



Open Communication Interface for Road Traffic Control Systems
Offene Schnittstellen für die Straßenverkehrstechnik

OCIT-C Center to Center

Data

OCIT-C_Data_V1.2_R1

OCIT Developer Group (ODG)&Partner

OCIT® Registered trade mark of AVT STOYE, Siemens, Stührenberg, SWARCO

OCIT-C Center to Center

Data

Document: OCIT-C_Data_V1.2_R1

Editor: ODG & Partner

Contact: www.ocit.org

Copyright © 2016 ODG. Subject to modifications. Documents with a more recent version or revision level replace all contents of the previous versions.

Contents

Introduction	7
1.1 Terms and Abbreviations	10
2 Definitions of objects and schemas.....	12
2.1 Object.....	12
2.2 Object types	12
2.3 Object identification.....	12
2.3.1 Object identification	13
2.3.2 Object filter.....	14
2.4 Global structures (global.xsd)	14
2.4.1 Element Description	15
2.4.2 Element Rel_Ids	15
2.4.3 Element Location	16
2.4.4 State	17
2.4.5 Timeline	18
3 Data catalogue	19
3.1 Traffic messages.....	19
3.2 Traffic data	20
3.2.1 Subsection, sections, traffic areas.....	21
3.2.2 Detectors and detector groups	22
3.2.2.1 Object type: TrafficData_detector_Description.....	22
3.2.2.2 Object type: TrafficData_detector_currentValue	23
3.2.2.3 Object type: TrafficData_detectorGroup_Description.....	24
3.2.2.4 Object type: TrafficData_detectorGroup_currentValue	24

3.2.2.5	Object type: TrafficData_detectorGroup_calculatedValue	24
3.2.2.6	Object type: TrafficData_trafficSubSection_Description	24
3.2.2.7	Object type: TrafficData_trafficSection_Description	25
3.2.2.8	Object type: TrafficData_trafficArea_Description	25
3.2.2.9	Object type: TrafficData_trafficSubSection_accumulatedValue.....	25
3.2.2.10	Object type: TrafficData_trafficSection_accumulatedValue	25
3.2.2.11	Object type: TrafficData_trafficArea_accumulatedValue.....	25
3.3	Parking data.....	26
3.4	Weather and environmental data.....	27
3.5	Cameras.....	27
3.6	Situations and strategies.....	29
3.7	Operating messages.....	30
3.8	Signs	33
3.8.1	Data model	33
3.8.2	Complex sign contents	35
3.9	Public transport passenger information	37
3.10	Traffic signal systems	37
3.10.1	Pre-defined commands - obsolete	38
3.10.2	Intersection Features.....	39
3.10.3	IntersectionCommand (modelled switching operations)	40
3.10.4	ControlMethodParameter	42
3.10.5	TSS status (IntersectionStatus).....	42
3.10.6	SGTimes.....	43
3.11	TSS raw data	43

3.11.1	Detector edges	44
3.11.2	Signal group states	46
3.11.3	Digital outputs in the traffic signal controller	47
3.11.4	User program values (AP-values)	48
3.11.4.1	Setting of requestable AP values	50
3.11.5	Extended PT telegram	50
3.11.6	Data from individual detectors	51
3.12	TSS supply data	52
3.13	Binary container	52
3.14	Project-specific extensions	53
3.14.1	Compatibility rules	53

Document history

Version Issue	Date	Distribution List	Comment
V1.1_R1	30.10.2014	PUBLIC	Version 1.1 Issue 1
V1.2_R1	16.12.2016	PUBLIC	Version 1.2 Issue 1 Referencing changed from OCIT-I to OCIT-C 2.3.1: OITD numbers are to be written in IP notation 2.3.2: Object filters are a standard function of the servers 3.11.4: Table with standardised AP values.

