The ActMAP to FeedMAP Framework
Automatic Detection and incremental updating for Advanced In-Vehicle Applications

Jan Löwenau[1], Sinisa Durekovic[2], Bernd Thomas[3], Hans-Ulrich Otto[4]

ABSTRACT. Up to date digital maps are a demanding requirement especially in the context of ADAS applications. This paper presents first results and applications from the FeedMAP project and how they can be used for increasing driving safety by integrating map deviation detection and incremental update technology into ADAS frameworks.

INTRODUCTION

The ActMAP framework provides concepts and methods for wireless distribution of incremental map updates for in-vehicle navigation and Advanced Driver Assistance System (ADAS) applications with the general goal to achieve highest up-to-dateness of an in-vehicle map database.

Although the ActMAP framework helps to shorten the time span between updates significantly, there is still space for improvement in terms of detecting map errors, changes in the real world, or giving attention to highly dynamic events like local warnings automatically. One basic assumption of the ActMAP framework is that such deviations, which lead to map updates, are detected and provided by the update suppliers. Since constantly checking wide areas of a road network is a time consuming and cost intensive process for update supplies, which obviously can only be done periodically and not in a permanently manner, the basic idea is to use the end customer's vehicle equipped with either a navigation system or ADAS application for the automatic detection of map deviations. The subject of the FeedMAP project is to provide a framework for implementing this solution, to achieve a even higher degree of map up-to-dateness for in-vehicle map databases.

THE ACTMAP – FEEDMAP LOOP

Up-to-date map data is a must for current and future navigation and Advanced Driver Assistance System (ADAS) applications. Today, digital maps are normally stored on DVDs, hard disks, or memory storage cards, with periodic updates only available with complete map version replacements. However, new mechanisms for updating maps have been investigated and some of them already reached the market. As the real world is changing every day, detecting changes to the road network quickly and at a low cost is a challenge. Although

[1] BMW Group Forschung und Technik, Hanauer Str. 46, 80992 Munich, Germany, e-mail: jan.loewenau@bmw.de
[2] NAVTEQ, Otto-Volger-Str 1, 65843 Sulzbach, Germany, e-mail: sinisa.durekovic@navteq.com
[3] NAVIGON AG, Berliner Platz 11, 97080 Würzburg, Germany, e-mail:bernd.thomas@navigon.com
[4] Tele Atlas Deutschland GmbH, Am Neuen Horizont 1, 31177 Harsum, Germany, e-mail: hans.ulrich.otto@teleatlas.com

mapmakers continuously survey the European road network for changes, map information is not always up-to-date or accurate. Major map suppliers estimate that road networks change by approximately 10-15% every year and by as much as 40 % in high-growth areas.

In addition to the surveys, mapmakers rely on road authorities and user feedback to detect the latest changes in the road network. The FeedMAP project studies the commercial and technical feasibility of map deviation feedback. In order to improve the flow of user map feedback, the FeedMAP concept uses all vehicles equipped with a digital map for deviation detection, i.e. to check if the maps are indeed accurate and complete.

The FeedMAP project is developing a framework that detects map anomalies when cars happen to come across a faulty map attribute and that will quickly update the maps of all other vehicles using a standardised mechanism for delivering incremental map updates based on the concepts developed in the ActMAP project.

FeedMAP's main focus is on static and semi-static changes to the road network. In addition, real-time status of reports on events such as traffic jams or accidents can be verified and returned to the information source. Figure 1 depicts the overview of the architecture of the ActMAP – FeedMAP framework.

