Mobile Device Integration with V2X Communication

Stefan Diewald[2], Tim Leinmüller[1], Boris Atanassow[1*],
Lutz-Peter Breyer[1] and Matthias Kranz[3]
[1]DENSO AUTOMOTIVE Deutschland GmbH, Info-Safety Engineering Department, Germany,
[t.leinmueller|b.atanassow|l.breyer]@denso-auto.de
[2]Technische Universität München, Distributed Multimodal Information Processing Group, Munich, Germany,
stefan.diewald@tum.de
[3]Luleå University of Technology, Department of Computer Science, Electrical and Space Engineering, Luleå, Sweden,
matthias.kranz@ltu.se

Abstract

Message processing and data visualization are key technologies in a V2X system, influencing both real and perceived performance, and usability of a system. V2X safety applications require frequent exchange of position information by means of so called Cooperative Awareness Messages (CAMs). These messages are generated, encoded and sent by all vehicles. Corresponding actions have to be executed on the receiving vehicles. The in-vehicle V2X system can be divided into several components to accomplish these tasks. Our approach is based on a "two-component" system, consisting of a vehicle-integrated V2X communication unit (onboard unit, OBU) and a personal portable device (PPD), such as a smartphone or tablet PC.

Subject of this work is the investigation how to distribute the workload and functionality between the two components. Our goal is to find a flexible solution that maximizes the overall performance and reliability of the system. We investigate and compare several message processing approaches and try to combine the strengths of both components. To ensure comparability, tests are carried out on the same hardware platform. As a result, we present our final setup that can handle an up to 60 times higher message rate compared to other investigated solutions.

Keywords

Vehicle-to-X (V2X) communication, visualization, mobile device, onboard unit, smartphone, PPD.

Introduction

One of the major questions in the context of Vehicle-to-X (V2X) communication are possibilities of visualization and communication of information to vehicle drivers and vehicle passengers. This information is distributed via different message types, e.g. the periodically distributed Cooperative Awareness Messages (CAMs) (1), or event-driven messages such as the Decentralized Environmental Notification Messages (DENMs) (2). With the tremendously increasing availability of powerful personal portable devices (PPD), such as smartphones or tablet PCs, transferring some of the visualization and communication tasks to the PPD of the user becomes an interesting alternative to vehicle integrated systems[1,2].

[1]http://www.miniusa.com/top-feature-cooper-coupe-connected-nav.html, last accessed July 25, 2012.
[2]http://reviews.cnet.com/8301-13746_7-57330460-48/2011-frankfurt-auto-show-mercedes-iphone-integration/, last accessed July 25, 2012.

New mobile devices are released every few days, often featuring higher computing power and better displays. According to HTC, today's smartphone model lifetime is six to nine months[3]. In 2011, the smartphone sales reached 472 million units[4]. With these smartphones, "always connected, personal and contextual" is a good summarization for the current and future mobile experience[5]. The portable devices provide services, such as location-specific information, user-preferred multimedia content, mobile Internet access or comprehensive communication, anywhere anytime. Their ongoing development towards ubiquitous companions (3) opens also several possibilities in the automotive domain (4).

In contrast to PPDs, vehicles are evolving much slower. This is an important factor to consider for the introduction of advanced information and communication technology (ICT) like V2X. Using a cheap, personal and exchangeable mass-market mobile devices as head unit add-on or replacement can bridge the lifecycle gap between automotive and consumer devices (4). It could be sufficient to have a gateway to the V2X communication system integrated in the vehicle, and the mobile device contains the fast changing technology, but provides the host for evolving new services and more powerful applications and services.

In addition, by combining time-critical, low-delay V2X communication data with high-delay (e.g. several hundred milliseconds) data from Internet services via the mobile data link of the portable device (2G, 3G, 3.5G or 4G network), enriched and contextualized information can be provided to the driver and passengers. For example, when the V2X system reports a blocked road or traffic jam on the route, the application on the mobile device could automatically calculate an alternative route by querying the public transportation schedules. Furthermore, in safety critical situations, it is an advantage when drivers are accustomed to the HMI partially being represented by the mobile device. Since users (i.e. drivers) are in general more acquainted with the interaction and visualization paradigms of their mobile devices than with car systems, this can support safety while driving.

