Kihl et al., 2007]
- MURU [Mo et al., 2006]
- RBVT-R [Nzouonta et al., 2009]

In comparison to proactive routing approaches, reactive strategies seem to fit the needs of vehicular ad-hoc networks in terms of routing overhead. However, the reactive approach introduces a delay for sending packages as unknown routes have to be discovered first. Moreover, as Füßler et al. found in their comparison of routing strategies [Füssler et al., 2002], the third category is superior to reactive approaches.

(Greedy) Position-based routing can be considered as a special case of the reactive approach since route discovery is only triggered upon request. The difference, however, lies in how the route is discovered. Algorithms of this class make the next-hop decision on every node depending on the local context. In contrast to the presented table-driven schemes, this greedy approach neither requires maintenance of routing tables nor generates waiting times until a route discovery process has been completed. This keeps routing overhead to a minimum. Every representative of this class exploits locations of the surrounding vehicles, hence the name position-based. In vehicular ad-hoc networks this location service is provided by periodic updates by the C2X neighborhood. Greedy position-based routing has drawn attention from a number of researchers in the past. The following list shows popular approaches but makes no claim to be complete:

- Anchor-based Street and Traffic Aware Routing (A-STAR) [Seet et al., 2004]
- Routing Protocol for Emergency Applications in Car-to-Car Networks using Trajectories (REACT) [Van de Velde et al., 2006]
- Greedy Traffic Aware Routing Protocol (GyTAR) [Jerbi et al., 2007]
- Fastest-Ferry Routing in DTN-enabled Vehicular Ad-hoc Networks (FFRDV) [Yu and Ko, 2009]
- Geographic Routing in City Scenarios (GPCR) [Lochert et al., 2005b]
- Movement-Based Routing Algorithm for Vehicle Ad Hoc Networks (MORA) [Granelli et al., 2006]
- Global Perimeter Stateless Routing with Movement Awareness (GPSR-MA) [Granelli et al., 2007]

Besides the greedy approach and exploitation of positions, some algorithms mentioned above make use of meta-information in order to support routing decisions in each node. Obvious information that can be derived from changing positions over time includes speed and heading. A routing scheme can then determine if a possible next forwarder is closer to the intended destination. Another approach is to consider intersections as anchor points for routing decisions. Buildings which block the line of sight to the next forwarder could potentially disrupt wireless communication. By trying to route the packet over street intersections the lower influence of buildings leads to an increased probability of reaching the destination faster in fewer hops.
Traffic density in the close vicinity is also an example of meta-information used by several

approaches in order to generate more reliable routing decisions. It obviously increases the likelihood of finding a suitable next hop. Aside from that, access to map data is also commonly used. For example, A-STAR [Seet et al., 2004] tries to route along major roads on which bus services are run. However, they do not consider specific schedules or times of day when no bus services are available. The following routing protocols make use of some kinds of meta-information for routing decisions:

- Anchor-Based Street and Traffic Aware Routing (A-STAR) [Seet et al., 2004]
- Greedy Traffic Aware Routing Protocol (GyTAR) [Jerbi et al., 2007]
- Connectivity-Aware Routing (CAR) [Naumov and Gross, 2007]
- GVGrid [Sun et al., 2006]
- Movement-Based Routing Algorithm for Vehicle Ad Hoc Networks (MORA) [Granelli et al., 2006]
- Greedy Perimeter Stateless Routing [Karp and Kung, 2000]
- Global Perimeter Stateless Routing with Movement Awareness (GPSR-MA) [Granelli et al., 2007]

Another important aspect of routing schemes is how they recover from errors: for example, when reaching a local maximum, e.g. no appropriate forwarder can be found. Some algorithms tackle this problem by starting from scratch, while others simply drop the packet or try to trace back the route back to where they came from. These recovery strategies become especially important in vehicular ad-hoc networks, since local maxima are very likely to occur. Three error-recovery strategies could be identified: *Reroute*, i.e. trace back the path until a more suitable forwarder can be found; *Wait*, i.e. do nothing until a suitable node appears in transmission range (some of the approaches drop the packet after a timeout); *Forward*, i.e. Forward to any node contained in the neighborhood table according to an algorithm, e.g. first node that appears counterclockwise in relation to the EGO's heading.

Table 2.1 summarizes the various routing schemes presented in this section and gives information about the meta-information they rely on. *Position* means the algorithm uses periodically-updated location information on cars in transmission range while *motion* refers to the data that can be inferred by looking at changed positioning information over time. Obviously, motion and heading both depend on positioning; hence all algorithms that exploit motion and heading information also use positioning. The column *map* indicates whether data from the road network influences the decision for the next forwarder in any form. Algorithms with a green checkmark in the column *intersections* try to route data over intersections in order to increase the probability of finding a suitable next hop. The column *traffic* indicates data from a traffic information service that can be queried by the algorithm in order to influence the next hop decision. Algorithms

			Meta-information					
Algorithm	**Class**	**Recovery**	**Position**	**Motion**	**Heading**	**Map**	**Inter-sections**	**Traffic**
AODV	reactive	reroute	✗	✗	✗	✗	✗	✗
A-STAR	greedy	reroute	✓	✗	✗	✓	✓	✗
CAR	reactive	wait / rr	✓	✓	✓	✗	✓	✗
DSR	reactive	reroute	✗	✗	✗	✗	✗	✗
GVGrid	reactive	reroute	✓	✓	✓	✗	✗	✗
ROMSGP	reactive	reroute	✓	✗	✗	✗	✗	✗
MOPR	greedy	forward	✓	✓	✓	✗	✗	✗
REACT	greedy	wait	✓	✓	✗	✓	✗	✗
GyTAR	greedy	forward	✓	✓	✓	✓	✓	✗
FFRDV	greedy	forward	✓	✓	✓	✓	✗	✗
GSR	reactive	no recovery	✓	✗	✗	✓	✓	✗
GPCR	greedy	forward	✓	✗	✗	✓	✓	✗
OLSR	proactive	reroute	✗	✗	✗	✗	✗	✗
ROVER	reactive	reroute	✓	✗	✗	✗	✗	✗
MURU	reactive	reroute	✓	✗	✗	✓	✗	✗
MORA	greedy	no recovery	✓	✓	✓	✗	✗	✗
GPSR	greedy	forward	✓	✗	✗	✗	✗	✗
GPSR-MA	greedy	forward	✓	✓	✓	✗	✗	✗
RBVT-P	proactive	no recovery	✓	✗	✗	✓	✓	✗
RBVT-R	reactive	no recovery	✓	✓	✗	✓	✓	✗
LGHS	greedy	wait	✓	✓	✓	✓	✓	✓
LGHS-CD	greedy	wait	✓	✓	✓	✓	✓	✓

Table 2.1.: The different routing schemes and their parameters analyzed in this section. The meta-information that the different approaches rely on are ordered from simple to complex from left to right, e.g. positioning is considered less complex than information about live traffic. Algorithms LGHS and LGHS-CD developed in this thesis (see Chapter 5) are shown in the last two rows. Legend: ✓: supported, ✗: not supported.

LGHS and LGHS-CD that are developed in this thesis are also included for comparison in the last two rows.

Obviously, all greedy approaches use positions of neighboring nodes, but even most reactive protocols rely on this type of information. Interestingly, only one algorithm makes use of traffic information: A-STAR [Seet et al., 2004] considers traffic densities along the route in order to discover more stable paths, although only statistics are applied as will be discussed below.

Addressing An important aspect for routing in ad-hoc networks is addressing the destination node. In traditional stationary networks, the (geographical) location of the sender and receiver are unimportant. The hierarchy of such networks is built upon network segments. In vehicular ad-hoc networks, however, the geographical position of the destination node is of the utmost importance. Here, two different addressing schemes are known:

- geographical multicast
- unicast

Geographical multicast can be considered as a special type of broadcast, but is limited to a geographical region. A routing scheme following this paradigm tries to route packets in the direction of the area of validity as declared in their headers. Multicast algorithms then broadcast and keep the information among the vehicles that are within the area of validity. State-of-the-art use cases like local danger warnings (cf. Chapter 3) rely on these types of algorithms.

Similar to wired networks, *unicast* algorithms realize point-to-point communication between two vehicles. This type of communication is considered more complex since global knowledge on network topology is needed in order to know the position of the destination vehicle. In reality however, this topology knowledge is significantly limited by the transmission range of the underlying wireless communication technology. In most cases the destination vehicle is either out of range, or moving out of range, during transmission. Knowledge about the location of the destination node therefore constitutes a major challenge for unicast communication in vehicular ad-hoc networks. Since the most promising routing algorithms follow a greedy approach, they do not manage routing tables. Hence, they cannot know about the position of the destination. Therefore, most routing schemes for vehicular ad-hoc networks either do not consider how to discover this location, e.g. [Seet et al., 2004], or presume a globally available "location service" like [Li et al., 2000] that can be queried in order to retrieve it, e.g. in [Jerbi et al., 2007]. Although no details on the implementation are given, this is certainly an interesting approach to approximate the location of the destination vehicle. Low-latency cellular radio like LTE or the upcoming 5G could therefore support direct communication between two vehicles that are not in transmission range.

In order to give the reader a better understanding of how the different approaches are composed, a selection of the cited routing strategies is presented in more detail.

DSR: Dynamic Source Routing The dynamic source routing protocol (DSR) [Johnson and Maltz, 1996] was one of the first attempts to propose an optimized routing protocol for vehicular ad-hoc networks. It belongs to the class of reactive protocols where route discovery processes are only triggered upon request. DSR is table driven and maintains a cache for already successfully discovered routes. When a node attempts to send a packet, it looks up the cache for a valid route and adds the hops to the header of the packet. In the case where no valid route can be found, the node broadcasts a route request packet. Every receiving node checks whether it is the destination. If not, it adds its own address to the packet header and forwards the packet. If it is the destination, it answers with a route reply. This message is sent back along the route until it reaches the sender. The nodes on the path to the destination can use a so-called promiscuous mode, which basically means that every received packet is sent to the upper layers even if the node is

not the intended addressee. DSR uses this technique in order to also store route requests in nodes other than the one that initiated the discovery process. In case a packet cannot be forwarded along a route contained in the cache, DSR notifies the sender, which then can trigger a route maintenance procedure. Promiscuous mode is also used here in order to allow other nodes to store this information in their local cache.
According to the authors, DSR makes the assumption of moderately moving nodes with respect to transmission latency and range. The route maintenance procedure makes clear why this assumption is important. Frequent topology changes of the network lead to more route discovery processes. Although an overload of the network is prevented by limiting the number of discoveries for each node, DSR is not feasible for practical use. Also note that DSR plans routes on the basis of node addresses, though it makes no use of any additional information specific to vehicular ad-hoc networks, e.g. position or moving directions of nodes.

A-STAR: Anchor-Based Street and Traffic Aware Routing Anchor-Based Street and Traffic Aware Routing (A-STAR) [Seet et al., 2004] is one of the examples which make use of meta-information in order to improve routing decisions. The algorithm is a member of the greedy position-based class and therefore chooses the next forwarder in each hop. The term *anchor-based* refers to the fact that A-STAR tries to deliver packets over intersections to the destination. Before sending a packet, the algorithm computes the intersections to be included according to an additional available map. The streets connecting these intersections are assigned a weight according to their estimated traffic density. A-STAR is especially tailored to city environments and therefore realizes this estimation with knowledge about bus lines. According to this information the weights are assigned. The authors call this a "statistically rated map". They point out that traffic density is subject to changes throughout the day. In order to adapt to these changes, they propose to use a "dynamically rated map" where real-time updates to local traffic conditions can be queried. When A-STAR reaches a local maximum during sending a packet along a precomputed route, the street connected by the intersections gets marked invalid and the sender is notified. All vehicles receiving this message also mark the street as invalid for a predefined time.
A-STAR is one of the examples where meta-information is considered to a high degree. The assumption of accessing a traffic service which provides dynamic maps, however, is unrealistic and also not evaluated in their work. Note that in Table 2.1 A-STAR is listed as an algorithm which does not make use of traffic information because they only estimate the situation according to road categories. However, their approach to using meta-information is promising. Chapter 5 builds up on these ideas when advanced algorithms for managing information flow within vehicular ad-hoc networks are elaborated.

GPSR-MA: Global Perimeter Stateless Routing with Movement Awareness Global Perimeter Stateless Routing with Movement Awareness (GPSR) [Granelli et al., 2007] is an extension of the greedy location-based algorithm GPSR [Karp and Kung, 2000]. It assumes the (geographical) location of the destination and forwards packets which are

closer to to it. Therefore, both GPSR and GPSR-MA rely on position updates from the neighboring nodes within transmission range. The motivation of GPSR-MA is the highly dynamic nature of a vehicular ad-hoc network. GPSR considers only the positions of the neighbors, not their velocity. Since the positions are only updated at intervals, the actual locations might differ. Considering possible speeds of vehicles, this could cover a significant amount of a node's transmission range. In the worst case, a node has already moved out of reach and is not available as a forwarder anymore. GPSR-MA tackles this problem by predicting the future position of surrounding vehicles according to their current speed and heading. The algorithm recovers from local maxima by applying the *right hand rule* and selecting the first car which is in reach clockwise in relation to its own heading.
While the approach of movement prediction is certainly interesting, it has to be noted that it is not based on the topology of the underlying road network. This results in possible inaccurate estimations as positions of nodes are easily predicted incorrectly.

Presentation and Session Layer

On the upper layers that are close to the actual application, it is essential to provide transparent access to the underlying communication technologies. Especially in the ad-hoc communication between vehicles, it is important to provide applications with tools that allow the assessment of the situation inside and outside of transmission range. The architectures of ETSI [European Telecommunications Standards Institute (ETSI), 2010a] and the related architecture of Kosch et al. [Kosch et al., 2012b] therefore introduce a facility layer which provides this functionality. The key facility on this layer is the Local Dynamic Map (LDM) as described in [European Telecommunications Standards Institute (ETSI), 2011a]. It provides an image of the ad-hoc network situation around the vehicle and contains information about moving and stationary objects, e.g. cars or road signs. The facility layer is responsible for receiving and storing updates from neighboring vehicles. It also generates updates of the EGO and communicates this information to the cars close-by. ETSI distinguishes in [European Telecommunications Standards Institute (ETSI), 2011a] four different categories of information to be stored in the LDM:

- **Permanent static data**: information on real-world objects that are not subject to change. This includes road topography and attributes as well as points of interest.
- **Transient static data**: contains information about real world objects that are "quasi static" such as traffic signs and positions of gantries.
- **Transient dynamic data**: includes information about real-world objects, that are subject to more frequent changes. Among others, ETSI mentions positions of roadwork or speed limits.
- **Highly dynamic data**: moving vehicles and updating dynamic traffic signs are considered highly dynamic and are contained in this category.

The fourth category of highly dynamic data in particular requires the frequent exchange of messages between vehicles in order to update the LDM. The standardization of these messages is an essential part of the functionality of a C2X system as communication must be vendor-independent. The standardization process, however, is still ongoing and is likely to change in the years to come. Currently, ETSI describes two elementary message types for vehicular ad-hoc communication: the *Common Awareness Message* (CAM) [European Telecommunications Standards Institute (ETSI), 2011b] and the *Decentralized Environmental Notification Message* (DENM) [European Telecommunications Standards Institute (ETSI), 2010b].

Common Awareness Message (CAM) This type of message provides "information of presence, positions, as well as basic status of communicating ITS stations to neighboring ITS stations that are located within a single hop distance" [European Telecommunications Standards Institute (ETSI), 2011b]. CAMs are generated automatically within the facility layer without intervention by an application and use a broadcast mechanism over the ad-hoc network. Note that it is defined as a *single-hop* broadcast. The LDM therefore contains information about every vehicle within transmission range. A general broadcast not limited to one hop would jam communication channels. Generation frequency of CAMs varies, depending on the application, between 1 and 10 Hz.
Applications are responsible for processing this data and taking action according to their assessment of the situation. Detailed information about the contents of a CAM is shown in Figure 2.16. The definition of ETSI distinguishes between a common header for CAMs and DENMs and CAM specific fields. The header contains versioning information as well as a message ID. Furthermore, the time of message generation is stored. A plausibility check is applied in order to determine whether a CAM is still valid. If true, the information is stored in the LDM, or it is discarded otherwise. Because the same header is used for CAMs and DENMs, the message ID is used to distinguish between these two types (0 = CAM, 1 = DENM).
The CAM-specific fields contain a unique ID which identifies the station and furthermore the geographic position of the station at the point in time when the message was generated. ETSI names four profiles which affect the content of the three remaining fields:

- **Profile basicVehicle**: a regular vehicle.
- **Profile basicIRS**: a stationary object, e.g. traffic sign or roadside station.
- **Profile emergencyVehicle**: fire truck, ambulance or police.
- **Profile publicTransportVehicle**: bus, train, etc.

Depending on the profile, the station characteristics contain three flags which indicate whether this station is able to change its position, if it is a member of a public authority and if it is physically relevant, e.g. other vehicles can collide with this object. Common vehicle parameters include speed, heading and acceleration. A number

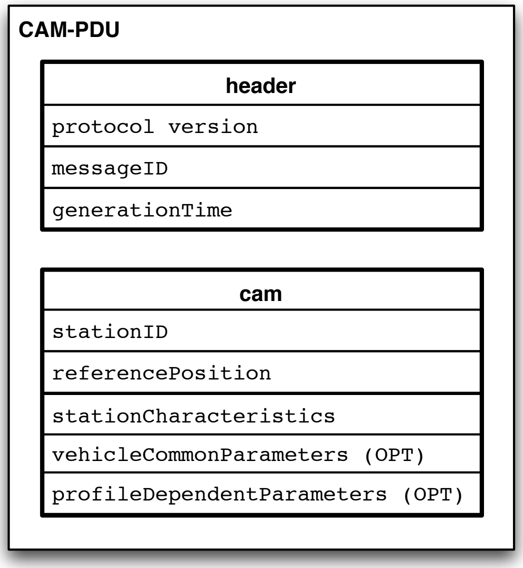


Figure 2.16.: Common Awareness Message specification according to ETSI [European Telecommunications Standards Institute (ETSI), 2011b]. Optional field are marked by "OPT".

of other mandatory and optional parameters are specified, like confidence values. For a full list of available parameters see, [European Telecommunications Standards Institute (ETSI), 2010b]. Profile-dependent parameters are stored in the last field, e.g. traffic light priorities for emergency vehicles.

Decentralized Environmental Notification Message (DENM) In contrast to CAMs, the second class of messages used for vehicular ad-hoc communication are event-triggered and not generated in regular intervals. They also use a geographical broadcast mechanism as described in Section 2.2.2. According to ETSI, "Decentralized Environmental Notification Messages (DENMs) are mainly used by the Cooperative Road Hazard Warning (RHW) in order to alert road users of detected events" [European Telecommunications Standards Institute (ETSI), 2010b]. Immediately after a critical situation is detected, the station broadcasts a DENM to the close vicinity. As long as the situation persists, the message is repeated at a given frequency. However, no minimum and maximum frequency is mentioned in [European Telecommunications Standards Institute (ETSI), 2010b]. The broadcasting is stopped either when event that caused it disappears, lifetime of the DENM expires or the generating station explicitly sends out a cancellation message. The disappearance of an event is to be seen from the EGO's perspective: imagine the car is detecting a slippery road. It then broadcasts DENMs in order to warn other road users. As soon as traction of the wheels comes back to normal, the application would stop sending out messages. As for CAMs, situation assessment is done by the applications, e.g. a road hazard is only valid for cars driving towards this event.

However, other vehicles may also forward the message, even if the corresponding event has been assessed as not relevant for this car.
Figure 2.17 shows the composition of a DENM according to ETSI [European Telecommunications Standards Institute (ETSI), 2010b]. The header is common for DENMs and CAMs. The DENM-specific fields are divided into three containers:

- **Management container**
  - *ActionID*: describes an unique identifier for an event. It is generated from the stationID and sequence number. This allows disambiguation of the same event types generated by two different vehicles. This information is used to aggregate two DENMs into one event that is stored in the LDM, e.g. the receiving station can detect if two DENMs are originating from the same vehicle and, if true, combines them into one event.
  - *DataVersion*: Using the field dataVersion a receiving station can determine if this message describes an update from the sending station. It is also used to mark a DENM as canceled using the value 255.
  - *ExpiryTime*: Another way to mark a DENM as invalid is to set the field expiryTime. When reached, corresponding DENMs are not forwarded anymore and are removed from the LDM.
  - *Frequency*: Receiving stations forward messages according to the value declared in the field frequency.
  - *Reliability*: Detection of events might not be reliable in some cases, e.g. weather warnings. A probability can be added using this field and can be used by applications to determine whether to trigger a notification to the driver.
  - *IsNegation*: This field can also be used to cancel the event described by a DENM.
- **Situation container**
  - *Traffic flow effect*: If available, the sending station can communicate the impact on traffic flow of the event connected to this DENM, e.g. if it is causing a traffic jam
  - *Situation*: The situation field describes the direct cause of this event.
  - *LinkedCause*: communicates a possible linked cause to the main event, e.g. a traffic jam as a consequence of a crash on the motorway.
  - *Severity*: This field classifies the event into one of four severity levels, from low to safety critical.
  - *EventCharacteristics*: includes event-specific characteristics, e.g. if the event is moving or time critical

- *VehicleCommonParameters*: As for CAMs, this includes speed, heading, and position. A full list of available parameters can be found in [European Telecommunications Standards Institute (ETSI), 2010b].
- *ProfileDependentParameters*: As for CAMs, this includes parameters specific to the selected profile. A full list of available parameters can be found in [European Telecommunications Standards Institute (ETSI), 2010b].

- **Location container**
  - *EventPosition*: the position of the event connected to the DENM. It can either be a geographic location or an area in which the event is valid. It can be used by applications for performing relevance checks but also to link several DENMs from different cars to one event.
  - *LocationReference*: This field tracks the position of the event over time in case it is changing position, e.g. non-stationary roadwork.

Relevance for this Thesis The description of message formats that are undergoing a standardization process at the moment for vehicular ad-hoc communication shows that they are tailored specifically to safety-related applications. While Chapter 3 analyzes common applications of the Push Paradigm in more detail, it can be concluded at this point that the established communication infrastructure on the upper OSI layers of the ad-hoc network stack is not intended to be applied to added-value services. CAMs are only used for a basic service of presence notification. The messages are sent only to the vehicles in communication range using a single-hop broadcasting mechanism. Although DENMs use multi-hop, their composition is not generic and lacks the ability to transport arbitrary payloads over the ad-hoc network.

2.2.3. Simulating Car2X Networks

The simulation of vehicular ad-hoc networks is an important element in the toolchain for the successful deployment of applications within this domain. For realizing field operational tests (FOTs) like sim^{TD}or DRIVE C2X described in Section 2.1.3, financial, bureaucratic and organizational hurdles have to be passed long before the actual test can be started. In sim^{TD} for example, more than one year of integration effort was necessary in order to integrate and prepare the system for the FOT. Over 30 developers met on a regular basis at the testing site in Friedberg/Frankfurt for so-called "integration workshops". Also, planning the actual FOT, organizing and paying test drivers and resolving legal issues have to be considered. Another important aspect of simulations is the ability to predict impacts of C2X enabled cars on traffic. This stresses the necessity for simulation of C2X networks. Therefore, this topic drew the attention of researchers long before the first FOTs were planned. In the beginning, the focus lay on the implementation of comprehensive and realistic simulation tools for wireless networks. With the progress of the technology, C2X-specific models for 802.11p were added.

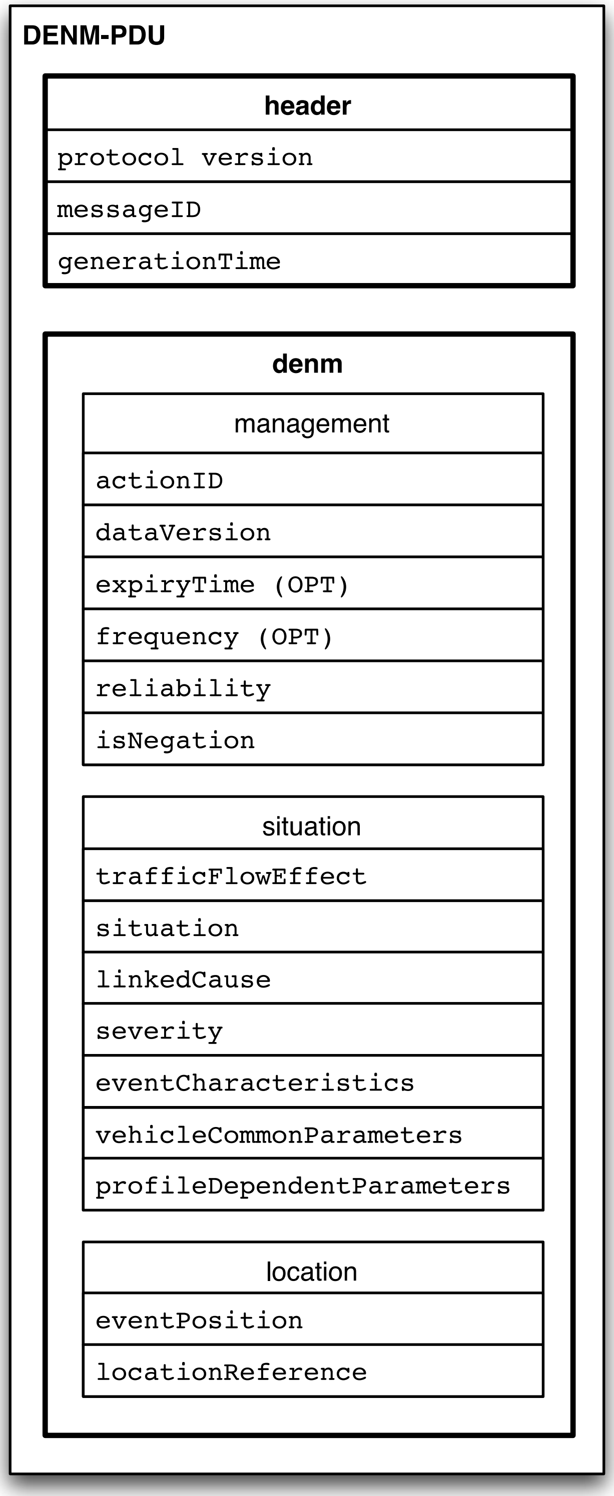


Figure 2.17.: Decentralized Environmental Notification Message specification according to ETSI [European Telecommunications Standards Institute (ETSI), 2010b]. Optional fields are marked by "OPT".

However, the ability to simulate a complete C2X system goes beyond the scope of wireless network simulators. While these only consider the realistic simulation of the network stack, they do not consider realistic vehicle movements on real maps or driver behavior. While some approaches include certain mobility models like random walk [Mangharam et al., 2005] or random waypoint [Johnson and Maltz, 1996], this is far from being realistic. In general, cars follow the road network according to traffic rules (right of way, traffic lights, etc.). Furthermore, movements of one car affect others, as collisions have to be avoided. In order to simulate more realistic scenarios, network and traffic simulators have to be coupled.
Within the scope of this thesis questions on human-factor issues are also addressed. Therefore, a driving simulator needs to be added. This leads to the following requirements for the coupling system:

1. **Extensibility** (mandatory): The coupling system must meet the mandatory requirement of being able to add the driving simulator.
2. **Development support** (mandatory): In this thesis, several novel C2X applications are developed. The coupling system must support the development of C2X based applications through a well-defined API.
3. **Availability** (mandatory): The coupling system must be freely available and supported by the maintainers. Note, that access to the source code is not necessarily required.
4. **Programming language** (optional): the system in this thesis will be developed in Java. Ideally, the coupling system is written in Java.

In this section the most common wireless network and traffic simulators that are widely used in the community are described. Furthermore, recent approaches for a synchronized execution of both are introduced. After that, an analysis with respect to their potential for contributing to the goals of this thesis is given. Already at this point, it can be stated that no coupling to a driving simulator exists so far. Therefore, a mandatory property of an available coupling system is the possibility to add another simulator.

Wireless network simulators

Table 2.2 presents an analysis of currently available state-of-the-art network simulators. The analysis focuses on their ability to handle the wireless communication technologies that were identified in Section 2.2.1 as suitable for application to the Pull Paradigm.

The Network Simulator ns-2 is among the most popular tools for simulation of wired and wireless networks [27]. Its development started by the end of the 80s and continued during the 90s. Ns-2 was focused on the simulation of wired networks. Because of its

[27] http://nsnam.isi.edu/nsnam/index.php/Main_Page, last accessed 24.11.2013

	SRC	WLAN			WiMAX	Mobile Radio		General Info	
	Blue-tooth	.11a	.11b/g	.11p	802.16	UMTS	LTE	Language	License
ns-2	✔*	✔	✔	✔*	✔*	✔*	✔*	C++/Otcl	GPL
ns-3	✔*	✔	✔	✔	✔*	✗	✔	C++/python	GPL
OPNET	✔	✔	✔	✗	✔	✔	✔	C++	com
OMNet++	✗	✔	✔	✔	✔*	✔*	✗	C++	APL
GloMoSim	✗	✔	✔	✔*	✗	✗	✗	PARSEC	own
QualNet	✔	✔	✔	✔*	✔*	✗	✗	C++	com
NCTUns	✗	✗	✔	✔	✔*	✗	✗	C++	own
GTnet	✔	✔	✔	✔	✗	✗	✗	C++	GPL
JiST/SWANS	✔	✔	✔	✔	✔*	✔*	✔*	Java	own

Table 2.2.: Network simulators for wireless communication technologies suitable for the Pull Paradigm according to the analysis in Section 2.2.1. Green checkmarks (✓): native support for a communication technology by the respective simulator; checkmark with star (✓*): support available using extensions; gray cells (✗): no support.

acceptance, community-contributed modules were written in order to also support Bluetooth, 802.11a/p as well as UMTS and LTE. It is written in C++ and comes with its own scripting language, Object Tcl. This supports the development of models without the necessity to compile ns-2 code before running the simulation. Ns-2 is distributed under the General Public License (GPL).
Ns-2 is still widely used today. While it is not being actively developed anymore, the software is still being maintained.
In 2006, it was succeeded by ns-3[28]. Unlike in ns-2, the development of ns-3 was focused on the ability to simulate wireless networks based on 802.11 out of the box. However, ns-3 does not support UMTS but it provides native support of the follow-up standard LTE. Furthermore, OTcl was replaced by the popular Python scripting language. Ns-3 is also available at no cost and is licensed under the GPL.

OPNET Modeler is a commercial network simulator that supports a wide range of protocols[29]. For wireless, almost all communication technologies are supported out of the box. However, simulation of 802.11p is not supported.

OMNeT++ is a discrete event simulation platform written in C++ [Varga, 2001], [Mallanda et al., 2005], [Varga and Hornig, 2008]. According to the authors, a key difference between ns-2 and OMNeT++ is its clear separation from simulation kernels and models [Varga, 2010]. While ns-2 intends to build a network simulator, OMNeT++ pro-

[28] http://www.nsnam.org/, last accessed 24.11.2013
[29] http://www.opnet.com/solutions/network_rd/modeler.html, last accessed 24.11.2013

vides a simulation platform which can be enhanced to a simulation framework by the community. Another important aspect of OMNeT++ is its ability to build *hierarchical* models and therefore provide the ability to reuse as much code as possible. The topology description language called NED is used to define models and their relationships. OMNeT++ provides an integrated development environment using NED as its native file format. The possibility of creating network topologies using a graphical editor supports the rapid development process for setting up simulations. A number of contributions have been made in order to integrate models into OMNET++. Support for wireless 802.11 is available out of the box, and UMTS comes as a plugin. However, no support is available for bluetooth and LTE.
Omnet++ is available under the Academic Public License and is therefore free for research use.

GloMoSim is a library-based simulator targeted explicitly at wireless networks and freely available for research purposes [Bajaj et al., 1999]. Each library simulates a certain protocol within the network stack and is written in PARSEC, a C-based parallel simulation language [Zeng et al., 1998]. One of the key features of GloMoSim is the parallel execution environment which results in high performance. Because it was built to primarily target wireless networks, it supports 802.11a/b/g out of the box. Support for .11p was added later using a separate library. Bluetooth and mobile radio are not included. The codebase of GloMoSim was commercialized in the Qualnet simulator[30] for which Bluetooth support is available. According to [Varga and Hornig, 2008], Qualnet's customers are primarily from military fields.

NCTUns is a simulator which explicitly added support for 802.11p in its 5.0 release [Wang and Lin, 2008], [Wang and Huang, 2012]. A differentiator from other solutions presented so far is the usage of the real Linux network stack. According to the authors, applications written for NCTUns can also run on a real network device without further modifications and it opens the possibility to generate more realistic network traffic. A graphical user interface enables the user-friendly setup of simulation runs. NCTUns is mainly mentioned here because it provides traffic simulation capabilities which are integrated into the simulator and provides native 802.11p support. Simulations can be run on real maps by using the SHAPE file format[31]. However, it lacks support for other important wireless communication technologies like Bluetooth or mobile radio.
NCTUns is written in C++ and available at no cost for academic use.

GTNetS is a network simulator developed at Georgia Tech [Riley, 2003a], [Riley, 2003b]. The author claims the usage of C in combination with OTcl leads to substantial memory consumption. In order to overcome these limitations and build up a scalable simulation

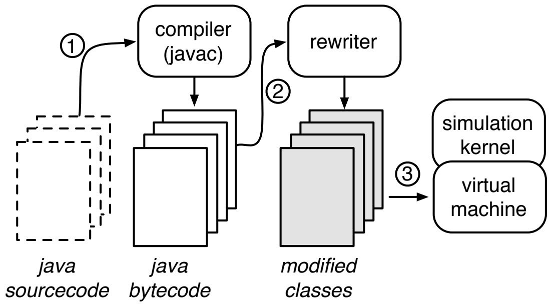
[30] http://www.qualnet.com, last accessed 24.11.2013
[31] http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf, last accessed 24.11.2013

environment, GTNetS was developed. Similar to the approach of OMNeT++ [Varga and Hornig, 2008] they use an object-oriented approach in order to enhance functionality of the simulator. Besides wired networks, 802.11b/g/p are supported. A later added model also supports bluetooth [Zhang and Riley, 2004].
GNetS is written in C++ and freely available under GPL.

Java in Simulation Time (JiST) is another notable approach to discrete event simulation which follows a different paradigm than the presented simulators above. JiST converts the Java Virtual Machine (JVM) into a simulation environment by introducing transparent execution semantics for programs written in Java [Barr et al., 2005a]. JiST programs do not require simulation-specific languages like OTcl for ns-2 or PARSEC for GloMoSim / Qualnet. In other words, JiST adds a simulation clock such that the JVM is then not executed in realtime but in simulation time. Advancing in time is then controlled by JiST. In order to achieve this, the JiST architecture distinguishes four components: compiler, virtual machine, rewriter and simulation kernel, as shown in Figure 2.18. The compiler and virtual machine are standard components available for the Java language. The execution in simulation time is handled by the rewriter and the simulation kernel. According to [Barr et al., 2005a] the rewriter basically is a dynamic class loader. It rewrites the program on the bytecode level by intercepting all class load requests and embedding the simulation time operations. This is done after compilation but before execution on the virtual machine. Hence, the rewriting is done only once. During execution, the modified classes are run in simulation time by the kernel, which is possible due to the injected time operations added during the rewriting step.
On top of JiST, Barr et al. built the Scalable Wireless Ad-hoc Network Simulation (SWANS) [Barr et al., 2005b]. SWANS is highly efficient and scales very well for large networks with a high number of nodes. This efficiency is not only due to the underlying JiST but also because of a technique the authors call "hierarchical binning". When a simulated signal is sent over the communication channel, the runtime has to consider all possible reception nodes after considering pathloss and fading. In a nutshell, hierarchical binning allows the computation of the set of receiving nodes proportional to the set size [Barr et al., 2005b].
SWANS supports all 802.11 derivates addressed in this thesis, and Bluetooth. Mobile radio can be added using plugins.

The approach of using virtual machines for simulation is efficient. [Barr et al., 2005b] evaluate the performance of JiST/SWANS against ns-2 and GloMoSim in terms of memory footprint and required time to run a 2-minute NDP simulation. They varied the number of nodes between 500 and 50,000. It turned out that for the simulation task, JiST/SWANS outperforms both competing simulators in terms of allocated memory and required simulation time.

It can be concluded at this point that JiST/SWANS is a very promising wireless



Nodes	Simulator	Time (s)	Memory (KB)
500	**SWANS**	**54**	**700**
	GloMoSim	82	5759
	ns2	7136	58761
	SWANS-hier	*43*	*1101*
5000	**SWANS**	**3250**	**4887**
	GloMoSim	6191	27570
	SWANS-hier	*430*	*5284*
50000	**SWANS**	**312019**	**47717**
	SWANS-hier	*4377*	*49262*

Figure 2.18.: **Left:** Java in Simulation Time (JiST) architecture. **Right:** Evaluation results of JiST/SWANS against GloMoSim and ns-2. Pictures adapted from [Barr et al., 2005b].

network simulator for this thesis, as it is written in pure Java and scales well for large networks. Necessary wireless communication technologies either come with the simulator or can be added using plugins.

Traffic simulators

As discussed above, the traffic simulator application is essential for realistic simulation of C2X networks. To be precise, *microscopic* simulation is necessary. In contrast to macroscopic traffic simulation, every car is modeled individually. This is essential since simulation of wireless communication is also based on individual nodes. Typical distinction criteria of such simulators are mobility models used and modeling of road infrastructure, e.g. traffic lights. Particularly interesting for this thesis is the ability to couple other simulators (e.g. external APIs are available) and to import existing, publicly available maps (Open Street Map). The description of these systems is kept at a higher level and mainly focuses on the distinction criterions introduced above. The primary goal is to identify a suitable simulator that can be coupled with one of the network simulators described in the previous section, which is analyzed in the next section. Next, available traffic simulators with respect to the goals of this thesis are described. A quick overview is available in Table 2.3.

VISSIM is a commercial, multimodal traffic simulation platform [Fellendorf and Vortisch, 2010]. It is designed to model a number of different types of traffic simultaneously, e.g. buses, pedestrians or cyclists. The road infrastructure can be modeled in arbitrary detail and various importers are available. According to the product description this includes the import of OSM data. Traffic lights and signs can also be added and are obeyed by the simulated traffic. To ensure more realistic simulation, traffic signal control provides interfaces to common standards used in the real world. Because it is a commercial product, no detailed information about mobility models could be found. VISSIM can be

	Features			General Information		
	OSM import	Infra-structure	Coupling	License	Language	Mobility model(s)
VISSIM	✔	✔	✔	com	C++	proprietary
CARISMA	✗	✗	✗	own	C++	RW / RD
VanetMobiSim	✗	✔	✔	GPL	Java	IDM-IM/-LC
SUMO	✔	✔	✔	GPL	C++	Krauß

Table 2.3.: Traffic simulators described in this section. Legend: ✓: feature is present, ✗: feature is not present.

enhanced by several modules that are sold by the vendor as separate products. One of these products explicitly provides Car2X functionalities and is described as a "programming interface". According to the vendor, applications based on Car2X technology can be added using Python or C++. The communication itself is modeled using a "stochastic model". No further information is given about simulation efficiency or scalability, despite the usual product descriptions. In particular, no evaluation against other tools is available.
Private transport vehicles can be modeled using individual attributes like length, width, acceleration and deceleration rates. They are placed randomly on the map and find their way through the network autonomously, e.g. no pre-defined routes are applied. For public transport, simulated arrival and departure times can be modeled.

CARISMA was developed by [Kosch, 2005] within the scope of a PhD thesis. In his thesis he elaborated on "situation adaptive communication in ad-hoc networks". The ability to simulate vehicle movements was part of the system. Before running a scenario, several parameters can be adjusted: scenario size, number of vehicles, acceleration options, etc. Vehicles' paths through the road networks are either determined at random, e.g. the car chooses a new direction at each intersection (random drunken) or by random walk. It is also important to note that no cars leave the simulation; they choose another destination on the map after they have reached their destination. Also as in VISSIM, vehicle characteristics like length and acceleration / deceleration are considered. Furthermore, CARISMA also models human driver behavior by varying the distance between vehicles according to the drivers' state. The author argues that human perception of distance to the car in front might be different from the actual distance. In contrast to VISSIM, maps can be loaded from SHAPE files which can be converted to OSM, but the overall simulation capabilities are limited by several factors: Streets are limited to two lanes and passing maneuvers are not considered at all. Aside from that, a simple conflict resolution at intersections is used and no traffic lights are considered. CARISMA is mentioned here because it is already a system which provides C2X and traffic simulation in one system without added coupling overhead. Because it is a self-contained system it does not support coupling. Furthermore, it not actively supported anymore and possibilities are limited due to the described factors.

VanetMobiSim is a recent approach for generating movement patterns for both macroscopic and microscopic traffic simulation [Haerri et al., 2011]. It inherits results from CanuMobiSim [Stepanov et al., 2005] and focuses on the development of highly realistic models. The authors integrated their previously-developed Intelligent Driver Model (IDM) [Treiber et al., 2000] into VanetMobiSim and augmented it with intersection management (IDM-IM) and lane changing behavior (IDM-LC). Basically, IDM is car-following model describing dynamics of a vehicle continuosly over time by a differential equation.
According to [Haerri et al., 2011], VanetMobiSim is able to generate vehicle movements for ns-2, GloMoSim and Qualnet.
VanetMobiSim is able to import maps from various formats, like Topologically Integrated Geographic Encoding and Referencing system (TIGER), Geographic Data File (GDF) or clustered Voronoi. TIGER maps are freely available and were completely merged with the Open Street Map project in 2007 and 2008[32]. However, TIGER is only available for the United States. Other countries are not considered. GDF is an interchange format used in several navigation systems. No direct support for the conversion to OSM is available. However, this feature can be indirectly added using eWorld[33] but since no native support is given, OSM import is marked as not available in Table 2.3.

SUMO is a popular and widely used microscopic traffic simulator developed by the German Aerospace Center (DLR) [Behrisch and Bieker, 2011; Krajzewicz et al., 2002]. It is available as open source and receives continuous support and enhancements by the initial developers as well as from the community. SUMO comes with a command-line tool called *netconvert* which converts different available map formats to its internal representation of road networks. Out of the tools presented here, import is available for VISSIM and OSM. Furthermore, *netgen* generates user-defined road topologies that are understood by SUMO.
A very important feature for coupling with other simulators is available in TraCI, the Traffic Control Interface. Using this socket connection, SUMO can be remotely controlled by other applications before, during and after a simulation run.
Vehicle routes through the network can be specified by departure and arrival properties, e.g. lane, velocity, entry point and exit point. In contrast to CARISMA, vehicles are removed from the map when they have reached their pre-defined destination. A number of additional properties are available. In order to simulate large-scale scenarios, origin/destination matrices can be used to generate individual routes for a high number of vehicles. SUMO uses a modified version of the Krauß car-following model [Krauß, 1998] in order to determine acceleration and deceleration of vehicles. It is also possible to

[32] http://wiki.openstreetmap.org/wiki/TIGER
[33] http://eworld.sourceforge.net/

simulate passing maneuvers. This feature was added during initial development. Details can be found in [Krajzewicz, 2010].
Like VISSIM, SUMO is able to simulate intermodal traffic. Especially interesting is the ability to model emergency vehicles by assigning priorities to certain cars at intersections. Road infrastructure like traffic lights is also supported and fully controllable via the TraCI interface. However, the authors point out that the network model is coarse in contrast to VISSIM and therefore not used by traffic engineers for evaluating real-life intersections [Behrisch and Bieker, 2011]. Also interesting, though less relevant for this thesis, is the ability to model person-based intermodal traffic. At an individual level, start, destination and allowed transport systems are given. SUMO then simulates the person's way through the network fulfilling the given transport constraints.
SUMO follows a very generic approach to traffic simulation, and thus is a promising option for this thesis.

Driving Simulators

A high-fidelity and accurate driving simulator is a crucial element for addressing the research questions of this thesis. Although a number of driving simulators exist, e.g. SILAB[34], VTI Foerst[35], CKAS Full Driving Simulator[36] they are under a commercial license and involve high costs and do not allow extensions. An open-source driving simulator that is targeted for research purposes is CARS [Kern et al., 2008]. It is written in Java and consists of three components: a visual map editor for creating the simulation environment, the driving simulator itself and a tool to analyze the driver's performance. Although availability and the programming language used meet the requirements for this thesis, CARS lacks important features like importing existing maps, buildings, traffic lights, etc. Also, development stopped in 2009.

A recent approach to provide a high-fidelity driving simulator to the research community is OpenDS [Math et al., 2012]. It is also written in Java and based on the JMonkeyEngine[37]. The engine supports Blender, Ogre and Wavefront formats and is able to load nearly any 3D environment. This has been demonstrated by importing a highly detailed model of the city of Saarbrücken. Data was provided by the land office. The simulator is currently under active development by DFKI in the context of the EU-funded project GetHomeSafe[38]. As it is open source and therefore easy to adapt, it contributes to the PADE system that is described in Chapter 7.

[Gajananan et al., 2013] present a framework that enables multiple users to simultaneously control cars in the same virtual environment. This opens the possibility to conduct distributed user studies over several countries. The framework introduces the scenario

[34] http://www.wivw.de/ProdukteDienstleistungen/SILAB/
[35] http://www.vti.se/en/research-areas/vehicle-technology/vtis-driving-simulators/vti-first-simulator/
[36] http://www.ckas.com.au/driver_training_simulators_43.html
[37] http://www.jmonkeyengine.org/
[38] http://www.gethomesafe-fp7.eu

		Traffic Simulators				
		SUMO	VISSIM	CARISMA	ITS Modeler	VanetMobiSim
Network Simulators	ns-2	TraNS	MSIECV	CARISMA	-	VanetMobiSim
	ns-3	VSimRTI / simITS / iTetris	VSimRTI	-	-	-
	OPNET	-	-	-	MOBYSIM	-
	OMNet++	VSimRTI	VSimRTI			
	GloMoSim	-	-	-	-	VanetMobiSim
	QualNet	-	-	-	-	VanetMobiSim
	NCTUns	-	-	-	-	-
	GTnetS	-	-	-	-	-
	JiST/SWANS	VSimRTI	VSimRTI	-	-	-

Table 2.4.: The simulators presented in this section. Cell(x,y): coupling system for network simulator x and traffic simulator y.

markup language (SML) which can be used to specify traffic events, e.g. accidents and to ensure reproducible situations. This is especially useful for controlling situations in distributed studies in order to compare the same situation over several participants. SML is a promising enhancement to OpenDS but multi-user studies are part of future work.

Conclusion

This section presented widely used network and traffic simulators. As outlined in the introduction a synchronization of both is necessary in order to model realistic behavior of vehicular ad-hoc networks. This section concludes with respect to the described network and traffic simulators and narrows down the possibilities to be considered by looking at already-existing coupling systems. In Table 2.4 the presented network simulators are listed in rows, traffic simulators in columns. Each cell contains name(s) of already existing coupling systems incorporating the two respective simulators. The following systems can be identified:

- Vehicle Simulation Runtime Infrastructure (VSimRTI) [Schünemann, 2011]
- Multiple Simulator Interlinking Environment for C2CC in VANETS (MSIECV) [Lochert et al., 2005a]
- VAnet Simulation Environment (VASE) [Hrizi and Filali, 2010]
- Traffic and Network Simulation Environment (TraNS) [Piorkowski and Raya, 2008]
- Simulation for Intelligent Transportation Systems (simITS) [Hrizi and Filali, 2010]

		Requirements			
		Extensibility	Development support	Availability	Programming Language
Coupling Systems	TraNS	✗	✔	✔	✔
	MSIECV	✗	✔	✗	✗
	simITS	✗	✗	✗	✗
	CARISMA	✗	✗	✗	✗
	MOBYSIM	✔	✔	✗	✗
	VanetMobiSim	✗	✗	✗	✔
	iTetris	✔	✔	✔	✗
	VASE	✗	✗	✗	✗
	VSimRTI	✔	✔	✔	✔

Table 2.5.: Analysis of the available simulator coupling systems with respect to the requirements introduced at the beginning of this section.

- Integrated Wireless and Traffic Platform for Real-Time Road Traffic Management Solutions (iTETRIS) [Krajzewicz et al., 2010]

Note that traffic simulation is included in CARISMA and VanetMobiSim and therefore are not listed as separate coupling systems.

As a first step, the network simulators NCTUns and GTnetS are excluded since no coupling systems are available for them. Furthermore, NCTUns supports neither Bluetooth nor mobile radio simulation. Next, the coupling systems are analyzed with respect to the requirements introduced at the beginning of this section in Table 2.5. It turns out that only two systems fulfill all mandatory requirements: VSimRTI and iTetris. However, VSimRTI is more generic. It is able to couple various simulators and is the only system that supports JiST/SWANS, which turned out to be the most suitable of the presented network simulators. Therefore, and because of the additional optional fourth requirement met, VSimRTI seems to be the most promising approach for this thesis.

3. Established ADAS Based on Unidirectional Car2X Communication

Numerous driver assistance systems can already be found in production cars. Antilock Braking System (ABS) and Electronic Stability Control (ESP) are common today even in subcompact cars. Higher classes integrate advanced driver assistance systems like Adaptive Cruise Control (ACC) or parking assistance. However, they all rely on sensors integrated into the vehicle and therefore no collaboration between OEMs is necessary for proper functioning. Section 2.1.5 showed that the situation for cooperative vehicles is different. Standards play an important role in successfully deploying C2X technology to the market. Implementations by car manufactures need to comply with those standards in order to be able to communicate vendor-independently. Not only do protocols need to be standardized, but also applications and their use cases. This chapter introduces unidirectional applications of the Car2X domain that are standardized by ETSI and therefore reflect the common view of stakeholders. These applications are then matched to implementations in the two largest field operational tests for Car2X Communication, sim^{TD} and DRIVE C2X since these have the highest chance for being integrated into cars after market introduction. Results of this chapter are twofold: first, not all applications specified by ETSI can be found in the FOTs. Second, a strong focus on information-centric, safety and traffic efficiency ADAS based on vehicular ad-hoc communication is found.

3.1. The ETSI Basic Set of Applications

The Intelligent Transport Systems Steering Group (ITSSG) of ETSI (see Section 2.1.5) advances interoperability of services in the ITS domain. In their Technical Report 102 638 [European Telecommunications Standards Institute (ETSI), 2009], ETSI specified a basic set of applications (BSA) and use cases within the Push Paradigm. Although the document focuses on applications working on the basis of C2C, C2I and I2C communication, it also contains use cases based on mobile networks (UMTS, LTE) and broadcast systems (DVB, DAB). ETSI distinguishes several application classes: active road safety, co-operate traffic efficiency, cooperative local services, and global internet services (see Table 3.1). Analyzing the use cases in Table 3.1 reveals a strong focus on road safety and traffic efficiency. Furthermore, interaction with most use cases is not intended: the driver is considered as an information receiver hence they are considered as information-centric.

Most research projects in the C2X domain subsume the application classes "co-operative local services" and "global internet services" into "Infotainment and Others". Thus, classification of use cases is continued as follows:

Application Class	Application	Use-Case
Active road safety	Driving assistance - Co-operative awareness	Emergency vehicle warning
		Slow vehicle indication
		Intersection collision warning
		Motorcycle approaching indication
	Driving assistance - Road Hazard Warning	Emergency electronic brake light
		Wrong way driving warning
		Stationary vehicle - accident
		Stationary vehicle - vehicle problem
		Traffic condition warning
		Signal violation warning
		Roadwork warning
		Collision risk warning
		Decentralized floating car data - Hazardous location
		Decentralized floating car data - Precipications
		Decentralized floating car data - Road adhesion
		Decentralized floating car data - Visibility
		Decentralized floating car data - Wind
Cooperative traffic efficiency	Speed management	Regulatory / contextual speed limits notification
		Traffic light optimal speed advisory
	Co-operative navigation	Traffic information and recommended itinerary
		Enhanced route guidance and navigation
		Limited access warning and detour notification
		In-vehicle signage
Cooperative local services	Location based services	Point of Interest notification
		Automatic access control and parking management
		ITS local electronic commerce
		Media downloading
Global internet services	Communities services	Insurance and financial services
		Fleet management
		Loading zone management
	ITS station life cycle management	Vehicle software / data provisioning and update
		Vehicle and RSU data calibration

Table 3.1.: The ETSI Basic Set of Applications (BSA) for the Push Paradigm.

- **Safety**
- **Traffic Efficiency**
- **Infotainment and Others**

It should be noted that various categorizations exist. For example, [Kosch et al., 2012b] employ another taxonomy. They distinguish "AutoNet Applications" according to their relation to different entities. These entities are either driving, vehicles, or passengers. The first category includes all applications that help improve driving and traffic on the road. This means that local danger warnings are included here, as well as traffic efficiency applications. The second category includes applications that serve the drivers' comfort and entertainment. Fleet management, remote software updates, etc. fall into the third category, vehicle related applications. It is obvious that although this categorization approach is different, it is just another perspective on a comparable range of applications. The BSA of ETSI reflects a common view of stakeholders on possible applications within the Push Paradigm, but a use case contained in this set does not imply a realization.

ETSI application	DRIVE C2X Function	sim^{TD} Function
Roadwork warning	Roadwork warning	Roadwork warning
Stationary vehicle: vehicle problem	Car breakdown warning	Obstacle warning
Emergency vehicle warning	Approaching emergency vehicle	Emergency vehicle warning
Decentralized floating car data: Visibility / Wind	Weather warning	Road weather warning
Emergency electronic brake light	Emergency electronic brakelight	Electronic brakelight
Stationary vehicle: accident	Post crash warning	Obstacle warning
-	Traffic jam ahead warning	Traffic jam ahead warning
-	-	Cross-traffic assistance
Regulatory / contextual speed limit notification / In-vehicle signage	In-vehicle signage/Speed limit	Traffic sign assistant
Traffic light optimal speed advisory	Green light optimal speed advisory	Traffic light assistant
-	-	Street Foresight
-	-	Roadworks information system
-	-	Enhanced navigation
-	-	Diversion management
-	-	Internet-based services

Table 3.2.: The ETSI Basic Set of Applications (BSA) matched to realized and evaluated applications in the two largest field operational tests for C2X Communication DRIVE C2X and sim^{TD}. Light orange: safety, light green: traffic efficiency, light blue: infotainment and others.

Research projects realized only a subset of the BSA by introducing well-defined selection processes, e.g. technical feasibility or deployment and time to market. In order to get a deeper understanding about applications and use cases within the Push Paradigm, this chapter considers those implemented and evaluated in two of the largest field operational tests for C2X Communication: the project DRIVE C2X on a European level, and sim^{TD} which took place in Germany (cf. Section 2.1.3). Since major European and German car manufactures were part of the consortium, the selected set of applications have the highest probability for a market introduction as soon as C2X technology is available in production models. This chapter continues by matching use cases of the ETSI BSA set to applications in DRIVE C2X and sim^{TD} in order to reveal functions being focused on in state-of-the-art research.

Table 3.2 shows ETSI BSA applications in the first column. DRIVE C2X functions which were selected for full impact assessment are listed in the second column. Despite different naming, use cases of DRIVE have a corresponding counterpart in ETSI BSA. The traffic jam ahead warning is the only case where DRIVE enhances ETSI BSA. The third column shows that sim^{TD} enhances the BSA of ETSI. Also, a traffic jam ahead warning exists here as well. In the category of safety-related use cases sim^{TD} also evaluated C2X based cross-traffic assistance. In the category of traffic efficiency, the project came up with several novel applications. sim^{TD} also introduced a function in the

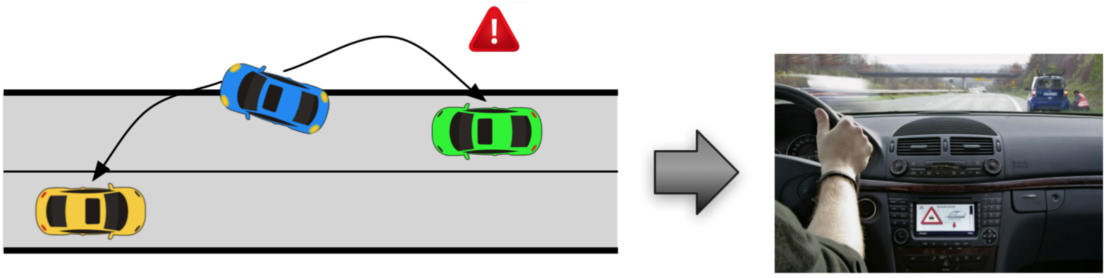


Figure 3.1.: Local danger warning on the example of a broken down vehicle. By turning on warning lights, broken down vehicles broadcast DENMs (blue). On a positive relevance check by a receiving car (green), a warning message is shown. The example on the right shows the implementation within the research project WILLWARN.

category of infotainment and others. These are summarized as “Internet-based services”. The reason for the enhanced function list of sim^{TD} is the method by which they where selected. This selection process also included questionnaires to stakeholders in which they where asked to contribute additional applications. In terms of categorization, Table 3.2 also shows an even stronger focus on safety and traffic efficiency than ETSI BSA. In the following, a more detailed description is given on the basis of sim^{TD} functions. As shown, this research project not only covers a wide range of use cases of ETSI BSA but also introduces novel functions.

3.2. Safety

The initial idea of connected vehicles was to improve road safety. By making cars aware of each other, crashes in different situations could be avoided. All applications which help to improve traffic safety belong in this category.

3.2.1. Local Danger Warnings

Use Cases 1-4 and 6-7 can be summarized as “Local Danger Warnings”. In these scenarios, the driver is informed about local traffic events, like-slow moving or broken-down vehicles, before actually arriving at the location (Figure 3.1). When driving on a motorway at high speed and with poor visibility, these use cases can contribute significantly to increased road safety. Information about a specific traffic event can either be sent out by the causing vehicle using DENMs or broadcast by traffic authorities using mobile connections. In C2X research, the following local traffic events are recognized: broken down vehicle, slow-moving vehicle, pedestrians on the road, traffic jam ahead warning, approaching emergency vehicle warning, and road weather information.