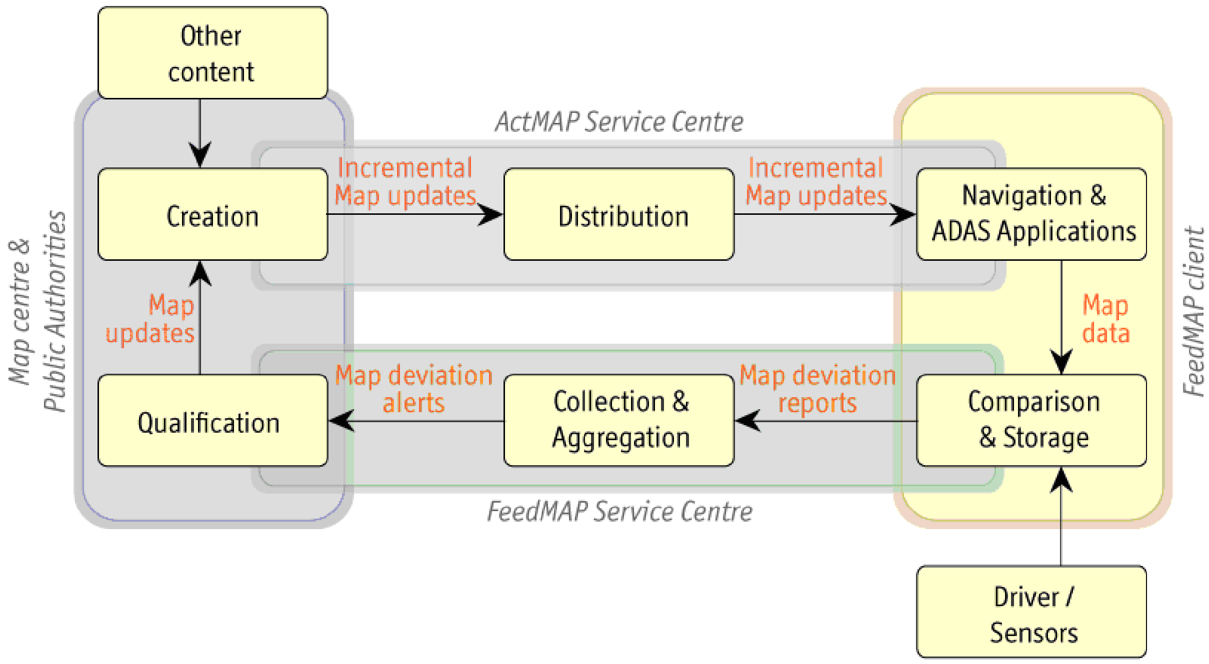

Figure 1. ActMAP-FeedMAP framework

Coupling the ActMAP and FeedMAP framework is conceptually a reasonable step for the following reasons:

- Faster availability of map updates due to automatic, permanent, and global area-wide monitoring of map errors and real world road changes.
- Minimizing maintenance (data acquisition) costs for update suppliers.
- Increasing the quality of maps in general and specifically by reliable update information given from public authorities and other trustworthy Location Based Content (LBC) providers.

- Better quality of service for the end customer due to increased up-to-dateness of maps and additional services like dynamic content updates.

Considering the core functionalities of the ActMAP and the FeedMAP system the main roles of each system is clearly characterized as Update Distributor and Update Detector (Figure 1). Whereas update suppliers like map centres or content providers in general take over the role of an Update Evaluator/Provider. Hence they have to decide if FeedMAP updates derivable from FeedMAP deviation alerts meet a special quality criteria, such that they can be propagated via the ActMAP services to in-vehicle applications.

MAP DEVIATION DETECTION

FeedMAP/ACTMAP loop starts when the difference between ground truth and content of the digital map is detected. Difference may be in absence of real-world entity in the digital map, presence of digital map entity that does not exists in reality or in difference between value of entity attribute stored in the digital map and actual real-world value of the attribute.

In the FeedMAP framework, those differences are called Map Deviations and they are described in XML-formatted data structures referred to as a Map Deviation Report (MDR). These Map Deviations are detected by 'FeedMAP Clients' (FMC).

A FeedMAP Client generally fits into two categories:

1. Car Probes are FeedMAP clients equipped with sensors and algorithms that are used in Deviation Detection.
2. Public Authorities (PA) are FeedMAP clients that generate 'Map Deviation Reports' as well. However, since PA initiates or at least keeps official records about many attributes contained in the Digital Map, they are the reference source for that information.

Map Deviation Detection algorithms implemented in car probes can be generally grouped into three categories: autonomous, manual, and joint detection.

Autonomous detection does not involve any conscious driver action; source of the data that indicates the deviation is only provided by different car sensors. An example of the Autonomous detection is detection of link 'travel time' errors. This attribute tells to the navigation system the average time necessary to transverse one map link and it is one of most important attributes used in fastest route calculation. Car probe can simply measure time necessary for car to drive along one link and if this value significantly differs from the value stored in the digital map, Map Deviation Report can be generated.

Manual deviation detection algorithms rely only on driver interventions. Detection of Scenic Routes or changes in Point-of-Interest attributes (telephone number, opening hours …) are typical examples. In general, all manual detection algorithms can be automated by use of hardware sensors and by applying complex software algorithms, but in most cases such approach is not feasible.

Joint detection algorithms are those algorithms where the system detects the deviation, but some driver confirmation is necessary. This confirmation can be explicit (when the system asks the driver for confirmation), or implicit (when driver action confirms the assumption of system).

For some deviation types, different deviation detection methods can be developed. Systems can rely on more or less sophisticated hardware sensors and or more or less complex software algorithms in deviation detection. Of course, more advanced sensors and more advanced algorithms will reduce the need for drivers' involvement in Deviation Detection. Since today's 'Driver Workplace' is already fairly complex, one should avoid manual detections and joint detections with explicit confirmations because they increase the drivers' workload.

DEVIATION REPORT ANALYSIS

It is the task of the FeedMAP Service Centre (FMSC) to receive *Map Deviation Reports* (MDRs), to analyse them and to decide if and when a *Map Deviation Alert* (MDA) is to be generated and sent to the Map Centre.

Within The FeedMAP project, Tele Atlas is implementing a FMSC, that will receive and analyse MDRs from NAVIGON's FeedMAPSensor, as described later in this document.

The workflow in the FMSC looks as follows.