In our project, we have investigated such a setup. It consists of a vehicle-integrated V2X communication platform linked to an Android-based tablet PC. In this setup, it is essential to have an appropriate, that means fast and reliable, V2X message processing system that allows supplying the PPD with the required data. In this paper, we explore different approaches and discuss how the message processing can be distributed among the two components. Our key objective is to achieve good performance and reliability while keeping the system flexible and extensible. We provide quantitative results from our measurements allowing us to compare the performance of the different approaches.

The remainder of the paper is structured as follows: we first give a short summary of similar approaches and situate our approach in this context. We then present the setup of our system. In the next section, three different work splits are described in detail and the respective results are presented. We conclude with a summary of our findings and with implications on future work.

Related Work

Grimm provides a high level discussion of a similar setup (5). In Grimm's approach, the V2X features, including decoding and encoding of the V2X messages, are completely hosted by the smartphone, what makes it a mandatory part of the system. In contrast, our work analyzes the possibilities of work split between the components in detail.

The *CODAR viewer* is a situation-aware driver assistance system developed by Kranz et al. (6). It collects V2X data from road site units (RSUs) and displays the traffic information to the driver via the vehicle's HMI. By presenting real-time information about traffic incidents and the positions of other V2X equipped vehicles, the system creates cooperative awareness. That allows the driver to make more informed decisions, since s/he also has information about the context of other road users.

[3] http://money.cnn.com/2011/01/31/technology/new_smartphone/index.htm, last accessed January 11, 2012.
[4] http://www.gartner.com/it/page.jsp?id=1924314, last accessed July 25, 2012.
[5] http://www.appcelerator.com/company/survey-results/mobile-developer-report-january-2011/, last accessed January 13, 2012.

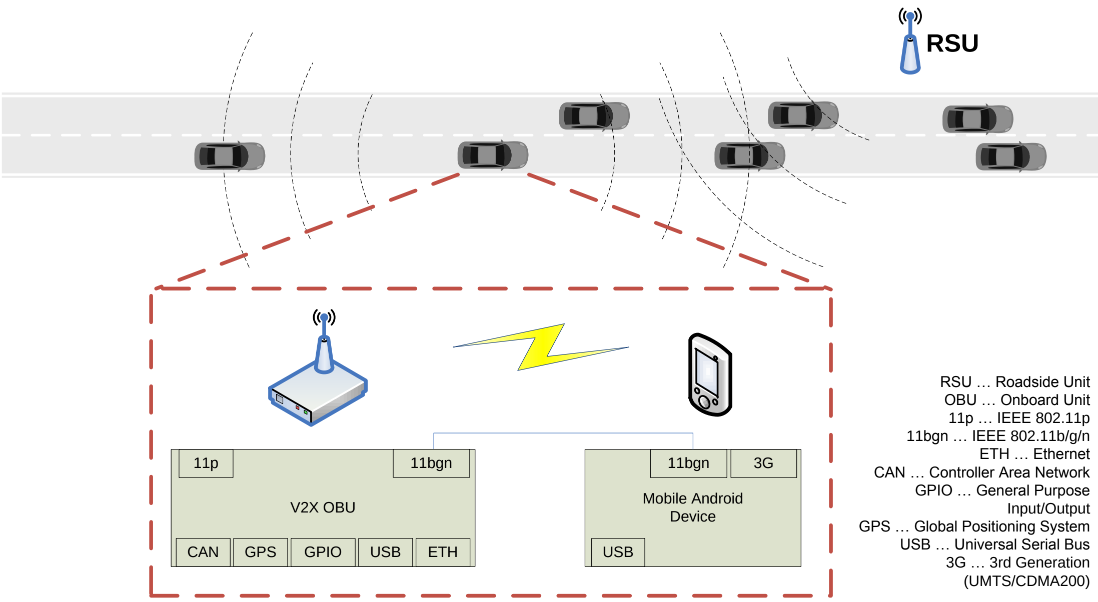

Fig. 1: System overview including a detailed view of our in-vehicle setup.