Introduction

OCIT-C stands for Open Communication Interface for Road Traffic Control Systems - Center to Center. OCIT-C covers the communication functions between central traffic control and traffic guidance systems:

- Traffic control centers and traffic management centers (urban, regional, interregional)
- Traffic engineer work place with traffic control centers
- Parking guidance systems, parking facility systems
- Roadworks management systems
- Local internet users (city info online)

The definition and maintenance of the OCIT-C interface is carried out by the ODG and their partners.

OCIT-C provides a standard which perfectly complements OCIT-O. All requirements for traffic control up to the overarching traffic management are covered by using OCIT-C and OCIT-O for communication from control centers to field devices.

OCIT-C is therefore geared towards practical requirements. With its low implementation costs, it's also suitable for use in solutions with small budgets.

The featured properties of OCIT-C are:

- Exchange protocol with a simple request-response communication pattern (direct querying of data) based on the SOAP standard.
- Definition of a comprehensive data model in the process data area containing all partial sections of the traffic control and traffic guidance, use of OCIT-C supply data models for TSS.
- System integration and desired adaptations are governed in advance by project planning.
- Conformity tests for the protocol are carried out in a test environment provided at www.ocit.org. Tests of entire implementations (protocol and data contents) are carried out on a project-specific basis.
- Expansions to the DATEX II components are possible based on your project requirements.

The OCIT-C interface is open and can be used in various systems, predominantly in the road traffic technology sector. The aim of this document is to describe the standardised data that can be transferred via OCIT-C. The aim of this document is not to describe the protocol. This is described in the document "OCIT-C protocol".

The following communication modules are currently standardized:

- **Traffic messages**
This communication module is used to transfer messages regarding roadworks, events, and faults with the following attributes: Location, time period, daily validity, status, classification, impact, e.g. capacity reduction, restrictions, e.g. speed or weight, referencing regarding overlapping messages, type of recording (manual or automatic) and free text.
- **Traffic data**
This communication module is used to transfer measurements or derived values from various detection devices.
- **Parking data**
This communication module is used to transfer data from parking facilities: Name, location, short and long term parks, occupancy, vacant parking spots, forecasted values (trend) and status of the parking facility.
- **Weather and environmental data**
This communication module is used to transfer data from respective detection facilities:
 - Sensor type with the attributes location, point or road related measurement values,
 - Data corresponding to the TLS: Precipitation (intensity, likelihood), temperature (air, ground), wind, humidity, air pressure, visibility, state of the lane surface, sunrise, sunset, brightness, and
 - additional data: Radiation balance, cloud cover level, ozone, sulphur dioxide, benzene, soot, carbon monoxide, dust, weather forecast.
- **Cameras**
This communication module is used to transfer data to control monitoring cameras and to transfer images.
- **Situations and strategies**
This communication module is used to transfer the descriptions of traffic situations or control strategies and input for strategic traffic control measures.
- **Operating messages**
This communication module is used to transfer operating messages from traffic equipment, such as from traffic signal systems.
- **Signs**
This communication module is used to transfer data for controlling segment and full matrix signs: Status messages from the sign (status, time stamp, temperature and contents) commands to the sign (contents). A differentiation of the contents is made through: Prismatic signs, full matrix displays, display points and their individual displays. For full matrix displays, transferred are: the complete layout (text and image positions, colors, font etc.) and the contents of the layout elements.
- **Public transport passenger information**
This communication module is used to transfer road, line, or public transport stop related data of the public transport network: Location, VDV passenger information, public transport stop and line, type of PT vehicle (bus, tram, etc.), expected arrival, actual arrival at the public transport stop, status of the PT vehicle.

- **Traffic signal systems (TSS)**
This communication module is used to transfer data as input for controlling traffic signal systems: Description, actual status, predefined switching operations, parameters for traffic and cycle related signaling.
- **TSS raw data**
This communication module is used to efficiently transfer a large amount of confidential values from traffic signal light systems: Detector edges, signal group statuses, digital outputs, user program values, PT data (additional R09 telegram), individual detector data.
- **TSS supply data**
This communication module is used for supplying data to traffic signal systems.
- **Binary container**
This communication module is used to transfer binary encoded data. Data with a transferred description (name type) identifies the type of data in the container and thus makes it possible for the recipient to process.
- **Project-specific extensions**
of communication modules are possible, however must correspond with certain rules (see chapter 3.14).