The local danger warnings described in this section are not exhaustive. For example, [Réolon, 2007] presents a local danger warning ontology (LDWO) that adds a number

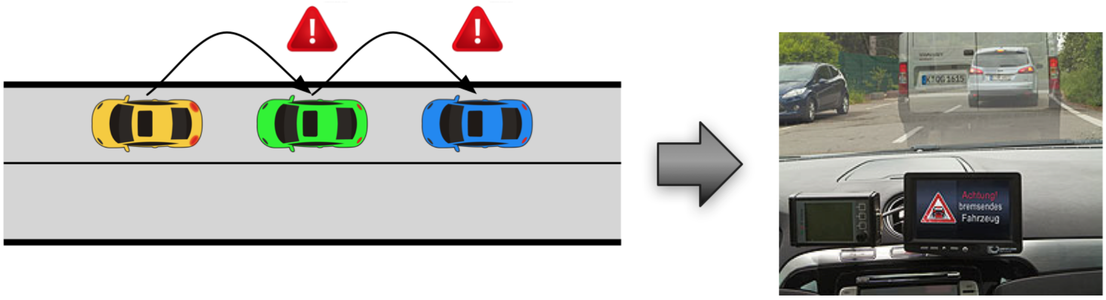


Figure 3.2.: Electronic brake assistant according to ETSI: A car driving ahead brakes suddenly (yellow) while another vehicle is blocking the line of sight (green). The driver (blue) receives a warning on the HMI and is able to avoid a crash. Implementation in sim^{TD} is shown in the overlay on the right (Image source: Ford).

of additional types, e.g. icy road. However, this section puts emphasis on standardized applications and their use cases. Therefore, these additional types are not considered.

3.2.2. Electronic Brake Light

Rear-end collisions are dangerous situations on the roads and account for a significant number of accidents today. The electronic brake light aims to prevent rear-end collisions by informing the driver through visual, auditory or haptic modalities when drivers engage brakes for safety. This use case is specifically intended for situations where line of sight between the EGO and the car ahead is blocked by another road user (see Figure 3.2). It is important to note that there is no autonomous braking of the vehicle. The assessment of the situation remains to be made by the driver.

3.2.3. Cross-Traffic Assistance

This use case provides assistance in side collision scenarios. Algorithms project paths of vehicles within the C2X neighborhood and match them to the path of the EGO vehicle. In case of an impending crash, a warning is issued to the driver. In most implementations, the scenario involves two cars: one driving on a major road, the other approaching from a minor road. The latter violates right-of-way and receives a warning. The driver of the car on the major road receives no warning. A modified version of this scenario also involves motorcycles. In a vendor-specific implementation by BMW, additional modalities are used in order to draw attention to the motorcyclist: The horn, headlamps and additional LED warning lights mounted on the motorcycle come on automatically (Figure 3.3). In contrast to standard issued by ETSI, this realization also autonomously triggers 30% of the car's braking power for one second. Usage of light and horn is probably not always reasonable as it might distract other road users not involved. This will be discussed in detail in Section 6.6.

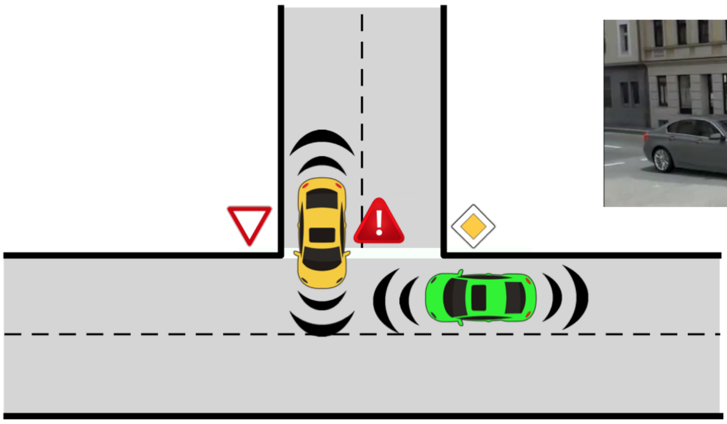


Figure 3.3.: Cross traffic assistance according to ETSI. If the system detects that the car driver is unlikely to yield, a warning is issued. In the example of the BMW (upper right), the additional warning lights and horn on the motorcycle come on automatically. Picture: BMW.

3.3. Traffic-Efficiency

A second class of use cases within the Push Paradigm is concerned with traffic efficiency. Here, C2X technology is used to improve the traffic flow within areas of high traffic density. Most use cases are based on C2C and C2I / I2C communication.

3.3.1. Traffic Sign Assistant

The traffic sign assistant specified by ETSI informs the driver about current valid traffic signs (see Figure 3.4, right). The ETSI standard suggests using I2C communication in order to transmit necessary data to the car. In implementations, this approach is applied to dynamic traffic signs on motorways. In cases of static traffic signs, a database is used and updated periodically using cellular networks. It contains GPS locations of signs as well as well as an approach path on which the vehicle is matched. Depending on the individual road situation, this use case can also be classified as a safety application. In the event of a traffic violation, it is intended to warn the driver, e.g. when exceeding speed limits.

3.3.2. Traffic Light Assistant

This scenario uses I2C communication in order to broadcast traffic light timing information to cars in transmission range. From this and onboard sensor data, the application logic calculates an adequate speed that is just right to pass synchronized traffic lights. Remaining red time can be shown while waiting for the green signal (see Figure 3.4, right). Of course, recommended speed does not exceed currently valid limits and remaining green time is never presented. This use case also addresses a safety aspect: In case

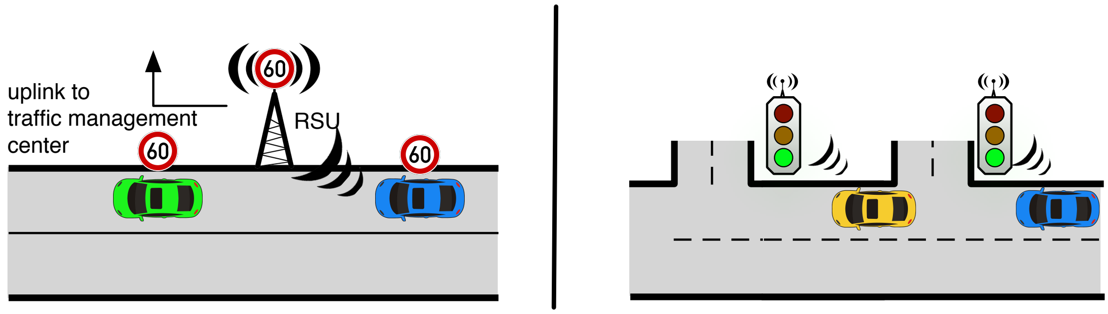


Figure 3.4.: **Left:** Traffic sign assistant in the Push Paradigm: A roadside unit broadcasts speed-limit information using CAMs. According to ETSI, a wired connection to the traffic management center is established in order to cope with dynamic sign content, **right:** Traffic light assistant. With the help of status information broadcast by road infrastructure, the application running in the car calculates and recommends optimal speeds to the driver for passing upcoming traffic lights during the green phase.

of an impending red light violation, the driver can also be warned.

3.3.3. Street Foresight

Street foresight visualizes highly dynamic traffic flow information in the close vicinity. This enables the driver to adapt her route to current traffic situations. Moreover, location information of other vehicles is accumulated into a “self-experienced” map. Street foresight is based on C2C communication and has been introduced in sim^{TD}. Neither ETSI BSA or other research projects considered this use case.

3.3.4. Roadworks Information System

The sim^{TD}-specific use case roadworks information system optimizes traffic flow in the presence of roadwork. It presents information about the geometry of the roadwork and traffic flow during the approach. It supports the driver in choosing a lane with less traffic. Additional information is also available (length, reason, additional traveling time, etc.). The geometry is calculated using vertical acceleration data from vehicles that previously passed the construction site. In order to communicate this geometry to other cars, either I2C communication or cellular networks can be used. The first relies on RSUs located at the beginning and end of the construction area (see Figure 3.5). The second follows a database approach similar to the traffic sign assistant. All necessary data is downloaded using a mobile connection.

3.3.5. Diversion Management & Enhanced Navigation

Diversion management connects electronic overhead gantry signs and vehicles using I2C communication. On-time information about traffic events issued by traffic control cen-

Figure 3.5.: Roadworks information system in sim^{TD}. The yellow car delivers data from vertical acceleration sensors to an RSU when leaving the work area. Its geometry is computed and transmitted to the blue car entering the roadworks. While driving through the roadworks, information on geometry, traffic flow and estimated time to end is displayed. The image in the middle shows the visualization within sim^{TD} for the green car.

ters is available even when signs have been passed. The driver is able to adapt her route according to this information. In sim^{TD}, in Germany well-known dWiSta[1] signs are equipped with I2C communication technology in order to realize this use case.

Using information about traffic events along a planned route, the enhanced navigation re-calculated automatically. The driver was informed using the HMI, which displayed the initial and re-planned routes. This use case was specific to sim^{TD}.

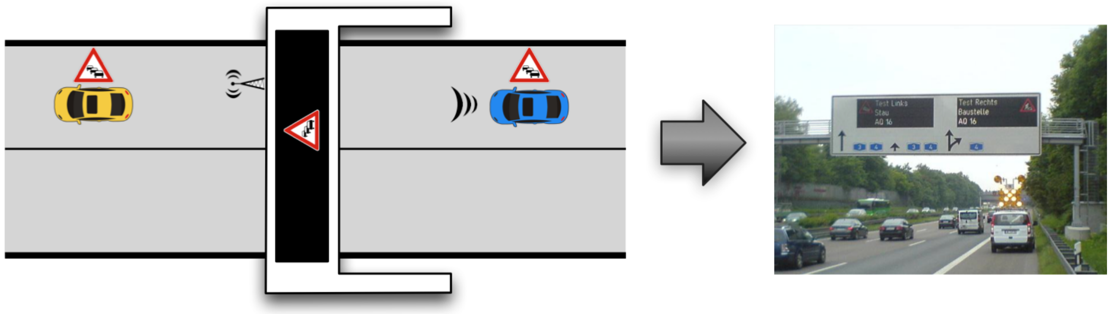


Figure 3.6.: Electronic overhead gantry sign (dWista) used on German motorways. I2C technology broadcasts information to the vicinity. Drivers are able to access this information within the area of validity of the sign.

3.4. Infotainment and Other

Infotainment use cases within the Push Paradigm are often based on cellular networks, not C2C or C2I communication. sim^{TD} introduced several added-value services, which

[1]dWiSta: Dynamische Wegweiser mit integrierten Stauinformationen (dynamic direction sign with integrated traffic jam information)

are presented here.

3.4.1. Internet-Based Services

sim^{TD} evaluated the following use cases in this category:

- **Weather forecast** included detailed information on today's forecast as well as the next three days for several regions in Germany.
- **Urban information** displayed details about local events, e.g. exhibitions. Turn-by-turn navigation to an event could be activated on the detail screen of the POI.
- **Traffic events** were displayed on individual road segments indicating traffic jams or accidents. Details for each event could be obtained by clicking POI icons on a map.
- **Webcams** provided visual access to important transport hubs. Images were updated at regular intervals; see Figure 3.7.
- **Parking information** was available by clicking on POI icons on the map. Colors indicated their current occupancy. Turn-by-turn navigation to a parking garage could be activated on the detail screen of the respective POI.

All information for use cases presented in this category were downloaded using UMTS and updated at regular intervals. Only webcam pictures were checked for a more recent version while the user viewed a cached image. If a newer version existed, the image was downloaded and updated on the HMI.

Figure 3.7.: One example of Internet-based services in sim^{TD}. By clicking on one of the POI icons in the left image (red circles), still images of the location were downloaded via UMTS.

3.5. Car2X Applications in Products

In a broader sense, several Car2X based applications have already been introduced to the market during the last few years. Vehicle manufacturers equipped their cars with internet access and provide mostly added-value services using mobile radio technology, e.g. UMTS or LTE. However, up to now vendor-independent communication and standardized applications have not been available. Manufacturers deliver their own services and they are targeted to deliver information to the driver, or from the car to the manufacturer. Direct communication between cars, regardless of the communication technology used, is not in a productive state and still subject to further research. In the following, a representative selection of currently available products based on Car2X is given:

- News, weather, local point of interest search, access to office features
- Live traffic information integrated in navigation systems
- Notification to towing services in case of a breakdown
- Connection to vendor service center
- Integration of social networks
- Road tolls
- Music and video streaming

In order to give the reader a better understanding, two concrete products from two major German car manufacturers are described. Note that the individual products may have additional features like navigation but are not listed here since the focus is on connected cars.

3.5.1. Mercedes COMAND Online

Mercedes subsumes all functionalities regarding connected cars under the product name COMAND Online[2]. All communication is relayed over a mobile phone connected via Bluetooth to the car. In contrast to BMW Connected Drive, no dedicated SIM card is installed into the car.

Basic functionalities of COMAND Online include internet browsing (only in a stationary vehicle), weather information, and Google local search. Mercedes also runs an app store that provides access to extended functionalities that can be installed by the customer. Among others, this includes Facebook, stock applications, and information about parking lots. Dynamic traffic information is also available, but it is received via Traffic Message Channel (TMC) which is a wireless broadcast technology that is based on a non-audible frequency of common radio broadcasting. COMAND Online also comes with an additional safety feature: As soon as sensors detect a crash, e.g. the airbag released, the car's position is transmitted to a human operator who then tries to call the driver and/or requests medical assistance (eCall).

[2] http://techcenter.mercedes-benz.com/de_DE/comand_online/detail.html

The driver interacts with the system using the COMAND Controller, a turn-and-push dial mounted in the center console. Alternatively, shortcuts to important functionalities are accessible via dedicated buttons around the controller.

3.5.2. BMW Connected Drive

BMW Connected Drive[3] consists of safety, comfort, and infotainment functions. Internet connectivity is achieved by an integrated SIM card that uses either UMTS or LTE. Alternatively, an internet connection is established by a smartphone via Bluetooth, but according to the product description, most of the services are available using the integrated SIM card. Another interesting feature is an integrated hotspot that shares the internet connection with up to 8 co-drivers over consumer WiFi.

BMW offers a remote app for smartphones that provides access to comfort functions like climate control or fuel level. Furthermore, the car can be locked/unlocked, or located on a map. Another interesting feature includes route planning at home and then sending it to the car. When the driver returns to her car, she can immediately start her trip.

Infotainment functionality offers the possibility to access emails, text messages, and contacts. They can be read to the driver or composed and sent using speech. Other services include news, Google local search, Facebook, and a calendar. As with Mercedes, an automatic emergency call is integrated.

3.6. Conclusion

In this chapter established ADAS/IVIS based on unidirectional Car2X Communication were described. The analysis of applications that have been standardized by ETSI has shown a strong focus on safety aspects. Another large group of applications addresses traffic efficiency. The comparison of these standards to actually implemented and tested ADAS/IVIS in two of the the largest field operational tests for Car2X Communication revealed an information-centric alignment. The third group of applications, infotainment and other, is not based on vehicular ad-hoc communication. For example, parking and urban information tested in sim^{TD} rely on cellular networks. A review of already available products in the context of Car2X Communication also showed the connection to this wireless communication technology. Furthermore, it can be seen as a transfer of existing technology into the car, e.g. accessing emails.

In the following the findings of Chapters 2 and 3 are reviewed as a whole and missing aspects in current state of the art Car2X networks are derived. From these results, necessary extensions are identified to implement the Pull Paradigm.

[3] http://www.bmw.de/de/topics/faszination-bmw/connecteddrive-2013/ubersicht.html

4. Overcoming the Limitations of Current Car2X Networks

In the previous chapter, the state of the art in Car2X Communication networks on application level as well as the underlying technical foundations were described. Following up these findings and in order to be able to introduce a novel Car2X paradigm, missing aspects for user-centric applications based on vehicular ad-hoc communication are identified in the following.

4.1. Human Factors

From a human factors perspective, the established Car2X technology of the Push Paradigm is not open to users. The previous chapter clearly showed the technical focus of research in this area. In past years work on protocols and standards was mainly driven by pre-commercial development. The effect on drivers and passengers was almost not considered. The "last mile" to the user, the Human Machine Interface, was of comparably little significance. With the execution of field operational tests that were conducted in real traffic, comprehensive work on human factors was needed not only because of legal reasons but also to improve user acceptance. For example, in sim^{TD}, 400 drivers inexperienced with the system and technology in general were paid to participate and deal with Car2X from a user's perspective. Their subjective ratings were included in the project results afterwards. Inconsistent, non-intuitive or not adequate timely presentation of functions would have significant effects on user acceptance. DFKI, especially the author of this thesis as main developer of the human-machine interface, pioneered work in this area. With the development of an HMI Interface tailored to Car2X applications, the groundwork was laid to involve drivers and let them experience the technology. However, since only applications within the Push Paradigm were developed in sim^{TD}, the driver remained an information receiver. The HMI is included as a module in the Car2X development platform that is going to be introduced in Chapter 7. The sim^{TD} HMI served as the visualization component for all relevant functions during the field test and was considered a robust and reliable system component. In the context of this thesis it will also include novel Pull applications.

It should be made very clear at this point that the HMI is mainly used to present results of a Pull request and to integrate Pull applications into the existing diversity of ADAS/IVIS. The author is aware of the fact that the analysis and decomposition of complex requests to the ad-hoc network requires a comprehensive input concept for

Europe, USA, Japan: A total of 79 projects				
		Focus		
		Vehicle human	Infrastructure human	Technical
Application type	Safety	6	1	34
	Traffic Efficiency	2	1	23
	Infotainment	0	0	3
	Independent	1	1	28

Figure 4.1.: Relevant results of a state of the art survey of Car2X Communication projects by [Strandén et al., 2008b]. 79 research projects were analyzed. The focus of every application type is clearly technical, while the category infotainment is least considered.

automotive HMIs, e.g. a multimodal interface as presented in [Castronovo et al., 2010b; Kern et al., 2010]. Results of the previous Chapters show that such a concept does not exist at the moment in Car2X research, not least because of the main application areas of the Push Paradigm (also see [Castronovo et al., 2010a]). However, this thesis focuses on enabler technologies for implementing the Pull Paradigm. The input of complex Pull requests while driving is left open for future work. In this thesis, only *Meta-Pull* requests are considered. Here, the user inputs a simpler request which is then augmented by information that is already available, e.g. width of the car (recall Bob's search for an adequate parking spot in Section 1.1).

4.2. Applications

From an application point of view, the current state of the art is primarily focused on safety- and traffic-related use cases. This was clearly shown by looking at standardized applications by ETSI and the C2C CC. These were mapped to the use cases that have been implemented in two of the largest FOTs for Car2X Communication. As these research projects prepare roll-out strategies for upcoming market introduction, they represent even more the focus of the community. Here too, selected applications for evaluation were to a large extent from the categories of safety and traffic efficiency. These results are also supported by a state-of-the-art survey of wireless vehicular communication projects [Strandén et al., 2008b]. They analyzed 79 Car2X projects that were conducted in Europe, the USA and Japan as these are considered to be the most dominant countries in this research area. Every project was classified according to a developed template which lists some basic information on the project like duration and size as well as research foci. Particularly interesting are columns *Vehicle Human* and *Infrastructure Human*. These describe the focus of the individual application type listed in rows. The column *Vehicle Human* then describes a Car2X project which focuses on driver and/or passenger, while the column *Infrastructure Human* focuses on operator or management persons. In case a project matches a certain area of work, the respective cell is marked by an "x". Note that the meanings of application types are the same as the ones introduced in Chapter

3.
Figure 4.1 shows relevant results for metric *total scope* which describes *were efforts have been made and where they have not.* The number in a cell represents the accumulated number of all projects in all evaluated countries. It clearly shows a focus on traffic efficiency and safety but also shows that mostly technical questions are addressed. Furthermore, infotainment applications on the basis of Car2X are also out of scope at the moment. Among other results, Strandén et al. conclude that projects in Europe and USA end with a demonstrator and theoretical results, e.g. standards, while Japan puts results in production and deployment. The full analysis can be found in [Strandén et al., 2008a].

In summary, user-centric Car2X-based applications are currently not in focus.

4.3. Technology, Protocols and Architectures

The previous Chapters also showed that the strong focus on safety and traffic efficiency is also reflected by the underlying technology, protocols and architectures.
On the facility layer, for example, two standardized data structures are used for ad-hoc communication: CAMs and DENMs. The contents of a DENM in particular are tailored to safety applications as explicitly described in the message format definition by ETSI [European Telecommunications Standards Institute (ETSI), 2010b]. Its fields include information like “severity” or “trafficFlowEffect” which clearly addresses safety issues. These message formats are not generic enough to realize interactive, added-value services on the basis of vehicular ad-hoc communication. This is also backed up by state-of-the-art literature where infotainment applications are excluded from using ad-hoc communication technologies: *“In-car services refer to features such as in-car entertainment [...]. These features do not use vehicular communication and therefore are not further detailed.”* [Popescu-Zeletin et al., 2010]. On an architectural level, this could also be verified by looking at the protocol architecture of an OBU specified by the C2C CC [Baldessari et al., 2007]. According to this, infotainment applications do not use the C2C transport layer. They are realized using “other radio” such as UMTS.

Routing algorithms in this domain also are optimized for quick information dissemination across the ad-hoc network. Several research efforts try to predict vehicle movement [Granelli et al., 2007] or use maps to route data packets over intersections [Seet et al., 2004]. Two major limitations of these apporaches could be identified. First, the special characteristics of vehicular ad-hoc networks are not fully exploited. Second, direct communication between nodes lacks approaches to discover the intended communication partner and keep a stable route between sender and receiver. Stable, low latency, and bidirectional communication between multiple cars, however, is a key requirement for applications of the Pull Paradigm. Therefore, this problem is addressed in more detail in Chapter 5, where algorithms advanced algorithms for managing information flow in C2X networks are elaborated.

While the potential of ad-hoc communication for infotainment applications is obvious, e.g. low latency times, the current state of the art lacks support for this type of applications on the technological, architectural and protocol levels.

4.4. Simulation

As outlined in Section 2.2.3, a realistic simulation of Car2X networks can only be achieved by coupling network and traffic simulators. This has been done in the past by several researchers in the community. The approaches were analyzed, and according to mandatory and optional requirements of this thesis, VSimRTI has been identified to be the most promising one. So far, primarily technical questions were answered using these coupling systems. However, studying effects on drivers by considering limitations of Car2X technology at the same time cannot be addressed by current approaches, as they do not integrate a driving simulator. The generic architecture of VSimRTI allows this integration. Therefore, Chapter 7 addresses this topic and adds OpenDS [Math et al., 2012].

This leads to the following drawbacks of current Car2X networks:

1. Added-value services on the basis of Car2X Communication technology are not considered.
2. User-centric applications that acquire information from the (ad-hoc) network on the fly do not exist.
3. The content of CAMs and DENMs is specifically tailored to safety-related applications. They lack generality for non-safety applications using vehicular ad-hoc communication.
4. Unicast ad-hoc communication does not address the problem of discovering the target node.
5. Routing algorithms do not fully exploit special characteristics of vehicular ad-hoc networks in order to ensure reliable communication.
6. Simulation of Car2X networks lacks integration of a driving simulator that enables the evalation of user-centric applications.

4.5. The "Pull" Paradigm: Adding Bidirectional Communication to Car2X Networks

In the previous section, current limitations of Car2X networks with respect to the Pull Paradigm were presented. This leads to necessary enhancements which are shown in Figure 4.2.
On the network level, a new message type that reflects Pull requests needs to be defined,

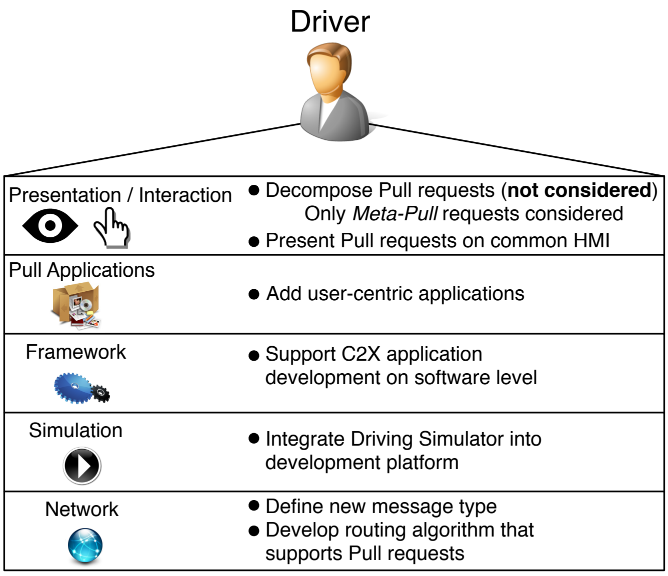


Figure 4.2.: Necessary enhancements of today's Car2X Communication networks in order to implement the proposed Pull Paradigm.

as CAMs and DENMs are tailored to safety applications. Although unicast communication is currently supported, questions of information source discovery and retrieval are currently unanswered. Both need to be stable and reliable in order to enhance the user experience with Pull applications. This challenge was already introduced in Section 1.2. Because the answer to this problem is an essential pillar for an implementation of the Pull Paradigm, it will be investigated in detail in the next chapter. Advanced algorithms for information management in Car2X networks will be elaborated. This addresses **RQ1** as defined in Section 1.3.
It has been shown that current main application areas focus on safety and traffic efficiency. Therefore, novel applications within the Pull Paradigm have to be defined. Chapter 6 is dedicated to this topic and therefore to addressing **RQ2**.

Currently, there exists no end-to-end software platform that enables the development of Car2X applications. In order to support development on the software level, a corresponding framework is developed. It features communication support and also integrates a common HMI framework that coordinates parallel-running ADAS/IVIS. By doing so, developed applications are well integrated into the vehicle. Note, that only the presentation part is addressed in this thesis. Although, the HMI framework is able to accept input from the driver, the interpretation of complex Pull requests is not considered in this thesis. Only *Meta-Pull* requests are assumed (cf. Section 4.1). The software platform is introduced in Chapter 7 as the Car2X Platform for Application Development and Evaluation (C2X PADE) and addresses **RQ4** and **RQ5**. In the same context, simulation tools are added in order to accommodate **RQ3**.

5. Advanced Algorithms for Managing Information Flow in Car2X Networks

5.1. Mapping Car2X Networks to Unknown Fully Dynamic Graphs

Reliable information source discovery and retrieval within vehicular ad-hoc networks were identified as key requirements for implementing the Pull Paradigm in Section 1.2. It was also argued that current routing protocols integrated into the Car2X network stack do not fulfill these requirements (cf. Chapter 4). Therefore, this chapter introduces advanced algorithms for managing information flow in dynamic environments. The problem is mapped onto the domain of graph theory, where nodes denote vehicles and edges wireless connections between them. This describes a *dynamic* graph since the topology is constantly changing, possibly very quickly. There are several advantages of this approach. First, the concentration can be on the real problem of information source discovery and retrieval without being restraint from technical details of Car2X Communication. Second, results from graph theory and artificial intelligence can be integrated. This broader view enlarges the scope of existing ideas from other domains that can possibly contribute to a more optimal solution. Third, visibility within the research community is not limited to a single domain.

Two algorithms are introduced: linked graphs heuristic search (LGHS) in Section 5.3 and then its extension LGHS-CD (CD = Cycle Detection) in Section 5.4. LGHS tackles the problem of shortest path discovery within dynamic graphs. The term "linked graphs" is introduced because LGHS relies on two graphs that correlate with each other: a dynamic one that captures the ad-hoc network and a static one that is derived from the road network. LGHS-CD adds cycle detection to LGHS, e.g. it discovers cycles between the *information seeking* and the *information providing* node where the latter is unknown at the beginning. This work is published in [Castronovo et al., 2012, 2013a]. Furthermore, the groundwork for LGHS and LGHS-CD was laid in the context of two Master's Theses [Del Fabro, 2013; Kunz, 2011]. Both were supervised by the author of this thesis.

In terms of the Pull Paradigm, LGHS discovers multi-hop routes between two cars while LGHS-CD adds information search and retrieval to ad-hoc networks. Furthermore, LGHS-CD introduces a decentralized information management component that speeds up future requests.

5.2. Terms and Definitions

Term	Summarized meaning
$G = (V, E)$	A graph G consisting of set of vertices V and edges E. Weighted and directed graphs are assumed.
$v \in V$	A vertex v is an intersection point of the graph. It denotes the starting and endpoints of edges.
$e = (v_1, v_2, w) \in E$	An edge e denotes a connection between two vertices. $v_1 \in V$ is called the predecessor of e, $v_2 \in V$ is called the successor of e and w denotes the weight of e (sometimes also written as $e.w$).
$\tilde{G} = (\tilde{V}, \tilde{E})$	A fully dynamic graph $\tilde{G}$ consisting of vertices $\tilde{V}$ and edges $\tilde{E}$. All of the operations mentioned above are allowed.
$\tilde{v} \in V$	A vertex of the dynamic graph. This is similar to the vertices in a non-dynamic graph, with the exception that $\tilde{V}$ may change over time.
$\tilde{e} = (\tilde{v_1}, \tilde{v_2}, w) \in \tilde{E}$	Every edge models a connection between two vertices. As in the non-dynamic case, $\tilde{v_1}$ is called the predecessor, $\tilde{v_2}$ is called the successor and w is the weight of the edge.
P	A path defined as a set of vertices $v_1, ..., v_n \in V$ on a non-dynamic, directed and weighted graph G.
$\tilde{P}$	A path defined as a set of vertices $\tilde{v_1}, ..., \tilde{v_n} \in \tilde{V}$ on a dynamic, directed and weighted graph $\tilde{G}$.
N	The set of vertices with connections to / from a single vertex $v \in V$ which are of length 1, i.e. $\{u \in V \mid \exists e \in E : e = (v, u, w)\}$. This set is specific to every node $v \in V$.
$\tilde{N}$	The set of vertices with connections to / from a single vertex $\tilde{v} \in \tilde{V}$ which are of length 1, i.e. $\{\tilde{u} \in \tilde{V} \mid \exists \tilde{e} \in \tilde{E} : \tilde{e} = (\tilde{v}, \tilde{u}, w)\}$. This set is specific to every node $\tilde{v} \in \tilde{V}$.

Table 5.1.: Overview of graph theory terms

In order to provide a common view and also be able to evaluate and compare approaches for vehicular ad-hoc networks to those in the artificial intelligence/graph theory domain, a common terminology is developed. It can then be applied to both domains. The terminology is used throughout this chapter whenever referring to the Pull Paradigm, routing protocols and shortest path algorithms.

Let $G = (V, E)$ be a weighted, directed graph with vertices $v \in V$ and edges $e \in E$. V and E are defined as finite sets and assume positive, real numbers for all weights. Whenever G is referred to, a static graph is assumed, e.g. its topology and weights do not change over time, especially not during shortest path computation.

$\tilde{G} = (\tilde{V}, \tilde{E})$ denotes a *dynamic* graph with vertices $\tilde{v} \in \tilde{V}$ and edges $\tilde{e} \in \tilde{E}$. In such a graph the following set of operations is allowed on its vertices and edges:

- insert a vertex $\tilde{v}$ into $\tilde{V}$

- remove a vertex $\tilde{v}$ from $\tilde{V}$
- insert an edge $\tilde{e}$ into $\tilde{E}$
- remove an edge $\tilde{e}$ from $\tilde{E}$
- change the edge weight of an $\tilde{e} \in \tilde{E}$

Generally speaking, all notations containing the symbol '$^\sim$' refer to elements of dynamic graphs. Since all of the above operations are allowed, it is a *fully dynamic graph* whose topology is constantly changing.
Discrete points in time T are defined and a partial ordering on these points in time is assumed: $\forall t_1, t_2 \in T$ if $t_1 < t_2$.
Shortest path $P = [(p_0, p_1), ..., (p_{n-1}, p_n)]$ on G connecting two vertices v, v' is defined such that

$$\sum_{i=0}^{n-1} (p_i, p_{i+1}).w$$

is minimal among all possible paths leading from v to v'.

Furthermore, $\tilde{N}$ denotes the neighbor set of a vertex $\tilde{v}$. $\tilde{N}$ contains all $\tilde{u} \in \tilde{V}$ for which the following holds:

$$\{\tilde{u} \in \tilde{V} | \exists \tilde{e} \in \tilde{E} : \tilde{e} = (\tilde{v}, \tilde{u})\}$$

Given an edge $e = (u, v)$ of a directed graph, u is called the **predecessor** and v the **successor** (in static and dynamic graphs).

Finally, the following functions are defined which describe topology changes in a highly dynamic graph. They enable the modeling of vehicle movements, road networks and wireless connections:

1. A function $\tilde{f} : \tilde{V} \times T \rightarrow E \times \mathbb{R}^+$ assigns a vertex $\tilde{v} \in \tilde{G}$ to exactly one edge $e \in E$ at a point in time $t \in T$. Moreover, it generates an edge $(v, \tilde{v})$ between vertex $\tilde{v}$ and the predecessor of assigned edge e with domain-specific weight. For example, the weight of $(v, \tilde{v})$ can be interpreted as the distance vehicle $\tilde{v}$ travelled along road e. In the application domain $\tilde{f}$ realizes a specific mobility model.

   The domain of $\tilde{f}$ is $\tilde{V} \times T$, thus enabling the algorithm to derive the neighboring topology for a certain $\tilde{v} \in \tilde{V}$ and a discrete point in time $t \in T$. For the algorithm, $\tilde{f}$ is restricted to vertices contained in $\tilde{N}$, the set of direct neighbors to a $\tilde{v} \in \tilde{V}$. Furthermore, queries to future graph topology return null. Also note, that the dynamic graph topology changes continuously but queries to $\tilde{f}$ are restricted to

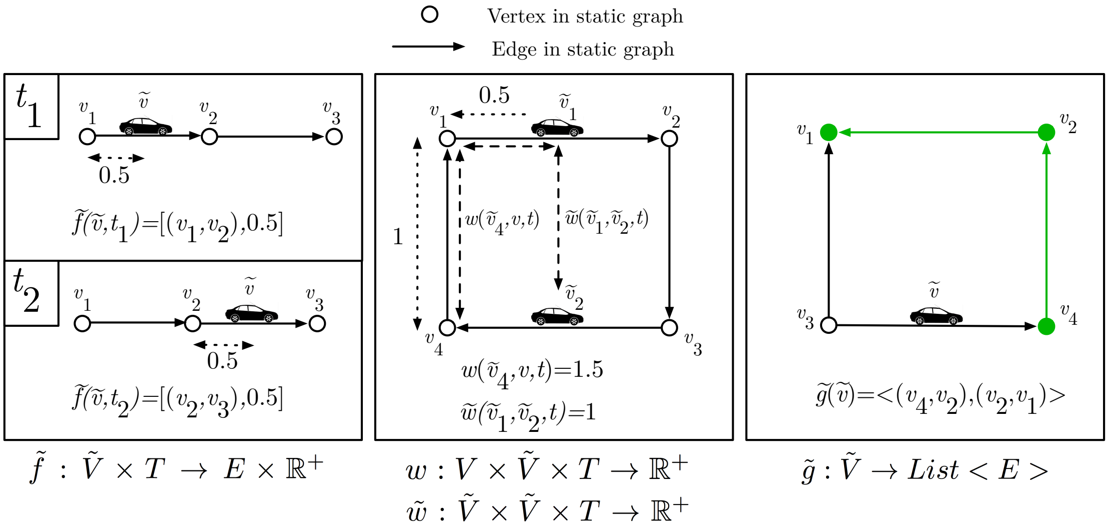


Figure 5.1.: Visualization of $\tilde{f}$, w, and $\tilde{g}$.

discrete points in time. This means, that only a *snapshot* of the neighboring topology can be retrieved which diverges (in the worst case) from the actual topology the time-span between two discrete points in time t and $t+1$. Figure 5.1 (left) visualizes $\tilde{f}$.

2. A function $w : V \times \tilde{V} \times T \to \mathbb{R}^+$ gives information about the correlation between the two graphs G and $\tilde{G}$ and returns a weight of an edge $(v, \tilde{v})$ at time $t \in T$. As for $\tilde{f}$, only queries to direct neighbors are possible, e.g. elements in $\tilde{N}$. In contrast to $\tilde{f}$ this weight can be interpreted as the euclidian distance of a vehicle to an intersection. A similar function $\tilde{w} : \tilde{V} \times \tilde{V} \times T \to \mathbb{R}^+$ returns the weight of an edge formed by two dynamic vertices, e.g. the straight-line distance to another vehicle. A visualization of w and $\tilde{w}$ can be found in Figure 5.1 (middle).

3. A function $\tilde{g} : \tilde{V} \to List < E >$ describes an ordered sequence of edge assignments for any $\tilde{v} \in \tilde{N}$ to edges $e_1, ..., e_n \in E$. Note, that $\tilde{g}$ gives no information about time. In vehicular ad-hoc networks this realizes the most probable path of a vehicle. An example of $\tilde{g}$ is depicted in Figure 5.1 (right).

Table 5.1 comprises the introduced terms and serves the reader as a reference throughout this chapter.

5.3. LGHS: Discovering Shortest Paths to Information Sources

Shortest path problems on graphs have been thoroughly studied in the artificial intelligence and graph theory literature. Single-source shortest path (SSSP) or all-pair shortest path problems with positive edge weights can be solved using widely known algorithms, such as Dijkstra's Algorithm [Dijkstra, 1959] and A* [Russell and Norvig, 2003]. There are a large number of applications: network protocols use them in order to route data packets from one physical location to another and many planning systems in artificial intelligence implement variants of these algorithms. Furthermore, they are applied to agents and computer games.

The problem becomes significantly more challenging when certain dynamic operations on the graph are allowed while searching, such as insertions or deletions of vertices as well as changes of edge weights. This problem drew the attention of research in artificial intelligence and related graph theory fields. Among others, approaches by [Nannicini and Liberti, 2008; Koenig et al., 2004; Misra and Oommen, 2004; Cicerone et al., 2003; Demetrescu and Italiano, 2003; Frigioni et al., 2000] assume global knowledge on the graph operations performed. This means they are only applicable on graphs for which the topology and all executed operations are known at every point in time. A short overview of three of these approaches is given below.

[Koenig et al., 2004] propose an incremental version of the popular A* algorithm based on a dynamic gridworld. This world consists of cells which denote vertices. Edges are drawn between neighboring cells. The algorithm continuously finds shortest paths between a start cell s_{start} and a goal cell s_{goal}. Their approach is incremental since they reuse results from previous searches. However, they assume global knowledge about the gridworld: The algorithm holds data structures for the start distance of every cell as well as a list of traversable and blocked cells. In the domain of cooperative vehicles this assumption cannot be made: Transmission range is limited and therefore no global knowledge of the graph can be derived.

The approach of [Misra and Oommen, 2004] transfers the problem into the domain of learning automata. They establish three components: the learning automaton (LA), which is instantiated in every vertex; the random environment (RE); and a penalty/reward system. RE changes edge weights using stochastic information and LA constantly interacts with this environment by guessing whether or not a node belongs to the shortest path. The LA constantly receives rewards or penalties depending on whether its guess was right or not. Although the distribution of the weight changes by the RE is unknown to LA, it is allowed to retrieve a snapshot of the whole graph and its current edge weights. Basically, this snapshot, Dijkstra's algorithm and the received penalties/rewards are used for shortest path computation. It is obvious that in Car2X networks such a snapshot of the whole graph is not available.

[Frigioni et al., 2000] proposes fully dynamic algorithms by counting the vertices affected by changes to the graph. When increasing the weight of an edge, the affected vertices are those which change the distance from the start vertex. The algorithm marks vertices according to their status after a graph operation; white marked vertices were not affected by an operation. Neither the distance from the source nor the parent in the shortest path tree changed. Red-marked vertices increased their distance from the source, while the distance of the ones marked pink remained constant, but they replaced the old parent in the shortest path tree. Obviously, this approach also relies on global knowledge about the graph topology and the operations on it.

This knowledge can neither be assumed nor derived in vehicular ad-hoc networks by the very nature of the domain. Consider vertices as vehicles and edges as wireless connections between two vehicles: The task is to send data from car A to car B, and B is not in A's transmission range. While searching a path to B, all of the operations above can occur in arbitrary order and number, because the topology of the underlying graph is subject to fast deletion and insertions of vertices and the algorithm has no information about these changes. Hence, it is referred to as a "fully dynamic problem" (Dynamic Single-Source Shortest Path Problem, or DSSSP). In this example, the dynamic operations may even break the connection between vertices. Furthermore, it is not efficient and sometimes even not feasible to visit every vertex after a graph operation. Network bandwidth is considered a limited resource in vehicular ad-hoc networks, which is also a reason why the cited algorithms are not applicable here.

In the following subsections, a combined uniform and heuristic search algorithm is proposed, which is able to solve the single-source shortest path problem under *unknown* topology changes in fully dynamic graphs. In order to do so, two graphs are instantiated: The first, $G = (V, E)$, is static. The single-source shortest path problem is solved on G using Dijkstra's algorithm. The second, $\tilde{G} = (\tilde{V}, \tilde{E})$, is subject to the described graph operations. Afterwards, *domain specific* relations between the two graphs are exploited and a shortest path identified on G is approximated in $\tilde{G}$. The algorithm only depends on *local* information about graph operations in order to heuristically maintain or alter the initial path found in G.

5.3.1. Algorithm Overview

LGHS is divided into three phases. At first, these phases are executed in the vertex where shortest path computation is started. The result is a $\tilde{v} \in \tilde{N}$ which is selected to be included in $\tilde{P}$. After that, the algorithm continues in the previously chosen $\tilde{v} \in \tilde{N}$ and continued until a goal condition is reached, e.g. the destination vertex is found. Each phase consists of several steps which are summarized in Table 5.2:

- **Phase 1** checks for termination and estimates future graph topology.
- **Phase 2** computes heuristics on which the decision for selecting the next vertex in $\tilde{P}$ is based.
- **Phase 3** selects the next vertex for $\tilde{P}$ from candidates contained in $\tilde{N}$. The decision is based on heuristics from Phase 2.

At the beginning of **Phase 1** it checks whether the destination vertex $\tilde{d} \in \tilde{V}$ is a direct neighbor, e.g. an element of $\tilde{N}$. In this case, no shortest path computation is necessary and the algorithm terminates after appending $\tilde{d}$ to $\tilde{P}$. If the destination vertex is not an element of $\tilde{N}$, the algorithm proceeds by solving SSSP on G using a modified Dijkstra strategy (Phase 1, step 2). The Dijkstra implementation takes the number of dynamic vertices assigned to a static edge into account: The more vertices are assigned to the same edge, the higher the preference for including this edge in shortest path computed by Dijkstra's algorithm. This is done before heuristics computation in order to accommodate fast topology changes within the graph.

Next, the future graph topology is estimated. This is especially necessary at low update rates by $\tilde{f}$ to the vertices in $\tilde{N}$. A vehicular ad-hoc network can be seen as a highly dynamic graph, and as a consequence this fact has to be taken into account. A car traveling at 36 m/s on a motorway covers a substantial amount of a car's transmission range between two position updates. Hence an update by $\tilde{f}$ could delete an edge in $\tilde{G}$.

The following three steps in **Phase 2** are crucial for the algorithm since the choice of the next vertex for $\tilde{P}$ depends on them. For every vertex in $\tilde{N}$, three heuristics assess its value for reaching the goal vertex. Heuristic 1 prefers dynamic vertices that are "close" to elements of previously calculated shortest path P. Heuristic 2 tries to match future dynamic graph topology with shortest path P, while heuristic 3 calculates the progress of dynamic vertices along P. As a result, every vertex in $\tilde{N}$ gets assigned a value within $[0, 1]$ where higher values denote higher priority in selecting the next vertex for $\tilde{P}$. Each heuristic is multiplied by a weight of α, β and γ respectively. The computations of the three heuristics are described in detail below.

In **Phase 3** a vertex in $\tilde{N}$ for $\tilde{P}$ is selected based on the heuristics computed in Phase 2. If no suitable vertex can be identified, the algorithm recovers by restarting after the next update of $\tilde{f}$.

5.3.2. Phase 1: Estimating Future Graph Topology

Due to the highly dynamic nature of the graph, future graph topology is estimated using information provided by functions $\tilde{f}$ and $\tilde{g}$, which were defined in Section 5.2. After this step, three heuristics are computed in order to find candidates for the next vertex in shortest path $\tilde{P}$. Note that computation of those heuristics is based on the *estimated* topology, which is the result of this step, and not on the *actual* topology. This is justified by discrete update intervals of $\tilde{f}$ (see Section 5.2, definition of $\tilde{f}$).

The pseudocode for future topology estimation is given in Figure 5.2 and is executed for every $\tilde{v} \in \tilde{N}$. $\tilde{f}$ is queried for retrieving the assigned edge e and the weight of edge $(v, \tilde{v})$ at discrete points in time t and $t-1$. Using this information, $\tilde{f}(\tilde{v}, t).w - \tilde{f}(\tilde{v}, t-1).w$ is calculated. The result is added to the edge $\tilde{v}$ is assigned to. The following cases have to be distinguished:

- $f(\tilde{v}, t).e = f(\tilde{v}, t-1).e$ (At time t and $t-1$ $\tilde{f}$ assigns $\tilde{v}$ to the *same* $e \in E$)

#	Step	Action(s)
1	Check necessity of Single Source Shortest Path computation	Check if a $\tilde{v} \in \tilde{N}$ is the destination. Add $\tilde{v}$ to $\tilde{P}$ and terminate if true or continue otherwise.
2	Solve single-source shortest path problem on G	Change weights of edges in G according to number of dynamic vertices in $\tilde{N}$ assigned to edges in G. Use Dijkstra's algorithm to compute the shortest path from source to destination.
3	Update neighbor weights	Estimate future topology of $\tilde{G}$ with the help of previous updates by $\tilde{f}$.
4	Heuristic 1: Calculate edge weights between $v \in P$ and $\tilde{v} \in \tilde{N}$	Prefer vertices with minimal weight between $v \in P$ and $\tilde{v} \in \tilde{N}$. Weight the result with predefined α.
5	Heuristic 2: Calculate matches of $\tilde{v} \in \tilde{N}$ with edges in P	Prefer vertices that will be assigned to more edges of P with the help of $\tilde{g}$. Weight the result with predefined β.
6	Heuristic 3: Calculate edge weights along P	Prefer vertices with higher accumulated edge weights along P. Weight the result with predefined γ.
7	Update $\tilde{P}$	Select a vertex $\tilde{v} \in \tilde{N}$ according to computed heuristics in steps $4-6$ and add it to $\tilde{P}$, or wait until next update from f to $\tilde{N}$ if no vertex can be selected. Repeat steps $1-7$ until the destination is reached.

Table 5.2.: High-level steps and actions of the search algorithm.

- $f(\tilde{v},t).e \neq f(\tilde{v},t-1).e$ (At time t and $t-1$ $\tilde{f}$ assigns $\tilde{v}$ to *different* $e \in E$)

This affects the computation of Δw (lines 4-11, Figure 5.2). If the estimated future weight exceeds the edge weight of the currently assigned $e \in E$, $\tilde{g}$ is used in order to retrieve the most probable future assignment of $\tilde{v}$ (lines 12-20, Figure 5.2). Otherwise the estimated weight is stored for $\tilde{v}$.

5.3.3. Phase 2: Computing Heuristics

Phase 2 computes three heuristics for every $\tilde{v} \in \tilde{N}$. Each of the heuristics approximates the shortest path found in G either more quickly or more stably in $\tilde{G}$. However, there is a tradeoff between path stability and discovery time. Therefore, the individual heuristic is weighted by α, β and γ respectively in order to find a balance between the two options depending on what is required for the intended application. As described in Section 4.5, the "Pull" Paradigm requires as stable paths as possible. Consequently, heuristics are optimized for more stable paths instead of less time. Higher heuristic values denote higher priority in the selection process for the next vertex of $\tilde{P}$. The three heuristics are

```
for all (ṽ ∈ Ñ) do
  w_t ← f̃(ṽ, t).w
  w_{t−1} ← f̃(ṽ, t − 1).w
  if (f̃(ṽ, t).e == f̃(ṽ, t − 1).e) then
    Δw ← w_t − w_{t−1}
    w_est ← w_t + Δw
  else
    w_{t−1} ← f̃(ṽ, t − 1).e.w − f̃(ṽ, t − 1).w
    Δw ← w_t + w_{t−1}
    w_est ← w_t + Δw
  end if
  if (w_est > f̃(ṽ, t).e.w) then
    E ← g̃(ṽ)
    for (i ← 0; i < E.length − 1; i + +) do
      if (E[i] == f̃(ṽ, t).e) then
        ṽ.e ← E[i + 1]
        ṽ.w ← w_est − E[i].w
      end if
    end for
  else
    ṽ.w ← w_est
  end if
end for
```

Figure 5.2.: Estimate future graph topology

combined into a single value. In Phase 3 this value is used to decide which $\tilde{v} \in \tilde{N}$ is added to $\tilde{P}$. After this, the algorithm continues in this selected vertex.

Heuristic 1

The pseudocode for heuristic 1 is presented in Figure 5.3 (left). The idea here is to use function w in order to identify vertex $\tilde{v} \in \tilde{N}$ with minimum weight to a vertex $v \in P$. It is obvious to prefer these vertices for $\tilde{P}$ since they approximate initial calculated path P from Phase 1. The code is executed for $\forall \tilde{v} \in \tilde{N}$ from the previous step and computes $\forall v \in P$, the weight of an edge between a vertex $\tilde{v} \in \tilde{G}$ and a vertex $v \in G$ (line 4-7). Only values up to w_{max} are considered which is the total weight of P. (lines 8-10). Heuristic 1 is then defined as $m_1 = \alpha * (1 - \frac{w_{(v,\tilde{v})}}{w_{max}})$ (line 11). Recall that higher values denote higher priority in selecting the next vertex for $\tilde{P}$. Heuristic 1 optimizes stability of path $\tilde{P}$ since it prefers vertices $\tilde{v} \in \tilde{V}$ which correlate to vertices of P. In vehicular ad-hoc networks this can be interpreted as preferring vehicles that are close to street intersections. Normally, these indicate direction changes of vehicles driving on the road. This approach also has another advantage: Especially in a city environment where buildings block wireless transmissions, the probability of establishing stable data transmission to neighboring cars is increased. Figure 5.3 (right) gives a graphical view of heuristic 1.

for all $(\tilde{v} \in \tilde{N})$ **do**
  $w_{(v,\tilde{v})} \leftarrow \infty$
  for all $(v \in P)$ **do**
    if $(w(v,\tilde{v},t) < w_{(v,\tilde{v})})$ **then**
      $w_{(v,\tilde{v})} \leftarrow w(v,\tilde{v},t)$
    end if
    if $(w_{(v,\tilde{v})} > w_{max})$ **then**
      $w_{(v,\tilde{v})} \leftarrow w_{max}$
    end if
  end for
  $\tilde{v}.h_1 \leftarrow \alpha * (1 - \frac{w_{(v,\tilde{v})}}{w_{max}})$
end for

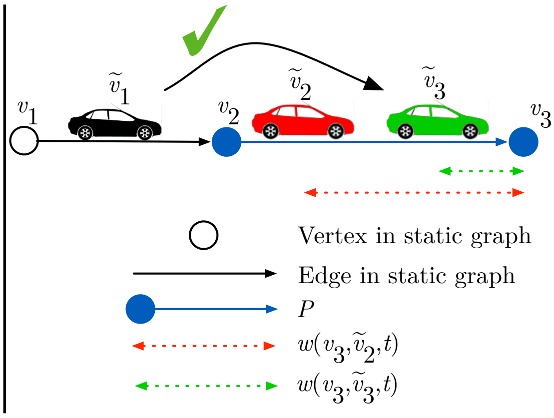


Figure 5.3.: **Left**: Heuristic 1: Proximity of $\tilde{v} \in \tilde{N}$ to $v \in P$. **Right**: Visualization of heuristic 1. Because $w(v_3, \tilde{v}_3, t) < w(v_3, \tilde{v}_2, t)$, $\tilde{v}_3$ is scored higher than $\tilde{v}_2$.

Heuristic 2

Heuristic 2 is computed as shown in Figure 5.4 (left). As for heuristic 1, the code is executed $\forall \tilde{v} \in \tilde{N}$. Function $\tilde{g}$ defined in Section 5.2 gives the most probable edge assignments of vertices in $\tilde{V}$ to edges in E. The idea of heuristic 2 is to identify edges of P that are contained in the list returned by $\tilde{g}(\tilde{v})$. The more matching edges the higher the preference for adding a vertex to $\tilde{P}$.

Recall that G is directed. Therefore, the direction of an edge is taken into account: An exact match results in score ω and τ if the assigned edge is a non-exact match. An exact match is given when edge (v_1, v_2) is contained in $\tilde{g}(\tilde{v})$ and also in P; a non-exact match when edge (v_2, v_1) is in $\tilde{g}(\tilde{v})$ and (v_1, v_2) in P. If the algorithm is not able to find a match at all, it assigns a score of 0. Note, that $\omega > \tau$ is required.

$Score_{max}$ is defined as $\sum_{i=0}^{|P|-1} \omega$ and heuristic 2 as $\beta * \frac{score}{score_{max}}$, yielding a value in the range $[0, 1]$ where larger values denote higher priority in selecting the next vertex for $\tilde{P}$. Finally, the weight β is assigned and heuristic h_2 is stored in $\tilde{v}$ (line 14). Heuristic 2 tries to optimize path stability by preferring those $\tilde{v} \in \tilde{V}$ whose own path closely matches path P. In vehicular ad-hoc network terms, this can be interpreted as preferring a vehicle that can carry the message closer to the destination.

Heuristic 3

The pseudocode for the third heuristic is given in Figure 5.5 (left). For every $\tilde{v} \in \tilde{N}$ it sums up weights starting at the first edge belonging to P and continuing until the assigned $e \in E$ of $\tilde{v}$ (lines 5-13). For vertices in $\tilde{N}$ which are not assigned to an edge belonging to P, it selects vertex v of P with the least weight to the assigned edge of $\tilde{v}$ (lines 15-30). The heuristic itself is then defined as $\frac{w_{(e,\tilde{v})}}{w_P}$ where $w_{(e,\tilde{v})}$ is the weight calculated and w_P the total weight of P.

As before, this calculates a value in the interval $[0, 1]$ which is multiplied by a weight

for all $(\tilde{v} \in \tilde{N})$ **do**
  $score_{\tilde{v}} \leftarrow 0$
  for $(i \leftarrow 0; i < P.size - 2; i++)$ **do**
    $p_1 \leftarrow P[i]$
    $p_2 \leftarrow P[i+1]$
    if $((p_1, p_2) \in \tilde{g}(\tilde{v}))$ **then**
      $score_{\tilde{v}} \leftarrow score_{\tilde{v}} + \omega$
    else
      if $((p_2, p_1) \in \tilde{g}(\tilde{v}))$ **then**
        $score_{\tilde{v}} \leftarrow score_{\tilde{v}} + \tau$
      end if
    end if
  end for
  $\tilde{v}.h_2 \leftarrow \beta * (\frac{score_{\tilde{v}}}{score_{max}})$
end for

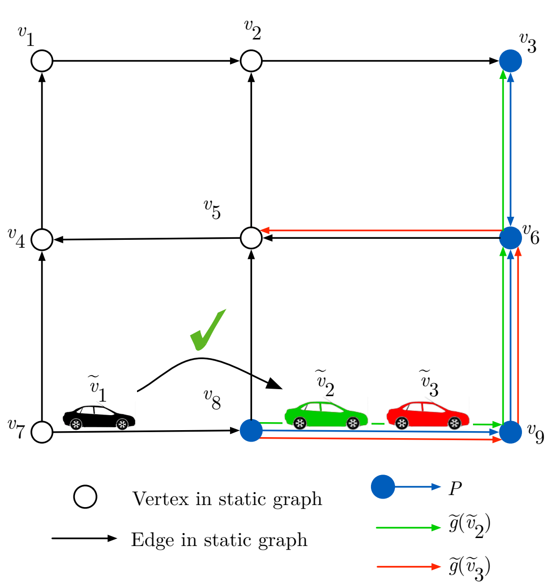


Figure 5.4.: **Left**: Heuristic 2: Assigncaments of $\tilde{v} \in \tilde{N}$ to edges of P. **Right**: Visualization of heuristic 2. According to $\tilde{g}$, $\tilde{v}_2$ (green) is assigned to more edges belonging to P in comparison to $\tilde{v}_3$ (red). Therefore, heuristic 2 scores $\tilde{v}_2$ higher than $\tilde{v}_3$.

of γ (line 31). Also here, larger values denote higher priority in selecting the next vertex for $\tilde{P}$. In contrast to heuristic 1 and 2, this heuristic tries to optimize progress towards the destination by choosing the next vertex $\tilde{v} \in \tilde{V}$ that is furthest along P and therefore optimizes for fast shortest path discovery. In vehicular ad-hoc networks this means choosing the vehicle furthest along the guiding path on the street map.

5.3.4. Phase 3: Selecting Best Neighbor

After calculating the heuristics, the vertex in $\tilde{N}$ for $\tilde{P}$ is selected. Let H be the set containing the sums of h_1, h_2, h_3 $\forall \tilde{v} \in \tilde{N}$. Then, the next vertex for $\tilde{P}$ is defined as the vertex with the largest sum in H. If the next vertex is $\tilde{P}[\tilde{P}.length - 1]$ (the current vertex), a recalculation is triggered after the next update of $\tilde{f}$.

