TECHNICAL SPECIFICATION

ISO/TS 18234-4

First edition
2006-06-01

Traffic and Travel Information (TTI) — TTI via Transport Protocol Expert Group (TPEG) data-streams —

Part 4: Road Traffic Message (RTM) application

Informations sur le trafic et le tourisme (TTI) — Messages TTI via les flux de données du groupe d'experts du protocole de transport (TPEG) —

Partie 4: Application de message de trafic sur route (RTM)

Reference number
ISO/TS 18234-4:2006(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

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

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.

In other circumstances, particularly when there is an urgent market requirement for such documents, a technical committee may decide to publish other types of normative document:

- an ISO Publicly Available Specification (ISO/PAS) represents an agreement between technical experts in an ISO working group and is accepted for publication if it is approved by more than 50 % of the members of the parent committee casting a vote;
- an ISO Technical Specification (ISO/TS) represents an agreement between the members of a technical committee and is accepted for publication if it is approved by 2/3 of the members of the committee casting a vote.

An ISO/PAS or ISO/TS is reviewed after three years in order to decide whether it will be confirmed for a further three years, revised to become an International Standard, or withdrawn. If the ISO/PAS or ISO/TS is confirmed, it is reviewed again after a further three years, at which time it must either be transformed into an International Standard or be withdrawn.

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/TS 18234-4 was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*.

ISO/TS 18234 consists of the following parts, under the general title *Traffic and Travel Information (TTI) — TTI via Transport Protocol Expert Group (TPEG) data-streams:*

- *Part 1: Introduction, numbering and versions*
- *Part 2: Syntax, Semantics and Framing Structure (SSF)*
- *Part 3: Service and Network Information (SNI) application*
- *Part 4: Road Traffic Message (RTM) application*
- *Part 5: Public Transport Information (PTI) application*
- *Part 6: Location referencing applications*

Introduction

TPEG technology uses a byte-oriented stream format, which may be carried on almost any digital bearer with an appropriate adaptation layer. TPEG-messages are delivered from service providers to end-users, and are used to transfer application data from the database of a service provider to a user’s equipment.

This document describes the Road Traffic Message application in detail.

It should be remembered that the TPEG-RTM has been derived from earlier work that resulted in the RDS-TMC standards (EN ISO 14819-2). Upon analysis, RDS-TMC can be seen to drift into other application areas, where it covers a few public transport, parking and weather messages. TPEG-RTM is just one of several applications required to provide a fully comprehensive traffic and travel information service, for example a service is likely to need public transport information, parking information and weather information – these are or will be the subject of other TPEG-application specifications.

Nevertheless, TPEG-RTM, where reasonable, has included the ability to convey similar content to RDS-TMC, in order to offer considerable backwards compatibility and the prospect of automatically generating RDS-TMC messages from TPEG-RTM messages.

The Broadcast Management Committee of the European Broadcast Union (EBU) established the B/TPEG project group in autumn 1997 with the mandate to develop, as soon as possible, a new protocol for broadcasting traffic and travel-related information in the multimedia environment. The TPEG technology, its applications and service features are designed to enable travel-related messages to be coded, decoded, filtered and understood by humans (visually and/or audibly in the user’s language) and by agent systems.

One year later in December 1998, the B/TPEG group produced its first public specifications. Two documents were released. Part 2 (TPEG-SSF, CEN ISO/TS 18234-2) described the Syntax, Semantics and Framing structure, which will be used for all TPEG applications. Part 4 (TPEG-RTM, CEN ISO/TS 18234-4) described the *first* application, for Road Traffic Messages.

CEN/TC 278/WG 4, in conjunction with ISO/TC 204/WG 10, established a project group comprising the members of B/TPEG and they have continued the work concurrently since March 1999. Since then two further parts have been developed to make the initial complete set of four parts, enabling the implementation of a consistent service. Part 3 (TPEG-SNI, CEN ISO/TS 18234-3)) describes the Service and Network Information Application, which is likely to be used by all service implementations to ensure appropriate referencing from one service source to another. Part 1 (TPEG-INV, CEN ISO/TS 18234-1) completed the work, by describing the other parts and their relationships; it also contains the application IDs used within the other parts.

In April 2000, the B/TPEG group released revised Parts 1 to 4, all four parts having been reviewed and updated in the light of initial implementation results. Thus a consistent suite of specifications, ready for wide scale implementation, was submitted to the CEN/ISO commenting process.

In November 2001, after extensive response to the comments received and from many internally suggested improvements, all four parts were completed for the next stage: the Parallel Formal Vote in CEN and ISO. But a major step forward has been to develop the so-called TPEG-Loc location referencing method, which enables both map-based TPEG-decoders and non map-based ones to deliver either map-based location referencing or human readable information. Part 6 (TPEG-Loc, CEN ISO/TS 18234-6) is now a separate specification and is used in association with the other parts of CEN ISO/TS 18234 to provide comprehensive location referencing. Additionally Part 5, the Public Transport Information Application (TPEG-PTI, CEN ISO/TS 18234-5), has been developed and been through the commenting process.

This Technical Specification, CEN ISO/TS 18234-4, provides a full specification provides a full specification for the Road Traffic Message application.

During the development of the TPEG technology a number of versions have been documented and various trials implemented using various versions of the specifications. At the time of the publication of this Technical Specification, all parts are fully inter-workable and no specific dependencies exist. This Technical Specification has the technical version number TPEG-RTM_3.0/003.

Traffic and Travel Information (TTI) — TTI via Transport Protocol Expert Group (TPEG) data-streams —

Part 4: Road Traffic Message (RTM) application

1 Scope

This document establishes the method of delivering Road Traffic Messages within a TPEG service. The TPEG-RTM application is designed to allow the efficient and language independent delivery of road information directly from service provider to end-users. The information provided relates to event and some status information on the road network and on associated infrastructure affecting a road journey. For example, limited information about abnormal operation of links in the network may be included, such as ferries, lifting-bridges, etc.

The term “application” is used in TPEG specifications to describe specific applications, such as in this case the Road Traffic Message application, which comprises three information containers: the Message Management Container, the Application Event Container and the TPEG-Location Container. The first two Containers are fully described herein and the TPEG-Location Container is described in CEN ISO/TS 18234-6.

Each TPEG application (e.g. TPEG-RTM) is assigned a unique number that is called the application identification (AID). An AID is defined whenever a new application is developed. The AID is used within the TPEG-Service and Network Information Application (CEN ISO/TS 18234-3) to indicate how to process TPEG content and allows routing of data to an appropriate application decoder.

AID = 0001 is assigned to the TPEG-Road Traffic Message application, described in this specification.

A hierarchical methodology has been developed to allow the creation of messages from a set of TPEG-RTM tables, which are essentially word oriented and cover most needs. Many of the TTI descriptive words, in the TPEG-RTM tables, were obtained from the DATEX dictionary (ENV 13106), which embodies European TTI knowledge of the last ten years or more, including a deconstruct of the phrase oriented RDS-TMC events list (EN ISO 14819-2). These TPEG-RTM tables (essentially word oriented data object dictionaries) comprise a wide ranging ability to describe a TTI event and some status information, introducing new precision in a number of areas such as “vehicle types”, “positional information on the carriageway” and “diversion routing advice”.

NOTE Explicit backwards compatibility with the RDS-TMC events list (EN ISO 14819-2) could not be achieved since some “update classes”, such as “29 Reference to Audio Broadcasts” and “30 Service Messages”, fall outside the TPEG-RTM remit.

2 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/IEC 7498-1, *Information technology — Open Systems Interconnection — Basic Reference Model: The Basic Model*

ISO 8601, *Data elements and interchange formats — Information interchange — Representation of dates and times*

ISO/TS 18234-1, *Traffic and Travel Information (TTI) — TTI via Transport Protocol Expert Group (TPEG) data-streams — Part 1: Introduction, Numbering and Versions*

ISO/TS 18234-2, *Traffic and Travel Information (TTI) — TTI via Transport Protocol Expert Group (TPEG) data-streams — Part 2: Syntax, Semantics and Framing Structure (SSF)*

ISO/TS 18234-3, *Traffic and Travel Information (TTI) — TTI via Transport Protocol Expert Group (TPEG) data-streams — Part 3: Service and Network Information (SNI) Application*

ISO/TS 18234-6, *Traffic and Travel Information (TTI) — TTI via Transport Protocol Expert Group (TPEG) data-streams — Part 6: Location Referencing application*

3 Terms and definitions

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

NOTE Definitions in this specification are in some cases derived from definitions found in the DATEX Data Dictionary (ENV 13106). TPEG-RTM is completely focussed on delivering messages to end-users, so for this key operational reason some definitions have a different meaning from that found in the DATEX Data Dictionary. These differences are highlighted in this section.

3.1
cross reference information (CRI)
pointer to one or more messages in the same, or another TPEG service

3.2
event description (EVE)
part of a message describing the event, unplanned or planned occurrence affecting the road or transport network, (for example the transport network in the case of a ferry carrying vehicles between parts of the road network) or status information, including qualifiers and quantifiers

NOTE This definition varies from the DATEX Data Dictionary definition (ENV 13106).

3.3
locations
see CEN ISO/TS 18234-6 for full details of the location referencing container explanations

3.4
location referencing
method for referencing locations to facilitate the exchange of location related information between different systems

3.5
message
collection of coherent information sent through an information channel. Describes an event or a collection of related events, or status information and including message management information. The latter is contained in the message header

3.6
message expiry time (MET)
date and time in accordance with EN ISO 8601 when the message should be deleted from all TPEG-decoders (used for message management purposes)

3.7
message generation time (MGT)
date and time stamp in accordance with EN ISO 8601 originated at the actual time and point of message generation (used for message management purposes)

3.8
message identifier (MID)
unique identifier for a sequence of versions of one message relating to a particular event of a particular service component

3.9
position
defines where an event has taken place in relation to the road: driving lane 1, hard shoulder, central reservation, etc. The driving lanes are numbered according to the usual local practice, i.e. driving lane 1 is the lane nearest to the hard shoulder. In countries which drive on the left, driving lanes are hence numbered from left-to-right, and in countries driving on the right, from right-to-left

3.10
severity factor (SEV)
Amount of disruption to traffic likely to be caused by a particular event

NOTE This definition varies from the DATEX Data Dictionary definition (ENV 13106).

3.11
start time (STA)
date and time in accordance with EN ISO 8601 at which an event, or status information, began or is scheduled to begin (used for presentation to the end-user)

NOTE This definition varies from the DATEX Data Dictionary definition (ENV 13106).

3.12
status
characteristic of an element of the transport system for which at all times a value can be established. Status relates to an information stream. Values can be normal or deviating from normal

3.13
stop time (STO)
date and time in accordance with EN ISO 8601 at which an event, or status information, ended or is scheduled to end (used for presentation to the end-user)

NOTE This definition varies from the DATEX Data Dictionary definition (ENV 13106).

3.14
time schedule information (TSI)
gives information about the time schedule for repetitive events within the start and stop time

3.15
unverified information (UNV)
indicates that a message includes information from an unverified source

3.16
version number (VER)
serial number to distinguish successive messages having a particular message identifier. Version numbers are used incrementally, allowing the progress of an event to be tracked from first notification (VER = 0), through updates, to eventual cancellation (VER = 255)

NOTE This definition varies from the DATEX Data Dictionary definition (ENV 13106).

4 Symbols and abbreviations

For the purposes of this Technical Specification, the following abbreviations apply.

4.1
AID
Application Identification

4.2
BPN
Broadcast, Production and Networks (an EBU document publishing number system)

4.3
B/TPEG
Broadcast/TPEG (the EBU project group name for the specification drafting group)

4.4
CEN
Comité Européen de Normalisation

4.5
CRI
Cross Reference Information (see 3.1)

4.6
DAB
Digital Audio Broadcasting

4.7
DVB
Digital Video Broadcasting

4.8
EBU
European Broadcasting Union

4.9
ETSI
European Telecommunications Standards Institute

4.10
EVE
Event Description (see 3.2)

4.11
ILOC
Intersection location

4.12
INV
Introduction, Numbering and Versions (see CEN ISO/TS 18234-1)

4.13
IPR
Intellectual Property Right(s)

4.14
ISO
International Organization for Standardization

4.15
MET
Message Expiry Time (see 3.6)

4.16
MGT
Message Generation Time (see 3.7)

4.17
MID
Message Identifier (see 3.8)

4.18
OSI
Open Systems Interconnection

4.19
PTI
Public Transport Information (see CEN ISO/TS 18234-5)

4.20
RDS-TMC
Radio Data System – Traffic Message Channel

4.21
RFU
Reserved for future use (not necessarily abbreviated)

4.22
RTM
Road Traffic Message application (this specification)

4.23
SEV
Severity Factor (see 3.10)

4.24
SNI
Service and Network Information application (see CEN ISO/TS 18234-3)

4.25
SSF
Syntax, Symantics and Framing Structure (see CEN ISO/TS 18234-2)

4.26
STA
Start Time (See 3.11)

4.27
STO
Stop Time (see 3.13)

4.28
TPEG
Transport Protocol Experts Group

4.29
TSI
Time Schedule Information (see 3.14)

4.30
TTI
Traffic and Travel Information

4.31
UAV
Unassigned value

4.32
UNV
Unverified Information (see 3.15)

4.33
UTC
Coordinated Universal Time

4.34
VER
Version Number (see 3.16)

5 RTM application overview

5.1 Introduction

The TPEG-Road Traffic Message application allows for a wide range of TPEG-decoder types and presentation possibilities to be supported. It may support simultaneously a wide range of TPEG-decoder types, from sophisticated agent TPEG-decoders serving navigation systems, through to simple TPEG-decoders only able to decode 'top-level' information. Some of the possibilities include digital map-based TPEG-decoders, GPS TPEG-decoders without digital map, and in-vehicle, fixed or portable TPEG-decoders without either GPS or digital map. Road traffic messages may be presented to the user in many different ways, including by text, by synthesized speech, graphically, or in route calculation.

Each road traffic message will need to include at least some of the following elements to satisfy the user requirements for information upon which decisions may be based:

- To whom is the information targeted,
- The geographical location to which the information relates,
- The position on the roadway, or adjacent area affected,
- The event being described,
- How severe is the effect of the described message on the journey,
- Whether the information has been verified,
- The time period for which the message remains valid,
- The consequence the message has on expected journey time in the form of delay information,
- Advice on alternative routes,
- Alternative travel options using other modes of transport by cross-references to other applications,
- Further associated information.