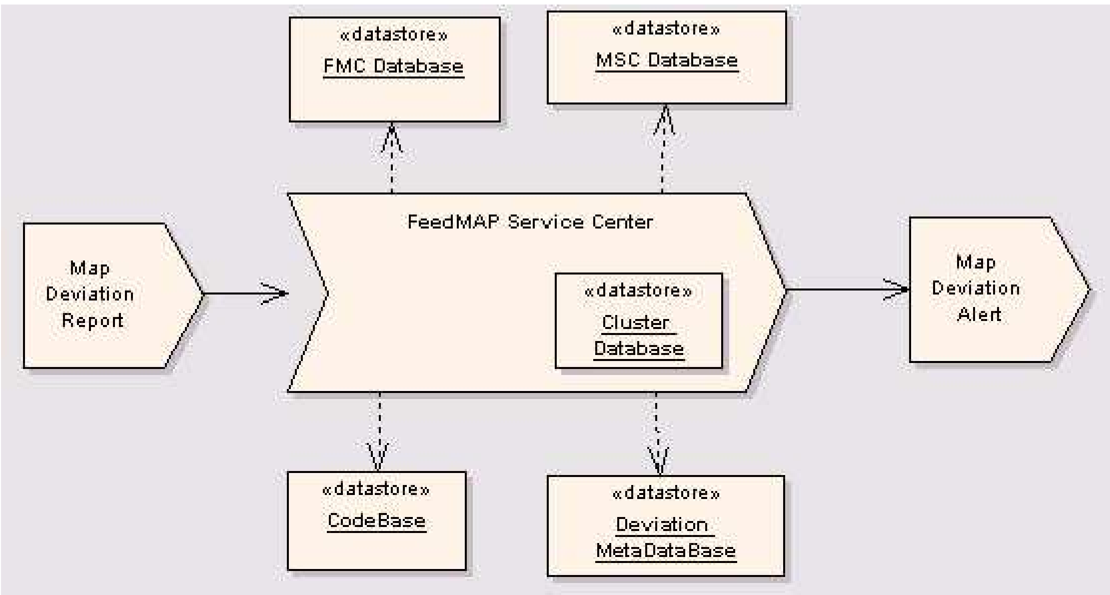

Figure 2. FMSC data flow

The Deviation MetaDataBase is the place where configuration data for verification and processing for each Deviation Type is stored.

The CodeBase is a repository of algorithms. Here it is specified what algorithm is to be used to check the quality of the MDR and to analyze clusters of particular deviation type.

The Cluster Database is used as a temporary store for the MDRs. In that database, MDRs will be grouped into Clusters. Each Cluster is a collection of MDRs that describes same real-world deviation types. Each MDR belongs to exactly one cluster.

The MSC Database (see Figure 2) stores information about different map centers where to send the final MDAs to. The FMC Database is the place where relevant data about FeedMAP Clients is stored.

The following example in Figure 3 illustrates how a MDR looks like. The client with abstract ID = 42 delivers a repost about a deviation found. It relates to a database from TELE ATLAS, with the release GERMANY-2007Q1, updated the last time on 2007-07-20 at 00:42:42am. The client reports a likely wrong oneway, observed on 2007-08-02 at 00:57am. For the detection, the system used a GPS, odometer and gyro sensors. The GPS quality was quite low, HDOP=12.5. The client also delivers a geographic location as a sequence of latitude and longitude pairs.


```
<?xml version="1.0" encoding="UTF-8"?>
<MapDeviationTransferRequest
        xmlns:adtlt="http://www.location.com/adtlt"
        xmlns:gml="http://www.opengis.net/gml"
        xmlns="http://www.feedmap.com"
        version="1.2.1">
    <MapDeviationReport
            vendor="TELE ATLAS"
            release="GERMANY-2007Q1"
            lastUpdate="2007-07-20T00:42:42.000+00:00">
        <DeviationType>WRONG_ONEWAY</DeviationType>
        <TimeStamp>2007-08-02T00:57:00.000+00:00</TimeStamp>
        <SensorList>
            <GPS>
                <HDOP>12.5</HDOP>
            </GPS>
            <Odometer>true</Odometer>
            <Gyro>true</Gyro>
        </SensorList>
        <ClientID>42</ClientID>
            <DeviationLocation locationQuality="0.1">
                  <gml:LineString
                    srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
                        <gml:coordinates decimal="." cs="," ts=" ">
                              9.1566292,45.4839422 9.1567121,45.4840502
                              9.1567666,45.4841599 9.1567997,45.4842592
                              9.1567941,45.484399 9.1567668,45.4844842
                              9.156712,45.4845783 9.1566383,45.484663
                        </gml:coordinates>
                  </gml:LineString>
            </DeviationLocation>
          </MapDeviationReport>
</MapDeviationTransferRequest>
```


Figure 3. MDR Example

How a FMSC operates to deal with MDRs is illustrated by Figure 4. During the FMSC Start-up, all necessary initializations will be preformed. In the main FMSC loop, the system will wait for one of three events:

- Shutdown request
- Timer expiration
- Arrival of the new MDR

If a shutdown is requested, the shutdown procedure will clean used resources. If the timer has expired, the system will check all the clusters in Cluster Expiration Check procedure. Old clusters that did not produce MDAs in predefined time, as well as old obsolete clusters will be deleted. When a new MDR has arrived, it will be processed in up to four steps:

1. Validation will check if MDR is “good enough” to be processed.
2. Clustering will assign MDR to the existing or new Cluster.
3. Cluster Processing will analyze updated Cluster(s).