Löper et al. use a different combination of V2X communication and nomadic devices for their automated car prototype (7). In their approach, vehicle-to-infrastructure (V2I) communication is used for acquiring the current state and the time to the next phase change of traffic lights around the car. The user's portable device can communicate with the 3G equipped vehicle through its mobile data connection. This connection can, for example, be used for sending a "come here" command to the automated vehicle.

Summarizing related work, a detailed analysis of different workload splits in a two-component OBU-PPD system for V2X communication has not been done yet. Therefore, we analyze different options in the remainder of this paper.

System Overview

An overview of the system setup is depicted in Figure 1. The setup consists of a V2X communication onboard unit (V2X OBU) responsible for the communication with other V2X units, and an Android-based tablet PC as PPD.

Hardware elements

For reasons of simplicity, our prototype OBU consists of two physical devices, the DENSO Wireless Safety Unit (WSU) and a commercial, off-the-shelf IEEE 802.11b/g/n WLAN access point. The WSU features a PowerPC central processing unit (CPU) running at $400\ \mathrm{MHz}$, and has $128\ \mathrm{MB}$ random access memory (RAM). It supports IEEE 802.11p (8) for V2X communication. The utilized WLAN access point can either be run in access point (AP) mode which allows connecting WLAN clients to it, or in Ethernet adapter mode which allows a transparent connection of Ethernet devices to any other defined WLAN access point. We use a 7-inch Samsung Galaxy Tab GT-P1000 running Android 2.3.3. It is equipped with a $1\ \mathrm{GHz}$ single core ARM architecture CPU and has $512\ \mathrm{MB}$ RAM.

Connectivity between OBU and PPD

Today's mobile devices are supporting several wired and wireless communication technologies that could be used for connecting the PPD with the OBU. The most common technologies are WLAN (IEEE 802.11a/b/g/n), Bluetooth, and Universal Serial Bus (USB).

Since all three mentioned technologies support the Internet protocol (IP), we have chosen to use IP-based communication for our system. That way, we can use all three interconnection types. To be able to forward all incoming V2X messages, the connection has to support at least a transmission rate of $27\ \mathrm{Mbit/s}$ which is the maximum specified transmission rate for IEEE 802.11p. For slower connections, such as USB 1.1 which only supports up to $12\ \mathrm{Mbit/s}$, filtering would have to be applied on the OBU.

In order to combine data from Internet sources with V2X information coming from the OBU, the PPD has to be connected to the OBU and at the same time to the Internet. However, current mobile devices normally only allow using either mobile data or WLAN. A possible workaround is putting the PPD in the so called WiFi tethering mode. In this mode, the PPD acts as modem and WLAN access point. That way, the OBU's WLAN module could connect to the PPD which then would be connected to both, the Internet and the OBU.

For our evaluation of work splits, we have chosen WLAN. The access point is running in access point mode and the PPD is connected to it via IEEE 802.11g. The connection allows a maximum transmission rate of $54\ \mathrm{Mbit/s}$.

Evaluation of Work Splits

In a two-component system, work can be split in different ways. The split should take into account weaknesses and strengths of the individual components, as well as the options for interlinking the components.

We compare different work splits with respect to the following criteria:

- Performance - messages per second
- Flexibility - effort required to add/update message formats and types
- Battery usage - load on PPD

For benchmarking the work splits, we have chosen the decoding process of Cooperative Awareness Messages (CAMs) (1). The CAM format is defined in Abstract Syntax Notation One (ASN.1). CAMs are repeatedly broadcasted by all vehicles within the V2X network. They provide information about presence, position, vehicle type and other basic information. Vehicles that receive the CAMs are aware of their neighbors and can use the information to evaluate their own situation. Therefore, it is fair to assume that they will constitute the major load on V2X systems in non-emergency situations, i.e. in normal use.