Different road users will have disparate needs and interests in various road traffic messages. Different road users may have disparate needs and interests in various road traffic messages. Some may be useful to many users while others may be relevant to only a few users, for example drivers of heavy goods vehicles. Structure in the coding of messages provides for each to be suitably client-filtered. Filtering can be based on many elements, including phrases, attributes, location, times, and the severity of the message on the journey. No additional coding structure is needed.

NOTE Part of each TPEG-message is a location reference. TPEG technology uses one location referencing system across all applications, known as TPEG-Loc (CEN ISO TS 18234-6). This has the potential of enabling messages from different TPEG streams to be linked by their common location. Each message will be about a particular location. The location may be quite specific, a single point on the road network, a road segment between two given points, or it may be a more general area, often with more vague boundaries. The way in which the location is coded is important as it allows information to be filtered by TPEG-decoders and integrated with route planning and navigation systems.

The descriptive phrase and attribute part of the message about an incident allows a user to make a judgement about the likely progress of a journey, and may either directly or indirectly provide advice allowing travel plans to be revised. To allow appropriate decisions to be made, various data about the incident may be required. If for example, an accident occurs, in general the effect the incident causes will change over time. Immediately following an accident, there will be some disruption to traffic flow, the disruption will increase as traffic builds up behind the incident, then begin to lessen as the accident is cleared, and eventually traffic flow will return to normal.

Each incident has a unique reference number (MID), and the changing progress of an incident is tracked by including a VER with each message. The service provider will allocate a new MID and VER = 0 for a new message, subsequent updates to the same event are indicated by allocation of the next higher VER. A MID and version number 255 has the effect of cancelling all earlier versions of the same message.

There are a few particular things to note about MID and VER.

The first is that VER do NOT “wrap around” from version 255 to version 0. In the unlikely event that more than 254 updates to a specific incident is required, service providers must generate a ‘new’ message, using a new MID (and VER = 0), and cancel the earlier message using Version number = 255. A road traffic message uses two mandatory elements: MID and VER=255, which, used in combination, cancels earlier sent messages with the same message ID.

The shortest non-cancellation message contains MID, VER, LOC and EVE; it should be noted that once a location reference is used, one or more corresponding event descriptions must be included.

The second thing to note is that a message identification number, once used, and then cancelled, must not be re-used until as long a time-period as possible has elapsed. Ideally, a service provider should use all 65 535 possible message identification numbers before re-using a previously used MID.

This use of message identification numbers and version numbers will ensure that TPEG-decoders can unambiguously identify the latest versions of each road traffic message, even if messages are received by the TPEG-decoder ‘out of sequence’, when for example an earlier version of a message arrives after a subsequent version, which updates the information that was originally transmitted.

Message identifier and version are the two elements that are mandatory for every message. They are used for message management purposes in the user’s TPEG-decoder, and are not intended for direct display to the user.

All other elements of a message are optional, used when appropriate. These include elements relating to time, the specific or general location to which the message relates, and which particular driving lanes or carriageway are affected. The service provider is also able to make a judgement on the severity of the effect the incident may have upon journey times, and whether an authoritative reporter has verified the information. As a result of a particular message, a user may wish to access more information, perhaps a suggested diversion route, or even to study alternative modes of transport. An easy means of accessing additional information, for example public transport timetables, within a different TPEG application is provided with the cross-referencing information.

5.2 TPEG-message concept

TPEG applications follow an overall concept, which is indicated by the diagrams in this section to give a quick and easily understood human concept, before a more technical description is given.

TPEG event messages may be seen as being built from three different parts, or containers, each with its own clear task: a message management container, an application event container (in this application, the RTM Container) and a location Ccontainer, a shown in Figure 1. (Location referencing details are described in TPEG-Loc (CEN ISO/TS 18234-6).

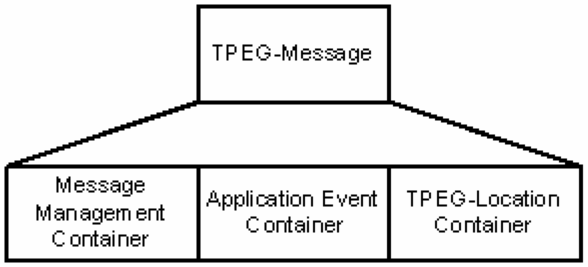

Figure 1 — The three containers

The message management container handles all the elements that allow message tracking, quick identification, validity and other "administrative" tasks. The elements in the application event container are used to describe, with the end-user in mind, the reason for the message, what has happened, and what an end-user may wish to know. The location container describes the location, route or an area for which the event message is applicable.

Regardless of delivery method, it is assumed that a TPEG-decoder will "see" a number of TPEG-messages, one after the other, where they may be messages defined by one or more applications. Figure 2 shows this concept where two applications: TPEG-PTI (CEN ISO/TS 18234-5) and TPEG-RTM messages are streamed together.

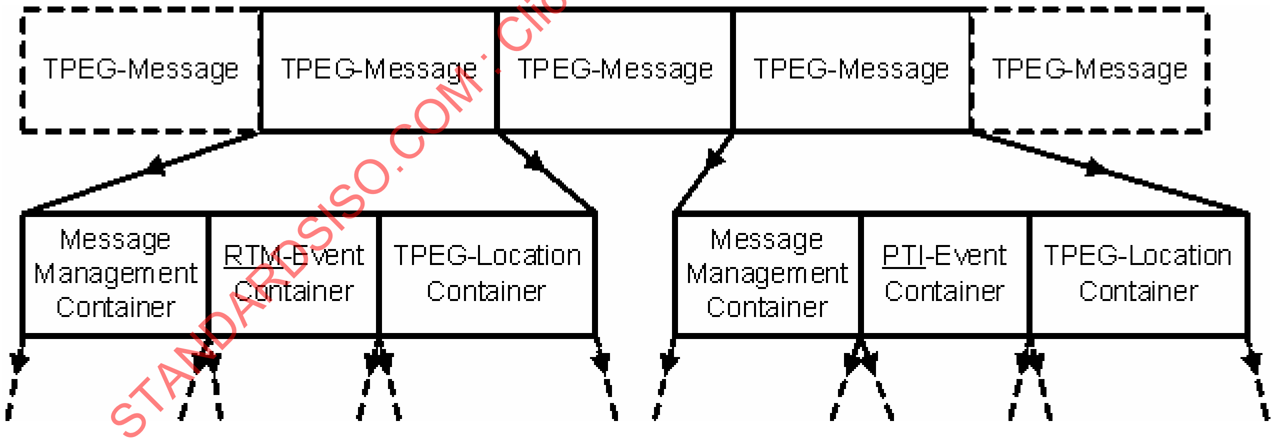

Figure 2 — TPEG-messages showing message management, event and location containers

Where a TPEG-message is one carrying traffic and travel information, Figure 2 also shows that it comprises three "containers": one for the message management, one for event content (e.g. "Accident – Buses running slowly, etc.) and one for the location content (both machine readable and human understandable data).

5.3 TPEG-messages delivering additional information

TPEG-messages may also contain vital information for the full use of a service. For example, a special form of message which is only a message management container (without any associated location container) may be inserted to cancel an existing message (see 5.4.1). This concept is illustrated, in Figure 3.

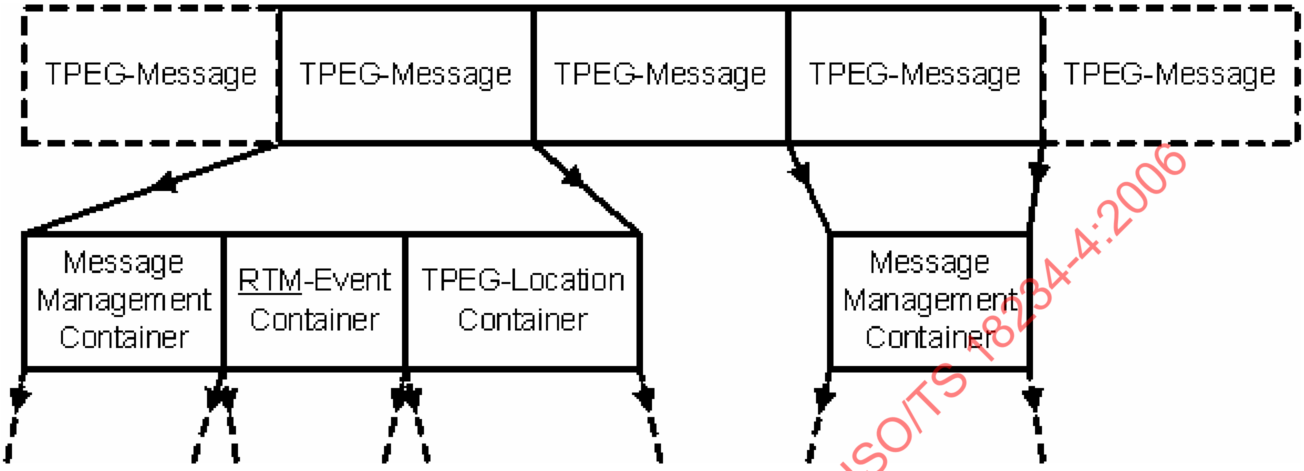

Figure 3 — Delivering additional non-message information

5.4 Elements of a TPEG road traffic message

Most elements of a road traffic message are optional, sent only if specifically required. Thus a TPEG-message container may include various elements according to the following descriptions. Figure 4 shows a TPEG-RTM message, which has three containers.

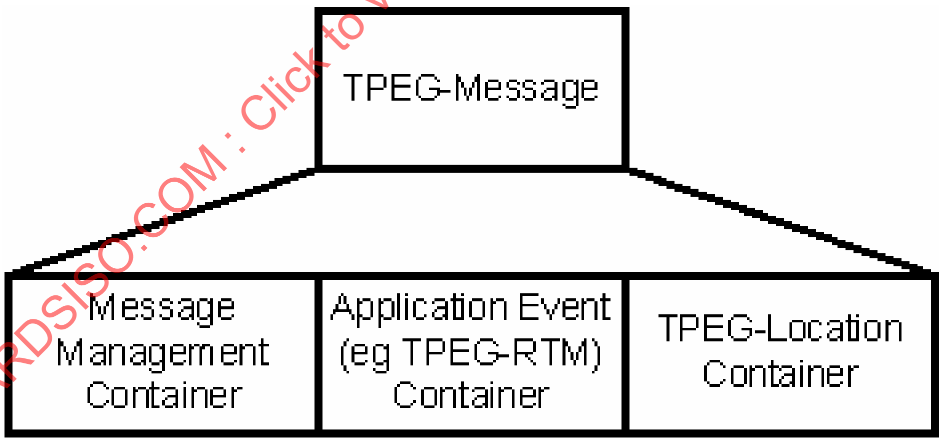
TPEG-Message
Message Management Container
Application Event (eg TPEG-RTM) Container
TPEG-Location Container

Figure 4 — TPEG-RTM message normally has three containers

5.4.1 Cancellation message

A special case, used to cancel an earlier message, only comprises the two mandatory elements, message identifier and version number = 255. Note it does not have an associated location container. This is shown in Figure 5.

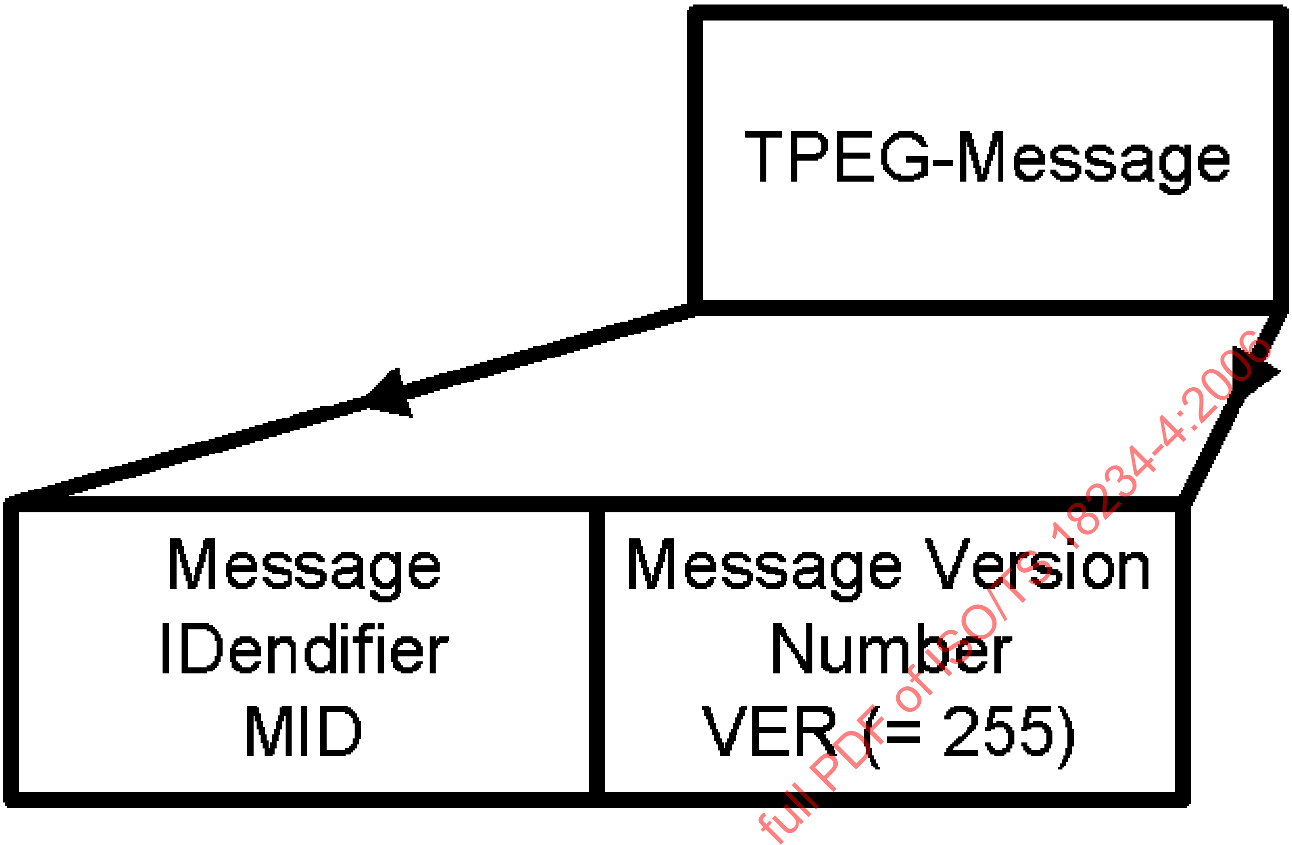

Figure 5 — Cancellation message

5.4.2 Short RTM-messages

Often a message will include only a few elements. The shortest, non-cancellation message may contain only four elements: MID, VER, EVE and LOC.