4. If it happens during the Cluster Processing that the Cluster contains enough quality data, MDA will be issued by Map Deviation Alert Factory.

After Cluster Expiration Check or MDR Processing, the system will go back into Main FMSC Loop.

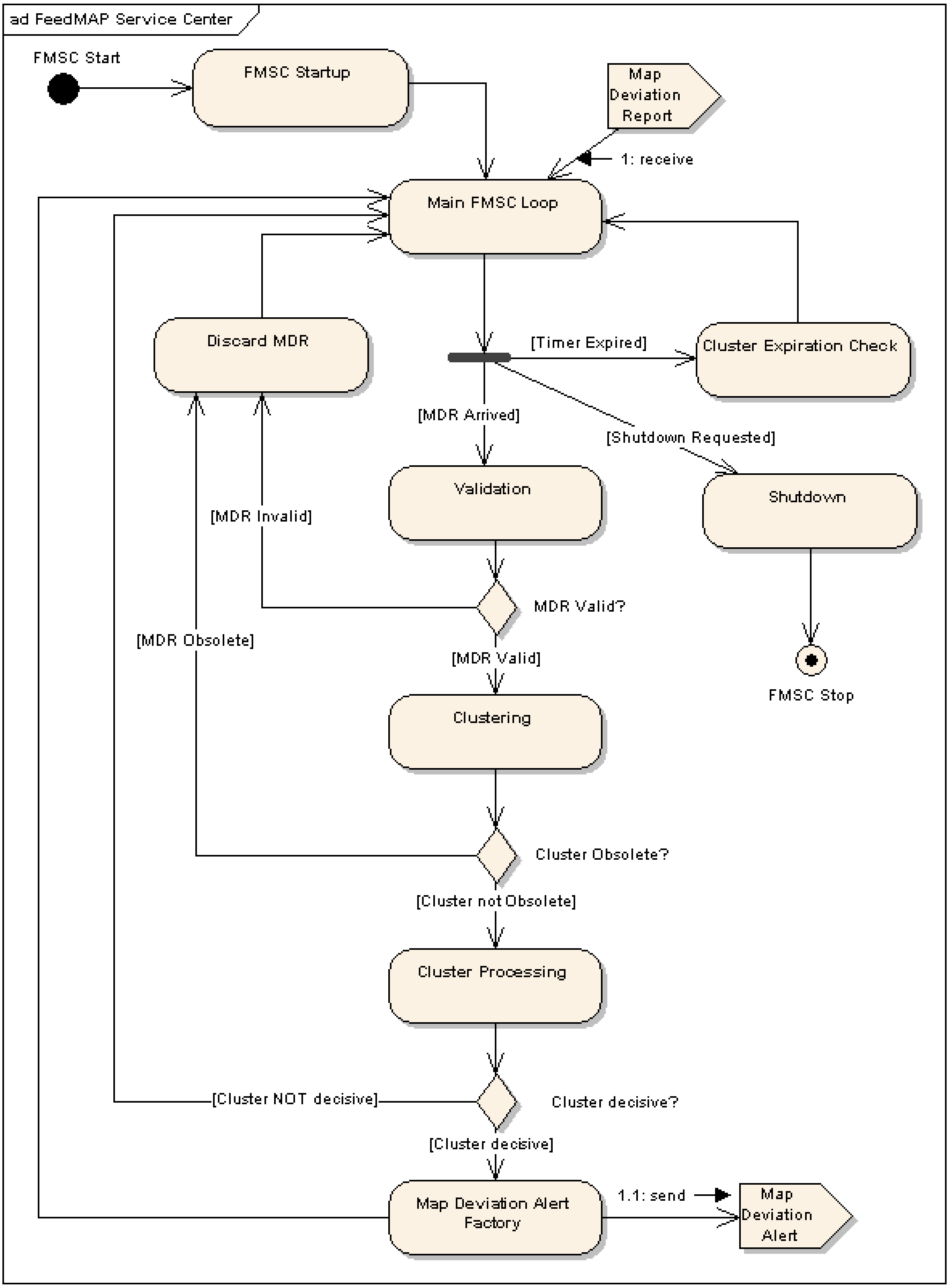

Figure 4. FMSC generalised algorithm

APPLICATIONS AND FEEDMAP-ADAS TEST SCENARIOS

Within the FeedMAP project different implementations of the FeedMAP Clients, Service Centres and applications are developed. In this paper we focus on ADAS related applications and ADAS development platforms based on the concept of the ADAS Horizon. The ADAS Horizon comprises a solution for providing digital map information about the most probable path the vehicle will take to the vehicles CAN-bus (Controller Area Network bus). This concept was developed in the EU research project MAPS & ADAS.