Note: All data, including big data are transferred using the OCIT-C protocol.

1.1 Terms and Abbreviations

Term / Abbreviation	Description
AP	User program
Client	A program which wishes to use services offered by other (servers) and actively opens them to do so.
DATEX II	Specifications of Technical Committee 278 of the European Committee for Standardization (CEN) for the exchange of traffic-related data between traffic control centers.
FTP	File Transfer Protocol, a network protocol for transferring files
http	HyperText Transfer Protocol, a protocol for transferring data over a network.
TSS	Traffic signal system
Method	The algorithms assigned to a class of objects. Also used as a synonym for function, procedure, command, action.
PT	Public Transport
OCIT	Open Communication Interface for Road Traffic Control Systems
OCIT-C	Open Communication Interface for Road Traffic Control Systems - Center to Center. OCIT-C covers the communication functions between central traffic control and traffic guidance systems.
OCIT-O	OCIT outstations Interface between traffic control centres and traffic signal controllers for controlling and supplying the traffic signal controllers.
ODG	OCIT Developer Group
OSI	Open Systems Interconnection Reference Model, a communication model of the International Organization for Standardization (ISO) for communication protocols in computer networks.
OTS 2	Open Traffic Systems, Version 2
Server	A program that offers certain services and passively waits on incoming calls (from clients) to do so.
SOAP	Simple Object Access Protocol, it is a protocol which enables data to be exchanged between systems. SOAP uses the "Remote Procedure Call", through which it enables the functions in other computers to be called.

	See http://www.w3.org/TR/SOAP
SSL	Secure Socket Layer.
SoapServer-Interface	Soap and Protocolmanager on the server side
SoapClient-Interface	Soap and Protocolmanager on the client side
Protocolmanager	Protocol layer used for implementing commands in the buffer
TLS	Technical delivery terms for roadway stations. The TLS are a standard for the structure of traffic control systems on major German Federal highways. Editor: German Federal Highway Research Institute
TCP / IP	Transmission Control Protocol / Internet Protocol, a family of network protocols for the Internet.
VDV	Verband Deutscher Verkehrsunternehmen (Association of German Transportation Companies)
WSDL	Web Services Description Language, a platform / programme language and protocol-independent description language for network services (web services) for exchanging messages based on XML.
XML	Extensible Markup Language, a markup language for presenting structured data in the form of text. XML is used among other things for a platform and implementation-independent exchange of data between computer systems. An XML document is made up of text characters, in the most basic case in ASCII coding, and is therefore machine-readable. It does not contain binary data. The XML specification is published by the World Wide Web Consortium (W3C) as a recommendation.
XSD	XML schema, a recommendation of the World Wide Web Consortium (W3C) for defining structures for XML documents. The structure is described in the form of an XML document. Furthermore, it supports a large number of data types. The XSD schema language describes data types, individual XML schema instances (documents) and groups of such instances. A specific XML schema is called an XSD (XML Schema Definition) and the file usually has the ending ".xsd".

2 Definitions of objects and schemas

2.1 Object

An object is a unit that can be clearly identified, which the data to be exchanged relates to.

Example: a specific unit, such as a TSS, detector, parking building, traffic message

2.2 Object types

An object type denotes a requestable or configurable unit between the interface partners. It is not possible to request partial quantities of the object types.

In general, data from a data type (e.g. detector values) are divided into the following partial aspects. These partial aspects constitute the object types below - i.e. requestable units:

- Descriptive data are an object type
- Dynamic data are one or more object types
- Commands are a separate object type
- Object types are defined in the schema definitions. The annotations of the root elements of the data type to be transferred document the identification string of the object type in the form *objecttype: <Objecttypeidentification>*.