5.4. LGHS-CD: Context-Aware Information Retrieval and Dissemination

Linked Graphs Heuristic Search with Cycle Detection (LGHS-CD) augments LGHS presented in Section 5.3 with the ability to return to the vertex in which the shortest path computation started, e.g. it starts in the information seeking node, discovers the information providing node and returns to the information seeking node. In graph theory

$w_P \leftarrow P.totalWeight$
for all $(\tilde{v} \in \tilde{N})$ **do**
  $e_{\tilde{v}} \leftarrow \tilde{f}(\tilde{v}, t).e$
  $w_{(e,\tilde{v})} \leftarrow 0$
  if $(e_{\tilde{v}} \in P)$ **then**
    for $(i \leftarrow 0; i < P.size - 2; i++)$ **do**
      if $(e_{\tilde{v}} == (P[i], P[i+1]))$ **then**
        $w_{(e,\tilde{v})} \leftarrow w_{(e,\tilde{v})} + \tilde{f}(\tilde{v}, t).w$
        abort for-loop
      else
        $w_{(e,\tilde{v})} \leftarrow w_{(e,\tilde{v})} + (P[i], P[i+1]).w$
      end if
    end for
  else
    $leastWeight \leftarrow \infty$
    $v \leftarrow nil$
    for $(i \leftarrow 0; i < P.size - 1; i++)$ **do**
      if $(w(P[i], \tilde{v}, t) < leastWeight)$ **then**
        $leastWeight \leftarrow w(P[i], \tilde{v}, t)$
        $v \leftarrow P[i]$
      end if
    end for
    for $(i \leftarrow 0; i < P.size - 2; i++)$ **do**
      if $(P[i] == v)$ **then**
        abort for-loop
      else
        $w_{(e,\tilde{v})} \leftarrow w_{(e,\tilde{v})} + edge(P[i], P[i+1]).w$
      end if
    end for
  end if
  $\tilde{v}.h_3 \leftarrow \gamma * (\frac{w_{(e,\tilde{v})}}{w_P})$
end for

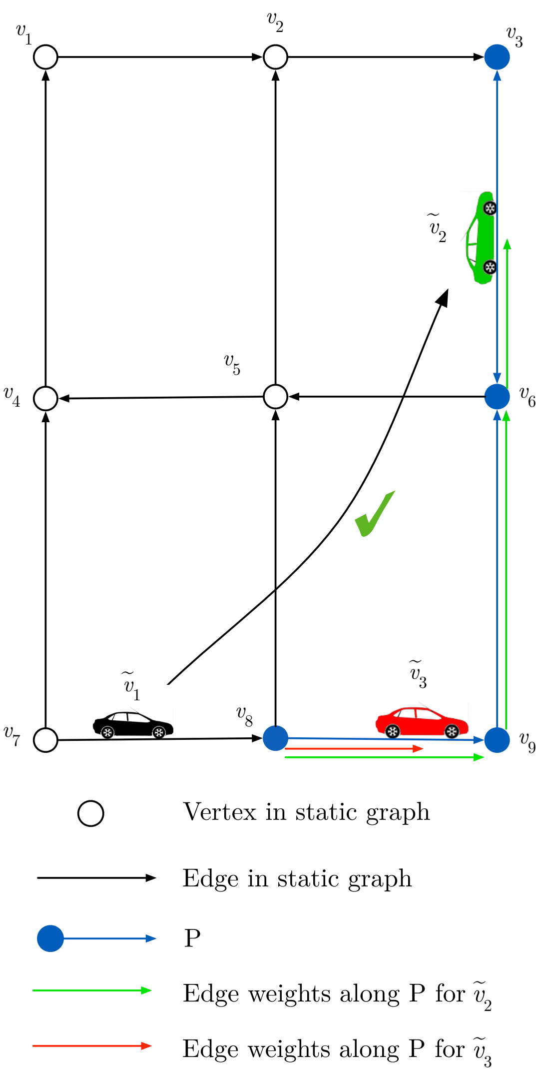


Figure 5.5.: **Left**: Heuristic 3: Edge weights along P. **Right**: Visualization of heuristic 3. $\tilde{v}_2$ is preferred since the accumulated weight of edges along P (blue) is higher than for $\tilde{v}_3$.

terms, the algorithm's goal is to start at $\tilde{v}_{start}$, discover a $\tilde{v}_{goal}$ fulfilling condition i and finding the shortest path back to v_{start}. The path to $\tilde{v}_{start}$ may contain already-visited vertices during discovery of $\tilde{v}_{goal}$; hence the cycle is formally a *closed walk*. $\tilde{v}_{goal}$ is discovered without a priori knowledge within the dynamic graph. Information about successful queries is stored in a distributed manner within the dynamic graph. Therefore, $\tilde{v} \in \tilde{V}$ carry additional *crumbs* which refer to other $\tilde{v} \in V$ that are elements of a *route* to $\tilde{v}_{goal}$. Metaphorically speaking, the size of the crumbs increases with decreasing distance to the information providing node. Formally, the values of the crumbs are getting higher with fewer vertices remaining on the route to $\tilde{v}_{goal}$. Both notations are used in the rest of this chapter. Crumbs are stored in each vertex after an information discovery succeeded. The term has its origin in the popular fairytale Hänsel und Gretel who leave breadcrumbs behind as they are hiking deeper into the forest. Afterwards, they collect the breadcrumbs in order to find their way out [Grimm and Grimm, 2007].

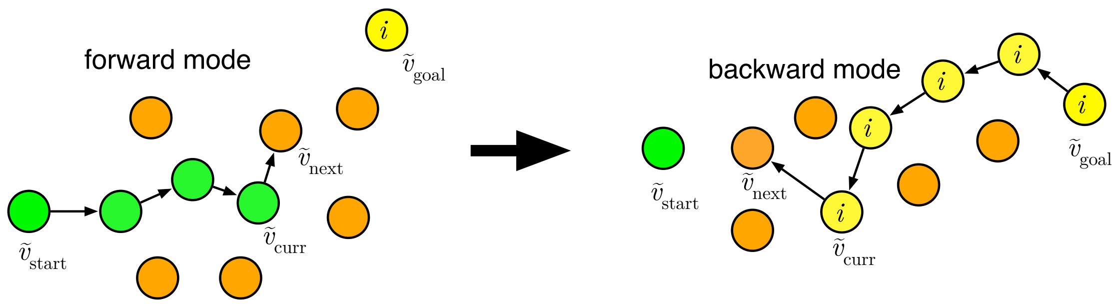


Figure 5.6.: **Left**: forward mode and visited vertices to $\tilde{v}_{goal}$ that fulfills condition i; **right**: backward mode, in which yellow colored vertices are now part of the route and can also fulfill i, if indicated by results of the benefit function. Note the topology change of $\tilde{G}$; $\tilde{f}$ and $\tilde{g}$ are used to approximate a path from $\tilde{v}_{goal}$ to $\tilde{v}_{start}$.

5.4.1. Algorithm Overview

Similar to LGHS, LGHS-CD is executed in vertices consecutively. The algorithm is divided into three phases: forward mode, backward mode and route maintenance. An overview is given in Figure 5.6. *Forward mode* is started in $\tilde{v}_{start}$. If available, crumbs are considered for determining $\tilde{v}_{next}$; otherwise, an application-specific approximation is used. Furthermore, information from $\tilde{f}$ and $\tilde{g}$ on previously visited $\tilde{v}$ is forwarded from the current vertex v_{curr} to $\tilde{v}_{next}$. This is exploited in *backward mode* when $\tilde{v}_{goal}$ has been discovered and a path back to $\tilde{v}_{start}$ has to be found. This enables LGHS-CD to approximate the sub-graph between $\tilde{v}_{goal}$ and $\tilde{v}_{start}$. In backward mode crumbs are stored in each visited vertex to optimize future cycle discoveries. This generates a route to vertices fulfilling i. The more such discoveries succeed, the higher the probability LGHS-CD can rely on already-existing crumbs. Shorter paths between visited vertices in backward mode and $\tilde{v}_{goal}$ result in higher values for the crumbs. The goal condition i

is connected to data d that is relevant for $\tilde{v}_{start}$. Therefore, in backward mode a benefit function calculates the local relevance of $i.d$ for $\tilde{v}_{curr}$. Upon exceeding a predefined threshold, $i.d$ is stored in the vertex. The same benefit function is used to calculate a global relevance, e.g. importance for other vertices. If relevant, $i.d$ is distributed to all $\tilde{v} \in \tilde{N}$. This ensures dissemination of $i.d$ in the dynamic graph. A time-to-live is applied in order to remove expired data.
Route maintenance is a twofold process: first, crumb size is decreased over time. Second, route maintenance handles dynamic graph operations, e.g. vertices that enter / leave the route are added / removed. Also topology changes of the route have to be considered due to the high dynamics of the graph.

5.4.2. Phase 1: Forward Mode

Forward mode is divided into three steps which in turn consist of two sub-steps each (see Table 5.3). Step 1 checks for the goal condition and determines $\tilde{v}_{next}$. Step 2 is a fallback in case no crumbs exist in $\tilde{v}_{curr}$ or $\tilde{v} \in \tilde{N}$. The algorithm then relies on LGHS for determining $\tilde{v}_{next}$. Step 3 stores the local context of $\tilde{v}_{curr}$ to M.

When the algorithm is started in a $\tilde{v} \in \tilde{V}$ it generates local memory M that contains essential data for completion of cycles in unknown highly dynamic graphs. Since a $\tilde{v} \in \tilde{V}$ has only restricted knowledge about graph topology, M cannot be accessed globally and is therefore forwarded to $\tilde{v}_{next}$ after the completion of forward mode. M also stores the *local context* of every vertex visited in forward mode during step 3. The Contents of M are then used in backward mode in order to approximate the sub-graph between $\tilde{v}_{goal}$ and $\tilde{v}_{start}$ in order to assess the shortest path back to $\tilde{v}_{start}$.

Step 1: Determination of $\tilde{v}_{next}$ Using Crumbs

Step 1a starts with a check for the termination condition. If the current vertex fulfills i and holds a non-expired version $i.d$, either $\tilde{v}_{goal}$ was discovered, or a vertex which stored $i.d$ during backward mode. LGHS-CD then terminates forward mode and continues with backward mode (see Section 5.4.3). In step 1b, $\tilde{v}_{next}$ is determined. Therefore, LGHS-CD looks for available crumbs in $\tilde{v}_{curr}$ and all $\tilde{v} \in \tilde{N}$. In case crumbs exist, the algorithm jumps immediately to step 3 and continues in the vertex containing the largest crumb after completing step 3. With no crumbs, restricted knowledge on graph topology and dynamic operations, it is not possible to determine a path to $\tilde{v}_{goal}$. Step 2 therefore provides a fallback solution in case no crumbs exist. In this case, $\tilde{v}_{goal}$ is approximated using *application-specific* data. It is then the application's responsibility to set $\tilde{v}_{goal}$. This approximation is highly dependent on the application domain. In this thesis, LGHS-CD is applied to C2X networks. Here, the most probable geographic location of the destination is used. For example, the on-demand traffic information system described in Section 6.1.1 assesses the location of a traffic event depending on its type: a traffic jam, for example, is most likely in front of the car. In such a case $\tilde{v}_{next}$ is determined as a car driving in the same direction of the EGO and located ahead of it.

Table 5.3.: The three steps of forward mode.

Forward mode		
Step	Sub-step	Action(s)
Step 1	1a	Check whether the current vertex is the $\tilde{v}_{goal}$ that fulfills i. If so, proceed with backward mode.
	1b	Check whether crumbs exist in $\tilde{v} \in \tilde{N}$. If so, continue in $\tilde{v}$ with the largest crumb after completing step 3. If no crumbs exist, approximate destination and proceed with step 2.
Step 2	2a	Use LGHS to solve DSSSP to $\tilde{v}_{goal}$.
	2b	Best neighbor decision according to result of LGHS. If no best neighbor selection possible, wait for next update of $\tilde{f}$ to $\tilde{N}$.
Step 3	3a	Store $\Delta d = \tilde{f}(\tilde{v}_{curr}, t-1) - \tilde{f}(\tilde{v}_{curr}, t-2)$ and mapping list $m = \tilde{g}(\tilde{v}_{curr})$ in $M.\tilde{v}_{curr}$.
	3b	Copy M to $\tilde{v}_{next}$ and continue forward mode in $\tilde{v}_{next}$.

Step 2: Determination of $\tilde{v}_{next}$ using LGHS

As described above, step 2 provides a fallback solution in case no crumbs are available in $\tilde{v} \in \tilde{N}$ or $\tilde{v}_{curr}$. Consequently, $\tilde{v}_{next}$ could not be determined in step 1. This determination is done in step 2.

The result of step 1 is that either an approximated $\tilde{v}_{goal}$ or a route has been identified by considering crumbs. Step 2 considers this result and uses the original LGHS algorithm in order to determine $\tilde{v}_{next}$ out the neighboring vertices. Here, the three heuristics introduced in Section 5.3.3 are computed. According to the result, $\tilde{v}_{next}$ is selected. In case no $\tilde{v} \in \tilde{N}$ is suitable, step 2 also uses the same fallback as LGHS by waiting for the next update of $\tilde{f}$ to $\tilde{N}$.

Step 3: Update Local Memory M

In one of the two previous steps $\tilde{v}_{next}$ was determined. In step 3 important information about the local context of $\tilde{v}_{curr}$ is stored to M. Function $\tilde{f}$ is queried over the last two time intervals and the result is stored as Δd. The idea is to retrieve weight changes of $\tilde{v}_{curr}$ in correlation to edges in the static graph $\tilde{G}$ over time. Note that in order to get more accurate values, this process can be repeated over the last n updates of $\tilde{f}$. Using the definitions introduced in Section 5.2 it is likely that the assigned edge $e \in E$ of a $\tilde{v}$ changed between to updates of $\tilde{f}$. This has to be taken into account when calculating Δd. Formally, this leads to the following:

$$\Delta d = \begin{cases} |\tilde{f}(\tilde{v}_{curr}, t-1).w - \tilde{f}(\tilde{v}_{curr}, t-2).w| & \tilde{f}(\tilde{v}_{curr}, t-1).e = \tilde{f}(\tilde{v}_{curr}, t-2).e \\ \tilde{f}(\tilde{v}_{curr}, t-2).e.w - \tilde{f}(\tilde{v}_{curr}, t-2).w + \tilde{f}(\tilde{v}_{curr}, t-1).w & \tilde{f}(\tilde{v}_{curr}, t-1).e \neq \tilde{f}(\tilde{v}_{curr}, t-2).e \end{cases}$$

Furthermore, mapping list $m = \tilde{g}(\tilde{v}_{curr})$ is stored in M and copied to $\tilde{v}_{next}$. This data is then used in backward mode to discover a stable path back to $\tilde{v}_{start}$ in $\tilde{G}$. After completion of phase 3, LGHS-CD continues with phase 1 of forward mode in $\tilde{v}_{next}$.

5.4.3. Phase 2: Backward Mode

Table 5.4.: The three steps of backward mode.

Backward mode		
Step	Sub-step	Action(s)
Step 1	1a	Check whether $\tilde{v}_{start} \in \tilde{N}$. If so, $\tilde{v}_{next} = \tilde{v}_{start}$ and continue with step 3. Otherwise continue to step 1b.
	1b	Use data stored in M to approximate subgraph between $\tilde{v}_{curr}$ and $\tilde{v}_{start}$.
Step 2	2a	Use LGHS to solve DSSSP, update neighbor weights and compute heuristics.
	2b	Best neighbor decision according to result of LGHS. If no best neighbor selection is possible, wait for next update of $\tilde{f}$ to $\tilde{N}$.
Step 3	3a	Create or update crumbs for $i.d$ in $\tilde{v}_{curr}$.
	3b	Compute local and global benefit and disseminate/store $i.d$ accordingly.

Backward mode is started as soon as LGHS-CD discovers a $\tilde{v} \in \tilde{V}$ fulfilling the goal condition i. The task of the backward mode is to trace back to $\tilde{v}_{start}$ and deliver $i.d$. It can be safely assumed that the subgraph formed by $\tilde{v}_{goal}$, $\tilde{v}_{start}$, and the vertices visited during forward mode is constantly changing while backward mode is executed. This fact is addressed by using local context data collected during forward mode. LGHS-CD approximates the subgraph using all Δd values and mapping lists m. In order to speed up future requests to i, the Algorithm generates crumbs that form a route to $\tilde{v}_{goal}$. Furthermore, local and global benefits are calculated; hence $\tilde{v}_{curr}$ might not only become part of the route but also fulfill i by storing $i.d$. This supports future cycle discoveries and ensures dissemination of $i.d$ throughout the dynamic graph. A summary of the three steps of backward mode is given in Table 5.4.

Step 1: Subgraph Approximation

Similar to forward mode, step 1a checks for termination by searching for $\tilde{v}_{start}$ in all $\tilde{v} \in \tilde{N}$. LGHS-CD then sets $\tilde{v}_{next} = \tilde{v}_{start}$, executes step 3 and terminates. If $\tilde{v}_{start}$ is not an element of $\tilde{N}$ the Algorithm approximates the subgraph formed by $\tilde{v}_{curr}$, $\tilde{v}_{start}$ and all vertices contained in M that were visited during forward mode (step 1b). The pseudocode for this procedure is given in Figure 5.7. It uses information stored in M to calculate the most probable assignments of $\tilde{v} \in \tilde{V}$ to $e \in E$. Furthermore, weights of virtual edges $(\tilde{v}, v)$ are approximated (cf. Section 5.2). The Algorithm distinguishes two cases:

1. $\tilde{v} \in \tilde{N}$: In this situation the Algorithm can rely on more accurate information since $\tilde{f}$ can be queried (lines 17 – 20). Therefore, data stored in M is discarded.
2. $\tilde{v} \notin \tilde{N}$: Estimate correlation of $\tilde{v}$ to G (edge assignment and weight of virtual edge, cf. Section 5.2). Similar to Figure 5.2 of LGHS where future graph topology is calculated, the assigned edge of $\tilde{v}$ in static graph G needs to be determined according to estimated weight Δw_{est} (lines 5 – 13).

The results of subgraph approximation are then used for further processing. More precisely, information gathered during forward mode is used to approximate edge assignment of $\tilde{v}_{start}$ to $e \in E$. The original LGHS algorithm can then rely on this approximation and determine the shortest path back to $\tilde{v}_{start}$. LGHS-CD assumes that the dynamic of the graph is similar for neighboring vertices. For example, in the application domain this refers to cars close to each other on the same road traveling with comparable speed. This fact is exploited by averaging over all collected Δd during forward mode (line 1 of Figure 5.7). The estimated weight change Δw_{est} can then be calculated from Δd and the time elapsed since forward mode started (Δt, line 4).

Step 2: LGHS

Step 2a uses LGHS to solve DSSSP to $\tilde{v}_{start}$. While in forward mode LGHS was only a fallback solution, it is called in every run of backward mode. It works on estimated graph topology which resulted from step 1. Estimated values for $\tilde{v}_{start}$ are especially important for LGHS to succeed. Note, that $\tilde{v}_{start}$ is also contained in M, as it was added during forward mode. Hence, approximated correlations to G are available to LGHS in backward mode.
Step 2b selects $\tilde{v}_{next}$ according to results from LGHS. Also here, the algorithm waits for the next update of $\tilde{f}$ to vertices in $\tilde{N}$ if LGHS returns $\tilde{v}_{next} = \tilde{v}_{curr}$.

Step 3: Crumbs, Benefit Computation, and Dissemination

Step 3a of backward mode creates crumbs which support future queries. Additionally, local and global benefit is computed.
When entering step 3, the algorithm increases an already existing crumb value for $i.d$ by a global constant c_{inc}. If $\tilde{v}_{curr}$ contains no crumb for $i.d$, an initial value m_{init} is calculated according the following equation:

$$m_{init} = max(0, c_{def} \cdot \rho(M.n_{bck}) - c_{dec} \cdot c) \tag{5.1}$$

$\Delta w_{avg} \leftarrow \frac{1}{|M|} \cdot \sum_{i=1}^{|M|} M.\tilde{v}_i.\Delta d$
for all $(\tilde{v} \in M)$ **do**
  if $(\tilde{v} \notin \tilde{N})$ **then**
    $\Delta w_{est} \leftarrow \Delta w_{avg} \cdot \Delta t$
    if $(\Delta w_{est} > M.\tilde{v}.e.w)$ **then**
      $E \leftarrow M.\tilde{g}(\tilde{v})$
      for $(i \leftarrow 0; i < E.length - 1; i++)$ **do**
        if $(E[i] == \tilde{v}.e)$ **then**
          $\tilde{v}.e \leftarrow E[i+1]$
          $\tilde{v}.w \leftarrow \Delta w_{est} - E[i].w$
        end if
      end for
    else
      $\tilde{v}.e \leftarrow M.\tilde{v}.e$
      $\tilde{v}.w \leftarrow \Delta w_{est}$
    end if
  else
    $\tilde{v}.e \leftarrow \tilde{f}(\tilde{v}, t).e$
    $\tilde{v}.w \leftarrow \tilde{f}(\tilde{v}, t).w$
  end if
end for

Figure 5.7.: Subgraph approximation in backward mode

with

$$c = t_{curr} - M.t_{bck} \tag{5.2}$$

and

$$\rho : \mathbb{N}^+ \setminus \{1\} \to \mathbb{R}, x \longmapsto \begin{cases} 1 - \frac{x}{l^2} & x \leq l \\ 1 - \frac{x}{m_{max}} & x > l \end{cases} \tag{5.3}$$

The intuition behind equation 5.1 is to calculate an initial crumb value m_{init} by decreasing a default value c_{def} dependent on the number of vertices visited in backward mode so far ($M.n_{bck}$). This is reflected by function ρ: Up to a threshold l, ρ is close to 1, which in turn results in a constant value for m_{init}. Global parameter l declares the number of vertices that should carry crumb values close to c_{def}. When this threshold is exceeded, ρ converges to $-\infty$, resulting in $m_{init} = 0$. Here, m_{max} defines the maximum number of vertices carrying crumbs.

Term $c_{dec} \cdot c$ additionally subtracts a value c_{dec} c times where c denotes the time-steps elapsed since the backward mode started ($M.t_{bck}$). This information is contained in M and therefore locally available to the algorithm. By applying this term to equation 5.1, a lifetime for crumbs is added, after which they become invalid. This addresses the high dynamics of the graph.

1: $w_{path} \leftarrow Path(\tilde{v}_{curr}.e.v_0, \tilde{v}_{goal}.e).w$
2: $b_{path} \leftarrow 0$
3: $b_{\tilde{g}} \leftarrow 0$
4: $b_{\tilde{v}} \leftarrow 0$
5: **if** $(w_{path} \leq w_{max})$ **then**
6: $\quad b_{path} \leftarrow \frac{w_{path}}{w_{max}} \cdot 100$
7: **end if**
8: **if** $(\tilde{v}_{goal}.e \in \tilde{g}(\tilde{v}_{curr}))$ **then**
9: $\quad b_{\tilde{g}} \leftarrow 100$
10: **end if**
11: $b_{\tilde{v}} \leftarrow \frac{b_{path}+b_{\tilde{g}}}{2}$

Figure 5.8.: Benefit Computation

In step 3b of backward mode a benefit value is calculated which is used for determining local storage of $i.d$ or dissemination to all $\tilde{v} \in \tilde{N}$. The concept of a *message benefit function* was originally introduced in [Adler et al., 2006; Eichler et al., 2006] and used for information dissemination in vehicular ad-hoc networks. This is adapted to LGHS-CD. [Eichler et al., 2006] consider message context (e.g. message age, time since last reception), vehicle context (e.g. driving direction), and information context (distance to source, time of day) to compute the benefit. While the message context can be mapped to the additional term $c_{dec} \cdot c$ in equation 5.1, vehicle and information context can be seen as b_{path} and $b_{\tilde{g}}$ as described in the following.

Figure 5.8 shows the exact procedure by which benefit is computed. The basis for benefit of a vertex $b_{\tilde{v}}$ is the two values b_{path} and $b_{\tilde{g}}$ which are determined independently from each other. The average value then results in $b_{\tilde{v}}$. They are computed as follows:

- b_{path}: This component of the overall benefit $b_{\tilde{v}}$ gives an approximation of the "distance" between $\tilde{v}_{curr}$ and $\tilde{v}_{goal}$. The "closer" $\tilde{v}_{curr}$ is to $\tilde{v}_{goal}$, the higher the benefit. Formally, this is achieved by considering w_{path} as the weight of shortest path on static graph G between current vertex $\tilde{v}_{curr}$ and goal vertex $\tilde{v}_{goal}$ (line 1). Recall that $\tilde{v}_{goal}$ met goal condition i and carries data $i.d$. Along with $i.d$, a global maximum weight w_{max} is defined for which $i.d$ is valid. In case w_{path} is lower or equal to w_{max}, benefit b_{path} is computed (lines 5 – 7). $\tilde{b}_{path}$ gives a result within the interval of [0,100].

- $\tilde{b}_{\tilde{g}}$: The second component of $b_{\tilde{v}}$ considers future edge assignments of $\tilde{v}_{curr}$ to edges in static graph G. In cases where a (future) edge assignment matches the edge $\tilde{v}_{goal}$ is assigned to, $i.d$ is considered relevant for $\tilde{v}_{curr}$. Formally, this is achieved by exploiting function $\tilde{g}$. $b_{\tilde{g}}$ then is either 0 or 100 (lines 8 – 10).

The overall benefit $b_{\tilde{v}}$ is then defined as the average value of b_{path} and $b_{\tilde{g}}$ (line 11).

5.4.4. Phase 3: Route Maintenance

Decreasing Crumb Sizes

During visiting vertices in backward mode, already existing routes are strengthened or initialized according to a rather complex operation presented in equation 5.1. The counterpart, decreasing crumb sizes over time, also needs to be considered. However, it can be achieved with less effort: LGHS-CD starts decreasing the size of crumbs by a constant value after a pre-defined time. This ensures fading of routes when no requests in backward node reach a vertex. Also, start time of decrease and amount can be adapted depending on the application / information type.

Handling Dynamic Graph Operations

Route maintenance is an important part of LGHS-CD since routes leading to a $\tilde{v}_{goal}$ fulfilling goal condition i are constantly changing due to graph dynamics. Phase 3 is therefore not consecutively executed but is triggered by events on the dynamic graph. More precisely, phase 3 is activated when

- a non-member vertex enters the route,
- a member vertex leaves the route, and
- the sequence of member vertices changes along the route.

members in this context identify all vertices carrying crumbs to a vertex fulfilling goal condition i.

Non-member vertex enters the route Given a vertex $\tilde{v}_{new}$ and a subset $\tilde{V}_T \subseteq \tilde{V}$ where $\tilde{V}_T$ contains all vertices forming the route, $\tilde{v}_{new}$ enters a route if and only if:

$$\exists \tilde{a}, \tilde{b} \in \tilde{V}_T, \exists \tilde{v}_{new} \in \tilde{N} : \tilde{f}(\tilde{v}_{new}, t).e.v_0 \in P(\tilde{f}(\tilde{a}, t).e.v_0, \tilde{f}(\tilde{b}, t).e.v_0) \wedge \tilde{v}_{new}.i \neq \emptyset$$

with $P(v_1, v_2)$ shortest path between vertices $v_1, v_2 \in G$. If this condition is fulfilled, a crumb has to be generated for $\tilde{v}_{new}$ by one of its neighboring vertices. However, the event of a new member entering the route is detected by all neighboring vertices. In order to prevent surrounding vertices from generating crumbs simultaneously, a synchronization procedure is applied. This addresses possible limitations of the application domain. For example, in vehicular ad-hoc networks wireless communication bandwidth is considered a limited resource. Therefore, communication overhead is reduced by limiting the nodes transmitting to the new member of the route. Besides this synchronization step, a value for the crumb has to be calculated.

The entire procedure is described in Figure 5.9, left. For all elements in $\tilde{N}$ the weight to $\tilde{v}_{new}$ is calculated using function $\tilde{w}$ (line 5). If it is smaller than the weight between

1: $c_{sum} \leftarrow 0$
2: $cnt \leftarrow 0$
3: **for all** $(\tilde{n} \in \tilde{N} \setminus \{\tilde{v}_{new}\})$ **do**
4:     **if** $(crumb(i, \tilde{n}, t) \neq \emptyset)$ **then**
5:         **if** $(w(\tilde{n}, \tilde{v}_{new}) < w(\tilde{v}_{curr}, \tilde{v}_{new}))$ **then**
6:             *abort*
7:         **end if**
8:         $c_{sum} \leftarrow c_{sum} + getCrumb(i, \tilde{n}, t)$
9:         $cnt \leftarrow cnt + 1$
10:     **end if**
11: **end for**
12: $c_{avg} \leftarrow \frac{c_{sum}}{cnt}$
13: $setCrumb(i, \tilde{v}_{new}, c_{avg})$

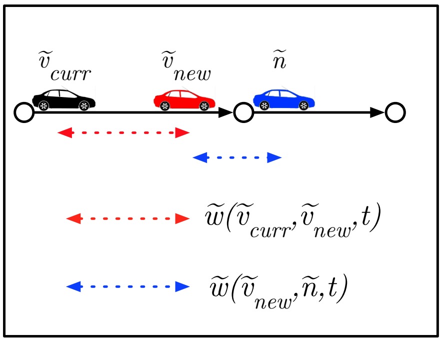


Figure 5.9.: **Left**: the procedure for incorporating a new route member. **Right**: Visualization of the synchronization step which tries to minimize the vertices that set a crumb in the new vertex of the route. In the optimal case, this is done only by the "closest" vertex.

the current vertex $\tilde{v}_{curr}$ and $\tilde{v}_{new}$, it is expected that another vertex sets the appropriate crumb and the procedure is aborted Figure 5.9, line 6. Otherwise the crumb value of the neighboring vertex is retrieved (line 8). In lines 12 − 13 the average crumb value in $\tilde{v}_{new}$ is set. This avoids situations where all neighboring vertices create crumbs in a $\tilde{v}_{new}$. However, the procedure cannot prevent this completely because every vertex has only local knowledge of the dynamic graph topology which is limited to the elements in $\tilde{N}$. Formally, two vertices will both create crumbs in a $\tilde{v}_{new}$ if the following condition is fulfilled:

$$\exists \tilde{v}_1, \tilde{v}_2, \tilde{v}_{new} \in \tilde{V} : \tilde{v}_{new} \in \tilde{v}_1.\tilde{N} \wedge \tilde{v}_{new} \in \tilde{v}_2.\tilde{N} \wedge \tilde{v}_1 \notin \tilde{v}_2.\tilde{N} \wedge \tilde{v}_2 \notin \tilde{v}_1.\tilde{N}$$

Member vertex leaves the route If a member vertex is *leaving* the route it removes its crumb autonomously since this situation can be detected locally. A vertex continuously checks $\tilde{N}$ for crumbs belonging to the same route after each update of $\tilde{f}$. As soon as there is no $\tilde{v} \in \tilde{N}$ that contains a crumb, the current vertex considers itself not a member of the route anymore and deletes its crumbs. Supposing that the vertex reenters the route, it will be notified by already-existing members according to the procedure shown in Figure 5.9.

Positions of member vertices change along the route Since "larger" crumbs are required for vertices "closer" to $\tilde{v}_{goal}$, they have to be continuously adapted to topology changes of the dynamic graph. With updates of $\tilde{f}$ to vertices in $\tilde{N}$, the sequence of route members is likely to change. This results in incorrect crumb sizes that do not reflect the actual order; hence a third route maintenance procedure is applied. Formally, such a situation can be described as follows (where $\tilde{v}_x.c$ denotes crumbs sizes stored in $\tilde{v}_x$):

$$\exists \tilde{v}_1, \tilde{v}_2, \tilde{v}_{goal} : \tilde{v}_1.c > \tilde{v}_2.c \wedge P(\tilde{v}_1, \tilde{v}_{goal}).w > P(\tilde{v}_2, \tilde{v}_{goal}).w$$

Both affected vertices can detect this situation locally by querying $\tilde{N}$. In this case both replace their own crumb value with that of the conflicting vertex in $\tilde{N}$ in order to reflect the change in sequence. $\tilde{f}$ then updates the changed crumb values to $\tilde{N}$ in all neighboring vertices that are possibly also part of the route.

5.5. Example Runs

In order to give the reader a better understanding of how the algorithms work in practice, an example using a concrete situation and precise numbers is given in this section. A simple grid for the static graph is used and an edge weight of 1 is assumed for every edge. The shortest path P is depicted in blue color. The weight which is returned by $\tilde{f}$ is one out of $\frac{1}{2}, \frac{1}{3}, \frac{2}{3}$ (see Figure 5.10).

5.5.1. LGHS

For LGHS the focus is on heuristic computation and therefore estimation of future graph topology is skipped (part of Phase 1) for clarity. Except for the destination only the vertices of the dynamic graph which are in $\tilde{N}$ are shown. The tables in each step give the results of $\tilde{f}$ and $\tilde{g}$ as well as the heuristic computation. The vertex on which the algorithm is currently running is marked by a "$*$" in the **first** column, the next vertex selected for $\tilde{P}$ by a "$*$" in the **last** column. For the calculation of edge weights using function w, weights of the static graph are used (see Figure 5.10). Every heuristic is weighted equally ($\frac{1}{3}$). ω and τ for heuristic 2 are set to 2 and 1 respectively.

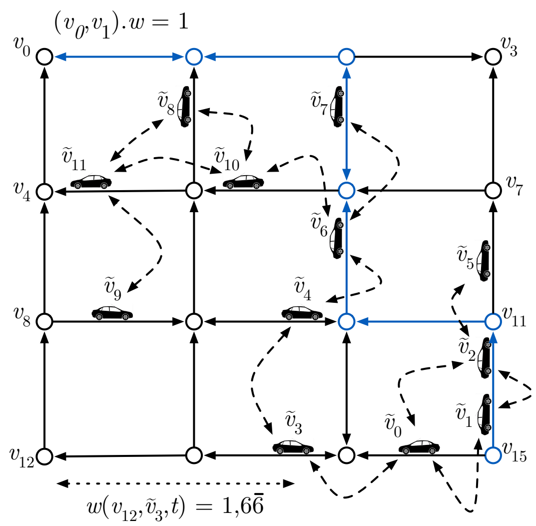
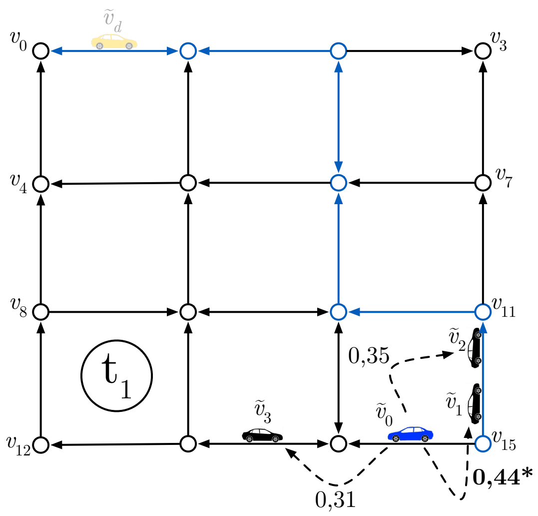


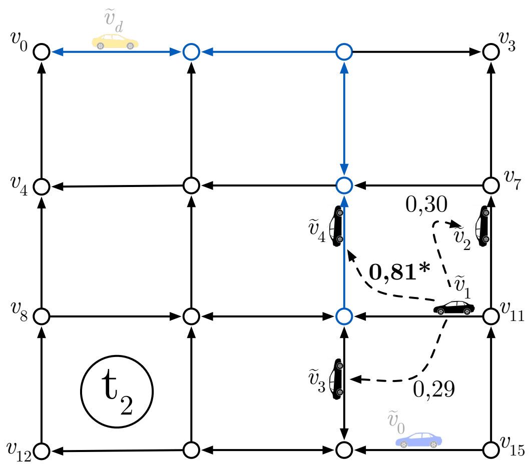
Figure 5.10.: Properties of the static and the dynamic graph used for the example runs.



$P = [(v_{15}, v_{11}), (v_{11}, v_{10}), (v_{10}, v_6), (v_6, v_2), (v_2, v_1), (v_1, v_0)]$
$\tilde{P} = [(\tilde{v}_0, \tilde{v}_1)]$

Comments: Normally, Dijkstra's algorithm would compute shortest path P on G as $[(v_{14}, v_{10}), (v_{10}, v_6), ...]$ but since the number of $\tilde{v}$ assigned to (v_{15}, v_{11}) is higher than those for (v_{14}, v_{10}), the extension to Dijkstra's algorithm adapts weights for shortest path computation in G dynamically. In this case, (v_{15}, v_{11}) is preferred over (v_{14}, v_{10}), which results in shortest path P containing edge (v_{15}, v_{11}). $\tilde{v}_1$ then scores higher than $\tilde{v}_2$ because $\tilde{g}(v_1, t_1)$ matches more edges of P than $\tilde{g}(v_2, t_1)$ (heuristic 2).

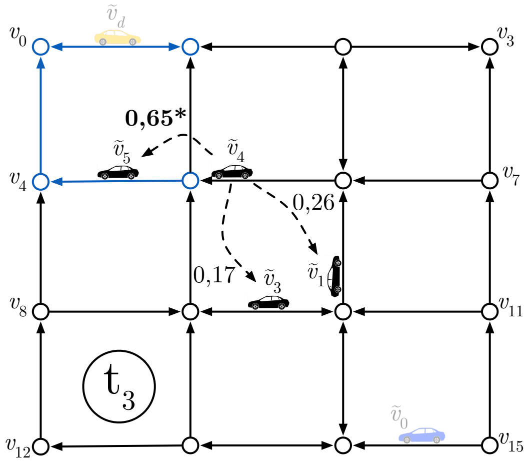
$\tilde{N}$	$\tilde{f}(\tilde{v}_x, t_1)$	$\tilde{g}(\tilde{v}_x)$	H_1	H_2	H_3	$\sum_{i=1}^{3} H_i$
$\tilde{v}_0^*$	$\{(v_{15}, v_{14}), \frac{1}{2}\}$	$[(v_{14}, v_{13}), (v_{13}, v_{12})]$	–	–	–	–
$\tilde{v}_1$	$\{(v_{15}, v_{11}), \frac{1}{3}\}$	$[(v_{11}, v_{10}), (v_{10}, v_6)]$	$\frac{1}{3}(1 - \frac{0{,}\overline{3}}{6}) \approx 0,31$	$\frac{1}{3} \cdot \frac{4}{12} \approx 0,11$	$\frac{1}{3} \cdot \frac{0{,}\overline{3}}{6} \approx 0,02$	**0,44***
$\tilde{v}_2$	$\{(v_{15}, v_{11}), \frac{2}{3}\}$	$[(v_{11}, v_7), (v_7, v_3)]$	$\frac{1}{3}(1 - \frac{0{,}\overline{3}}{6}) \approx 0,31$	$\frac{1}{3} \cdot \frac{0}{12} = 0$	$\frac{1}{3} \cdot \frac{0{,}\overline{6}}{6} \approx 0,04$	0,35
$\tilde{v}_3$	$\{(v_{13}, v_{14}), \frac{1}{2}\}$	$[(v_{14}, v_{10}), (v_{10}, v_6)]$	$\frac{1}{3}(1 - \frac{1{,}5}{6}) \approx 0,25$	$\frac{1}{3} \cdot \frac{2}{12} \approx 0,06$	$\frac{1}{3} \cdot \frac{0}{6} = 0$	0,31



$P = [(v_{10}, v_6), (v_6, v_2), (v_2, v_1), (v_1, v_0)]$
$\tilde{P} = [(\tilde{v}_0, \tilde{v}_1), (\tilde{v}_1, \tilde{v}_4)]$

Comments: At t_2, $\tilde{v}_1$ is assigned to (v_{11}, v_{10}). Three other vertices of $\tilde{G}$ are in $\tilde{N}$. Since edges of $\tilde{g}$ for all three $\tilde{v}$'s do not match an edge of P, heuristic 2 is 0. Heuristic 1 and heuristic 3 decide that $\tilde{v}_4$ is the next vertex in $\tilde{P}$ since it is "closest" to a vertex of P and currently is the only $\tilde{v}$ assigned to an edge of P.

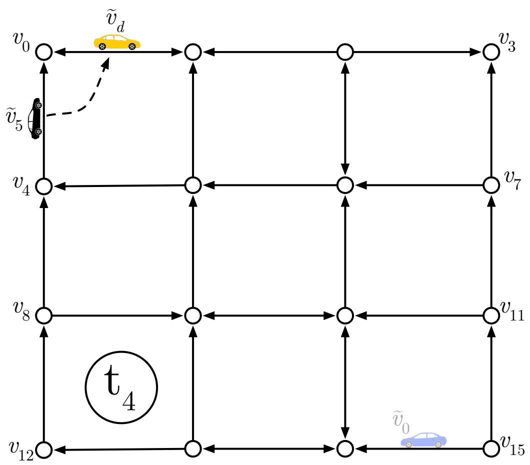
$\tilde{N}$	$\tilde{f}(\tilde{v}_x, t_2)$	$\tilde{g}(\tilde{v}_x)$	H_1	H_2	H_3	$\sum_{i=1}^{3} H_i$
$\tilde{v}_1^*$	$\{(v_{11}, v_{10}), \frac{1}{3}\}$	$[(v_{10}, v_6), (v_6, v_5)]$	–	–	–	–
$\tilde{v}_2$	$\{(v_{11}, v_7), \frac{2}{3}\}$	$[(v_7, v_3), (v_3, v_2)]$	$\frac{1}{3}(1 - \frac{1{,}\overline{3}}{4}) \approx 0,22$	$\frac{1}{3} \cdot \frac{0}{8} = 0$	$\frac{1}{3} \cdot \frac{1}{4} \approx 0,08$	0,30
$\tilde{v}_3$	$\{(v_{14}, v_{10}), \frac{1}{2}\}$	$[(v_{10}, v_9), (v_9, v_5)]$	$\frac{1}{3}(1 - \frac{0{,}5}{4}) \approx 0,29$	$\frac{1}{3} \cdot \frac{0}{8} = 0$	$\frac{1}{3} \cdot \frac{0}{4} = 0$	0,29
$\tilde{v}_4$	$\{(v_{10}, v_6), \frac{2}{3}\}$	$[(v_6, v_5), (v_5, v_1)]$	$\frac{1}{3}(1 - \frac{0{,}\overline{3}}{4}) \approx 0,31$	$\frac{1}{3} \cdot \frac{0}{8} = 0$	$\frac{1}{3} \cdot \frac{0{,}\overline{6}}{4} = 0,5$	**0,81***



$$P = [(v_5, v_4), (v_4, v_0), (v_0, v_1)]$$
$$\tilde{P} = [(\tilde{v}_0, \tilde{v}_1), (\tilde{v}_1, \tilde{v}_4), (\tilde{v}_4, \tilde{v}_5)]$$

Comments: At time t_3, $\tilde{v}_4$ is assigned to (v_6, v_5). Note that initial shortest path P was changed due to the absence of assigned vertices to edge (v_5, v_1). P is now calculated over (v_5, v_4). The score of $\tilde{v}_5$ is significantly higher than for $\tilde{v}_1$ and $\tilde{v}_3$. In this case, all three heuristics score for $\tilde{v}_5$: The vertex is close to a vertex of P and $\tilde{g}$ matches the future path of $\tilde{v}_5$.

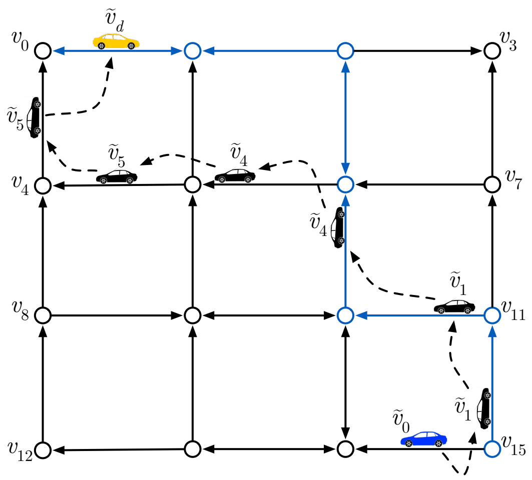
$\tilde{N}$	$\tilde{f}(\tilde{v}_x, t_3)$	$\tilde{g}(\tilde{v}_x)$	H_1	H_2	H_3	$\sum_{i=1}^{3} H_i$
$\tilde{v}_1$	$\{(v_{10}, v_6), \frac{1}{3}\}$	$[(v_6, v_5), (v_5, v_4)]$	$\frac{1}{3}(1 - \frac{1,\overline{6}}{3}) = 0,15$	$\frac{1}{3} \cdot \frac{2}{6} \approx 0,11$	$\frac{1}{3} \cdot \frac{0}{3} = 0$	$0,26$
$\tilde{v}_3$	$\{(v_{10}, v_9), \frac{1}{2}\}$	$[(v_9, v_5), (v_5, v_1)]$	$\frac{1}{3}(1 - \frac{1,5}{3}) \approx 0,17$	$\frac{1}{3} \cdot \frac{0}{6} = 0$	$\frac{1}{3} \cdot \frac{0}{3} = 0$	$0,17$
$\tilde{v}_4^*$	$\{(v_6, v_5), \frac{2}{3}\}$	$[(v_5, v_1), (v_1, v_0)]$	–	–	–	–
$\tilde{v}_5$	$\{(v_5, v_4), \frac{1}{2}\}$	$[(v_4, v_0), (v_0, v_1)]$	$\frac{1}{3}(1 - \frac{0,5}{3}) \approx 0,28$	$\frac{1}{3} \cdot \frac{4}{6} \approx 0,22$	$\frac{1}{3} \cdot \frac{0,5}{3} \approx 0,06$	$\mathbf{0,56}^*$



$$P = []$$
$$\tilde{P} = [(\tilde{v}_0, \tilde{v}_1), (\tilde{v}_1, \tilde{v}_4), (\tilde{v}_4, \tilde{v}_5), (\tilde{v}_5, \tilde{v}_d)]$$

Comments: At t_4 the goal condition is fulfilled, $\tilde{v}_d$ is an element of $\tilde{N}$. The algorithm terminates.

$\tilde{N}$	$\tilde{f}(\tilde{v}_x, t_4)$	$\tilde{g}(\tilde{v}_x)$	H_1	H_2	H_3	$\sum_{i=1}^{3} H_i$
$\tilde{v}_4$	$\{(v_6, v_5), \frac{2}{3}\}$	$[(v_6, v_7), (v_7, v_{11})]$	–	–	–	–
$\tilde{v}_d^*$	$\{(v_0, v_1), \frac{1}{2}\}$	$[(v_1, v_2), (v_2, v_3)]$	–	–	–	–



$$
\begin{aligned}
P = \quad & [(v_{15}, v_{11}), (v_{11}, v_{10}), (v_{10}, v_6), \\
& (v_6, v_2), (v_2, v_1), (v_1, v_0)] \\
\tilde{P} = \quad & [(\tilde{v}_0, \tilde{v}_1), (\tilde{v}_1, \tilde{v}_4), (\tilde{v}_4, \tilde{v}_5), (\tilde{v}_5, \tilde{v}_d)]
\end{aligned}
$$

Comments: The initially calculated path P and $\tilde{P}$ in comparison. The picture shows the flexibility of the algorithm. It dynamically adapts shortest path P when it cannot be fully approximated in $\tilde{G}$.

5.5.2. LGHS-CD

The next example run shows how forward and backward mode of LGHS-CD are executed. This section should give the reader a practical example of how routes and values for crumbs are computed. LGHS heuristic values are attached to each arrow but details on computations are skipped for simplicity. The next vertex selected by LGHS is marked by a "*". The same simple grid for the static graph is used as in the previous section and a weight of 1 for every edge is assumed. The weight which is returned by $\tilde{f}$ is out of $\frac{1}{2}, \frac{1}{3}, \frac{2}{3}, 0$. Except for the destination vertex $\tilde{v}_{goal}$, only vertices contained in $\tilde{N}$ are shown. The remaining topology of the dynamic graph is unknown. The contents of local memory M as well as results of each step of the forward/backward mode are shown in the respective tables. The following default values are used: initial crumb value ($c_{def} = 100$), crumb value decrease after each step $c_{dec} = 1$, maximum weight for benefit computation ($w_{max} = 5$), threshold for dissemination: 50, threshold for local storage: 50. Furthermore, decrease of crumb values is started after three steps in backward mode ($l = 3$). The subgraph approximation is reduced to $\tilde{v}_{start}$ for simplicity.

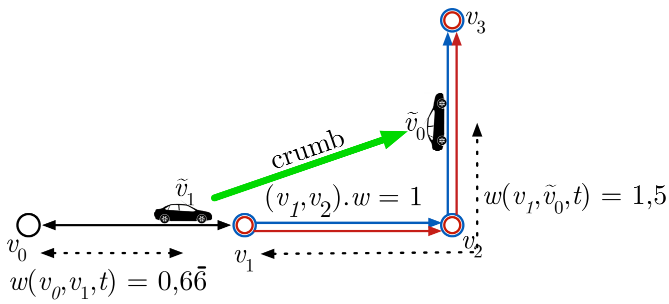
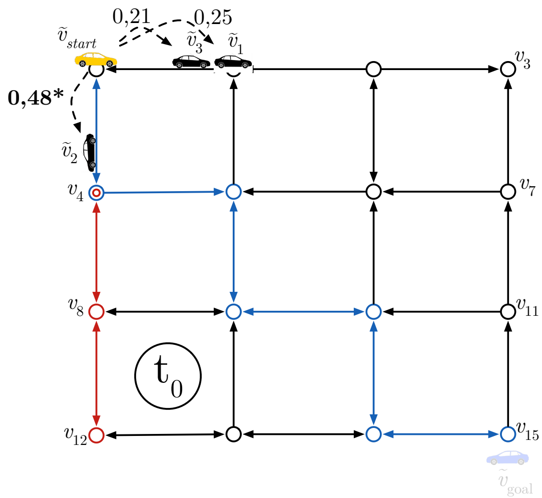


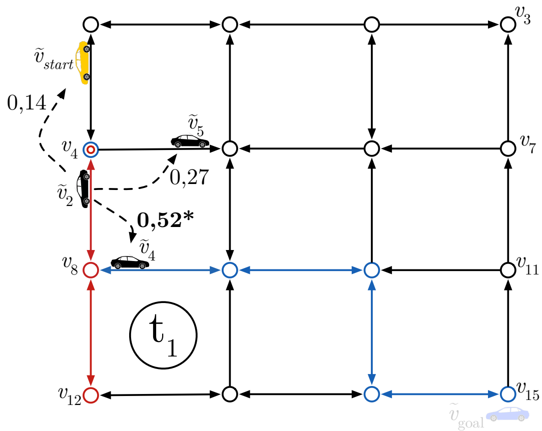
Figure 5.11.: Properties of the static and the dynamic graph used for the example run of LGHS-CD. Green arrows: crumbs in backward mode, red arrows: visualization of $\tilde{g}(v_x)$, blue arrows: shortest path on static graph.



Forward mode execution in $\tilde{v}_{start}$	
step	action(s)
1a	$\tilde{v}_{start}$ does not fulfill i
1b	No crumbs. $\rightarrow$ step 2a
2a	$v_{goal} \approx \{(v_{15}, v_{14}), 0\}$
2b	Set $\tilde{v}_{next} = \tilde{v}_2$
3a	Update M
3b	Continue in $\tilde{v}_2$

M			
$\tilde{v}_x$	$\tilde{f}(v_x, t_0)$	Δd	$\tilde{g}(\tilde{v}_x)$
$\tilde{v}_{start}$	$\{(v_0, v_4), 0\}$	$\frac{1}{3}$	$\{(\tilde{v}_4, \tilde{v}_8), (\tilde{v}_8, \tilde{v}_{12})\}$

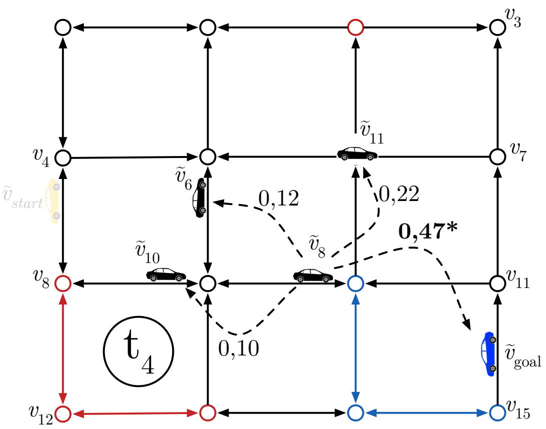
Comments: LGHS-CD is triggered in $\tilde{v}_{start}$. At this point, no crumbs exist and $\tilde{v}_{goal}$ is approximated by the application at $\{(v_{15}, v_{14}), 0\}$. LGHS determines $\tilde{v}_2$ as best neighbor. Δd for $\tilde{v}_0$ is calculated from $\tilde{f}(\tilde{v}_0, t-1) - \tilde{f}(\tilde{v}_0, t-2)$. In this case it is assumed $\tilde{v}_{start}$ was previously assigned to $\{(v_1, v_0), \frac{2}{3}\}$ and $\{(v_1, v_0), \frac{1}{3}\}$. Δd is therefore $\frac{1}{3}$.



Forward mode execution in $\tilde{v}_2$	
step	action(s)
1a	$\tilde{v}_2$ does not fulfill i
1b	No crumbs. $\rightarrow$ step 2a
2a	$v_{goal} \approx \{(v_{15}, v_{14}), 0\}$
2b	Set $\tilde{v}_{next} = \tilde{v}_4$
3a	Update M
3b	Continue in $\tilde{v}_4$

M			
$\tilde{v}_x$	$\tilde{f}(v_x, t_0)$	Δd	$\tilde{g}(\tilde{v}_x)$
$\tilde{v}_{start}$	$\{(v_0, v_4), \frac{1}{3}\}$	$\frac{1}{3}$	$\{(\tilde{v}_4, \tilde{v}_8), (\tilde{v}_8, \tilde{v}_{12})\}$
$\tilde{v}_4$	$\{(v_4, v_8), \frac{1}{3}\}$	$\frac{2}{3}$	$\{(\tilde{v}_4, \tilde{v}_8), (\tilde{v}_8, \tilde{v}_{12})\}$

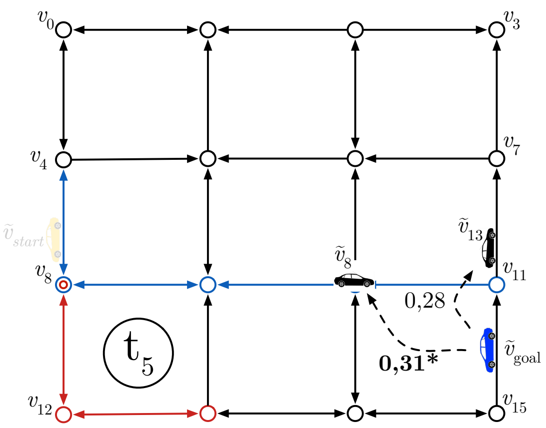
Comments: In t_1 the situation is similar to t_0. Δd is calculated as $\frac{2}{3}$ this time.



Forward mode execution in $\tilde{v}_8$	
step	action(s)
1a	$\tilde{v}_8$ does not fulfill i
1b	No crumbs. $\rightarrow$ step 2a
2a	$v_{goal} \approx \{(v_{15}, v_{11}), \frac{1}{2}\}$
2b	Set $\tilde{v}_{next} = \tilde{v}_{goal}$
3a	Update M
3b	Continue in $\tilde{v}_{goal}$

M			
$\tilde{v}_x$	$\tilde{f}(v_x, t_0)$	Δd	$\tilde{g}(\tilde{v}_x)$
$\tilde{v}_{start}$	$\{(v_4, v_8), \frac{1}{3}\}$	$\frac{1}{3}$	$\{(\tilde{v}_8, \tilde{v}_{12}), (\tilde{v}_{12}, \tilde{v}_{13})\}$
$\tilde{v}_4$	$\{(v_4, v_8), \frac{1}{3}\}$	$\frac{2}{3}$	$\{(\tilde{v}_4, \tilde{v}_8), (\tilde{v}_8, \tilde{v}_{12})\}$
$\tilde{v}_{10}$	$\{(v_8, v_9), \frac{2}{3}\}$	0	$\{(\tilde{v}_9, \tilde{v}_5), (\tilde{v}_9, \tilde{v}_1)\}$
$\tilde{v}_6$	$\{(v_9, v_5), \frac{1}{3}\}$	$\frac{1}{3}$	$\{(\tilde{v}_5, \tilde{v}_1), (\tilde{v}_1, \tilde{v}_0)\}$
$\tilde{v}_8$	$\{(v_9, v_{10}), \frac{2}{3}\}$	$\frac{1}{3}$	$\{(\tilde{v}_{10}, \tilde{v}_6), (\tilde{v}_6, \tilde{v}_2)\}$

Comments: Jump in time to t_4. LGHS-CD is now active in $\tilde{v}_8$. During the previous steps $\tilde{v}_{10}$ and $\tilde{v}_6$ were visited. M now contains information on these vertices.