The investigated work splits are:

1) PPD as HMI and V2X Message Processing Unit
2) OBU as Facility Services Provider and PPD as HMI Provider
3) Hybrid Approach for Handling Facility Services

The three work split configurations are depicted in Fig. 2. Since mobile devices are not equipped with the V2X access technology IEEE 802.11p, this functionality is for all cases provided by the OBU. In order to maximize the overall system stability and the interoperability with other V2X communication devices, we have chosen to implement the essential networking and transport functionalities in the OBU's V2X stack. Both, the Basic Transport Protocol (BTP) (9) and GeoNetworking (GNW) (10) are currently going through the final revision process at the European Telecommunications Standards Institute (ETSI). As soon as the revisions are completed, it can be expected that the protocols will not be changed for years, as it is common for automotive systems. Keeping these functions in the OBU means that we do not have to test the interoperability and the long-term stability for every new functionality we add to the PPD. In addition, this design decision ensures that the basic V2X functionalities, such as message forwarding, can also be provided by the OBU when no PPD is coupled to it.

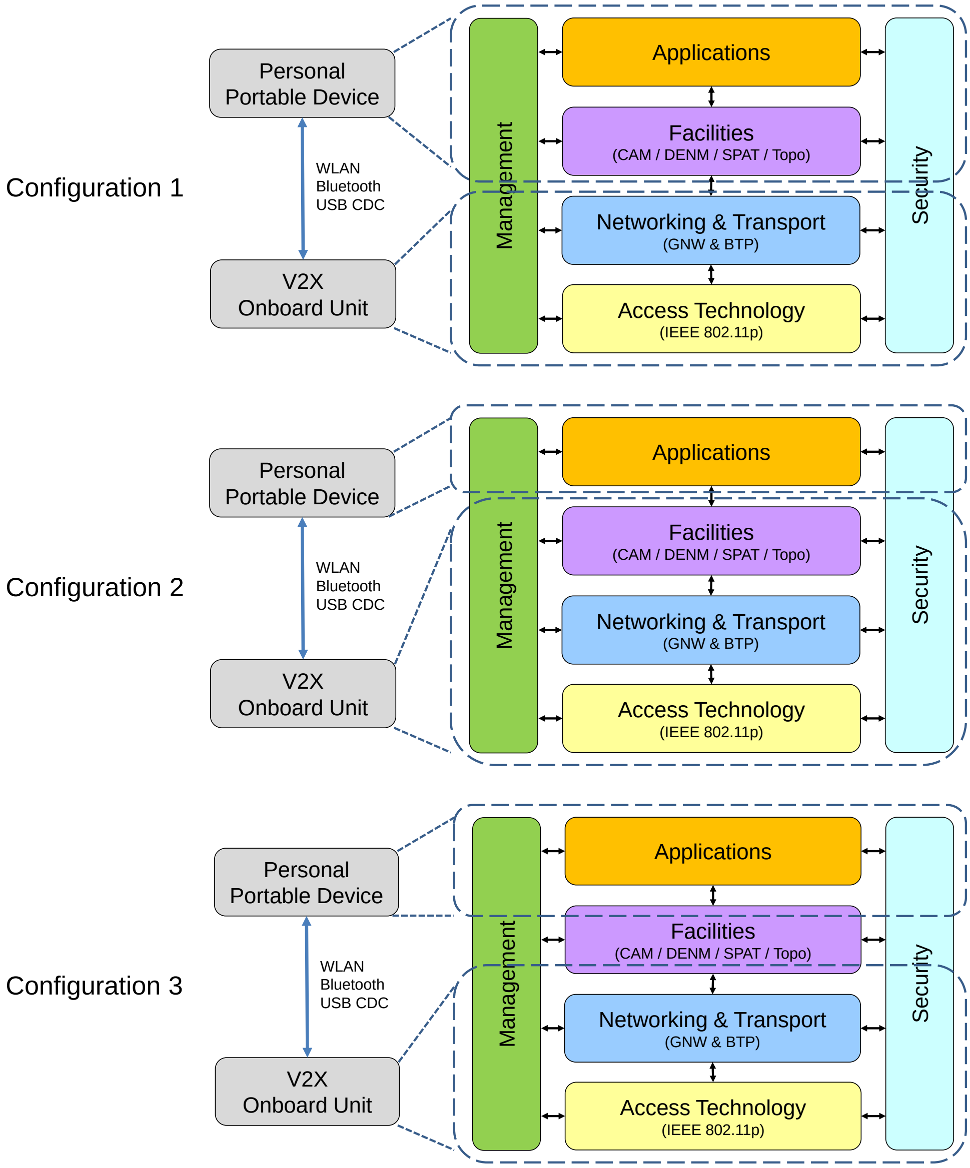

Fig. 2: Overview of the three investigated work splits.

Configuration 1: PPD as HMI and V2X Message Processing Unit

The first investigated work split is depicted in Fig. 2, Configuration 1. The OBU acts mainly as V2X message gateway. All incoming data that is intended for the ITS station is directly passed to the PPD. Forwarding the raw message directly to the PPD is the most flexible and future-proof solution. Instead of upgrading the OBU's software at a garage when new message types are introduced, it is sufficient to supply a software update for the PPD which can be installed by the end-user.

For the first configuration, we have used the open-source ASN.1 Java framework *BinaryNotes*[6] for decoding incoming messages directly on the PPD. The first benchmark has been conducted under low CPU load. 1,000 decodes last on average about 9500 ms which leads to an average decode rate of 105 Hz. Measurements during active map rendering with *mapsforge*[7] and distance calculations between the position received in the CAMs and a given point, lead to an average achievable CAM decode rate of 41 Hz. Considering a CAM transmission rate of 10 Hz per vehicle, only four stations could be supported by this solution. Thinking of dense highway traffic, this number is much to low for a viable traffic awareness system.

[6]http://bnotes.sourceforge.net/, last accessed August 31, 2012.
[7]http://code.google.com/p/mapsforge/, last accessed August 12, 2012.