5.4.3 RTM-messages — All possible elements

Road traffic messages could theoretically include *all* the elements and containers shown in Figure 6.

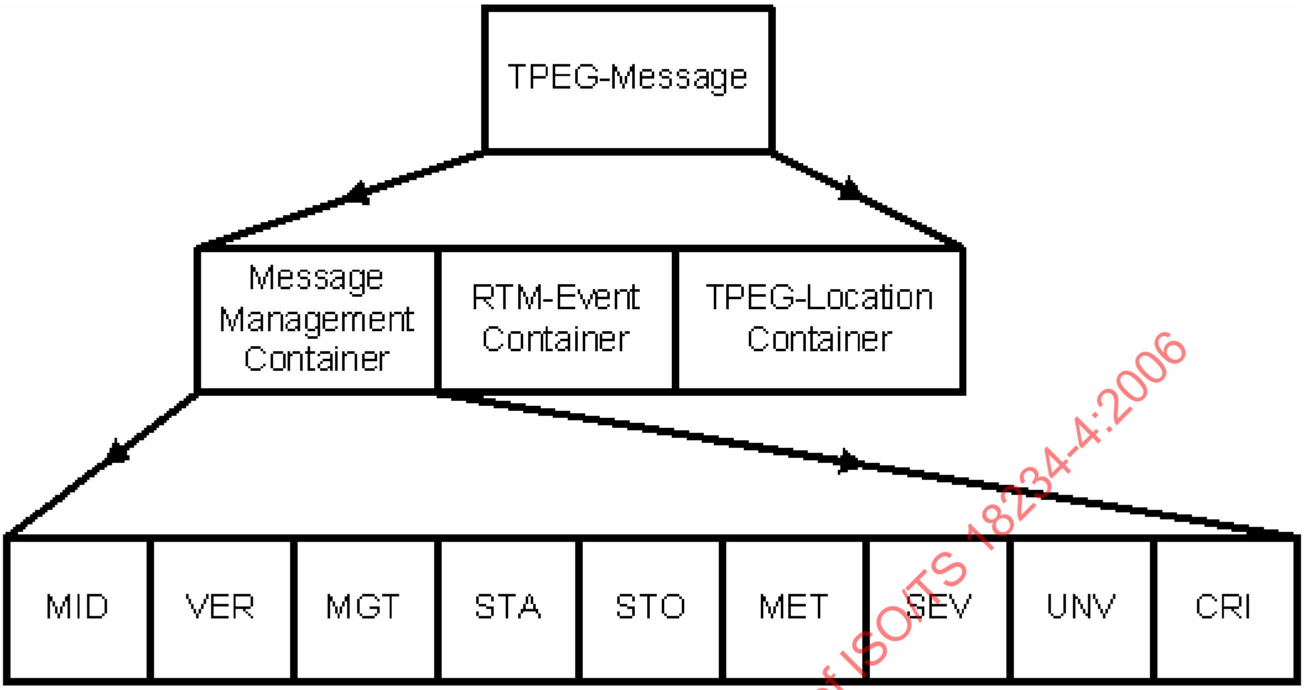

Figure 6 — RTM messages – Message management elements

5.4.4 Declarative coding

NOTE Every TPEG-RTM element is declaratively coded, and therefore the order of the elements is not defined.

This method of coding allows a road traffic message to contain one or more event description elements, according to the need to describe various aspects of the event. For example, an accident with appropriate detail and the resultant network performance with appropriate detail and a suggested diversionary route may be combined together within the same message.

5.5 Message management

5.5.1 Dates and times

Date and time references are included within the road traffic message application to serve two distinct purposes. The first is for use by the TPEG-decoder, for both presentation to the user and for filtering of information. The second is essentially for message management, and is not intended for display to the end-user. Date and time should be expressed with an accuracy of ± 1 second, referenced to Universal Co-ordinated Time (UTC).

NOTE 1 For the use of the term UTC, see ITU Recommendation 535-1; *Use of the term UTC*.

NOTE 2 For simplicity in reading this specification the complete phrase ‘date and time’ is often shortened to only: ‘time’. It should be noted however that the coding algorithm for date and time always causes both date and time to be conveyed in a message.

5.5.1.1 Start and stop times

The most obvious use is when describing events with scheduled start and end times, for example the times during which a road is planned for closure for roadworks. Provision is made to specify both a start time and a stop time for each message. These times are intended for presentation to the user, and possibly may be used as an element in message filtering by the end-user’s TPEG-decoder.

In the case of a message relating to an unplanned incident, for example an accident, it may be unnecessary to specify a start time as the incident has already occurred. The end time of an unplanned incident is likely to change over the lifetime of a message, since it may not be known with any certainty, sometimes it may be difficult to specify a stop time.

5.5.1.2 Message expiry time

Another use of time information is for message management purposes.

A mechanism is required to ensure that old messages are eventually purged from TPEG-decoders. This is because there is no guarantee that, once a TPEG-decoder has received a message, it will receive an update message, or cancellation message. A service provider sending the message expiry time to detail when the TPEG-decoder should delete a message will therefore achieve message purging.

The message expiry time is not presented to the end-user; it is only for message deletion purposes in the TPEG-decoder.

5.5.1.3 Message generation time

A message generation time may also be included with each message. This is not intended for display to the end-user, as its primary purpose is to enable a service provider to track a message through the distribution and broadcast infrastructure from end-to-end.

If a service provider intends to make use of the service component reset feature (see 5.5.1.5) provided by the service and network information (TPEG-SNI) application (CEN ISO TS 18234-3), the inclusion of the message generation time with every non-cancellation (MID and VER = 255) message is mandatory.

5.5.1.4 Time schedule information

For certain types of messages there is a need to give a schedule for when the event takes place. For single events this time information can be handled by the start and stop time functionality but for certain types, for example a schedule for the opening times of a tunnel or alternative starting times for convoy driving there is a need to be able to send schedules for the events. These times are intended for presentation to the user.

5.5.1.5 Service component reset

Another means of deleting messages is possible with the service component reset (SCR) feature provided in the service and network information (TPEG-SNI) application (CEN ISO TS 18234-3). Use of SCR by a service provider is optional.

The service component reset is delivered in the guide to the service table for a specific service component used for the RTM application. On receipt of the SCR for the RTM application, the TPEG-decoder knows that all messages older than the <time_t> value in the SCR table are no longer valid. Hence, the SCR feature provides the ability for a service provider to delete several messages with a single command, as may exceptionally be required in order to reset the message stock following a major problem with the message generation system.

5.5.2 Road traffic message effect and reliability

Road traffic messages essentially contain two primary pieces of information: what has occurred, and where. For an end-user or agent system to make judgements about the effect on travel, it is often necessary to provide further information. A service provider may optionally provide additional information.

5.5.2.1 Severity factor

A major factor in accessing which messages to present is how seriously the incident is likely to affect the journey. This is a judgement that can only be assessed by the service provider who will choose one of five levels to represent the disruption to travel. The severity factor is determined by the service provider taking into account the seriousness of the incident, weighted by the traffic density at the location affected, to represent the disruption to travel. The severity factor is likely to change over the lifetime of an event as it develops.

The severity factor may be one element that is assessed and used by broadcast systems and bearers to determine message repetition rates, or even which messages to ignore entirely should the total number of messages exceed capacity on lower data-rate bearers.

5.5.2.2 Unverified information

Although most service providers compile information received from reliable sources, on occasion, due to the potential importance of the information, it may be appropriate to send a message about an incident that has not yet been verified. A particular message may be coded to indicate to the end-user that it has not yet been verified. The TPEG-decoder should present the information to the end-user, indicating that the message has not been verified.

5.5.3 Cross-reference information concept

Transport Infrastructures are complex and a single incident may have a profound affect on the performance of many different parts of the infrastructure. Cross-reference information (CRI) fields in the road traffic message application would allow each message to be cross-referenced to other messages, either within the road traffic message application, or in other TPEG applications. For example, a road/rail level crossing, which has “stuck half open and half closed”, may affect travel for drivers, bus and coach users, and rail passengers alike. In this case, CRI may be used to cross-reference from a public road traffic message application to a private road traffic message application for bus operators, and to a public transport information application, which is rail focussed.

The method of cross-referencing will be described in a subsequent version. A new <rtm_component> would be used to convey the cross-referencing information of a message.

5.6 Event description

A major objective of service providers in the development of intelligent information systems has been the ability to present descriptions of road traffic messages to end-users, independent of language. Currently the most suitable method of achieving this essential objective is to use a standardized catalogue of descriptive words or phrases, each represented by a short-form code. A TPEG-decoder interprets this code by reference to a stored table, and it can then present the phrase appropriately to the end-user. Presentation can be verbal or in written text form in a language determined by the user, or represented symbolically by an icon.

In TPEG-RTM, the entire event information is built up using descriptive “words” or very short 2/3 word phrases, with numeric and other quantifiers and qualifiers, selected from a hierarchical classification structure.

At the highest level (level one), there are a small number of distinct classes, (for example, ACCIDENT, VISIBILITY, ROAD CONDITIONS) which have been chosen to give the end-user a rapid understanding of the message content. Each level one class then has a number of lower-level branches allowing increased message detail to be described. This allows total flexibility for the service provider to send as much or as little detail as they want (or have), and independently, the end-user system to derive and present increasing levels of detail.

Events are not restricted to being described by reference to only a single level one class. For example, a message describing freezing snow may appear under both “WEATHER” and “ROAD CONDITIONS” classes.

EXAMPLE This concept of a message using several levels may be illustrated by considering this example of a message:

"An articulated lorry has overturned in the slow-vehicle lane."

Level 1	ACCIDENT	(ONE)
Level 2	SLOW-VEHICLE LANE	
Level 2	VEHICLE	(ONE)
Level 3	OVERTURNED	
Level 3	HEAVY GOODS VEHICLE	
Level 4	ARTICULATED	

At its simplest, a TPEG-decoder could just advise the end-user of a SINGLE ACCIDENT ahead. A TPEG-decoder decoding to the second level would be able to enhance this description to advise that the ACCIDENT involved a SINGLE VEHICLE, in the SLOW-VEHICLE LANE. Level 3 information allows the detail of OVERTURNED and HEAVY GOODS VEHICLE to be added, and level 4 adds further the fact that it is ARTICULATED.

Suppliers of TPEG-decoders have the freedom to present information to whatever level they feel appropriate; any may choose different levels for different aspects of the message. A service provider, similarly, can choose the amount of detail to send. The hierarchical approach is useful for distribution on bearers of different capacity. High capacity ones may carry all levels of the message, whilst lower ones may carry fewer levels, but still retain the essential (top level) elements of a message.

The hierarchical coding structure is future-proof, since it allows a new message component to be added and used by a service provider. Existing TPEG-decoders, although not able to decode it, can represent the message component fully to one level higher.

The method also results in efficient coding where many of the event tables can be used under several level-one class messages, for example, position, vehicle-type, etc. This reduces the number of reference tables that need to be stored in the TPEG-decoder.

5.7 Location Referencing

Messages relate to real-world objects that are called locations. Each message requires a description of the location, to which the message relates. Simple TPEG-decoders without navigational systems or digital maps will require the location to be expressed as character strings, using familiar place names and road numbers to be presented either as text or speech to the end-user. Intelligent systems, such as digital mapped-based TPEG-decoders, require the location information to be expressed in a machine-readable coded form.

Within the "location co-ordinates" elements, TPEG-Loc combines both requirements in a way that permits machine interpretation while simultaneously ensuring useful, human understandable, analogues are available. The specification for location referencing is fully described in TPEG-Loc (CEN ISO TS 18234-6).

6 RTM container

6.1 Structure of road traffic messages

Here follows an indication of the structure of a TPEG-road traffic message in a hierarchical sense and an "internal hierarchical index" to the message content — message management, event description — structures and finally coding, for the message management and RTM-event containers, as shown in Figure 7.

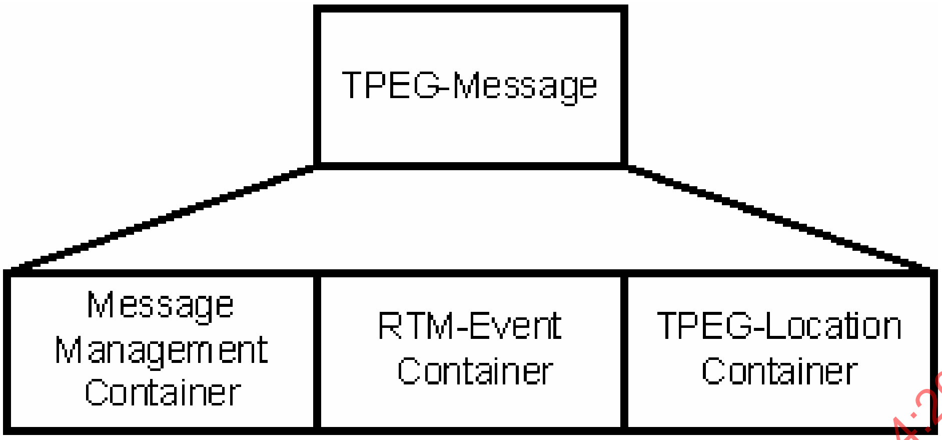

Figure 7 — TPEG-RTM message containers

MANDATORY ELEMENTS	
road_traffic_message (message_id, version_number,[ ]..)	: Section 7.1
DATE AND TIME ELEMENTS	
message_generation_time	: Section 7.2
start_time	: Section 7.2
stop_time	: Section 7.2
message_expiry_time	: Section 7.2
EFFECT AND RELIABILITY ELEMENTS	
severity_factor	: Section 7.3
unverified_information	: Section 7.3
LOCATION ELEMENTS	
TPEG-Loc	: (CEN ISO/TS 18234-6)
TIME INFORMATION ELEMENTS	
repetitive_time([ ])	: Section 7.2
non_repetitive_time	: Section 7.2
EVENT DESCRIPTIVE ELEMENTS	
accidents ([ ])	: Section 8.2.1
obstructions ([ ])	: Section 8.2.2
activities ([ ])	: Section 8.2.3
road_conditions ([ ])	: Section 8.2.4

network_performance ([ ])	: Section 8.2.5
network_conditions ([ ])	: Section 8.2.6
facilities_performance ([ ])	: Section 8.2.7
moving_hazard ([ ])	: Section 8.2.8
security_alert ([ ])	: Section 8.2.9
public_transport_information ([ ])	: Section 8.2.10
visibility ([ ])	: Section 8.2.11
weather ([ ])	: Section 8.2.12
diversion_advice ([ ])	: Section 8.2.13

6.2 Notation

The structure of road traffic message application is developed in a stylized indented form, which shows the hierarchy of the various data structures, the sub-components that are required and those that are optional and repeatable.

