INTERNATIONAL STANDARD

ISO 22837

First edition
2009-01-15

Vehicle probe data for wide area communications

Données de sonde du véhicule pour les communications de surfaces étendues

Reference number
ISO 22837:2009(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 22837 was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*.

Introduction

This International Standard specifies the core and initial sets of probe data elements and example probe messages, and a framework for defining and extending these probe data elements and probe messages. It facilitates the development and operation of probe vehicle systems by providing a standard set of probe data elements and probe messages along with the basic data framework to extend the set.

This International Standard provides a reference architecture for probe vehicle systems and for probe data, a basic data framework for probe data elements and probe messages, the definition of core data elements, the definition of an initial set of additional probe data elements, and the definition of example probe messages.

This International Standard facilitates the work of system developers and operators who need to specify probe data elements and probe messages:

There are many ways that probe data elements and probe messages could be defined. This International Standard provides a concrete and common way to define probe data elements and probe messages. The standard also facilitates communication and mutual understanding among the developers and the operators of probe vehicle systems.

The ability to develop probe vehicle systems in a consistent and uniform manner reduces development time and cost. If a particular probe vehicle system requires additional probe messages that are not yet part of the standard, the existence of a common framework for defining probe data elements/messages helps system developers to develop probe vehicle systems in a uniform way.

Probe data will be collected from many vehicle makes and models from many vehicle manufacturers. This standard provides a basic data framework for handling probe data elements/messages and the concrete definition of major probe data components that help collect and process probe data consistently.

It should be noted that this International Standard does not prescribe a physical communication medium for transmitting probe messages to or from vehicles. This International Standard is intended to be independent of any particular communication medium and to be compatible with any medium that is selected by system developers.

Core data elements and normative probe data elements are covered in Clauses 7 and 8 respectively. Subclause 5.3 and Annex A are of interest to users familiar with information modelling.

Vehicle probe data for wide area communications

1 Scope

This International Standard specifies the following.

- **Reference architecture for probe vehicle systems and probe data.** This reference architecture provides a general structure for probe vehicle systems within which a wide range of actual probe vehicle systems can be built whose physical characteristics may differ (e.g., in their choice of communications medium). The reference architecture is used to:
  - clarify the major building blocks and logical interconnections of probe vehicle systems for which this standard will be used;
  - categorize probe data in accordance with the information model described below.
- **Basic data framework for probe data elements and probe data.** This framework specifies how to define probe data elements and probe messages. Specifically it provides the following.
  - Rules for mapping information models (as defined in ISO 14817) of probe data to probe data elements/messages. The information models show the logical structure of entities and concepts involved in probe data.
  - Required characteristics of probe data elements and probe data messages.
  - The notation for probe data elements/messages (in XML).
  - Rules for using core data elements and basic data elements (see below), and extensions of data elements in each application domain.
- **Core data element definitions.** Core data elements are basic descriptive elements intended to appear in every probe message. These are the location and the time at which the probe data was sensed.
- **Initial set of probe data elements.** These elements will be commonly used in typical probe data enabled application domains, such as traffic, weather, and safety. Standardizing these probe data elements facilitates the development of probe vehicle systems and the distribution of probe data. This is not intended to be an exhaustive listing of probe data elements.
- **Example probe messages.** These messages define how probe data elements are combined to convey information to probe processing centres. This is not intended to be an exhaustive listing of probe messages.

Figure 1 depicts the scope described above.

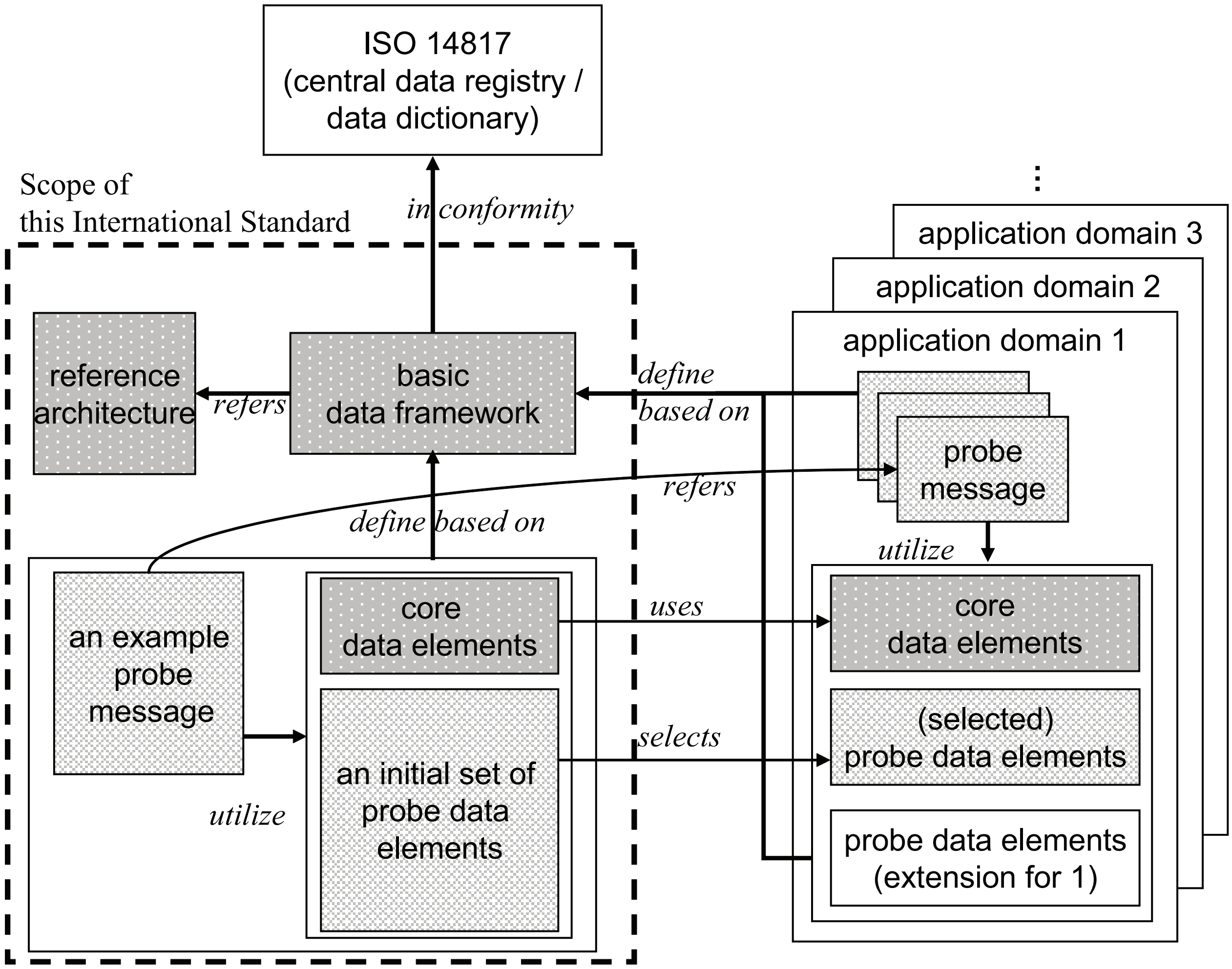

Figure 1 — Scope of this International Standard

To completely define probe processing, the standardization of probe data elements and probe messages is not sufficient. Standards are also required for processed probe data (the output of probe processing) and downlink elements and messages (to convey these results to vehicles and other users). This International Standard prescribes an initial set of probe data elements that are important for transmission from vehicles to land-side processing centres. The standardization of other probe data issues will be addressed in future work.

2 Conformance

Conforming probe data elements shall be defined based on the basic data framework.

Conforming probe messages shall be defined based on the basic data framework, and include core data elements among its members.

Conforming systems do not need to use all probe data elements in this International Standard.

Developers of probe systems may define probe data elements in addition to those listed in the normative portion of this International Standard as extensions. Parties who create extensions to the standard should be cautioned, however, that probe data elements defined outside of this International Standard may not be recognized by all probe data processing centres.

New normative probe data elements may be added to this International Standard through the ISO process.

It is not required to send confidence data in order to conform to this International Standard. If confidence is not known, the field shall be left blank.

3 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 9000, *SI units and recommendations for the use of their multiples and of certain other units*

ISO/IEC 8824-1:2002, *Information technology — Abstract Syntax Notification One (ASN.1): Specification of basic notation*

ISO 14817, *Transport information and control systems — Requirements for an ITS/TICS central Data Registry and ITS/TICS Data Dictionaries*

4 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

4.1
probe vehicle system
system consisting of vehicles which collect and transmit probe data and land-based centres which collate and process data from many vehicles to build an accurate understanding of the overall roadway and driving environment

4.2
vehicle sensor
device within a vehicle that senses conditions inside and/or outside the vehicle or that detects actions that the driver takes

4.3
probe data
vehicle sensor information, formatted as probe data elements and/or probe messages, that is processed, formatted, and transmitted to a land-based centre for processing to create a good understanding of the driving environment

4.4
probe data element
data item included in a probe message

4.5
core data element
probe data element which appears in all probe messages

4.6
probe message
structured collation of data elements suitable to be delivered to the onboard communication device for transmission to a land-based centre

NOTE It is emphasized that a probe message will not contain any information that identifies the particular vehicle from which it originated or any of the vehicle's occupants, directly or indirectly. In delivering a probe message to be transmitted by the onboard communication device, the onboard data collection system will request that the message be packaged and transmitted without any vehicle or occupant identifying information.

4.7
processed probe data
data from probe data messages which has been collated and analysed in combination with other data

5 Reference architecture

5.1 General

The reference architecture for this International Standard consists of the reference architecture for probe vehicle systems and the reference architecture for probe data.

The reference architecture for probe vehicle systems is designed to present an initial categorization of system components and the relationship among them.

The reference architecture for probe data is designed to present the initial categorization of probe data. The reference architecture for probe data is also referred as the information model for probe data.

5.2 Reference architecture for probe vehicle systems

The reference architecture for probe vehicle systems presents the initial categorization of system components and their relationships from a conceptual point of view. A component is depicted as a UML class and represents an encapsulation of functions and data that is conceptually considered as an individual entity in the probe vehicle system. A relationship is depicted as a UML association and represents potential control and/or data flow among components.

Figure 2 shows the overall structure of the reference architecture for probe vehicle systems.

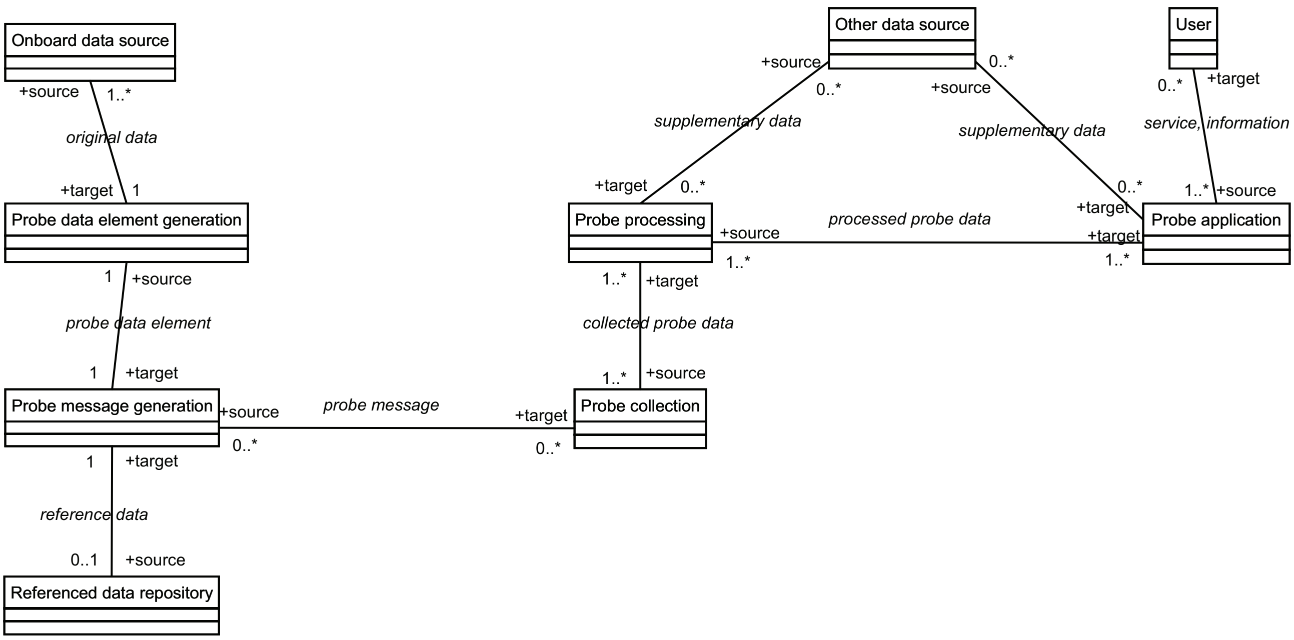

Figure 2 — Reference architecture for probe vehicle systems (overall structure)

The reference architecture comprises the following components.

- **Onboard data source.** The onboard data source provides original data that will become a probe data element. Original data may be raw sensor data or data from other onboard applications. Onboard data sources may be (various types of) sensors, onboard systems, and so on.
- **Probe data element generation.** Probe data element generation creates probe data elements from original data. All of the following cases are included.
  1) No processing (probe data element is identical to original data).
  2) Normalize original data (probe data element is the result of performing a calculation or transformation on original data).
  3) Process original data to generate a new type of data (multiple items of original data are processed, possibly over a time period, to produce the probe data element, e.g. “traffic jam detected”).
- **Probe message generation.** Probe message generation creates and formats probe messages from probe data elements and sends them to probe collection. Here, “send” is at the application layer, not the communication layer. Probe message generation manages the timing of sending messages as an application issue. Actual message transmission out the vehicle is left to the communication layer. Probe message generation may refer stored reference data, to assist with data transformation or to help determine whether a probe message should be sent.
- **Referenced data repository.** Referenced data repository holds data for reference by the probe message generator.
- **Probe collection.** Probe collection is a land-side activity that receives probe messages sent by vehicles and extracts probe data from these messages.
- **Probe processing.** Probe processing receives collected probe data from probe collection and processes it (for example, using analysis and fusion). Probe processing does not receive any information from probe collection that identifies the vehicle or driver.
- **Probe application.** Application which uses information produced by probe processing.
- **Other data source.** Other data source provides additional data that is used for probe processing and/or by probe applications. Other data sources may be road authorities, police, weather information providers, and so on.
- **User.** Entity that receives services and/or information produced from probe data. Users may be drivers, road authorities, police, weather services, public agencies, individual users (of cell phones, PDAs), and so on.