For measuring the device's CPU load, CAMs arriving with a rate of 20 Hz have been used. The average CPU load for the decoding process has been between 32% and 41% on the Samsung Galaxy Tab P1000. Considering a short trip with the mobile device not connected to a power source, this would heavily drain the battery.

We expect that using a commercial ASN.1 decoding library, such as MARBEN's TCE-Java[8] or OSS Nokalva's ASN.1/Java[9], would lead to better results in terms of performance and battery usage. However, maximum flexibility by means of handling facilities and applications on the PPD would always lead to a heavy battery drain. In addition, novel driver assistance systems or safety features integrated in the vehicle's system could also make use of the V2X data. For this setup, the PPD would either have to share the encoded data again with these system, or the new components would have to handle the messages themselves.

Configuration 2: OBU as Facility Services Provider and PPD as HMI Provider

In order to maximize the overall performance, we decided to evaluate a second work split which is depicted in Fig. 2, Configuration 2. In contrast to Config. 1, the PPD is only responsible for running the applications and for providing the HMI. The facilities, such as CAM and DENM services, have been completely shifted to the OBU. For decoding CAMs, the C/C++ ASN.1 library from Objective Systems[10] has been used. The CAM decoding benchmark on the OBU lead to an average CAM decoding rate of $41,500$ Hz with a maximum CPU load of about 78%. This is about 400 times faster than with *BinaryNotes* on the PPD.

For forwarding the decoded V2X messages (together with management and statistical data) to the PPD, an efficient message format has to be chosen. Since lengthening the data many times over would cause a communication bottleneck between the PPD and the OBU, human readable formats, such as Extensible Markup Language (XML) or JavaScript Object Notation (JSON), are ineligible. However, several performance and space optimized binary serialization libraries are available that offer libraries for C/C++ and Java. Based on previous performance examinations[11] and availability of documentation, we have chosen *Protocol Buffers (ProtoBuf)*[12] for exchanging the data between the OBU and the PPD. ProtoBuf allows for a fast and flexible message creation that supports different scalar value types as well as lists and optional fields. The test CAM encoded with ASN.1 packed encoding rules (PER) has a size of 24 B. A *ProtoBuf* message holding the same values plus message type identifier and some statistical data is 46 B which is still less than double the length of the ASN.1 PER encoded version.