Extended Adaptive Speed Recommendation

The BMW application "Adaptive Speed Recommendation" is a typical example of ADAS applications whose usability strongly depends on the correctness of the map data. The Adaptive Speed Recommendation Info function provides additional, detailed information on the stretch of road the driver is currently covering. To provide this helpful support, a traffic sign graphics in the instrument cluster – head up display and navigation display - in the cockpit informs the driver of the speed limit at his current location (see Figure 2).

Figure 5. BMW Vehicle used in the FeedMAP framework

Adaptive Speed Recommendation is extended with map deviation detection, reporting and dynamic map update capabilities and it is now active part of the ActMAP-FeedMAP chain. Figure 6 shows a block diagram of the developed ADASRP[5] platform connected to the BMW sensor CAN box wrt. the vehicle data.

[5] NAVTEQ Advanced Driver Assistance System Research Platform 2006 is Windows-based framework application for hosting various ADAS solutions.

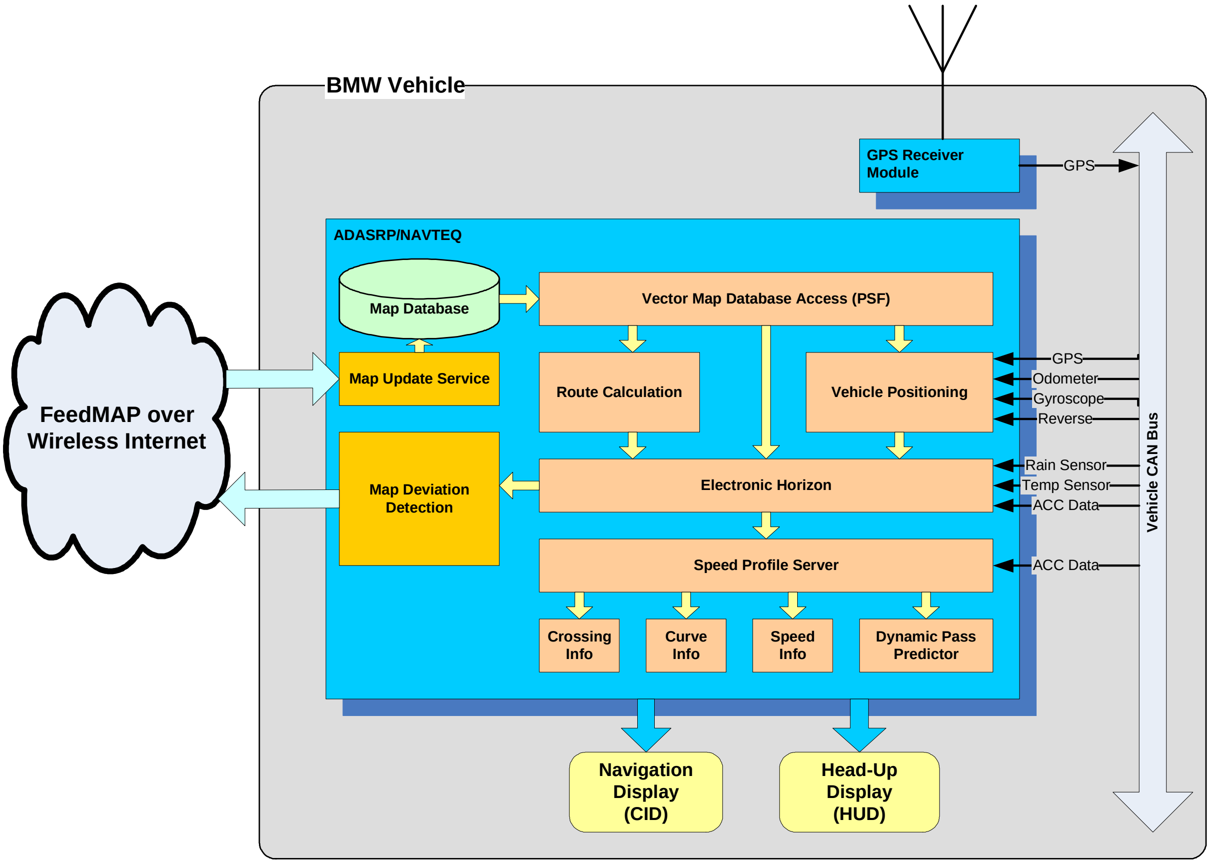

Figure 6. Hardware and software modules used in the BMW prototype test vehicle

Legal Speed Limit Deviation Detection is a typical example of complex joint detection system with implicit driver confirmation. The system developed by BMW and NAVTEQ is based on BMW/NAVTEQ Speed Profile Server/Adaptive Speed Recommendation (ASR) system that informs the driver about maximum recommended speed. This recommendation is calculated by taking in to account the current speed, vehicle dynamic characteristics (braking, acceleration), drivers' preference (relaxed, normal, sport driving) as well as the number of information about road ahead: curves, crossings, slope, and, of course, Legal Speed Limits.

In addition, the system monitors the driver's behaviour – namely speed, and its response to the speed warnings given by the adaptive speed recommendation.