Each relationship in this reference architecture is represented as a data and/or control flow, defined as follows:

- **Original data.** Data used for probe data generation. Original data may be raw sensor data or data from other onboard applications.
- **Reference data.** Data stored in a repository and referred to for probe data generation. Reference data may be (among other things) historical data and/or statistical data.
- **Probe data element.** The result of formatting original data into a form suitable to be incorporated as an element of a probe message. (For the general definition of probe data element, see 4.4.)
- **Probe message.** A message in the application layer. A probe message consists of several probe data elements (always including core data elements) that convey meaningful information to centre-side probe collection components. (For the general definition of probe message, see 4.6.)

- **Collected probe data.** Probe data collected by the probe collection component, to be sent to probe processing components.
- **Supplementary data.** Data from other data sources (non-vehicle) that is also used in probe processing and/or by probe applications.
- **Processed probe data.** The result of fusing and analysing probe data in combination with supplementary data; the result of probe processing. (For the general definition of processed probe data, see 4.7.)
- **Service information.** The value-added result of combining processed probe data with supplementary data for delivery to users.

5.3 Reference architecture for probe data (information model)

The reference architecture for probe data represents the initial categorization of probe data from a conceptual point of view.

The reference architecture for probe data consists of multiple packages in UML notation. Each package includes conceptual entities that are identified from a specific point of view. Each conceptual entity is depicted as a UML class.

Figure 3 is the overall package structure of the reference architecture for probe data.

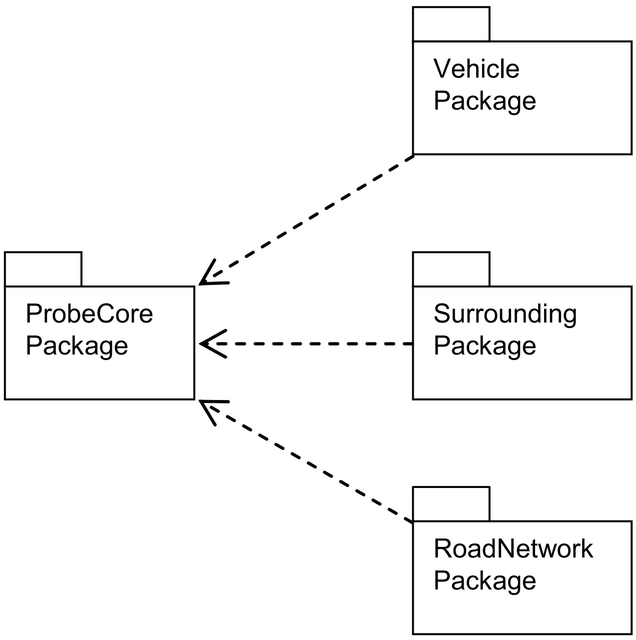

Figure 3 — Reference architecture for probe vehicle systems (Package structure)

Each package includes the conceptual entities from a different specific viewpoint. Each conceptual entity is an object class. Each class has properties. Figure 4 shows the conceptual entities for the ProbeCorePackage, for example. All of the packages and entities are explained in Annex A.

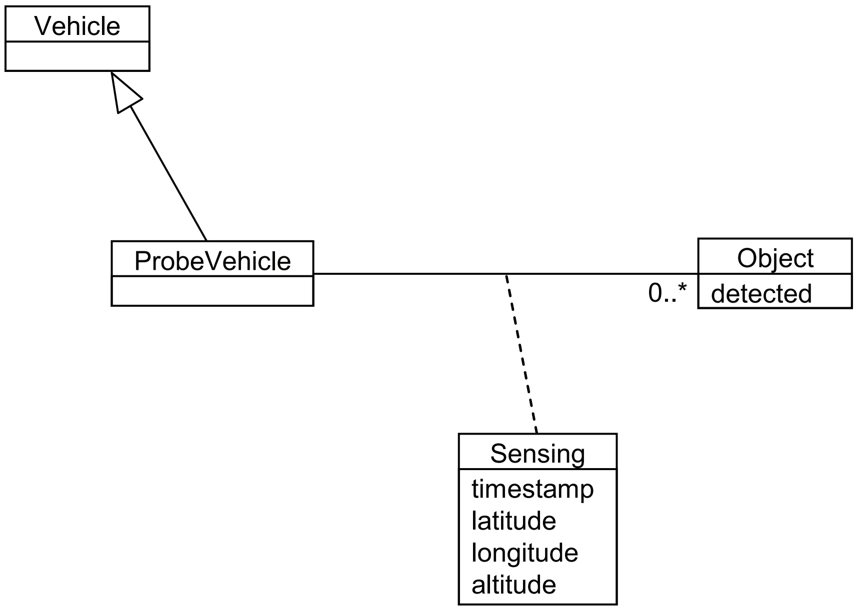

Figure 4 — Entities of ProbeCorePackage

6 Basic data framework

6.1 General

The basic data framework specifies the requirement for defining probe data elements and probe messages.

6.2 Probe data element

The following are the requirements for defining a probe data element.

- A probe data element is shown by a pair consisting of a property of a class and a value domain. That is, it is expressed in the form of ObjectClassTerm.propertyTerm:value-domain-term.

- Each probe data element shall have the following meta-attributes; those basic meta-attributes specified as mandatory in ISO 14817 or defined as optional in ISO 14817 but mandatory for probe data elements.

  — **descriptive name**: A name of the probe data element in the form of "ObjectClassTerm.propertyTerm:value-domain-term". Descriptive name is used for the identification of the probe data element.

  — **ASN.1 name**: The ASN.1 Name shall be the name of a data concept expressed as a valid "typereference" as defined in 11.2 of ISO/IEC 8824-1:2002.

  — **ASN.1 object identifier**: A unique ASN.1 object identifier in accordance with ISO/IEC 8824-1.

  — **definition**: A statement in natural-language text that expresses the essential meaning of the probe data element and assists humans in differentiating the data element from all other data elements.

  — **descriptive name context**: A designation of the ITS/TICS functional area within which the descriptive name is relevant. The descriptive name context for each probe data element is "probe".

— **data concept type**: A categorization of the kind of data concept. The data context type of each probe data element is “data element”.

— **standard**: The alphanumeric designation of the standard, or other reference, that defines and describes the probe data element, typically the functional Data Dictionary standard that defines the probe data element.

— **data type**: The logical representation of the probe data element as expressed as a valid data concept instance of an ASN.1 data type.

— **format**: A natural language description of the logical layout of the data concept to facilitate interchange of data.

— **unit of measure**: Units shall be defined in accordance with ISO 1000. For units of enumeration, such as equipment or units of issue, the standard measure shall be defined using this meta-attribute.

— **valid value rule**: A natural language text definition of the rule(s) by which permissible legal instances of a probe data element are identified.

— **data quality**: Specifies the details of data quality for a probe data element. Multiple items may be required to describe data quality, with some items being qualitative and others quantitative.

- When a probe data dictionary is registered to the data registry, it shall comply with ISO 14817; administrative meta-attributes which are mandatory shall be described.

6.3 Probe messages

The following are the requirements for defining a probe message.

- Each probe message consists of a set of probe data elements sent to a centre from the vehicle as a unit.
- Each probe message consists of core data elements and probe data elements.
- Each probe data element included in each probe message satisfies the necessary conditions for probe data elements described above.
- Each probe message shall have the basic meta-attributes defined as mandatory in ISO 14817.

— **descriptive name**: A name of the probe data message in the form of “MessageTerm:message”. Descriptive name is used for the identification of a probe message.

— **ASN.1 name**: The ASN.1 name shall be the name of a data concept expressed as a valid “typereference” as defined in 11.2 of ISO/IEC 8824-1:2002.

— **ASN.1 object identifier**: A unique ASN.1 object identifier in accordance with ISO/IEC 8824-1.

— **definition**: A statement in natural-language text that expresses the essential meaning of the probe data message and assists humans in differentiating the message from all other messages.

— **descriptive name context**: A designation of the ITS/TICS functional area within which the descriptive name is relevant. Descriptive name context for each probe data message is “probe”.

— **data concept type**: A categorization of the kind of data concept. The data context type of each probe data message is “message”.

— **architecture reference**: The name of one or more ITS/TICS Architecture “architecture flow”(s) with corresponding architecture source (subsystem or terminator) and architecture destination (subsystem or terminator) into which this data concept can be meaningfully categorized in whole or in part.

- **architecture name**: The designator (e.g., the title or number) of an ITS/TICS or other architecture that contains the architecture reference(s).
- **architecture version**: The version number of an ITS/TICS or other architecture that contains the architecture reference(s).
- **metadata source**: Indicates whether or not each data element in the message is defined in this dictionary; here “direct”, which means all probe data elements in probe data messages are defined in this dictionary.
- **priority**: Indicates whether a message should receive priority treatment. If applicable, the priority scheme and/or the priority of the message may be specified.
- **frequency / message mode**: Indicates the expected timing or rate of occurrence of an instance of this message. Additionally, indicates the message mode for periodic messages.
- **referenced data frames**: A set of data frames which are involved in the message. Multiples allowed.
- **referenced data elements**: A set of data elements which are involved in the message. To identify these referenced data elements, their descriptive names are used.
- **data type**: The logical representation of the message as expressed as a valid message instance of an ASN.1 data type. The text of this meta-attribute shall consist of a complete and syntactically correct ASN.1 module definition.

6.4 Notation

Probe data elements and probe messages defined in this framework are defined in the following XML format.

```
<?xml version="1.0" encoding="UTF-8" ?>

<probe_dictionary>

<probe_data_element descriptive_name="(descriptive name written here)">

    <ASN.1_name> <!-- ASN.1 name--> </ASN.1_name>

    <ASN.1_object_identifier> <!-- ASN.1 object identifier --> </ASN.1_object_identifier>

    <definition> <!-- definition --> </definition>

    <descriptive_name_context>probe</descriptive_name_context>

    <data_concept_type>data element</data_concept_type>

    <standard> <!-- standard, if any --> </standard>

    <data_type> <!-- data type --> </data_type>

    <format> <!-- format --> </format>

    <unit_of_measure> <!-- unit of measure --> </unit_of_measure>

    <valid_value_rule> <!-- valid value rule --> </valid_value_rule>

    <data_quality> <!-- data quality --> </data_quality>

</probe_data_element>
```

```
<probe_message descriptive_name="(descriptive name written here)">

    <ASN.1_name> <!-- ASN1. name --> </ASN.1_name>

    <ASN.1_object_identifier> <!-- ASN.1 object identifier --> </ASN.1_object_identifier>

    <definition> <!-- definition --> </definition>

    <descriptive_name_context>probe</descriptive_name_context>

    <data_concept_type>message</data_concept_type>

    <architecture_reference> <!-- architecture reference --> </architecture_reference>

    <architecture_name> <!-- architecture name --> </architecture_name>

    <architecture_version> <!-- architecture version --> </architecture_version>

    <metadata_source> <!-- metadata source --> </metadata_source>

    <priority> <!-- priority --> </priority>

    <frequency> <!-- frequency --> </frequency>

    <referenced_data_frames> <!--referenced data frames --> </referenced_data_frames>

    <referenced_data_elements>

    <probe_data_elementref

        refdescriptive_name="(descriptive name of referenced element written here)"/>

    </referenced_data_elements>

    <data_type> <!-- data type --> </data_type>

</probe_message>

</probe_dictionary>
```

7 Core data elements

7.1 Concept of core data elements

“Core data elements” are meta-information added to all probe data elements and probe messages. Core data elements consist of a timestamp which shows the time when the probe element was sensed and a location stamp which specifies the vehicle’s location at the time the probe element was sensed.

Probe data elements are core data elements and normalized raw sensor data based on “an initial set of probe data elements” (see Clause 8).

The concept of core data elements is shown in Figure 5.

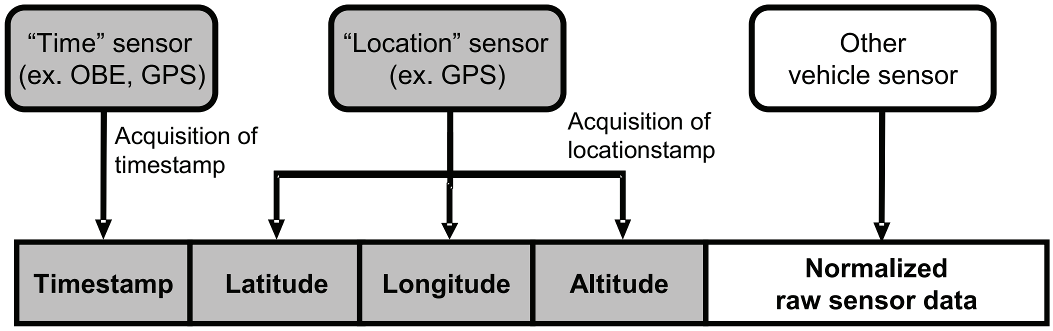

Figure 5 — Concept of core data elements

7.2 Timestamp

The timestamp is expressed as the elapsed time since January 1, 1970 (it is the same as "UNIX epoch time"). This method makes it easy to calculate the elapsed time between two timestamps.

7.3 Locationstamp

7.3.1 General

The locationstamp specifies the geographic location of the vehicle at the time that a sensor reading was collected. It consists of "Latitude", "Longitude" and "Altitude".