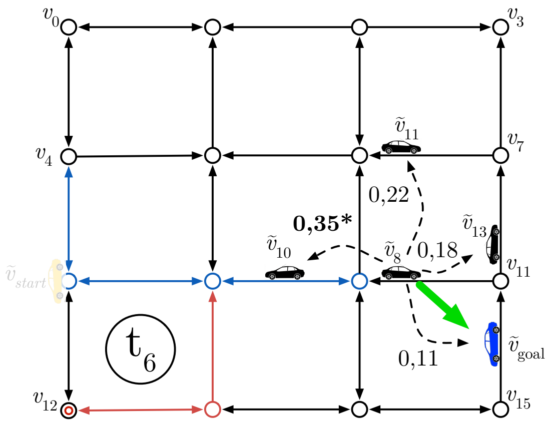


Backward mode execution in $\tilde{v}_{goal}$	
step	action(s)
1a	$\tilde{v}_{start} \notin \tilde{N}$
1b	$\tilde{v}_{start} \approx \{(v_4, v_8), \frac{2}{3}\}$
2a	$h(\tilde{v}_8) = \mathbf{0,31}^*, h(\tilde{v}_{13} = 0,28)$
2b	Set $\tilde{v}_{next} = \tilde{v}_8$
3a	No crumbs
3b	Disseminate to $\tilde{v}_8$, $\tilde{v}_{13}$

M			
$\tilde{v}_x$	$\tilde{f}(v_x, t_0)$	Δd	$\tilde{g}(\tilde{v}_x)$
$\tilde{v}_{start}$	$\{(v_4, v_8), \frac{2}{3}\}$	$\frac{1}{3}$	$\{(\tilde{v}_8, \tilde{v}_{12}), (\tilde{v}_{12}, \tilde{v}_{13})\}$
$\tilde{v}_4$	$\{(v_4, v_8), \frac{1}{3}\}$	$\frac{2}{3}$	$\{(\tilde{v}_4, \tilde{v}_8), (\tilde{v}_8, \tilde{v}_{12})\}$
$\tilde{v}_{10}$	$\{(v_8, v_9), \frac{2}{3}\}$	0	$\{(\tilde{v}_9, \tilde{v}_5), (\tilde{v}_9, \tilde{v}_1)\}$
$\tilde{v}_6$	$\{(v_9, v_5), \frac{1}{3}\}$	$\frac{1}{3}$	$\{(\tilde{v}_5, \tilde{v}_1), (\tilde{v}_1, \tilde{v}_0)\}$
$\tilde{v}_8$	$\{(v_9, v_{10}), \frac{2}{3}\}$	$\frac{1}{3}$	$\{(\tilde{v}_{10}, \tilde{v}_6), (\tilde{v}_6, \tilde{v}_2)\}$

Comments: $\tilde{v}_{goal}$ fulfills i. Backward mode is started. Approximation of edge assignment for $\tilde{v}_{start}$ is as follows:

- $\Delta w_{avg} = \frac{1}{5} \cdot (\frac{1}{3} + \frac{2}{3} + 0 + \frac{1}{3} + \frac{1}{3}) = \frac{1}{3}$
- $\Delta w_{est} = \Delta t \cdot \frac{1}{3} = (5-0) \cdot \frac{1}{3} = \frac{5}{3}$
- With $\tilde{g}(\tilde{v}_{start})$ contained in M and $\Delta w_{est} \Rightarrow \tilde{v}_{start} \approx \{(v_4, v_8), \frac{2}{3}\}$



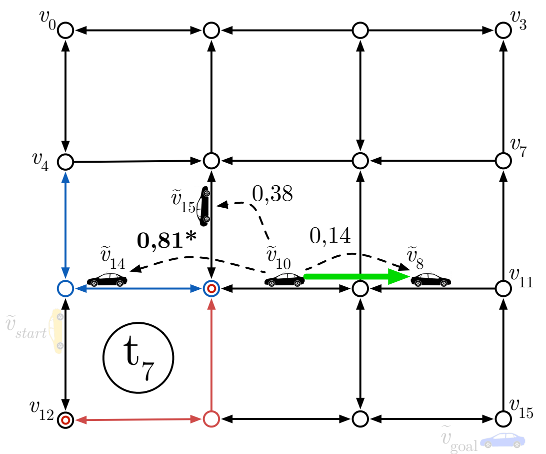
Backward mode execution in $\tilde{v}_8$	
step	action(s)
1a	$\tilde{v}_{start} \notin \tilde{N}$
1b	$\tilde{v}_{start} \approx \{(v_8, v_{12}), 0\}$
2a	$h(\tilde{v}_{10}) = \mathbf{0,35}^*, h(\tilde{v}_{11}) = 0,22,$ $h(\tilde{v}_{13}) = 0,18, h(\tilde{v}_{goal}) = 0,11$
2b	Set $\tilde{v}_{next} = \tilde{v}_{10}$
3a	set crumb value 74
3b	disseminate to $\tilde{v}_8, \tilde{v}_{11}, \tilde{v}_{13}$

M			
$\tilde{v}_x$	$\tilde{f}(v_x, t_0)$	Δd	$\tilde{g}(\tilde{v}_x)$
$\tilde{v}_{start}$	$\{(v_8, v_{12}), 0\}$	$\frac{1}{3}$	$\{(\tilde{v}_{12}, \tilde{v}_{13}), (\tilde{v}_{13}, \tilde{v}_9)\}$
$\tilde{v}_4$	$\{(v_4, v_8), \frac{1}{3}\}$	$\frac{2}{3}$	$\{(\tilde{v}_4, \tilde{v}_8), (\tilde{v}_8, \tilde{v}_{12})\}$
$\tilde{v}_{10}$	$\{(v_8, v_9), \frac{2}{3}\}$	0	$\{(\tilde{v}_9, \tilde{v}_5), (\tilde{v}_9, \tilde{v}_1)\}$
$\tilde{v}_6$	$\{(v_9, v_5), \frac{1}{3}\}$	$\frac{1}{3}$	$\{(\tilde{v}_5, \tilde{v}_1), (\tilde{v}_1, \tilde{v}_0)\}$
$\tilde{v}_8$	$\{(v_9, v_{10}), \frac{2}{3}\}$	$\frac{1}{3}$	$\{(\tilde{v}_{10}, \tilde{v}_6), (\tilde{v}_6, \tilde{v}_2)\}$

Comments: Approximation of edge assignment for $\tilde{v}_{start}$ is as follows:

- $\Delta w_{est} = \Delta t \cdot \frac{1}{3} = (6 - 0) \cdot \frac{1}{3} = 2$
- With $\tilde{g}(\tilde{v}_{start})$ contained in M and Δw_{est} $\Rightarrow$ $\tilde{v}_{start} \approx \{(v_8, v_{12}), 0\}$

In t_6, $\tilde{v}_{start}$ is approximated as $\{(v_8, v_{12}), 0\}$ (using Δw_{avg} of t_5). Since $\tilde{v}_8$ contains no crumb, the initial value m_{init} is calculated with $M.n_{bck} = 1$ (one vertex has been visited since backward mode), $c = 1$ and default values as described in the introduction to this section. Using equation 5.1 this yields $100 \cdot (1 - \frac{1}{2^2}) - 1 \cdot 1 = 74$. According to Figure 5.8, the benefit is calculated in step 3b. Since $\tilde{g}(\tilde{v}_8)$ contains edge (v_4, v_8) that $\tilde{v}_{start}$ is most probably assigned to, $i.d$ gets stored locally and disseminated to all $\tilde{v} \in \tilde{N}$.



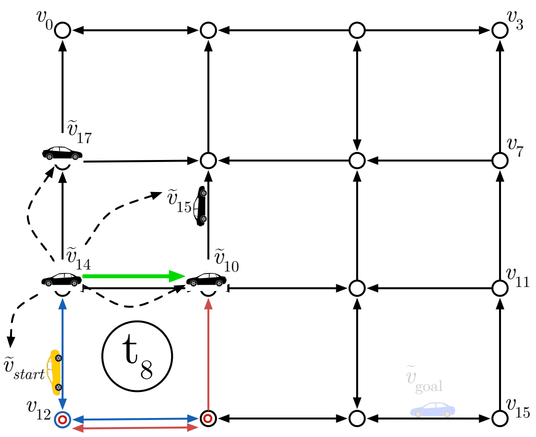
Backward mode execution in $\tilde{v}_{10}$	
step	action(s)
1a	$\tilde{v}_{start} \notin \tilde{N}$
1b	$\tilde{v}_{start} \approx \{(v_8, v_{12}), \frac{1}{3}\}$
2a	$h(\tilde{v}_{14}) = \mathbf{0,81}^*, h(\tilde{v}_{15}) = 0,38,$ $h(\tilde{v}_8) = 0,14$
2b	Set $\tilde{v}_{next} = \tilde{v}_{14}$
3a	set crumb value 48
3b	no dissemination

M			
$\tilde{v}_x$	$\tilde{f}(v_x, t_0)$	Δd	$\tilde{g}(\tilde{v}_x)$
$\tilde{v}_{start}$	$\{(v_8, v_{12}), \frac{1}{3}\}$	$\frac{1}{3}$	$\{(\tilde{v}_{12}, \tilde{v}_{13}), (\tilde{v}_{13}, \tilde{v}_9)\}$
$\tilde{v}_4$	$\{(v_4, v_8), \frac{1}{3}\}$	$\frac{2}{3}$	$\{(\tilde{v}_4, \tilde{v}_8), (\tilde{v}_8, \tilde{v}_{12})\}$
$\tilde{v}_{10}$	$\{(v_8, v_9), \frac{2}{3}\}$	0	$\{(\tilde{v}_9, \tilde{v}_5), (\tilde{v}_9, \tilde{v}_1)\}$
$\tilde{v}_6$	$\{(v_9, v_5), \frac{1}{3}\}$	$\frac{1}{3}$	$\{(\tilde{v}_5, \tilde{v}_1), (\tilde{v}_1, \tilde{v}_0)\}$
$\tilde{v}_8$	$\{(v_9, v_{10}), \frac{2}{3}\}$	$\frac{1}{3}$	$\{(\tilde{v}_{10}, \tilde{v}_6), (\tilde{v}_6, \tilde{v}_2)\}$

Comments: Approximation of edge assignment for $\tilde{v}_{start}$ is as follows:

- $\Delta w_{est} = \Delta t \cdot \frac{1}{3} = (7-0) \cdot \frac{1}{3} = \frac{7}{3}$
- With $\tilde{g}(\tilde{v}_{start})$ contained in M and $\Delta w_{est} \Rightarrow \tilde{v}_{start} \approx \{(v_8, v_{12}), \frac{1}{3}\}$

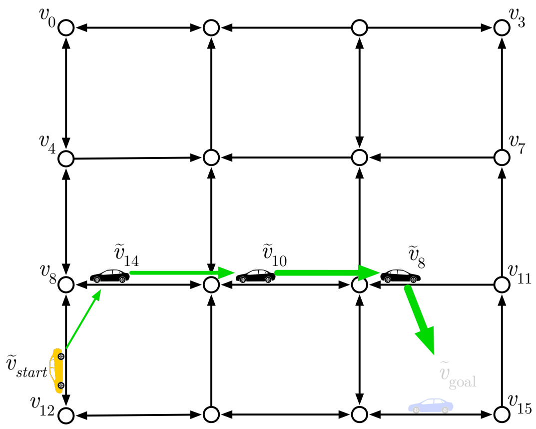
In t_7, $\tilde{v}_{start}$ is approximated as $\{(v_8, v_{12}), \frac{1}{3}\}$ (using Δw_{avg} of t_5). Since $\tilde{v}_{10}$ contains no crumb, the initial value m_{init} is calculated with $M.n_{bck} = 2$, $c = 2$ and default values as described in the introduction to this section. Using equation 5.1 this yields $100 \cdot (1 - \frac{2}{2^2}) - 1 \cdot 2 = 48$. According to Figure 5.8, the benefit is calculated in step 3b: $b_{\tilde{v}} = \frac{40+0}{2} = 20$. This is below the threshold of 50. Hence, $i.d$ is neither distributed nor stored locally.



Backward mode execution in $\tilde{v}_{14}$	
step	action(s)
1a	$\tilde{v}_{start} \in \tilde{N} \rightarrow \tilde{v}_{next} = \tilde{v}_{start}$
1b	-
2a	-
2b	-
3a	set crumb value 22
3b	disseminate to $\tilde{v}_{10}, \tilde{v}_{15}, \tilde{v}_{17}$

M			
$\tilde{v}_x$	$\tilde{f}(v_x, t_0)$	Δd	$\tilde{g}(\tilde{v}_x)$
$\tilde{v}_{start}$	$\{(v_8, v_{12}), 0\}$	$\frac{1}{3}$	$\{(\tilde{v}_{12}, \tilde{v}_{13}), (\tilde{v}_{13}, \tilde{v}_9)\}$
$\tilde{v}_4$	$\{(v_4, v_8), \frac{1}{3}\}$	$\frac{2}{3}$	$\{(\tilde{v}_4, \tilde{v}_8), (\tilde{v}_8, \tilde{v}_{12})\}$
$\tilde{v}_{10}$	$\{(v_8, v_9), \frac{2}{3}\}$	0	$\{(\tilde{v}_9, \tilde{v}_5), (\tilde{v}_9, \tilde{v}_1)\}$
$\tilde{v}_6$	$\{(v_9, v_5), \frac{1}{3}\}$	$\frac{1}{3}$	$\{(\tilde{v}_5, \tilde{v}_1), (\tilde{v}_1, \tilde{v}_0)\}$
$\tilde{v}_8$	$\{(v_9, v_{10}), \frac{2}{3}\}$	$\frac{1}{3}$	$\{(\tilde{v}_{10}, \tilde{v}_6), (\tilde{v}_6, \tilde{v}_2)\}$

In t_8, $\tilde{v}_{goal}$ is contained in $\tilde{N}$. Steps 1b, 2a and 2b are left out. Initial crumb value is calculated with $M.n_{bck} = 3$, $c = 3$ and default values as described in the introduction to this section. Using equation 5.1 this yields: $100 \cdot (1 - \frac{3}{2^2}) - 1 \cdot 3 = 22$. $i.d$ gets stored locally and distributed to neighboring vertices since $\tilde{v}_{start}$ is assigned to an edge contained in $\tilde{g}(\tilde{v}_{14})$.



Result after eight time steps. The route was established during backward mode through $\tilde{v}_{14}, \tilde{v}_{10}, \tilde{v}_8$. The closer to v_{goal}, the larger is the crumb value as indicated by the width of the green arrows. Maintenance procedures ensure that the route keeps being consistent with changes to the topology of the dynamic graph.

5.6. Complexity Analysis

5.6.1. LGHS

In this section, the runtime complexity of LGHS is analyzed using the established O-notation. The analysis gives the complexity for one run on a $\tilde{v} \in \tilde{V}$. Table 5.5 shows the sets that are used during the complexity analysis.

Set	**Semantic**
$\lvert\tilde{N}\rvert$	Number of vertices in $\tilde{N}$
$\lvert\tilde{P}\rvert$	Number of vertices in shortest Path P on static G
$\lvert V\rvert$	Number of vertices in static graph G
$\lvert E\rvert$	Number of edges in static graph G
$\lvert\tilde{V}\rvert$	Number of vertices in dynamic graph $\tilde{G}$
$\lvert\tilde{E}\rvert$	Number of edges in dynamic graph $\tilde{G}$
g	Number of edges returned by $\tilde{g}$. In most cases the size of this set is smaller than $\lvert E\rvert$

Table 5.5.: Notations for sets used for complexity analysis

First, the complexity of functions $\tilde{f}$, $\tilde{g}$ and w is considered. For the first two $O(1)$ is assumed. Both $\tilde{f}$ and $\tilde{g}$ can be implemented as lookup lists and therefore require constant time for accessing an element. In the application domain, w calculates a distance between two cars. This can also be seen as a basic operation and thus $O(1)$ can be assumed.

Complexity of Phase 1

In Phase 1 the algorithm checks for the destination vertex by looking at all elements in $\tilde{N}$. The complexity of this operation is proportional to $|\tilde{N}|$ and thus is in $O(|\tilde{N}|)$. In the worst case $|\tilde{N}|$ contains all vertices of $\tilde{G}$, which results in $O(|\tilde{V}|)$ for Phase 1.

The next step of Phase 1 is to compute shortest path P to the destination vertex on static graph G using Dijkstra's algorithm. According to [Cormen et al., 2000a] the running time of a standard Dijkstra implementation is $O(|V|^2)$. However, [Cormen et al., 2000a] describes a modified implementation using Fibonacci heaps [Cormen et al., 2000b] resulting in a running time of $O(|V|lg|V| + |E|)$.

The last step of Phase 1 is to estimate future graph topology (Figure 5.2). The for-loop is run $|\tilde{N}|$ times. The nested for-loop in line 14 g times. The worst case, when the $\tilde{g}$ returns all edges in G and $|\tilde{N}|$ contains all vertices of $|\tilde{G}|$, results in complexity class $O(|\tilde{V}| * E)$.

Complexity of Phase 2

Phase 2 computes three heuristics (Figures 5.3, 5.4 and 5.5 (left)). All of these three algorithms loop over all elements in $\tilde{N}$ and P. Since they do not depend on each other, they were implemented in one outer and one inner for-loop for reasons of performance optimization. It can therefore be concluded that the complexity using this approach is in $O(|P||\tilde{N}|)$. Assuming the worst-case scenario where $|P|$ contains all edges of G, this can be written as $O(|E||\tilde{N}|)$

Complexity of Phase 3

The best neighbor decision can be integrated into Phase 2 by storing the vertex with the highest heuristic score after each loop over $\tilde{N}$. Therefore, Phase 3 does not contribute to the complexity analysis.

Overall Complexity

Combining the individual results yields $O(|\tilde{V}| * E + |\tilde{V}| + |E| + |V|lg|V|)$. The term $|\tilde{V}| + |E|$ can be neglected. Therefore, the overall complexity of LGHS is

$$O(|\tilde{V}| * E + |V|lg|V|)$$

The interpretation of this result is that the running time of the algorithm is proportional to the number of vertices in $\tilde{V}$ and edges in E. It scales logarithmically with the number of vertices in V. This indicates that the algorithm is efficient enough for large static and dynamic graphs.

5.6.2. LGHS-CD

In this section the overall complexity of LGHS-CD is analyzed. As maintenance mode runs independently from the forward and backward modes, it is considered separately. The section proceeds by looking at the single steps of forward and backward mode and then combining them into an overall complexity. As for LGHS, O-notation is used.

Table 5.6.: Forward mode complexity analysis

Forward mode - complexity analysis		
Step	Action summary	Complexity
1a	Check goal condition	$O(1)$
1b	Check for crumbs	$O(log(\|\tilde{V}\|))$
2a	Run LGHS	$O(\|\tilde{V}\| \cdot \|E\| + \|V\|log(\|V\|))$
2b	LGHS best neighbor decision	$O(1)$
3a	Store delta	$O(1)$
3b	Copy M to $\tilde{v}_{next}$	$O(1)$
Overall complexity: $O(\|\tilde{V}\| \cdot \|E\| + \|V\|log(\|V\|))$		

Forward and Backward Mode

The individual steps of forward mode are listed in Table 5.6. The check for goal condition (1a) is basically a comparison operation and can be achieved in constant time; hence the complexity is $O(1)$. In step 1b the largest crumb is to be determined. In a straightforward approach every $\tilde{n} \in \tilde{N}$ is checked, resulting in runtime $O(|\tilde{N}|)$. However, this can be optimized by storing all values of known crumbs in a tree structure which then yields logarithmic runtime for insert (set new crumb), delete (remove crumb), and read (find largest crumb). Although the update operation is done in constant time, the frequency of the other operations is higher and therefore a binary tree is the optimal choice. In the worst case, when the whole graph is contained in $\tilde{N}$, complexity rises from $O(|\tilde{N}|)$ to $O(|\tilde{V}|)$.
Steps 2a and 2b are only executed if no crumb is available. However, they are included for determining worst case complexity. Step 2a is basically LGHS, whose runtime was identified in Section 5.6.1 as $O(|\tilde{V}| \cdot |E| + |V|log(|V|))$, and 2b (use results of LGHS to determine best neighbor) is in $O(1)$.
Computation of Δd, mapping list m, and storing this information in $\tilde{v}_{next}$ can be completed in constant time. Hence, step 3 is in $O(1)$.

The overall worst case runtime of forward mode is the same as for LGHS: $O(|\tilde{V}| \cdot |E| + |V|log(|V|))$. This only applies for cases where no crumbs are available in a vertex and LGHS is used as a fallback. For all other calls to forward mode, only step 1 and 3 are executed and runtime is in $O(log(|\tilde{N}|))$.

In backward mode, step 1a checks all elements of $\tilde{N}$ for $\tilde{v}_{start}$. As for step 1b of forward mode, depending on the data structure used for realization of $\tilde{N}$ this can be optimized to $O(log(|\tilde{N}|))$. Also here, complexity can change to $O(log(|\tilde{V}|))$ if all vertices of the dynamic graph are contained in $\tilde{N}$.

Approximation of the subgraph in step 1b iterates over all elements in M. Furthermore, all edges for a vertex contained in M are examined in a worst-case scenario. Therefore, subgraph approximation is in $O(|M| \cdot |E|)$. If every vertex $\tilde{G}$ was visited during forward mode, this changes the complexity to $O(|\tilde{V}| \cdot |E|)$. This is, however, unlikely in practice, but since the worst-case scenario is analyzed, $O(|\tilde{V}| \cdot |E|)$ for subgraph approximation is assumed.

Step 2 of backward mode is basically an LGHS run. As determined in Section 5.6.1, complexity is $O(|\tilde{V}| \cdot |E| + |V|log(|V|))$.

Step 3a creates or updates an existing crumbs. Both operations can be done in constant time. Creating a crumb involves computing the initial value according to equation 5.1. Updating a crumb just increments the current value by a global constant c_{inc}. In step 3b benefit is computed (Figure 5.8). The element check in line 8 can be realized using a lookup table. The rest of the algorithm consists of a simple comparison in line 5, and assignments. Therefore, benefit computation takes constant time ($O(1)$).

Overall complexity for backward mode of LGHS-CD is $O(log(\tilde{|V|}) + |\tilde{V}| \cdot |E| + |\tilde{V}| \cdot |E| + |V|log(|V|))$. This can be further simplified to:

$$O(|\tilde{V}| \cdot |E| + |V|log(|V|))$$

Combining complexities of forward and backward mode results in:

$$O(2 \cdot (|\tilde{V}| \cdot |E| + |V|log(|V|))) \in O(|\tilde{V}| \cdot |E| + |V|log(|V|))$$

LGHS-CD therefore is in the same complexity class as LGHS. It scales proportionally with the number of vertices in the dynamic graph and edges in the static graph but logarithmically with the number of vertices in the static graph.

Route Maintenance

consists of three independent operations that are executed when necessary. However, the conditions introduced in Section 5.4.4 need to be checked periodically when $\tilde{f}$ updates $\tilde{N}$. Since all these conditions rely on the neighbors $\tilde{N}$, the complexity is in $O(\tilde{N})$. As for forward and backward mode, $\tilde{N}$ contains all vertices of $\tilde{G}$. Hence, worst-case runtime for route maintenance is in $O(\tilde{V})$.

5.7. The Relation of LGHS and LGHS-CD to this Thesis

This chapter presented LGHS and its extension LGHS-CD which will contribute to implementation of the Pull Paradigm on the network level in Chapter 7. The problem of information discovery and retrieval was mapped to graph theory and solved within this domain. In Section 4.3 the following missing aspects in unidirectional vehicular ad-hoc networks with respect to the network level were identified:

Table 5.7.: Backward mode complexity analysis

Backward mode - complexity analysis		
Step	Action summary	Complexity
1a	Check termination	$O(log(\|\tilde{V}\|))$
1b	Approximate subgraph	$O(\|\tilde{V}\| \cdot \|E\|)$
2a	Run LGHS	$O(\|\tilde{V}\| \cdot \|E\| + \|V\|log(\|V\|))$
2b	LGHS best neighbor decision	$O(1)$
3a	Create / Update crumb	$O(1)$
3b	Benefit computation	$O(1)$
Overall complexity: $O(\|\tilde{V}\| \cdot \|E\| + \|V\|log(\|V\|))$		

- Unicast ad-hoc communication does not address the problem of discovering the target node.
- Routing algorithms do not fully exploit special characteristics of vehicular ad-hoc networks in order to ensure reliable communication.

Both are filled by the algorithms presented in this chapter. LGHS-CD introduced the concept of crumbs and routes in order to “guide” requests to the respective information-providing node(s) and speed up future discovery procedures. Special characteristics of vehicular ad-hoc networks, e.g. high mobility, are approached using comprehensive route maintenance procedures, future graph topology approximation, and the combination of uniform-cost and heuristic search.

In Chapter 7 LGHS and LGHS-CD will be integrated as routing algorithms into the Car2X network stack in order to support the Pull Paradigm on network level. The next chapter deals with application categories and use cases for the Pull Paradigm.

6. Introducing a Novel Car2X Paradigm

This chapter approaches the Pull Paradigm from the application level and builds on previous results from Chapter 5. Table 6.1 shows four novel application categories that are proposed in this thesis. Each of them consists of four use cases. All of them follow the Pull Paradigm and therefore consider all artifacts that are part of the Car2X networks as an information resource. Additionally, two advanced applications of the Push Paradigm are proposed. Both of them incorporate highly vulnerable road users, in this case bicyclists, into the Car2X community. In this chapter each use case of each category is further detailed. Descriptions follow a common structure: *Overview* gives a short summary of the use case, and *Implementation Aspects* describes technical considerations that have to be taken into account for implementation. This mainly addresses special requirements with respect to the Pull Paradigm and refers to solutions that were described in the previous Chapters. Finally, the *Example Scenario* presents the use case from user's perspective.

Pull Services	1 on-demand	1.1	On-Demand Traffic Information System
		1.2	Car2X-Enabled Charging Station for Electric Cars
		1.3	Intelligent Start-Stop
		1.4	Right of Way Conflict Resolution
	2 environmental displays	2.1	Environmental Displays Optimized for Viewer Perspective
		2.2	Virtual Agents for Environmental Displays
		2.3	Context-Aware Environmental Displays
		2.4	Multiple Reality Environmental Displays
	3 social	3.1	C2X Like Me Button
		3.2	Eco-Score Communities
		3.3	Ad-hoc Rides
		3.4	Linking Environmental Displays and Social Networks
	4 personalized	4.1	Majority Decisive Environmental Displays
		4.2	Visual Bookmarking in Environmental Displays
		4.3	Live Video Feeds
		4.4	Microscopic Parking Lot Assistance
Additional Push Services	5 vulnerable road users	5.1	Visibility Enhancement for Bicycles
		5.2	E-Bike Combined Battery and Communication Module

Table 6.1.: Pull application categories and respective use cases of next generation vehicular ad-hoc networks.

Furthermore, the concept of multiple reality is introduced in Section 6.6. Here, perceivable characteristics of real objects are changed in such a way that information transfer is *directed*, e.g. only the intended addressee is able to perceive the changes. This concept will support additional Push applications mentioned above in order to incorporate bicyclists into the Car2X community. A use case is described in Section 6.5.1, and a first multiple reality implementation follows in Section 8.5.

The last section of this chapter chooses use cases for implementation by applying a selection process derived from the literature.

6.1. On-Demand Information Access

The first category concerns use cases in which the driver or vehicle actively requests information from the C2X network (other cars or infrastructure). The desired information is then fetched dynamically and presented to the driver. Applications use vehicular ad-hoc communication to retrieve accurate and up-to-date information from surrounding cars or infrastructure. A major difference from established ADAS is that the applications are user-centric. Information is requested from the ad-hoc network on-demand. This information could either be acquired by sensors or derived from them.

6.1.1. On-Demand Traffic Information System

Overview The On-Demand Traffic Information System uses vehicular ad-hoc communication to retrieve accurate and up-to date information from surrounding cars or infrastructure. It is a prime example of a Pull application: Information is explicitly requested and retrieved over the ad-hoc network.

Implementation Aspects From an implementation point of of view, this kind of application poses several challenges on the network and facility layer. As shown, today's protocols are primarily focused on information dissemination. Within this category of Pull applications, however, point-to-point communication between two vehicles is needed, e.g. between information seeking car and information provider. This presumes unicast communication between vehicles that are most likely not within transmission range of each other. Using a broadcast over the ad-hoc network is not feasible since it would jam communication channels. Thus, there is missing an adequate routing algorithm on the network layer. Furthermore, there is the question of information discovery: When desired information is available from a car outside transmission range, this car has to be discovered and information must be transported back over a highly dynamic network. In order to approach these challenges, LGHS-CD will be used and integrated into the network layer of the Car2X stack (see Section 7.6).
Current message types like CAMs and DENMs are also tailored to safety applications. DENMs in particular contain such application-specific fields. Therefore, in Chapter 7 new message types will be defined for Pull applications.

Example Scenario The user is interested in information about the traffic jam she is experiencing at the moment. Using the on-demand traffic information system, she asks about the remaining length of this traffic jam. Normally, such a traffic event is formed by slow-moving or stationary vehicles. This information can be used to detect the beginning

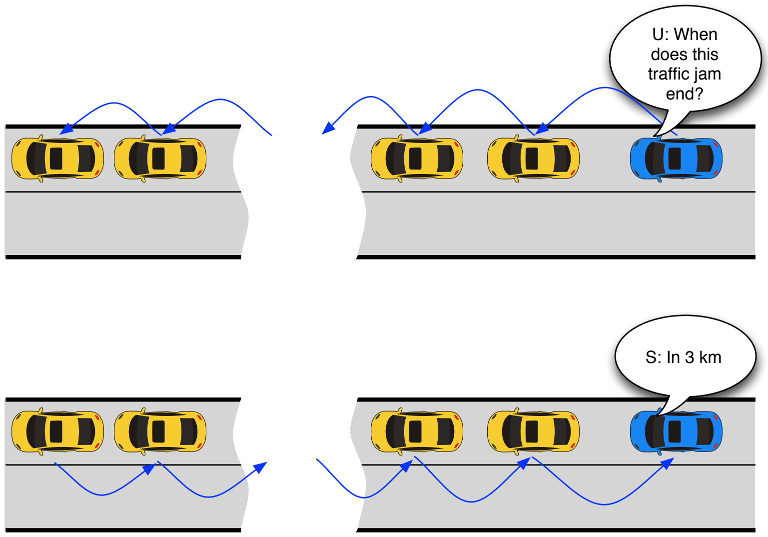


Figure 6.1.: Example scenario of the On-Demand Traffic Information System.

of a traffic jam. Since vehicle positions are known, an estimate of upcoming traffic can be transported back to the EGO car.

6.1.2. Car2X-Enabled Charging Station for Electric Cars

Overview Today's charging stations for electric cars are solely meant to transfer electricity to the car that is connected. They lack the ability to communicate with cars even over wired charging connections. This very "thin" relationship offers potential for new applications in the Car2X domain. By introducing the Car2X-enabled charging station, a variety of use cases can be realized. Even before the car connects, the station can adjust charging voltage, announce its compatibility with the approaching car, or process payments for charging. Using Car2X technology is even reasonable during the time the car is connected. Existing charging stations need to be upgraded to a roadside station while wired data transfer between the two parties requires new standards for plugs and communication protocols.

Implementation Aspects Presence notification of a charging station can be achieved using CAMs. However, it is necessary to add the type "charging station" to the possible types of roadside units. Additionally, a charging station needs to announce its services to surrounding cars. Types of services also need to be included in the list of available services.
Communication between charging station and car has to be realized as unicast transmission. LGHS can be used for this purpose. Depending on the final application, either ad-hoc communication or mobile radio (to increase the transmission range of the charging station) can be used.

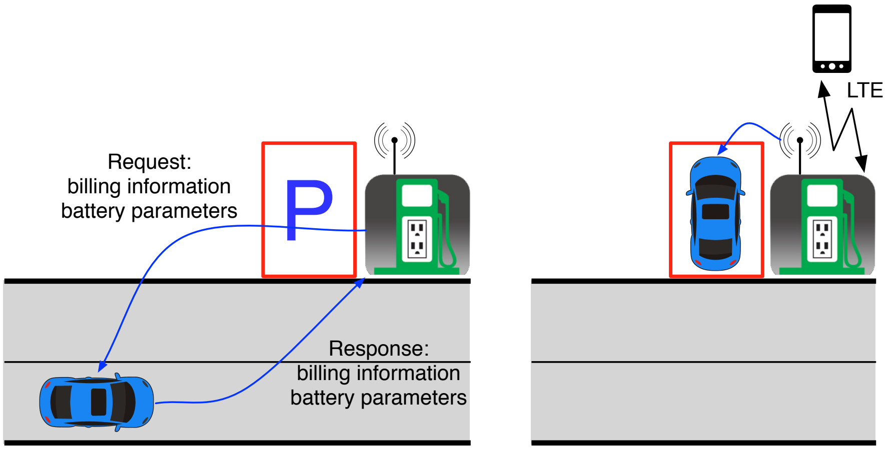


Figure 6.2.: Example scenario of a Car2X-enabled charging station. On the right picture, the driver is informed about current charging state on her smartphone.

Example Scenario In an example scenario, a car is approaching the transmission range of a Car2X-enabled charging station. Since the car's voltage is low, it asks the driver whether she wants to take this opportunity. The driver is able to reserve this charging station by paying a small "reservation fee". The station indicates this by blocking its attached parking space to other cars. Furthermore, provider, billing information, and charging parameters are requested from the car.
During charging, car and station stay connected and the driver is able to view current charging status, remaining time, etc. This scenario is also depicted in Figure 6.2.

6.1.3. Intelligent Start-Stop

Overview Engine Start-Stop Systems are becoming more and more common even in middle-sized cars. They turn off the engine during an intermediate stop. However, in situations where the car stops only for a few seconds, it might be more efficient to keep the engine running. Using Car-to-Infrastructure as well as Car-2-Car Communication this situation can be avoided. By requesting environmental information, remaining time to engine start can be estimated. If this time is below a calculated threshold, the engine can stay running. Note that this use case does not require user interaction.

Implementation Aspects The cars waiting ahead of the EGO may not all be in transmission range. Therefore, the exact number can be acquired using LGHS-CD presented in Chapter 5 and counting the hops.

Example Scenario When the car stops at a traffic light, this use case keeps the engine running if it is determined that fuel consumption for a engine restart is higher than for running the engine for x seconds, where x is calculated from remaining red time and number of cars in front of the EGO vehicle.

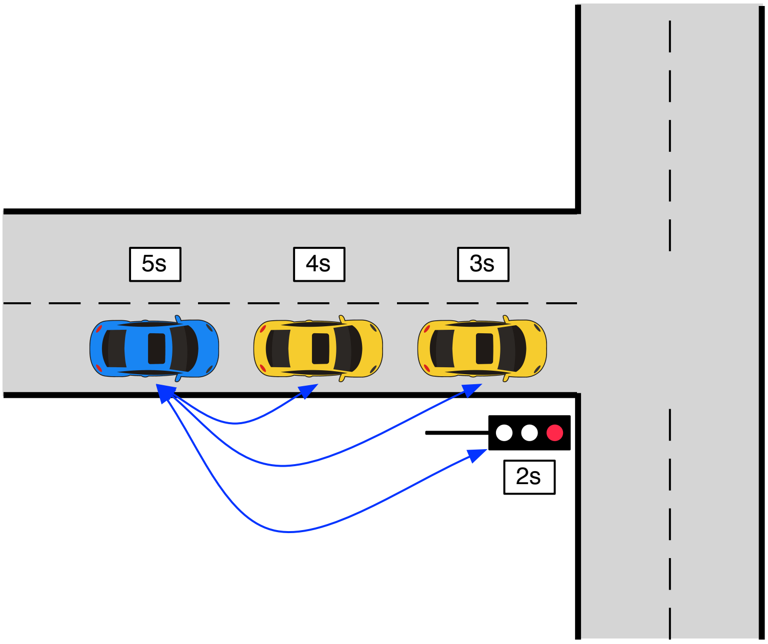


Figure 6.3.: Example scenario of the Intelligent Start-Stop use case. Blue (EGO) car requests number of cars ahead as well as remaining red time from traffic light. From this information, remaining stationary time for EGO is calculated and fed into the Start-Stop system.

6.1.4. Right of Way Conflict Resolution

Overview In ambiguous traffic situations Car2X technology can be used to resolve conflicts. In a first phase, such a situation is detected. During the second phase, applications on the cars involved communicate in order to resolve the situation according to a pre-defined scheme. In the third and last phase the driver is informed about the negotiated resolution. Following the principles of the Pull Paradigm, the first phase can either be initiated by the application itself, or the driver can request a resolution from the system. In the first case, the application needs to constantly scan for ambiguous traffic situations, while in the second case this overhead is avoided.

Example Scenario In a situation where four cars approach an intersection at the same time as shown Figure 6.4, right of way is undetermined according to traffic rules. Drivers need to resolve this conflict by communicating face to face, which might be misunderstood. A Car2X application as proposed can identify the conflict and resolve it without ambiguity for all parties.

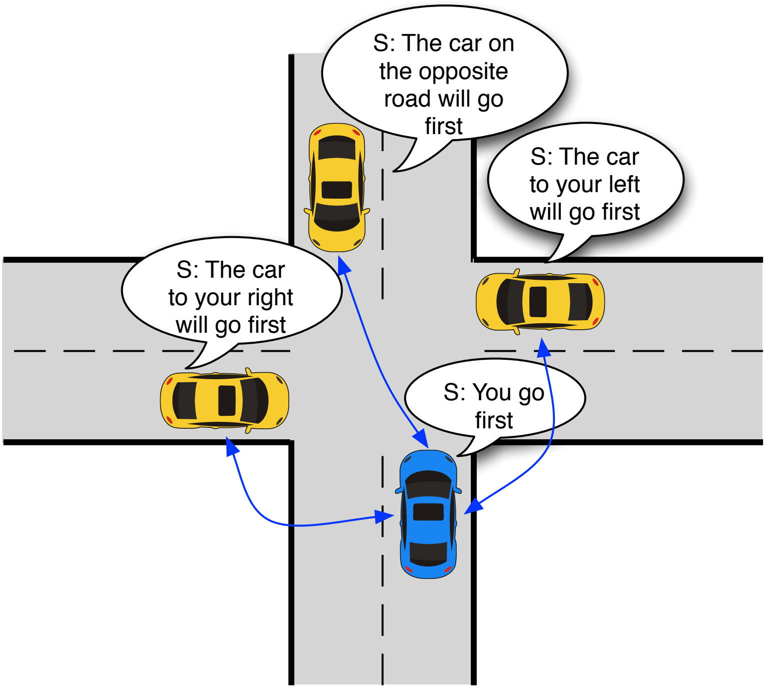


Figure 6.4.: Right of way conflict resolution example.

6.2. Car2X-Enabled Environmental Displays

The category of Car2X-enabled Environmental Displays describes use cases that make use of so called environmental displays. With decreasing costs for large displays, they have become more and more common for outdoor and indoor use during recent years. Typical indoor applications are interactive information boards for shopping malls. Environmental displays for outdoor use were introduced at bus stops or even in front of traffic lights as a dynamic substitute for billboards. They mainly show advertisements or information about current bus / train schedules. The use cases of this category aim for integration of environmental displays into the Car2X Community.

6.2.1. Environmental Displays Optimized for Viewer Perspective

Overview From within a moving car, the environment is perceived as highly dynamic even at low speeds. Visual perception capabilities of the driver are limited in such situations. This use case optimizes presentations of an environmental display by adapting to the current viewing angle of a moving car. Optimization is achieved by distorting the image such that the image appears as if it is being views straight on.

Implementation Aspects Technically, viewing angle can be determined by using information contained in standardized Common Awareness Messages (see Section 2.2).

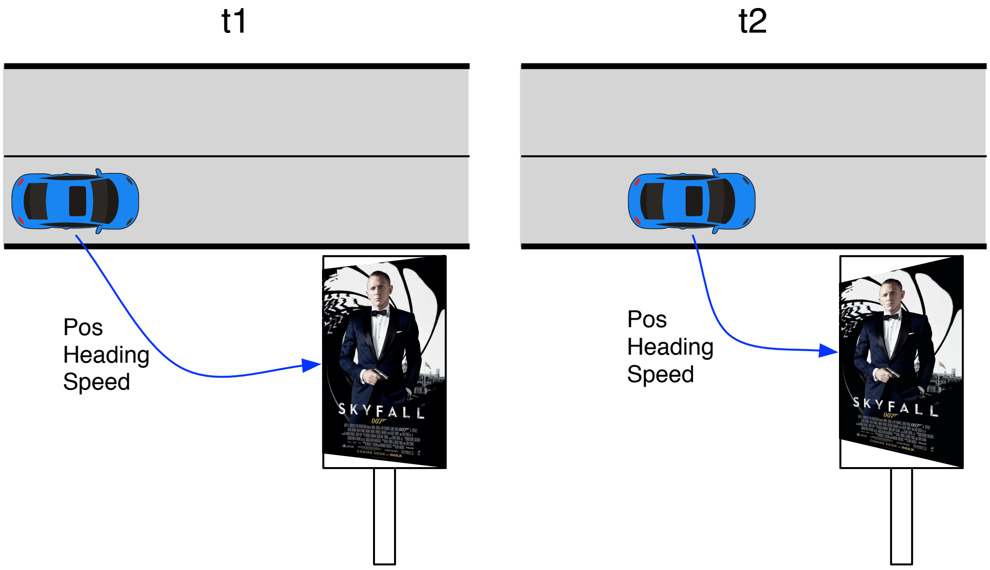


Figure 6.5.: Example scenario of an environmental display optimized for viewer perspective.

However, update frequency of CAMs is typically 1 Hz and thus too low to adapt in real time. Depending on the current speed limit, the effect is likely to appear stagnant. This makes it necessary to predict path and position during two updates. This prediction is already part of an established safety-related use case described in Section 3.2. Cross traffic assistance also predicts future vehicle positions by relying on CAM information.

Example Scenario The car in figure 6.5 is driving along a road with an environmental display on the right side. The display shows an advertisement for a movie. The image is distorted dynamically according to the viewing angle calculated from CAM information.

6.2.2. Virtual Agents for Environmental Displays

Overview The second use case of this category employs a virtual agent during interaction with the environmental display. The idea is to use an abstract representation of an agent within a car, e.g. a cartoon character. It is able to leave the car and fetch information for the driver. While outside the car, it transforms into a person-like character that becomes visible on the environmental display, naturally "fetches" desired information and jumps back into the car. This visualizes the Car2X concept in a certain way and makes it a more real experience. The jumping-out animation should be synchronized between the two communicating parties, e.g. transition between the in-car screen and the environmental display should be visible. The concept of virtual characters that are present on various physical displays has already been researched in the past, e.g. [Kruppa, 2006]. However, to the best of my knowledge, it has not been investigated using Car2X Communication.

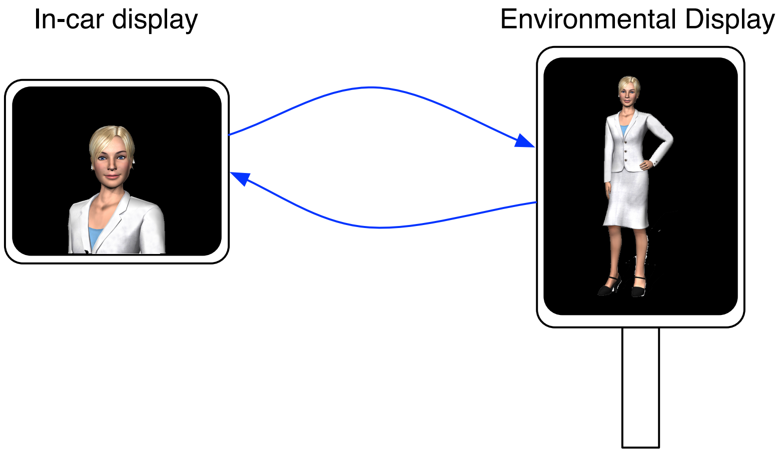


Figure 6.6.: Example scenario of an environmental display using a Virtual Character to visualize information transfer.

Implementation Aspects This use case makes clear that certain Pull application cannot be implemented using standardized Car2X messages. The fields of CAMs are neither applicable nor intended for this kind of communication. When using ad-hoc communication, synchronization between environmental display and car as well as information transfer needs to be realized. In order to be able to implement such an application, new message types and a different communication paradigm is needed.

Example Scenario The driver is interested in an advertisement shown on an environmental display. While standing at a traffic light she says "Please give me more information about this movie". The system confirms the request and negotiates animation speed and virtual character details with the environmental display. After completion, the car part of the application signals the start of the animation.

6.2.3. Context-Aware Environmental Displays

Overview The connection between environmental displays and cars offer the possibility to optimize content presentation depending on traffic situations. Animations, for example, attract attention but also distract the driver from her primary task. Using the possibilities of Car2X Communication, the complexity of the content presentation can be adapted to the situation: During the red phase of traffic lights, short video clips can be shown on the environmental display while during the green phase only static information is displayed. Another possibility is the exploitation of larger font sizes for faster-moving cars. While in the first example the traffic light communicates with the display (Infrastructure to Infrastructure Communication), the second example uses Car to Infrastructure Communication.

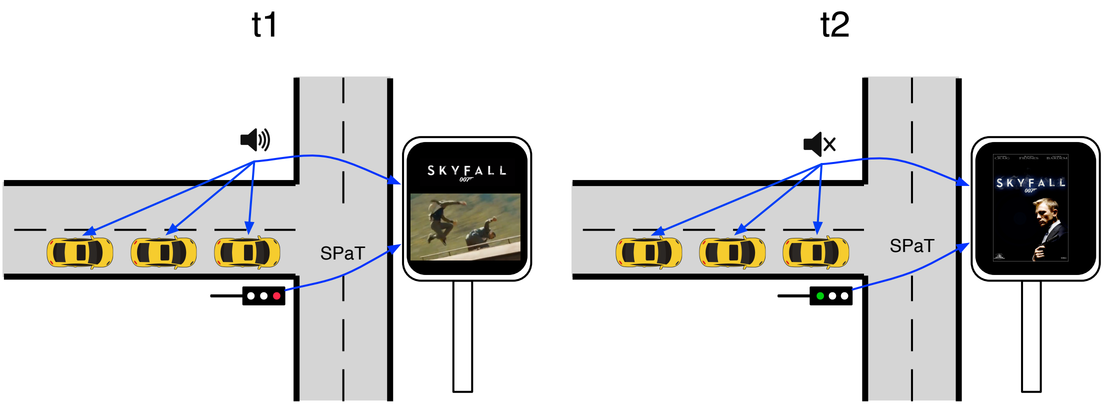


Figure 6.7.: Example scenario of an environmental display using context information for content presentation.

Implementation Aspects Context-aware environmental displays can be realized using CAMs and SPaTs. The latter (signal phase and timing messages) are used in applications like the green light optimal speed advisory. They contain status information from traffic lights. These can be used by the application in order to distinguish between red and green phases.

Example Scenario Figure 6.7 depicts an example scenario for this use case. T1 on the left shows a red phase. The environmental display recognizes this situation by evaluating a SPaT message and presents a movie trailer. At the same time, audio is broadcast using consumer WLAN, e.g. 802.11g which provides higher quality of service and reduces packet loss to a minimum. Cars that have enabled this service will be able not only to watch the trailer during the red phase but also to listen to it. When the traffic light turns green in t2, the environmental display stops the trailer and presents a still picture, e.g. movie poster. Also, audio broadcast is disabled.

6.2.4. Multiple Reality Environmental Displays

Overview Using the concept of multiple reality that will be introduced in Section 6.6, this use case exploits individual perception of the physical objects. The idea uses environmental displays augmented by multiple reality as traffic signs. This enables a personalized one-to-one information transfer on publicly visible objects either because information presented is not relevant for others or because privacy aspects have to be considered.

Implementation Aspects As for all use cases that are based on multiple reality, the possibilities for technical realization are crucial. Within this thesis, a first multiple reality implementation is presented in Chapter 8 that uses infrared light to exclude non-participants from the communication.

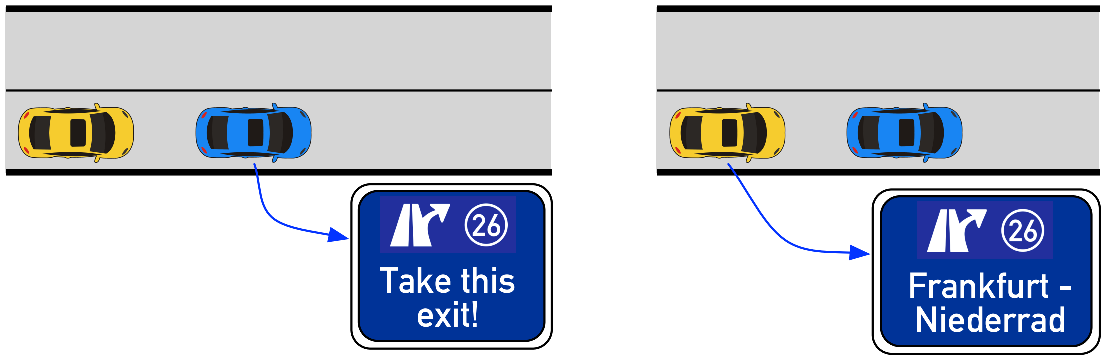


Figure 6.8.: Example scenario of an environmental display augmented by multiple reality. The text "Take the next exit" can only be perceived by the driver in the blue car. The driver of the yellow car (right picture), perceives the name of the exit.

Example Scenario Figure 6.8 illustrates this use case. On the left an individualized one-to-one traffic sign is shown to the blue car. It indicates the driver to take the next exit. In the right picture, the view for the driver in the yellow car is depicted. She only sees the name of exit "Frankfurt - Niederrad" as this exit is not relevant for her. The two pictures in Figure 6.8 show how the content of the signs in perceived by each driver *at the same time*.

6.3. Linking Vehicular Ad-hoc Networks with Online Social Networks

People use social networks on a daily basis. The mobile versions allow them to interact from anywhere and cars are no exception. The profiles that users upload on social networks feature structured lists of personal information, preferences and characteristics. This opens an opportunity for finding common interests between people within the same social network. The introduction of this to the Car2X domain means that the fact that people are in the same location can be exploited (also see [Castronovo et al., 2011b; Mitrevska et al., 2012]). Thus, by combining both, limitations of anonymous and socially-isolated driving can be overcome. This category of use cases provides examples of how social networks can be introduced in the existing domain of Car2X.

6.3.1. C2X Like Me Button

Overview It has become common practice to "like" certain objects on the internet. This refers to nearly every virtual building block of the world wide web: blogs, images, products and so on. The process of "liking" triggers an entry on the personal social network page of the user and thus the information is shared with a previously-defined

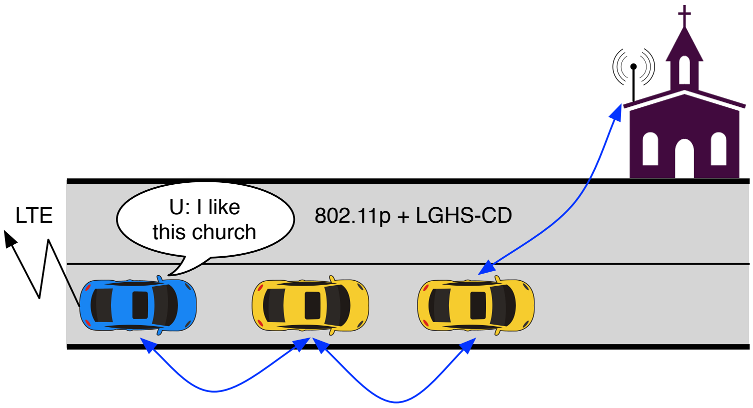


Figure 6.9.: C2X Like Me Button example scenario. After the user states her request, 802.11p and LGHS-CD are used to determine the physical object. Via LTE the liked church is posted on the user's personal page.

group of people. The transfer of this process to real-life objects can enhance the driving experience and Car2X is a key technology for realization. The idea is basically the possibility to like any physical object in view of the driver: other cars, buildings, shops, points of interests.

Implementation Aspects From an implementation point of view, this use case is challenging for Car2X technology. Wireless ad-hoc communication as well as mobile radio can be used for realization. The latter would require a model of the environment to be downloaded into the car. When the user asks to like an object this model is queried and shared among friends. The usage of ad-hoc communication requires objects to be equipped with this technology. Then, the object to be liked has to be identified and disambiguated. This demands new message types as standardized CAMs or DENMs only contain limited information. Also, an ad-hoc search process among the physical, Car2X-enabled objects has to be started. As this is not possible in state-of-the-art Car2X networks, LGHS-CD can implement this feature.

Example Scenario In Figure 6.9 an example scenario is shown. The user wishes to like a church in front of her. As the building is not in transmission range, LGHS-CD over wireless ad-hoc networking is used to discover the desired object. On success, LGHS-CD returns the information and the application running on the vehicle updates the status page of the user via LTE.

6.3.2. Eco-score Communities

Overview Eco-friendly driving is a key factor for curbing carbon dioxide emissions. Following the goal of the social use-case class, the driving style of friends is used to motivate or challenge drivers to drive in a more eco-friendly way. This introduces an eco score and reminds friends of their current status within the community.

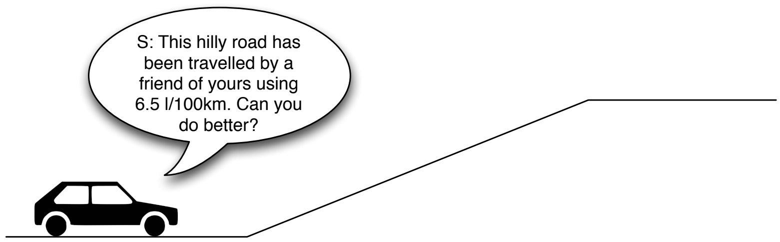


Figure 6.10.: Example scenario of an eco-score community.

Implementation Aspects This use case can be implemented using cellular radio like LTE.

Example Scenario In Figure 6.10 an example scenario is shown. The driver is approaching a hill and the system has access to the fuel consumption of friends within the community. Using this information the system can provide this score to the driver before she reaches the hill and challenge her to do better.

6.3.3. Ad-hoc Rides

Overview Navigation systems often come as standard equipment in today's cars and are therefore disseminated widely. If activated, the system is aware about the current route and destination. This information is already exploited in an established Car2X use case. The traffic light assistant (see Section 8.1.4) which was implemented for sim$^{\mathrm{TD}}$ highlights the most probable path when approaching an intersection. One of the criteria for the most probable path is the route set in the navigation system. The proposed use case ad-hoc rides also uses this information and sends it to a central service. Users outside the car who want a ride connect their smartphone to this service. The application then brings together car drivers as "mobility providers" and pedestrians as "mobility users". Mobility is provided spontaneously (ad-hoc) and as a service.

Implementation Aspects From an implementation point of view this use case can be realized via a central service using cellular technologies but also ad-hoc communication is conceivable. However, this can only be reliable in dense traffic. For the latter, LGHS-CD can be used for discovering an information provider.

Example Scenario An ad-hoc example is shown in Figure 6.11. In this example, driver and pedestrian are connected to a central service. After the pedestrian communicates her need a ride to the central service, cars with a matching route are notified. The driver can then decide whether to accept or reject the request. If the driver agrees, the application brings together mobility provider and mobility user.

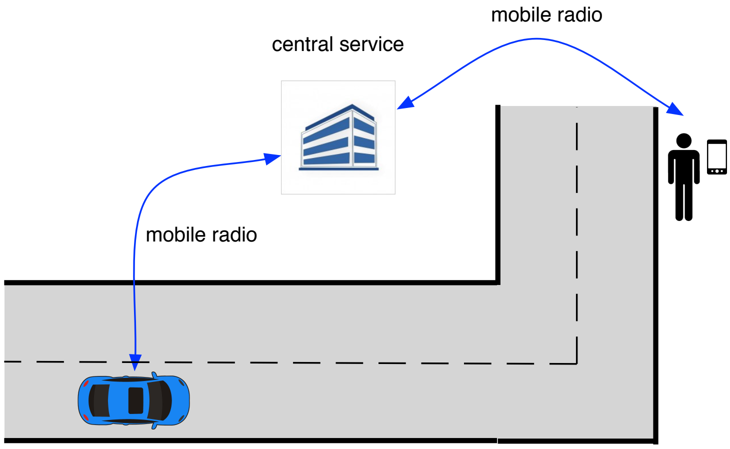


Figure 6.11.: Example scenario of an ad-hoc ride.

6.3.4. Linking Environmental Displays and Social Networks

Overview This use case creates a bridge between environmental display, car, and social network. First, the driver signals her interest in the content currently shown on the display. The car then requests a "content hint" from the environmental display using vehicular ad-hoc communication. After that, the application queries an online service via mobile radio (LTE) in order to retrieve additional information on the content. This approach is robust against transmission errors caused by the car traveling out of range of the display. The introduced "content hint" can be transmitted in one single packet. The social network is also connected using LTE.

Implementation Aspects As for other applications within the Pull Paradigm, unicast communication between the environmental display and the car is required. This is not possible using the established Car2X network stack. Therefore, new message types and unicast oriented communication needs to be added in order to implement this application. Communication between online services and the car can be realized using state-of-the art IP-based protocols over mobile radio. Furthermore, data fields contained in regular CAMs are also not applicable to this scenario. An additional field for the presence information of environmental display has to be added.

Example Scenario Bob has a C2X-equipped car. He drives to work and passes by a billboard. It shows a commercial about the new James Bond movie. Bob asks his car: "What is this movie?" The in-vehicle system recognizes the question refers to the billboard and sends a content request C2X message to the billboard using C2X Communication technology. The billboard responds with a message about the current content: name of the movie and the cast. The in-vehicle system tells Bob this is a James Bond movie and asks if he wants tickets. Bob answers: "yes and please ask Alice if she wants to join". The system now sends another message to the billboard asking about purchase

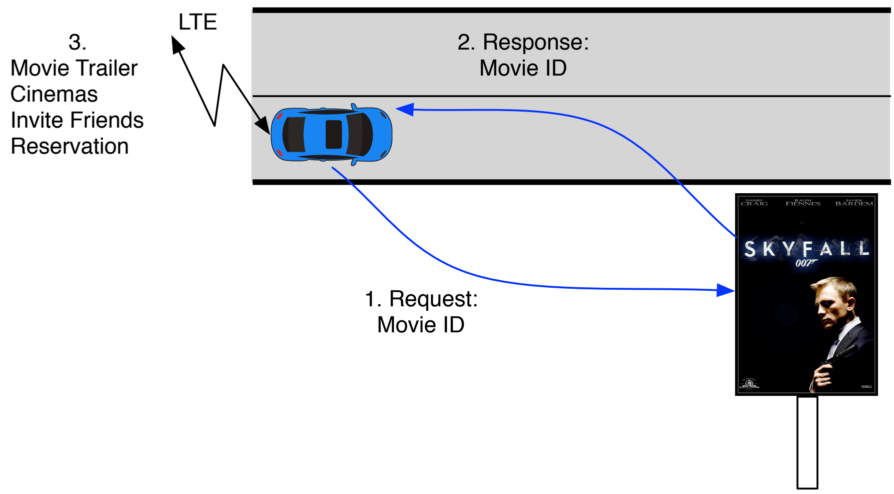


Figure 6.12.: Example use case involving environmental display and social network. 1,2: Request movie id from display and response using vehicular ad-hoc communication. 2: Contact online service via mobile radio (LTE). Retrieve trailer, make reservations after asking friends to join.

information. The system also connects with a social network and sends a message to Alice asking if she is interested in going to see the movie with Bob. The billboard responds with a link for buying tickets. The in-vehicle system proceeds with this information.

6.4. Personalizing Car2X Applications

Personalizing a driver's workplace is a common feature of the modern car. Besides providing comfort, personalization also increases safety while driving. Different types of sensors provide important data that form the basis for adaptive applications. C2X technology can be used to complement the variety of sources to provide adaptive systems.

6.4.1. Majority-Decisive Environmental Displays

Overview The connection between vehicles and environmental displays allows a highly dynamic content presentation. This was already shown in use case 2.3 (context-aware environmental displays). Another interesting application in this context are environmental displays that select their presentations according to certain metadata received by vehicles in transmission range. According to this metadata, presentations would be personalized to the majority of the receivers.