For evaluating the overall system performance, we have measured the average maximum combined ASN.1 decoding and ProtBuf encoding on the OBU. The average result was about $15,900$ Hz including all integer to float conversions so that the PPD does not have to perform any preprocessing on the messages. On the PPD, the ProtoBuf decoding of the generated message has been benchmarked, again while rendering a map on the PPD. We have measured an average maximum achievable CAM processing rate of 3500 Hz on the Samsung Galaxy Tab, which is more than 80 times faster as Config. 1. In another benchmark with additional position processing and data reasoning, we could handle data from more than 75 vehicles at the same time without influencing any other process on the OBU and the PPD.

The CPU load caused by the message handling on the PPD was again evaluated with CAMs at 20 Hz. On average, the ProtoBuf message decoding caused a CPU load of 2.3% (averaged over an interval of 10 about minutes). In contrast to the first configuration, the effect of message decoding on battery drain is almost negligible.

[8] http://www.marben-products.com/asn.1/tce_java.html, last accessed August 28, 2012.
[9] http://www.oss.com/asn1/products/asn1-java/asn1-java.html, last accessed August 28, 2012.
[10] http://www.obj-sys.com/asn1-compiler.shtml, last accessed September 6, 2012.
[11] http://code.google.com/p/thrift-protobuf-compare/wiki/BenchmarkingV2, last accessed September 05, 2012.
[12] http://code.google.com/p/protobuf/, last accessed August 14, 2012.

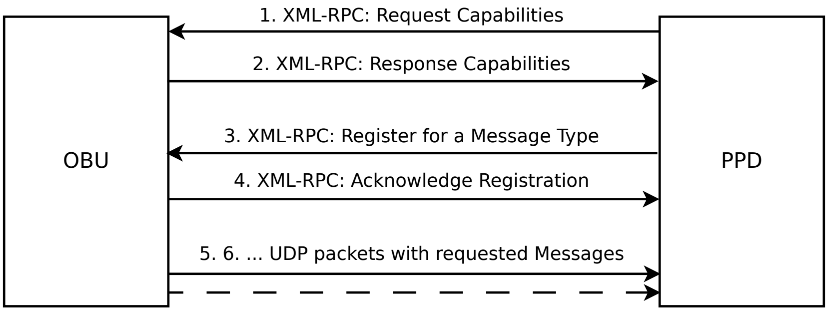

Fig. 3: A XML-RPC server on the OBU allows to register for different message types. Besides already decoded messages, the PPD can also request raw messages on any BTP port.

However, the flexibility of this solution is very low. The introduction of new message types requires updating the OBU. The need to update both, the OBU as well as the PPD, can easily lead to compatibility issues when, for example, only one device's software is updated.

Configuration 3: Hybrid Approach for Handling Facility Services

In order to gain flexibility, we have developed a third configuration (depicted in Fig. 2, Configuration 3) that combines the strengths of the previous configurations. This has been realized by enabling facility handling on the OBU and the PPD. Basically, the message decoding functionality remains on the OBU, with the difference that the PPD can also request messages on any BTP port in raw format. Additionally, the compatibility problem when having updated only the software of the PPD, or even just of the OBU, is addressed by this approach.

The elementary part of the configuration is a XML remote procedure call (RPC) server running on the OBU. When the PPD is coupled to the OBU, it can send a XML-RPC request to this server, asking for the version and capabilities of the OBU's software. The OBU's response contains the software version of the OBU and a list of message types and versions that can be decoded on the OBU. Based on this information, the PPD can either request decoded messages or raw messages that are received by the OBU on the desired BTP port.