In order to use a position effectively, it is necessary to know how accurate the stated position is believed to be. This is expressed as the "confidence" in the measured value. Location confidence is represented in metres and two decimal places of metres. It is not mandatory to use all of the available decimal places. Location confidence is expressed as an average error range from the point shown by "latitude", "longitude" and "altitude". For example, when GPS is used, location confidence is the "estimated position error" (EPE) as defined in GPS ICD 200[4]. This confidence is represented in terms of a value domain.

In Table 1, "[L...H]" indicates a range of values between "L" and "H".

In Table 2, value domains that are used for representing confidence are listed.

7.3.2 Latitude

Latitude is expressed in degrees as a real number. Positive numbers correspond to north latitudes and negative numbers to south latitudes.

7.3.3 Longitude

Longitude is expressed in degrees as a real number. Positive numbers correspond to east longitudes and negative numbers to west longitudes.

7.3.4 Altitude

Altitude is expressed in metres, relative to sea level, as an integer number. Positive numbers correspond to altitudes above sea level, and negative numbers correspond to altitudes below sea level.

Table 1 — Core data elements

Name	Description	Data source	Data type	Format	Unit of measure	Valid value rule	Data quality
Sensing.timestamp: real	Based on 1 January, 1970 (same as "UNIX epoch time").	Geographical location sensor (e.g. GPS)	REAL	REAL	second	real	n.a.
Sensing.latitude: lctn-in-degree-with-confidence	Latitude of the vehicle observed by the location position sensor, and accuracy of a sensor.	Geographical location sensor (e.g. GPS)	LocationDegreeWithConfidence ::= SEQUENCE { degree REAL, confidence REAL }	latitude [–90…90] Format of confidence is expressed in REAL.	degree, millimetre	real [–90…90], confidence is any real number	n.a.
Sensing.longitude: lctn-in-degree-with-confidence	Longitude of the vehicle observed by the sensor, and accuracy of a sensor.	Geographical location sensor (e.g. GPS)	LocationDegreeWithConfidence ::= SEQUENCE { degree REAL, confidence REAL }	longitude [–180…180] Format of confidence is expressed in REAL.	degree, millimetre	real [–180…180], confidence is any real number	n.a.
Sensing.altitude: lctn-in-altitude-with-confidence	Altitude of the vehicle observed by the sensor, and accuracy of a sensor.	Geographical location sensor (e.g. GPS)	LocationAltitudeWithConfidence ::= SEQUENCE { altitude INTEGER, confidence REAL }	INTEGER Format of confidence is expressed in REAL.	metre, metre	integer [–65 535…65 535], confidence is any real number	n.a.

Table 2 — Value domain

Descriptive name	Definition	Data type	Format	Unit of measure	Valid value rule
lctn-in-degree-with-confidence	Expression of location in degrees with expression of the confidence for this location data	LocationDegreeWithConfidence ::= SEQUENCE { degree REAL, confidence REAL }	Value expressed in degrees and millimetres	degree, millimetre (confidence)	degree is a real number from –180 to 180, and confidence is any real number
lctn-in-altitude-with-confidence	Expression of location in altitude with expression of the confidence for this location data	LocationAltitudeWithConfidence ::= SEQUENCE { altitude INTEGER, confidence REAL }	Value expressed in a pair of metres	metre, metre (confidence)	altitude is an integer from –65 535 to 65 535, and confidence is any real number

8 Normative data elements

8.1 General

This clause lists normative probe data elements. These data elements are derived from onboard sensors and onboard applications and are available (or can be made available) on the vehicle databus. These data elements relate to application areas such as traffic, weather, road conditions, safety, and navigation.

This list of data elements is intended to be sufficient for first-generation probe vehicle systems. Developers of specific probe vehicle system implementations may not use all of the data elements listed.

8.2 Data elements

In Table 3, data elements are listed along with key parameters. The XML representation is provided in Annex B.

Some data elements have a confidence value. If a data element has a value V, and the data element also has a confidence value C, this means that the reporting vehicle believes the data value is between $V–C$ and $V+C$.

In Table 3, "[L...H]" indicates a range of values between "L" and "H".

In Table 4, value domains that are used for data element definitions are listed.

Table 3 — Normative probe data elements

Name	Description	Data source	Data type	Format	Unit of measure	Valid value rule	Data quality

Environment. temperature:qty-degrees-Celsius-with-confidence	Ambient air temperature with confidence	Temperature sensor	QtyDegreesCelsiusWithConfidence ::= SEQUENCE {
			degrees INTEGER,
			confidence INTEGER
			}
				Air temperature. Format of confidence is expressed in degrees Celsius.	degree Celsius, degree Celsius.	integer [–49...50, FFFF = unknown], integer [0...20]	n.a.

Wiper.status: integer	Activation of wipers and operational mode (intermittent, slow, fast)	Wiper switch	INTEGER	0 = no wiper active, 1 = intermittent, 2 = slow, 3 = fast	concatenated code	integer [0...3]	n.a.
Environment. rainfallIntensity: integer	Rate at which rainfall occurs, expressed in units of depth per unit of time.	Rain sensor	INTEGER	Rainfall intensity is expressed in millimetres/hour	millimetre per hour	integer [0…999]	n.a.

ExteriorLights. status:code-exterior-light-status	Status of all vehicle exterior lights	Light status information on vehicle data bus	CodeExteriorLightStatus ::= SEQUENCE {
			parkinglight INTEGER,
			lowbeam INTEGER,
			highbeam INTEGER,
			foglights INTEGER,
			automaticlightcontrol INTEGER,
			turnhazardsignal INTEGER
			}
				For parkinglight, lowbeam, highbeam, foglights, automaticlightcontrol:
				0 = OFF, 1 = ON
				For turnhazardsignal:
				0 = off,1 = left,
				2 = right, 3 = hazard
					code	integer [0...1],
						integer [0...1],
						integer [0...1],
						integer [0...1],
						integer [0...1],
						integer [0...3]
							n.a.

Table 3 (*continued*)

Name	Description	Data source	Data type	Format	Unit of measure	Valid value rule	Data quality
Environment.light Condition:integer	Outside light conditions: degree of light in environment (bright to dark)	Light level sensor	INTEGER	Segmentation of direct measurement of light energy (lux) 0 = 0 to 1 lux 1 = 2 to 100 lux 2 = 101 to 1 000 lux 3 = 1 001 to 30 000 lux 4 = 30 001 to 50 000 lux 5 = 50 001 to 80 000 lux 6 = 80 001 to 100 000 lux 7 = over 100 000 lux	code	integer [0…7]	n.a.

Vehicle.velocity: rt-velocity-with-confidence	Vehicle velocity with confidence data	Velocity: vehicle speed sensor
		Confidence: an accuracy figure inserted in onboard database by car-maker
			RateVelocityWithConfidence :: = SEQUENCE {
			velocity INTEGER,
			confidence INTEGER
			}
				Current vehicle speed. Format of confidence is expressed in metres/second.	metre per second, metre per second	integer [0...99], integer [0...100]	n.a.

Obstacle.detected: boolean	Obstacle presence	Forward obstacle sensing system	BOOLEAN	Obstacle in current lane of vehicle at or near current position 1 = obstacle present	code	0 or 1	n.a.
Obstacle.distance: integer	Obstacle distance from vehicle detecting and reporting the obstacle	Forward obstacle sensing system	INTEGER	Distance from reporting vehicle at time of reporting	decimetre	integer [0…999]	n.a.
Obstacle.direction: integer	Obstacle direction from vehicle detecting and reporting the obstacle	Forward obstacle sensing system	INTEGER	Azimuth relative to forward direction of vehicle	degree	integer [–90… 90]	n.a.

Table 3 (*continued*)

Name	Description	Data source	Data type	Format	Unit of measure	Valid value rule	Data quality
AntiLockBrake System.status: boolean	activation of anti-lock braking system, which indicates poor road traction	ABS system	BOOLEAN	1 = ABS activated	code	0 or 1	n.a.
TractionControl System.status: boolean	activation of traction control system, which indicates poor road traction	Traction control system	BOOLEAN	1 = TC activated	code	0 or 1	n.a.
VehicleStability Control.status: boolean	activation of vehicle stability control (also called electronic stability program), which is indicative of hazardous conditions	VSC/ESP system	BOOLEAN	1 = ESP/VSC activated	code	0 or 1	n.a.
Vehicle.GForce: integer	Vertical G force to indicate a pothole or bumpy road; reported based on exceeding a threshold	Vertical G force sensors	INTEGER	Vertical G force as measured at wheel Signed.	tenth of G	integer [–99…99]	n.a.
Vehicle.acceleration: rt-acceleration-with-confidence	detection of acceleration/ deceleration by vehicle, which exceeds a defined threshold, with confidence	Acceleration: longitudinal accelerometer. Confidence: an accuracy figure inserted in onboard database by car-maker.	RateAccelerationWithConfidence :: = SEQUENCE { acceleration INTEGER, confidence INTEGER }	Detection of acceleration by vehicle, which exceeds a defined threshold. Signed (format of confidence is expressed in cm/s^2).	centimetre per second squared, centimetre per second squared	integer [0...3 000], integer [0...1 000]	n.a.
Brake.status: integer	Brake Activation and Force	Brake force signal	INTEGER	Activation of vehicle brakes either by driver or automatic system, and the degree of braking, expressed as a percentage of full braking force.	scale [0…99]	integer [0…99]	n.a.

Table 3 (*continued*)

Name	Description	Data source	Data type	Format	Unit of measure	Valid value rule	Data quality
Brake.boostAssist: integer	Engagement of brake boost assist function, as an indication of an emergency situation	Brake boost assist system	INTEGER	0 = not activated 1 = activated	code	integer [0…1]	n.a.

Vehicle.yawRate: rt-yaw-rate-with-confidence	Rate of change of yaw angle, which can indicate hazardous or emergency situations, with confidence	Yaw rate: yaw rate sensor.
		Confidence: an accuracy figure inserted in onboard database by car-maker.
			RateYawRateWithConfidence :: = SEQUENCE {
			yaw-rate INTEGER,
			confidence INTEGER
			}
				Rate of change of yaw angle. Format of confidence is expressed in degrees/second.	degree per second, degree per second	integer [0...359], integer [0...359]	n.a.

FuellingSystem. fuelConsumption: integer	current fuel consumption	Onboard fuel consumption data from ECU	INTEGER	Fuel consumption rate	millilitre per minute	integer [0…999]	n.a.
FuellingSystem. averageFuel Consumption: integer	average fuel consumption	Onboard fuel consumption data from ECU	INTEGER	Average fuel consumption rate	millilitre per minute	integer [0…999]	n.a.
Vehicle.stoppage Time:integer	time during which vehicle stopped with engine idling	Derived from vehicle clock and vehicle velocity	INTEGER	Time elapsed	tens of seconds	integer [0…999]	n.a.
Vehicle.engine Stopped Time:integer	period of time that the engine is stopped	Vehicle clock and engine ECU	INTEGER	Time elapsed	minute	integer [0…999]	n.a.

Vehicle.lateral Acceleration:rt-acceleration-with-confidence	lateral acceleration experienced by vehicle, which can indicate an emergency manoeuvre, with confidence	Lateral accelero-meter	RateAccelarationWithConfidence :: = SEQUENCE{
			lateralAcceleration INTEGER,
			confidence INTEGER
			}
				Rate of change of lateral acceleration. Format of confidence is expressed in cm/s^2.	centimetre per second squared, centimetre per second squared	integer [0... 3 000], integer [0...1 000]	n.a.

Table 3 (*continued*)

Name	**Description**	**Data source**	**Data type**	**Format**	**Unit of measure**	**Valid value rule**	**Data quality**
Path.exception Variance:integer	Vehicle path on road is different from path indicated in map database, indicating map database needs to be updated. This data element is only a flag that this situation is detected.	Navigation system	INTEGER	Once detected, vehicle reports variance every 20 m or once per second, whichever is greater. 1 = path exception detected	code	integer [0…1]	n.a.

Vehicle.direction: qty-direction-with-confidence	Current direction of vehicle travel.
	This data element supports other data elements such as obstacle detection.
		Navigation system or onboard compass	QtyDirectionWithConfidence :: = SEQUENCE {
			direction INTEGER,
			confidence INTEGER
			}
				Degrees relative to North: format of confidence is expressed in tenths of degrees.	tenth of degree, tenth of degree	integer [0…3 600], integer [0…1 000]	n.a.

Table 3 (*continued*)

Name	Description	Data source	Data type	Format	Unit of measure	Valid value rule	Data quality
Vehicle.vehicle Type:integer	Type of vehicle	Configuration in onboard computer	INTEGER	0 = unknown, 1 = passenger car, 2 = light truck, 3 = heavy truck (> 5 000 kg) (axis count unspecified), 4 = bus, 5 = motorcycle, 6 = articulated truck, 7 = car with trailer, 8 = truck with trailer (axis count unspecified), 9 = high sided vehicle, 10 = heavy truck (axis count = 2), 11 = heavy truck (axis count = 3), 12 = heavy truck (axis count = 4), 13 = heavy truck (axis count = 5), 14 = heavy truck (axis count = 6), 15 = heavy truck (axis count = 7), 16 = truck with trailer (axis count = 2), 17 = truck with trailer (axis count = 3), 18 = truck with trailer (axis count = 4), 19 = truck with trailer (axis count = 5), 20 = truck with trailer (axis count = 6), 21 = truck with trailer (axis count = 7), 22 to 255 = local definition	code	integer [0…255]	n.a.