For instance, a digital map may contain legal speed limit of 60 km/h for some road segment but the driver consistently keeps the speed of 100 km/h despite speed warnings given by the ASR. In this case, a legal speed limit deviation detection will assume that the data about 60 km/h speed limit is wrong and it generates an appropriate map deviation report.

On the other side, let's assume that the driver keeps the speed of around 60 km/h along one particular road segment where the legal speed limit is 100 km/h. Does the map legal speed limit has a wrong attribute value or is there another reason for the lower speed?

The BMW/NAVTEQ Legal Speed Limit Deviation Detection system uses different sensors and algorithms to answer this question: Adaptive Cruise Control (ACC) radar to indicate presence of slow vehicle in front; rain and temperature sensor to detect heavy weather conditions that may force the driver to drive slowly, road geometry in front that may reveal sharp curves etc.

Updating the ADAS Horizon

The NAVIGON AG extended within the FeedMAP project their ADAS application (MapSensor), which provides digital map information about the most probable path the vehicle will take to the vehicles CAN-bus, with FeedMAP client functionality. The resulting application called FeedMAPSensor (Figure 7), is evaluated in a joint test-site with Volvo Technology (VTEC), Tele Atlas, and Swedish Road Administration (SRA).

The FeedMAPSensor application is installed on a VTEC truck, which is equipped with various sensors that send information to the truck's CAN bus. Such sensors, apart from standard velocity sensors and GPS, comprise vision lane detection modules, driver workload sensors, and radar. The sensors' information are used to assist the deviation detection algorithms in autonomous, joint and manual mode.

Besides reporting detected deviations to the FeedMAP Service Centre the detected deviations are maintained by a special update store concept, such that deviations can be sent to the vehicles CAN-bus to include warnings (as for instance geometry changes) for ADAS end-applications. Hence the FeedMAPSensor application extends the ADASA horizon by map updates and detected deviation information. Consequently ADAS applications based on the horizon information benefit from improved map data, since the system memorizes the changes in the road network and this information can be used to alert the driver about map changes and/or directly can include the update information into their ADAS algorithms. Finally the overall driving security and safety can be increased.

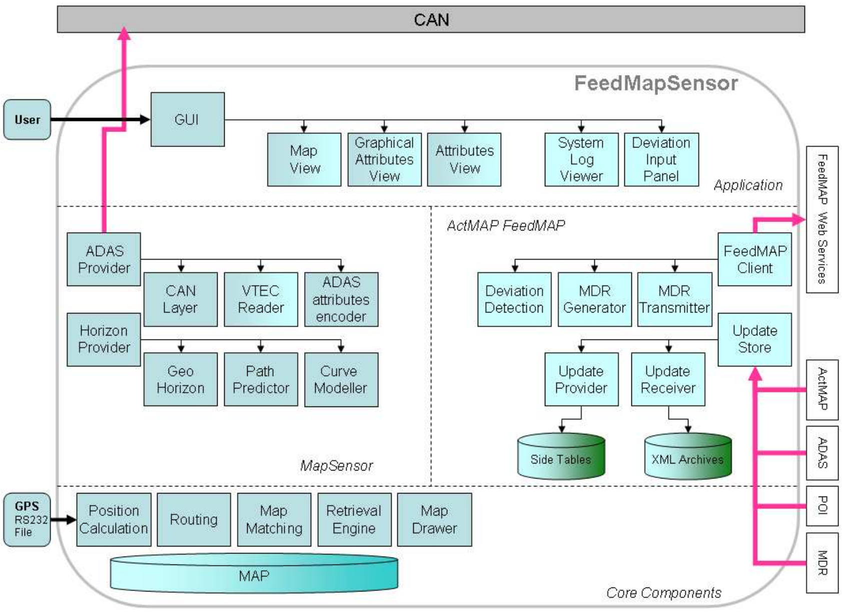

Figure 7. NAVIGON FeedMAP Extended MapSensor Provider

The FeedMAPSensor comprises the manual detection of 4 deviation types and automatic detection of 5 deviation types. The deviation types and detection methods are briefly described in the following:

A GUI deviation detection panel is used for the *Road Works Deviation Detection*. By simple means a manual detection to record start- and end location of road construction sites as well as the type of the construction is provided. With the combined use of the deviation input panel and interactive “point, click, and select” operations on the map, additional three deviation types are manually recorded by the user. These three deviation types comprise reporting status modifications like *new, changed, moved, and deleted for Point of Interests (POI), Traffic Signs, and Speed Limits*. Although manual detection is of important the main focus is clearly on the automatic detection of deviations, since this reduces workload and minimizes the risk of disturbing the driver. In the following the automatic detection of the five deviation types currently supported by the FeedMAPSensor are explained.

The FeedMAPSensor automatically detects *Wrong Road Geometry* and visualizes these detections in the digital map view (left window Figure 8). If the vehicle will approach a location where a deviation was previously detected, the deviation information will also be send to the ADAS horizon and displayed as additional horizon attributes (middle and right window Figure 8) .