A sample communication scenario between the OBU and the PPD is depicted in Fig. 3. The PPD first requests the capabilities and then registers for a message type (for example, *raw*, *CAM*, or *DENM*) on a BTP port. When the request can be fulfilled by the OBU, it acknowledges the PPD's request and starts sending either already decoded or raw messages to the PPD. In that way, multiple devices can register for multiple message types. In case of raw message forwarding, performance and battery usage results from Configuration 1 are valid. When the OBU decodes the messages, performance and battery usage statistics from Configuration 2 apply.

Conclusion

In this paper, we have described a two-component setup for integrating mobile devices with V2X communication. Based on this setup, we have investigated three possible work split configurations. Starting from a very flexible and intuitive approach, we have optimized our solution in two steps. Our final setup keeps the maximum flexibility but can handle a more than 60 times higher message rate while conserving battery of the user's PPD.

In conclusion, combining the strengths of both worlds seems to provide a viable solution for market introduction of V2X systems. The modules implemented in the OBU ensure system stability and reliability for safety critical applications. The flexibility of modules implemented in the PPD allows for rapid development of new applications and continuous improvement of user perception.

With DriveAssist (11), we have created a first research prototype of an Android-based driver assistance and awareness system that can be used for further investigations on this area. In future work, we plan, among other things, analyzing security risks and solutions of such a two-component setup similar to previous work from Leinmüller et al. (12, 13).

References

1. ETSI TS 102 637-2, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service," Technical Specification V1.2.1, March 2011.
2. ETSI TS 102 637-3, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service," Technical Specification V1.1.1, September 2010.
3. M. Kranz, A. Schmidt, and P. Holleis, "Embedded Interaction: Interacting with the Internet of Things," *IEEE Internet Computing*, vol. 14, no. 2, pp. 46–53, Mar.–Apr. 2010.
4. S. Diewald, A. Möller, L. Roalter, and M. Kranz, "Mobile Device Integration and Interaction in the Automotive Domain," in *AutoNUI: Automotive Natural User Interfaces Workshop at the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2011)*, Nov.–Dec. 2011.
5. D. K. Grimm, "Smartphone-Integrated Connectivity Applications for Vehicular Ad-hoc Networks," in *Proceedings of 18th ITS World Congress*, 2011.
6. M. Kranz, A. Franz, M. Röckl, L. Andreas, and T. Strang, "CODAR Viewer - A Situation-Aware Driver Assistance System," in *Pervasive 2008 Demonstration*, ser. Advances in Pervasive Computing, vol. 236. Austrian Computer Society, May 2008, pp. 126–129.
7. C. Löper, A. Catala-Prat, J. Gacnik, J. Schomerus, and F. Koster, "Vehicle Automation in Cooperation with V2I and Nomadic Devices Communication," in *Proceedings of the 14th International Conference on Intelligent Transportation Systems (ITSC)*, Oct. 2011, pp. 650–655.
8. IEEE, "802.11p-2010 - IEEE Standard for Information technology – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments," IEEE, Tech. Rep., 2010.
9. ETSI TS102 636-5-1, "Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 5: Transport Protocols; Sub-part 1: Basic Transport Protocol," Technical Specification V1.1.1, February 2011.
10. ——, "Intelligent Transport Systems (ITS); Vehicular communications; GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 1: Media-Independent Functionality," Technical Specification V1.1.1, June 2011.
11. S. Diewald, A. Möller, L. Roalter, and M. Kranz, "DriveAssist - A V2X-Based Driver Assistance System for Android," in *Automotive HMI Workshop at Mensch und Computer 2012*, Sep. 2012.
12. T. Leinmüller, E. Schoch, and C. Maihöfer, "Security Issues and Solution Concepts in Vehicular Ad Hoc Networks," in *Proceedings of the Fourth Annual Conference on Wireless On demand Network Systems and Services (WONS 2007)*, Jan. 2007.
13. T. Leinmüller, R. K. Schmidt, E. Schoch, A. Held, and G. Schäfer, "Modeling Roadside Attacker Behavior in VANETs," in *Proceedings of 3rd IEEE Workshop on Automotive Networking and Applications (AutoNet 2008)*, 2008.