Table 3 (*continued*)

Name	**Description**	**Data source**	**Data type**	**Format**	**Unit of measure**	**Valid value rule**	**Data quality**
Vehicle.vehicle Usage:integer	Usage of vehicle	Configuration in onboard computer	INTEGER	0 = unknown, 1 = private use, 2 = taxi, 3 = commercial, 4 = public transport, 5 = emergency services, 6 = patrol services, 7 = road operator, 8 = snow plough, 9 = hazmat (hazardous material), 10 = other, 11 to 255 = local definition	code	integer [0…255]	n.a.
Vehicle.sudden Steering Manoeuver: integer	Detection of emergency steering manoeuvre made by driver, based on: angular rotation rate of steering wheel, which exceeds a defined threshold, or lateral acceleration.	Steering angular velocity sensor and/or steering angle sensor	INTEGER	Angular rotation rate of steering wheel	degree per second	integer [0…359]	n.a.
LaneMark. detected:integer	Status of lane marking detection	Derived from lane detection systems	INTEGER	0 = lane marking not detected 1 = lane marking detected	code	integer [0...1]	n.a.
Road. longitudinal SlopeScale: integer	Current longitudinal slope scale of vehicle travel	Gyroscope	INTEGER,	Degrees relative to the horizontal	tenth of degree	integer [–899…900]	n. a.

Table 3 (*continued*)

Name	Description	Data source	Data type	Format	Unit of measure	Valid value rule	Data quality

Seatbelt.status: code-seatbelt-status	Seat-belt status reported consistent with AMI-C message set. Addresses all possible seat locations.
	This data element may be used for data cleansing and would not typically be aggregated.
		Seat-belt occupancy sensors or seat-belt closure detection	CodeSeatbeltStatus ::= SEQUENCE {
			driver INTEGER,
			middlefront INTEGER,
			passenger INTEGER,
			secondrowleft INTEGER,
			secondrowmiddle INTEGER,
			secondrowright INTEGER,
			thirdrowleft INTEGER,
			thirdrowmiddle INTEGER,
			thirdrowright INTEGER,
			fourthrowleft INTEGER,
			fourthrowmiddle INTEGER,
			fourthrowright INTEGER,
			fifthrowleft INTEGER,
			fifthrowmiddle INTEGER,
			fifthrowright INTEGER
			}
				For each seat,
				0 = not equipped,
				1 = non-fastened,
				2 = fastened
					concatenated code	integer [0…2],
						integer [0…2],
						integer [0…2],
						integer [0…2],
						integer [0...2],
						integer [0...2],
						integer [0...2],
						integer [0...2],
						integer [0...2],
						integer [0...2],
						integer [0...2],
						integer [0...2],
						integer [0...2],
						integer [0...2],
						integer [0...2]
							n.a.

Door.status: boolean	Indication that one or more vehicle doors are open. This data element may be used for data cleansing and would not typically be aggregated.	Door sensors	BOOLEAN	Data expresses “door open warning” is on or off 0 = off 1 = on	code	0 or 1	n.a.
Trunk.status: boolean	Vehicle trunk is open. This data element may be used for data cleansing and would not typically be aggregated.	Trunk latch sensor	BOOLEAN	0 = closed, 1 = open	code	0 or 1	n.a.
ParkingBrake. status:boolean	Parking brake engaged. This data element may be used for data cleansing and would not typically be aggregated.	Parking brake engagement switch	BOOLEAN	0 = off, 1 = on	code	0 or 1	n.a.

Table 4 — Value domain

Descriptive name	Definition	Data type	Format	Unit of measure	Valid value rule

qty-degrees-Celsius-with-confidence	Value expressed in degrees Celsius, with confidence data.	QtyDegreesCelsiusWithConfidence :: = SEQUENCE {
		degrees INTEGER,
		confidence INTEGER
		}
			Value expressed in a pair of degrees Celsius	degree Celsius, degree Celsius	integer [−∞...∞],
					integer [0...10]

code-exterior-light-status	Status of a set of vehicle exterior lights	CodeExteriorLightStatus ::= SEQUENCE {
		parkinglight INTEGER,
		lowbeam INTEGER,
		highbeam INTEGER,
		foglights INTEGER,
		automaticlightcontrol INTEGER,
		turnhazardsignal INTEGER
		}
			Value expressed in a sequence of numbers	code	

rt-velocity-with-confidence	Velocity with confidence data	RateVelocityWithConfidence :: = SEQUENCE {
		velocity INTEGER,
		confidence INTEGER
		}
			Value expressed in a pair of meters/second	metre per second, metre per second	integer [0...∞],
					integer [0...100]

rt-acceleration-with-confidence	Acceleration with confidence data	RateAccelerationWithConfidence :: = SEQUENCE {
		acceleration INTEGER,
		confidence INTEGER
		}
			Value expressed in a pair of cm/s^2	centimetre per second squared, centimetre per second squared	integer [0...∞],
					integer [0...10 000]

rt-yaw-rate-with-confidence	Yaw rate with confidence data	RateYawRateWithConfidence :: = SEQUENCE {
		yaw-rate INTEGER,
		confidence INTEGER
		}
			Value expressed in a pair of degrees/second	degree per second, degree per second	integer [0...359],
					integer [0...359]

qty-direction-with-confidence	Direction with confidence data	QtyDirectionWithConfidence :: = SEQUENCE {
		direction INTEGER,
		confidence INTEGER
		}
			Value expressed in a pair of tenths of degrees	tenth of degree, tenth of degree	integer [0…3 600],
					integer [0…1 000]

Table 4 (*continued*)

Descriptive name	Definition	Data type	Format	Unit of measure	Valid value rule

code-seatbelt-status	Status of a set of each seat-belt	CodeSeatbeltStatus ::= SEQUENCE {
		driver INTEGER,
		middlefront INTEGER,
		passenger INTEGER,
		secondrowleft INTEGER,
		secondrowmiddle INTEGER,
		secondrowright INTEGER,
		thirdrowleft INTEGER,
		thirdrowmiddle INTEGER,
		thirdrowright INTEGER,
		fourthrowleft INTEGER,
		fourthrowmiddle INTEGER,
		fourthrowright INTEGER,
		fifthrowleft INTEGER,
		fifthrowmiddle INTEGER,
		fifthrowright INTEGER
		}
			Value expressed in a sequence of numbers	code	

Annex A
(normative)

Reference architecture for probe data (information model) for normative data elements

A.1 General

This annex provides a reference architecture for probe data (an information model). This information model provides ObjectClassTerms and propertyTerms as described in basic data framework. See 5.3 and Clause 6 for more detail.

A.2 Notation

The information model is depicted using the subset of UML identified in ISO 14817. See ISO 14817 and the OMG Unified Modeling Language Specification for more detail.

A.3 Information Model

This clause explains the contents of the information model.

The information model includes every data element concept that is required to define probe data elements in this International Standard. The information model also includes additional data element concepts to clarify the meaning of each data element concept and relations among them.

A.4 Package Structure

Figure A.1 shows the package structure of the information model.

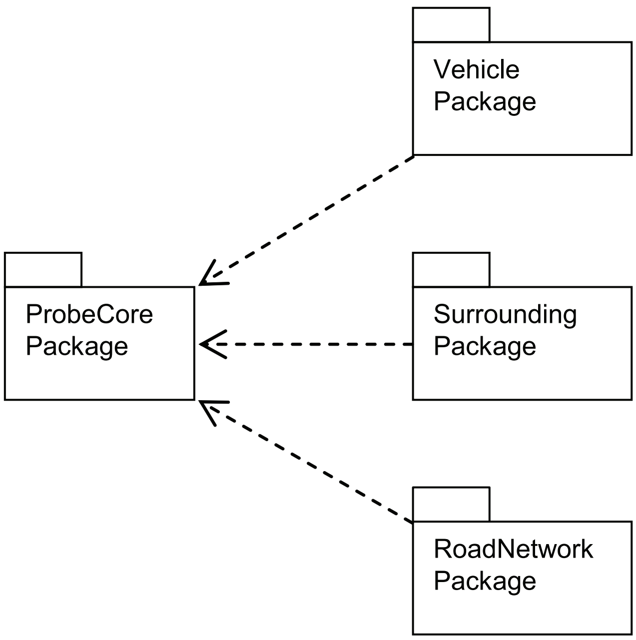

Figure A.1 — Package structure of information model

“ProbeCorePackage” defines data element concepts for core data elements. As this package defines the core concepts, every other package depends on this package.

“VehiclePackage” defines data element concepts that correspond to properties of the vehicle itself and components of the vehicle.

“SurroundingPackage” defines data element concepts that correspond to properties of the road, environment and every object physically surrounding the vehicle.

“RoadNetworkPackage” defines data element concepts that correspond to properties of the road network.

A.5 ProbeCorePackage

Figure A.2 shows “ProbeCorePackage” and Table A.1 gives its class descriptions.

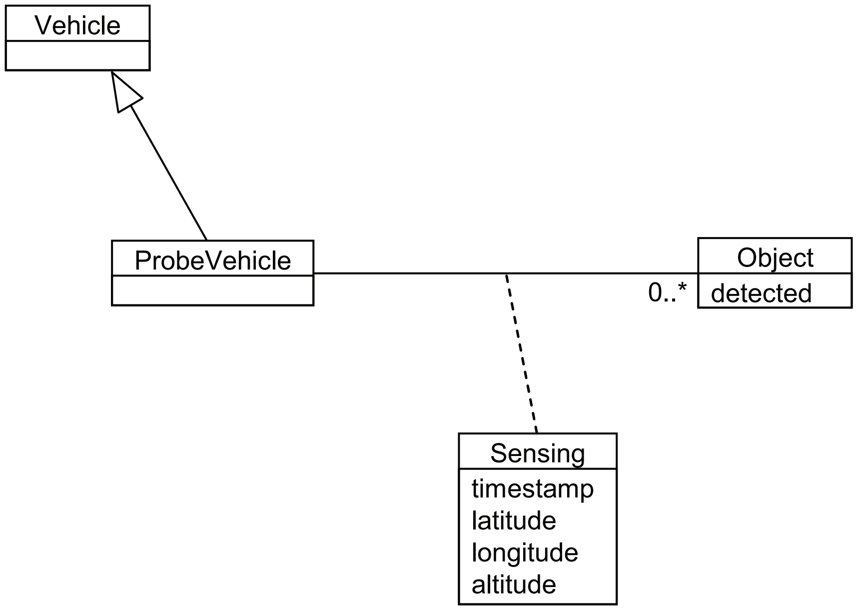

Figure A.2 — ProbeCorePackage

“ProbeCorePackage” defines data element concepts for core data elements.

This package describes the core structure of probe data; i.e. “ProbeVehicle” senses some “Object”. “Sensing” is an association class that holds sensing properties, i.e. “timestamp” and “locationstamp” (“latitude”, “longitude”, and “altitude”).

Every class defined in other packages is a subclass of “Object”, as they are sensed by “ProbeVehicle”, and we can define “timestamp” and “locationstamp” (that corresponds to core data elements) for every sensing event.

Table A.1 — Class descriptions for ProbeCorePackage

Class	Property	Description
Vehicle		A self-propelled conveyance that runs on tires, such as an automobile, truck or bus.
ProbeVehicle		A “Vehicle” capable of sensing, collecting and processing data about itself and its environment.
Object		Anything whose properties can be sensed by a “ProbeVehicle”.
	detected	The sensing of an “Object” (or its properties) by a “ProbeVehicle”.
Sensing		The event by which some property of an “Object” is sensed or collected by a “ProbeVehicle”.
	timestamp	The time of the sensing.
	latitude	The latitude of the “ProbeVehicle” at the time of the sensing.
	longitude	The longitude of the “ProbeVehicle” at the time of the sensing.
	altitude	The altitude of the “ProbeVehicle” at the time of the sensing.

The property “detected” indicates that for all classes (subclasses of “Object” implicitly have properties that indicate the event in which the object is detected). For example, class “Vehicle” has properties “detected” (detection of the object itself). Note that this International Standard uses the naming rule of ISO 14817, in which the usage of upper case and lower case in the property name is strictly defined.

A.6 VehiclePackage

Figure A.3 shows the “VehiclePackage” and Table A.2 gives its class descriptions.

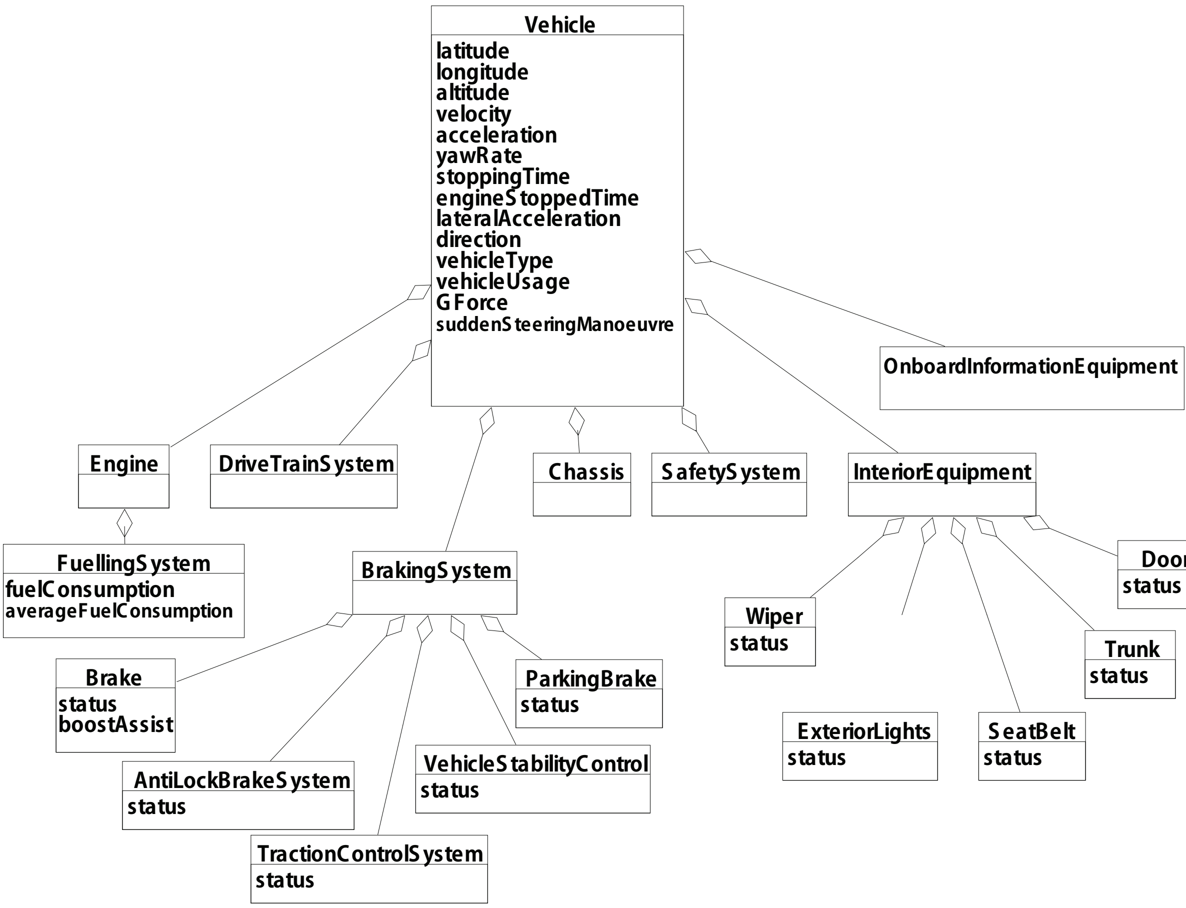

Figure A.3 — VehiclePackage

“VehiclePackage” defines data element concepts that correspond to properties of the vehicle itself and components of the vehicle.