EXAMPLE:

Appointment (Date, Time)

Date (year, month, day)

Time (hour, minute, second)

In the above example, the data structure called ‘appointment’ has two components, ‘date’ and ‘time’.

Date is itself composed of components ‘year’, ‘month’, and ‘day’. Similarly, ‘time’ is composed of components ‘hour’, ‘minute’, and ‘second’.

It may well be that it is not strictly necessary to include time in every appointment. Furthermore, when time is included, the second field may not be necessary. To illustrate this, the data structures can be written thus:

Appointment (Date, [ ]..)

Time (hour, minute, [ ]..)

second

Date (year, month, day)

The square brackets in the appointment structure indicate where optional elements listed directly below may be included. These options are indented. In the example, there is only one option, that of ‘time’. Time itself has two fixed elements, and one optional element. The only option is ‘second’, which is shown indented. The data structure ‘date’ is separate.

Taking this concept further, it can be shown that both time and date may be designed as options in the appointment structure, minutes and seconds within time, and year and month within date.

Appointment ([ ] ..)

Date (day, [ ]..)

month

year

Time (hour, [ ]..)

minutes

seconds

6.3 RTM application component frame

6.3.1 Coding syntax and semantics

The road traffic message application is coded according to the syntax and semantics described in TPEG-SSF (CEN ISO/TS 18234-2).

6.3.2 Component frame

Section 6.3.3 defines the road traffic message application component frame. The service component id (scid) is allocated dynamically by the TPEG-SNI application, as described in TPEG-SNI (CEN ISO/TS 18234-3).

Each <road_traffic_message> element should only occur at most once, in the component frame.

The CRC check of the n* <road_traffic_message> is two bytes long, and is based on the ITU polynomial $x^{16} + x^{12} + x^5 + 1$.

The RTM Component Data CRC is calculated from all the bytes of the RTM component data (i.e. n*<road_traffic_message>), including the "Number of RTM messages" field.

6.3.3 Coding of the RTM application component frame

<component_frame(x)>:=	: Road Traffic Message Application
<intunti>(scid),	: Service Component Identifier (scid = x)
<intunli>,	: Length of application data in bytes
<crc>,	: Header CRC
	: Application Data
<intunti>(n),	: Number of RTM messages
n * **<road_traffic_message>**,	: Road Traffic Message messages
<crc>;	: CRC check of all messages

7 Message management container

7.1 Mandatory elements

A road traffic message consists at least of two mandatory elements, the message identifier and a version number.

7.1.1 Message identifier (MID)

The message identifier is described in 3.8.

message_id (number)	{range: 0 .. 65535}

7.1.2 Version number (VER)

The version number is described in 3.16.

version_number (number)	{range 0.. 255}

7.2 Date and time elements

Date and time elements are described in Section 3. The following sections, 7.2.1 to 7.2.4, are coded according to TPEG-SSF (CEN ISO/TS 18234-2).

7.2.1 Message generation time (MGT)

The message-element message generation time is described in 3.7

message_generation_time (date and time)	{resolution: seconds}

7.2.2 Start time (STA)

The message-element start time is described in 3.11.

start_time (date and time)	{resolution: minutes}

7.2.3 Stop time (STO)

The message-element stop time is described in 3.13.

stop_time (date and time)	{resolution: minutes}

7.2.4 Message expiry time (MET)

The message-element message expiry time is described in 3.6.

message_expiry_time (date and time)	{resolution: minutes}

7.2.5 Time schedule information (TSI)

The RTM-component time schedule information is described in 3.14

repetitive_time (time)	{resolution: minutes}
non_repetitive_time (date and time)	{resolution: seconds}

7.3 Severity and reliability elements

7.3.1 Severity factor (SEV)

Severity factor is described in 3.10.

severity_factor (number) {range: see TPEG table rtm31}

If the severity factor element is not sent, ‘unspecified’ should be assumed: this equates to code 255 in TPEG table rtm31.

7.3.2 Unverified information (UNV)

Unverified information is described in 3.15.

unverified_information (number) {range: see TPEG table rtm46}

If the unverified information element is not sent, ‘verified’ should be assumed: this equates to code 255 in TPEG table rtm46.

7.4 Coding of the message management container

7.4.1 Road traffic message

```
<road_traffic_message>:=

    <intunli>(mid)                                   : Message ID

    <intunti>(ver)                                   : Version number

    <intunli>                                        : Number of bytes following the length indicator

    <bitswitch>(selector)                            : Message elements supplied

    if (selector = xxxxxxx1) <time_t>                : Message generation time

    if (selector = xxxxxx1x) <time_t>                : Start time

    if (selector = xxxxx1xx) <time_t>                : Stop time

    if (selector = xxxx1xxx) <time_t>                : Message expiry time

    if (selector = xxx1xxxx) <rtm31>                 : Severity factor, see TPEG table rtm31

    if (selector = xx1xxxxx)  <intunlo>              : Reserved for future use

    if (selector = x1xxxxxx) <rtm46>                 : Unverified information, see TPEG table rtm46

    if (selector = 1xxxxxxx) <rtm_components> : Road traffic message components
```

7.4.2 RTM components template

```
<rtm_components>:=

   <intunti>(n)                                : Number of components

   n * <rtm_component()>                       : Road traffic message component
```

7.4.3 RTM component template

<rtm_component(x)>:=	: Road traffic message component template
<intunti>(id)	: Identifier (id)
<intunli>(n)	: Length, n, of component data in bytes
n * **<byte>**	: Component data

7.4.4 Time schedule information

7.4.4.1 Coding of repetitive time

<rtm_component(70)>:=	: Repetitive Time information
<intunti>(id)	: Identifier, id = 70 hex
<intunli>(n)	: Length, n, of component data in bytes
<intunti>	: Hour, 0..23
<intunti>	: Min, 0..59
<intunli>	: Duration in minutes, 0 to 10079 (max. one week)
<day_mask>	: see TPEG-SSF (CEN ISO/TS 18234-2), 6.1.2.2

NOTE Repetitive time information shall be bounded by the message start and stop time (STA and STO).

7.4.5 Coding of non-repetitive time

<rtm_component(71)>:=	: Non-Repetitive Time information
<intunti>(id),	: Identifier, id = 71 hex
<intunli>(n),	: Length, n, of component data in bytes
<intunti>(m),	: Number of operating times
m * **<non_rep_time>**,	: Start time and duration

NOTE Non-repetitive time information shall be bounded by the message start and stop time (STA and STO).

<non_rep_time>:=	: Non-Repetitive Time type
<time_t>,	: Start date and time
<duration>;	: Duration in seconds (see TPEG-SFF, Section 6.1.2.3)
	: 0 = a start time without duration

7.4.6 RTM component – TPEG-Loc link

<rtm_component(90)>:=	: TPEG-Location Referencing (see CEN ISO/TS 18234-6)
<intunti>(id),	: Identifier, id = 90 hex
<intunli>(n),	: Length, n, of component data in bytes
<tpeg_loc_container>;	: TPEG-Location Container

8 Event container

8.1 Event description

As described in 5.6, the event description uses a hierarchical approach to define events in a number of levels, which allow recursive coding.

8.2 Level one classes

The level one classes are listed in Table 1.

Table 1 — Level one classes

Class	Description
Accident	Descriptions of situations in which road users (vehicles, animals and people) do not behave in a predictable or safe manner and either impact with each other or the roadside infrastructure and in some cases may leave the road.
Obstructions	Descriptions of situations in which road users (vehicles, animals and people) or other causes (man-made or environmental) make it difficult or impossible for other road users to progress along that part of roadway.
Activities	Descriptions of events (particularly involving people) that can have an impact on the road traffic.
Road conditions	Descriptions of changes to the properties of the surface of the road (disruption to the surface, adhesion reduction or changes to the road markings) that may affect the progress of the road user.
Network performance	Descriptions of the effect on road users (delay, flow or speed) that arise out of external events.
Network conditions	Descriptions of changes to network conditions planned, imposed or advised by the road network operator that affect the drivers, vehicles and routing.
Facilities performance	Descriptions of the changes (for any reason) to the availability of control, assistance and roadside services.
Moving hazard	Descriptions of situations in which non-stationary road users (vehicles, animals and people) make it hazardous for others to use the road.
Security alert	Descriptions of situations which may have safety implications for the road user.
Public transport Information	Descriptions of abnormal operation of all forms of public transport which may affect the road user's journey.
Visibility	Descriptions of changes to the normal obscurity or lighting conditions that may affect the road user's ability to see the road or other road users ahead.
Weather	Descriptions of weather situations that are affecting or may affect the progress of the road-user.
Diversion advice	Descriptions of alternative routes for use advised via the service provider.

In 8.2.1 – 8.2.13, each level one class and its lower levels are fully notated together with references to TPEG tables using rtmxx notation.

In the case of ‘numbers’ the intended range is shown in {example range: ..}, flushed right.

‘Numag’ is a general method for coding quantities of objects; this is described in TPEG-SSF (CEN ISO/TS 18234-2).

8.2.1 Accident

```
accidents (number_of, [ ].. )                  {example range: 0..20}

        position (position/rtm10 )

        animals (number_of, [ ].. )            {example range: numag, 0..10 000}

                position (position/rtm10 )

                animal_problem (animal_problem/rtm23, [ ].. )

                animal_info (animal_type/rtm21, animal_size/rtm22, [ ].. )

        vehicles (number_of, [ ].. )           {example range: numag, 0..10 000}

                position (position/rtm10 )

                vehicle_problem (vehicle_problem/rtm03, [ ].. )

                vehicle_info (vehicle_type/rtm01, vehicle_subtype, [ ].. )

        people (number_of, [ ].. )             {example range: numag, 0..1 000 000}

                position (position/rtm10 )

                people_problem (people_problem/rtm20, [ ].. )

                people_info (people_type/rtm19, [ ].. )
```

8.2.2 Obstructions

```
obstructions (number_of, [ ].. )               {example range: 0..20}

        position (position/rtm10 )

        animals (number_of, [ ].. )            {example range: numag, 0..10 000}

                position (position/rtm10 )

                animal_problem (animal_problem/rtm23, [ ].. )

                animal_info (animal_type/rtm21, animal_size/rtm22, [ ].. )

        vehicle (number_of, [ ].. )            {example range: numag, 0..10 000}

                position (position/rtm10 )

                vehicle_problem (vehicle_problem/rtm03, [ ].. )

                vehicle_info (vehicle_type/rtm01, vehicle_subtype, [ ].. )
```

```
    people (number_of, [ ].. )              {example range: numag, 0 ..1 000 000}
        position (position/rtm10 )
        people_problem (people_problem/rtm20, [ ].. )
        people_info (people_type/rtm19, [ ].. )
    object (number_of, [ ].. )              {example range: numag, 0..1 000}
        position (position/rtm10 )
        object_problem (object_problem/rtm12, [ ].. )
```

8.2.3 Activities

```
activities (number_of, [ ].. )              {example range: 0..20}
    position (position/rtm10 )
    activity (activity_type/rtm24, activity_subtype, [ ].. )
    people (number_of, [ ]..)               {example range: numag, 0..1 000 000}
        position (position/rtm10)
        people_problem (people_problem/rtm20 )
        people_info (people_type/rtm19 [ ].. )
```

8.2.4 Road conditions

```
road_conditions ( [ ].. )
    position (position/rtm10 )
    surface (general_magnitude/rtm31, surface_condition/rtm18, [ ].. )
    adhesion (general_magnitude/rtm31, adhesion_condition/rtm39, [ ].. )
marking (marking_condition/rtm15, [ ].. )
```

8.2.5 Network performance

```
network_performance ( [ ].. )
    performance (network_performance/rtm34, [ ] )
        length_affected (number)        {example range: 0..100 000 in*10 metres}
    speed (number)      {example range: 0..60 in*0,5 metres/sec}
    delay (number) {example range: 0..65 000 in*1 min}
    travel_time (number)                {example range: 0..65 000 in*1 min}
```

8.2.6 Network conditions

```
network_conditions ( [ ].. )
        position (position/rtm10 )
        regulation (regulation/rtm45, regulation_quantifier (number/numag), [ ].. )
                length_affected (number)      {example range: 0..100 000 in *10 metres}
                regulation (condition_status/rtm47, [ ].. )
        restriction (restriction/rtm49, [ ].. )
                length_affected (number)      {example range: 0..100 000 in *10 metres}
                restriction (condition_status/rtm47, [ ].. )
        roadworks (roadworks/rtm50,  [ ].. )
                length_affected (number)      {example range: 0..100 000 in *10 metres}
                roadworks (condition_status/rtm47, [ ].. )
```

8.2.7 Facilities performance

```
facilities_performance ( [ ].. )
        traffic_control (traf_con_type/rtm42, traf_con_status/rtm43, [ ].. )
                position (position/rtm10 )
        roadside_assistance (road_assist_type/rtm32, road_assist_status/rtm33, [ ].. )
        roadside_services (road_serv_type/rtm37, road_serv_status/rtm38, [ ].. )
```

8.2.8 Moving hazard

```
moving_hazards (number_of, [ ].. )            {example range: 0..20}
        position (position/rtm10 )
        animals (number_of, [ ].. )           {example range: numag, 0..10 000}
                position (position/rtm10 )
                animal_problem (animal_problem/rtm23, [ ].. )
                animal_info (animal_type/rtm21, animal_size/rtm22, [ ].. )
        vehicles (number_of, [ ].. )          {example range: numag, 0..10 000}
                position (position/rtm10 )
                vehicle_problem (vehicle_problem/rtm03, [ ].. )
                vehicle_info (vehicle_type/rtm01, vehicle_subtype, [ ].. )
```

people (number_of, [ ]..) {example range: numag, 0..1 000 000}

position (position/rtm10)

people_problem (people_problem/rtm20, [ ]..)

people_info (people_type/rtm19, [ ]..)