Implementation Aspects The standardized common awareness message is not entirely suitable for a realization of this use case. Although already contained information can be used (speed, heading, etc.) there are other interesting parameters that are not available, e.g. gender or age of the driver. Therefore, either a new message type or modified

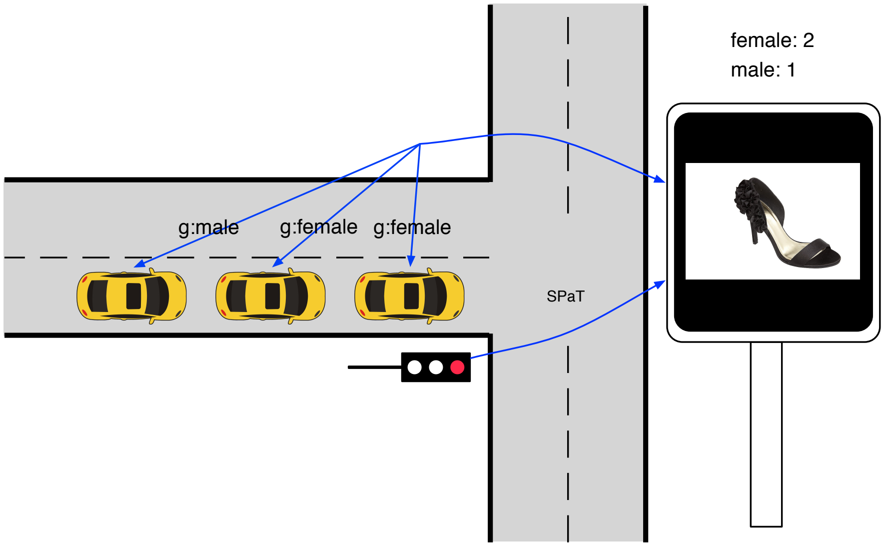


Figure 6.13.: Example scenario of a majority-decisive environmental display showing a personalized advertisement tailored to a certain group of people.

CAMs need to be defined. In order to limit communication overhead, environmental displays would request such “augmented messages” and vehicles only in this case would send enhanced information.

Example Scenario In Figure 6.13 an example of a majority-decisive environmental display is shown. During the red phase of a traffic light, the environmental display requests metadata “gender” from Car2X enabled vehicles. Since the majority of received messages reveal female drivers looking in the direction of the environmental display, an advertisement tailored to women is displayed.

6.4.2. Visual Bookmarking in Environmental Displays

Overview The term “visual bookmarking” has been established in desktop browsing. It allows the user to store a visual reference to web pages. Additionally, tags can be attached to the bookmark. This makes it easier to find a web page later on. The use case “visual bookmarking in environmental displays” transfers this idea to the automotive domain. It allows the driver to store visual content of the display as well as pointers to additional information upon request. At a later point in time the driver can then safely browse the content and retrieve additional information.

Implementation Aspects From an implementation point of view this use case relies on two wireless communication technologies: mobile radio and vehicular ad-hoc communication. First, the car requests an ID from the display. Second, a picture and pointers to additional information are downloaded using an internet connection. This makes the

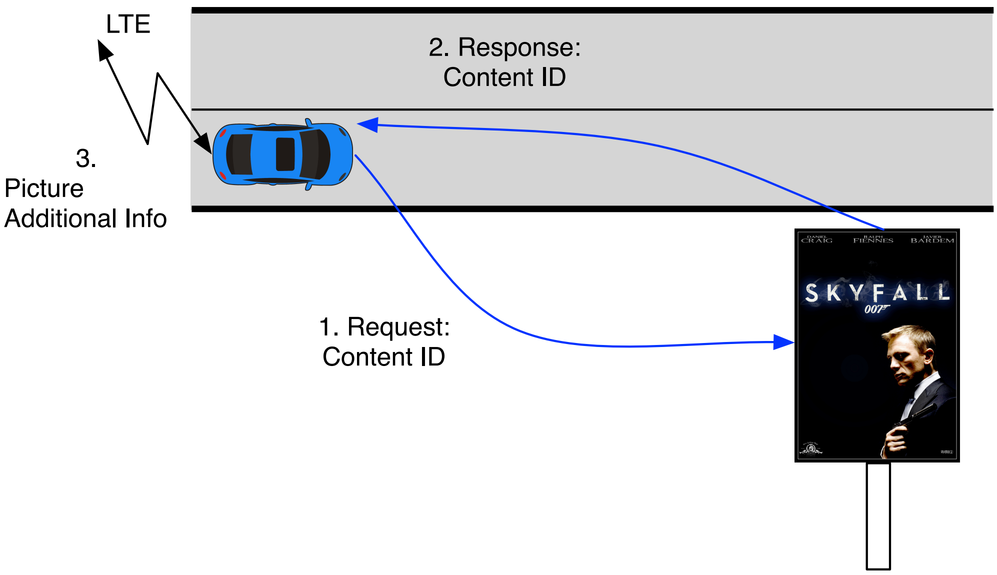


Figure 6.14.: Example scenario of a visual bookmark. Upon request, the content ID is retrieved from the environmental display. Additional information is downloaded using an alternate wireless communication technology.

application robust against the limited transmission range of the environmental display as relevant information can be retrieved in one single packet. As for other Pull applications, unicast communication is needed.

Example Scenario An example scenario is shown in Figure 6.14. After retrieving several visual bookmarks the driver can browse stored content at a later point in time.

6.4.3. Live Video Feeds

Overview Access to webcams was one of the added-value services evaluated in sim^{TD}. This gave drivers visual access to transport nodes. The connection was realized via cellular networks (UMTS) and pictures were downloaded at regular intervals, i.e. no live video (also see Section 3.4). Other approaches by [Gorius et al., 2013, 2012] provide video streaming over 3G/4G networks. The novel use case "live video feeds" proposes a similar functionality using the vehicular ad-hoc network for providing real-time video. A driver can request video streams from other cars and the feed is streamed over other cars.

Implementation Aspects This use case poses several challenges from an implementation point of view. It needs a stable connection from requesting to sending car. Additionally, the route between these two cars might change with network dynamics, i.e. a stable routing algorithm is required. Implementing LGHS-CD (see Chapter 5) as such a routing algorithm within the network layer allows a realization of this use case.

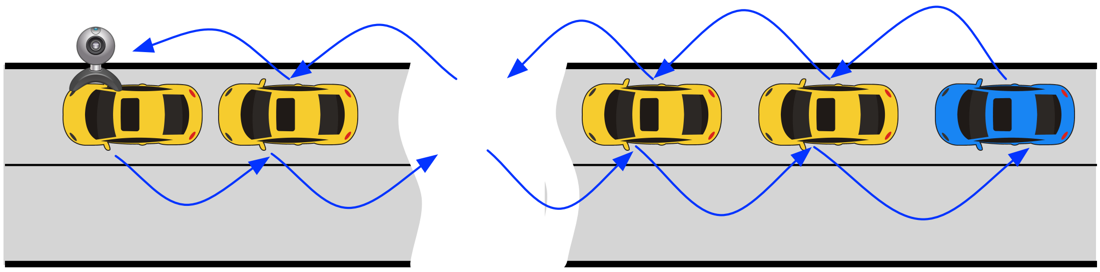


Figure 6.15.: Example scenario of C2X live video feeds.

Example Scenario In Figure 6.15 an example scenario is shown. The blue car requests a video stream from the car driving in front. As it is out of communication range, LGHS-CD is used to discover a stable and shortest path to the sender. Once found, LGHS-CD delivers the stream to the requesting car. Note that the route maintenance procedure of LGHS-CD copes with fast topology changes of the vehicular ad-hoc network.

6.4.4. Microscopic Parking Lot Assistance

Overview Searching for a free parking lot can be cumbersome during rush hour. Some cities guide traffic by showing the availability on public displays throughout the urban space. This macroscopic support is useful for driving to the right parking lot. However, upon arriving it is up to the driver to find an empty spot. If there are only few available it is like looking for a needle in a haystack. The Car2X-based parking lot use case is an assistance system that provides microscopic and up-to-date status of spaces of a parking lot. It supports the driver in finding the way from the entrance to the assigned parking lot. The use case assumes an "always-on" feature in every parked vehicle, e.g. every car is able to send and receive messages while the engine is powered off. This is an important feature of electric cars.
When requesting an empty parking lot, the car sends out a query over the vehicular ad-hoc network. Parked cars can then return empty spots to the requesting car.

Implementation Aspects This use case can be realized by using several sensors available in modern vehicles. It is important to mention that a GPS signal is most likely not available in a parking garage. In order to be able to guide the driver to her parking space, a map is generated by merging sensor data from cars that previously entered the parking garage. This approach is similar to the dynamically generated roadwork geometry from the roadwork information system (see Section 3.3.4). In this use case, however, the map of a parking garage is not derived from GPS data but from vehicle speed, steering angle, and barometric height measurement. The last sensor data reflects to the number of available parking levels.
An empty spot can also be detected by proximity sensors. This use case relies on point-to-point ad-hoc communication which is not possible in state-of-the-art Car2X networks.

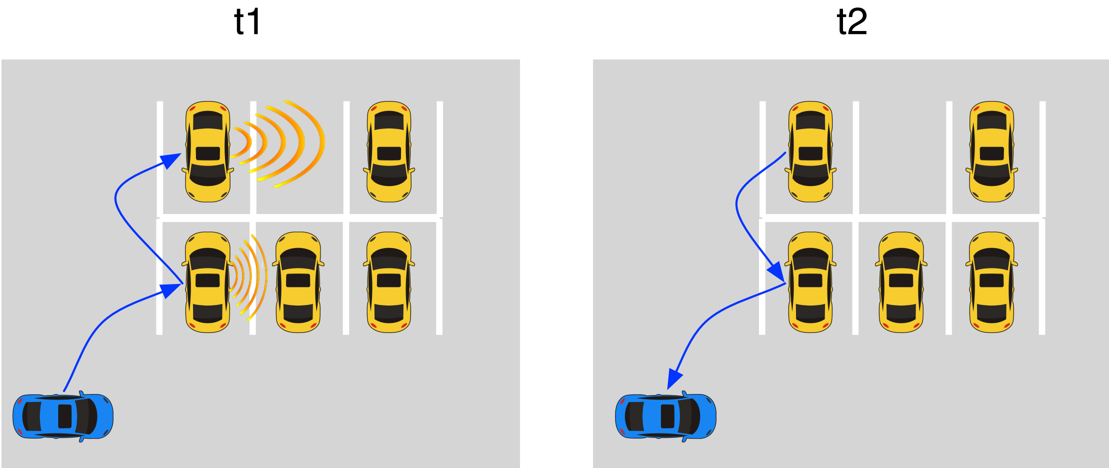


Figure 6.16.: Example scenario of microscopic parking lot assistance.

Therefore, LGHS-CD is essential on the network layer. New message types for Car2X based information requests have also to be defined on the facility layer.

Example Scenario In Figure 6.16 a simple example of this use case is shown to give the reader an idea how it works. It is simplified in the sense that only a 2D representation of the problem is given, i.e. different levels of a parking garage are not shown. The blue car is requesting an empty parking space and LGHS-CD is used to retrieve the desired information over the ad-hoc network. In this example the first car that receives the request is sensing a blocked space and forwards the message. The answer is then generated by a car that is parking next to an empty spot.

6.5. Additional Push Applications

This section presents additional Push applications that involve highly vulnerable road users. Up to now, bicyclists have not been included in the Car2X-Community.

6.5.1. Visibility Enhancement for Bicycles

Overview The use case of cross-traffic assistance within the Push Paradigm increases safety in potentially dangerous situations at intersections (see Section 3.2.3). A variant of this use case is the cross-traffic assistance of BMW that includes a motorcycle. In the intended scenario the motorcycle is driving on a major road and approaches an intersection. At the same time a car also reaches the same intersection. If the car's driver is unlikely to yield, the visibility of the motorcycle increases. Headlights and additional mounted LED warning lights come on. At the same time the car autonomously issues 30% of the car's braking power.

This additional Push use case transfers this concept to bicycles as pictured in Figure 6.17. The situation is similar to the cross-traffic assistance but includes a bicycle. What needs to be considered is that bicycles do not come with a large amount of technology. This use case therefore proposes a small-footprint architecture that relies on existing components where possible.

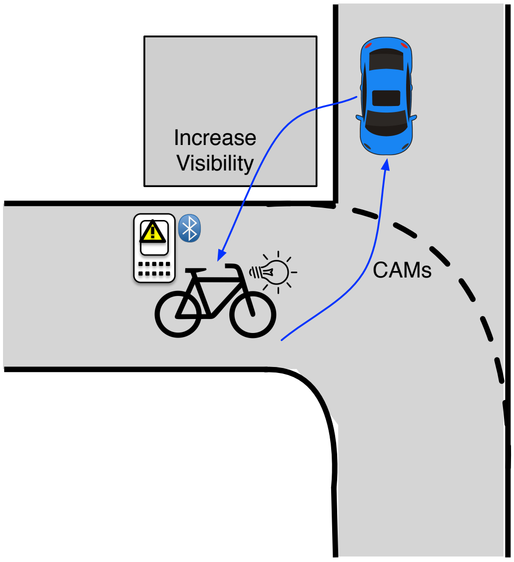


Figure 6.17.: Visibility Enhancement for bicycles. Application logic on the car (blue) detects a potentially dangerous situation and requests visibility enhancement from the bike.

Implementation Aspects Although this use case is related to cross-traffic assistance, implementation poses several challenges. The small-footprint architecture on the bike has to be carefully designed. Since only the car assesses the situation, existing message types for requesting visibility enhancement are not applicable. As for other (Pull) use cases new message types have to be defined. Furthermore, visibility enhancement technologies for bikes need to be considered. A detailed discussion of this topic will follow in Section 6.6, where the multiple reality model is introduced.

Example Scenario In Figure 6.17, sensor data that is included in CAMs is acquired from the bicyclist's smartphone. It is connected via Bluetooth to the communication module on the bike. Because computing power on the bike is limited, situational assessment is solely done on the car which — in case of an impending crash — then requests visibility enhancement from the bike.

In Section 8.5 an implementation of this use case will be presented.

6.5.2. E-Bike Combined Battery and Communication Module

Overview Recent advances in bike technology led to the availability of electric bikes (e-bikes) that feature an electric-powered engine which assists the rider during cycling. Batteries on the bicycle's rack or frame supply the engine with power. With the availability of electric power on bikes they can potentially carry additional information technology, such as a communication module. Similar to the visibility enhancement for bicycles, this use case makes use of smartphone sensors. But with the additional technology present on an e-bike, the smartphone also gains access to the bike's sensors. This bi-directional information transfer enables a variety of applications (see example scenario below).

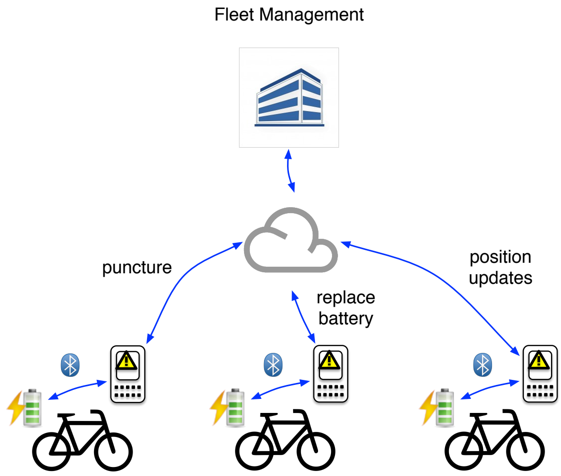


Figure 6.18.: Use case for the combined battery and communication module: fleet management of e-bikes.

Implementation Aspects Besides the considerations presented in Section 6.5.1, the integration of a communication module into the e-bike's battery is a major challenge for this use case. It is further discussed in Section 10.3 where a demonstrator of this use case is described.

Example Scenario The proposed scenario is a fleet management system for e-bikes as depicted in Figure 6.18. With the smartphone being able to access the battery and other bike sensors, it can use its mobile connection to inform the fleet management of the current bike status. The fleet management, on the other hand, can then send out maintenance teams to a specific to a bike. Using the smartphone's GPS receiver the exact position can be determined. This also enables the fleet management to verify and change the distribution of the fleet. Payments for rental companies can be processed via the smartphone.

Of course, safety-relevant functions can be added as well. The small-footprint architecture as outlined in 6.5.1 can be combined with the battery. Section 8.5 will analyze the requirements of such an architecture and present an implementation.

A demonstration of the fleet management component was given to the public at Mobile Word Congress (MWC) 2013, Barcelona, Spain. A full description of the demonstrator is presented in Section 10.3.

6.6. The Multiple Reality Model

Car2X-based applications and IVIS come with a variety of capabilities to notify/warn the driver of upcoming events. The common practice is to use visual or auditory modalities. Visual notifications are often realized as displays but also warning lights that are integrated into the car's dashboard. In modern cars there are also head-up displays (HUDs) where information is projected on the windshield. So far, this technology has been used for displaying information that can be obtained from onboard sensors, like navigation instructions or current speed. To the best of my knowledge HUDs have not been combined with Car2X technology.

Besides information that is displayed within the car, Chapter 3 describes a use case which differs from these common notification approaches: The lateral cross-traffic assistant uses auditory and visual means of other road users involved. In a nutshell, in a case of an impending crash, the system on the motorcycle switches on headlights, additional LED warning lights and the horn. Looking deeper into the system, it turns out that both applications, the one running on the motorcycle and the one running on the car, act independently from each other. The assessment of critical situations relies on beaconing information. This approach to information delivery has an obvious drawback as it attracts the attention of other, uninvolved road users as well. This might also result in unnecessary distraction from the primary task. In the following, the term "full reality" refers to applications using this approach.

In order to avoid distracting other road users, one could use "augmented reality". Here, data from video cameras are used, the pictures analyzed to find certain patterns. Then, overlays or warning messages/tones are generated. In contrast to the described lateral cross-traffic assistant, other parties remain uninvolved. Another advantage is the possibility of individual information delivery by using only one information resource. However, this approach requires a fairly large amount of computational power, and tracking while driving is also error prone.

The Multiple Reality Model combines benefits of both approaches: Information transfer is *directed* to the intended addressee while tracking objects is robust at the same time. (see Table 6.2). This goal is achieved by changing the perceivable characteristics of objects. In contrast to the described example of lateral cross traffic assistance, both parties do not assess a situation independently. Following the "Pull Paradigm" the perceivable characteristics are changed upon request using Car2X Communication. The

<table>
<tr><td></td><td>Augmented Reality</td><td>Multiple Reality</td><td>Full Reality</td></tr>
<tr><td>Tracking robustness</td><td>objects have to be recognized and tracked via error-prone pattern recognition</td><td colspan="2">very robust as perceivable characteristics are changed in real environment</td></tr>
<tr><td>Directed</td><td colspan="2">directed</td><td>not directed</td></tr>
<tr><td>Perceivable by</td><td colspan="2">perceivable only by intended addressee</td><td>perceivable by everyone</td></tr>
</table>

Table 6.2.: Analysis of the described approaches to visibility enhancement in means of robustness, addressee direction and perceivable characteristics. Both augmented and full reality show drawbacks (red) while the proposed multiple reality approach combines benefits of both, augmented and full reality.

increased visibility of a multiple-reality-enhanced cross-traffic assistance is only visible for the intended addressee. The information request then contains details on how perceivable characteristic are to change in order to direct the visibility enhancement. For a first proof-of-concept implementation see Figure 8.17.

6.7. Selecting Use Cases for Implementation

This chapter presented four novel application categories for next-generation Car2X networks. Each category was supplemented by a number of example use cases that further described the categories. Furthermore, two additional safety applications were presented. Within the scope of this thesis three use cases will be implemented for demonstration purposes. In order to select the top three among the ones presented in this chapter, a well-established selection process is applied. This process revealed these three use cases:

- On-demand traffic information system
- Context aware environmental displays
- Visibility enhancement for bicycles

The following describes the selection process and gives information on how this decision was reached.

6.7.1. The Analytic Hierarchical Process (AHP)

Among others, the Analytic Hierarchy Process (AHP) [Saaty, 1987] has a profound background in mathematics and psychology and was applied to numerous decision problems in the past. Results of the AHP are weights (w_i) and individual scores (s_i) for each criterion that was found relevant for alternatives of a decision. Each alternative is then assigned a total score S by the formula

Intensity of importance on an absolute scale	Definition	Explanation
1	Equal importance	Two activities contribute equally to the objective
3	Moderate importance of one over another	Experience and judgement strongly favor one activity over another
5	Essential or strong importance	Experience and judgement strongly favor one activity over another
7	Very strong importance	An activity is strongly favored and its dominance demonstrated in practice
9	Extreme importance	The evidence favoring one activity over another is of the highest possible order of affirmation
2,4,6,8	Intermediate values between the two adjacent judges	When compromise is needed
Reciprocals	If activity i has one of the above numbers assigned to it when compared with activity j, the j has the reciprocal value when compared with i	
Rationals	Ratios arising from the scale	If consitency were to be forced by obtaining n numerical values to span the matrix

Table 6.3.: Fundamental scale of the AHP reproduced from [Saaty, 1987].

$$S = \sum_{i=0}^{n} w_i s_i$$

such that

$$\sum_{i=0}^{n} w_i = 1$$

with n as the number of criteria. Alternatives in the context of this thesis are of course the use cases presented in this chapter. The AHP was also applied in project sim^{TD} for selecting functions that should be evaluated during the field operational test. Additionally, a direct selection of use cases accompanied the AHP. The interested reader is pointed to a summarized description of this process and results in [Schaaf et al., 2011]. Detailed information on criteria, weights, and scores can be found in one of the publicly available project deliverables [Passmann et al., 2009]. The AHP was mainly chosen because some of the criteria were already elaborated in the project sim^{TD} and there found to be adequate for a comparable decision problem.

The process of solving a decision problem with the AHP starts by generating a hierarchy of criteria that are considered relevant for the problem. In a next step priorities of the individual criteria are assigned by pairwise comparison of these (sub-)criteria. This generates priorities that can be synthesized to a weight of a criterion which is assigned to the parent node in the hierarchy. Pairwise comparison judges relative importance of

Criterion	Comment	Weight (w_i)	Rank
Pull Paradigm	The use-case demonstrates the Pull Paradigm well	0,275	1
Experienceability	Benefit of this use-case is perceivable and visible to	0,232	2
Vehicle Demo	Prospects of demonstration in test-vehicle are good	0,229	3
Implementation	Use-case can be implemented with low effort	0,125	4
Simulator Demo	Prospects of demonstration in simulator are good	0,066	5
Usability	The use-case is intuitive, can be learned and is usable	0,052	6
Market acceptance	Prospects of market acceptance are good	0,020	7

Table 6.4.: Weights calculated from comparison matrix.

two (sub-)criteria A and B on a "fundamental scale" defined by AHP (see Table 6.3), e.g. if criterion A is considered *extremely important* over B it is assigned a value of 9. This value is entered at position (i, j) in a matrix where i and j denote row and column of criteria A and B respectively. According to the author of AHP in [Saaty, 1987], this leads to a matrix of order n and $n(n-1)/2$ pairwise comparisons, where again n is the number of criteria. The resulting matrix is reciprocal and values on the diagonal are always 1 because it always contains criteria compared to themselves (see also Table 6.3).

6.7.2. Applying the AHP to Select Use-Cases for Implementation

Without giving mathematical details of the AHP here, literature derives weights of assigned priorities by calculating the principal and normalized eigenvector of the matrix [Saaty, 1987]. With this local scale, the priority of the parent is weighted resulting in a global scale. Depending on the depth of the hierarchy tree this has to be repeated until all levels of sub-criteria have contributed to the parent criterion.

Before the next section applies the AHP to the decision problem of which use cases to implement in this thesis, a note about consistency: Consider a comparison matrix where a_{ij} denotes the preference of criterion i over criterion j and a_{jk} the preference of criterion j over criterion k. Then one would expect the value for a_{ik} to be $a_{ij}a_{jk}$. However, in practice this is not always the case, leading to inconsistency in the matrix. The AHP measures this phenomenon by the consistency ratio (CR). Details on how CR is calculated can be found in [Saaty, 1987] but it should be noted at this point that the

Category	#	Name	Score
1 on-demand	**1.1**	**On-Demand Traffic Information System**	**0,946**
	1.2	Car2X-Enabled Charging Station for Electric Cars	0,698
	1.3	Intelligent Start-Stop	0,348
	1.4	Right of Way Conflict Resolution	0,423
2 environmental displays	2.1	Environmental Displays Optimized for Viewer Perspective	0,733
	2.2	Virtual Agents for Environmental Displays	0,684
	2.3	**Context-Aware Environmental Displays**	**0,925**
	2.4	Multiple Reality Environmental Displays	0,634
3 social	3.1	C2X Like Me Button	0,614
	3.2	Eco-Score Communities	0,444
	3.3	Ad-hoc Rides	0,457
	3.4	Linking Environmental Displays and Social Networks	0,758
4 personalized	4.1	Majority Decisive Environmental Displays	0,535
	4.2	Visual Bookmarking in Environmental Displays	0,722
	4.3	Live Video Feeds	0,686
	4.4	Microscopic Parking Lot Assistance	0,631
5 additional Push	**5.1**	**Visibility Enhancement for Bicycles**	**0,922**
	5.2	E-Bike Combined Battery and Communication Module	0,877

Table 6.5.: Final scores according to the AHP. Selected use cases are boldfaced.

AHP considers a comparison matrix consistent if CR is .10 or less.

With applying the AHP to the selection problem "Which of the presented use cases are most appropriate for demonstrating the results of this thesis?" seven criteria were found to be relevant. Note that for simplification only one level of hierarchy was used. Furthermore, calculations were made with the help of a freely available Excel sheet that implements one level of the AHP[1].

Some of the seven criteria were taken from the use-case selection procedure in the project sim^{TD} but specific ones for this thesis were added, e.g. how well a use case demonstrates the Pull Paradigm. The complete list is shown in Table 6.4 along with resulting criteria weights. As one would expect, use cases demonstrating the Pull Paradigm are rated higher than, for example, the criterion "market acceptance". It was also found, that experienceability and (specific to this thesis) an implementation in a test vehicle is important. These three criteria will contribute most to a decision. The comparison matrix and individual scores for criteria on the use-case level can be found in Appendix A. The CR for the comparison matrix was calculated as .068 which is below the threshold of .1 and therefore the result can be considered consistent according to the AHP.

The result after assigning scores to each criterion on the use-case level is shown in Table 6.5. Again, detailed results can be found in Appendix A. By far, the on-demand traffic information system (score: .946), context aware environmental displays (score: .925), and visibility enhancement for bicycles (.922) scored highest and were selected in order

[1]http://bpmsg.com/new-ahp-excel-template-with-multiple-inputs/

to demonstrate the results of this thesis. The reason for this selection lies in the judgment of the three main criteria (vehicle demo, pull paradigm, and experienceability). Hardware resources at DFKI include two test vehicles and a 70-inch environmental display. This explains the high grades for these use cases. Although the on-demand traffic information system cannot be shown with more than two cars, experienceability is considered high. Availability of the environmental display also led to a high ranking of all use cases where such displays are involved. Implementations are described in Chapter 8.

7. PADE: Simulating and Deploying Bidirectional Car2X Applications

7.1. Structure of PADE

In this chapter, the Car2X platform for application development and evaluation (PADE) is designed and implemented. It constitutes a software platform that supports development of bidirectional Car2X applications and contributes to the practical part of this thesis. Furthermore, it is well-integrated in the established technology of the Push Paradigm and thus is also supported. Hence, it addresses Research Question 4 as defined in Section 1.3. The architecture of PADE is based on the layered composition of the Car2X network stack, but greatly enhances it by adding additional software modules. The modules that contribute to the layers can be deployed either individually or in arbitrary combinations depending on the intended application. Since this thesis considers human factors and the specific requirements of the underlying communication technology in a combined way, PADE modules are present on each layer except PHY and MAC. The MAC and PHY layers are taken from the existing stack because no requirements for the Pull Paradigm were identified in Chapter 4. The modules of PADE are:

- Application Module
- Abstraction Module
- Presentation Module
- Messaging Support Module
- LGHS-CD Module
- Simulation Module
- Communication Module

The **Application Module** provides programming interfaces for implementing compatible Car2X applications in PADE. The framework connects to the abstraction module and delivers messages and events from the network to the application. Furthermore, methods for building and sending messages are available.

The **Abstraction Module** introduces a new layer in the Car2X network stack that abstracts from the underlying runtime environment. C2X PADE supports the development of applications that either run in simulation or are integrated into a test vehicle.

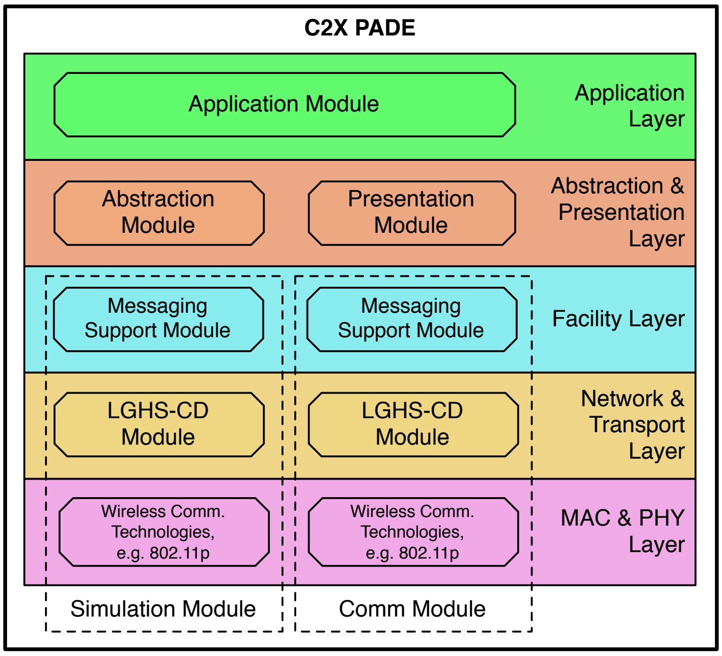


Figure 7.1.: PADE's layered architecture. It introduces a novel *abstraction layer* and enhances established layers of the Car2X network stack in order to support Pull applications.

Both runtime environments are by no means compatible in terms of timing, messaging formats, protocols, and APIs. The abstraction module of PADE provides a common C2X message format to the application layer and maps it to the requirements of the simulated or integrated facility layer respectively. The simulation module is based on discrete simulators, e.g. a simulated clock advances time in discrete steps, whereas each time-step is possibly not in real time. However, non-simulated wireless communication is obviously executed in real time. This affects how the clock is implemented in the application depending on the runtime environment.
Networking and transportation protocols also differ and need to be abstracted for the application. The same applies to APIs to both runtime environments. The abstraction layer of PADE provides these features.

The **Presentation Module** realizes a framework which is able to model, build and integrate Human Machine Interfaces for automotive applications. It is specifically tailored to Car2X-based applications but has also been successfully applied to other automotive applications (cf. Chapter 8). Besides the actual framework, it specifies a workflow for HMI design and implementation which has been constantly improved in several Car2X research projects (see also Chapter 8). Its full potential in the field is exploited when several independent running applications compete for limited resources on the HMI. It then behaves as a coordinator assigning HMI resources and on-time presentation slots to the applications while not overloading the display at the same time. To do this, it chooses appropriate "display strategies" for requested presentations. Display strategies are predefined in the "presentation model" of the application that requests the presentation. The definition of these presentation models is part of the workflow process for the HMI part of an application. For presentation management, the module instantiates a priority concept and considers the current context before changing the UI state. The basic implementation uses the current HMI state but can be exchanged for other plugins when additional information is required, e.g. cognitive load of the driver [Endres, 2012]. The presentation module of PADE mainly addresses Research Question 5 as defined in Section 1.3. Note that the focus is on *presenting* results of Pull requests rather than managing the input (cf. Section 4.1). The presentation module is published in [Castronovo et al., 2013b]

The **Messaging Support Module** is integrated into the facility layer and provides a new message type that is introduced for Car2X Pull Applications. For established Car2X networks, the message types Common Awareness Message (CAM) and Decentralized Environmental Notification Message (DENM) were presented in Section 2.2.2. As discussed, they are tailored to safety applications and their applicability to interactive applications of the Pull paradigm is limited. Therefore, a new message format that reflects information pulled from the ad-hoc network is introduced: Requested Environmental Message (REM). The messaging support module integrates REMs into the existing message types, thus ensuring compatibility of the facility layer for Pull applications. The messaging support module communicates with the abstraction module, enabling the abstraction layer to provide communication facilities to the applications. In Section 7.5, the structure of REMs and their relation to other message types is presented in detail.

The **LGHS Module** implements LGHS and LGHS-CD developed in Chapter 5 and integrates them into the network layer of the Car2X stack. The integration into the Scalable Wireless Ad-hoc Network Simulator (SWANS) is presented in Section 7.6. Furthermore, an obstacle model is developed in order to be able to consider the influence of buildings on wireless communication. LGHS-(CD) are realized as routing algorithms and replace the default routing strategy in SWANS. They therefore fulfill the requirements of the Pull Paradigm on the network level. The implementation of LGHS-(CD) addresses Research Question 1 (see Section 1.3).

The **Simulation Module** is one of the two runtime environments of PADE. It integrates four different simulators and builds upon the Vehicle Simulation Runtime Infrastructure (VSimRTI). The topic of simulating Car2X networks was thoroughly discussed in Section 2.2.3 and VSimRTI was identified as the most generic software to integrate different simulators. The coupling of simulators was a mandatory requirement, since the complex relations in vehicular ad-hoc networks cannot be modeled by only a single simulation. Besides traffic, network, and environment simulator, PADE also includes a driving simulation in order to be able to address effects of Car2X Pull applications on drivers. Here, OpenDS, the Open Source Driving Simulator, that is currently developed

at DFKI is used. Up to a certain number of simulated cars, the simulation module is able to simulate in real time depending on the computing power of the PC and the complexity of the scenario (around 50 cars on a 2.2GHz Intel processor with 4 GB of RAM running Windows 7). The integration of OpenDS is discussed in detail in Section 7.7. The simulation module addresses Research Question 3 (see Section 1.3).
The module combines three different simulators in real time for evaluating applications before running them in a test vehicle. Using a state-of-the-art 802.11p simulator and a microscopic traffic simulation, this module is able to deliver precise and realistic results even on large networks. Since Pull applications are more focused on the user, this module integrates an advanced driving simulation, opening the possibility to test effects of novel Car2X based applications on drivers while considering the specific limitations of vehicular ad-hoc networks. The wireless communication is transparent to the application, which offers the possibility of transferring the application to the vehicle with minimal effort.

The **Communication Module** operates on the lower layers and realizes the communication with other units in a real vehicular ad-hoc network. It offers a framework for developing Car2X-based applications and common functionality for sending and receiving messages over the ad-hoc network. It is built upon the Car2X stack implementation of NEC and conforms to established standards of the ETSI and Car2Car Communication Consortium [Festag et al., 2009]. As for the Presentation Module, the Communication Module has also been successfully applied to Car2X Push applications.

Developing a novel Car2X application using PADE involves modeling and implementing the requirements of the application module and (if desired) presentation module. The developed application can then be executed either by the simulation or by the communication module. Thanks to the abstraction module, no adaptations for switching between runtime environments are necessary. In the following Sections each layer of PADE and its modules are explained in detail, while Chapter 8 continues by showing how example applications are developed using PADE.

7.2. Presentation Layer

Figure 7.2 outlines the information flow within the Presentation Module. Before a presentation can start applications send "presentation requests". These are validated against the corresponding Presentation Model. If this validation succeeds a presentation task is created and stored in an internal database. The Presentation Module is time-based and has no situation knowledge. This means, that an application specifies start and end time of a presentation and does not send spatial information for presentation requests. Presentation tasks which have reached their start time are handed to an internal data structure by the presentation timer. Validator and presentation timer are able to notify applications on the current state of their presentation request. According to the current HMI context and priorities of the presentation tasks, possible conflicts are resolved (cf. Section 7.2.6). If a task has been selected for presentation, available "display strategies"

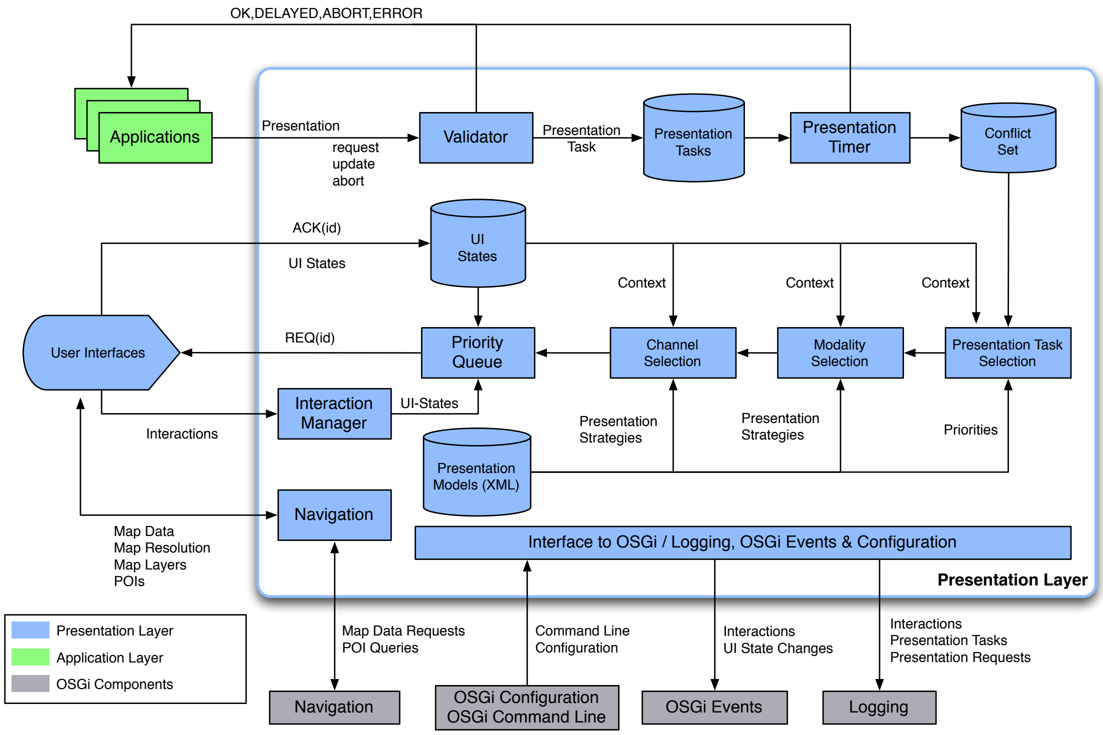


Figure 7.2.: Presentation Module of PADE

defined in the model are considered in order to select presentation modality and -channel. The user interface updates its state constantly after receiving new tasks. Synchronized communication ensures that the Presentation Module is always assessing the current display context.

7.2.1. Presentation Models

A Presentation Model contains essential information for a successful presentation on the HMI. The following list reflects all information contained in a Presentation Model:

- ID
- Name
- Public interface
- Private parameters
- Display strategies
- Automotive user interface markup language (AUML)
- Inheritance

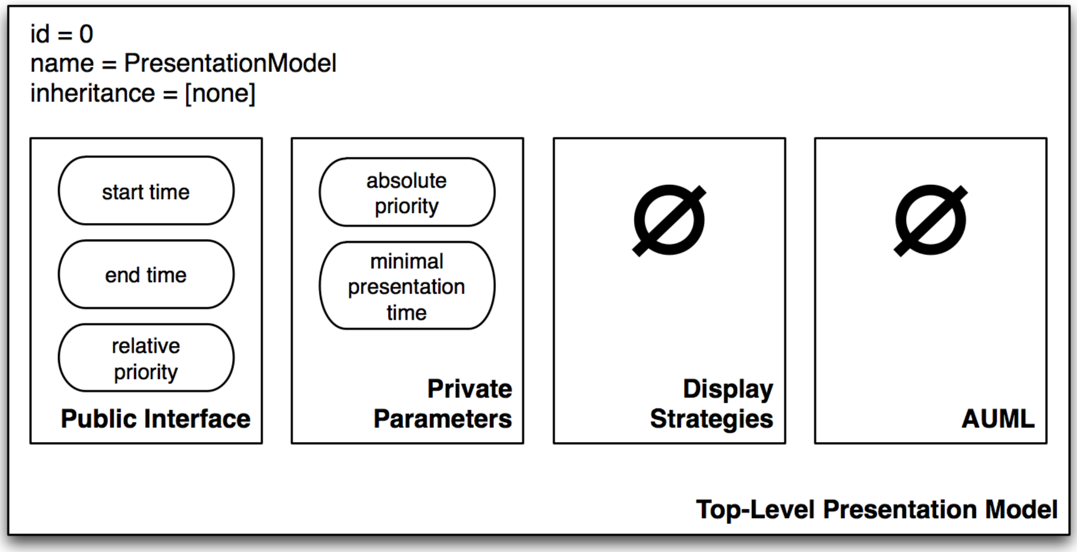


Figure 7.3.: Top-Level Presentation Model.

The ID of a presentation model is used as an identifier by the Validator during a presentation request. With this ID the presentation model to which this request is validated against can be looked up. The public interface defines mandatory and optional parameters as well as their valid domains in order to request presentations. The different types of parameters are explained in detail in the next section. Private parameters can be specified and are used to process presentation requests internally. For example, these parameters can be used to implement the priority concept explained in Section 7.2.4. Presentation models specify at least one, but in most cases more, so-called "display strategies". Such a strategy occupies certain, possibly limited, resources on the HMI for this presentation, e.g. visual, haptic or auditory channels. The automotive user interface markup language (AUML) defines the communication protocol for displaying presentation tasks on the user interface(s). In the automotive domain more than one physical interface may exist: Besides common displays, there are warning lights, indicators for current speed, fuel level, etc. A powerful feature of presentation models is the possibility to build up hierarchical structures. They inherit information (parts of the public interface, display strategies, etc.) from higher-ranking models.

The top-level presentation model contains the information shown in Figure 7.3. All others in the system inherit from this model. This reflects the fact that start and end time as well as model ID of a presentation are mandatory parameters when creating a Presentation Task. Relative priority is modeled as an optional parameter (cf. 7.2.2). Instantiating this top-level presentation model results in an empty presentation since neither a display strategy nor AUML is specified. Figure 7.4 shows a simplified version of presentation model inheritance for Pull applications in the category of local danger

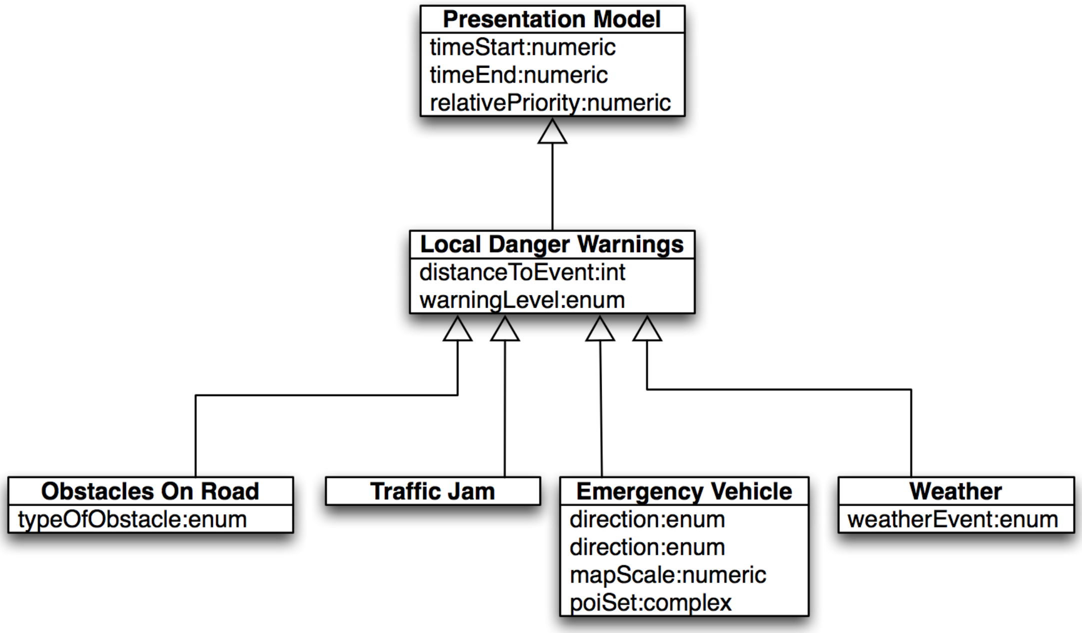


Figure 7.4.: Example inheritance for C2X Push applications in the category of local danger warnings. Inheritance is marked by arrows. Simplified version displays only the public interface.

warnings. Every local danger warning inherits from a higher-ranking model parameters distanceToEvent and warningLevel, which are common for all four presentation models. When necessary, the models specify application-specific parameters which are not shared among other models. Note that in this simplified picture AUML is not included.

7.2.2. Parameter Types

Presentation models offer the possibility to specify basic parameter types or complex types similar to the type structure in higher programming languages. Every parameter specifies its name, type, valid domain (either as list or domain range) and an optional default value. Parameters without default value are mandatory when requesting a presentation. This Sections describes possible parameter types and their syntax.

Basic Types

The available basic types are:

- Numeric
- Boolean
- String

- Enumeration

Numeric parameters are either integer or float values. The definition includes their domain range and an optional default value. String parameters are textual values. A list of valid values can be specified. If empty, no constraints for parameter values are applied. As for other parameters, a default value can be specified. An enumeration is a list of allowed elements either of type String or Numeric. Boolean parameters can either be true or false. An example taken from the reference implementation for a C2X based traffic sign assistant is shown below in order to clarify the syntax. In this implementation, presentation models are stored in XML files.

```
<Parameter name="tsValidLanes" type="numeric" min="1" max="7" default="1" />
<Parameter name="tsCode" type="enum" values="10100,10200" default="10100" />
<Parameter name="distanceBar" type="boolean" default="false" />
<Parameter name="tsAddonContent" type="string" values="" default="" />
```

Complex Types

Complex types cannot be represented by basic types and/or need internal processing before the presentation task can be displayed to the user. A practical example would be an application which visualizes the Car2X neighborhood table on the display. Since the application possesses no knowledge about the dimensions of its assigned drawing area it hands the raw GPS coordinates to the Presentation Module. Before the presentation task is displayed or updated, the Presentation Module converts the GPS coordinates into screen coordinates. The identifier `complex` followed by a fully qualified class name indicates this parameter type. In this class several methods have to be implemented:

- `isParameterValueValid()`: This method is dynamically called by the Validator and checks whether the submitted value for this parameter is correct according to the presentation model. It returns true on successful validation, false otherwise.
- `expandVariables()`: This method is called when a task has been selected for presentation. It does the necessary computation on parameter values and modifies the AUML by enriching the current (possibly transformed) value of a parameter to the AUML.
- `cleanup()`: This method is called on system shutdown in order to free up memory, stop threads cleanly or perform other necessary cleanup tasks. This is especially necessary when the Presentation Module is integrated into a larger system where restart or shutdown is triggered by a framework and memory/thread management is mandatory.

An example definition of a complex type in a presentation model is given as follows (taken from the reference implementation for an approaching emergency vehicle warning):

```
<Parameter name="mapScale" type="complex:de.simtd.impl.system.hmi.
presentationmodel.parameter.F214MapScale" default="500" />
```

7.2.3. Presentation Tasks

After a successful presentation request the application receives a unique identifier for the created presentation task. This identifier can be used by the application to update and cancel the presentation at any time, even if it is not yet started. Note that method `expandVariables()` described above is only called when a task was selected for presentation. This avoids unnecessary computation of parameter values.

7.2.4. Priority Concept

The priority concept is an essential tool for resolving presentation conflicts. When developing C2X applications using PADE, every presentation model is assigned a "priority range" on a global scale in the interval of $(0, 100)$, and a default priority. The application can vary its current priority within its designated range using the parameter `relativePriority`. This parameter is inherited from the top-level presentation model (cf. Figure 7.3) and has a domain range of $(-100, 100)$, which means the absolute priority is not exposed to the application. This approach allows to change the priorities of the models during testing without the need to adapt the application. A relative priority of 0 maps to the default priority of the model; all other values are mapped using Figure 7.5. The priority range reflects the fact that applications may have higher or lower priority depending on the current situation, e.g. a congestion warning coming up in 3 km might be lower in priority than a broken down vehicle right in front of the EGO car. Within the concept of PADE the assessment of such situations is the responsibility of the application, since the Presentation Module relies on time and not on spatial information. On the other hand, applications do not possess any knowledge of presentation task realization in the HMI. This separation of situation assessment and display management is an essential concept and maintained throughout the modules of PADE. Priority values and ranges that were applied in sim$^{\mathrm{TD}}$ can be found in Figure 8.1.2.

7.2.5. Automotive User Interface Markup Language (AUML)

PADE's Presentation Module uses the automotive user interface markup language (AUML) in order to communicate with the user interface. AUML is an XML-based communication protocol which is tailored to automotive user interfaces and is specified within display strategies in presentation models. One powerful feature of AUML is the concept of *back-referencing*: Variable values that are defined in the public or private parameter section of a presentation model can be backreferenced by writing '$[variable name]$' for XML element names, attribute values or CDATA. These variables are then substituted by the current value of a parameter when a task has been selected for presentation. The method `expandVariables()` of a complex parameter implementation (cf. Section 7.2.2) receives a reference to the individual node in the AUML tree, manipulates it and returns the AUML. After all variables have been substituted by their current values, the AUML is sent to the registered handler and presented to the user.

$abs \leftarrow def$
if $(rel > 0)$ **then**
  $abs \leftarrow abs + \frac{rel*(max-def)}{100}$
end if
if $rel < 0$ **then**
  $abs \leftarrow abs - \frac{rel*(min-def)}{100}$
end if
if $abs > max$ **then**
  $abs \leftarrow max$
end if
if $abs < min$ **then**
  $abs \leftarrow min$
end if

Figure 7.5.: From relative to absolute priority. Legend: abs = absolute priority, rel = relative priority, min = minimum priority according of priority range, max = maximum priority of priority range, def = default priority.

High-Level Addressing

AUML uses high-level addressing in order to communicate with the UI. It moves parts of the implementation effort to the UI side since it needs to be parsed and interpreted by the specific application. The advantage of high-level addressing is that any existing UI framework can be used and full bandwidth of the UI language can be exploited. Of course, backreferencing is also available.

High-level addressing recognizes channels and layouts. A channel is a communication interface with the user, e.g. display or haptic. Furthermore, with the help of channels a communication interface can be modeled in further detail and address a sub-part of this interface, e.g. a specific area on a display. Both components of the Presentation Module, UI and the underlying framework, retrieve a taxonomy of available channels at startup. A channel (or parts of it) can be referenced within the display strategy section of a presentation model. An example is given in Section 7.2.7.

Layouts are UI implementations of applications. They can be addressed by their name or fully qualified class names and are loaded dynamically by the UI framework which delivers a subtree of the AUML, the display strategy, to the UI application code. Here, application-specific parsing is required to instantiate and update the UI. Currently, high-level addressing is implemented in Adobe Flash and allows development of additional presentation models with low effort on UI side. This concept has been used for sim^{TD}, where over 30 presentation models were implemented. Additionally, the HMI was included in several demonstrators at DFKI (see Chapter 8). The flexible approach of high-level addressing helped keep overhead for additional user interfaces low.

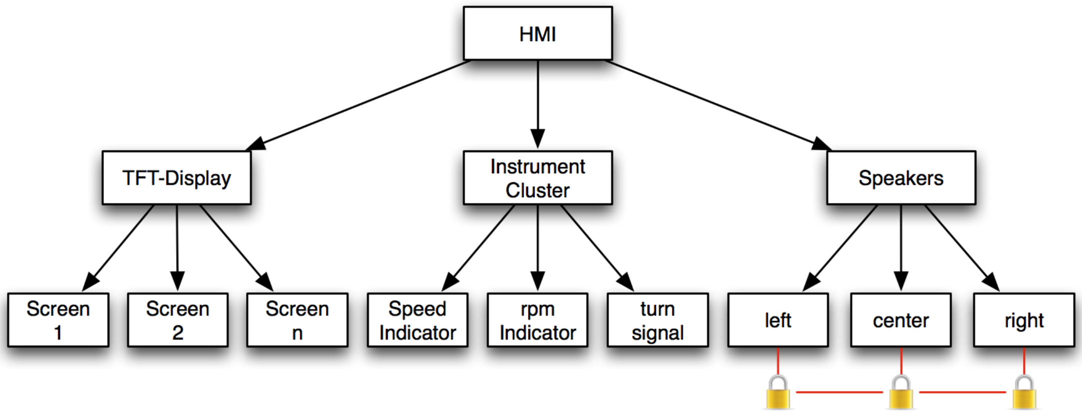


Figure 7.6.: A simple Allocation Tree. Usability constraints are symbolized by locks: In this example, the available speakers must not be allocated by different presentation tasks at the same time.

7.2.6. Conflict Identification & Resolution

Since all applications in the system run independently from each other and have no knowledge of UI resources, presentation conflicts are possible. The following components ensure the most efficient usage of the UI: display strategies, HMI allocation trees, and priorities which are assigned to each application (cf. Section 7.2.4). Among application-specific parameters, display strategies define the UI resources an application occupies during presentation. Within one strategy several presentation channels might be defined, e.g. visual and auditory, at the same time. The strategies are defined in the presentation model and are ordered in ascending priority.

An *allocation tree* is a hierarchical model of all available HMI resources (see Figure 7.6). Every leaf in this tree can be allocated by exactly one presentation task. Using *usability constraints* it is possible to restrict certain allocations. Taking the example tree in Figure 7.6, it is not possible to allocate the speakers to different presentation tasks as this would degrade comprehension and thus the overall usability. UI resources can be referenced within display strategies. Identifying presentation conflicts then boils down to comparing the current allocation of the HMI and utilized UI resources of a display strategy which has been selected for presentation. When no conflict is detected, the selected task is presented using the first display strategy defined. Conflict resolution is done by comparing priorities of both tasks and switching the lower prioritized presentation to its alternative strategy, if available. In case no conflict resolution can be identified, one of two fallback strategies is employed:

1. Abort presentation and notify application
2. Postpone presentation by x ms and notify application about the delay, where x is a configuration option

Applications can specify which fallback strategy should be used when creating a presentation task. Note that this can also be overridden by a configuration option.

7.2.7. Example Presentation Model

Listing 7.1 shows an example presentation model for a Push application (cross-traffic assistant) as it was defined for the project sim^{TD} (see Section 2.1.3). Note that this model is based on another one which defines parts of the public interface for a whole use-case category. It specifies two display strategies. The first allocates two resources on the HMI: "mainScreen" and "symbolArea", the second only the resource "symbolArea". In sim^{TD} the upper part of the display is divided into six parts where additional information can be displayed. Using the wildcard character "*" within the channel element, any of these six positions is valid for this strategy.

Listing 7.1: Example presentation model for sim^{TD}-function cross traffic assistant

```xml
<PresentationModel name="CrossTrafficAssistant" appId="224" basedOn="HF22.xml">
 <OsgiParameters>
  <!-- 0 = Left, 1 = Right -->
  <Parameter name="direction" type="enum" values="0,1" default="0"/>
  <!-- 0 = Car, 1 = Motorcycle, 2 = Truck -->
  <Parameter name="vehicleType" type="enum" values="0,1,2" default="0"/>
 </OsgiParameters>
 <InternalParameters>
  <Parameter name="globalPrioritisation" type="int" min="60" max="100" default="90" />
  <Parameter name="minPresentationTime" type="int" min="0" max="max" default="3000" />
 </InternalParameters>
 <DisplayStrategies>
  <DisplayStrategy id="0">
   <displayTask>
    <channel name="GUI:mainScreen">
     <layout name="F$appId$">
      <warningLevel>$warningLevel$</warningLevel>
      <direction>$direction$</direction>
      <vehicleType>$vehicleType$</vehicleType>
     </layout>
    </channel>
   </displayTask>
  </DisplayStrategy>
  <DisplayStrategy id="1">
   <displayTask>
    <channel name="GUI:symbolArea:*">
     <layout name="F$appId$">
      <warningLevel>$warningLevel$</warningLevel>
      <direction>$direction$</direction>
      <vehicleType>$vehicleType$</vehicleType>
     </layout>
    </channel>
   </displayTask>
  </DisplayStrategy>
 </DisplayStrategies>
</PresentationModel>
```

7.3. Application Layer

From a developer's point of view, the application module of PADE provides the necessary interface and methods for implementing Car2X applications and provides access to the the ad-hoc neighborhood. This section describes their structure on a technical level and gives programmers an understanding of how to use the PADE platform in practice.

The application module uses the programming language JAVA v1.7. Implementing a new application using PADE involves *extending* the `Application` class and *implementing* the `ApplicationInterface`. The PADE framework then calls method `receiveMessage()` when a message has been received. Alternatively, application developers can implement the `timerCall()` method which is called every 100ms by PADE. Within this method an application should check whether new, unprocessed messages have arrived in the queue. The `Application` class provides access to an `operationHandle` which wraps all methods for managing messages and sending them over the ad-hoc network. It is instantiated depending on which runtime environment is executed (real or simulated).

A PADE application also has access to the Presentation Module introduced in Section 7.2. Using its reference to the presentation service, applications can create, update, or cancel presentation tasks. Furthermore, the application can subscribe to input events and UI state changes in order to respond with appropriate actions.

7.4. Abstraction Layer

In order to abstract from the underlying runtime environment, this layer wraps necessary data structures and methods for application development and maps them either to the API of NEC C2X SDK or VSimRTI. They include access to the neighborhood table, methods for message composing, support for sending messages, and information about the EGO vehicle. The relation of abstraction components to other PADE components is shown in Figure 7.10.

7.4.1. Composing Messages

PADE currently supports three types of messages. Existing CAMs and DENMs have been modeled on the platform level, are compatible with established standards and can easily be mapped to data structures available in VSimRTI and NEC C2X SDK. The new type of Requested Environmental Notification Message (REM; see the next Section) reflects Pull requests. Currently, no such message exists in either of the underlying runtime environments. In order to make this type of message available to applications, it has been wrapped in the payload of already-existing message formats.

For the simulation part, content of a REM is encapsulated in the payload of the DENM data structure provided by VSimRTI. This is only possible because fields of this message type are not fixed to the ones specified by ETSI. The API of VSimRTI offers the possibility to transport arbitrary payloads when DENMs are used. On the receiving vehicle, the REM is extracted and pushed back to the application, e.g. `receiveMessage()` is called (see Section 7.3).