The properties of the vehicle itself are represented as properties of “Vehicle”. Components of the vehicle are categorized into “Engine”, “DriveTrain”, “Chassis”, “Brake”, “Safety”, “Interior” and “Information”. Concrete vehicle components such as “Brake” and “Wiper” are depicted as a part of this categorization.

Table A.2 — Class descriptions for VehiclePackage

Class	Property	Description
Vehicle		See “ProbeCorePackage”.
	latitude	See “ProbeCorePackage”. (Though this property is used for defining core data elements, it is also described here for completeness.)
	longitude	See “ProbeCorePackage”. (Though this property is used for defining core data elements, it is also described here for completeness.)
	altitude	See “ProbeCorePackage”. (Though this property is used for defining core data elements, it is also described here for completeness.)
	velocity	Driving speed.
	acceleration	Rate of increase in driving speed.
	yawRate	Rate of vehicle horizontal rotation.
	stoppingTime	The time at which driving ends.
	engineStoppedTime	The time at which the engine is turned off.
	lateralAcceleration	Rate of increase in lateral motion.
	direction	The direction of vehicle travel.
	vehicleType	The type of “Vehicle”.
	vehicleUsage	The usage of “Vehicle”.
	GForce	Measured at wheel to indicate a pothole or bumpy road.
	suddenSteeringManoeuvre	Emergency steering manoeuvre made by driver.
Engine		A machine that produces power by burning fuel internally. It consists of the engine proper, fuelling system and other components.
FuellingSystem		A system in charge of fuel management.
	fuelConsumption	The rate of fuel usage.
	averageFuelConsumption	Average rate of fuel usage.
DriveTrainSystem		A system that transfers the engine power to the tires. It includes the transmission, propeller shaft, differential and drive shaft.
Chassis		A portion of a vehicle excluding the body, engine and drive train. It consists of the suspension system, steering system and the running gear, including the tires and other components.
BrakingSystem		The braking system consists of the foot brake and parking brake systems and various control systems such as an antilock braking system (ABS), traction control system (TCS), vehicle stability control (VSC) system and others.

Table A.2 (*continued*)

Class	property	description
Brake		Equipment to make the vehicle slow or stop.
	status	Status of “Brake”.
	boostAssist	Brake boost assist function.
AntiLockBrakeSystem		Equipment that prevents tire locking from sudden braking or stopping on slippery pavement.
	status	Status of “AntiLockBrakeSystem”.
TractionControlSystem		Equipment that prevents slipping.
	status	Status of “TractionControlSystem”.
VehicleStabilityControl		Equipment that helps provide stable driving.
	status	Status of “VehicleStabilityControl”.
ParkingBrake		Auxiliary brake for parking. Side brake.
	status	Status of “ParkingBrake”.
SafetySystem		Safety equipment is divided between passive safety systems that mitigate occupant injury in the event of a traffic crash and active safety systems that reduce the likelihood of a traffic crash.
InteriorEquipment		The devices, instruments and components used in the cabin. It comprises various control switches for the headlamps, turn signal indicators, fog lamps, windshield wipers, doors, trunk, seat-belts and so on.
Wiper		Equipment that clears or cleans front or rear glass.
	status	Status of “Wiper”.
SeatBelt		Safety belt to protect an occupant in a crash.
	status	Status of “SeatBelt”.
Trunk		An area for stowing luggage.
	status	Status of “Trunk”.
Door		Entrance to a vehicle.
	status	Status of “Door”.
OnboardInformation Equipment		This term refers to devices that provide information to the driver, such as a navigation system and other equipment.
ExteriorLights		Vehicle exterior lights.
	status	Status of parking light, low beam, high beam, footlights, automatic light control and turn-hazard signal.

A.7 SurroundingPackage

Figure A.4 shows the “SurroundingPackage” and Table A.3 gives its class descriptions.

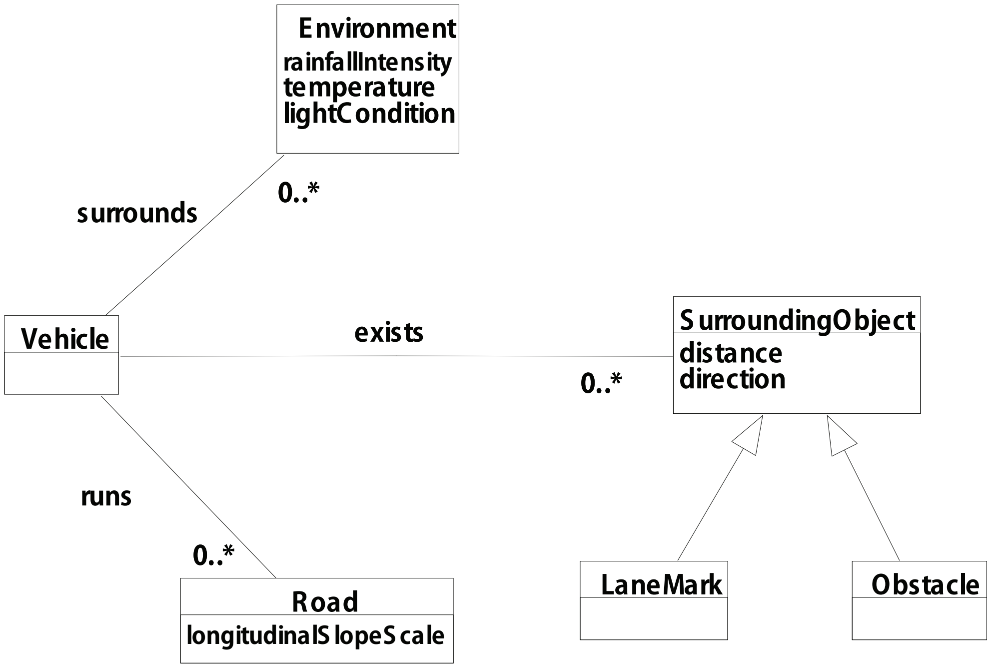

Figure A.4 — SurroundingPackage

“SurroundingPackage” defines data element concepts that correspond to properties of the road, the driving environment and other objects physically surrounding the vehicle.

These surrounding physical entities are represented as “Environment”, “Road”, and “SurroundingObject”. They can be represented as subclasses of these classes and parts of these classes.

Table A.3 — Class descriptions for SurroundingPackage

Class	Property	Description
Vehicle		See “ProbeCorePackage”.
Environment		The circumstances or conditions around the “Vehicle”.
	rainfallIntensity	Rate at which rainfall occurs around the “Vehicle”.
	temperature	Ambient air temperature around the “Vehicle”.
	lightCondition	Degree of lightness around the “Vehicle”.
Road		Surfaces on which the “Vehicle” runs.
	longitudinalSlopeScale	Degrees relative to the horizontal.
SurroundingObject		Object which is in the vicinity of the “Vehicle”.
	distance	Distance between the “Vehicle” and the object.
	direction	Direction of the object relative to the Vehicle.
Obstacle		A nearby Object in the Vehicle’s way.
	detected	The fact that the “Obstacle” is sensed, which indicates the presence of “Obstacle(s)”. (This property is inherited from “Object”.)
	distance	“Obstacle” distance from “Vehicle”. (This property is inherited from “SurroundingObject”.)
	direction	The direction of the “Obstacle” as measured from the “Vehicle”. (This property is inherited from “SurroundingObject”.)
LaneMark		Lane marking on the left or right side of the lane in which the “Vehicle” is travelling.
	detected	The fact that the “Lanemark” is sensed, which indicates the presence of “LaneMark”. (This property is inherited from “Object”.)

A.8 RoadNetworkPackage

Figure A.5 shows the “RoadNetworkPackage” and Table A.4 gives its class descriptions.

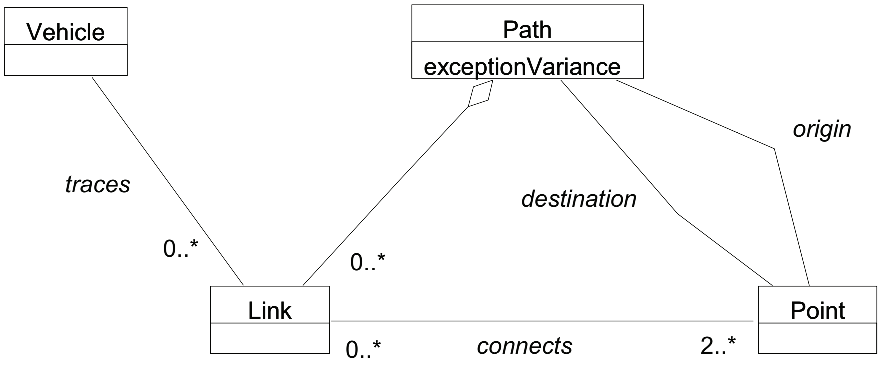

Figure A.5 — RoadNetworkPackage

“RoadNetworkPackage” defines data element concepts that correspond to properties of the road network.

This package represents the logical network of road structures that is represented as a set of “Points” and “Links”.

Table A.4 — Class descriptions for RoadNetworkPackage

Class	Property	Description
Vehicle		See “ProbeCorePackage”
Path		The connection between origin and destination. May consist of multiple “Links”.
	exceptionVariance	Vehicle path on road is different from path indicated in map database
Link		The connection between two “Points”.
Point		Point on the road, especially at intersections.

Annex B
(normative)

Core data elements in XML format

The following is the XML definition of core data elements.

```
<?xml version="1.0" encoding="UTF-8"?>

<probe_data_dictionary>

    <probe_data_element descriptive_name="Sensing.timestamp:real">

        <ASN.1_name>Sensing-timestamp</ASN.1_name>

        <ASN.1_object_identifier>{ 1 0 22837 000 000 }</ASN.1_object_identifier>

        <definition>Based on January 1, 1970 (it is same as "UNIX epoch time").</definition>

        <descriptive_name_context>probe</descriptive_name_context>

        <data_concept_type>data element</data_concept_type>

        <standard>EMPTY</standard>

        <data_type>REAL</data_type>

        <format>REAL</format>

        <unit_of_measure>sec</unit_of_measure>

        <valid_value_rule>real</valid_value_rule>

        <data_quality>n.a.</data_quality>

    </probe_data_element>

    <probe_data_element descriptive_name="Sensing.latitude:lctn-in-degree-with-confidence">

        <ASN.1_name>Sensing-latitude</ASN.1_name>

        <ASN.1_object_identifier>{ 1 0 22837 000 001 }</ASN.1_object_identifier>

        <definition>Latitude of the vehicle observed by the location position sensor, and accuracy of a
sensor.</definition>

        <descriptive_name_context>probe</descriptive_name_context>

        <data_concept_type>data element</data_concept_type>

        <standard>EMPTY</standard>

        <data_type>LocationDegreeWithConfidence ::= SEQUENCE {

    degree REAL,
```

```
confidence REAL }</data_type>
    <format>latitude [-90...90]. Format of confidence is expressed in REAL..</format>
    <unit_of_measure>degree, millimetre</unit_of_measure>
    <valid_value_rule>real [-90...90]. confidence is any real number.</valid_value_rule>
    <data_quality>n.a.</data_quality>
</probe_data_element>
<probe_data_element descriptive_name="Sensing.longitude:lctn-in-degree-with-confidence">
    <ASN.1_name>Sensing-longitude</ASN.1_name>
    <ASN.1_object_identifier>{ 1 0 22837 000 002 }</ASN.1_object_identifier>
    <definition>Longitude of the vehicle observed by the sensor, and accuracy of a sensor.</definition>
    <descriptive_name_context>probe</descriptive_name_context>
    <data_concept_type>data element</data_concept_type>
    <standard>EMPTY</standard>
    <data_type>LocationDegreeWithConfidence ::= SEQUENCE {
degree REAL,
confidence REAL }</data_type>
    <format>longitude [-180...180]. Format of confidence is expressed in REAL.</format>
    <unit_of_measure>degree, millimetre</unit_of_measure>
    <valid_value_rule>real [-180...180]. confidence is any real number.</valid_value_rule>
    <data_quality>n.a.</data_quality>
</probe_data_element>
<probe_data_element descriptive_name="Sensing.altitude:lctn-in-altitude-with-confidence">
    <ASN.1_name>Sensing-altitude</ASN.1_name>
    <ASN.1_object_identifier>{ 1 0 22837 000 003 }</ASN.1_object_identifier>
    <definition>Altitude of the vehicle observed by the sensor, and accuracy of a sensor.</definition>
    <descriptive_name_context>probe</descriptive_name_context>
    <data_concept_type>data element</data_concept_type>
    <standard>EMPTY</standard>
    <data_type>LocationAltitudeWithConfidence ::= SEQUENCE {
altitude INTEGER,
```

confidence REAL }</data_type>

<format>INTEGER. Format of confidence is expressed in REAL.</format>

<unit_of_measure>metre, metre</unit_of_measure>

<valid_value_rule>integer [-65535...65535]. confidence is any real number.</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

</probe_data_dictionary>

Annex C
(normative)

Data elements in XML format

The following is the XML definition of data elements.