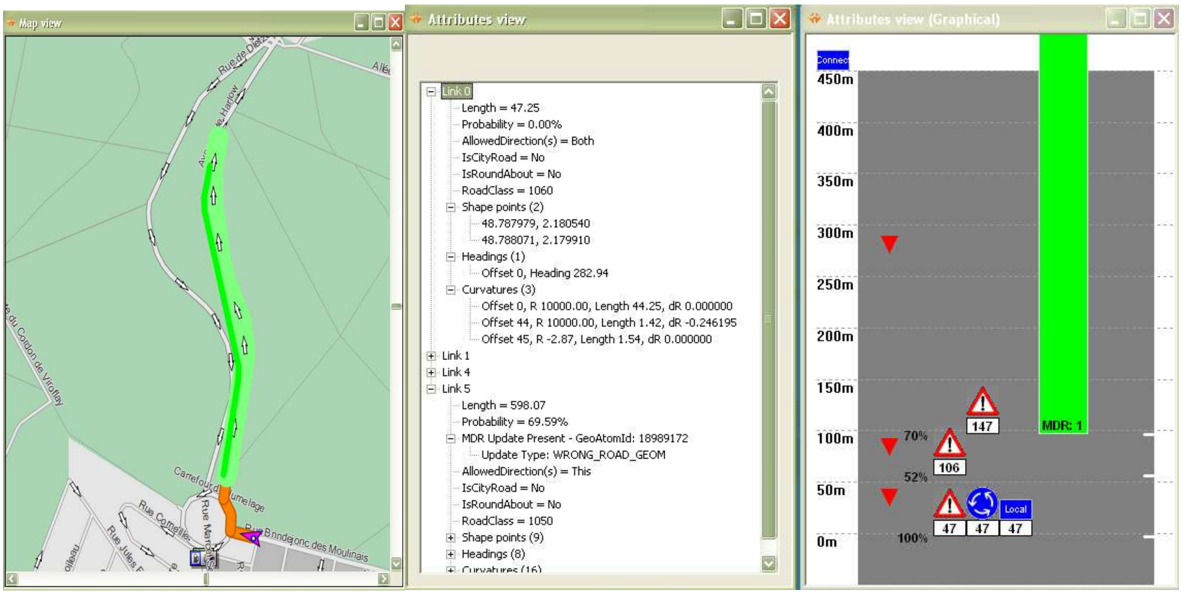

Figure 8. Detected Deviation WRONG_GEOMETRY and ADAS Horizon

Speed Limit Deviation Detection is performed in two different modes, automatic and manual. Automatic detection is based on speed information given by the GPS receiver and additional (optional) radar information about the speed of vehicles in front of the truck.

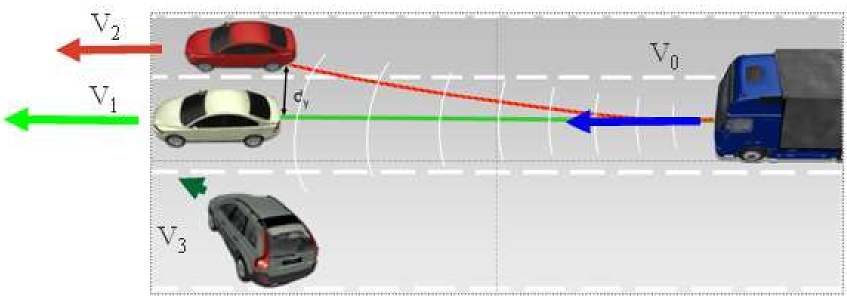

Figure 9. Radar Usage for WRONG SPEED LIMIT detection

Since the measured speed information solely based on the GPS information might lead to false assumptions on possible speed limit changes, because their might be congestions, illegal speeding by the driver, or the driver is simply driving very slow, the use of radar improves the speed limit detection heuristics. Therefore radar information is read from the trucks CAN bus providing information about the speed of objects in the near vicinity of the truck (Figure 9). The detected speed information from the trucks speed is compared with speed limit attributes attached to the map. Radar and Truck speed information is used for different speed limit detection models to estimate changed speed limits.

The detection of *Missing Roads* is also supported and requires a close interaction with map matching components of the core system. The information about the new roads geometry is collected and reported to the FeedMAP Service Centre. As with all detected map changes this information is also available as update data to the ADAS horizon and thus can be used by ADAS applications. The left graphics of Figure 10 shows the map where the vehicle (triangle) is obviously driving on a missing road. The right window shows the map after driving the same road a second time with the new detected road drawn in green.