When running inside a vehicle, data structure `DataRequestGUCMessage` offered by the NEC C2X SDK is used to transport REMs. Also here, arbitrary payloads can be specified and unicast communication is possible. As for the simulation, the receiving vehicle checks the type of payload and pushes the REM to the application by calling `receiveMessage()`. This abstracts sending and receiving procedures for the application.

Alternatively, PADE also integrates the network stack that comes with the operating system (see Figure 7.10). Therefore, applications can also be built upon consumer wifi (802.11a/b/g/n) or cellular networks (UMTS/LTE).

From an implementation point of view, this is done on the application level. Normally, one would expect this to be on the network layer, but since neither VSimRTI nor NEC C2X SDK is open source, it needed to be realized on the upper layers.

7.4.2. Node Properties

Properties of the EGO vehicle are stored in the `UnitData` class. It provides access to vehicle data such as global position, address of the node, and vehicle type (car, motorcycle but also road side unit). The abstraction layer of PADE retrieves necessary information from VSimRTI and NEC C2X SDK in order to provide a common interface to these properties from the application level.

7.5. Facility Layer

On the facility level the messaging support module introduces a new message type to the Car2X network stack that reflects Pull requests. This requested environmental message (REM) is derived from DENMs but contains additional slots. Furthermore, two types of REMs are introduced: REM-req and REM-rep. The first is generated by the application running on a car *requesting* information from the ad-hoc network; the second is sent by the node holding the requested information as a reply. The individual fields of REM-req and REM-rep are shown in Figure 7.7. Fields that were adopted from the DENM can be looked up in Section 2.2.2.

7.5.1. Requested Environmental Message (REM)

The additional fields of a REM are the following:

- Header
  - `messageID`: Unique id of this REM

  - `messageSequenceNumber`: In order to support requests and replies which exceed the maximum size of a REM (typically 1024 bytes) the content is split over several messages. The message sequence number identifies a message in this sequence.
  - `messageCount`: The total number of messages in this sequence.

- Management
  - `maxHops`: The maximum number of hops allowed for this request / reply. Receiving nodes may drop this message if *maxHops* exceeds a given threshold.

- Request / reply
  - `Sender`: Reference to sending node. This is for example used by the application to identify a previously issued request. This field is copied from REM-req to REM-rep by the node holding the information. Note that it is also renamed recipient for clarity.
  - `previousSenders`: Stores local information on previous hops that can be used to trace back the route to the requesting car. It can also be used to approximate the dynamics of the ad-hoc network.
  - `requestContent` / `requestReply`: Payload of this request or reply, respectively.

- Location
  - `senderPosition` / `recipientPosition`: The sender of the request stores its position in a REM-req. This field is then copied to REM-rep by the receiving application. This information can be used to locate the recipient.

Implementation note: REMs can be built by applications in order to send and receive Pull requests. Normally, not all fields would be exposed to the application layer. For example, messageSequenceNumber and messageCount are handled by lower layers in traditional network stacks. But since NEC C2X SDK is a closed-source product, implementation was done on the application layer. This leads to minor timing overhead since messages need to be pushed up to the application layer for processing.

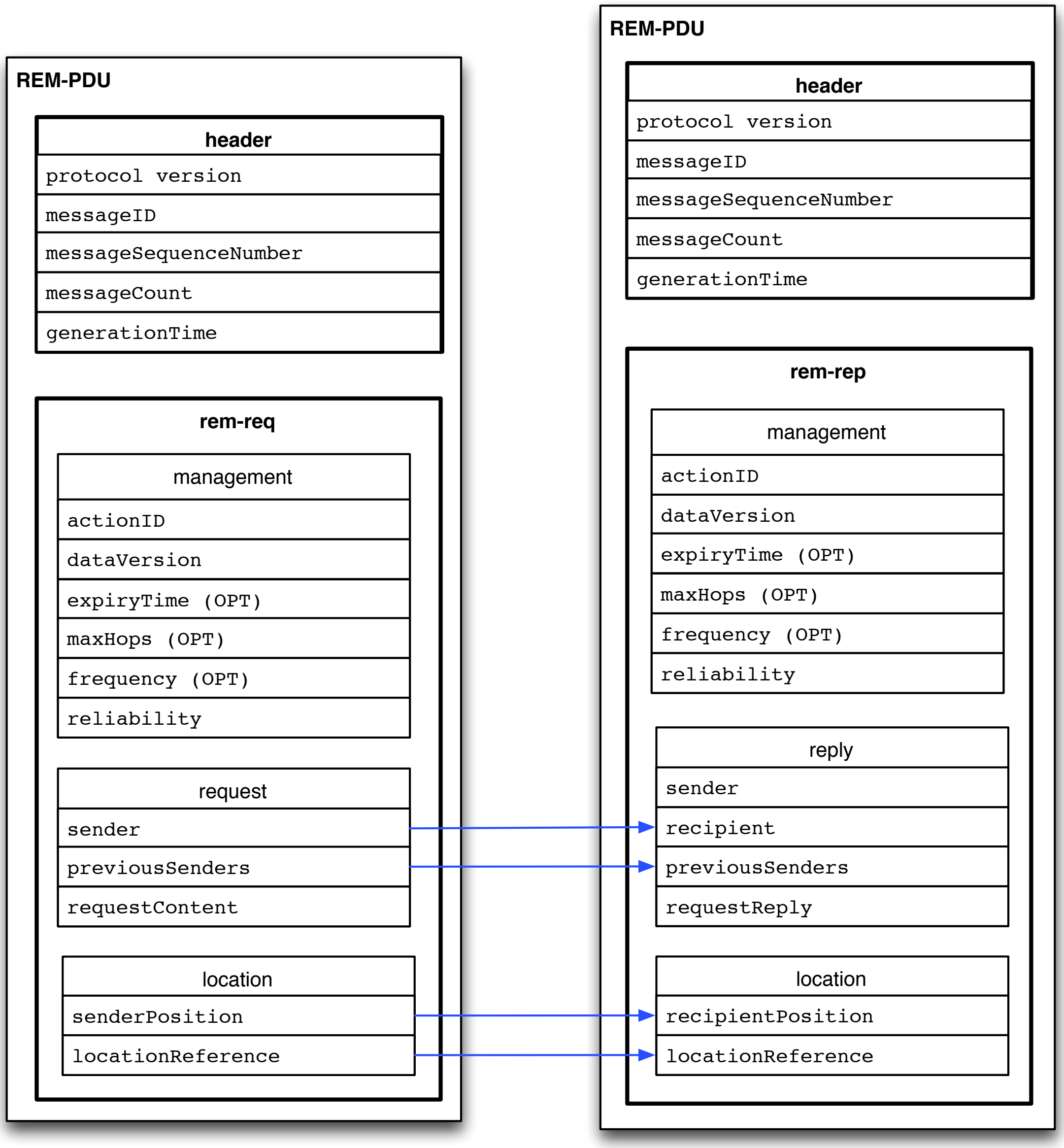


Figure 7.7.: Requested Environmental Message (REM). Left: Request; right: Reply. Arrows indicate fields that are copied from REM-req to REM-rep by the node holding the requested information.

7.6. Network Layer

This section describes the implementation and integration of LGHS-CD as routing algorithm in SWANS as well as necessary modifications that were made to internal components of the network simulator. A visual representation is shown in Figure 7.8. Here, the internal composition of a network node in SWANS is presented, as well as its communication with the added interfaces that are described in the following subsections. Together, these components constitute the LGHS-CD module of PADE.

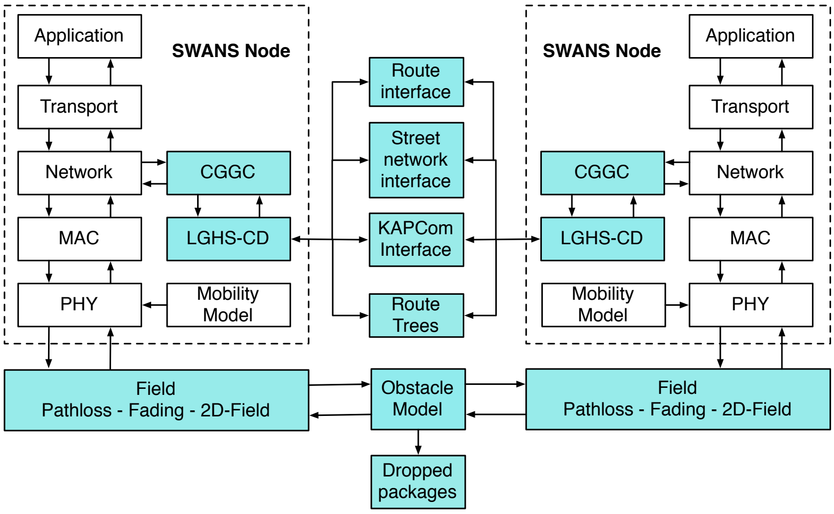


Figure 7.8.: Integration of LGHS-CD into the Scalable Wireless Ad-hoc Network Simulator. Blue-shaded components were added or modified in order to implement the LGHS-CD module of PADE.

7.6.1. LGHS-CD Integration into the Network Simulator SWANS

Although SWANS provides interfaces to implement customized routing algorithms via jist.swans.route.RouteInterface, they are not considered by VSimRTI in the simulation module. Thus, only the internal routing protocol of SWANS, namely Cached Greedy GeoCast (CGGC) is possible in combination with VSimRTI. The code of SWANS is open source but VSimRTI uses a custom version that contains enhancements by the University of Ulm that add support particularly for vehicular ad-hoc networks. Fraunhofer FOKUS agreed to provide the source code of this custom version. Therefore, a hook could be added to CGGC such that the LGHS-CD module is instantiated instead of executing CGGC. Calls to methods of CGGC are then simply forwarded to the LGHS-CD module.
Another modification was made to the jist.swans.misc.Protocol class in order to collect and send beaconing information from other nodes. This information was also used by LGHS-CD to build up the neighbor list, as access to VSimRTI methods from within SWANS is restricted.

7.6.2. Satisfying Requirements of LGHS-CD

LGHS-CD relies on certain information regarding dynamic graph operations, static graph topology, and edge assignments. In this context, dynamic graph operations are movements, insertions and deletions of cars by the simulator. The static graph is formed by the underlying road network and edge assignments are cars traveling on roads. In order to fulfill requirements of LGHS-CD, an interface to the street network was built that is used by the simulation module, namely the traffic simulator SUMO. There are no access restrictions for LGHS-CD, in order to support the mandatory shortest path computations during the individual phases of the algorithm.
A similar interface was built for routes that vehicles are assigned to during the simulation. These are used for predicting future topology of the dynamic graph, e.g. future positions of cars. SUMO provides this information via the traffic control interface (TraCI). A special focus has been set on access restrictions, e.g. queries to future routes return null. Furthermore, the route interface checks the neighbors of the calling SWANS node, since only queries to nodes within the transmission range are allowed. These requirements result from the functions defined in Section 5.2.

7.6.3. Modeling Obstacles

Another important addition to SWANS is an obstacle model that was implemented in order to consider the influence of buildings on radio signals. Concrete walls and other materials influence propagation of radio signals significantly. The obstacle model within the LGHS-CD module performs a line-of-sight check whenever a packet is sent to another node. If an obstacle is identified, the packet is dropped. This is an efficient approach to include the effect of buildings in the simulation. In reality, radio signals may be reflected by different materials, e.g. glass, and therefore be able to be detected by the recipient. However, a more realistic model would reach the limits of real-time simulation and increase the complexity. A line-of-sight check puts more restrictions on the simulation than reality. Obstacles were not considered by SWANS and therefore were added as new functionality. The data source of the model, e.g. buildings, was derived from OpenStreetMap and stored in an internal database.

7.6.4. Integrating KAPCom for Information Management

LGHS-CD uses KAPCom, developed in the context of the PhD thesis of Michael Feld [Feld, 2011] and serves as an information and knowledge management component. The LGHS-CD module of PADE uses the KAPCom interface depicted in Figure 7.8 for storing and retrieving data. KAPCom provides a generic network interface for communication. Note that for performance reasons, all communication between KAPCom and SWANS nodes is routed through one interface, thus using one connection. When simulating a large number of nodes within SWANS, load and allocated memory increase significantly. For the same reason, only one KAPCom instance is used for all nodes. However, information stored by a specific node can only retrieved or modified by this node. This was realized by adding categories in KAPCom's ontology, each representing a specific node. During startup of SWANS, the LGHS-CD module registers each node added by the simulator in KAPCom below the previously generated *allVehicles* category and submits a unique ID for this node. This ID is then referred to during simulation when the node accesses its data. Besides its ID, the node stores the creation time, data object, area of validity, and position where the information was retrieved. The LGHS-CD module running on the node can then process REMs by accessing KAPCom, e.g. respond to an information request either positively or negatively.

7.6.5. Implementing Route Management of LGHS-CD

Section 5.4.4 described how LGHS-CD handles dynamic graph operations in order to keep routes to information sources up to date. Recall that route maintenance is an important part of LGHS-CD since the graph is highly dynamic and routes are therefore constantly changing. Route maintenance is always necessary when

- a non-member node enters the route,
- a member node leaves the route, and
- the sequence of member vertices changes along the route.

The LGHS-CD module is notified in these cases because of the modifications made to jist.swans.misc.Protocol that ensure transfer of beaconing information to the module. All three events trigger route maintenance operations described in Section 5.4.4 and occur very frequently, most likely with every advance of the simulation clock. It is therefore essential to apply efficient data structures that offer cheap basic operations such as search, insert, and delete. The most necessary operation is search: During a simulation run crumb values decrease constantly and requests trigger a check for the information within the node (search). Operations insert and delete are only required when information is stored locally (insert) or a crumb value drops below the threshold (delete).
Basic operations are always needed when crumb values of a route are increased or decreased. All of these operations can be achieved within $O(log\, n)$ using state-of-the-art red-black trees [Cormen et al., 2009] and are referred to as *RouteTrees* (see Figure 7.8).

7.7. Simulation Module

The simulation module uses the existing vehicle simulation runtime infrastructure (VSim-RTI) that was identified in Section 2.2.3 as the most promising approach for this thesis. It provides possibilities to couple arbitrary, discrete simulators. The individual simulators for traffic and communication were also identified in Section 2.2.3 as SUMO and

JiST/SWANS, respectively. Furthermore, OpenDS [Math et al., 2012], the open source driving simulator developed at DFKI, is added to the simulator community.

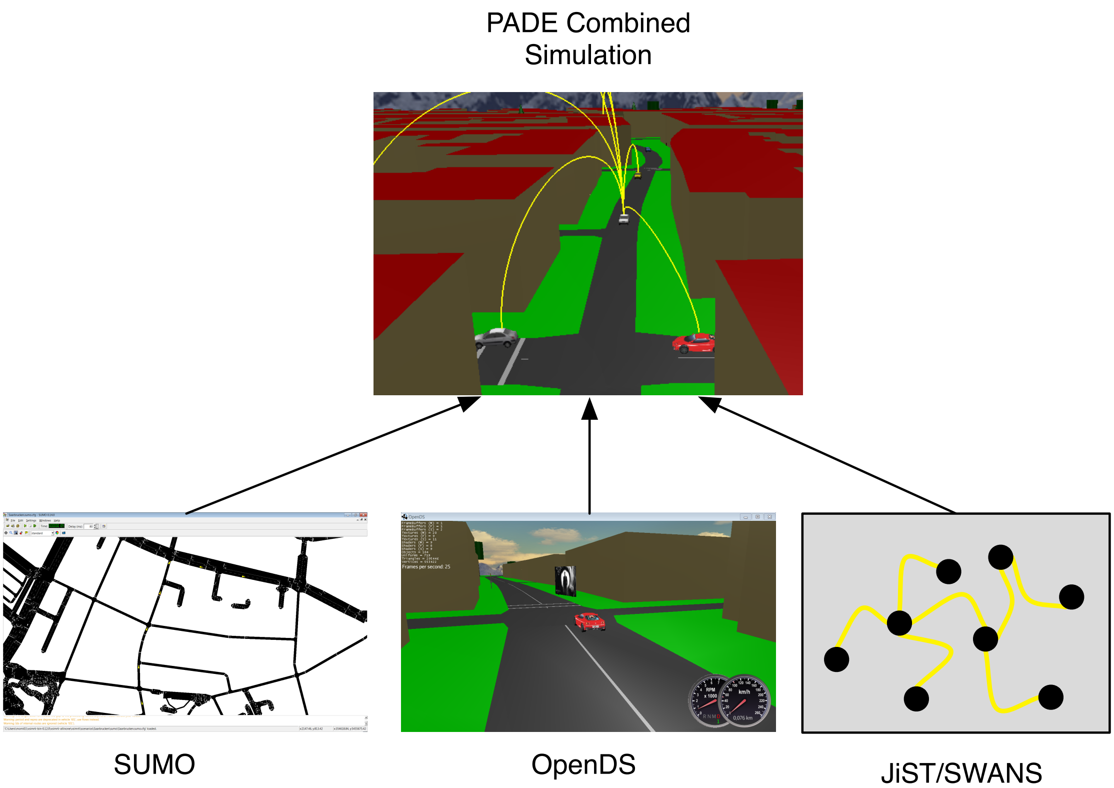


Figure 7.9.: Screenshot of a combined simulated scenario with OpenDS, SUMO, JiST/SWANS in PADE. Wireless communication is indicated by yellow arcs.

7.7.1. Adding OpenDS to VSimRTI

VSimRTI uses the High Level Architecture (HLA) for simulator coupling [Symington and Morse, 2000]. In this architecture, two components are necessary for additional simulators: *Ambassador* on the simulator and *Federate* on the runtime infrastructure. Ambassador subscribes to message types it needs to receive from the runtime infrastructure. Figure 7.10 shows the basic information flow between these two components in the case of OpenDS and VSimRTI. Every vehicle added by SUMO gets visualized in OpenDS. Its movements are controlled by SUMO. Furthermore, OpenDS Ambassador receives Car2X messages that are sent and received by JiST/SWANS. They are used to visualize communication between vehicles for demonstration purposes (see Figure 7.9). OpenDS Ambassador creates a node in JiST/SWANS for the EGO vehicle the user is driving in OpenDS, which is then considered during communication simulation.

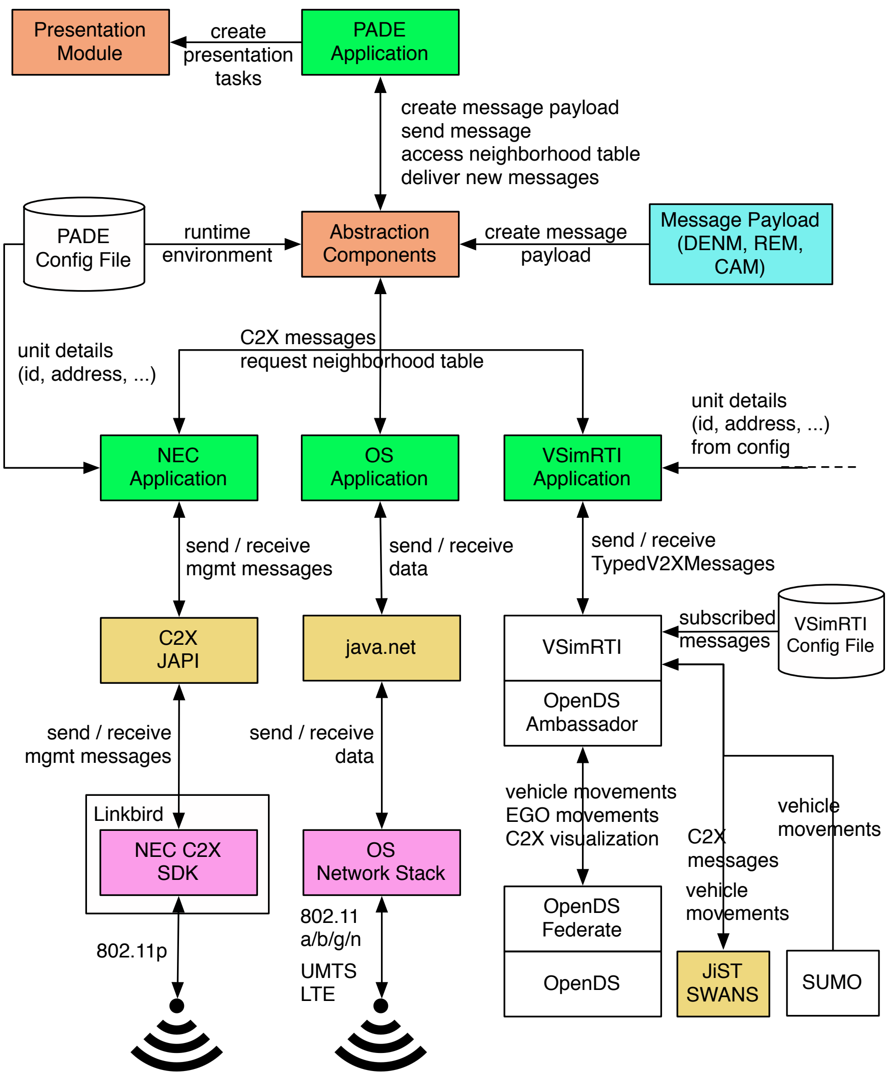


Figure 7.10.: PADE components and data flow.

7.8. Communication Module

The communication module is a NEC Linkbird MX, a hardware platform for C2X prototyping. It is equipped with two 802.11p cards, each consisting of one control and one communication channel. The operating system is Debian Linux with NEC C2X SDK installed, which realizes the lower part of the communication stack [Festag et al., 2009]. Applications are able to communicate using a Java API (JAPI) provided by NEC. The API specifies a low-level protocol based on management messages in order to retrieve essential information from the Linkbird, e.g. EGO position or neighborhood entries. The abstraction layer of PADE exposes these functionalities to PADE applications but creates a common format for every runtime environment.

8. Developing PADE Applications

PADE and parts thereof has been applied to several systems, from demonstrators, to use in several bachelor's/master's/PhD theses, to large-scale field operational tests. This chapter gives a detailed description of the individual projects and the role of PADE within the systems.

8.1. Push Applications in the sim^{TD} Field Operational Test

The presentation module of PADE has been successfully applied to a large-scale field operational test. A fleet of 120 Car2X-enabled vehicles from all major German car companies participated in this field test around and within the hessian metropolitan area of Frankfurt/Main. For a detailed description of the sim^{TD} project, please see Section 2.1.3.

Over a period of 3 years, 34 presentation models were discussed with members of the consortium, designed and implemented. Each member was responsible for one or more C2X Push applications. The following sections group the use cases in larger areas of interest and presents the most important graphical realizations as well as an abstract view of the presentation models and their inheritance. This section also presents interesting aspects from PADE's perspective that came up during presentation model development.

8.1.1. General HMI Concept in sim^{TD}

For graphical representation on the touch display and to incorporate all sim^{TD}-functions, it was decided to create three virtual screens: navigation, main, and options screens. The navigation screen showed a map and guidance instruction. Furthermore, interactive Push applications were accessible via five buttons. The main screen was primarily used for showing the highest-prioritized functions. Using the options screen, the driver could set different options related to the HMI, start a driver survey, communicate with the sim^{TD}-control center, and inform other drivers about obstacles on the road. The three virtual screens were arranged side-by-side and could be accessed using virtual buttons (see Figure 8.1).

Most presentations request the main screen as their default presentation strategy. As on the main screen only one application can be displayed at a time, a "symbol area" was created, which was included on top of each screen. Hence, six additional slots for condensed presentations were available in the allocation tree and were part of alternative presentation strategies. Priority ranges were defined and optimized for each application before the field test started. The presentation module of PADE that was presented in

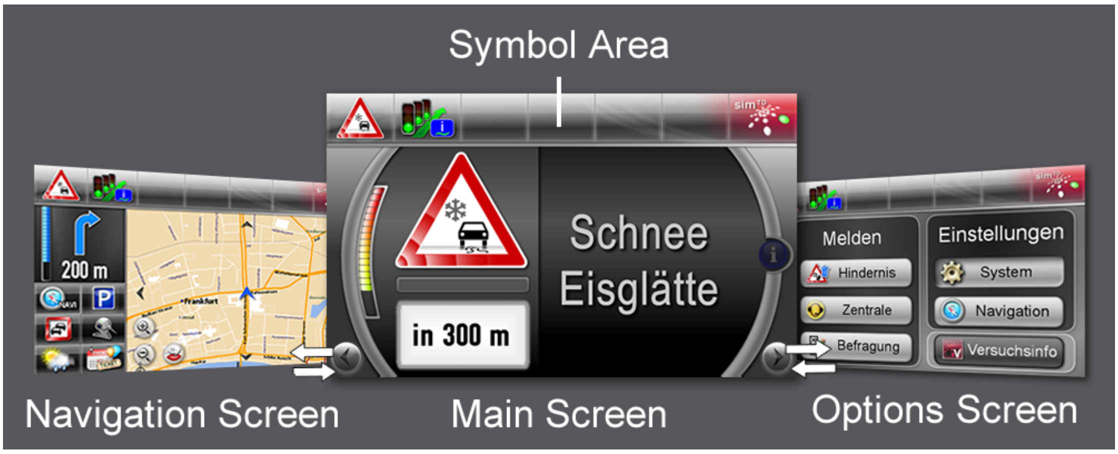


Figure 8.1.: Illustration of the sim^{TD} HMI depicting three virtual screens and the symbol area.

Section 7.2 has been successfully applied to the sim^{TD} project. It was running reliably in all 120 cars, and 400 test drivers experienced it on a daily basis.

A special focus was placed on consistency of presentations over all implemented presentation models. All warning functions were assigned two urgency levels. These were supported by audio output. Furthermore, the second level was supported by red coloring as well (see Figure 8.1.5).
Another consistent element was the presentation of spatial distance and progress. Several sim^{TD} functions needed to display the approach to relevant traffic situations or progress within, e.g. obstacles. The spatial approach was always represented by a multi-colored segmented *vertical* progress bar on the left of the main screen (see Figure 8.1). Progress within roadwork and validity areas of traffic signs was always indicated by a blue segmented *horizontal* progress bar (see Figure 8.4, lower picture). A detailed analysis of the spatial approach and progress visualizations in the automotive context was published in collaboration with colleagues at DFKI at INTERACT 2013 [Mahr et al., 2013].

8.1.2. Priority Distribution of sim^{TD} Presentation Models

The priority concept of PADE introduced in Section 7.2.4 was applied to all 34 presentation models of the sim^{TD} system. Priority ranges and default values were initially determined in a workshop with the respective project partners for the relevant functions. During the integration phase of the project on the test track in Friedberg, the initial priority values were redefined in an iterative process. Efforts over two years ensured a harmonized priority distribution of the sim^{TD} functions. Finally, test drives in the city of Frankfurt, motorways, and rural roads verified correct priority management of the sim^{TD} HMI before the field operational test. The final priority distribution of the most relevant functions is shown in Figure 8.2.

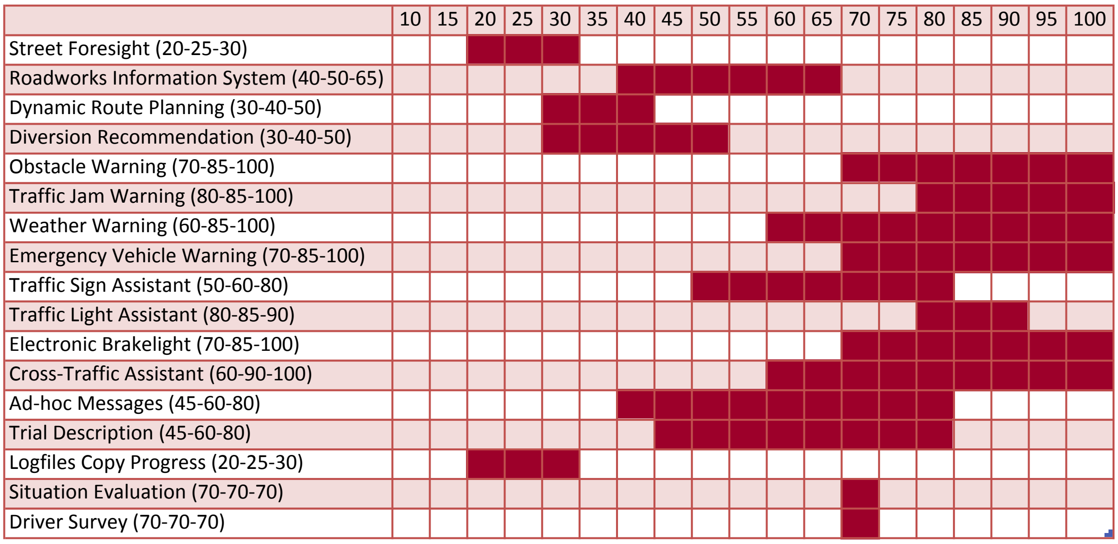


Figure 8.2.: Priority distribution of the most relevant sim^{TD} presentation models as used in the field operational test.

8.1.3. Traffic Flow Information

Street Foresight

The Push application Street Foresight gathers beaconing information from the close vicinity and calculates traffic flow. More interestingly, it combines the information into a self-experienced map. The display handles two different presentation types: motorway and city, where the latter shows the EGO vehicle centered, whereas the former moves it to the bottom of the screen. Using this Push application the driver was able to assess the traffic close by and possibly adapt her route with more accurate information than by using publicly available traffic information services.

The presentation model inherits directly from the top-level model but overwrites parameter presentationType which is used to indicate city or motorway display. Furthermore, the presentation model of this function is particularly interesting since it shows the strong separation of the presentation and application layer in C2X PADE. The complex parameters *edgeObjects* and *carObject* contained information on how to draw the map on the screen. PADE's concept provides that there is no situation assessment on the presentation layer, e.g. presentations are planned on the basis of time and not on spatial information. This implies that lower layers use spatial but not time or display information. Therefore, complex parameters were used to do the necessary conversion between application and presentation layer formats. The function submitted Universal Transverse Mercator (UTM) coordinates; this is 2-dimensional Cartesian coordinate system that divides the surface of the earth into sixty zones, each 6 degrees longitude wide. Because only PADE's presentation layer possesses information about drawing areas on

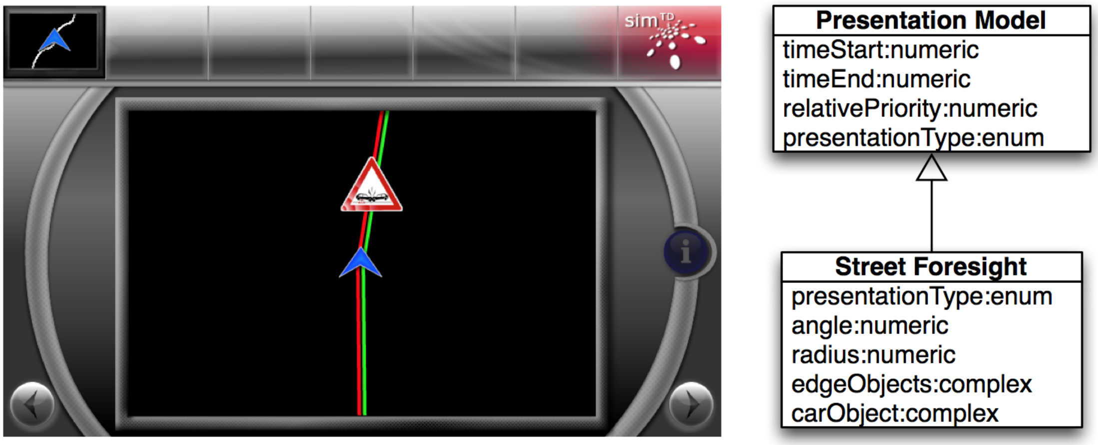


Figure 8.3.: Presentation model and visualization of Push application Street Foresight in sim^{TD}.

the display, UTM was translated to screen coordinates within the `expandVariables()` method (see Section 7.2.2). The coordinate translation was necessary on every update from the function. Since street foresight was always present while driving, this method needed special attention in terms of efficiency. Together with the function developers and accountable developers from the sim^{TD} OSGi framework, this method was highly optimized to ensure a resource-efficient conversion. The optimization was continuously monitored by detailed profiling before the field test which confirmed that no significantly higher load was added to the CPU of the car PC.

Roadworks Information System

Figure 8.4 shows the presentation model and visualization of the Roadworks Information System. The function was represented in two views: The first visualized the approach to the work area by counting down remaining distance and showing an overview of (the initial) work area's geometry. The second showed progress within the work area. Geometry was displayed in detail on the right of the screen and was moved consistently with the car's progress as it advanced through the work area. Microscopic traffic flow information on lane level were symbolized as colored arrows. Moreover, general information was available using the blue info button to the right. The two views were realized by overriding parameter presentationType from the top presentation model.

The function calculated geometry dynamically from the first n cars that drove through the work area. Their vertical acceleration was used to derive the necessary information. Data of other cars were also used to determine an estimated time for passing through the work area.

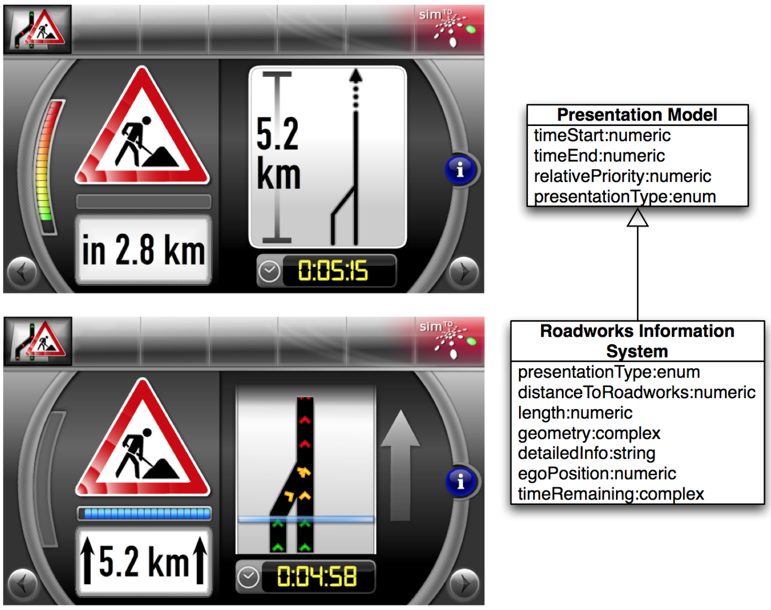


Figure 8.4.: Presentation model and visualization of Push application Roadworks Information System in sim^{TD}.

Roadworks Geometry Description Language During the discussion with the application developers involved, a descriptive XML-language for mapping raw vertical acceleration data from cars to an easyly-understandable graphical visualization was developed (contained in the complex parameter “geometry”). It consisted of an ordered set of “tiles” where each tile was assigned a set of attributes, e.g. length in meters. The following types of tiles were known:

- straight
- lanes split
- diagonal
- lanes merge

Figure 8.5 shows a graphical representation of the defined tile types.

Each tile features origin and destination in the range from 0 to 5. In order to illustrate this, Figure 8.6 shows a graphical representation of a “straight” tile with different origins. The blue lines were not visible on the HMI and are used for simplification only. Obviously, the straight and diagonal tile each feature one origin and destination, while the split lane has one origin and two destinations. Analogously, merged lanes have two origins and one destination.

Using the introduced description language it is furthermore possible to superimpose tiles in order to model more complex geometries. It enables, for example, drawing types

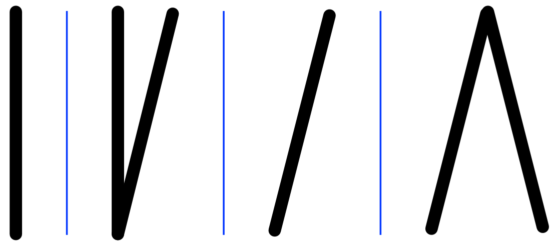


Figure 8.5.: Types of tiles that are recognized in the roadworks geometry description language. From left to right: Straight, lanes split, diagonal, lanes merge.

Figure 8.6.: Straight tile with starting point 0 (left) and 1 (right).

straight and panning in one tile by assigning the same id to two tiles. Every XML element contains an attribute `length` that describes the length in meters of a tile. The Attribute `trafficFlow` is used to render traffic density information of this element. In Figure 8.4 this is shown as green, yellow, and red arrows in the lower picture.

The method `isParameterValueValid()` was used to perform a consistency check of the submitted roadwork geometry. For example, it is not possible to set destination x of tile n-1 to a value where no origin in tile n follows. This would severely affect plausibility of the function and potentially distract the driver. In case an inconsistent geometry is submitted, the presentation model displayed a straight line on the HMI. The same representation was used in case the function was not able to acquire sufficient data over the network.

Within the method `expandVariables()` of complex parameter roadworksGeometry the XML tree of the geometry description was attached to the AUML and sent to the UI afterwards. Figure 8.7 gives an example of a roadwork geometry described in XML and the corresponding graphical representation.

8.1.4. Traffic Flow Control

Traffic Light Assistant

The traffic light assistant is an implementation of the Push use case described in Section 3.3.2. During discussions with the responsible sim^{TD} partner Volkswagen, it has been decided that the following features should be included in the presentation model of the

```
<roadworksGeometry>
  <tile id="0" type="straight" length="50" origin="2" trafficFlow="3" />
  <tile id="1" type="panning" length="30" origin="2" destination="1" trafficFlow="3" />
  <tile id="2" type="straight" length="30" origin="1" annotation="" trafficFlow="3" />
  <tile id="3" type="splitting" length="50" origin="1" destination="0_1" trafficFlow="3_2" />
  <tile id="4" type="straight" length="50" origin="0" trafficFlow="2" />
<tile id="4" type="straight" length="50" origin="1" trafficFlow="3" />
  <tile id="5" type="combination" length="50" origin="0_1" destination="1" trafficFlow="3_2" />
  <tile id="6" type="straight" length="50" origin="1" trafficFlow="3" />
<roadworksGeometry>
```

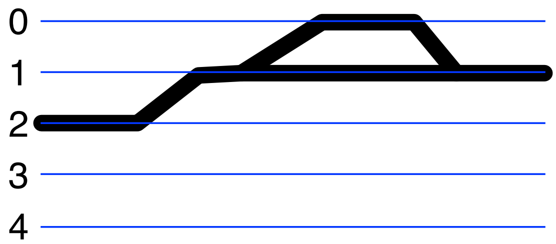


Figure 8.7.: XML structure of a roadwork geometry with corresponding graphical representation below.

function:

- Up to three traffic lights that display either recommended speed to cross while green or the remaining red time.
- Intersection geometry: either X or T intersection.
- Most probable path of the car, if available. It was determined by exploiting navigation data or the turn signal.
- Static "green arrow". In Germany, some traffic lights can always be driven past when turning right. The function had the capability to represent this in the graphical realization.

Figure 8.8 shows this presentation model in simplified form. It is directly derived from the main presentation model. In addition to this function in the category traffic efficiency, a warning functionality was included. In case of an impending red light violation, an appropriate warning message was displayed (Figure 8.8, bottom). Warning tones and coloring were consistent with the concept for local danger warnings.

8.1.5. Local Danger Warnings

The function group of local danger warnings was divided into four sub-functions:

- Obstacles: warning before objects blocking the road, like lost cargo or animals.

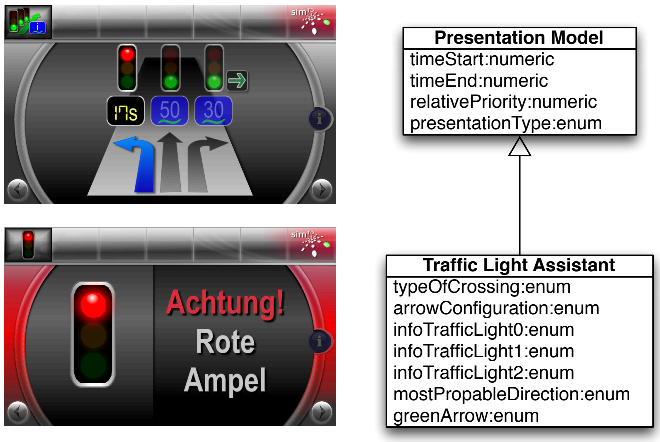


Figure 8.8.: The traffic light assistant in sim^{TD}. Additionally, a warning function in case of an impending red light violation was included.

- Traffic Jam: warning before stationary cars on the motorway.
- Approaching emergency vehicle warning.
- Weather, wind, ice, and snow warnings.

The division into these four groups was also reflected when presentation models were designed. Inheritance was used to derive all four groups from a "local danger warnings" model that featured common parameters (warning level, distance to event). Each local danger warning then added its specific parameters, if any. The warning concept in sim^{TD} featured two urgency levels. In Figure 8.9 a broken down vehicle warning is shown at the info (left) and warning level (right). Together with additional red elements in the visual representation, two distinctive warning tones were played (cf. [Cao et al., 2009, 2010]). Additionally, spatial approach to traffic situations was indicated by text, e.g. in 300 m, and a colored, vertical progress bar (also see [Mahr et al., 2013]).

8.1.6. Local Danger Warnings Issued by Other Drivers

In addition to presenting local danger warnings in sim^{TD}, drivers were also able to notify other road users about broken down vehicles, pedestrians, animals, and obstacles on the road. These four options were available via a submenu on the options screen. After pressing one of the buttons a confirmation dialog was shown which disappeared automatically after a given time in order to minimize interactions. Technically, the OSGi whiteboard pattern was used to notify other applications about the interaction event and these, in turn, sent out DENMs to the Car2X neighborhood. This functionality of PADE's presentation module shows that it is not restricted to output but can also receive input from the user. Therefore, simple Pull requests can be realized as well (see Figure 10.1b).

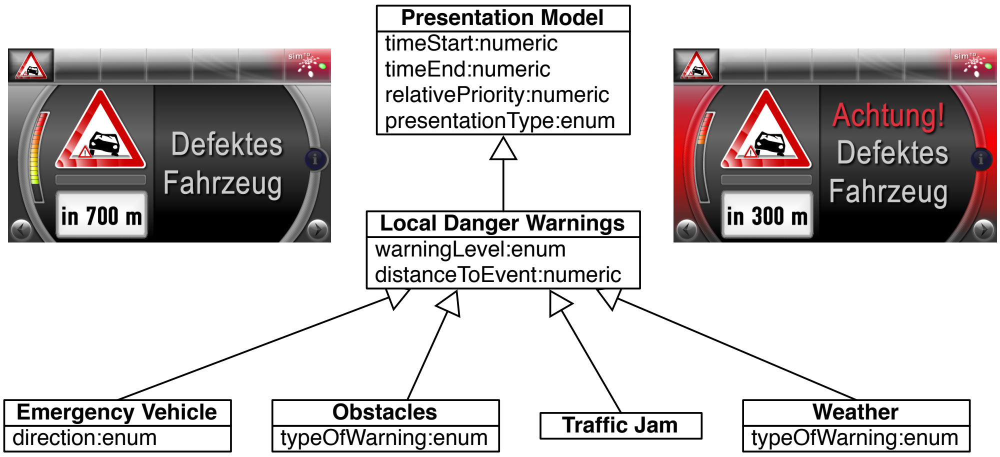


Figure 8.9.: (Simplified) presentation model and visualization of the function group of local danger warnings in sim^{TD}.

8.1.7. Driver Assistance

Traffic Sign Assistant / Warning

Implementation of the traffic sign assistant (see Section 3.3.1 for a description) in sim^{TD} included around 100 static and dynamic traffic signs as well as 60 additional signs. Using the parameter *tsCode* in the presentation model, the individual signs were identified. It was realized as an enumeration parameter and included the official main and sub-codes of the German road traffic provisions, e.g. main code 274 speed limit and sub-code 56 for a speed limit of 60 km/h. In order to include this in one parameter, the following formula was applied: main-code * 100 + sub-code. Analogous coding was applied to additional signs (parameter *tsAddonCode*).

Contrary to most other presentation models, the main display strategy of the traffic sign assistant was to use the symbol area. However, since a warning functionality also was included, the traffic sign assistant was able to give a hint as to the presentation strategy to use. This did not necessarily change the display strategy but informed the presentation planning component about the function's intentions. It could nevertheless be overridden by other running presentations. The traffic sign assistant was the only function for which this feature was enabled.

Following the consistent concept for visualizing spatial approach and progress in the sim^{TD}-HMI, the function was able to assign validity and distance information. Distance to a traffic sign was also presented as a vertical blue bar in the symbol area. Figure 8.10 shows a simplified version of the presentation model and graphical representation of the function.

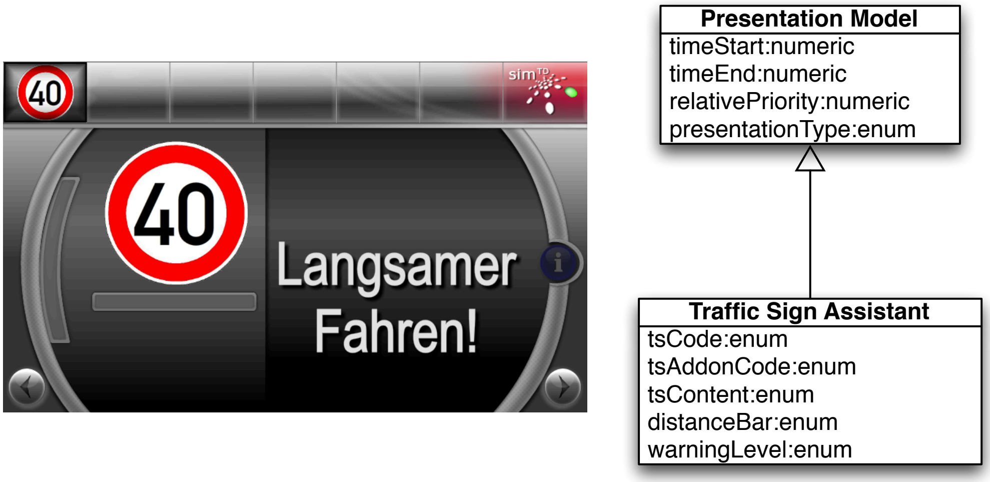


Figure 8.10.: (Simplified) presentation model and visualization of the traffic sign assistant implementation in sim^{TD}.

8.1.8. Location-Based Services

Location-based services in sim^{TD} provided access to information about parking, traffic, weather, webcams, and local events. By selecting one of these categories, point-of-interest icons were shown on the map. When selecting one of these POIs, detailed information about it was presented on the navigation screen.

From PADE's perspective this use case is particularly interesting because presentations were not triggered by an external event but by the user interacting with a POI on the screen. Three components of the sim^{TD}-system needed to communicate in order to realize these use cases: function, presentation module of PADE and navigation. Figure 8.11 depicts a graphical overview of these relations.

After system startup, the function downloaded POI information via the vehicle's UMTS connection. This information contained the position and its meta-information depending on the POI type, e.g. for a POI of category traffic, the cause, affected motorway and length (if applicable) were retrieved. The function then created POIs using the public navigation interface. However, using the navigation API it was not possible to store information other than location, name, and POI image. Therefore, presentation models in PADE were designed to represent the meta-data, e.g. length of a presented traffic situation. Using a unique POI ID created by the navigation API it was possible for the presentation module to access the elements later on.

When the user selected a use case of location-based services, the navigation API was queried for the specific POI layer, e.g. webcams. It returned a flat image that superimposed the base map (see Figure 8.12, left). This image contained all POIs that were

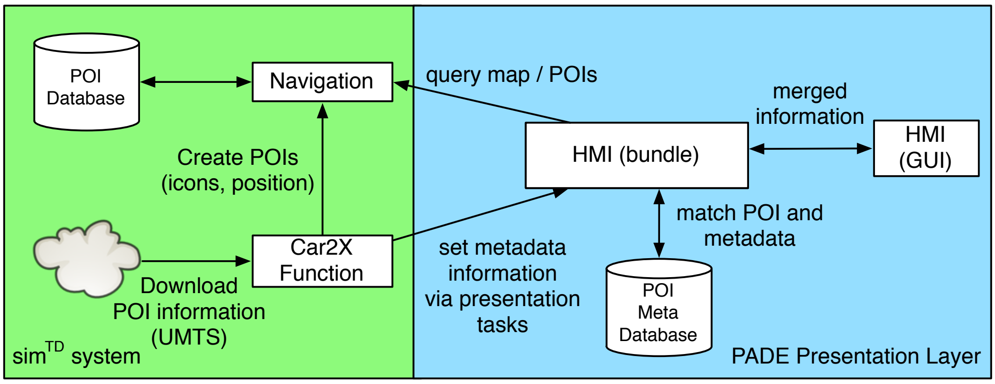


Figure 8.11.: Relations between PADE's presentation module and components of the sim$^{\mathrm{TD}}$ system that were established for location-based services.

created earlier by the function.

The user was now able to select one of the presented POIs. Since the navigation service returned a flat image, it was not possible to identify the requested POI. The navigation API was used to retrieve POI ids that were found below the x-,y-coordinate of the interaction. After querying the POI meta-database that contained information created after the system startup, the graphical representation of the AUML was shown on the display (Figure 8.12, right).

Figure 8.12.: Left: Transparent POI layer "webcams" placed on top of the base map retrieved from the navigation service. Right: Presentation task after interaction with a POI.

8.2. On-Demand Traffic Information System

In Section 6.7, the on-demand traffic information system was one of the three use cases that were selected for demonstrating results of this thesis. This section describes the implemented scenario. The particular implementation is called the advanced turning assistant and demonstrates the Pull Paradigm by requesting sensor data from another car. In this example, the situation is as pictured in Figure 8.13: The blue car approaches a

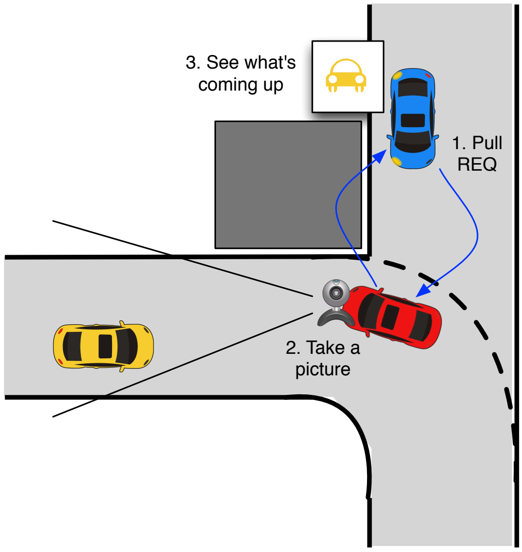


Figure 8.13.: PADE Demonstrator "Advanced Turning Assistant": Blue car requests still picture (1) taken by webcam in another car (2) in order to assess situations occluded from her perspective (3).

major road and is expected to yield. At the same time, the red car driving on the major road is turning into the road, but the view is occluded from the blue car's perspective. Using the advanced turning assistant, the driver of the blue car requests a still picture of the street that is not visible from her perspective from the Car2X neighborhood. The red car responds to the request and makes the driver of the blue car aware of the yellow car which is not in the view field of the EGO car.

From the perspective of LGHS-CD, the goal is to discover a vehicle that is in an optimal angle to the EGO car, e.g. in Figure 8.13 almost orthogonal. Therefore, the request contains position and heading of the EGO. The goal condition is checked in receiving cars by comparing whether their own position and heading is optimal compared to the received information. Recall that LGHS-CD uses LGHS with an approximation of $\tilde{v}_{goal}$ if no crumbs are available. In the example of the advanced turning assistant this approximation can be done by forwarding the request to vehicles that are close to the intersection in front of the EGO vehicle.

A special challenge during implementation was the size of the picture. Maximum payload of a REM is 1024 bytes. A picture using jpeg compression in VGA resolution is typically significantly larger in size. Therefore, image data had to be split over several packages. To transfer the requested image, around 50 REMs had to be sent. Therefore, additional fields were added to a REM. Every message contained its sequence number as well as the total number of messages in this sequence. The receiving application then issued a presentation task after all messages arrived, or after a time-out, only parts of the image. The request was considered to have failed if more than 50% of messages in a sequence were lost. When the first message was received an estimated arrival time of remaining packets was calculated. If this time was exceeded, the corresponding message was re-requested.

This message-splitting procedure was added to the PADE platform itself and is therefore also available to other applications. It is always used when the payload of a REM exceeds the maximum allowed size. Note that the procedure had to be implemented on the application layer since the NEC implementation of the Car2X network stack is closed source and there is no possibility to add this procedure to the network layer. The fragmentation and assembly of messages on the application level led to a minor decrease in performance but proved that the Pull paradigm is realizable by enhancing existing Car2X technology.

8.3. Interaction with Environmental Displays

In Section 6.7, the Pull use case "context aware environmental displays" (see Section 6.2.3) was selected as the second demonstrator for this thesis. In this particular implementation, the demonstrator combined several components developed at DFKI. The use case is described as follows: A driver of a car is waiting at a red traffic light while an environmental display shows advertisements for two different movies. The user's gaze on one of the movies is highlighted by the environmental display giving immediate visual feedback of what is in the focus of her attention. If desired, additional information can be accessed and a ticket reservation can be made by multimodal interaction. If the user asks for details about the movie, a trailer is started on the environmental display while audio is played within the car.

PADE's view on this demonstrator is presented in Figure 8.14. Other components involved (colored in gray) are:

- SiAMDP: A multimodal dialog platform developed in the DFKI project SiAM [Nesselrath and Feld, 2013] that is specifically tailored to automotive applications. It is built upon OSGi, a service platform for Java that enables modular application development.
- EyeVius: A system for precise real-time situation understanding. Among others, it uses an environment model and an embedded Tobii eye tracker to identify gazing targets of the user.

For the visual representation of the ticket reservation process, PADE's presentation module is used and connected via the OSGi service platform to SiamDP. Six additional presentation models were designed each of them representing a step in the ticket reservation process.

EyeVius makes the system aware of the movie that is currently in the user's focus of attention. It is connected as a *device* to SiamDP. On focus change, SiamDP notifies a C2X PADE application, which then generates a REM and sends it to the environmental display via 802.11p. The same applies when the dialog system asks to start the movie trailer. Audio is not streamed but started separately in the car. On the environmental display, another PADE application waits for REM requests and notifies the poster application to either start the movie trailer or to change the focused part of the poster.

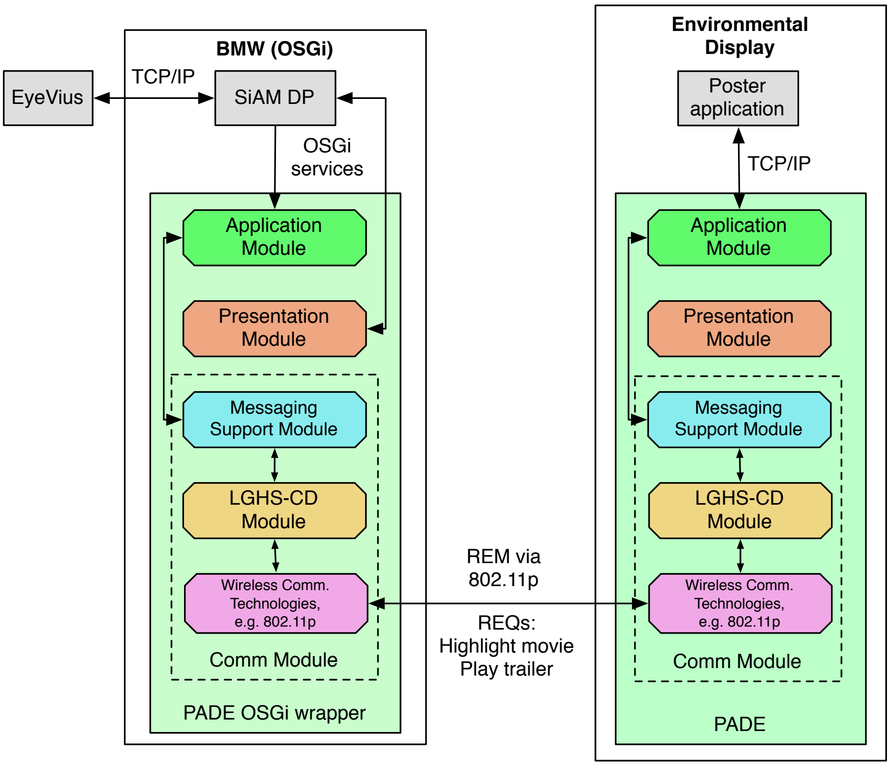


Figure 8.14.: Technical setup of the PADE demonstrator "Interaction with Environmental Displays".

8.4. Personalized Applications

The presentation module of PADE has been used by other work at DFKI in the context of speaker classification and adaption. It helped to demonstrate applications for adaptive automotive HMI and was chosen because of its flexible approach to presentation models. This section gives a short overview of developed applications and their context.

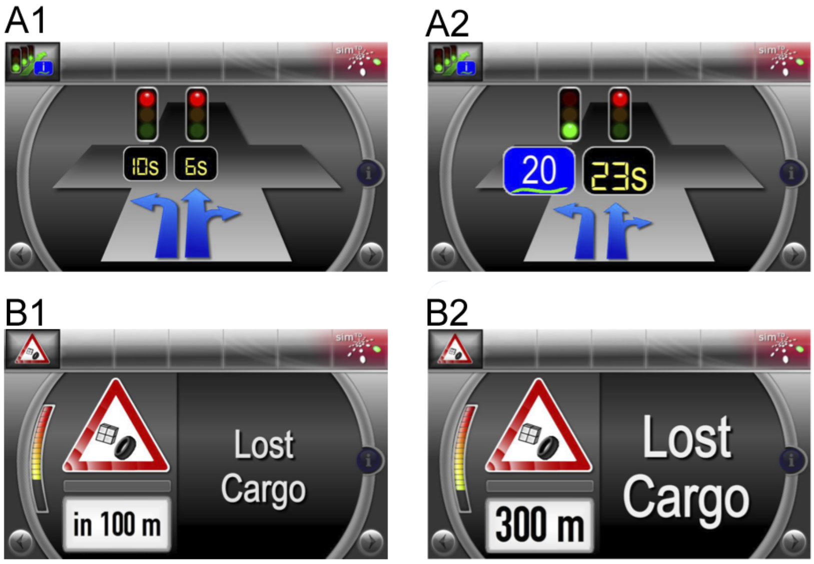


Figure 8.15.: Personalization of PADE's presentation module. A1/B1: Unmodified version; A2/B2: Modified presentation model for people with visual impairments. Image source: [Feld, 2011].