<probe_data_element descriptive_name="Environment.temperature:qty-degrees-Celsius-with-confidence">

<ASN.1_name>Environment-temperature</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 010 }</ASN.1_object_identifier>

<definition>ambient air temperature with confidence</definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>QtyDegreesCelsiusWithConfidence :: = SEQUENCE {

degrees INTEGER,

confidence INTEGER }</data_type>

<format>air temperature. Format of confidence is expressed in degrees Celsius.</format>

<unit_of_measure>degree Celsius, degree Celsius.</unit_of_measure>

<valid_value_rule>integer[-49...50, FFFF = unknown], integer[0...20]</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="Wiper.status:integer">

<ASN.1_name>Wiper-status</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 035 }</ASN.1_object_identifier>

<definition>activation of wipers and operational mode (intermittent, slow, fast)</definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>INTEGER</data_type>

<format>0 = no wiper active, 1 = intermittent, 2 = slow, 3 = fast</format>

<unit_of_measure>concatenated code</unit_of_measure>

<valid_value_rule>integer [0...3]</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="Environment.rainfallIntensity:integer">

<ASN.1_name>Environment-rainfallIntensity</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 009 }</ASN.1_object_identifier>

<definition>rate at which rainfall occurs, expressed in units of depth per unit of time.</definition>

<standard>EMPTY</standard>

<data_type>INTEGER</data_type>

<format>Rain fall intensity is expressed in millimetres/hour</format>

<unit_of_measure>millimetre per hour</unit_of_measure>

<valid_value_rule>Integer [0…999]</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element> <probe_data_element descriptive_name="ExteriorLights.status:code-exterior-light-status">

<ASN.1_name>ExteriorLights-status</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 011 }</ASN.1_object_identifier>

<definition>Status of all vehicle exterior lights</definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>CodeExteriorLightStatus ::= SEQUENCE {

parkinglight INTEGER,

lowbeam INTEGER,

highbeam INTEGER,

foglights INTEGER,

automaticlightcontrol INTEGER,

turnhazardsignal INTEGER }</data_type>

<format>For parkinglight, lowbeam, highbeam, foglights, automaticlightcontrol: 0=OFF, 1=ON; For turnhazardsignal: 0=off,1=left, 2=right, 3=hazard</format>

<unit_of_measure>code</unit_of_measure>

<valid_value_rule>integer[0...1], integer[0...1], integer[0...1], integer[0...1], integer[0...1], integer[0...3] </valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="Environment.light Condition:integer”>

<ASN.1_name>Environment-lightCondition</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 008 }</ASN.1_object_identifier>

<definition>outside light conditions: degree of light in environment (bright to dark)</definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>INTEGER</data_type>

<format>Segmentation of direct measurement of light energy (lux); 0 = 0 to 1 lux, 1 = 2 to 100 lux, 2 = 101 to 1 000 lux, 3 = 1 001 to 30 000 lux, 4 = 30 001 to 50 000 lux, 5 = 50 001 to 80 000 lux, 6 = 80 001 to 100 000 lux, 7 = over 100 000 lux</format>

<unit_of_measure>code</unit_of_measure>

<valid_value_rule>integer [0…7]</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="Vehicle.velocity:rt-velocity-with-confidence">

<ASN.1_name>Vehicle-velocity</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 032 }</ASN.1_object_identifier>

<definition>vehicle velocity with confidence data</definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>RateVelocityWithConfidence :: = SEQUENCE {

velocity INTEGER,

confidence INTEGER }</data_type>

<format>current vehicle speed. Format of confidence is expressed in meters/second.</format>

```
        <unit_of_measure>metre per second, metre per second</unit_of_measure>
        <valid_value_rule>integer[0...99], integer[0...100]</valid_value_rule>
        <data_quality>n.a.</data_quality>
    </probe_data_element>
    <probe_data_element descriptive_name="Obstacle.detected:boolean">
        <ASN.1_name>Obstacle-detected</ASN.1_name>
        <ASN.1_object_identifier>{ 1 0 22837 000 015 }</ASN.1_object_identifier>
        <definition>obstacle presence</definition>
        <data_concept_type>data element</data_concept_type>
        <standard>EMPTY</standard>
        <data_type>BOOLEAN</data_type>
        <format>obstacle in current lane of vehicle at or near current position; 1 = obstacle present</format>
        <unit_of_measure>code</unit_of_measure>
        <valid_value_rule>0 or 1</valid_value_rule>
        <data_quality>n.a.</data_quality>
    </probe_data_element>
    <probe_data_element descriptive_name="Obstacle.distance:integer">
        <ASN.1_name>Obstacle-distance</ASN.1_name>
        <ASN.1_object_identifier>{ 1 0 22837 000 017 }</ASN.1_object_identifier>
        <definition>obstacle distance from vehicle detecting and reporting the obstacle</definition>
        <descriptive_name_context>probe</descriptive_name_context>
        <data_concept_type>data element</data_concept_type>
        <standard>EMPTY</standard>
        <data_type>INTEGER</data_type>
        <format>distance from reporting vehicle at time of reporting</format>
        <unit_of_measure>decimetre</unit_of_measure>
        <valid_value_rule>integer [0…999]</valid_value_rule>
        <data_quality>n.a.</data_quality>
    </probe_data_element>
    <probe_data_element descriptive_name="Obstacle.direction:integer">
```

```
        <ASN.1_name>Obstacle-direction</ASN.1_name>
        <ASN.1_object_identifier>{ 1 0 22837 000 016 }</ASN.1_object_identifier>
        <definition>obstacle direction from vehicle detecting and reporting the obstacle</definition>
        <data_concept_type>data element</data_concept_type>
        <standard>EMPTY</standard>
        <data_type>INTEGER</data_type>
        <format>azimuth relative to forward direction of vehicle</format>
        <unit_of_measure>degree</unit_of_measure>
        <valid_value_rule>integer [-90… 90]</valid_value_rule>
        <data_quality>n.a.</data_quality>
    </probe_data_element>
    <probe_data_element descriptive_name="AntiLockBrakeSystem.status:boolean">
        <ASN.1_name>AntiLockBrakeSystem-status</ASN.1_name>
        <ASN.1_object_identifier>{ 1 0 22837 000 004 }</ASN.1_object_identifier>
        <definition>activation of anti-lock braking system, which indicates poor road traction</definition>
        <descriptive_name_context>probe</descriptive_name_context>
        <data_concept_type>data element</data_concept_type>
        <standard>EMPTY</standard>
        <data_type>BOOLEAN</data_type>
        <format>1 = ABS activated</format>
        <unit_of_measure>code</unit_of_measure>
        <valid_value_rule>0 or 1</valid_value_rule>
        <data_quality>n.a.</data_quality>
    </probe_data_element>
    <probe_data_element descriptive_name="TractionControlSystem.status:boolean">
        <ASN.1_name>TractionControlSystem-status</ASN.1_name>
        <ASN.1_object_identifier>{ 1 0 22837 000 022 }</ASN.1_object_identifier>
        <definition>activation of traction control system, which indicates poor road traction</definition>
        <descriptive_name_context>probe</descriptive_name_context>
        <data_concept_type>data element</data_concept_type>
```

<standard>EMPTY</standard>

<data_type>BOOLEAN</data_type>

<format>1=TC activated</format>

<unit_of_measure>code</unit_of_measure>

<valid_value_rule>0 or 1</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="VehicleStabilityControl.status:boolean">

<ASN.1_name>VehicleStabilityControl-status</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 034 }</ASN.1_object_identifier>

<definition>activation of Vehicle Stability Control (also called Electronic Stability Program), which is indicative of hazardous conditions</definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>BOOLEAN</data_type>

<format>1=ESP/VSC activated</format>

<unit_of_measure>code</unit_of_measure>

<valid_value_rule>0 or 1</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="Vehicle.GForce:integer">

<ASN.1_name>Vehicle-gForce</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 027 }</ASN.1_object_identifier>

<definition>Vertical G force to indicate a pothole or bumpy road; reported based on exceeding a threshold</definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>INTEGER</data_type>

<format>Vertical G force as measured at wheel. Signed.</format>

<unit_of_measure>tenth of G</unit_of_measure>

<valid_value_rule>integer [-99 … 99]</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="Vehicle.acceleration:rt-acceleration-with-confidence">

<ASN.1_name>Vehicle-acceleration</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 024 }</ASN.1_object_identifier>

<definition>detection of acceleration/deceleration by vehicle, which exceeds a defined threshold, with confidence</definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>RateAccelerationWithConfidence :: = SEQUENCE {

acceleration INTEGER,

confidence INTEGER }</data_type>

<format>detection of acceleration by vehicle, which exceeds a defined threshold. signed. Format of confidence is expressed in cm/s^2</format>

<unit_of_measure> centimetre per second squared, centimetre per second squared </unit_of_measure>

<valid_value_rule>integer[0...3000], integer[0...1000]</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="Brake.status:integer">

<ASN.1_name>Brake-status</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 006 }</ASN.1_object_identifier>

<definition>Brake Activation and Force</definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>INTEGER</data_type>

<format>activation of vehicle brakes either by driver or automatic system, and the degree of braking, expressed as % of full braking force</format>

<unit_of_measure>scale [0…99]</unit_of_measure>

<valid_value_rule>integer [0…99]</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="Brake.boostAssist:integer">

<ASN.1_name>Brake-boostAssist</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 005 }</ASN.1_object_identifier>

<definition>Engagement of brake boost assist function, as an indication of an emergency situation</definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>INTEGER</data_type>

<format>0 = not activated, 1 = activated</format>

<unit_of_measure>code</unit_of_measure>

<valid_value_rule>integer [0…1]</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="Vehicle.yawRate:rt-yaw-rate-with-confidence">

<ASN.1_name>Vehicle-yawRate</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 033 }</ASN.1_object_identifier>

<definition>Rate of change of yaw angle, which can indicate hazardous or emergency situations, with confidence</definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>RateYawRateWithConfidence :: = SEQUENCE {

yaw-rate INTEGER,

confidence INTEGER }</data_type>

<format>Rate of change of yaw angle. Format of confidence is expressed in degrees/second.</format>

<unit_of_measure>degree per second, degree per second</unit_of_measure>

<valid_value_rule>integer[0...359], integer[0...359]</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="FuellingSystem.fuelConsumption:integer">

<ASN.1_name>FuellingSystem-fuelConsumption</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 013 }</ASN.1_object_identifier>

<definition>current fuel consumption</definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>INTEGER</data_type>

<format>fuel consumption rate </format>

<unit_of_measure> millilitre per minute</unit_of_measure>

<valid_value_rule>integer [0…999]</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="FuellingSystem.averageFuelConsumption:integer">

<ASN.1_name>FuellingSystem-averageFuelConsumption</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 012 }</ASN.1_object_identifier>

<definition>average fuel consumption</definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>INTEGER</data_type>

<format>average fuel consumption rate</format>

<unit_of_measure> millilitre per minute</unit_of_measure>

<valid_value_rule>integer [0…999]</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>


```
<probe_data_element descriptive_name="Vehicle.stoppageTime:integer">
    <ASN.1_name>Vehicle-stoppageTime</ASN.1_name>
    <ASN.1_object_identifier>{ 1 0 22837 000 029 }</ASN.1_object_identifier>
    <definition>time during which vehicle is stopped with engine idling</definition>
    <descriptive_name_context>probe</descriptive_name_context>
    <data_concept_type>data element</data_concept_type>
    <standard>EMPTY</standard>
    <data_type>INTEGER</data_type>
    <format>time elapsed</format>
    <unit_of_measure>tens of seconds</unit_of_measure>
    <valid_value_rule>integer [0…999]</valid_value_rule>
    <data_quality>n.a.</data_quality>
</probe_data_element>
<probe_data_element descriptive_name="Vehicle.engineStoppedTime:integer">
    <ASN.1_name>Vehicle-engineStoppedTime</ASN.1_name>
    <ASN.1_object_identifier>{ 1 0 22837 000 026 }</ASN.1_object_identifier>
    <definition>period of time that the engine is stopped</definition>
    <descriptive_name_context>probe</descriptive_name_context>
    <data_concept_type>data element</data_concept_type>
    <standard>EMPTY</standard>
    <data_type>INTEGER</data_type>
    <format>time elapsed</format>
    <unit_of_measure>minute</unit_of_measure>
    <valid_value_rule>integer [0…999]</valid_value_rule>
    <data_quality>n.a.</data_quality>
</probe_data_element>
<probe_data_element descriptive_name="Vehicle.lateralAcceleration:rt-acceleration-with-confidence">
    <ASN.1_name>Vehicle-lateralAcceleration</ASN.1_name>
    <ASN.1_object_identifier>{ 1 0 22837 000 028 }</ASN.1_object_identifier>
```

<definition>lateral acceleration experienced by vehicle, which can indicate an emergency manoeuvre, with confidence</definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>RateAccelarationWithConfidence :: = SEQUENCE{

lateralAccelaration INTEGER,

confidence INTEGER }</data_type>

<format>Rate of change of lateral acceleration. Format of confidence is expressed in cm/s^2</format>

<unit_of_measure> centimetre per second squared, centimetre per second squared </unit_of_measure>

<valid_value_rule>integer[0... 3000], integer[0...1000]</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="Path.exception Variance:integer">

<ASN.1_name>Path-exceptionVariance</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 019 }</ASN.1_object_identifier>