2.3 Object identification

Object identification occurs over one to three levels:

- Object type
- Source information (source of the object). This level is optional.
- Identifier (id)

The actual identifier must be unique within the object type. If the source is used, then the identifier must be unique within the object type and the source.

The identifiers of an object are generally negotiated, common knowledge between the interface partners (e.g. detectors, signs, parking garages).

Other object types (e.g. traffic messages) must be generated and deleted dynamically via the interface. This way it is not possible to pre-define the identifier for this type of objects. For the traffic messages, the identifiers and source information are used for unique identification.

2.3.1 Object identification

The object identification of the OCIT-C in terms of the identifier is open for all data sources. Therefore, the IDs can be freely definable in principle.

OCIT-I/O envisages certain classification criteria for TSS-related data. In order to get a clear illustration of the classification criteria for TSS-related data, it provides a regulation for the formation of OCIT-C IDs, derived from the classification criteria of OCIT-I/O. This illustration is especially useful when using OCIT-O as a protocol for connecting external systems.

The ID of a measurement value in the Identifier/Ident field, for objects whose measurement is clearly defined by the object type, e.g. traffic data (→ Chapter 3.2), traffic signal systems (→ Chapter 3.10) and raw data (→ Chapter 3.11), should be structured as follows:

- J<SystemNo>_<SubsystemNo>_<UnitNo >_<ObjectNo>

Whereby "J" stands for the special structure of the primary key. The mark "J" may only be used in conjunction with the format described here.

The address of a measurement value in the Id field is not used here and it usually repeats the value of the field Identifier/Ident.^{1 2}

For AP values (→ Chapter 3.11.4), the ID in the Identifier/Ident field should be structured as follows:

- J<SystemNo>_<SubsystemNo>_<UnitNo>_<OITD-Number>_<ObjectNo>

The identifiers <SystemNo>, <SubsystemNo>, <UnitNo> and <ObjectNo> have been taken from the "OCIT-I" standard, as well as the term of the OITD number. The OITD numbers are to be written in IP notation. (57.102)

If there is no object number available or it is a one-off object, "1" is used as the object number.

For systems which neither use the SystemNo nor the SubsystemNo in the primary key, rather only the UnitNo, the following structure is also permissible:

- J<UnitNo_<ObjecNo for traffic data and traffic signal systems
- J<UnitNo>_<OITD-Number>_<ObjectNo> for AP values. The <OITD-Number> should be structured as follows:
- <Member-Number>.<Subnumber>

Examples of a complete address are therefore:

- J1_12_22555_17
for the detector with channel number 17 on intersection 1_12_22555

¹ This for example makes it possible to identify an isolated measurement value (without the protocol part from protokoll.xsd) in log files.

² Alternatively, the data supplier is able to use a local identifier.

- J1_12_22555_57.102_3
for the AP value "102" of the Verkehrs-Systeme AG (Member 57) on intersection 1_12_22555; the AP value has the channel number 3

These examples with the short address form appear as follows:

- J22555_17
- J22555_57.102_3

2.3.2 Object filter

Object filters are a standard server function and must be supported as specified here. When calling data through the client, an optional filter list can be specified using the protocol's "inquireAll" and "get" methods. The objects are addressed in the filter list using an identifier in the same way as for object identification. These identifiers are made up of:

- Source information (source of the object). This level is optional.
- Identifier (ident)

It is permissible to use only one part of the primary key used for the identifier in the Identifier/Ident field, whereby the primary key in the server is filtered starting from links.

