SAFESPOT INTEGRATED PROJECT - IST - 4 - 026963 - IP

DELIVERABLE

SP3 - SINTECH - SAFESPOT Innovative Technologies

Local dynamic map specification

Deliverable No. (use the number indicated on technical annex)		D3.3.3	
SubProject No.	SP3	SubProject Title	SAFESPOT Innovative Technologies
Work Package No.	WP3	Work Package Title	Specification
Task No.	T3.3.3	Task Title	Local Dynamic Maps Specifications
Authors (per company, if more than one company provide it together)		Bosch - Alison Yuen, Christopher Brown, Christian Zott Daimler Chrysler - Andreas Hiller Ibeo - Florian Ahlers NAVTEQ - Kees Wevers (editor), Stéphane Dreher Technical University Munich - Tobias Schendzielorz Tele Atlas - Christine Bartels TNO - Zoltan Papp, Bart Netten	
Status (F: final; D: draft; RD: revised draft):		F	
Version No:		07	
File Name:		SF-D333-local-dynamic-map-spec-v06.doc	
Planned Date of submission according to TA:		01-07-2007	
Issue Date:		18-04-2008	
Project start date and duration		01 February 2006 - 48 Months	

Executive summary

Enabled by advances in communication technology, cooperative systems are seen as the next logical step beyond autonomous driving assistance systems, for safety and comfort applications for road traffic. In cooperative systems, autonomous information from stored digital maps and from vehicle sensors is supplemented with cooperative information received via radio links from other vehicles and the infrastructure. A new spatial database concept, named local dynamic map, reflecting all relevant static, temporary and dynamic information in the perception vicinity of a stationary object (road side unit) or moving object (vehicles and other road users), is considered as a core element of cooperative systems. The local dynamic map is a highly dynamic data store with a relation to the road network, which enables storage and updating of objects including type, position and other characteristics, and retrieval of selected information for further processing and situation analysis, like calculation of trajectories, and detection of hazardous obstacles and potential conflicts with other road users. If the object that maintains the local dynamic map is moving, the map window is moving as well, with the object as its centre point. The local dynamic map is constructed on top of a digital map database for ITS applications, and conceived as a four layer structure with increasing dynamics, and specified as a logical object model, which may serve as the basis for specifying the application programming interface, and for its actual implementation. The four layers represent respectively: (1) the static (semi-permanent) digital map database; (2) similar static information that is not (yet) incorporated in the digital map database; (3) temporary and dynamic information (like weather and traffic conditions); and (4) dynamic and highly dynamic information concerning moving objects (vehicles, vulnerable road users and animals). This document provides the specification of the local dynamic map.

The local dynamic map has a central position in the architecture of the SAFESPOT system. Only the sensor data processing and fusion module of the system will have write access to the local dynamic map for transactions. All modules and applications of the system, including the data processing and fusion module, will have access to the local dynamic map for queries. This implies that all temporary and dynamic information that is stored in the local dynamic map passes through data processing and fusion. For instance, sensor information that is used by an application, is not directly retrieved from the sensor, but as processed information from the local dynamic map.

The local dynamic map may be implemented as a relational or an object-oriented database. The static map data of layer 1 may be used by directly accessing the physical storage format of the digital map database or by transferring the static map data for the perception horizon also to the relational or an object-relational database. Positions of dynamic objects will be maintained both as absolute position and as map-matched position to the map database road centre, possibly with lateral offset.

The local dynamic map is described as a logical object-oriented model in a UML (Unified Modelling Language) static structure diagram. On top of the main class hierarchy is the WorldObject, with main subclasses Feature and DynamicObject. Feature and its subclasses cover all map related static objects (layers 1 and 2). The class DynamicObject has two main subclasses, ConceptualObject and MovingObject. ConceptualObject and its subclasses cover the temporary weather and traffic related in-

formation of layer 3. The class MovingObject covers the dynamic and highly dynamic information of layer 4. It has subclasses LivingObject (with subclasses Pedestrian, Bicyclist and Animal), Trailer and MotorVehicle (with subclasses TwoTrackVehicle and PoweredTwoWheeler). In addition to the main hierarchy various classes are defined that are associated with classes in the main hierarchy. Examples are the GeometryObject (associated with WorldObject) to express the geometry of an object, MotionState and Trajectory (associated with MovingObject) and Driver (associated with MotorVehicle). Another class associated with MotorVehicle is EgoMotorVehicle, covering information that is only present in the own (ego) local dynamic map. This document provides class diagrams of various parts of the complete class diagram. In addition the whole structure of the model is described in a text diagram, and the details of each of the classes (positing in the hierarchy, attributes and operations) in a set of tables.

The application program interface (API) will consist of two parts, a transaction part and a query part. The transaction part needs to be able to create (including attribute initialisation) and remove instances of all defined DynamicObject types (layers 3 and 4), and to set and change object attributes values (including uncertainty values) of dynamic objects that are stored in the local dynamic map, as well as of static objects for which dynamic attributes are defined. The query part of the API needs to be able to extract information from the local dynamic map concerning all object types (i.e. including the static map and map related information of layers 1 and 2). It needs to be able to query the database using filters for geographic location (geometry objects of various kinds as well as map database links), object types and object attributes. Database queries for moving objects in relation to map database links need to be able to include, in addition to the link to which the ego object is associated, relevant links ahead of the moving object (much like an ADAS Horizon). This enables the system to be aware of moving objects on those links that may be encountered by the ego object in view of their driving directions. The API may be described in terms of SQL statements, object-oriented query statements, or more generic function calls. Although this document provides a description of the API, precise definition of the API was still underway at the time of writing the final version of this document. In addition to queries on demand, a subscription mechanism will be defined for automatic notification, which will push information concerning objects to applications when certain predefined conditions are met. Although this mechanism is sometimes referred to as continuous querying, it will actually reduce the number of regular active queries that applications need to make to avoid the risk of using dated information, and thereby reduce computational load on the system.