<definition>vehicle path on road is different from path indicated in map database, indicating map database needs to be updated. This data element is only a flag that this situation is detected.</definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>INTEGER</data_type>

<format>Once detected, vehicle reports variance every 20 m or once per second, whichever is greater. 1 = path exception detected</format>

<unit_of_measure>code</unit_of_measure>

<valid_value_rule>integer[0…1]</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="Vehicle.direction: qty-direction-with-confidence">

<ASN.1_name>Vehicle-direction</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 025 }</ASN.1_object_identifier>

<definition>current direction of vehicle travel. This data element supports other data elements such as obstacle detection</definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>QtyDirectionWithConfidence :: = SEQUENCE {

direction INTEGER,

confidence INTEGER }</data_type>

<format>degrees relative to North, format of confidence is expressed in tenths of degrees.</format>

<unit_of_measure>tenth of degree, tenth of degree</unit_of_measure>

<valid_value_rule>Integer [0…3600], Integer [0…1000]</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="Vehicle.vehicleType:integer">

<ASN.1_name>Vehicle-vehicleType</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 031 }</ASN.1_object_identifier>

<definition>Type of vehicle </definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>INTEGER</data_type>

<format>0=Unknown, 1=passenger car, 2=light truck, 3=heavy truck (&gt 5 000 kg) (axis count unspecified), 4=bus, 5=motorcycle, 6=articulated truck, 7=car with trailer, 8=truck with trailer (axis count unspecified), 9=high sided vehicle, 10=heavy truck (axis count =2), 11=heavy truck (axis count =3), 12=heavy truck (axis count =4), 13=heavy truck (axis count =5), 14=heavy truck (axis count =6), 15=heavy truck (axis count =7), 16=truck with trailer (axis count =2), 17=truck with trailer (axis count =3), 18=truck with trailer (axis count =4), 19=truck with trailer (axis count =5), 20=truck with trailer (axis count =6), 21=truck with trailer (axis count =7), 22 to 255 = local definition</format>

<unit_of_measure>code</unit_of_measure>

<valid_value_rule>integer [0…255]</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="Vehicle.vehicleUsage:integer">

<ASN.1_name>Vehicle-vehicleUsage</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 036 }</ASN.1_object_identifier>

<definition>Usage of vehicle </definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>INTEGER</data_type>

<format>0=Unknown, 1=private use, 2=taxi, 3=commercial, 4=public transport, 5=emergency services, 6=patrol services, 7=road operator, 8=snow plough, 9=hazmat (hazardous material), 10=other, 11 to 255 = local definition</format>

<unit_of_measure>code</unit_of_measure>

<valid_value_rule>integer [0…255]</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="Vehicle.suddenSteeringManoeuvre:integer">

<ASN.1_name>Vehicle-suddenSteeringManoeuvre</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 030 }</ASN.1_object_identifier>

<definition>Detection of emergency steering manoeuvre made by driver, based on angular rotation rate of steering wheel, which exceeds a defined threshold; or based on lateral acceleration</definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>INTEGER</data_type>

<format>angular rotation rate of steering wheel</format>

<unit_of_measure>degree per second</unit_of_measure>

<valid_value_rule>integer [0…359]</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="LaneMark.detected:integer">

<ASN.1_name>LaneMark-detected</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 014 }</ASN.1_object_identifier>

<definition>Status of lane marking detection</definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>INTEGER</data_type>

<format>0 = lane marking not detected, 1 = lane marking detected</format>

<unit_of_measure>code</unit_of_measure>

<valid_value_rule>integer [0...1]</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="Road. longitudinal SlopeScale:integer">

<ASN.1_name>Road-longitudinalSlopeScale</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 020 }</ASN.1_object_identifier>

<definition>Current longitudinal slope scale of vehicle travel</definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>INTEGER,</data_type>

<format>Degrees relative to the horizontal</format>

<unit_of_measure>tenth of degree</unit_of_measure>

<valid_value_rule>integer [-899…900]</valid_value_rule>

<data_quality>n. a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="Seatbelt.status: code-seatbelt-status">

<ASN.1_name>Seatbelt-status</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 021 }</ASN.1_object_identifier>

<definition>Seat-belt status reported consistent with AMI-C message set. Addresses all possible seat locations. This data element may be used for data cleansing and would not typically be aggregated.</definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>CodeSeatbeltStatus ::= SEQUENCE {

driver INTEGER, middlefront INTEGER, passenger INTEGER,

secondrowleft INTEGER, secondrowmiddle INTEGER, secondrowright INTEGER,

thirdrowleft INTEGER, thirdrowmiddle INTEGER, thirdrowright INTEGER,

fourthrowleft INTEGER, fourthrowmiddle INTEGER, fourthrowright INTEGER,

fifthrowleft INTEGER, fifthrowmiddle INTEGER, fifthrowright INTEGER }</data_type>

<format>For each seat, 0=not equipped, 1=non-fastened, 2=fastened</format>

<unit_of_measure>concatenated code</unit_of_measure>

<valid_value_rule>integer[0…2], integer[0…2], integer[0…2], integer[0…2], integer[0…2], integer[0…2], integer[0…2], integer[0…2], integer[0…2], integer[0…2], integer[0…2], integer[0…2], integer[0…2], integer[0…2], integer[0…2]</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="Door.status:boolean">

<ASN.1_name>Door-status</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 007 }</ASN.1_object_identifier>

<definition>indication that one or more vehicle doors is open. This data element may be used for data cleansing and would not typically be aggregated.</definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>BOOLEAN</data_type>

<format>data expresses "door open warning" is on or off: 0=off, 1=on</format>

<unit_of_measure>code</unit_of_measure>

<valid_value_rule>0 or 1</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="Trunk.status:boolean">

<ASN.1_name>Trunk-status</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 023 }</ASN.1_object_identifier>

<definition>vehicle trunk is open. This data element may be used for data cleansing and would not typically be aggregated.</definition>

<descriptive_name_context>probe</descriptive_name_context>

<standard>EMPTY</standard>

<data_type>BOOLEAN</data_type>

<format>0=closed, 1=open</format>

<unit_of_measure>code</unit_of_measure>

<valid_value_rule>0 or 1</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

<probe_data_element descriptive_name="ParkingBrake.status:boolean">

<ASN.1_name>ParkingBrake-status</ASN.1_name>

<ASN.1_object_identifier>{ 1 0 22837 000 018 }</ASN.1_object_identifier>

<definition>parking brake engaged. This data element may be used for data cleansing and would not typically be aggregated.</definition>

<descriptive_name_context>probe</descriptive_name_context>

<data_concept_type>data element</data_concept_type>

<standard>EMPTY</standard>

<data_type>BOOLEAN</data_type>

<format>0=off, 1=on</format>

<unit_of_measure>code</unit_of_measure>

<valid_value_rule>0 or 1</valid_value_rule>

<data_quality>n.a.</data_quality>

</probe_data_element>

Annex D
(informative)

Examples of probe messages

D.1 General

This annex lists typical examples of probe messages.

D.2 Simple element message

Any probe data element may be sent as a simple element message.

EXAMPLE A probe message that carries vehicle velocity consists of the following data elements.

Sensing. timestamp:real	Sensing.latitude: lctn-in-degree-with-confidence	Sensing. longitude:lctn-in-degree-with-confidence	Sensing.altitude: lctn-in-altitude-with-confidence	Vehicle.velocity: rt-velocity-with-confidence

core data elements

Figure D.1 — Example of simple element message

D.3 Multiple element message

Some combinations of probe data elements can be useful for specific probe processing applications. The following are typical messages that convey combinations of probe data elements.

Table D.1 — Data elements for multiple element messages

Application domain	Data elements (excluding core data elements)
Traffic	Vehicle.velocity:rt-velocity-with-confidence Vehicle.direction:qty-direction-with-confidence
Weather	Wiper.status:integer Environment.rainfallIntensity:integer Environment.temperature:qty-degrees-Celsius-with-confidence
	ExteriorLights.status:code-exterior-light-status Environment.temperature:qty-degrees-Celsius-with-confidence
Vehicle Operation	TractionControlSystem.status boolean Environment.temperature:qty-degrees-Celsius-with-confidence
Safety	Obstacle.detected:boolean Obstacle.distance:integer Obstacle.direction:integer Vehicle.velocity:rt-velocity-with-confidence Vehicle.direction:qty-direction-with-confidence

EXAMPLE A probe message for the traffic application domain listed in Table D.1 consists of the following data elements.

Sensing.timestamp:real	Sensing. latitude:lctn-in-degree-with-confidence	Sensing. longitude:lctn-in-degree-with-confidence	Sensing.altitude: lctn-in-altitude-with-confidence	Vehicle.velocity: rt-velocity-with-confidence	Vehicle.direction: qty-direction-with-confidence

core data elements

Figure D.2 — Example of multiple element message

Annex E
(informative)

Probe processing context model

E.1 General

This annex provides an informal description of the activities of probe processing, as well as an informal description of the kinds of activities that are expected to be going on alongside probe processing. In brief, probe processing builds an aggregate picture of the driving environment based on sensor readings from many vehicles and data from a variety of other sources. An important characteristic of probe processing is that data from vehicles has all identification of the vehicle and its occupants removed prior to processing.

E.2 Probe processing model

Probe data begins as raw sensor data from onboard sensors. Items sensed can include: vehicle location, vehicle speed, environmental conditions such as air temperature, rainfall intensity, light level and pavement conditions (how rough, how slippery, etc.). This data is gathered by an onboard data collection system which may pre-process the sensor data and which formats it for transmission.

Probe messages are usually sent only by exception, i.e. if the vehicle sensors detect a situation that is different from the currently known situation. This helps to greatly minimize communications traffic, probably the most expensive aspect of probe processing. In this model, the vehicle maintains an onboard database that contains the currently known information about the driving environment. If sensor readings are consistent with this information, no message is sent. If the sensor readings are significantly different from this information, a probe message will be sent. If land-side processing centres advise the vehicle of an updated situation, this is recorded in the onboard database. Usually this will have the effect of suppressing further probe messages regarding this situation until a further change is detected.

If a message is to be sent, the onboard data collection system delivers the probe message to the onboard communications device for transmission. The transmitted probe message is received by a data message processing operation, which routes the received probe messages to a probe processing centre.

The probe processing centre fuses the data reported from many vehicles along with supplementary data from other data sources. Some of the resulting processed probe data may be sent directly to public agencies to advise them of pavement problems, traffic jams, etc. Processed probe data is also sent to application providers to be further processed, formatted, and delivered to a variety of users. One important set of users are vehicles, who are advised of new situations in the driving environment in their vicinity via the data message processing operation. Other application results may be similarly delivered to the vehicle, for use by onboard applications. The probe processing centre can also send requests to vehicles in an area to adjust the way that probe data is sent (e.g. asking for more frequent reports about the driving environment in a particular area).

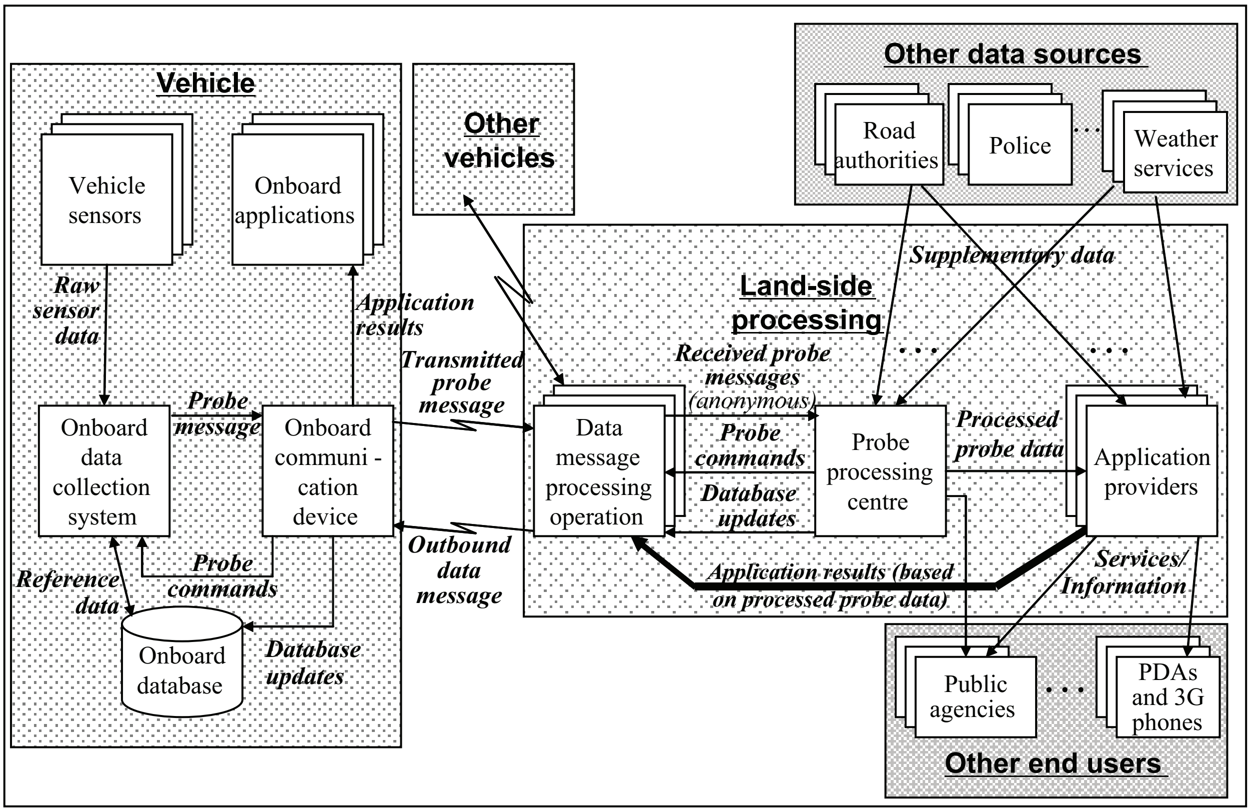

Figure E.1 — Probe processing model

E.3 Probe processing context model

E.3.1 General

Many other activities may be going on alongside probe processing. Some of these are shown in Figure E.2. The most important point is that many other kinds of messages, besides probe-related messages, can be moving between the vehicle and the land side. Many of these messages will necessarily identify vehicles and their occupants (for example, automatic crash notification messages, requests for location-based services, or other telematics services which require a fee to be paid).