8.2.9 Security alert

security_alert (security_alert/rtm36, [ ]..)

8.2.10 Public transport information

public_transport_info (public_transp_type/rtm40, public_transp_status/rtm41, [ ]..)

8.2.11 Visibility

visibility ([ ]..)

obscurity (obscurity_problem/rtm17, visibility_distance (number), [ ]..)
{example range: 0..2 500 in*10 metres}

visual_acuity (acuity_problem/rtm13, [ ]..)

lighting (lighting_problem/rtm14, [ ]..)

length_affected (number) {example range: 0..100 000 in*10 metres}

8.2.12 Weather

weather ([ ]..)

precipitation (general_magnitude/rtm31, precip_problem/rtm29, [ ]..)

wind (wind_speed(number), wind_problem/rtm30, [ ]..)
{example range: 0..150 km/h in 0,5m/s}

temperature (temperature (number), [ ]..) {example range: -40°C..+100 °C}

8.2.13 Diversion advice

diversion_advice ([ ]..)

vehicle_info (vehicle_type/rtm01, vehicle_subtype, [ ]..)

regulation (regulation/rtm 45, regulation_quantifier (number/numag))

vehicle_current_position (position/rtm10)

advice (condition_status/rtm47, advice_type/rtm35, [ ]..)

routing ([ ]..)

location_container(location_container/tpeg_loc_container) (CEN ISO/TS 18234-6)

for (distance/m) {range in metres * 10)

8.3 Coding of the events container

8.3.1 Accident

<rtm_component(80)>:=	: Accident event class
<intunti>(id),	: Identifier, id = 80 hex
<intunli>(n),	: Length, n, of component data in bytes
<intunti>,	: Number of accidents
m * **<accident_component()>**;	: Accident components

8.3.1.1 Accident component template

<accident_component(x)>:=	: Accident component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.1.2 Position

<accident_component(00)>:=	: Position of accident(s) component
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm10>;	: Position of accident(s), TPEG table rtm10

8.3.1.3 Animals

<accident_component(01)>:=	: Animal component
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<numag>,	: Number of animals
m * **<animal_component()>**;	: Animal components

8.3.1.3.1 Animal component template

<animal_component(x)>:=	: Animal component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.1.3.2 Animal position

<animal_component(00)>:=	: Animal position
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm10>;	: Position, TPEG table rtm10

8.3.1.3.3 Animal problem

<animal_component(01)>:=	: Animal problem
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm23>,	: Problem with animals, TPEG table rtm23
m * **<animal_problem_component()>**;	: Animal problem components (UAV)

8.3.1.3.3.1 Animal problem component template

<animal_problem_component(x)>:=	: Animal problem component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.1.3.4 Animal information

<animal_component(02)>:=	: Animal information
<intunti>(id),	: Identifier, id = 02 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm21>,	: Type of animals involved, TPEG table rtm21
<rtm22>,	: Size of animals involved, TPEG table rtm22
m * **<animal_information_component()>**;	: Animal information components (UAV)

8.3.1.3.4.1 Animal information component template

<animal_information_component(x)>:=	: Animal information component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.1.4 Vehicles

<accident_component(02)>:=	: Vehicle component
<intunti>(id),	: Identifier, id = 02 hex
<intunti>(n),	: Length, n, of component data in bytes
<numag>,	: Number of vehicles
m * **<vehicle_component()>**;	: Vehicle components

8.3.1.4.1 Vehicle component template

<vehicle_component(x)>:=	: Vehicle component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.1.4.2 Vehicle position

<vehicle_component(00)>:=	: Vehicle position
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm10>;	: Position, TPEG table rtm10

8.3.1.4.3 Vehicle problem

<vehicle_component(01)>:=	: Vehicle problem
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm03>,	: Problem with vehicles, TPEG table rtm03
m * **<vehicle_problem_component()>**;	: Vehicle problem components (UAV)

8.3.1.4.3.1 Vehicle problem component template

<vehicle_problem_component(x)>:=	: Vehicle problem component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.1.4.4 Vehicle information

<vehicle_component(02)>:=	: Vehicle information
<intunti>(id),	: Identifier, id = 02 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm01>(type),	: Type of vehicles involved, TPEG table rtm01
<**intunti**>(subtype),	: Subtype of vehicles involved, see table in 8.4.3
m * **<vehicle_information_component()>**;	: Vehicle information components (UAV)

8.3.1.4.4.1 Vehicle information component template

<vehicle_information_component(x)>:=	: Vehicle information component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.1.5 People

<accident_component(03)>:=	: People component
<intunti>(id),	: Identifier, id = 03 hex
<intunti>(n),	: Length, n, of component data in bytes
<numag>,	: Number of people involved
m * **<people_component()>**;	: People components

8.3.1.5.1 People component template

<people_component(x)>:=	: People component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.1.5.2 People position

<people_component(00)>:=	: People position
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm10>;	: Position, TPEG table rtm10

8.3.1.5.3 People problem

<people_component(01)>:=	: People problem
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm20>;	: Problem with people, TPEG table rtm20
m * **<people_problem_component>**;	: People problem components (UAV)

8.3.1.5.3.1 People problem component template

<people_problem_component(x)>:=	: People problem component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.1.5.4 People information

<people_component(02)>:=	: People information
<intunti>(id),	: Identifier, id = 02 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm19>,	: Type of people involved, TPEG table rtm19
m * **<people_information_component()>**;	: People information components (UAV)

8.3.1.5.4.1 People information component template

<people_information_component(x)>:=	: People information component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.2 Obstructions

<rtm_component(81)>:=	: Obstruction event class
<intunti>(id),	: Identifier, id = 81 hex
<intunli>(n),	: Length, n, of component data in bytes
<intunti>,	: Number of obstructions
m * **<obstruction_component()>**;	: Obstruction components

8.3.2.1 Obstruction component template

<obstruction_component(x)>:=	: Obstruction component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.2.2 Position

<obstruction_component(00)>:=	: Position of obstruction(s) component
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm10>;	: Position of obstruction(s), TPEG table rtm10

8.3.2.3 Animals

<obstruction_component(01)>:=	: Animal component
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<numag>,	: Number of animals
m * **<animal_component()>;**	: Animal components

8.3.2.3.1 Animal component template

<animal_component(x)>:=	: Animal component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.2.3.2 Animal position

<animal_component(00)>:=	: Animal position
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm10>;	: Position, TPEG table rtm10

8.3.2.3.3 Animal problem

<animal_component(01)>:=	: Animal problem
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm23>,	: Problem with animals, TPEG table rtm23
m * **<animal_problem_component()>**;	: Animal problem components (UAV)

8.3.2.3.3.1 Animal problem component template

<animal_problem_component(x)> =	: Animal problem component template
<intunti>(id)	: Identifier, id = x hex
<intunti>(n)	: Length, n, of component data in bytes
n * **<byte>**	: Component data

8.3.2.3.4 Animal information

<animal_component(02)>:=	: Animal information
<intunti>(id),	: Identifier, id = 02 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm21>,	: Type of animals involved, TPEG table rtm21
<rtm22>,	: Size of animals involved, TPEG table rtm22
m * **<animal_information_component()>**;	: Animal information components (UAV)

8.3.2.3.4.1 Animal information component template

<animal_information_component(x)>:=	: Animal information component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.2.4 Vehicles

<obstruction_component(02)>:=	: Vehicle component
<intunti>(id),	: Identifier, id = 02 hex
<intunti>(n),	: Length, n, of component data in bytes
<numag>,	: Number of vehicles
m * **<vehicle_component()>**;	: Vehicle components

8.3.2.4.1 Vehicle component template

<vehicle_component(x)>:=	: Vehicle component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.2.4.2 Vehicle position

<vehicle_component(00)>:=	: Vehicle position
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm10>;	: Position, TPEG table rtm10

8.3.2.4.3 Vehicle problem

<vehicle_component(01)>:=	: Vehicle problem
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm03>,	: Problem with vehicles, TPEG table rtm03
m***<vehicle_problem_component>**;	: Vehicle problem components (UAV)

8.3.2.4.3.1 Vehicle problem component template

<vehicle_problem_component(x)>:=	: Vehicle problem component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.2.4.4 Vehicle information

<vehicle_component(02)>:=	: Vehicle information
<intunti>(id),	: Identifier, id = 02 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm01>(type),	: Type of vehicles involved, TPEG table rtm01
<intunti>(subtype),	: Subtype of vehicles involved, see table in 8.4.3
m * **<vehicle_information_component()>**;	: Vehicle information components (UAV)

8.3.2.4.4.1 Vehicle information component template

<vehicle_information_component(x)>:=	: Vehicle information component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.2.5 People

<obstruction_component(03)>:=	: People component
<intunti>(id),	: Identifier, id = 03 hex
<intunti>(n),	: Length, n, of component data in bytes
<numag>,	: Number of people involved
m * **<people_component()>**;	: People components

8.3.2.5.1 People component template

<people_component(x)>:=	: People component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.2.5.2 People position

<people_component(00)>:=	: People position
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm10>;	: Position, TPEG table rtm10

8.3.2.5.3 People problem

<people_component(01)>:=	: People problem
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm20>,	: Problem with people, TPEG table rtm20
m * **<people_problem_component>**;	: People problem components (UAV)

8.3.2.5.3.1 People problem component template

<people_problem_component(x)>:=	: People problem component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.2.5.4 People information

<people_component(02)>:=	: People information
<intunti>(id),	: Identifier, id = 02 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm19>,	: Type of people involved, TPEG table rtm19
m * **<people_information_component()>**;	: People information components (UAV)

8.3.2.5.4.1 People information component template

<people_information_component(x)>:=	: People information component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.2.6 Objects

<obstruction_component(04)>:=	: Object component
<intunti>(id),	: Identifier, id = 04 hex
<intunti>(n),	: Length, n, of component data in bytes
<numag>,	: Number of objects involved
m * **<object_component()>**;	: Object components

8.3.2.6.1 Object component template

<object_component(x)>:=	: Object component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.2.6.2 Object position

<object_component(00)>:=	: Object position
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm10>;	: Position, TPEG table rtm10

8.3.2.6.3 Object problem

<object_component(01)>:=	: Object problem
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm12>,	: Problem with object, TPEG table rtm12
m * **<object_problem_component>**;	: Object problem components (UAV)

8.3.2.6.3.1 Object problem component template

<object_problem_component(x)>:=	: Object problem component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.3 Activities

<rtm_component(82)>:=	: Activities event class
<intunti>(id),	: Identifier, id = 82 hex
<intunli>(n),	: Length, n, of component data in bytes
<numag>,	: Number of activities
m * **<activities_component()>**;	: Activities components

8.3.3.1 Activities component template

<activities_component(x)>:=	: Activities component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.3.2 Position

<activities_component(00)>:=	: Position of activities component
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm10>;	: Position of activities, TPEG table rtm10

8.3.3.3 Activity

<activities_component(01)>:=	: Activity component
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm24>(type),	: Activity type, TPEG table rtm24
<intunti>(subtype)	: Subtype of activity involved, see table in 8.4.4
m * **<activity_component()>**;	: Activity components (UAV)

8.3.3.3.1 Activity component template

<activity_component(x)>:=	: Activity component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.3.4 People

<activities_component(02)>:=	: People component
<intunti>(id),	: Identifier, id = 02 hex
<intunti>(n),	: Length, n, of component data in bytes
<numag>,	: Number of people involved
m * **<people_component()>**;	: People components

8.3.3.4.1 People component template

<people_component(x)>:=	: People component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.3.5 People position

<people_component(00)>:=	: People position
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm10>;	: Position, TPEG table rtm10

8.3.3.6 People problem

<people_component(01)>:=	: People problem
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm20>,	: Problem with people, TPEG table rtm20
m * **<people_problem_component>**;	: People problem components (UAV)

8.3.3.6.1 People problem component template

<people_problem_component(x)>:=	: People problem component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.3.7 People information

<people_component(02)>:=	: People information
<intunti>(id),	: Identifier, id = 02 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm19>,	: Type of people involved, TPEG table rtm19
m * **<people_information_component()>**;	: People information components (UAV)

8.3.3.7.1 People information component template

<people_information_component(x)>:=	: People information component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.4 Road Conditions

<rtm_component(83)>:=	: Road conditions event class
<intunti>(id),	: Identifier, id = 83 hex
<intunli>(n),	: Length, n, of component data in bytes
m * **<road_conditions_component()>**;	: Road conditions components

8.3.4.1 Road conditions component template

<road_conditions_component(x)>:=	: Road conditions component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.4.2 Position

<road_conditions_component(00)>:=	: Position of road condition(s) component
<intunti>(id)	: Identifier, id = 00 hex
<intunti>(n)	: Length, n, of component data in bytes
<rtm10>	: Position of road condition(s), : TPEG table rtm10

8.3.4.3 Surface

<road_conditions_component(01)>:=	: Surface
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm31>,	: General magnitude, TPEG table rtm31
<rtm18>,	: Surface condition, TPEG table rtm18
m * **<surface_component()>;**	: Surface component(s) (UAV)

8.3.4.3.1 Surface component template

<surface_component(x)>:=	: Surface component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.4.4 Adhesion

<road_conditions_component(02)>:=	: Adhesion
<intunti>(id),	: Identifier, id = 02 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm31>,	: General magnitude, TPEG table rtm31
<rtm39>,	: Adhesion condition, TPEG table rtm39
m * **<adhesion_component()>;**	: Adhesion component(s) (UAV)