8.4.1. Adaptation of an In-Vehicle Information System

In [Feld, 2011] a speaker classification framework for non-intrusive user modeling is developed. In order to demonstrate the capabilities of the system, a series of systems have been augmented by personalization aspects. Among these, PADE's presentation module was enhanced in order to support perceptibility of graphical elements of the sim^{TD} functions. Therefore, presentation models implemented for the sim^{TD}system were augmented by parameters for personalization. The adapted AUML was then interpreted by the UI in order to create personalized visualizations of the presentation model. Three sim^{TD} functions were chosen for demonstration: obstacle warning, traffic light assistant, and internet-based services (parking information). Adaptations to the graphical UI included an additional layout for normal and impaired vision, e.g. larger font size for warnings and the traffic light assistant. Furthermore, parking information distinguished between female and male drivers. In the first case, parking spaces for women were highlighted.

Presentation models of the three functions were therefore enhanced by age class and gender of the user. Additionally, further parameters controlled the speed of the text-to-speech (TTS) synthesis and the voice of the system. Results are shown in Figure 8.15.

8.4.2. HMI Adaptation for People with Visual Impairment

Figure 8.16.: Personalization of PADE's presentation module. The presentation model has been enhanced by shapes in order to support colorblind people. Image source: [Majstorovic, 2013].

In [Majstorovic, 2013], PADE's presentation component is optimized for people with visual impairments. The system retrieves a user model from KAPcom which is a result of the speaker classification framework developed in [Feld, 2011] and changes the user interface depending on the disability of the driver. The focus is on color blindness, but low vision, e.g. retinal problems and dyslexia are also considered. The goal of the thesis is to adapt the sim^{TD} HMI for colorblind people without restarting the system. Two studies were conducted: First, generic parameters and rules for adaptation of the user interface for colorblind people were identified. The second was to confirm applied adaptations.

Studies by [Majstorovic, 2013] thesis suggest adaptations by means of colors, positions, shapes, and patterns in order to support colorblind people using the sim^{TD} functions. She presents several principles that are applied to the presentation models. For example, shapes are used in order to distinguish between visual elements with confusing colors.

The following functions were enhanced in order to support the principles mentioned above:

- Cross traffic assistant
- Electronic brake assistant
- Traffic light assistant
- Traffic jam warning
- Parking information

In Figure 8.16 an example for the traffic light assistant is described. Detailed information can be found in [Majstorovic, 2013].

8.5. Netbike: Implementing the Multiple Reality Model

In this section a first proof-of-concept implementation of the Multiple Reality model (see Section 6.6) is given. A corresponding use case has been introduced and selected for implementation in Section 6.7.

The implementation exploits special characteristics of infrared light and is integrated into a novel form of conspicuity enhancement for vulnerable road users (VRUs). The communication module of PADE is used in order to incorporate bicycles into the existing network and is therefore standard compliant. Current approaches turn on flashing lights and horns in order to increase the visibility of motorcycles (cf. 3.2.3). However, this approach comes with various drawbacks: First, the solution is visible and audible by other, uninvolved road users as well. This could lead to serious distraction. Second, current solutions are expensive and not tailored to the needs of bicyclists. Third, the power consumption is designed for motorcycles, not for bicycles where powerful batteries are absent. Netbike addresses these disadvantages: By augmenting Car2X-based cross-traffic assistance with Multiple Reality, the very nature of MR directs cross-traffic assistance warnings to the specific addressee, hence overcoming current disadvantages such as possible distraction of uninvolved road users. Furthermore, the integration in PADE uses low-cost consumer hardware and is therefore very attractive for bicyclists.

System Overview

Figure 8.17 depicts an overview of the Netbike system. It consists of two parts, one running on the car, the other on the bicycle. The part on the bicycle consists of a smartphone, a communication unit and the conspicuity enhancement module. Relevant sensor data (positioning, heading, etc.) are obtained from the smartphone which is connected via Bluetooth and also serves as HMI for the bicyclist. The obtained sensor data are then broadcast by PADE's communication module to the vicinity. In this first implementation the car is equipped with an infrared camera. It is placed on the hood of the car and captures the whole scene. Two infrared detectors (see Section 8.5.1) recognize the visibility enhancement (here infrared light) of the bicycle and display the scene with the bicyclist surrounded by a red triangle on a head down display (HDD) in the car. Since there is restricted computing power available on the bicycle, only the car assesses the situation. In case of an impending crash it sends out a Pull request to the bicycle to switch on the conspicuity enhancement.

Communication Module

In the case of the Netbike system, the communication module of PADE is realized using consumer hardware (MikroTik Routerboard RB433UAH with a 680 MHz MIPS processor). It runs a custom built version of OpenWRT[1], a popular Linux distribution for embedded devices. For wireless connectivity, a standard 802.11a/b/g card and Bluetooth

[1] https://openwrt.org/

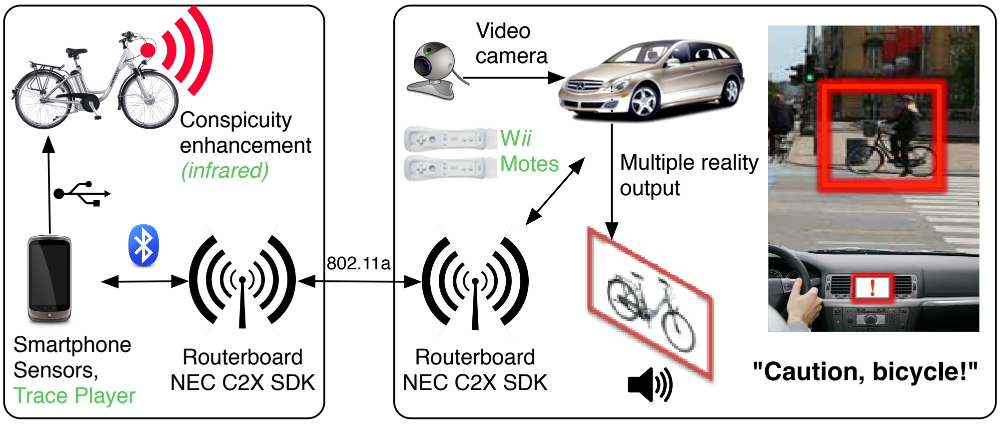


Figure 8.17.: High-level overview of the Netbike System. Left: modules on the bicycle, right: Modules on the car.

are available. 802.11a is used in ad-hoc mode in order to communicate with other cars; Bluetooth connects the smartphone and the on-board unit and transmits sensor data (position, speed, heading).

Communication is handled by a custom-compiled NEC Car2X Communication SDK[Festag et al., 2009], kindly provided by NEC Europe Ltd., for the PADE platform. In this setup, 802.11a is used for communicating with other road users. The frequency of 802.11a and the standard itself is close to 802.11p (see Table 2.15).

Netbike Client

Cross-traffic assistance systems rely on sensor data such as positioning, heading and speed. These data are broadcast to the close vicinity and are then assessed by an application logic which uses these data in order to identify critical situations. Vehicles can easily be equipped by such sensors. Weight and costs are minimal in comparison to costs of a car. Also, power consumption is not relevant for such sensors.
However, on the bicycle the situation is different: Although people consider cross-traffic assistance as a reasonable extension for their bicycle, they may not be willing to spend a large amount of money for this feature.

The Netbike system realizes a sensor module in the form of a ordinary smartphone. Relevant sensors like GPS, speed and heading are available on these platforms. It is linked via Bluetooth into the communication module. The smartphone also serves as an HMI for the bicyclist. Obtained sensor data are then broadcast by the communication module to the vicinity. In this implementation a Google Nexus One is used, running Android 2.1. Car2X-compliant communication is realized by the on-board unit. The smartphone itself is not suited for the following reasons: First, smartphone operating systems do not allow access to lower layers which would be required in order to configure

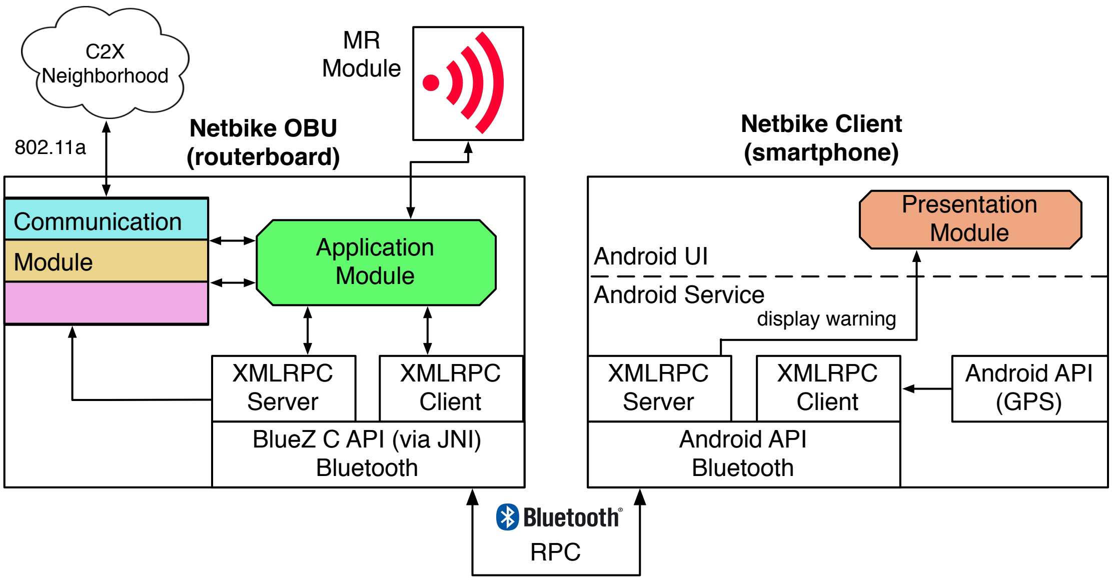


Figure 8.18.: Data flow and software components in the Netbike system.

the wireless hardware. Second, wireless transmission power of a smartphone is limited. Third, every additional smartphone operating system requires a reimplementation of the application logic.

Software Architecture

Figure 8.18 shows individual software components and the information flow of the Netbike system. remote procedure calls are used for communication between Netbike client (smartphone) and on-board unit (routerboard). Since bidirectional communication is required, server and clients have been implemented on both ends. The common standard XML-RPC[2] and the implementation by the Apache Software Foundation[3] were used for realizing remote procedure calls.

During startup, the communication module advertises its services via Bluetooth. The application module of PADE provides access to relevant functionalities such as the neighborhood table and message management. Applications can also register remote procedure calls in order to transparently communicate with the Netbike client. For Bluetooth communication, the Android API already provides the necessary tools for realizing RPCs. However, on the routerboard, the Bluetooth stack of the Linux operating system had to be exposed to Java using JNI (Java Native Interface).

[2] http://xmlrpc.scripting.com/
[3] http://ws.apache.org/xmlrpc/

Power Supply

The system is powered by a motorbike battery of type CT6B-3/YTZ7S. It provides a 12 volt power supply with a capacity of 6 Ah (= 12 x 6W h = 96W h). A maximum consumption of 32W by the Routerboard ensures three hours of operation. However, the average power consumption of the C2X module is far lower than 32W, which increased operation time on one battery charge.

8.5.1. Multiple Reality Module

In this first proof-of-concept implementation, specific characteristics of infrared light are exploited in order to enhance conspicuity of the cyclist. Only the detection module mounted on the car is able to recognize the signal of the LEDs due to the requested and unique pulsation pattern: The wavelength of infrared light is not perceivable for the human eye, and thus will not confuse or distract other uninvolved road users, but can easily be detected and tracked by the corresponding equipment, such as infrared cameras. Installed LEDs emit infrared light of about 940nm wavelength at an angle of 160°. They were chosen to match the most sensitive frequency range of the detection module. Maximum power dissipation was 75mW at a maximum forward current of 50mA. In this configuration the forward current is about 33mA.

The prototype was equipped with 4 LEDs which were installed on the front of the bicycle and connected by wire to the on-board unit. Mounting them at an angle of 90° to each other resulted in a total viewing angle of 250°.

The installed infrared LEDs cannot be accessed from the routerboard directly. In order to control the module, the Arduino platform was used[4]. Arduino is an open-source platform based on Atmel AVR micro-controllers. A board usually has several pins for input/output, a USB or serial-port, a small amount of flash memory and a processor unit.

8.5.2. Detection module

For detecting and tracking the movement of the cyclist, infrared cameras are necessary. In this first proof-of-concept implementation, two remote controls from the Wii gaming-console by Nintendo[5] (wiimotes) realized infrared detectors (see Figure 8.17). These remotes have become popular in the past years because, apart from the low costs, the infrared cameras are very sophisticated and offer an API in order to implement one's own applications.

A built-in processor runs an 8x sub-pixel analysis of the 128x96 pixel input signal and provides an overall resolution of 1024x768 pixels. In the prototype system the two wiimotes are mounted on the hood of the car at a distance of 0.6 meters from each other

[4] `http://www.arduino.cc/`
[5] `http://wii.nintendo.com`

and connected to the onboard computer of the car via Bluetooth. To obtain useful data, a smoothing-filter over the last two received coordinates is applied to the raw signal: $coord_t = 0.5 * coord_t + 0.25 * coord_{t-1} + 0.25 * coord_{t-2}$.

Because of the overlapping field of view, either one or both of the cameras will detect the light emitted from the bicycle. In the first case, the detected coordinates from just one camera are used; in the second case the average between the two corresponding coordinate pairs (i.e. the average of the x-coordinates and the average of the y-coordinates) is computed.

The whole scene is also tracked by a camera mounted on the hood of the car. It shows the entire scene on a HDD in the car. From sensor fusion of the wii-mote coordinates and camera image, an overlay around the bicycle is computed and shown to the driver. Note that the camera and HDD is only used in this proof-of-concept implementation. The Multiple Reality model itself does not specify how the perceivable characteristics have to be changed in order to increase visibility. A future implementation could use special windshields similar to Head-Up-Displays (HUD) but without the drawbacks of Hybrid Reality, e.g. tracking robustness.

The Netbike system described in this section was demonstrated at the International Conference on Intelligent User Interface (IUI) [Castronovo et al., 2011a] and the International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutoUI) [Castronovo and Endres, 2010].

9. A Simulation-Based Evaluation of LGHS and LGHS-CD

In this chapter, the core components of PADE, LGHS and LGHS-CD, are evaluated. They were implemented as routing scheme and integrated into the network layer of a wireless ad-hoc network simulator. Both algorithms were first optimized on a sample scenario and then executed on street networks of German cities. While LGHS was tested against two reference algorithms, these were not available for LGHS-CD. Therefore, LGHS-CD was evaluated against an upper and a lower bound.

9.1. Implementation of $\tilde{f}$, w and $\tilde{g}$

Recall that the graph $G = (V, E)$ is represented by the underlying city map. Intersections are vertices $v \in V$; road segments denote edges $e \in E$. $\tilde{G} = (\tilde{V}, \tilde{E})$ is spanned by the vehicular ad-hoc network. Vehicles are represented by $\tilde{v} \in \tilde{V}$; edges $\tilde{e} \in \tilde{E}$ are considered as wireless connection between two vehicles. The abstractly defined functions $\tilde{f}$, w and $\tilde{g}$ of Section 5.2 are then implemented as follows: Given a time t, $\tilde{f}$ assigns vehicles to specific road segments. Weights are calculated from the distance to the beginning of the assigned road segment. In the application domain of communicating vehicles, $\tilde{f}$ is responsible for controlling and updating positions of vehicles therefore realizing vehicle movements over time. Function w returns the distance between a vehicle in transmission range and an intersection on the city map. Furthermore, the vehicles include the road segments which they have travelled as well as their most probable path in their position updates. This realizes function $\tilde{g}$.

9.2. Evaluation of LGHS

LGHS performed a point-to-point sending task in a simulator for evaluation of vehicular ad-hoc network applications (C2X Simulation Runtime Infrastructure or short VSimRTI [Naumann et al., 2009]). VSimRTI coordinates different simulators and constitutes middleware between the individual simulators (cf. Section 2.2.3). For realistic simulation, the wireless network simulator JiST/SWANS [Barr et al., 2005a] [Barr et al., 2005b] and the traffic simulation SUMO [Behrisch and Bieker, 2011; ?] were used. Furthermore, the effect of buildings on wireless transmission is considered as an extension to the simulator (cf. Section 7.6). Two vehicles are only able to communicate if and only if there is a line of sight between them. This serves as a lower bound on the connection between the

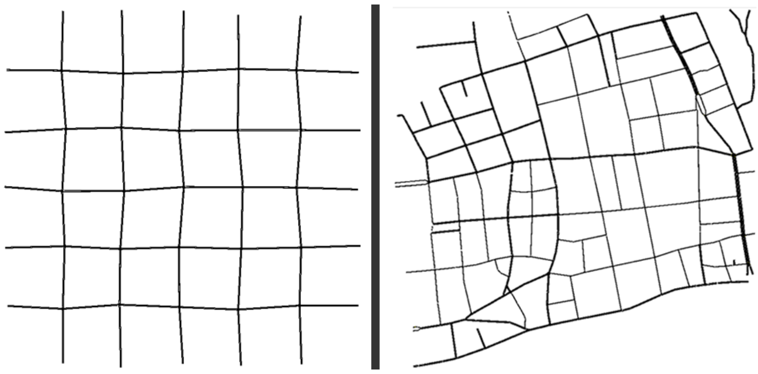


Figure 9.1.: Static graphs used for parameter optimization and evaluation. **Left:** Randomly generated, used for parameter optimization (1200m x 1200m). **Right:** Graph generated out of realistic road network of the city of Heidelberg, Germany used for evaluation (1200m x 1200m).

vehicles in the network simulator.

The performance of LGHS is compared to A-STAR [Seet et al., 2004] and GPSR-MA [Granelli et al., 2007] in terms of path discovery ratio (PDR) and path discovery time (PDT). The first metric measures the ratio $PDR = \frac{SuccessfulShortestPathDiscoveries}{TotalShortestPathSearches}$; the second indicates time from the beginning to the end of the shortest path search $PDT = t_{end} - t_{start}$. Both algorithms were described in Section 2.2.2.

9.2.1. Scenario

Evaluation was done in a scenario where every vehicle starts one shortest path computation to a given car, e.g. the center of the underlying city map. The information providing car remained stationary while all others were driving along a route on the map.

Weights for heuristics 1 – 3 introduced in Section 5.2 were optimized on a randomly generated map shown in Figure 9.1 (left). Weights and values for ω and τ, which were found to be optimal for LGHS, are given in Table 9.1. The goal of the optimization was a high PDR.

After optimization, evaluation was carried out on a map generated out of an existing city environment (Heidelberg, Germany; Figure 9.1, right). The Evaluation runtime was 120 seconds and the shortest path computation was started after 20 seconds simulation time. This ensured a realistic and even distribution of vehicles on the map. n vehicles per second were placed on the map by the simulator and removed after they completed their route, where $n \in \{1, 2, 3, 4\}$. For $n = 1$ this resulted in 120 path computations. Every run was repeated three times for every n and algorithm. This means there were $120 * 3 = 360$ path computations for $n = 1$ in total per algorithm and 720, 1080, 1440 for $n = 2, 3, 4$ respectively, resulting in a total of 3600 path searches per algorithm.

Table 9.1.: Values of the various weights used during the evaluation. The three on the left side denote the weights for the different heuristics while the two on the right were used for the score computation of heuristic 2 (see Figure 5.4, left).

Heuristic Weight		Heuristic 2 score	
weight	value	score	value
α	0.5	ω	2
β	0.3	τ	1
γ	0.2		

9.2.2. Results and Analysis

Results in Figure 9.3 show that LGHS outperforms A-STAR and GPSR-MA in terms of PDR. As expected, PDR increases with increasing traffic density. A-STAR obviously benefits from the available information about the underlying city map (the static graph G). GPSR-MA lacks this kind of information, which results in a lower PDR. As LGHS also takes local information about graph operations in $\tilde{G}$ (the vehicular ad-hoc network) into account, it scores higher on PDR than both A-STAR and GPSR-MA. The results are statistically significant ($p < .001$ according to a $\chi^2 test$).

However, in terms of PDT, GPSR-MA is superior to A-STAR and LGHS. Both A-STAR and LGHS re-schedule path searching in a vertex when no suitable successor vertex could be identified for the shortest path. GPSR-MA's behavior in such a situation is to greedily select a next vertex out of the neighboring vertices and do no re-scheduling at all. This also explains low PDR for GPSR-MA. Interestingly, PDT increases for GPSR-MA but decreases for both A-STAR and LGHS when there are more vehicles on the graph. In this case, both A-STAR and LGHS have an increased number of vertices available for choosing the next vertex for the shortest path, and the probability of finding a suitable one also increases since re-computing time for heuristics is lower than the re-scheduling interval PDT decreases in this case. Because LGHS considers local information about graph operations, it supersedes A-STAR in terms of PDT over time which results in a lower number of re-schedules.

Figure 9.2 depicts Voronoi visualizations for all three algorithms after a run with $n = 4$. Black dots mark starting positions for shortest path computation. Enclosing colored areas denote PDT to the center of the map for an individual path in ms (more red areas mark higher PDT). After a threshold of 15000 ms, path computation was stopped and marked as *failed*. One can clearly see the short PDT but low PDR of GPSR-MA: Paths are found quickly or not at all. Re-scheduling in cases when no successor vertex for the shortest path can be identified results in stepwise PDTs. Results of LGHS reflect high PDR even for large path lengths due to exploiting local information on dynamic graph operations.

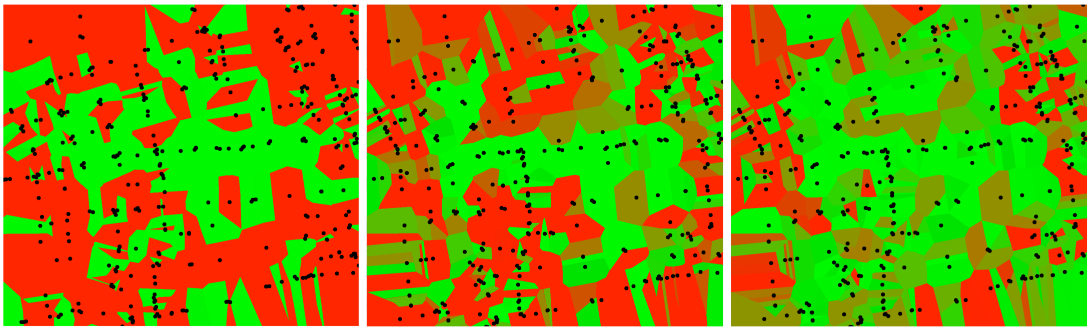


Figure 9.2.: Voronoi diagrams visualizing PDR and PDT. Dots mark the nodes in which the path computation started; color denotes the average time until the shortest path completed in the enclosed region (the greener the faster. The redder the slower. Dark red: not successful). **Left:** GPSR-MA ($n = 4$), fast but unreliable; **middle:** A-STAR ($n = 4$), slower but more reliable than GPSR-MA due to taking correlation of $\tilde{G}$ and G into account; **right:** LGHS, $n = 4$, taking correlation between $\tilde{G}$ and G as well as local information on dynamic graph operations into account outperforms GPSR-MA and A-STAR.

9.3. Evaluation of LGHS-CD

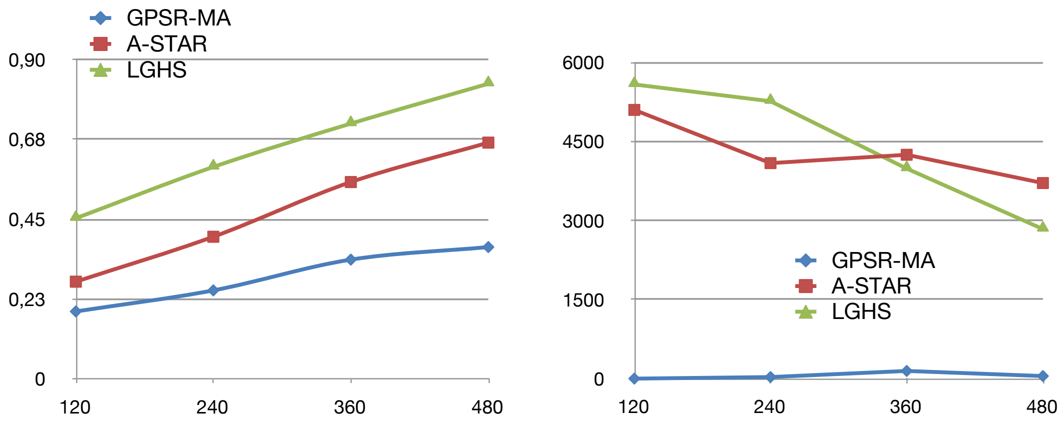
For evaluation of LGHS-CD, a similar technical environment as for LGHS has been built: The algorithm was integrated into the Transportation Layer of the Car2X network stack while VSimRTI [Schünemann, 2011] coordinated simulation of SUMO [Behrisch and Bieker, 2011] and JiST/SWANS [Barr et al., 2005a], [Barr et al., 2005b].
Metrics for measuring performance of LGHS-CD were defined such that effects of routes and crumbs could be observed over time. The performance improvement over time was an important factor for evaluation of LGHS-CD. The following hypotheses were stated:

1. LGHS-CD will develop a significantly higher *cycle discovery ratio (CDR(t))* over time than the reference algorithms.

2. LGHS-CD will develop a significantly lower *average cycle weight (ACW(t))* over time than the reference algorithms.

3. The average *number of visited vertices (NVV(t))* per cycle of LGHS-CD will be less than for the reference algorithms.

Analogous to LGHS, CDR(t) is defined as $\frac{NumberOfSuccessfulCycleDiscoveries}{NumberOfTotalCycleDiscoveries}$ for a time interval $(0, t)$. Due to the effect of crumbs, higher CDRs can be expected when more cycle discoveries succeed. Furthermore, ACW(t) describes the average weight of cycles over all *discovered* cycles during a simulation run. Weight of a cycle is defined as the sum of all edge weights that were collected between two $v \in V$ during forward and backward mode. Since $i.d$ is distributed over the dynamic graph and stored in certain vertices during



(b) PDT

(a) PDR

Figure 9.3.: Path Discovery Ratio (PDR) and Path Discovery Time (PDT) results for all three algorithms. The x-axis encodes the number of path searches, the y-axis encodes the percentage of successful path searches (PDR) and time in ms (PDT). GPSR-MA does not reschedule when no suitable successor vertex for the shortest path can be found. Increasing the number of vertices in the graph affects PDT for A-STAR and LGHS.

backward mode, ACW is expected to decrease over time. For the same reason, the average number of visited vertices per cycle (NVV) is also expected to be smaller over time.

In contrast to LGHS, no algorithm is available for discovering least-weighted cycles on unknown fully dynamic graphs. Therefore, LGHS-CD is evaluated against a lower and an upper bound. In this case, the lower bound is the baseline referred to as *Random Walk (RW)*. This approach randomly selects a $\tilde{n} \in \tilde{N}$ as next vertex during forward mode and checks for goal condition i. Once it discovers a vertex fulfilling i, the same random procedure is done for backward mode. Two additional constraints had to be introduced in order to guarantee termination. First, the number of visited vertices per cycle discovery was restricted to 256 per mode. This results in a maximum of 512 execution steps for forward and backward mode. Second, during the next vertex decision, RW must not select a vertex visited during the previous step. This prevented "bouncing" between two vertices. As an additional restriction for RW, no benefit computation was done and therefore, $i.d$ was not distributed or stored in other vertices.
For the upper bound, a modification of the original LGHS algorithm is used such that $\tilde{v}_{goal}$ and its correlation to G is known during forward mode. Analogously, $\tilde{v}_{start}$ is known during backward mode. It is important to note that this is an upper bound but not the optimal solution. LGHS' still needs to discover shortest paths on the unknown fully dynamic graph. The optimal solution would need to know future dynamic graph topology in order to compute optimal cycles. However, the dynamics of $\tilde{G}$ is nondeterministic in

the evaluation setup used; hence this information is not available. However, discovering $\tilde{v}_{goal}$ and $\tilde{v}_{start}$ using crumbs is one of the key features of LGHS-CD. Therefore, LGHS' with exact knowledge of $\tilde{v}_{goal}$ and $\tilde{v}_{start}$ serves as a suitable upper bound.

Referring to hypotheses 1 – 3, this means that the following results are expected, where $>*$ denotes significantly greater, and $<*$ significantly less than:

- CDR(t): LGHS-CD $>*$ RW, LGHS-CD $<*$ LGHS'
- ACW(t): LGHS-CD $<*$ RW, LGHS-CD $>*$ LGHS'
- NVV(t): LGHS-CD $<*$ RW, LGHS-CD $>*$ LGHS'

9.3.1. Scenario

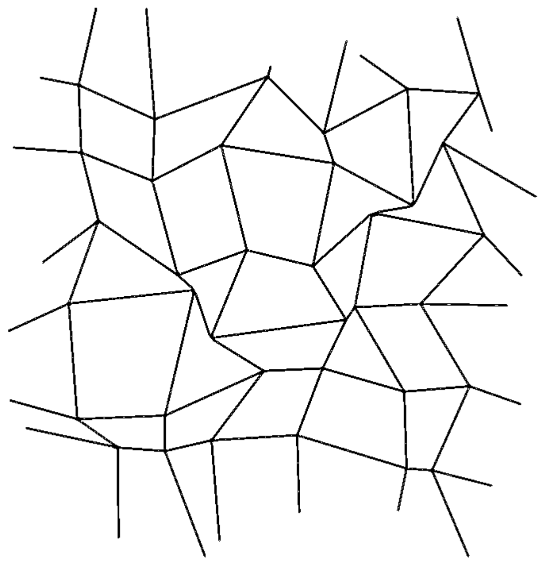


Figure 9.4.: Optimization graph G for LGHS-CD. As for LGHS it was derived from a grid but features more variance length of edges to get varying cycles.

As for LGHS, parameter optimization was done on a randomly generated (static) graph as shown in Figure 9.4. In contrast to LGHS, this graph contains a high variance of edge weights, which was necessary for generating routes with different weights. Dimensions of the graph in the simulation environment were also set to 1200x1200 m. During the optimization process, the following values for LGHS-CD were determined:

- Default value for a new crumb in a vertex $c_{def} = 100$.
- Decrease crumb values by 2 during backward mode: $c_{dec} = 2$.
- Increment existing crumb values by 20 during backward mode: $c_{inc} = 10$.

- Decrease unused crumbs every 5 time-steps: $t_{dec} = 5$.

Determined values for c_{def}, c_{dec} and t_{dec} are balanced in such a way that crumb values decrease just enough over time in order to distinguish current routes from previous ones. If c_{def} was too low and decreased quickly over time, a current route would vanish too early. In the opposite case, where c_{def} is too high and crumb values decrease too slowly, variation between individual crumbs would not differ enough to distinguish the routes.
Crumbs and routes defined by them can only affect performance of LGHS-CD when previous cycle discoveries succeeded. This is due to the fact that crumbs are only set during backward mode. Therefore, an improvement over time is expected. In order to show this improvement over time, simulation runtime was increased by 120s to 240s in comparison to evaluation of LGHS. Every algorithm used for the evaluation stopped after 256 visited vertices. This prevented non-terminating instances of the cycle-detection procedure.
The three evaluation maps as well as the corresponding SUMO networks are shown in Figure 9.5. In contrast to LGHS, evaluation of LGHS-CD was done on maps of three different cities: first, Frankfurt am Main / Niederrad (FFM). The chosen street network is part of the sim^{TD} test field and allows simulation in a similar scenario in which the field operational test was conducted. The second city is Heidelberg (HD). LGHS was already evaluated on this network. The third network was constructed from the campus roads of Saarland University in Saarbrücken (SB).

The simulation parameters were set as follows:

- Graph dimensions: 1200 x 1200 m.
- Simulation runtime: 240 s.
- Initial location of $\tilde{v}_{start}$ was set to the center of the map.

Similar to LGHS, several runs per algorithm were conducted. n vehicles per second were added to the simulation where $n \in 1, 2, 3$. Each algorithm was run with these parameters on all three maps, resulting in 27 simulation runs. Cycle discovery was started once per vehicle 20 seconds after the car was added to the simulation. Data was analyzed after the simulation with respect to CDR, ACW, and NVV. One randomly selected car fulfilled the goal condition. In contrast to LGHS, it was not stationary but followed its assigned route that was set by the traffic simulator.

9.3.2. Results and Analysis

In the following discussion, results for all three metrics are covered. Each result graph shows one metric for all algorithms on an evaluation map. The term “Xps” refers to the added vehicles per second, e.g. 1ps. Note that all cycle discoveries were started with an offset of 20 $\pm$ 4 seconds in simulation time. Therefore, all result graphs are plotted with this delay on the x-axis.

Circuit Discovery Ratio (CDR)

Figure 9.6a shows CDR for the FFM scenario. At the beginning of the simulation no crumbs exist and several cycle discoveries fail. This explains why LGHS-CD can never result in 100% CDR. LGHS-CD reaches approximately 60% CDR at the end of the simulation in the 1ps condition. This is far below of 95% CDR for LGHS' in the 1ps condition. This indicates that the route maintenance cannot be fully exploited when fewer vehicles are on the map. It also is an indication for less accurate approximation of $\tilde{v}_{goal}$ during step 2 in forward mode. However, LGHS-CD scores consistently higher on CDR than RW in the 1ps condition. Crumbs therefore improve CDR compared to the baseline. The number of vehicles on the map has a high impact on the resulting CDR. For condition 2ps, crumb effects are immediately visible (high gradient of the plot). CDR of LGHS-CD outperforms the baseline at approximately 75 seconds of simulation time and reaches nearly 80% at the end of simulation (LGHS': 90%). The approach to LGHS' becomes even more clear in the 3ps condition. Here, LGHS-CD reaches 91% while LGHS' scores 95%.

The topology of the scenario HD depicted in Figure 9.5b differs from FFM in that it resembles more a grid while FFM contains connected clustered areas. This fact is also reflected in results of CDR as shown in Figure 9.6b. RW reaches 40% in the 3ps condition. 1ps and 2ps are even below that. The grid structure allows added vehicles to be distributed equally on the map. The probability of discovering the target vehicle is then lower than for the FFM scenario where clusters and thin connections in between exist. While LGHS-CD can compensate for this by relying on crumbs, RW cannot. This is reflected by higher CDR of LGHS-CD even in the 1ps condition. For 3ps, LGHS-CD is above 80% on the HD scenario at the end of simulation.

The network of the SB scenario (Figure 9.5c) features fewer streets within the same dimensions. This resulted in a high vehicle density even in the 1ps condition. Accordingly, all three algorithms result in higher CDR than for the other two scenarios. Still, RW is below LGHS-CD at the end of simulation. In this last scenario, the 3ps condition performed almost equally to LGHS', as shown in Figure 9.6c.

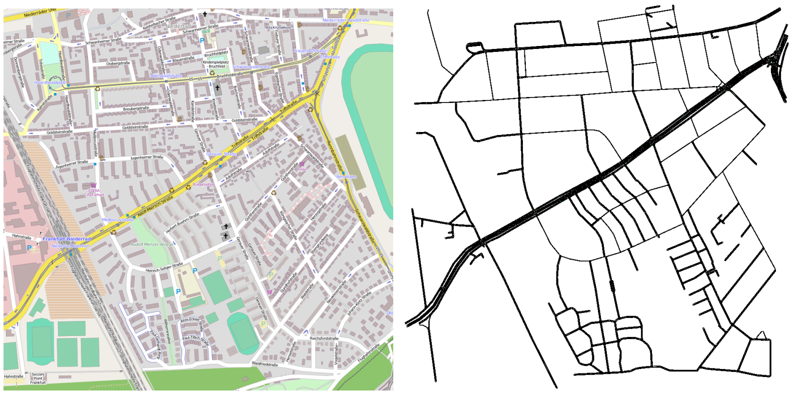
Average Cycle Weight (ACW)

The second metric, average cycle weight (ACW), is shown in Figure 9.7 for all three scenarios. Note that the y-axis has been scaled logarithmically in order to compare results of all algorithms in one graph. As shown in Figure 9.7a, RW results in high cycle weights (above 10,000) while cycle weights of LGHS-CD constantly decrease over time. At the end of simulation, ACW even falls below LGHS'. Crumbs on the route affect performance of LGHS-CD significantly with respect to ACW. RW cannot exploit this information.

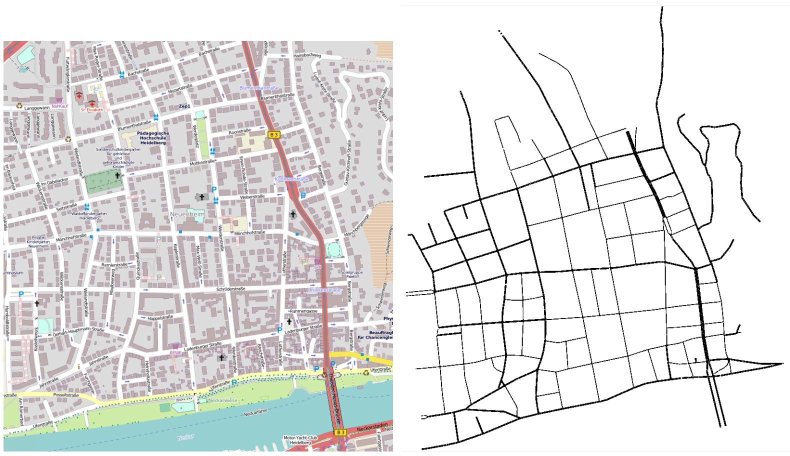
In scenario SB, ACW of LGHS-CD converges even faster to a value around 100 because of the more compact street network compared to the other two scenarios. For all three scenarios, LGHS-CD almost reaches the performance of LGHS', which is close to an optimal solution.

Number of Visited Vertices (NVV)

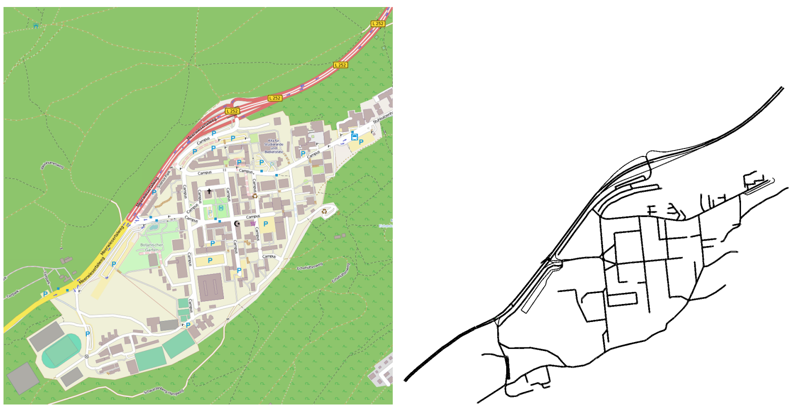
The third and final metric is the number of visited vertices (NVV). Recall that after a limit of 256 hops, cycle discovery was aborted. This results in a maximum result of 256 on the y-axes in Figure 9.8. Also here, the y-axes have been scaled logarithmically. The random approach of RW is close to this limit, as expected. LGHS-CD is far below this limit due to crumbs and subgraph approximation. Also for this metric, differences between the nearly optimal solution LGHS' and LGHS-CD are negligible.



(a) Evaluation map 1: Frankfurt am Main / Niederrad (FFM)

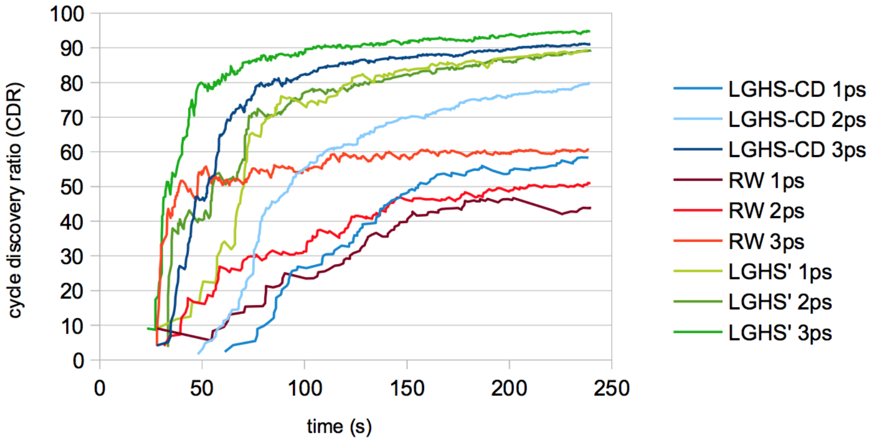


(b) Evaluation map 2: Heidelberg (HD)

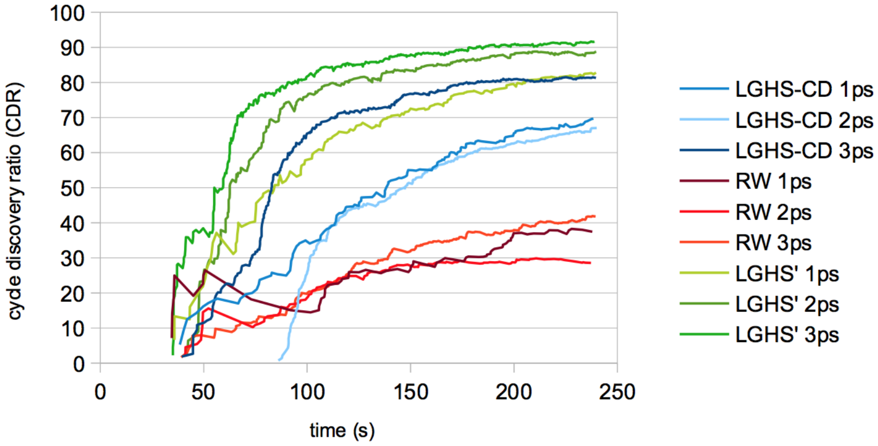


(c) Evaluation map 3: Saarland University, Saarbrücken Campus (SB)

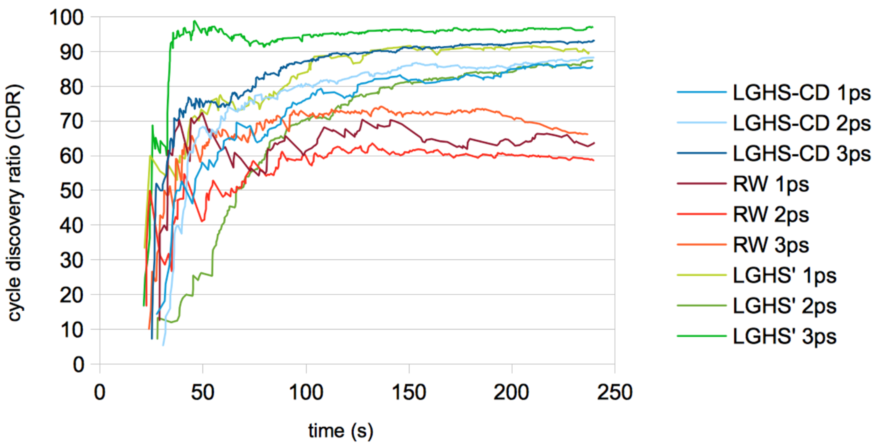
Figure 9.5.: Evaluation maps for LGHS-CD. Left: Open Streetmap views, right: SUMO networks.



(a) CDR for map FFM

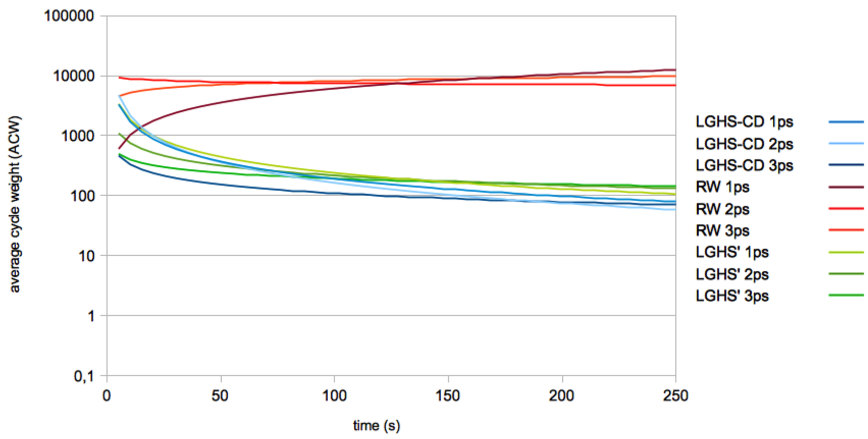


(b) CDR for map HD

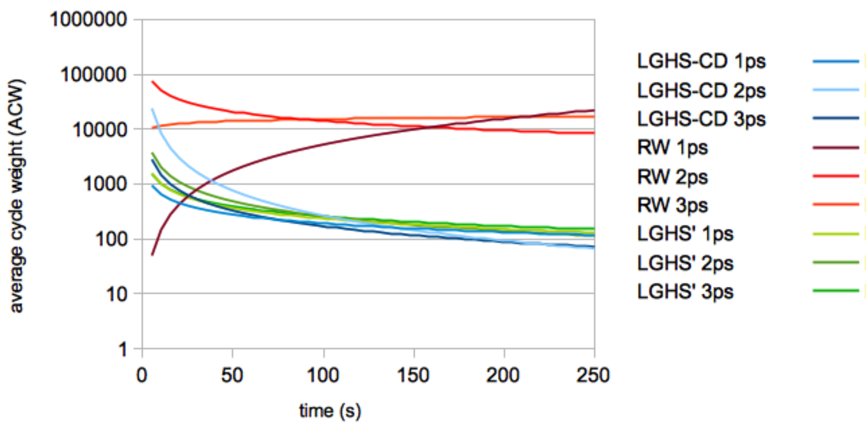


(c) CDR for map SB

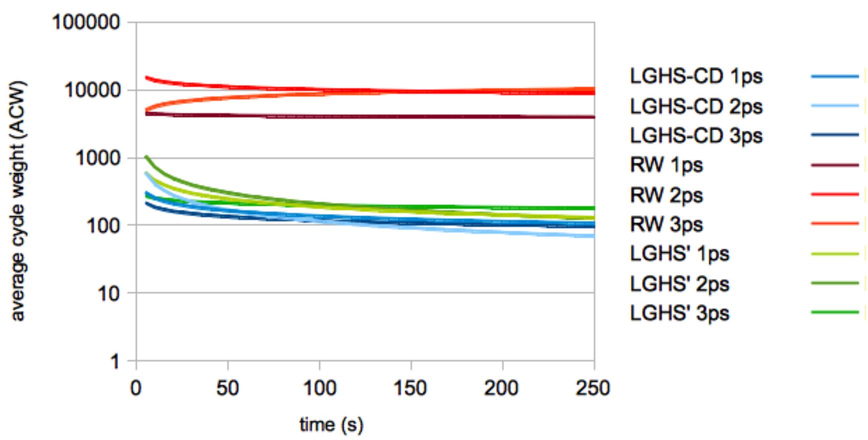
Figure 9.6.: Evaluation results for CDR on all three maps.



(a) ACW for map FFM

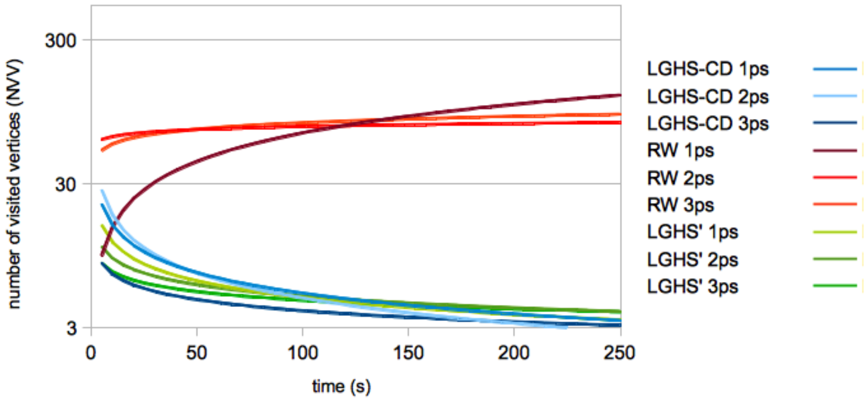


(b) ACW for map HD

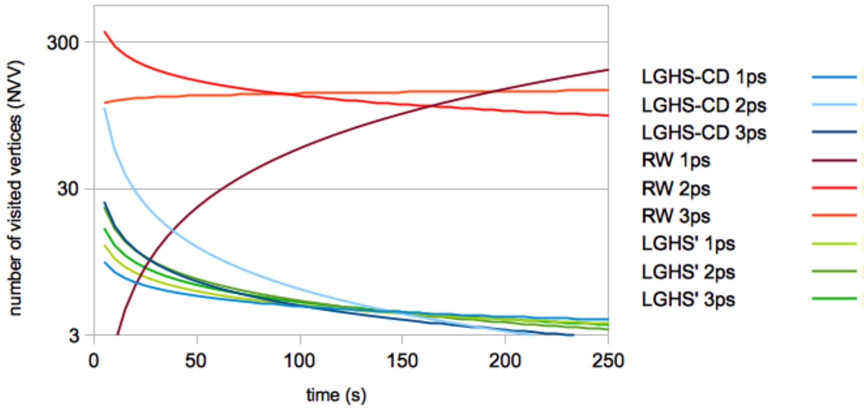


(c) ACW for map SB

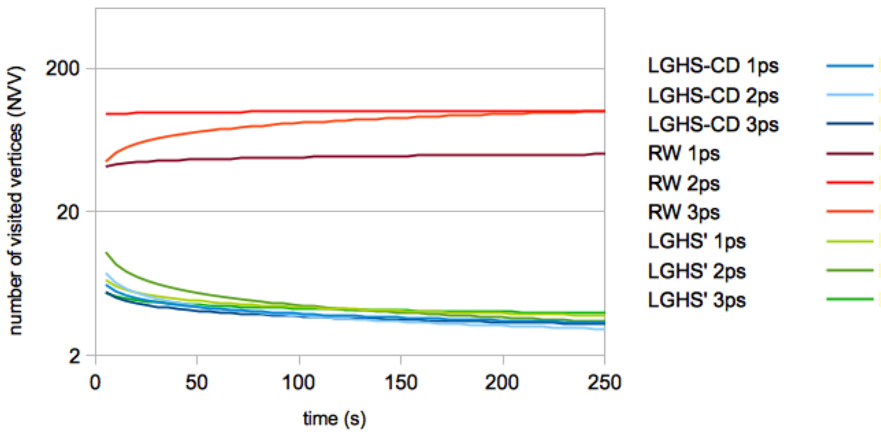
Figure 9.7.: Evaluation results for ACW on all three maps.



(a) NVV for map FFM



(b) NVV for map HD



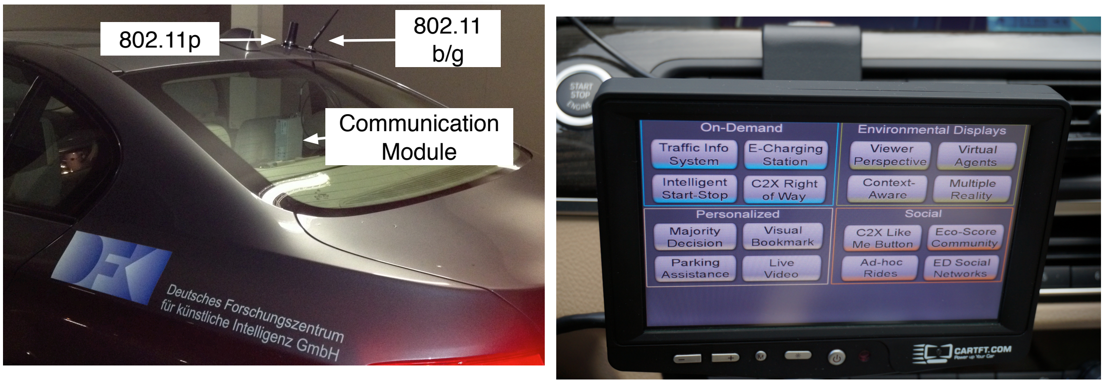
(c) NVV for map SB

Figure 9.8.: Evaluation results for NVV on all three maps.

10. Integrating PADE into Test Vehicles

10.1. Test Vehicle Integration

The PADE system has been integrated into the BMW 5 series test vehicle at DFKI. Demonstrators described in Section 8.2 and 8.3 became an integral part of the Advanced Driver Assistant (ADAS) Lab, one of the Living Labs at DFKI. A regular desktop PC (Intel core i7, Windows OS) was installed in the trunk of the test vehicle and served as the application unit. A professional voltage converter ensured a reliable power source. For the communication module, a Linkbird MX was connected via ethernet in order to establish communication between JAPI and C2X SDK (cf. Section 7.8).



(a) PADE's communication module and antennas for 802.11b/g/p.

(b) PADE's presentation module.

Figure 10.1.: PADE integration into the BMW 5 series test vehicle at the Advanced Driver Assistant Lab (ADAS).

In order to transfer data over regular WiFi, a USB network card with an external antenna was plugged into to the application unit. Both antennas (Linkbird and desktop PC) were led through one of the side windows and attached magnetically to the roof of the car. This ensured the best communication between vehicles and infrastructure. A second test vehicle (Smart fortwo) was equipped with hardware from sim^{TD}. The application unit was a NEXCOM VTC6200 embedded PC with an Intel ATOM CPU. The communication unit was sim^{TD}-specific hardware developed by Continental, but NEC was also responsible for providing the Car2X network stack. The Smart was used for demonstrating the on-demand traffic information system that was presented in Section 8.2.

10.2. sim^{TD} Field Operational Test

As described in Section 8.1, the presentation component of C2X PADE was applied to the sim^{TD} FOT and was experienced by over 400 drivers on a daily basis. During the final press event on June 30, 2013, where results of the project were presented to the public, several OEMs decided to show their own (graphical) implementation of the HMI while the underlying presentation component was left untouched. Among these were Opel and Volkswagen. Both of them wanted to show series-production readiness of Car2X technology and therefore fully integrated the HMI into their cars. Therefore, PADE's presentation component was adapted to fit fully into Opel's and VW's multimedia system and the OEMs were provided with the AUML specification to develop their own UI. This section describes their implementations.

10.2.1. Integration of PADE into the Volkswagen Golf VII

The seventh generation of the popular Golf by Volkswagen features a large touch-sensitive display on the center stack and a multifunctional display that is integrated into the instrument cluster. This is typically used by the IVIS to display information from the touch-sensitive display in iconified form, for example compact guidance instructions from the integrated navigation system.

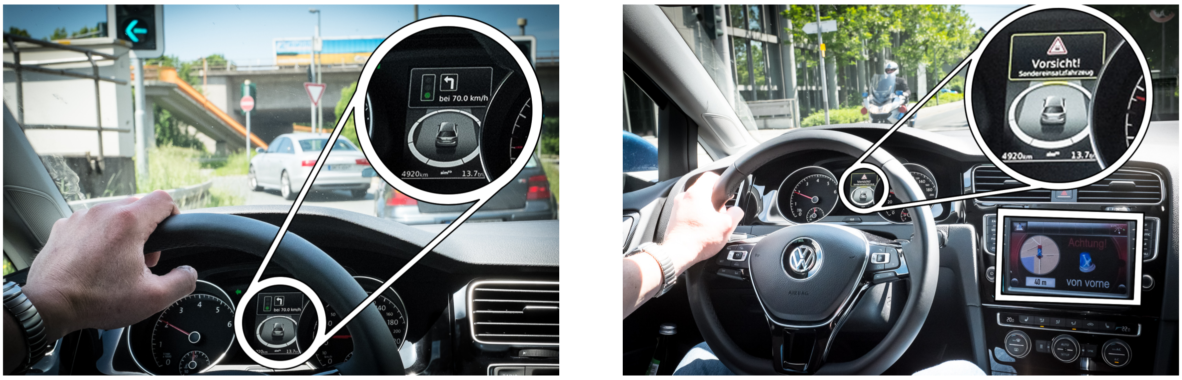


Figure 10.2.: PADE's presentation component integrated into the Volkswagen Golf VII. An additional display strategy representing the multifunctional display was added to the presentation model. **Left:** traffic light assistant; **right:** emergency vehicle warning. *(Image reproduced with the permission of Volkswagen AG.)*

Both of these displays were used when PADE was integrated into the on-board systems. On the center-stack display the fully-fledged version of the sim^{TD} HMI was presented (Figure 10.2, right) while the multifunctional display showed iconified versions. The underlying presentation engine controlled these two displays. Therefore, presentation models needed to be adapted in order to reflect this additional presentation strategy.

Because the AUML of the display strategy contained both screens, presentations appeared simultaneously. During the integration process, Volkswagen was provided with icons that were used in the symbol area on the sim^{TD} HMI. This ensured a consistent appearance of both implementations. Furthermore, the AUML specification as contained in the presentation model was given to the developers. Using this information, the instrument cluster representation was implemented. Figure 10.2 shows the resulting user interface as it was demonstrated to the public during the final event. In these two examples, a traffic light assistant and an approaching emergency vehicle warning are shown. However, the integration by Volkswagen featured all major sim^{TD} functions.

10.2.2. Integration of PADE into the Opel Insignia

Opel decided to replace the UI of the sim^{TD} reference HMI completely and built a graphical substitution that fitted the Opel Insignia's look and feel. However, as for the integration done by Volkswagen, PADE controlled the presentations and decided when to show what on the graphical UI. The implementation of Opel featured a main area like the reference HMI but only two additional slots for presentation on the symbol area. Presentation models were adapted to reflect this different layout. From a user's point of view, there was no difference from other IVIS that are installed in series models. PADE was fully integrated into the car's entertainment system. Two examples are shown in Figure 10.3: a road weather warning and a broken-down vehicle warning. The upper bar contains series production elements of Opel Insignia's HMI: current time and outside temperature. The left picture in Figure 10.3 also shows two other sim^{TD} functions in an iconified version: speed limit and roadworks information system. PADE's presentation engine put the highest prioritized function on the main area, here road weather warning.