Figure E.2 illustrates that the onboard data-collection system can assemble other kinds of messages besides probe messages, which will also be sent by the onboard communications device to the data message processing operation and routed to various destinations. This includes diagnostic information that is of interest to vehicle manufacturers, and requests and data related to a variety of other vehicle-oriented and user-oriented services.

Each of the entities and major data flows in the probe processing context model are defined in greater detail below. As applicable, parallels are drawn between items in this informative Context Model and corresponding items in the normative reference architecture that appears in Clause 5 of this International Standard.

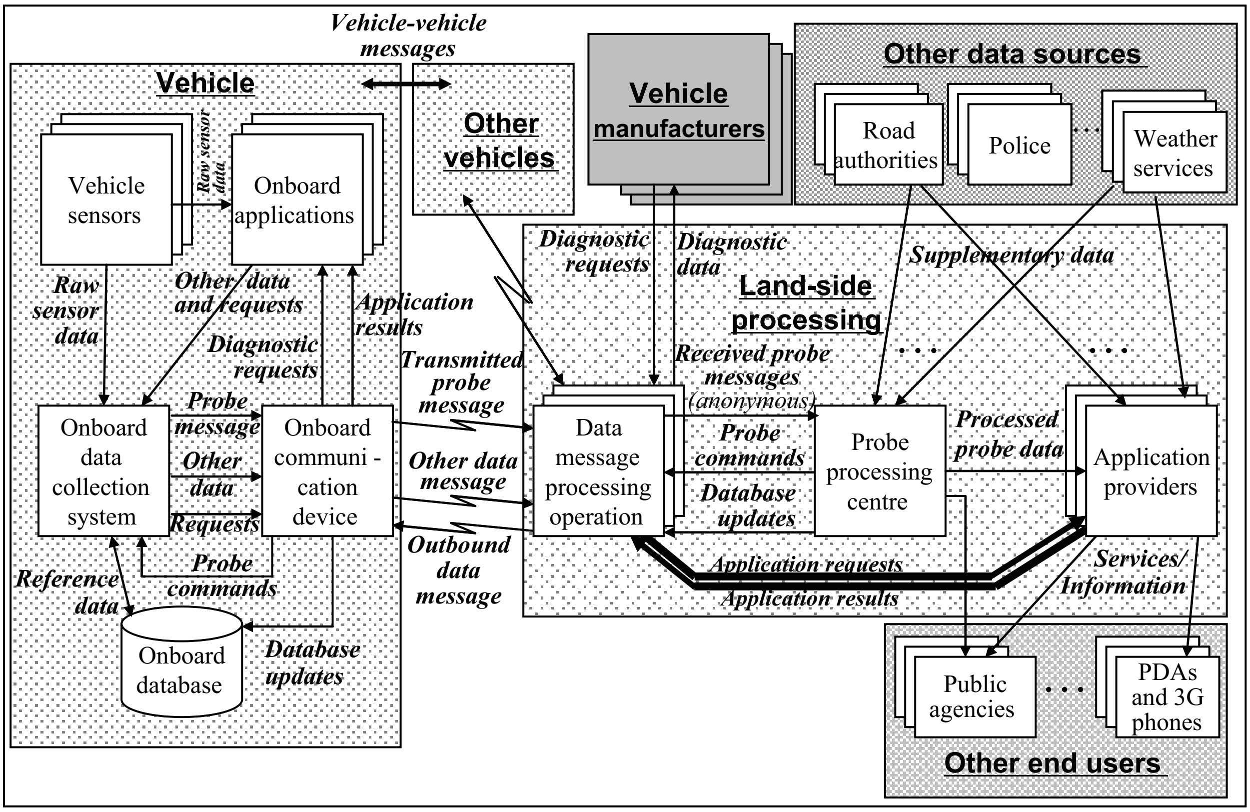

Figure E.2 — Probe processing context model

E.3.2 Entities

- **Vehicle sensors**

  — Devices on a vehicle that sense data about conditions inside and/or outside of the vehicle, as well as actions that the driver takes, such as turning on/off headlights or windshield wipers, or suddenly applying the brakes.

  — *This corresponds to "Onboard data source" and "Probe data element generation" in the reference architecture.*

- **Onboard application**

  — Any of several systems in the vehicle that can produce or request data, for example onboard diagnostics, the telematics hardware platform. Some onboard applications will help the driver or vehicle perform better or more safely by using results from external applications that use probe data.

- **Onboard data collection system**

  — An onboard system that accepts (1) data from **vehicle sensors** and (2) **other data** and **requests** from **onboard applications**. The onboard data collection system processes the data it receives. In some cases, the onboard data collection system compares and combines this data with data from an **onboard database.** The onboard data collection system formats the received and processed data for transmission to a central point.

  — The **onboard data collection system** also receives **probe commands** that originated from the **probe processing centre** to guide future data collection. For example, a **probe command** could instruct the **onboard data collection system** to start sending **probe messages** unconditionally rather than by exception.

  — *This corresponds to "Probe message generation" in the reference architecture.*

- **Onboard database**

  — An organized collection of information about the roadway system to which **onboard applications** and the **onboard data collection system** can refer. For example, the onboard database could contain expected travel speeds for road segments at various times of day. It could also contain real-time or predictive traffic information, information about road works, obstacles, other vehicles currently on the road, etc.

  — One purpose of the database is to allow the **onboard data collection system** to send a **probe message** only when **vehicle sensors** detect a situation different from the information in the **onboard database**.

  — *This corresponds to "Referenced data repository" in the reference architecture.*

- **Onboard communication device**

  — An onboard device that transmits data messages (**transmitted probe message** or **other data message**) to the **data message processing operation** and receives **outbound data messages** from the **data message processing operation**.

  — It also sends:

    — application results to various onboard applications

    — database updates to the onboard database

    — probe commands to the onboard data collection system

- **Other vehicles**

  — This entity appears in the diagram to illustrate the fact that many vehicles will provide data messages (including transmitted probe messages) to various **data message processing operations**.

- **Data message processing operation**

  — An entity which:

    i) Receives inbound data messages (each of which could be a **probe message** or **other data message**) from vehicles and routes them to various destinations depending on the type of message:

      — **Received probe messages** are sent to the **probe processing centre** for fusion and analysis. **Received probe messages** should not contain any information that identifies the vehicle or driver.

      — **Diagnostic data** is sent to **vehicle manufacturers. Diagnostic data** will contain information that identifies the vehicle.

      — **Application requests** of various kinds are sent to the appropriate **application provider**. Many of these messages will contain information that identifies the vehicle or the person making the **application request**.

    ii) Receives:

      — Probe commands and database updates from the probe processing centre

      — Application results from application providers

      — Diagnostic requests from vehicle manufacturers

    and transmits them as **outbound data messages** to the **onboard communications device** in vehicles.

- The **data message processing operation** may be operated by a communications provider or as one service of a telematics provider. It is expected that there will be multiple **data message processing operations**.
- *This corresponds approximately to "Probe collection" in the reference architecture, taking note that in the reference architecture, the only messages of interest are probe messages, while in the context model, many kinds of messages are present.*

- **Vehicle manufacturers**
  - In this diagram, vehicle manufacturers are simply the source of **diagnostic requests** to vehicles and the recipient of **diagnostic data** generated by vehicles. Unlike **probe messages**, messages containing **diagnostic data** will include the identity of the vehicle. These messages do not go to the **probe processing centre**.

- **Probe processing centre**
  - An entity that receives **received probe messages** from **data message processing operations.** It fuses and analyses data from these messages in combination with **supplementary data** from **other data sources** to produce **processed probe data** which it sends to **application providers** and to **public agencies**.
  - The **probe processing centre** also sends **probe commands** and **database updates** back to vehicles to guide future probe data collection.
  - *This corresponds to "Probe processing" in the reference architecture.*

- **Application provider**
  - An entity that transforms **processed probe data** and **supplementary data** into:
    - **Application results** that are sent to vehicles via a **message processing operation**
    - A variety of **services/information** that are delivered to **other end users**
  - *This corresponds to "Probe application" in the reference architecture.*

- **Other data sources**
  - Entities that provide data to the **probe processing centre** and to **application providers** to supplement data from **received probe messages** and **processed probe data**.
  - *This corresponds to "Other data source" in the reference architecture.*

- **Other end users**
  - Recipients of the services/information provided by **application providers.** Other end users include public agencies like road authorities or public safety agencies, and the users of other wireless devices like 3G cellular phones and wireless-enabled PDAs**.**
  - *This corresponds approximately to "User" in the reference architecture, noting that in the reference architecture "User" encompasses both the **vehicle** and **other end users**.*

E.3.3 Data flows

- **Raw sensor data**
  - Data produced by **vehicle sensors** and sent without further processing to the **onboard data collection system** and to **onboard applications**, as appropriate.
  - *This corresponds to "original data" and "probe data element" in the reference architecture.*
- **Other data and requests**
  - The output of various **onboard applications** that is also sent to the **onboard data collection system** for central processing.
- **Probe message**
  - The result of transforming and formatting **raw sensor data** into a form suitable to be delivered to the **onboard communication device** for transmission to a central point.
- **Other data**
  - Data, other than **probe messages**, that is transmitted to a central point for further routing and processing.
- **Requests**
  - Transactions from various **onboard applications** that request information or services from **application providers**.
- **Other data message**
  - All of the different kinds of data messages, other than **transmitted probe messages**, that the vehicle transmits to a central point for further routing and processing, including **other data** and **requests**.
- **Transmitted probe message**
  - The result of packaging and transmitting a **probe message** to the **data message processing operation** for delivery to the **probe processing centre**.
  - *This corresponds to "probe message" in the reference architecture.*
- **Received probe messages**
  - The collection of inbound data messages determined by the **data message processing operation** to contain probe data for delivery to the **probe processing centre**. It is emphasized that a **received probe message** should not contain any information that identifies the vehicle from which it originated or any of the vehicle's occupants.
  - *This corresponds to "collected probe data" in the reference architecture.*
- **Processed probe data**
  - The result of fusing and analysing data from **received probe messages** in combination with **supplementary data**.
  - *This corresponds to "processed probe data" in the reference architecture.*

- **Diagnostic requests**

  — Transactions created by **vehicle manufacturers** to request diagnostic information from vehicles via a **data message processing operation**. Transactions may be directed to all vehicles of a particular type or to a particular vehicle.

- **Diagnostic data**

  — Information about the performance or behaviour of a particular vehicle, directed by a **data message processing centre** to a **vehicle manufacturer**.

- **Application requests**

  — Requests for information or services originating from particular vehicles and directed by a **data message processing centre** to **application providers**.

- **Application results**

  — The information or services produced by **application providers** and directed to a particular vehicle via a **data message processing centre**. Within the vehicle, application results are routed from the **onboard communications device** to corresponding **onboard applications**.

- **Services/Information**

  — The value-added result of combining **processed probe data** with **supplementary data** for delivery to **vehicles** via a **data message processing operation** and to **other end users**.

  — *This corresponds to "service/information" in the reference architecture.*

- **Probe commands and database updates**

  — **Probe commands** and **database updates** are generated by the **probe processing centre,** to be sent to the vehicle from a **data message processing operation** via the **onboard communication device,** to support and guide subsequent probe processing in the vehicle. **Probe commands** and **database updates** are individually described in more detail below.

- **Outbound data message**

  — A message sent by the **data message processing operation** to the vehicle's **onboard communication device**, which could be:

  i) A **probe command** or **database update** transmitted to all the vehicles in a particular geographic area or augmented with information so that vehicles can determine whether a particular message is applicable to them.

  ii) An **application result** targeted to the particular vehicle(s) that requested them.

  iii) **A diagnostic request** from a **vehicle manufacturer**.

- **Database updates**

  — Information used to update the contents and characteristics of the **onboard database** to support and guide subsequent probe processing in the vehicle. This could include new and changed road segments, current or expected traffic conditions, current or expected weather and road condition information, etc.

- **Probe commands**

  — Directives from the **probe processing centre** that guide the collection and sending of **probe messages**. For example, a **probe command** could ask vehicles in a particular geographic area to starting sending **probe messages** unconditionally rather than by exception.

- **Vehicle-vehicle messages**

  — Information sent directly from one vehicle to another, for example to immediately warn nearby vehicles of a dangerous road condition.

- **Supplementary data**

  — Data from **other data sources** (non-vehicle) that is also used in probe processing.

  — *This corresponds to "supplementary data" in the reference architecture.*

- **Reference data**

  — Data from the **onboard database** which describes the latest-known situation. The **onboard data collection system** compares this reference data against current sensor readings to see if a change has occurred that needs to be reported as a **probe message**.

  — *This corresponds to "reference data" in the reference architecture.*

Bibliography

[1] ISO 15662, *Intelligent transport systems — Wide area communication — Protocol management information*

[2] ISO/IEC 19501:2005, *Information technology — Open Distributed Processing — Unified Modeling Language (UML) Version 1.4.2*

[3] ISO/TR 14813-5:1999, *Transport information and control systems — Reference model architecture(s) for the TICS sector — Part 5: Requirements for architecture description in TICS standards*

[4] *GPS Interface Control Document* (ICD 200) <http://www.navcen.uscg.gov/pubs/gps/icd200>

[5] *W3C Recommendation: Extensible Markup Language (XML) 1.0 (Second Edition)*

[6] *OMG Unified Modeling Language (UML) Specification Version 1.5,* March 2003

ICS 03.220.01; 35.240.60

Price based on 60 pages