Example of a filter identifier for the object type TrafficData_detector_currentValue:

- J1_12_22555
for all detectors on intersection 1_12_22555
- J1_12
for all detectors of subsystem 12 on system 1
- J1_12_22555_17
Only the measurements of the detector with channel number 17 on intersection 1_12_22555

If the short form of the address is used, it is only possible to filter by the unit number:

- J22555
for all detectors on intersection 22555
- J22555_17
Only the measurements of the detector with channel number 17 on intersection 22555

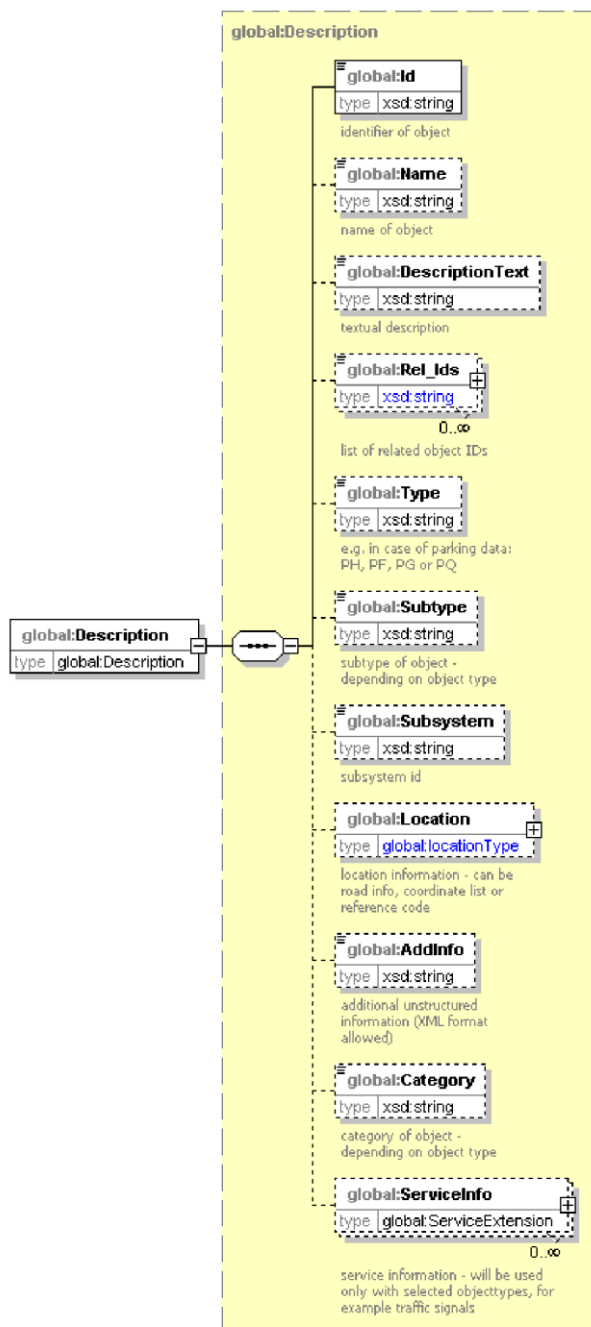
2.4 Global structures (global.xsd)

The exact description of the data model, as well as the elementary description of the attributes and structure elements takes place completely within the individual schema definitions in the form of XML schema definitions (XSD).

The globally used data structures will be defined in global.xsd. A brief description of the structure of global.xsd will be given here.

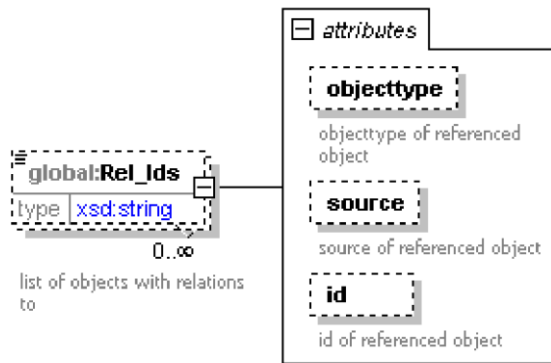
2.4.1 Element Description

All described data are entered into this structure. This data structure is referenced through the domain-specific object types.