8.3.4.4.1 Adhesion component template

<adhesion_component(x)>:=	: Adhesion component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.4.5 Marking

<road_conditions_component(03)>:=	: Marking
<intunti>(id),	: Identifier, id = 03 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm15>,	: Marking condition, TPEG table rtm15
m * **<marking_component()>;**	: Marking components (UAV)

8.3.4.5.1 Marking component template

<marking_component(x)>:=	: Marking component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.5 Network performance

<rtm_component(84)>:=	: Network performance event class
<intunti>(id),	: Identifier, id = 84 hex
<intunli>(n),	: Length, n, of component data in bytes
m * **<network_perform_component()>**;	: Network performance components

8.3.5.1 Network performance component template

<network_perform_component(x)>:=	: Network performance component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.5.2 Performance

<network_perform_component(00)>:=	: Performance
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm34>,	: Network Performance, TPEG table rtm34
m * **<performance_component()>**;	: Performance components

8.3.5.2.1 Performance component template

<performance_component(x)>:=	: Performance component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.5.2.2 Length of road affected

<performance_component(00)>:=	: Length of road affected
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<intunli>;	: Length of road affected (in metres * 10)

8.3.5.3 Speed

<network_perform_component(01)>:=	: Speed
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<intunti>;	: Speed (in m/s * 0,5)

8.3.5.4 Delay

<network_perform_component(02)>:=	: Delay
<intunti>(id),	: Identifier, id = 02 hex
<intunti>(n),	: Length, n, of component data in bytes
<intunli>;	: Delay (in minutes)

8.3.5.5 Travel time

<network_perform_component(03)>:=	: Travel time
<intunti>(id),	: Identifier, id = 03 hex
<intunti>(n),	: Length, n, of component data in bytes
<intunli>;	: Time (in minutes)

8.3.6 Network conditions

<rtm_component(85)>:=	: Network conditions event class
<intunti>(id),	: Identifier, id = 85 hex
<intunli>(n),	: Length, n, of component data in bytes
m * **<network_conditions_component()>**;	: Network conditions components

8.3.6.1 Network conditions component template

<network_conditions_component(x)>:=	: Network conditions component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.6.2 Position

<network_conditions_component(00)>:=	: Position
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm10>;	: Position, TPEG table rtm10

8.3.6.3 Regulation

<network_conditions component(01)>:=	: Regulation
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm45>,	: Regulation, TPEG table rtm45
<numag>,	: Regulation quantifier
m * **<regulation_component()>**;	: Regulation components

8.3.6.3.1 Regulation component template

<regulation_component(x)>:=	: Regulation component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.6.3.2 Length affected

<regulation_component(00)>:=	: Length affected
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<intunli>;	: Length of road affected (in metres * 10)

8.3.6.3.3 Regulation condition status

<regulation_component(01)>:=	: Regulation condition status
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm47>;	: Condition status, TPEG table rtm47

8.3.6.4 Restriction

<network_conditions_component(02)>:=	: Restriction
<intunti>(id),	: Identifier, id = 02 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm49>,	: Restriction condition, TPEG table rtm49
m * **<restriction_component()>**;	: Restriction components

8.3.6.4.1 Restriction component template

<restriction_component(x)>:=	: Restriction component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.6.4.2 Length affected

<restriction_component(00)>:=	: Length affected
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<intunli>;	: Length of road affected (in metres * 10)

8.3.6.4.3 Restriction condition status

<restriction component(01)>:=	: Restriction condition status
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm47>;	: Condition status, TPEG table rtm47

8.3.6.5 Roadworks

<network_conditions_component(03)>:=	: Roadworks
<intunti>(id),	: Identifier, id = 03 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm50>,	: Roadworks, TPEG table rtm50
m * **<roadworks_component()>**;	: Roadworks components

8.3.6.5.1 Roadworks component template

<roadworks_component(x)>:=	: Roadworks component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.6.5.2 Length affected

<roadworks_component(00)>:=	: Length affected
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<intunli>;	: Length of road affected (in metres * 10)

8.3.6.5.3 Roadworks condition status

<roadworks_component(01)>:=	: Roadworks condition status
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm47>;	: Condition status, TPEG table rtm47

8.3.7 Facilities performance

<rtm_component(86)>:=	: Facilities performance event class
<intunti>(id),	: Identifier, id = 86 hex
<intunli>(n),	: Length, n, of component data in bytes
m * **<facilities_performance_component()>**;	: Facilities performance components

8.3.7.1 Facilities performance component template

<facilities_performance_component(x)>:=	: Facilities performance component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.7.2 Traffic control

<facilities_performance_component(00)>:=	: Traffic control
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm42>,	: Traffic control type, TPEG table rtm42
<rtm43>,	: Traffic control status, TPEG table rtm43
m * **<traffic_control_component>**;	: Traffic Control components

8.3.7.2.1 Traffic control component template

<traffic_control_component(x)>:=	: Traffic Control component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.7.2.2 Traffic control position

<traffic_control_component(00)>:=	: Traffic control position
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm10>;	: Position, TPEG table rtm10

8.3.7.3 Roadside assistance

<facilities_performance_component(01)>:=	: Roadside assistance
<intunti>(id)	: Identifier, id = 01 hex
<intunti>(n)	: Length, n, of component data in bytes
<rtm32>	: Roadside assistance type, TPEG table rtm32
<rtm33>	: Roadside assistance status, TPEG table rtm33
m * **<roadside_assistance_component()>**	: Roadside assistance components (UAV)

8.3.7.3.1 Roadside assistance component template

<roadside_assistance_component(x)>:=	: Roadside assistance component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.7.4 Roadside services

<facilities_performance_component(02)>:=	: Roadside services
<intunti>(id),	: Identifier, id = 02 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm37>,	: Roadside services type, TPEG table rtm37
<rtm38>,	: Roadside services status, TPEG table rtm38
m * **<roadside_services_component()>**;	: Roadside services components (UAV)

8.3.7.4.1 Roadside services component template

<roadside_services_component(x)>:=	: Roadside services component template
<intunti>(id),	: Roadside services component id
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.8 Moving hazard

<rtm_component(87)>:=	: Moving hazard event class
<intunti>(id),	: Identifier, id = 87 hex
<intunli>(n),	: Length, n, of component data in bytes
<intunti>,	: Number of moving hazards
m * **<moving_hazard_component()>**;	: Moving hazard components

8.3.8.1 Moving hazard component template

<moving_hazard_component(x)>:=	: Moving hazard component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.8.2 Position

<moving_hazard_component(00)>:=	: Position of moving hazard(s)
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm10>;	: Position, TPEG table rtm10

8.3.8.3 Animals

<moving_hazard_component(01)>:=	: Animal component
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<numag>,	: Number of animals
m * **<animal_component()>;**	: Animal components

8.3.8.3.1 Animal component template

<animal_component(x)>:=	: Animal component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.8.3.2 Animal position

<animal_component(00)>:=	: Animal position
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm10>;	: Position, TPEG table rtm10

8.3.8.3.3 Animal problem

<animal_component(01)>:=	: Animal problem
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm23>,	: Problem with animals, TPEG table rtm23
m * **<animal_problem_component()>**;	: Animal problem components (UAV)

8.3.8.3.3.1 Animal problem component template

<animal_problem_component(x)>:=	: Animal problem component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.8.3.4 Animal information

<animal_component(02)>:=	: Animal information
<intunti>(id),	: Identifier, id = 02 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm21>,	: Type of animals involved, TPEG table rtm21
<rtm22>,	: Size of animals involved, TPEG table rtm22
m * **<animal_information_component()>**;	: Animal information components (UAV)

8.3.8.3.4.1 Animal information component template

<animal_information_component(x)>:=	: Animal information component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.8.4 Vehicles

<moving_hazard_component(02)>:=	: Vehicle component
<intunti>(id),	: Identifier, id = 02 hex
<intunti>(n),	: Length, n, of component data in bytes
<numag>,	: Number of vehicles
m * **<vehicle_component()>**;	: Vehicle components

8.3.8.4.1 Vehicle component template

<vehicle_component(x)>:=	: Vehicle component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.8.4.2 Vehicle position

<vehicle_component(00)>:=	: Vehicle position
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm10>;	: Position, TPEG table rtm10

8.3.8.4.3 Vehicle problem

<vehicle_component(01)>:=	: Vehicle problem
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm03>,	: Problem with vehicles, TPEG table rtm03
m***<vehicle_problem_component>**;	: Vehicle problem components (UAV)

8.3.8.4.3.1 Vehicle problem component template

<vehicle_problem_component(x)>:=	: Vehicle problem component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.8.4.4 Vehicle information

<vehicle_component (02)> =	: Vehicle information
<intunti>(id)	: Identifier, id = 02 hex
<intunti>(n)	: Length, n, of component data in bytes
<rtm01>(type)	: Type of vehicles involved, TPEG table rtm01
<intunti> (subtype)	: Subtype of vehicles involved, see table in 8.4.3
m * **<vehicle_information_component()>**	: Vehicle information components (UAV)

8.3.8.4.4.1 Vehicle information component template

<vehicle_information_component(x)>:=	: Vehicle information component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.8.5 People

<moving_hazard_component(03)>:=	: People component
<intunti>(id),	: Identifier, id = 03 hex
<intunti>(n),	: Length, n, of component data in bytes
<numag>;	: Number of people involved
m * **<people_component()>**;	: People components

8.3.8.5.1 People component template

<people_component(x)>:=	: People component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.8.5.2 People position

<people_component(00)>:=	: People position
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm10 >;	: Position, TPEG table rtm10

8.3.8.5.3 People problem

<people_component(01)>:=	: People problem
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm20>;	: Problem with people, TPEG table rtm20
m * **<people_problem_component>**;	: People problem components (UAV)

8.3.8.5.3.1 People problem component template

<people_problem_component(x)>:=	: People problem component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.8.5.4 People information

<people_component(02)>:=	: People information
<intunti>(id),	: Identifier, id = 02 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm19>,	: Type of people involved, TPEG table rtm19
m * **<people_information_component()>**;	: People information components (UAV)

8.3.8.5.4.1 People information component template

<people_information_component(x)>:=	: People information component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.9 Security alert

<rtm_component(88)>:=	: Security Alert event class
<intunti>(id),	: Identifier, id = 88 hex
<intunli>(n),	: Length, n, of component data in bytes
<rtm36>,	: Security Alert type, TPEG table rtm36
m * **<security_alert_component()>**;	: Security Alert components (UAV)

8.3.9.1 Security alert component template

<security_alert_component(x)>:=	: Security Alert component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.10 Public transport information

<rtm_component(89)>:=	: Public Transport Information event class
<intunti>(id),	: Identifier, id = 89 hex
<intunli>(n),	: Length n, of component data in bytes
<rtm40>,	: Public Transport type, TPEG table rtm40
<rtm41>,	: Public Transport status, TPEG table rtm41
m * **<public_transport_component()>**;	: Public Transport components (UAV)

8.3.10.1 Public transport component template

<public_transport_component(x)>:=	: Public Transport component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.11 Visibility

<rtm_component(8A)>:=	: Visibility event class
<intunti>(id),	: Identifier, id = 8A hex
<intunli>(n),	: Length, n, of component data in bytes
m * **<visibility_component()>**;	: Visibility components

8.3.11.1 Visibility component template

<visibility_component(x)>:=	: Visibility component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.11.2 Obscurity

<visibility_component(00)>:=	: Obscurity
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm17>,	: Obscurity problem, TPEG table rtm17
<intunti>,	: Visibility distance (in metres * 10)
m * **<obscurity_component()>**;	: Obscurity components (UAV)

8.3.11.2.1 Obscurity component template

<obscurity_component(x)>:=	: Obscurity component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.11.3 Visual acuity

<visibility_component(01)>:=	: Visual acuity
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm13>,	: Visual acuity problem, TPEG table rtm13
m * **<visual_acuity_component()>**;	: Visual acuity components (UAV)

8.3.11.3.1 Visual acuity component template

<visual_acuity_component(x)>:=	: Visual acuity component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.11.4 Lighting

<visibility_component(02)>:=	: Lighting
<intunti>(id),	: Identifier, id = 02 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm14>,	: Lighting problem, TPEG table rtm14
m * **<lighting_component()>**;	: Lighting components (UAV)

8.3.11.4.1 Lighting component template

<lighting_component(x)>:=	: Lighting component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.11.5 Length affected

<visibility_component(03)>:=	: Length affected
<intunti>(id),	: Identifier, id = 03 hex
<intunti>(n),	: Length, n, of component data in bytes
<intunli>;	: Length of road affected (in metres * 10)

8.3.12 Weather

<rtm_component(8B)>:=	: Weather event class
<intunti>(id),	: Identifier, id = 8B hex
<intunli>(n),	: Length, n, of component data in bytes
m * **<weather_component()>**;	: Weather components

8.3.12.1 Weather component template

<weather_component (x)>:=	: Weather component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.12.2 Precipitation

<weather_component(00)>:=	: Precipitation
<intunti>(id),	: Identifier, id = 00 hex
<intunti>(n),	: Length, n, of component data in bytes
<rtm31>,	: General magnitude, TPEG table rtm31
<rtm29>,	: Precipitation problem, TPEG table rtm29
m * **<precipitation_component()>**;	: Precipitation components (UAV)

8.3.12.2.1 Precipitation component template

<precipitation_component(x)>:=	: Precipitation component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.12.3 Wind

<weather_component(01)>:=	: Wind
<intunti>(id),	: Identifier, id = 01 hex
<intunti>(n),	: Length, n, of component data in bytes
<intunti>,	: Wind speed, m/s
<rtm30>,	: Wind problem, TPEG table rtm30
m * **<wind_component()>**;	: Wind components (UAV)

8.3.12.3.1 Wind component template

<wind_component(x)>:=	: Wind component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.12.4 Temperature

<weather_component(02)>:=	: Temperature
<intunti>(id),	: Identifier, id = 02 hex
<intunti>(n),	: Length, n, of component data in bytes
<intsiti>,	: Temperature (in degrees Celsius)
m * **<temperature_component()>**;	: Temperature components (UAV)