Figure 10.3.: PADE's presentation component integrated in an Opel Insignia. It consisted of a main screen and a symbol area with two slots that were arranged vertically. **Left:** road weather warning on the main screen and traffic sign assistant as well as roadworks information system on symbol area. **Right:** broken-down vehicle warning on main screen and traffic sign assistant on symbol area. *(Image reproduced with the permission of Opel AG.)*

10.3. E-Bike Fleet Management

A demonstrator for managing e-bike fleets was presented to the public from February 25th to 28th, 2013 in the context of the Mobile World Congress (MWC), Barcelona, Spain with 72,000 visitors[1]. It was based on the use case "E-Bike Combined Battery and Communication Module" as outlined in Section 6.5.2. The demonstrator showed how renting companies, authorities, and institutions can exploit latest developments in ICT to efficiently manage and maintain their bicycle fleets. The system covered basic fleet management (security, authentication), advanced fleet management (surveillance, maintenance), active safety, and added-value services. During the congress the demonstrator attracted international representatives from e-bike manufacturers as well as fleet managers of motorized two-wheelers. Additionally, possible safety features were presented in separate videos where the integration of vulnerable road users such as bicyclists into the Car2X community was shown. Especially in urban areas where e-bicyclists reach speeds that are above those of their unpowered counterparts, it can actively contribute to increased safety. Therefore, the demonstrator addressed common danger zones for cyclists and showed how they can be prevented using wireless ad-hoc communication, e.g. "the door prize": The door of a parked car is being opened at the same time a bicyclist is passing by. By using wireless communication between car and bicycle, this dangerous situation can be avoided by locking the doors for just a few seconds, giving the cyclist enough time to pass.

Figure 10.4.: E-bike fleet management demonstrated at the Mobile World Congress 2013, Barcelona, Spain. **Left:** battery with status LEDs; **right:** smartphone connected to the bike with controls for the bike's battery.

The following scenario was presented: A customer intends to rent an e-bike. After registering for the service and entering her payment information an application on her smartphone allows her to use the provider's services. When arriving at the rental station, the customer sends the id of an e-bike to the provider and receives an unlock code back. The smartphone then connects to the bike via Bluetooth and verifies the unlock code. If

[1]http://www.gsma.com/newsroom/gsma-mobile-world-congress-2013-sets-new-records

successful, the communication with the bike is established in both directions: The bike's sensors can be accessed by the smartphone app, while the bike gains internet access. The demonstrated app turned into an HMI for the e-bike. The user was able to switch on and control battery support while cycling, get trip information, etc. The bike confirmed connected smartphones with status LEDs. The GPS sensor of the phone was used to transmit the current position to the provider. Additionally, the demonstrator featured a fitness application as an example of a added-value service.

Technically, the bike's battery was enhanced by an Arduino board. The power supply was realized by wiring Arduino's Power over an Ethernet connection to the bike's battery. The Arduino was equipped with an additionally Bluetooth module to connect the smartphone. Arduino was also responsible for controlling the mounted LEDs. The hardware on the bike was completely integrated into the battery and only LEDs were visible to the visitors. A red LED indicated a locked battery while blue was used for a successful Bluetooth connection. A green LED confirmed a successful unlock procedure. The smartphone was mounted to the handlebar. Figure 10.4 shows the described demonstrator.

11. Conclusion

This thesis introduced the Pull Paradigm, a complimentary supplement to existing Car2X Communication networks. In contrast to the established Push Paradigm, where the focus is primarily on fast and efficient dissemination of information, the Pull Paradigm adds *interactivity* and therefore opens a wide range of additional application areas beyond today's safety and traffic-efficiency use cases.

Existing technologies, research, applications, and standards in the domain of Car2X were thoroughly analyzed. Numerous missing aspects were identified that prevent an immediate implementation of the Pull Paradigm with the current network stack. The main issues were information discovery procedures, bidirectional communication, added-value services, and standardized message formats that do not reflect Pull requests.

This led to a requirement list that defines necessary enhancements to almost every layer of the Car2X network stack in order to implement the Pull Paradigm. Essential components in this effort are two novel algorithms for managing information flow in vehicular ad-hoc networks. Both algorithms are formally specified with the help of graph theory and implemented as routing algorithms for vehicular ad-hoc networks. The algorithms have the following properties:

- **Linked graphs heuristic search (LGHS)** discovers shortest paths to information sources by only relying on *local* information about the network. Paths are first calculated on the street network using uniform-cost search. In a next step, three heuristics approximate this path in the ad-hoc network.
- **Linked graphs heuristic search with cycle discovery (LGHS-CD)** adds cycle discovery to LGHS and maintains *routes* to information sources in order to speed up future requests. Route maintenance is done in a distributed manner and no central management component is involved.

Both algorithms were evaluated on large, simulated city scenarios. The performance of LGHS under the tested conditions is superior to A-STAR [Seet et al., 2004] and GPRS-MA [Granelli et al., 2007], two routing algorithms for Car2X networks, by means of path discovery ratio. A-STAR and GPRS-MA were chosen because they also, like LGHS, try to exploit several characteristics of the vehicular ad-hoc network.
For evaluation of LGHS-CD no comparable algorithms could be identified. Therefore, LGHS-CD was tested against the (almost) optimal solution and the random walk. The (almost) optimal solution is LGHS provided with necessary information about $\tilde{v}_{start}$ and $\tilde{v}_{goal}$. The algorithm therefore knows exactly which forwarder is the optimal one. The

random walk implementation selects the next vertex of a path at random. Results show that LGHS-CD converges to the optimal solution over time on all three tested city scenarios.

Further important contributions to the implementation of the envisioned paradigm were developed:

- A new message format that reflects Pull requests has been defined. The proposed requested environmental notification message (REM) complements the existing message types common awareness message (CAM) and decentralized environmental notification message (DENM).
- A comprehensive automotive Human Machine Interface that integrates and coordinates ADAS/IVIS accessing limited resources of the HMI in parallel was developed. The focus is on *information presentation*. Analysis and decomposition of Pull requests is subject to future work.
- OpenDS, the open source driving simulator was added to an existing simulator coupling framework, thus enabling research on the Pull Paradigm from a user's perspective in the lab.

These components contribute to C2X PADE (the Car2X Platform for Application Development and Evaluation) which supports the development of Push and Pull applications. Using this platform, the following use cases were implemented in order to demonstrate the Pull Paradigm:

- The first demonstrator is an instantiation of the *on-demand traffic information system* that enables the driver to request an image from another car. Following the Pull Paradigm, the demonstrator exposes external cameras to other road users in order to function as an *on-demand traffic mirror* in turn-off situations.
- The second demonstrator shows *interaction with environmental displays (ED)*. The ED is showing advertisements for two movies. By requesting more information about one of them, the car sends REMs to the ED that trigger playback of the trailer. Furthermore, the focus of attention is tracked and the movie currently in the driver's focus is highlighted. The communication with the ED is done via the proposed REMs. The demonstrator is realized with the help of two other components developed at DFKI: SiAM DP, a multimodal dialogue platform for automotive applications and EyeVius, a system for referencing the outside environment.

In summary, existing technologies of the Push Paradigm were successfully extended in such a way that the Pull Paradigm could be implemented.

11.1. Research Questions Revisited

The following research questions were answered:

- **Is the proposed Pull Paradigm applicable to established Car2X Communication networks?** The current Car2X network stack was thoroughly analyzed and results revealed a direct application of the Pull Paradigm using available technology is not feasible. In particular, underlying network protocols lack the ability to discover information and to backtrack to the vehicle that started the request. Therefore, LGHS and its enhancement LGHS-CD were defined with the help of graph theory and implemented as routing algorithms in the Car2X network stack. The algorithm discovers shortest paths to information sources in vehicular ad-hoc networks and introduces a distributed path maintenance procedure that speeds up future requests. Furthermore, standardized message formats are not flexible enough to represent Pull requests. This led to the definition of requested environmental notification messages (REMs) that supplement the currently standardized CAMs and DENMs.

- **What are possible application areas for the Pull Paradigm considering specific limitations of Car2X Communication networks?** In this thesis, four categories of possible application areas in the context of the Pull Paradigm are proposed: on-demand, environmental displays, social, and personalized. Each category contains four use cases. All of them go beyond today's safety-related use cases and involve the driver in the communication with the environment. For example, the microscopic parking-lot assistant enables the driver to locate a free spot by querying the ad-hoc network. Of course, the list of use cases in this thesis cannot be complete but helps shape the Pull Paradigm on the application level. All proposed use cases are analyzed regarding their requirements to the underlying technology. These requirements also helped define necessary enhancements of today's Car2X Communication networks in order to implement the Pull Paradigm.

- **Are existing simulation tools applicable for integrating Pull applications?** In order to answer this question, existing tools for simulating Car2X networks were analyzed. It turned out that simulation of complex scenarios in vehicular ad-hoc networks requires at least a network and a traffic simulator. For both there exist solutions in the literature, e.g. the traffic simulator SUMO and the network simulator JiST/SWANS. Furthermore, *simulator coupling* is necessary. Also here, the thesis can build on existing technology, e.g. the vehicle simulation runtime infrastructure (VSimRTI) [Schünemann, 2011] which enables the coupling of time-discrete simulators and already integrates JiST/SWANS and SUMO. However, because applications of the Pull Paradigm are more focused on the user, a driving simulator is needed. Therefore, VSimRTI was used to add OpenDS, the open source driving simulator, to the simulation community. VSimRTI provides two interfaces that need to be implemented in order to connect additional simulators: the federate on the OpenDS side and the ambassador on the VSimRTI side. The federate subscribes to vehicle movements from SUMO and Car2X messages from JiST/SWANS. Using this information, simulated cars and network traffic can be visualized in OpenDS. Additionally, the user's EGO vehicle is represented in JiST/SWANS and is considered by the network simulation.

- **How can development of Pull applications be supported on the software level?** The thesis introduced C2X PADE, a software platform that supports development of Car2X (Push and Pull) applications. The platform adds a novel *abstraction layer* to the network stack that enables implementation of applications that are runtime independent, i.e. there is minimal effort to transfer the application from the lab to a test vehicle. Currently, two runtimes are supported: simulation and "real" Car2X communication. Messages can either be sent via 802.11p using a Car2X prototyping platform (NEC Linkbird MX), or using the wireless network stack of the operating system. Therefore, other wireless communication technologies such as 802.11a/b/g/n and UMTS/LTE are also integrated. For compatibility with other systems developed at DFKI, PADE can run as bundle within the well-known OSGi framework and offer its services to applications.
- **How can existing ADAS / IVIS and Pull applications be integrated in a common HMI framework?** C2X PADE contains a human-machine interface module that coordinates presentation requests of independent running applications. It introduces *presentation models* that contain all necessary information about the HMI part of an application, e.g. obligatory parameters and their allowed values for creating a presentation task. The programming interface is created dynamically out of the information contained in the presentation model. This module ensures *situation-aware* allocation of HMI resources by considering *presentation strategies* based on the current display context and priorities individually set by the application according to the current driving context. The human-machine interface module is also integrated into the OSGi framework and offers services for creating, updating and canceling presentation tasks. The module follows the OSGi whiteboard pattern in order to distribute interactions of the user to other applications which then can react to the input. It was applied to the sim^{TD} project and was installed in a fleet of 120 cars. The human-machine interface module was used by several OEMs during the project in order to demonstrate full integration of Car2X technology into their cars.
- **What are qualified evaluation metrics that measure performance of the Pull Paradigm?** LGHS and LGHS-CD are core components of the Pull Paradigm. The evaluation of these two algorithms introduced several metrics in order to measure the performance of Pull requests:
  - **Path discovery ratio (PDR) / cycle discovery ratio (CDR)** is the fraction of successful path/cycle discoveries versus all path/cycle discoveries and is a quantitive measure for Pull requests.
  - **Path discovery time (PDT)** measures the time for a path discovery.
  - **Average cycle weight (ACW) and number of visited vertices (NVV)** can be used for measuring the quality of discovered paths.

  These metrics were used to evaluate LGHS and LGHS-CD.

11.2. Applicability of the Pull Paradigm to First-Generation Car2X Networks

The proposed Pull Paradigm differs in essential points from established Car2X applications. In a nutshell, information source discovery and backtracking to the requesting car is an innovation introduced by the Pull Paradigm, while current Car2X networks rely on broadcast mechanisms. With the completion of field operational tests, Car2X technology will soon be ready for the market, which raises the question of whether the Pull Paradigm is applicable to first-generation Car2X networks. This section critically analyzes this topic.

- **Penetration rate** is already a major challenge of the Push Paradigm. Impact on traffic and safety can only be verified if a large number of vehicles is equipped with Car2X technology. The first-day use cases after the completion of sim^{TD} were chosen such that, even at low penetration rates, users would be able to realize the benefit. One of this use cases is the traffic light assistant. The obvious reason for choosing this use case is that traffic lights in highly-frequented inner-city areas can be equipped with with Car2X technology and first-day customers immediately recognize the value of the technology. Safety-related use cases based on vehicular ad-hoc communication are not considered in the first-generation Car2X networks. Some use cases of the Pull Paradigm require even higher penetration rates since only a strongly connected network can reliably answer requests. This concerns special use cases like the microscopic parking-lot assistant. Other use cases that directly communicate with road infrastructure, like environmental displays, are better suited for an application to first-generation Car2X networks. Similar to the traffic light assistant, customers instantly benefit from the technology.
- **Standards** play a major role in the domain of Car2X networks. In contrast to other advanced driver assistant systems, cooperative vehicles can only be successful if vendor-independent communication is possible. In order to apply the Pull Paradigm, new standards have to be developed. It was shown that this not only affects communication protocols but also message formats and applications. Assessing the efforts that were made during the first decade of the 21st century in order to come up with European-wide standards, it becomes clear that this poses a major hurdle to the Pull Paradigm. In order to overcome this limitation, one could think of a more service-oriented Car2X network across OEMs, mobile phone providers, and service providers. This is currently addressed in the German research activity CONVERGE where an "open, distributed, transregional/international connecting, provider-independent, scalable, flexible, secure and hybrid communicating Car2X system network"[1] is developed. Applications of the Pull Paradigm will greatly benefit from results of this project.
- **Bandwidth** is considered as a limited resource in vehicular ad-hoc networks. Although LGHS and LGHS-CD are both greedy algorithms with minimal overhead,

[1] http://www.converge-online.de/?id=020200&spid=en

applications of the Pull Paradigm should communicate on a dedicated channel in order to avoid interference with other Car2X-based ADAS. However, this means that additional standards are required.

- **Version management** also poses a major challenge with respect to the Pull Paradigm and its applicability to first-generation Car2X networks. While minor updates may not be a problem, major changes to protocols or the network stack might break compatibility. Updates also have to be coordinated across different OEMs. While it would be possible to rely on the car's mobile connection for delivering these updates, it is not clear how users would react. Some may have privacy or security concerns.

While this thesis showed the general feasibility of the Pull Paradigm by building upon existing technology, application to first-generation Car2X networks is only reasonable for use cases that send requests directly to other network entities, e.g. environmental displays. Pull applications using multi-hop requests are reserved for future Car2X networks.

11.3. Outlook

- Currently, the simulation module of PADE refreshes positions of vehicles in OpenDS at the time it receives updates by the simulation infrastructure. However, this affects perception of the environment by the driver steering the EGO vehicle in OpenDS: If the simulation framework does not update in real time, the situation might seem unrealistic. An interesting enhancement would be to control the real-time factor of the simulation. This might be easier for the case where simulation is too fast, but additional effort is necessary for increasing the real-time factor. This could be achieved by decreasing the model quality of the simulation.
- The presentation module of PADE will be further developed in the German research initiative CONVERGE[2]. The project aims to implement an open, service-oriented Car2X system network that connects OEMs, mobile phone providers, and service providers. Car2X messages are not only transmitted via C2C but also using C2I Communication, i.e. CAMs and DENMs can also be sent indirectly from vehicle to vehicle. This adds a variety of additional paths for message delivery. In CONVERGE, PADE's presentation module will be adapted to visualize these network paths for messages within the Car2X system network.
- The simulation module of PADE supports currently only one human driver. A valuable extension would be multi-user scenarios where a large number of drivers are sharing the same 3D environment. This can be achieved using the scenario markup language (SML) [Gajananan and Prendinger, 2013] which describes a method for modeling controlled traffic situations. The authors describe SML as part of a novel framework for driving simulator studies based on multi-user scenarios.

[2]http://www.converge-online.de

- The thesis focused on implementing the Pull Paradigm on the network level but also identified missing aspects of today's Car2X networks on the user interface level. Here, the input of complex Pull requests while driving is not considered at the moment. The author of this thesis is currently working on an integration of the presentation module into the SiAM-DP, a multimodal dialog platform for automotive applications. Multimodal interaction will greatly enhance the input possibilities for Pull requests.

- LGHS-CD currently only considers direct communication between vehicles. One could think of a hybrid approach that integrates RSUs in the shortest path calculations. Suppose cars also announce their presence to RSUs. Then, a central service could manage information flow between two cars over the two involved RSUs. This would create short paths over wired/WiMAX networks and potentially increase path discovery ratio.

- When considering potential business models for Car2X Pull applications, one could elaborate on the idea of micro-payments: while the Pull app can be downloaded free of charge from the OEM's store, users pay for every request. Of course, only successful queries are billed, e.g. the customer only pays if the microscopic parking-lot assistant finds a suitable spot.

- The combination of several wireless communication technologies could be valuable for applications of the Pull Paradigm. For example, the location of a target vehicle could be predetermined using cellular networks before sending the request over the ad-hoc network. This could make the introduced algorithms even more reliable in situations where no crumbs exist.

- Theoretically, the next generation of cellular radio networks (5G) will be able to support cell-based ADAS since latency times may be close or even faster than direct communication between vehicles. For example, if a car detects an inevitable crash it could trigger the release of the airbag in another car (wireless airbags). When applying this technology to the Pull Paradigm, one could think of "car drones" that are remotely controlled (driving 2.0).

A. Detailed Results of the Analytic Hierarchical Process

	Vehicle Demo	Simulator Demo	Experienceability	Usability	Pull Paradigm	Market acceptance	Implementation
Vehicle Demo	1	3	1	3	1	7	4
Simulator Demo	1/3	1	1/3	2	1/6	5	1/4
Experienceability	1	3	1	5	1	9	3
Usability	1/3	1/2	1/5	1	1/5	5	1/4
Pull Paradigm	1	6	1	5	1	9	4
Market acceptance	1/7	1/5	1/9	1/5	1/9	1	1/7
Implementation	1/4	4	1/3	4	1/4	7	1

Figure A.1.: Comparison matrix for all criteria used for the AHP.

On-Demand Traffic Information System		
Criterion	Criterion Weight	Criterion Score
Vehicle Demo	0,229	0,9
Simulator Demo	0,066	1
Experienceability	0,232	1
Usability	0,052	1
Pull Paradigm	0,275	1
Market acceptance	0,020	0,7
Implementation	0,125	0,8
Overall score	**0,946**	

Car2X-Enabled Charging Station for Electric Cars		
Criterion	Criterion Weight	Criterion Score
Vehicle Demo	0,229	0,8
Simulator Demo	0,066	0,8
Experienceability	0,232	0,4
Usability	0,052	0,9
Pull Paradigm	0,275	0,8
Market acceptance	0,020	0,7
Implementation	0,125	0,7
Overall score	**0,698**	

Figure A.2.: Detailed results of the AHP on use-case level.

Intelligent Start-Stop		
Criterion	Criterion Weight	Criterion Score
Vehicle Demo	0,229	0,3
Simulator Demo	0,066	0,6
Experienceability	0,232	0,2
Usability	0,052	0,8
Pull Paradigm	0,275	0,4
Market acceptance	0,020	0,8
Implementation	0,125	0,2
Overall score	**0,348**	

Right of Way Conflict Resolution		
Criterion	Criterion Weight	Criterion Score
Vehicle Demo	0,229	0,3
Simulator Demo	0,066	0,8
Experienceability	0,232	0,4
Usability	0,052	1
Pull Paradigm	0,275	0,5
Market acceptance	0,020	0,3
Implementation	0,125	0,1
Overall score	**0,423**	

Environmental Displays Optimized for Viewer Perspective		
Criterion	Criterion Weight	Criterion Score
Vehicle Demo	0,229	0,7
Simulator Demo	0,066	0,8
Experienceability	0,232	1
Usability	0,052	1
Pull Paradigm	0,275	0,6
Market acceptance	0,020	0,4
Implementation	0,125	0,5
Overall score	**0,733**	

Virtual Agents for Environmental Displays		
Criterion	Criterion Weight	Criterion Score
Vehicle Demo	0,229	0,9
Simulator Demo	0,066	0,7
Experienceability	0,232	1
Usability	0,052	0,8
Pull Paradigm	0,275	0,4
Market acceptance	0,020	0,5
Implementation	0,125	0,3
Overall score	**0,684**	

Context-Aware Environmental Displays		
Criterion	Criterion Weight	Criterion Score
Vehicle Demo	0,229	0,9
Simulator Demo	0,066	0,9
Experienceability	0,232	1
Usability	0,052	0,8
Pull Paradigm	0,275	1
Market acceptance	0,020	0,5
Implementation	0,125	0,8
Overall score	**0,925**	

Multiple Reality Environmental Displays		
Criterion	Criterion Weight	Criterion Score
Vehicle Demo	0,229	0,4
Simulator Demo	0,066	0,8
Experienceability	0,232	1
Usability	0,052	0,8
Pull Paradigm	0,275	0,7
Market acceptance	0,020	0,5
Implementation	0,125	0,1
Overall score	**0,634**	

C2X Like Me Button		
Criterion	Criterion Weight	Criterion Score
Vehicle Demo	0,229	0,5
Simulator Demo	0,066	0,7
Experienceability	0,232	0,8
Usability	0,052	0,4
Pull Paradigm	0,275	0,7
Market acceptance	0,020	0,8
Implementation	0,125	0,3
Overall score	**0,614**	

Eco-Score Communities		
Criterion	Criterion Weight	Criterion Score
Vehicle Demo	0,229	0,4
Simulator Demo	0,066	0,8
Experienceability	0,232	0,4
Usability	0,052	0,1
Pull Paradigm	0,275	0,5
Market acceptance	0,020	0,7
Implementation	0,125	0,4
Overall score	**0,444**	

Figure A.3.: Detailed results of the AHP on use-case level.

Ad-hoc Rides		
Criterion	Criterion Weight	Criterion Score
Vehicle Demo	0,229	0,4
Simulator Demo	0,066	0,4
Experienceability	0,232	0,5
Usability	0,052	0,4
Pull Paradigm	0,275	0,5
Market acceptance	0,020	0,7
Implementation	0,125	0,4
Overall score	**0,457**	

Linking Environmental Displays and Social Networks		
Criterion	Criterion Weight	Criterion Score
Vehicle Demo	0,229	0,7
Simulator Demo	0,066	0,7
Experienceability	0,232	0,8
Usability	0,052	0,6
Pull Paradigm	0,275	0,8
Market acceptance	0,020	0,7
Implementation	0,125	0,8
Overall score	**0,758**	

Majority Decisive Environmental Displays		
Criterion	Criterion Weight	Criterion Score
Vehicle Demo	0,229	0,1
Simulator Demo	0,066	0,8
Experienceability	0,232	0,8
Usability	0,052	0,9
Pull Paradigm	0,275	0,5
Market acceptance	0,020	0,7
Implementation	0,125	0,6
Overall score	**0,535**	

Visual Bookmarking in Environmental Displays		
Criterion	Criterion Weight	Criterion Score
Vehicle Demo	0,229	0,6
Simulator Demo	0,066	0,9
Experienceability	0,232	0,8
Usability	0,052	0,9
Pull Paradigm	0,275	0,7
Market acceptance	0,020	0,6
Implementation	0,125	0,7
Overall score	**0,722**	

Live Video Feeds		
Criterion	Criterion Weight	Criterion Score
Vehicle Demo	0,229	0,8
Simulator Demo	0,066	0,8
Experienceability	0,232	0,7
Usability	0,052	0,7
Pull Paradigm	0,275	0,7
Market acceptance	0,020	0,4
Implementation	0,125	0,4
Overall score	**0,686**	

Microscopic Parking Lot Assistance		
Criterion	Criterion Weight	Criterion Score
Vehicle Demo	0,229	0,1
Simulator Demo	0,066	0,9
Experienceability	0,232	0,7
Usability	0,052	0,7
Pull Paradigm	0,275	0,9
Market acceptance	0,020	0,7
Implementation	0,125	0,7
Overall score	**0,631**	

Visibility Enhancement for Bicycles		
Criterion	Criterion Weight	Criterion Score
Vehicle Demo	0,229	0,9
Simulator Demo	0,066	0,9
Experienceability	0,232	0,9
Usability	0,052	0,8
Pull Paradigm	0,275	1
Market acceptance	0,020	0,9
Implementation	0,125	0,9
Overall score	**0,922**	

E-Bike Combined Battery and Communication Module		
Criterion	Criterion Weight	Criterion Score
Vehicle Demo	0,229	0,9
Simulator Demo	0,066	0,1
Experienceability	0,232	1
Usability	0,052	0,8
Pull Paradigm	0,275	0,9
Market acceptance	0,020	0,9
Implementation	0,125	1
Overall score	**0,877**	

Figure A.4.: Detailed results of the AHP on use-case level.

Bibliography

Adler, C., Eichler, S., Kosch, T., Schroth, C., and Strassberger, M. (2006). Self-organized and Context-Adaptive Information Diffusion in Vehicular Ad Hoc Networks. In *3rd International Symposium on Wireless Communication Systems (ISWCS 06)*, pages 307–311, Valencia, Spain.

AKTIV Consortium (2006). PRESS RELEASE: AKTIV: An Active Approach to Mobility. Available at http://www.aktiv-online.org/deutsch/aktuell-pressemitteilungen.html. Technical report.

Appert, P.-Y., Soussain, G., and Hochede, E. (1998a). CONCERT - TR 1013 - Results and Main Findings on Integrated Payment and Information Integration: Marseilles Pilot, available at http://cordis.europa.eu/telematics/tap_transport/research/projects/concert.html, last accessed 24.11.2013. Technical report.

Appert, P.-Y., Soussain, G., and Hochede, E. (1998b). CONCERT - TR 1013 - Results and Main Findings on Pricing & Restraint: Barcelona, Bristol, Thessaloniki & Trondheim Pilots, available at http://cordis.europa.eu/telematics/tap_transport/research/projects/concert.html, last accessed 24.11.2013. Technical report.

Assenmacher, S. (2009). simTD: Field Operational Test for Determining the Effecitveness of Cooperate Systems. In *16th World Congress on Intelligent Transport Systems*, Stockholm, Sweden.

Bajaj, L., Takai, M., Ahuja, R., and Tang, K. (1999). Glomosim: A Scalable Network Simulation Environment. *Compare A Journal Of Comparative Education*, 28(1):154 – 161.

Baldessari, R., Bödekker, B., Brakemeier, A., Deegener, M., Festag, A., Franz, W., Hiller, A., Kellum, C., Kosch, T., Kovacs, A., Lenardi, M., Lübke, A., Menig, C., Peichl, T., Roeckl, M., Seeberger, D., Strassberger, M., Stratil, H., Vögel, H.-J., Weyl, B., and Zhang, W. (2007). CAR 2 CAR Communication Consortium Manifesto, available at http://www.car-to-car.org, last accessed 24.11.2013. Technical report.

Bankosegger, D., Fuchs, S., and Frötscher, A. (2010). COOPERS: Deliverable D16-IR 8200/8500: Final Report for Disseminating the Demonstration Achievements, Revised Business Development & Roll Out Strategy, available at http://www.coopers-ip.eu/index.php?id=150, last accessed 24.11.2013. Technical report.

Barr, R., Haas, Z. J., and van Renesse, R. (2005a). JiST: an efficient approach to simulation using virtual machines. *Software: Practice and Experience*, 35(6):539–576.

Barr, R., Haas, Z. J., and van Renesse, R. (2005b). Scalable Wireless Ad hoc Network Simulation. In Wu, J., editor, *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad hoc Wireless, and Peer-to-Peer Networks*, chapter 19, pages 297 – 311. Auerbach Publications.

Behrisch, M. and Bieker, L. (2011). SUMO-Simulation of Urban MObility-an Overview. In *SIMUL 2011, The Third International Conference on Advances in System Simulation*, pages 55 – 60, Barcelona, Spain.

Braess, H.-H. (1995). Prometheus - A Contribution to a Comprehensive Concept for Future Road Traffic. In Pauwelussen, J. and Pacejka, H. B., editors, *Smart Vehicles*, pages 3 – 36. Swets & Zeitlinger.

Brandenburg, K., Braun, A., Benz, T., Diekmann, T., Fritz, H., Künzel, G., Irion, J., Lanfranco, C., Lorei, M., Martini, S., Schulze, M., Schwarzhaupt, A., and Vollmer, D. (2000). CHAUFFEUR 2: Deliverable D1: System Requirements, System Design and Safety Concept. Technical report.

Cao, Y., Mahr, A., Castronovo, S., and Müller, C. (2009). On Timing and Modality Choice with Local Danger Warnings for Drivers. In Schmidt, A., Dey, A., Seder, T., Juhlin, O., and Kern, D., editors, *Proceedings of the 1st International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2009)*, pages 75–78, Essen, Germany. ACM.

Cao, Y., Mahr, A., Castronovo, S., Theune, M., Stahl, C., and Müller, C. (2010). Local Danger Warnings for Drivers: The Effect of Modality and Level of Assistance on Driver Reaction. In Rich, C., Yang, Q., Cavazza, M., and Zhou, M. X., editors, *Proceedings of the 2010 International Conference on Intelligent User Interfaces (IUI 2010)*, Hong Kong, China. ACM.

Car 2 Car Communication Consortium (2011). Memorandum of Understanding for OEMs within the CAR 2 CAR Communication Consortium on Deployment Strategy for cooperative ITS in Europe.

Castronovo, S. and Endres, C. (2010). Interface Outside: Extending the V2X Communication Framework for Vulnerable Road User Protection. In *Adjunct Proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2010)*, pages 2–2, Pittsburgh, PA, USA.

Castronovo, S., Endres, C., Fabro, T. D., Schnabel, N., and Müller, C. (2011a). Multimodal Conspicuity-Enhancement for E-Bikes (Demo): the Hybrid Reality Model of Environment Transformation. In Pu, P., Pazzani, M. J., André, E., and Riecken, D., editors, *Proceedings of the 2011 International Conference on Intelligent User Interfaces (IUI 2011)*, pages 433–434, Palo Alto, CA, USA. ACM.

Castronovo, S., Kunz, B., and Müller, C. (2012). A Combined Uniform and Heuristic Search Algorithm for Maintaining Shortest Paths on Fully Dynamic Graphs. In *4th International Conference on Agents and Artificial Intelligence (ICAART 2012)*, Vilamoura, Algarve, Portugal.

Castronovo, S., Kunz, B., and Müller, C. (2013a). Combining Uniform and Heuristic Search: Solving DSSSP with Restricted Knowledge of Graph Topology. In Filipe, J. and Fred, A., editors, *Agents and Artificial Intelligence*, pages 173–187. Springer Berlin Heidelberg.

Castronovo, S., Mahr, A., and Endres, C. (2010a). On In-Car User Interfaces for Car-2-X Pull-Applications: Design Considerations for HMIs. In *Adjunct proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications. International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI-2010)*, pages 1–1, Pittsburgh, PA, USA.

Castronovo, S., Mahr, A., and Müller, C. (2010b). Multimodal Dialog in the Car: Combining Speech and Turn-And-Push Dial to Control Comfort Functions. In *Proceedings of Interspeech 2010. Conference in the Annual Series of Interspeech Events (INTERSPEECH-2010)*, pages 510–513, Makuhari, Japan. ISCA, Makuhari, Japan.

Castronovo, S., Mahr, A., and Müller, C. (2013b). What, Where, and When? Intelligent Presentation Management for Automotive Human Machine Interfaces and Its Application. In Yamamoto, S., editor, *Human Interface and the Management of Information. Information and Interaction for Health, Safety, Mobility and Complex Environments, Lecture Notes in Computer Science vol. 8017*, pages 460 – 469. Springer Berlin Heidelberg.

Castronovo, S., Mitrevska, M., and Müller, C. (2011b). Addressing Road Rage: The Effect of Social Awareness in Every Day Traffic Situations – A User Study. In *Adjunct proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications. International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2011)*, pages 43–44, Salzburg, Austria.

Charles E. Perkins, E. M. R. (1997). Ad-hoc On-Demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90 – 100.

Cicerone, S., Stefano, G. D., Frigioni, D., and Nanni, U. (2003). A Fully Dynamic Algorithm for Distributed Shortest Paths. *Theoretical Computer Science*, 297:1–3.

Clausen, T. and Jacquet, P. (2003). Optimized Link State Routing Protocol, available at http://tools.ietf.org/html/rfc3626, last visited 23.11.2013. Technical report.

Coltun, R., Ferguson, D., Moy, J., and Lindem, A. (2008). OSPF for IPv6, available at http://tools.ietf.org/html/rfc5340, last visited 23.11.2013. Technical report.

Cormen, T. H., Leiserson, C. E., and Rivest, R. L. (2000a). Dijkstra's Algorithm. In *Introduction to Algorithms*, chapter 25.2, pages 527 – 531. The MIT Press, first edition.

Cormen, T. H., Leiserson, C. E., and Rivest, R. L. (2000b). Fibonacci Heaps. In *Introduction to Algorithms*, chapter 21, pages 420 – 439. The MIT Press, first edition.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms*. The MIT Press, third edition.

Day, J. D. and Zimmermann, H. (1983). The OSI Reference Model. *Proceedings of the IEEE*, 71(12):1334–1340.

Del Fabro, T. (2013). *Heuristic and Context-Aware Least-Weighted Circuit Discovery on Unknown Fully Dynamic Graphs*. Master's thesis, Department of Computer Science, Saarland University.

Demetrescu, C. and Italiano, G. F. (2003). A New Approach to Dynamic All Pairs Shortest Paths. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, STOC '03, pages 159–166, New York, NY, USA. ACM.

Dijkstra, E. W. (1959). A Note on Two Problems in Connection with Graphs. *Numerische Mathematik*, 1:269–271.

Ehrlich, J., Hautière, N., Kulla, M., Lovas, T., Oriz, D., Ahlers, F., Schendzielorz, T., Lautier, A., Marco, S., Spence, A., Mann, M., Vivo, G., and Scholliers, J. (2010). SAFESPOT Deliverable D.2.5.2: Final Report: Results on Test and Validation, available at http://www.safespot-eu.org/documents/SF_D2.5.2_Results_on_test_and_validation_v1.15.pdf, last accessed 24.11.2013. Technical report.

Eichler, S., Schroth, C., Kosch, T., and Strassberger, M. (2006). Strategies for Context-Adaptive Message Dissemination in Vehicular Ad Hoc Networks. In *Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services*, pages 1–9, San Jose, CA, USA.

Endres, C. (2012). *PresTK: Situation-Aware Presentation of Messages and Infotainment Content for Drivers*. PhD thesis, Department of Computer Science, Saarland University.

Endres, C. and Castronovo, S. (2010). Intelligent Environment Car: A New Perspective. In *Adjunct proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications. International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI-2010)*, pages 9–9, Pittsburgh, PA, USA.

Enkelmann, W., Bechler, M., Berninger, H., Bohnert, T. M., Brandes, R., Caliskan, M., Cosenza, S., Renesas, J. D., Häusler, F., Hayashi, M., Hinsberger, A., Lin, L., Irion, J., Kernchen, R., Kratzsch, C., Lübke, A., Mika, A., Nygren, N., Paulus, I., Radusch, I., and Stöbe, M. (2008). PRE-DRIVE C2X; Deliverable D4.1; Detailed Description of Selected Use-cases and Corresponding Technical Requirements. Technical report.

European Telecommunications Standards Institute (ETSI) (2009). ETSI TR 102 638 v1.1.1: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions. Technical report.

European Telecommunications Standards Institute (ETSI) (2010a). ETSI EN 302 665 v1.1.1: Intelligent Transport Systems (ITS); Communications Architecture. Technical report.

European Telecommunications Standards Institute (ETSI) (2010b). ETSI TS 102 637-3 v1.1.1: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service. Technical report.

European Telecommunications Standards Institute (ETSI) (2011a). ETSI TR 102 863 v1.1.1: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM); Rationale for and Guidance on Standardization. Technical report.

European Telecommunications Standards Institute (ETSI) (2011b). ETSI TS 102 637-2 v1.2.1: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service. Technical report.

Feld, M. (2011). *A Speaker Classification Framework for Non-intrusive User Modeling: Speech-based Personalization of In-Car Services*. Phd thesis, Department of Computer Science, Saarland University.

Fellendorf, M. and Vortisch, P. (2010). Microscopic Traffic Flow Simulator VISSIM. In *Fundamentals of Traffic Simulation*, pages 63 – 93. Springer Berlin Heidelberg.

Festag, A., Baldessari, R., Zhang, W., and Le, L. (2009). CAR-2-X Communication SDK - A Software Toolkit for Rapid Application Development and Experimentations. In *Proceedings of the IEEE Vehicular Networking and Applications Wokshop - Future Wireless Technologies for Vehicle Infrastructure Integration (VII) Applications (VehiMobil), co-located with IEEE International Conference on Communications (ICC)*, Dresden, Germany.

Frigioni, D., Spaccamela, A. M., and Nanni, U. (2000). Fully Dynamic Algorithms for Maintaining Shortest Path Trees. *Algorithms*, 34(2):251–281.

Füssler, H., Mauve, M., Hartenstein, H., Käsemann, M., and Vollmer, D. (2002). A Comparison of Routing Strategies for Vehicular Ad-Hoc Networks, available at http://www.cn.uni-duesseldorf.de/publications/library/Fuessler2002b.pdf, last accessed 24.11.2013. Technical Report TR-02-003, Department of Computer Science, University of Mannheim.

Gajananan, K., Nantes, A., Miska, M., Nakasone, A., and Prendinger, H. (2013). An Experimental Space for Conducting Controlled Driving Behavior Studies based on a Multiuser Networked 3D Virtual Environment and the Scenario Markup Language. *IEEE Transactions on Human-Machine Systems*, 43(4):345–358.

Gajananan, K. and Prendinger, H. (2013). National Institute of Informatics (NII): A Short Tutorial of the Scenario Markup Language, available at www.nii.ac.jp/TechReports/13-001E.pdf, last accessed 24.11.2013. Technical report, National Institute of Informatics (NII), Tokyo, Japan.

Gast, M. (2005). *802.11 Wireless Networks: The Definitive Guide, Second Edition.* O'Reilly Media.

Gorius, M., Shuai, Y., and Herfet, T. (2012). Dynamic media streaming over wireless and mobile IP networks. In *Consumer Electronics - Berlin (ICCE-Berlin), 2012 IEEE International Conference on*, pages 158–162.

Gorius, M., Shuai, Y., and Herfet, T. (2013). Dynamic media streaming with predictable reliability and opportunistic TCP-friendliness. In *Consumer Communications and Networking Conference (CCNC), 2013 IEEE*, pages 216–223.

Granelli, F., Boato, G., and Kliazovich, D. (2006). MORA: a Movement-Based Routing Algorithm for Vehicle Ad Hoc Networks. In *Proceedings of the 1st IEEE Workshop on Automotive Networking and Applications (AutoNet)*, San Francisco, California, USA.

Granelli, F., Boato, G., Kliazovich, D., and Vernazza, G. (2007). Enhanced GPSR Routing in Multi-Hop Vehicular Communications through Movement Awareness. *IEEE Communications Letters*, 11(10):781–783.

Grimm, J. and Grimm, W. (2007). *Hänsel und Gretel.* Hinstorff, first edition.

Haerri, J., Fiore, M., Filali, F., and Bonnet, C. (2011). Vehicular Mobility Simulation with VanetMobiSim. *Simulation*, 87(4):275–300.

Hedrick, C. (1988). Routing Information Protocol, available at http://tools.ietf.org/html/rfc1058, last accessed 24.11.2013. Technical report.

Hrizi, F. and Filali, F. (2010). An Integrated and Realistic Simulation Platform for Vehicular Networks. In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools)*, SIMUTools '10, pages 37–37, Torremolinos, Malaga, Spain. ICST (Institute for Computer Sciences, Social Informatics and Telecommunications Engineering).

ISO/IEC (1994). Information Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model, available at http://standards.iso.org/ittf/PubliclyAvailableStandards/s020269_ISO_IEC_7498-1_1994(E).zip, last accessed 24.11.2013. Technical report.

Jerbi, M., Senouci, S.-M., Meraihi, R., and Ghamri-Doudane, Y. (2007). An Improved Vehicular Ad Hoc Routing Protocol for City Environments. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 3972–3979, Glasgow, Scotland.

Johnson, D. B. and Maltz, D. A. (1996). Dynamic Source Routing in Ad Hoc Wireless Networks. In Imielinski, T. and Korth, H. F., editors, *Mobile Computing*, pages 153–181. Kluwer Academic Publishers.

Karp, B. and Kung, H. T. (2000). GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, MobiCom '00, pages 243–254, New York, NY, USA. ACM.

Kern, D., Mahr, A., Castronovo, S., and Müller, C. (2010). Making Use of Drivers Glances onto the Screen for Explicit Gaze-Based Interaction. In *Proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2010)*, pages 110–113, Pittsburgh, PA, USA. Carnegie Mellon University, ACM.

Kern, D., Müller, M., Schneegaß, S., Wolejko-Wolejszo, L., and Schmidt, A. (2008). CARS–Configurable Automotive Research Simulator. In *Automotive User Interfaces and Interactive Applications–AUIIA'08. Workshop at Mensch und Computer, Lübeck*, Lübeck, Germany.

Kihl, M., Sichitiu, M., Ekeroth, T., and Rozenberg, M. (2007). Reliable Geographical Multicast Routing in Vehicular Ad-Hoc Networks. In Boavida, F., Monteiro, E., Mascolo, S., and Koucheryavy, Y., editors, *Proceedings of the 5th international conference on Wired/Wireless Internet Communications (WWIC 07)*, WWIC '07, pages 315–325, Coimbra, Portugal. Springer Berlin Heidelberg.

Koenig, S., Likhachev, M., and Furcy, D. (2004). Lifelong Planning A*. *Artificial Intelligence*, 155(1-2):93–146.

Kompfner, P. (2010). D.CVIS.1.3 Final Activity Report, available at http://www.cvisproject.org/download/Deliverables, last accessed 24.11.2013. Technical report.

Kosch, T. (2005). *Situationsadaptive Kommunikation in Automobilen Ad-hoc Netzen*. PhD thesis, Department of Computer Science, Technische Universität München.

Kosch, T., Schroth, C., Strassberger, M., and Bechler, M. (2012a). A.2.1.2 Aktiv. In Kosch, T., Schroth, C., Strassberger, M., and Bechler, M., editors, *Automotive Internetworking*, pages 365 – 366. John Wiley & Sons, Ltd., first edit edition.

Kosch, T., Schroth, C., Strassberger, M., and Bechler, M. (2012b). *Automotive Internetworking*. John Wiley & Sons, Ltd., first edit edition.

Krajzewicz, D. (2010). Traffic Simulation with SUMO - Simulation of Urban Mobility. In Barceló, J., editor, *International Series in Operations Research and Management Science: Fundamentals of Traffic Simulation*, pages 269 – 294. Springer New York.

Krajzewicz, D., Blokpoel, R. J., Cartolano, F., Cataldi, P., Gonzalez, A., Lazaro, O., Leguay, J., Lin, L., Maneros, J., and Rondinone, M. (2010). iTETRIS-A System for the Evaluation of Cooperative Traffic Management Solutions. *Advanced Microsystems for Automotive Applications 2010*, pages 399–410.

Krajzewicz, D., Hertkorn, G., Rösel, C., and Wagner, P. (2002). SUMO (Simulation of Urban MObility); An Open-Source Traffic Simulation. In Al-Akaidi, A., editor, *Proc. of the 4th Middle East Symposium on Simulation and Modelling*, pages 183 – 187, Sharjah, UAE.

Krauß, S. (1998). *Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics*. Phd thesis, Department of Computer Science, University of Cologne.

Kruppa, M. (2006). *Migrating Characters: Effective User Guidance in Instrumented Environments*. Phd thesis, Department of Computer Science, Saarland University.

Kunz, B. (2011). *A Combined Uniform and Heuristic Search Algorithm for Maintaining Shortest Paths on Unknown Fully Dynamic Graphs*. Master's thesis, Department of Computer Science, Saarland University.

Li, J., Jannotti, J., De Couto, D. S. J., Karger, D. R., and Morris, R. (2000). A Scalable Location Service for Geographic Ad Hoc Routing. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom 00)*, pages 120–130, Boston, MA, USA. ACM.

Lochert, C., Caliskan, M., Scheuermann, B., Barthels, A., Cervantes, A., and Mauve, M. (2005a). Multiple Simulator Interlinking Environment for Inter Vehicle Communication. In *VANET 2005: Proceedings of the Second ACM International Workshop on Vehicular Ad Hoc Networks*, pages 87–88, Cologne, Germany.

Lochert, C., Hartenstein, H., Tian, J., Füssler, H., Hermann, D., and Mauve, M. (2003). A Routing Strategy for Vehicular Ad Hoc Networks in City Environments. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, volume 2000, pages 156 – 161.

Lochert, C., Mauve, M., Füssler, H., and Hartenstein, H. (2005b). Geographic routing in city scenarios. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(1):69.

Mahr, A., Castronovo, S., Math, R., and Müller, C. (2013). Assisting the Driver with Distance Estimation: Usability Evaluation of Graphical Presentation Alternatives for Local Traffic Events. In Kotzé, P., Marsden, G., Lindgaard, G., Wesson, J., and Winckler, M., editors, *Human-Computer Interaction – INTERACT 2013*, volume 8119 of *Lecture Notes in Computer Science*, pages 520 – 537. Springer Berlin Heidelberg.

Majstorovic, D. (2013). *Automotive User Interface Adaptation for People with Visual Impairment.* Master's thesis, Department of Computer Science, Saarland University.

Mallanda, C., Suri, A., and Kunchakarra, V. (2005). Simulating wireless sensor networks with OMNeT++, avalaible from https://www.omnetpp.org/publications, last accessed 24.11.2013.

Mangharam, R., Weller, D. S., Stancil, D. D., Rajkumar, R., and Parikh, J. S. (2005). GrooveSim: A Topography-Accurate Simulator for Geographic Routing in Vehicular Networks. In *Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks - VANET '05*, page 59, New York, NY, USA. ACM Press.

Math, R., Mahr, A., Moniri, M. M., and Müller, C. (2012). OpenDS: A new open-source driving simulator for research. In *Adjunct Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Appilcations*, pages 7–8, Portsmouth, NH, USA.

McDonald, M. and Richter, T. (2008). COOPERS: Deliverable D5 – A 2100 Summary Report on Safety Standards, Indicators to Improve the Safety Roads, avalaible at http://www.coopers-ip.eu/index.php?id=150, last accessed 24.11.2013. Technical report.

Mcquillan, J., Richer, I., and Rosen, E. (1980). The New Routing Algorithm for the ARPANET. *IEEE Transactions on Communications*, 28(5):711 – 719.

Misra, S. and Oommen, B. (2004). Stochastic Learning Automata-Based Dynamic Algorithms for the Single Source Shortest Path Problem. In Orchard, B., Chunsheng, Y., and Moonis, A., editors, *Proceedings of the 17th International Conference on Innovations in Applied Artificial Intelligence*, IEA/AIE'2004, pages 239–248, Ottawa, Canada. Springer Berlin Heidelberg.

Mitrevska, M., Castronovo, S., and Müller, C. (2012). Physical and Spiritual Proximity: Linking Car2X Communication with Online Social Networks. In Kun, A. L., Boyle, L. N., Reimer, B., and Riener, A., editors, *Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 12)*, pages 249–256, Portsmouth, NH, USA. ACM, New York, NY, USA.

Mo, Z., Zhu, H., Makki, K., and Niki, P. (2006). Muru: A Multi-Hop Routing Protocol for Urban Vehicular Ad Hoc Networks. In *Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services (IEEE Conference Proceedings)*, pages 1–8, San Jose, CA, USA.

Moy, J. (1998). OSPF Version 2, available at http://www.ietf.org/rfc/rfc2328.txt, last accessed 24.11.2013. Technical report.

Msadaa, I. C., Cataldi, P., and Filali, F. (2010). A Comparative Study Between 802.11p and Mobile WiMAX-Based V2I Communication Networks. In *Fourth International*

Conference on Next Generation Mobile Applications, Services and Technologies (NG-MAST 2010), pages 186–191, Amman, Jordan. IEEE.

Nannicini, G. and Liberti, L. (2008). Shortest Paths on Dynamic Graphs. *International Transactions in Operational Research*, 15(5):551–563.

Naumann, N., Schuenemann, B., and Radusch, I. (2009). VSimRTI - Simulation Runtime Infrastructure for V2X Communication Scenarios. In *Proceedings of the 16th World Congress and Exhibition on Intelligent Transport Systems and Services*, Stockholm, Sweden.

Naumov, V. and Gross, T. R. (2007). Connectivity-Aware Routing (CAR) in Vehicular Ad Hoc Networks. In *Proceedings of the 26th IEEE International Conference on Computer Communications (IEEE INFOCOM)*, pages 1919–1927, Anchorage, Alaska, USA.

Nesselrath, R. and Feld, M. (2013). Towards a Cognitive Load Ready Multimodal Dialogue System for In-Vehicle Human-Machine Interaction. In Terken, J., Riener, A., Schroeter, R., and Osswald, S., editors, *Adjunct Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2013)*, pages 49–52, Eindhoven, The Netherlands.

Nzouonta, J., Rajgure, N., and Borcea, C. (2009). VANET Routing on City Roads Using Real-Time Vehicular Traffic Information. *IEEE Transactions on Vehicular Technology*, 58(7):3609–3626.

Passmann, C., Schaaf, G., and Naab, K. (2009). Sichere Intelligente Mobilität Testfeld Deutschland Ausgewählte Funktionen, available at http://www.simtd.de, last accessed 24.11.2013. Technical report.

Pfliegl, R. (2010). COOPERS: Project Presentation, available at http://www.coopers-ip.eu/index.php?id=150, last accessed 24.11.2013. Technical report.

Piorkowski, M. and Raya, M. (2008). TraNS: Realistic Joint Traffic and Network Simulator for VANETs. *ACM SIGMOBILE Mobile Computing and Communications Review*, 12(1):31 – 33.

Popescu-Zeletin, R., Radusch, I., and Rigani, M. A. (2010). Applications of Vehicular Communication. In *Vehicular-2-X Communication: State-of-the-Art and Research in Mobile Vehicular Ad hoc Networks*, chapter 2, pages 5 – 38. Springer Berlin Heidelberg.

Réolon, F. (2007). *Kontextsensitive, P2P-basierte Erkennung und Darstellung von Gefahren in einer Motorradumgebung*. Diploma thesis, Department of Computer Science, Saarland University.

Richter, A. (2005). Wolfsburger Welle. In *Geschwindigkeitsvorgabe an Lichtsignalanlagen: Technische Aspekte Und Volkswirtschaftlicher Nutzen*, chapter 5.2, pages 34 – 40. Deutscher Universitäts-Verlag/GWV Fachverlage GmbH, Wiesbaden.

Riley, G. (2003a). Simulation of Large Scale Networks II: Large-Scale Network Simulations with GTNetS. In *Proceedings of the 2003 Winter Simulation Conference*, pages 676 –684, Washington, DC, USA.

Riley, G. (2003b). The Georgia Tech Network Simulator. In *Proceedings of the ACM SIGCOMM workshop on Models, methods and tools for reproducible network research (MoMeTools 03)*, pages 5 – 12. ACM.

Royer, E. E. M. and Toh, C.-K. (1999). A Review of Current Routing Protocols for A d Hoc Mobile Wireless Networks. *Personal Communications, IEEE*, 6(2):46–55.

Russell, S. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*, volume 60 of *Prentice Hall Series In Artificial Intelligence*. Prentice Hall.

Saaty, R. W. (1987). The analytic hierarchy process–what it is and how it is used. *Mathematical Modelling*, 9:161–176.

Schaaf, G., Passmann, C., Mönnich, J., Haug, M., and Niewels, F. (2011). A Method for Evaluation and Selection of C2X Communication Functions. In *8th ITS European Congress*, Lyon, France.

Schulze, M., Kosch, T., Kulp, I., Benz, T., Tomatis, A., Radusch, I., Noecker, G., Andreone, L., Kessel, T., and Klessen, C. (2010). PRE-DRIVE C2X; Deliverable D0.3; Final Report, available at http://www.drive-c2x.eu, last accessed 24.11.2013. Technical report.

Schulze, M., Mäkinen, T., Irion, J., Flament, M., and Kessel, T. (2008). IP PReVENT: Preventive and Active Safety Applications Integrated Project: IP_D15: Final report, available at http://ec.europa.eu, last accessed 24.11.2013. IP Deliverable FP6-507075.

Schünemann, B. (2011). V2X simulation runtime infrastructure VSimRTI: An assessment tool to design smart traffic management systems. *Computer Networks*, 55(14):3189–3198.

Schwartz, T., Castronovo, S., and Endres, C. (2010). A GPS-less Method to Find Your Parked Car. In *Adjunct proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2010)*, pages 26–26, Pittsburgh, PA, USA.

Seet, B.-c. C., Liu, G., Lee, B.-s. S., Foh, C.-h. H., Wong, K.-j. K.-J., and Lee, K.-k. (2004). A-STAR: A Mobile Ad Hoc Routing Strategy for Metropolis Vehicular Communications. In Mitrou, N., Kontovasilis, K., Rouskas, G. N., Iliadis, I., and Merakos, L., editors, *Proceedings of the Third International IFIP-TC6 Networking Conference*, volume 3042 of *Lecture Notes in Computer Science*, pages 989–999, Athens, Greece. Springer Berlin Heidelberg.

Siemens (1983). Siemens AG and Volkswagenwerk AG, Press Release: Autoscout - Leit- und Informationssystem für den Straßenverkehr, available at http://www.vwpix.org/archiv, last accessed 24.11.2013.

Stahlmann, R., Festag, A., Tomatis, A., Radusch, I., and Fischer, F. (2011). Starting European Field Tests for CAR-2-X Communication: The DRIVE C2X Framework. In *Proceedings for the 18th ITS World Congress Exhibition*, Orlando, FL, USA.

Stepanov, I., Marron, P. J., and Rothermel, K. (2005). Mobility Modeling of Outdoor Scenarios for MANETS. In *Proceedings of the 38th Annual Symposium on Simulation (ANSS 05)*, pages 312–322, San Diego, CA, USA. IEEE Computer Society.

Strandén, L., Ström Chalmers, E., and Uhlemann, E. (2008a). SAFER Project Report, Pre-Study no. AD4: Wireless Communications Vehicle-to-Vehicle and Vehicle-to-Infrastructure, available at http://www.chalmers.se/safer/EN/publications, last accessed 24.11.2013. Technical report.

Strandén, L., Uhlemann, E., and Ström, E. G. (2008b). State of the Art Survey of Wireless Vehicular Communication Projects. In *Proceedings of the 15th World Congress on Intelligent Transport Systems (ITS)*, volume 6, New York City, NY, USA.

Sun, W., Yamaguchi, H., Yukimasa, K., and Kusumoto, S. (2006). GVGrid: A QoS Routing Protocol for Vehicular Ad Hoc Networks. In *14th IEEE International Workshop on Quality of Service*, pages 130–139.

Symington, S. and Morse, K. (2000). IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Framework and Rules (IEEE Std 1516-2000). *IEEE Computer Society*.

Taleb, T., Sakhaee, E., Jamalipour, A., Hashimoto, K., Kato, N., and Nemoto, Y. (2007). A Stable Routing Protocol to Support ITS Services in VANET Networks. *IEEE Transactions on Vehicular Technology*, 56(6):3337 – 3347.

Treiber, M., Hennecke, A., and Helbing, D. (2000). Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805 – 1824.

Van de Velde, E., Blondia, C., and Campelli, L. (2006). REACT: Routing Protocol for Emergency Applications in Car-to-Car Networks using Trajectories. In *Proceedings of The Fifth Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, pages 145–151, Lipari, Italy.

Varga, A. (2001). The OMNeT++ Discrete Event Simulation System, available at http://www.omnetpp.org/download/docs/papers/esm2001-meth48.pdf, last accessed 24.11.2013. In *Proceedings of the European Simulation Multiconference (ESM 2001)*, Prague, Czech Republic.

Varga, A. (2010). OMNeT++. In Wehrle, K., Güneş, M., and Gross, J., editors, *Modeling and Tools for Network Simulation*, pages 35–59. Springer Berlin Heidelberg.

Varga, A. and Hornig, R. (2008). An Overview of the OMNeT++ Simulation Environment. In *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops (Simutools 08)*, pages 1 – 8, Marseille, France.

Walid Dabbous, C. H. (1988). PROMETHEUS: Vehicle to Vehicle Communications, available at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.22.2147, last accessed 24.11.2013.

Wang, S. and Huang, Y. (2012). NCTUns Distributed Network Emulator. *Internet Journal*, 4(2):61–94.

Wang, S.-Y. and Lin, C.-C. (2008). NCTUns 5.0: A Network Simulator for IEEE 802.11(p) and 1609 Wireless Vehicular Network Researches. In *Proceedings of the 2nd IEEE International Symposium on Wireless Vehicular Communications (WiVeC), co-located with the 68th IEEE Vehicular Technology Conference (VTC Fall)*, Calgary, Alberta, Canada.

Yu, D. and Ko, Y.-B. (2009). FFRDV: Fastest-Ferry Routing in DTN-Enabled Vehicular Ad Hoc Networks. In *11th International Conference on Advanced Communication Technology (ICACT 2009)*, pages 1410–1414, Phoenix Park, Korea.

Zeng, X., Bagrodia, R., and Gerla, M. (1998). GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks. In *Proceedings of the Twelfth Workshop on Parallel and Distributed Simulation (PADS 98)*, pages 154 – 161.

Zhang, X. and Riley, G. F. (2004). Bluetooth Simulations for Wireless Sensor Networks Using GTNetS. In *Proceedings of the the IEEE Computer Society's 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS 04)*, pages 375–382.

Zimdahl, W. (1983). *Wolfsburger Welle - Ein Projekt der VW Forschung.* Forschung Fahrzeugelektronik, Volkswagen AG.