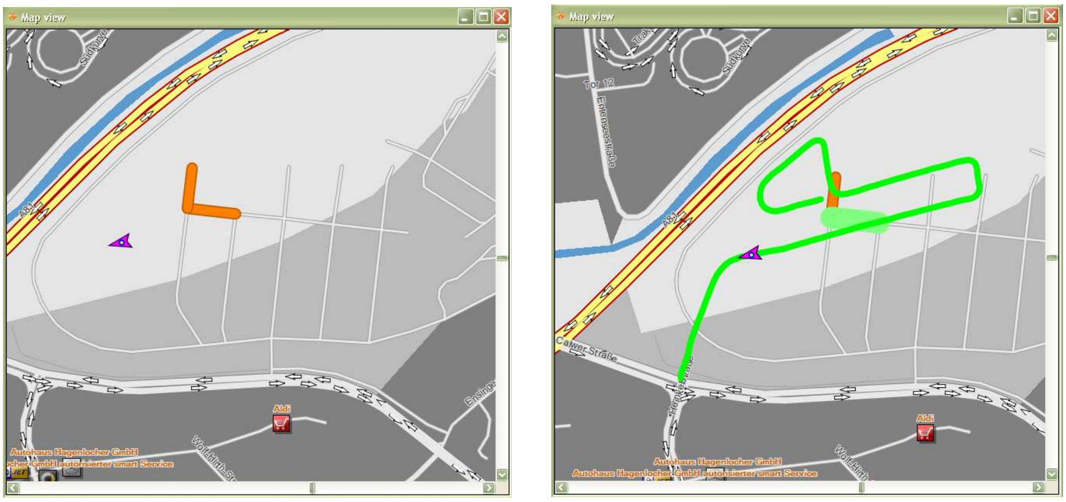

Figure 10. MISSING ROAD detection

Wrong or Missing Slope information scenarios are experienced as the slope assessment made by the truck's power train control units is compared to slope information in the map. The deviation detection algorithms under development will take in to account that slope estimations made from vehicle dynamic models alone or in combination with acceleration sensor(s) may vary in accuracy depending in the weight of the vehicle and might also suffer from braking of the vehicle since brake forces in some situations are hard to estimate correctly.

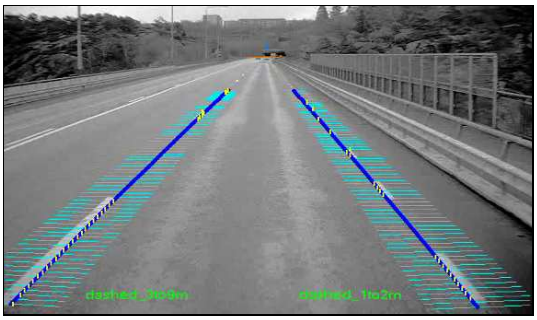

Figure 11. Image Lane Recognition

For *Lane Information Detection* the image processing unit from the EC research project SafeLane used by Volvo Technology is used. The vision system is mounted on a truck and provides real time information on the numbers of lanes and lane markers to the truck's CAN bus. The image processing unit is capable of detecting the width of lane, left and right lane marking type, left and right neighbour lane width, left and right neighbour lane marking (Figure 11). The data provided by this sensor system plus information given by the radar sensor is used to build a lane model. Then the detected lane information (markings, number of lanes) is compared to existing ADAS map attributes by the FeedMAPSensor's deviation detection algorithms. Independent of the case if some lane information is present in the digital map (ADAS attributes) the sensor data (vision system and radar) will be used to report an MDR containing lane information to the FeedMAP Service Centre.

Deviation detection is only one among many functionalities of the FeedMAPSensor. Beyond its functionality as FeedMAP detection probe it supports receiving and processing of various map related updates among which are ActMAP updates. Hence the FeedMAPSensor serves as ADAS development framework supporting a closed map updating loop.

CONCLUSIONS

At the conference the final version of BMW's ADAS solutions and NAVIGON's ADAS Advanced Programming Interface extended by the ActMAP-FeedMAP framework are presented.

Potential fields of deployment as, for example, precision navigation and ADAS applications (as shown in Figure 2) are demonstrated. Implementations of applications are showcased and the benefits of using the framework are explained.

Concluding it can be stated, that the deviation detection concept investigated and developed in the FeedMAP project and the updating concepts developed in the ActMAP project provide means to improve ADAS development frameworks and provide support for ADAS applications to improve quality in terms of up to date maps and increased driving safety and security.

REFERENCES

Loewenau, J. et al (2002), "Preliminary map actualisation requirements", *IST-2001-34141 ActMAP project*, Actual and dynamic MAP for transport telematics applications, (Nov. 30, 2002)

Otto H.U, (2005) editor, "Data Requirements". *Deliverable D1, MAPS & ADAS Consortium and ADASIS Forum*

Mezger K, (2005) editor, "ADASIS Functional Architecture". *Deliverable D2, MAPS & ADAS Consortium and ADASIS Forum*

Angenvoort J, (2005) editor, "Draft Specifications". *Deliverable D3, MAPS & ADAS Consortium and ADASIS Forum*.

Thomas B. et al (2007), "ActMAP White Paper and Interfaces to the FeedMAP framework", *Deliverable D3.2, IST-2005-027325 FeedMAP project*. Brussels, (Mar. 2007)

Visintainer F. et al (2006), "Preliminary requirements and strategies for map feedback", *Deliverable D2.1, IST-2005-027325 FeedMAP project*. Brussels, (Sep. 2006)

BOSCH (1997), "CAN Literature", http://www.semiconductors.bosch.de/en/20/can/3-literature.asp