8.3.12.4.1 Temperature component template

<temperature_component(x)>:=	: Temperature component template
<intunti>(id),	: Identifier, id = x hex
<intunti>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.13 Diversion advice

<rtm_component(8C)>:=	: Diversion Advice event class
<intunti>(id),	: Identifier, id = 8C hex
<intunli>(n),	: Length, n, of component data in bytes
m * **<diversion_advice_component()>**;	: Diversion advice components

8.3.13.1 Diversion advice component template

<diversion_advice_component(x)>:=	: Diversion advice component template
<intunti>(id),	: Identifier, id = x hex
<intunli>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.13.2 Vehicle information

<diversion_advice_component(00)>:=	: Vehicle information
<intunti>(id),	: Identifier, id = 00 hex
<intunli>(n),	: Length, n, of component data in bytes
<rtm01>(type),	: Type of vehicles involved, TPEG table rtm01
<intunti>(subtype),	: Subtype of vehicles involved, see table in 8.4.3
m * <**vehicle_information_component**()>;	: Vehicle information components (UAV)

8.3.13.3 Regulation

<diversion_advice_component(01)>:=	: Regulation
<intunti>(id),	: Identifier, id = 01 hex
<intunli>(n),	: Length, n, of component data in bytes
<rtm45>,	: Regulation, TPEG table rtm45
<numag>;	: Regulation quantifier

8.3.13.4 Current vehicle position

<diversion_advice_component(02)>:=	: Current vehicle position
<intunti>(id),	: Identifier, id = 02 hex
<intunli>(n),	: Length, n, of component data in bytes
<rtm10>;	: Position, TPEG table rtm10

8.3.13.5 Advice

<diversion_advice_component(03)>:=	: Advice
<intunti>(id),	: Identifier, id = 03 hex
<intunli>(n),	: Length, n, of component data in bytes
<rtm47>,	: Condition status, TPEG table rtm47
<rtm35>,	: Diversion advice type, TPEG table rtm35
m * <advice_component()>;	: Advice Components

8.3.13.5.1 Advice component template

<advice_component(x)>:=	: Advice component template
<intunti>(id),	: Identifier, id = x hex
<intunli>(n),	: Length, n, of component data in bytes
n * **<byte>**;	: Component data

8.3.13.5.2 Routing information

<advice_component(00)>:=	: Routing information
<intunti>(id),	: Identifier, id = 00 hex
<intunli>(n),	: Length, n, of component data in bytes
m * **<routing_component()>**;	: Routing components

8.3.13.5.2.1 Routing component template

<routing_component(x)>:=	: Routing component template
<intunti>(id),	: Identifier, id = x hex
<intunli>(n),	: Length, n, of component data in bytes
n * <**byte**>;	: Component data

8.3.13.5.2.2 Routing TPEG-Loc container

<routing_component(00)>:=	: Routing TPEG-Loc container
<intunti>(id),	: Identifier, id = 00 hex
<intunli>(n),	: Length, n, of component data in bytes
<tpeg_loc_container>,	: TPEG-Loc (CEN ISO/TS 18234-6)

8.3.13.5.2.3 For distance

< routing_component(01)>:=	: For distance
<intunti>(id),	: Identifier, id = 01 hex
<intunli>(n),	: Length, n, of component data in bytes
<intunli>;	: Length of road (in metres * 10)

8.4 RTM application primitives

8.4.1 TPEG tables – rtm01 to rtm50

Table 2 — TPEG tables – rtm01 to rtm50

<rtm01>:= **<intunti>**;	: Vehicle types : TPEG table rtm01
<rtm02>:= **<intunti>**;	: Works vehicles : TPEG table rtm02
<rtm03>:= **<intunti>**;	: Vehicle problems : TPEG table rtm03
<rtm04>:= **<intunti>**;	: Various activities : TPEG table rtm04
<rtm05>:= **<intunti>**;	: Pedal cycle types : TPEG table rtm05

<rtm06>:= **<intunti>**;	: Emergency vehicle types : TPEG table rtm06
<rtm07>:= **<intunti>**;	: Car types : TPEG table rtm07
<rtm08>:= **<intunti>**;	: Vehicle and trailer types : TPEG table rtm08
<rtm09>:= **<intunti>**;	: Light goods vehicle : TPEG table rtm09
<rtm10>:= **<intunti>**;	: Position : TPEG table rtm10
<rtm11>:= **<intunti>**;	: Heavy goods vehicle types : TPEG table rtm11
<rtm12>:= **<intunti>**;	: Object problem : TPEG table rtm12
<rtm13>:= **<intunti>**;	: Acuity problem : TPEG table rtm13
<rtm14>:= **<intunti>**;	: Lighting condition : TPEG table rtm14
<rtm15>:= **<intunti>**;	: Marking condition : TPEG table rtm15
<rtm16>:= **<intunti>**;	: Abnormal vehicle type : TPEG table rtm16
<rtm17>:= **<intunti>**;	: Obscurity problem : TPEG table rtm17
<rtm18>:= **<intunti>**;	: Surface condition : TPEG table rtm18
<rtm19>:= **<intunti>**;	: People type : TPEG table rtm19
<rtm20>:= **<intunti>**;	: People problem : TPEG table rtm20

<rtm21>:=	: Animal type
<intunti>;	: TPEG table rtm21
<rtm22>:=	: Animal size
<intunti>;	: TPEG table rtm22
<rtm23>:=	: Animal problem
<intunti>;	: TPEG table rtm23
<rtm24>:=	: Activity type
<intunti>;	: TPEG table rtm24
<rtm25>:=	: Fair type
<intunti>;	: TPEG table rtm25
<rtm26>:=	: Public gathering type
<intunti>;	: TPEG table rtm26
<rtm27>:=	: National or state occasion
<intunti>;	: TPEG table rtm27
<rtm28>:=	: Concert and cultural event
<intunti>;	: TPEG table rtm28
<rtm29>:=	: Precipitation problem
<intunti>;	: TPEG table rtm29
<rtm30>:=	: Wind problem
<intunti>;	: TPEG table rtm30
<rtm31>:=	: General magnitude
<intunti>;	: TPEG table rtm31
<rtm32>:=	: Road assistance type
<intunti>;	: TPEG table rtm32
<rtm33>:=	: Road assistance status
<intunti>;	: TPEG table rtm33
<rtm34>:=	: Network Performance
<intunti>;	: TPEG table rtm34
<rtm35>:=	: Diversion advice type
<intunti>;	: TPEG table rtm35

<rtm36>:=	: Security alert
<intunti>;	: TPEG table rtm36
<rtm37>:=	: Roadside services type
<intunti>;	: TPEG table rtm37
<rtm38>:=	: Roadside services status
<intunti>;	: TPEG table rtm38
<rtm39>:=	: Adhesion condition
<intunti>;	: TPEG table rtm39
<rtm40>:=	: Public transport type
<intunti>;	: TPEG table rtm40
<rtm41>:=	: Public transport status
<intunti>;	: TPEG table rtm41
<rtm42>:=	: Traffic control equipment
<intunti>;	: TPEG table rtm42
<rtm43>:=	: Traffic control equipment status
<intunti>;	: TPEG table rtm43
<rtm44>:=	: Sports event type
<intunti>;	: TPEG table rtm44
<rtm45>:=	: Regulation
<intunti>;	: TPEG table rtm45
<rtm46>:=	: Unverified Information
<intunti>;	: TPEG table rtm46
<rtm47>:=	: Condition_status
<intunti>;	: TPEG table rtm47
<rtm48>:=	: Motor cycle type
<intunti>;	: TPEG table rtm48
<rtm49>:=	: Restriction
<intunti>;	: TPEG table rtm49
<rtm50>:=	: Roadworks
<intunti>;	: TPEG table rtm50

8.4.2 TPEG tables (rtm01 to rtm50) indexing

The TPEG tables (rtm) numbers and code values have no order-significance and only have number values randomly assigned during the development process. In order to aid navigation of these TPEG tables, Table 3 and Table 4 provide an "internal index" to the tables, firstly in name order and secondly in number order.

Table 3 — TPEG tables (rtm01 to rtm50 Version 3.0) – ordered by name

Description	Table	Description	Table
abnormal_vehicle_type	16	position	10
activity_type	24	precipitation_problem	29
acuity_problem	13	public_gathering_type	26
adhesion_condition	39	public_transport_status	41
animal_problem_type	23	public_transport_type	40
animal_size	22	regulation_type	45
animal_type	21	restriction_type	49
car_type	07	roadside_assistance_status	33
concert_cultural_type	28	roadside_assistance_type	32
condition_status	47	roadside_services_status	38
diversion_advice_type	35	roadside_serv_type	37
emergency_vehicle_type	06	roadworks_type	50
fair_type	25	security_alert_type	36
general_magnitude	31	sports_event_type	44
heavy_goods_vehicle_type	11	surface_condition	18
light_goods_vehicle_type	09	traffic_control_status	43
lighting_problem	14	traffic_control_type	42
marking_condition	15	unverified_information	46
motor_cycle_type	48	various_type	04
national_events_type	27	vehicle_trailer_type	08
network_performance_status	34	vehicle_problem_type	03
object_problem	12	vehicle_type	01
obscurity_problem	17	wind_problem	30
pedal_cycle_type	05	works_vehicle_type	02
people_problem_type	20	*~ end of version 3.0 tables ~*	
people_type	19		

Table 4 — TPEG tables (rtm01 to rtm50 Version 3.0) – ordered by table number

Table	Description
01	vehicle_type
02	works_vehicle_type
03	vehicle_problem_type
04	various_type
05	pedal_cycle_type
06	emergency_vehicle_type
07	car_type
08	vehicle_trailer_type
09	light_goods_vehicle_type
10	position
11	heavy_goods_vehicle_type
12	object_problem
13	acuity_problem
14	lighting_problem
15	marking_condition
16	abnormal_vehicle_type
17	obscurity_problem
18	surface_condition
19	people_type
20	people_problem_type
21	animal_type
22	animal_size
23	animal_problem_type
24	activity_type
25	fair_type
26	public_gathering_type
27	national_events_type
28	concert_cultural_type
29	precipitation_problem
30	wind_problem
31	general_magnitude
32	roadside_assistance_type
33	roadside_assistance_status
34	network_performance_status
35	diversion_advice_type
36	security_alert_type
37	roadside_serv_type
38	roadside_services_status
39	adhesion_condition
40	public_transport_type
41	public_transport_status
42	traffic_control_type
43	traffic_control_status
44	sports_event_type
45	regulation_type
46	unverified_information
47	condition_status
48	motor_cycle_type
49	restriction_type
50	roadworks_type
	~ end of version 3.0 tables ~

8.4.3 TPEG table rtm01 – vehicle subtype cross-reference table

The vehicle subtype can only be decoded given the code for the vehicle type. Therefore a cross-reference table exists to inform a client decoder what TPEG table it should use to decode the vehicle subtype, given the code for the vehicle type.

NOTE Some vehicle types do not have a vehicle subtype table defined. In these cases a code (one byte, set to 0) for the vehicle subtype must be transmitted; however, TPEG-decoders should ignore the code for the vehicle subtype.

Table 5 — TPEG table rtm01 – vehicle subtype cross reference table

Code	CEN-English 'Word'	Entry in related TPEG table (subtype), table name:
0	unknown	(subtype = 0), *no TPEG table available*
1	car	TPEG table rtm07(...), car_type
2	light goods vehicle	TPEG table rtm09(...), light_goods_vehicle_type
3	heavy goods vehicle	TPEG table rtm11(...), heavy_goods_vehicle_type
4	public transport vehicle	TPEG table rtm40(...), public_transport_type
5	pedal cycle	TPEG table rtm05(...), pedal_cycle_type
6	emergency vehicle	TPEG table rtm06(...), emergency_vehicle_type
7	works vehicle	TPEG table rtm02(...), works_vehicle_type
8	abnormal vehicle	TPEG table rtm16(...), abnormal_vehicle_type
9	vehicle with trailer	TPEG table rtm08(...), veh_trailer_type
10	high-sided vehicle	(subtype = 0), *no TPEG table available*
11	minibus	(subtype = 0), *type is equal to subtype, see TPEG table rtm40*
12	taxi	(subtype = 0), *type is equal to subtype, see TPEG table rtm40*
13	tram	(subtype = 0), *type is equal to subtype, see TPEG table rtm40*
14	trolley-bus	(subtype = 0), *type is equal to subtype, see TPEG table rtm40*
15	train	(subtype = 0), *type is equal to subtype, see TPEG table rtm40*
16	post bus	(subtype = 0), *type is equal to subtype, see TPEG table rtm40*
17	school bus	(subtype = 0), *type is equal to subtype, see TPEG table rtm40*
18	military vehicle	(subtype = 0), *no TPEG table available*
19	motorcycle	TPEG table rtm48(...), motor_cycle_type
20	sledge	(subtype = 0), *no TPEG table available*

8.4.4 TPEG table rtm24 - activity subtype cross-reference table

The activity subtype can only be decoded given the code for the activity type. Therefore a cross-reference table exists to inform a client decoder what TPEG table it should use to decode the vehicle subtype, given the code for the vehicle type.

NOTE Some activity types do not have an activity subtype table defined. In these cases a code (one byte, set to 0) for the activity subtype must be transmitted; however, TPEG-decoders should ignore the code for the activity subtype.

Table 6 — TPEG table rtm24 - activity subtype cross-reference table

Code	CEN-English 'Word'	Entry in related TPEG *table* (subtype), table name
0	unknown	(subtype = 0), *no TPEG table available*
1	various	TPEG table rtm04(…), various_type
2	fair	TPEG table rtm25(…), fair_type
3	public gathering	TPEG table rtm26(…), public_gathering_type
4	sports event	TPEG table rtm44(…), sports_event_type
5	national event	TPEG table rtm27(…), national_events_type
6	concerts & cultural	TPEG table rtm28(…), concert_cultural_type

8.4.5 TPEG tables – structure and semantics

TPEG tables provide a list of the CEN-English ‘word’ with associated code value, and additionally comments, and where helpful, examples are given. The CEN-English ‘word’ describes a single entity as far as possible with a single word, but it is necessary to sometimes use a short phrase to describe the entity, e.g. light goods vehicle; nevertheless, TPEG tables are in essence tables of singular words. Where the coding allows multiplicity of the entity, then the CEN-English ‘word’ shall be singular. In other cases there are a number of logical plurals, e.g. roadworks, which are commented accordingly.

The key principle for the use of the CEN-English ‘word’ code value is that all client devices shall be designed to make their own assessment of the context and multiplicity, in order to deliver a semantically acceptable message in the chosen display language.

8.4.6 TPEG tables (rtm01 to rtm50 Version 3.0)

Table 7 — TPEG table rtm01: vehicle_type

Code	CEN-English 'Word'	Comments	Examples
0	unknown		
1	car		
2	light goods vehicle		e.g. van
3	heavy goods vehicle		e.g. lorry
4	public transport vehicle		e.g. bus and coach
5	pedal cycle		e.g. bicycle
6	emergency vehicle		
7	works vehicle		e.g. tractor and diggers
8	exceptional size vehicle		e.g. wide load
9	vehicle with trailer	including caravan	
10	high-sided vehicle	sensitive to bridge heights and lateral winds	
11	minibus		
12	taxi		
13	tram		
14	trolley-bus		
15	train		
16	post bus		
17	school bus		
18	military vehicle		
19	motorcycle	including moped	
20	sledge	animal drawn snow vehicle	
..	***~ end of version 3.0 ~***		
..			
255	vehicle	- the table default word -	

Table 8 — TPEG table rtm02: works_vehicle_type

Code	CEN-English 'Word'	Comments	Examples
0	unknown		
1	gritting vehicle		
2	snowplough		
3	salting vehicle		
4	white-lining vehicle		
5	resurfacing vehicle		
6	road roller	also known as: "steam roller"	
7	mobile crane		
8	construction vehicle		
9	farm tractor		
10	farm tractor and plough		
11	farm tractor and trailer		
12	combine harvester		
13	combine harvester and trailer		
14	track-laying vehicle	also known as: "bulldozer"	
15	road surface testing vehicle		
16	water tanker		
17	drain-cleaning vehicle		
18	road sweeping vehicle		
19	waste collection vehicle		
20	grass cutting machine		
21	tree cutting machine		
22	hedge cutting machine		
23	snow-blower		
..	***~ end of version 3.0 ~***		
..			
255	works vehicle	- the table default word -	

Table 9 — TPEG table rtm03: vehicle_problem_type

Code	CEN-English 'Word'	Comments	Examples
0	unknown		
1	rescue work		
2	jack-knifed		
3	overturned		
4	on fire		
5	spun round		
6	spillage		
7	driver on wrong carriageway	also known as a: "ghost driver"	
8	broken down		
9	shed load		
10	unlit		
11	brake failure		
12	stuck		
13	abandoned		
14	in convoy		
15	slow moving		
16	dangerously driven		
17	dangerous load		
18	prohibited		
19	slowing	also known as: "rubber-necking"	e.g. drivers observing accident
20	chasing		
21	excessive speed		
22	accident		
..	***~ end of version 3.0 ~***		
..			
255	vehicle problem	- the table default word -	

Table 10 — TPEG table rtm04: various_type

Code	CEN-English 'Word'	Comments	Examples
0	unknown		
1	demolition		
2	space launch		
3	eclipse		
4	blasting work		
5	maintenance work		
..	***~ end of version 3.0 ~***		
..			
255	activity	- the table default word -	

Table 11 — TPEG table rtm05: pedal_cycle_type

Code	CEN-English 'Word'	Comments	Examples
0	unknown		
1	motor-assisted pedal cycle		
2	tricycle		
3	tandem cycle		
4	pedal cycle and trailer		
5	pedal cycle and side-car		
6	unicycle		
7	kick sledge	e.g. in Swedish: "spark"	
..	***~ end of version 3.0 ~***		
..			
255	pedal cycle	- the table default word -	

Table 12 — TPEG table rtm06: emergency_vehicle_type

Code	CEN-English 'Word'	Comments	Examples
0	unknown		
1	ambulance		
2	fire engine		
3	police car		
4	breakdown recovery vehicle		
5	fire engine with escape ladder		
6	fire engine with hydraulic platform		
7	fire engine with water tank		
8	vehicle removal crane		
9	incident control vehicle		
10	salvage vehicle		
..	***~ end of version 3.0 ~***		
..			
255	emergency vehicle	- the table default word -	

Table 13 — TPEG table rtm07: car_type

Code	CEN-English 'Word'	Comments	Examples
0	unknown		
1	small car		
2	family car		
3	large car		
4	multi-purpose-vehicle	also known as: people carrier or space wagon	e.g. Renault Espace
5	limousine	extended length chassis	
6	camper car		
7	large 4-wheel drive vehicle		e.g. off-road capable
..	***~ end of version 3.0 ~***		
..			
255	car	- the table default word -	

Table 14 — table rtm08: vehicle_trailer_type

Code	CEN-English 'Word'	Comments	Examples
0	unknown		
1	car and caravan		
2	light goods vehicle and caravan		
3	heavy goods vehicle and caravan		
4	car and trailer		
5	light goods vehicle and trailer		
6	heavy goods vehicle and trailer		
7	bus and trailer		
..	***~ end of version 3.0 ~***		
..			
255	vehicle and trailer	- the table default word -	

Table 15 — TPEG table rtm09: light_goods_vehicle_type

Code	CEN-English 'Word'	Comments	Examples
0	unknown		
1	small van		
2	small pick-up truck		
3	medium size van	i.e. < 1,5 tonnes	
4	medium size pick-up truck	i.e. < 1,5 tonnes	
5	small motorcaravan	i.e. < 1,5 tonne	
6	vehicle carrying hazardous load		
..	***~ end of version 3.0 ~***		
..			
255	light goods vehicle	- the table default word -	

Table 16 — TPEG table rtm10: position

Code	CEN-English 'Word'	Comments	Examples
0	unknown		
1	driving lane 1		
2	driving lane 2		
3	driving lane 3		
4	driving lane 4		
5	driving lane 5		
6	driving lane 6		
7	driving lane 7		
8	driving lane 8		
9	driving lanes 1 and 2	logical plural	
10	driving lanes 2 and 3	logical plural	
11	driving lanes 3 and 4	logical plural	
12	driving lanes 4 and 5	logical plural	
13	driving lanes 5 and 6	logical plural	
14	driving lanes 6 and 7	logical plural	
15	driving lanes 7 and 8	logical plural	
16	driving lanes 1, 2 and 3	logical plural	
17	driving lanes 2, 3 and 4	logical plural	
18	driving lanes 3, 4 and 5	logical plural	
19	driving lanes 4, 5 and 6	logical plural	
20	driving lanes 5, 6 and 7	logical plural	
21	driving lanes 6, 7 and 8	logical plural	
22	driving lanes 1, 2, 3 and 4	logical plural	
23	driving lanes 2, 3, 4 and 5	logical plural	
24	driving lanes 3, 4, 5 and 6	logical plural	
25	driving lanes 4, 5, 6 and 7	logical plural	
26	driving lanes 5, 6, 7 and 8	logical plural	
27	driving lanes 1, 2, 3, 4 and 5	logical plural	
28	driving lanes 2, 3, 4, 5 and 6	logical plural	
29	driving lanes 3, 4, 5, 6 and 7	logical plural	
30	driving lanes 4, 5, 6, 7 and 8	logical plural	
31	driving lanes 1, 2, 3, 4, 5 and 6	logical plural	
32	driving lanes 2, 3, 4, 5, 6 and 7	logical plural	
33	driving lanes 3, 4, 5, 6, 7 and 8	logical plural	
34	driving lanes 1, 2, 3, 4, 5, 6 and 7	logical plural	
35	driving lanes 2, 3, 4, 5, 6, 7 and 8	logical plural	
36	off-road		
37	all driving lanes	logical plural	
38	central reservation		
39	hard shoulder		
40	service road		
41	local lane		
42	underpass		
43	fly over		

Code	CEN-English 'Word'	Comments	Examples
44	emergency lane		
45	bridge		
46	tunnel		
47	overtaking lane		
48	turning lane		
49	slip road		
50	toll plaza		
51	cycle lane		
52	through traffic lane		
53	filter lane		
54	bend		
55	hilltop		
56	car pool lane		
57	bus lane		
58	slow vehicle lane	also known as: "crawler lane"	
59	verges	logical plural	
60	roadside bank		
61	adjacent to road		
62	opposite carriageway		
63	exit slip road		
64	entry slip road		
65	express lane		
66	lay-by		
67	rest area		
68	service area		
69	around corner	i.e. referring to a 90 degree turn	
70	escape lane		
71	feeder road		
72	left-hand feeder road		
73	right-hand feeder road		
74	dyke		
75	shaded area		
76	sunny area		
77	left-hand turn lane		
78	right-hand turn lane		
79	bus stop		
80	set down area		
81	low lying area		
82	low altitude route		
83	high altitude route		
84	ascending route		
85	descending route		
86	around the bend	i.e. referring to a large radius corner	

Code	CEN-English 'Word'	Comments	Examples
87	weigh station		
88	north bound carriageway		
89	north-east bound carriageway		
90	east bound carriageway		
91	south-east bound carriageway		
92	south bound carriageway		
93	south-west bound carriageway		
94	west-bound carriageway		
95	north-west bound carriageway		
96	clockwise carriageway		
97	anti-clockwise carriageway		
98	junction		
99	left lane	for use in a left hand drive country	See 4.3.2
100	left lane	for use in a right hand drive country	See 4.3.2
101	right lane	for use in a left hand drive country	See 4.3.2
102	right lane	for use in a right hand drive country	See 4.3.2
103	middle lane		
104	one lane		
105	two lanes	logical plural	
106	three lanes	logical plural	
..	***~ end of version 3.0 ~***		
..			
255	on route	- the table default word -	

Table 17 — TPEG table rtm11: heavy_goods_vehicle_type

Code	CEN-English 'Word'	Comments	Examples
0	unknown		
1	medium size lorry	i.e. < 7,5 tonnes	
2	large lorry	i.e. > 7,5 tonnes	
3	medium lorry with close-coupled trailer		
4	large lorry with close-coupled trailer		
5	articulated lorry		
6	lorry cabin without articulated trailer		
7	car transporter		
8	low loader		
9	tanker		
10	vehicle carrying hazardous load		
11	large motorcaravan	i.e. > 1,5 tonne	e.g. recreational vehicle, also known as an "RV"
..	***~ end of version 3.0 ~***		
..			
255	heavy goods vehicle	- the table default word -	

Table 18 — TPEG table rtm12: object_problem

Code	CEN-English 'Word'	Comments	Examples
0	unknown		
1	various		
2	fallen tree		
3	fallen power line		
4	fallen telephone pole		
5	shed tyre tread		
6	shed wood		
7	shed opaque sheet		
8	shed ballast or sand		
9	loose road cones	logical plural	
10	avalanche		
11	mud slide		
12	rock fall		
13	land slide		
14	flood		
15	sewer overflow		
16	debris		
17	shed load		
18	spillage		
..	***~ end of version 3.0 ~***		
..			
255	object	- the table default word -	

Table 19 — TPEG table rtm13: acuity_problem

Code	CEN-English 'Word'	Comments	Examples
0	unknown		
1	sun glare		
2	snow glare		
..	***~ end of version 3.0 ~***		
..			
255	visibility problem	- the table default word -	

Table 20 — TPEG table rtm14: lighting_problem

Code	CEN-English ‘Word’	Comments	Examples
0	unknown		
1	failed lighting	i.e. not working at all	
2	faulty lighting	i.e. working partly or intermittently	
3	temporary lighting		
4	unlit		
..	***~ end of version 3.0 ~***		
..			
255	lighting problem	- the table default word -	

Table 21 — TPEG table rtm15: marking_condition

Code	CEN-English ‘Word’	Comments	Examples
0	unknown		
1	no lane marking		
2	changed lane marking		
3	temporary lane marking	i.e. for roadworks	
4	no reflective marking		e.g. in UK known as “cats-eyes”
5	traffic-calming indication		
6	crash barrier missing		
7	new road marking		
..	***~ end of version 3.0 ~***		
..			
255	road marking advice	- the table default word -	

Table 22 — TPEG table rtm16: abnormal_vehicle_type

Code	CEN-English ‘Word’	Comments	Examples
0	unknown		
1	high-sided		
2	high load		
3	heavy load		
4	wide load		
5	long load		
6	slow moving		
7	very slow moving		e.g. < 20 km/h
..	***~ end of version 3.0 ~***		
..			
255	abnormal vehicle	- the table default word -	

Table 23 — TPEG table rtm17: obscurity_problem

Code	CEN-English 'Word'	Comments	Examples
0	unknown		
1	rain		
2	fog		
3	smoke		
4	sleet		
5	insects	logical plural	
6	dust cloud		
7	insect swarm		
8	sand storm		
9	spray		
10	snow spray		
11	blizzard		
..	***~ end of version 3.0 ~***		
..			
255	visibility problem	- the table default word -	

Table 24 — TPEG table rtm18: surface_condition

Code	CEN-English 'Word'	Comments	Examples
0	unknown		
1	debris		
2	melting tar		
3	subsidence		
4	earthquake		
5	rock fall		
6	uneven surface		e.g. rutted surface
7	pot hole		
8	collapsed sewer		
9	burst water main		
10	frost damaged surface	i.e. in Swedish: tjälskador	e.g. ”spongy” surface
11	loose surface		e.g. from frost thawing
..	***~ end of version 3.0 ~***		
..			
255	road surface problem	- the table default word -	

Table 25 — TPEG table rtm19: people_typ

Code	CEN-English 'Word'	Comments	Examples
0	unknown		
1	children		
2	elderly		
3	handicapped		
4	traffic warden		
5	police officer		
6	fire fighter		
7	paramedic		
8	school children		
9	student		
..	***~ end of version 3.0 ~***		
..			
255	people	- the table default word -	

Table 26 — TPEG table rtm20: people_problem_type

Code	CEN-English 'Word'	Comments	Examples
0	unknown		
1	disorientated		
2	assembling		
3	rioting		
4	arriving		
5	leaving		
6	arriving and leaving		
7	observing	i.e. static	
8	marching		
9	directing traffic		
10	obstructing		
11	queuing		
12	playing		
13	injured		
14	trapped		
..	***~ end of version 3.0 ~***		
..			
255	people	- the table default word -	