2.4.2 Element Rel_Ids

The structure element `Rel_Ids` allows a description of the referencing or the topology under objects. Therefore, these structure elements refer to other objects and therefore include the complete identification comprising object type, source and ID.

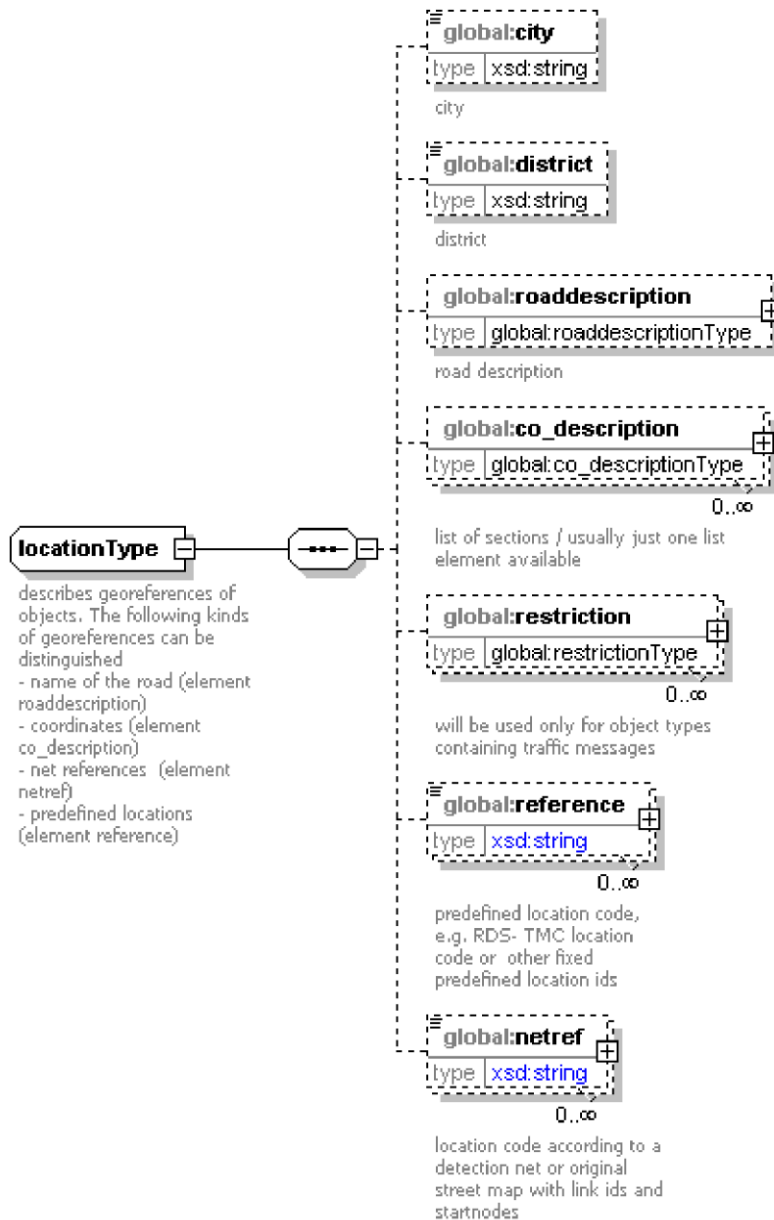


Name	Type	Use	Description
objecttype	xsd:string	optional	Object type of the referenced object
source	xsd:string	optional	Source of the referenced object
ID	xsd:string	optional	Identification of the referenced object

2.4.3 Element Location

This structure element allows objects to be geo-referenced. This can take place in different ways:

- Text description of the roads and crossing description
- Coordinates
- Pre-defined geo-references (e.g. TMC locations)
- Reference to networks (link-ids)



Each object description contains this structure. The occupancy must be matched in the respective project.

2.4.4 State

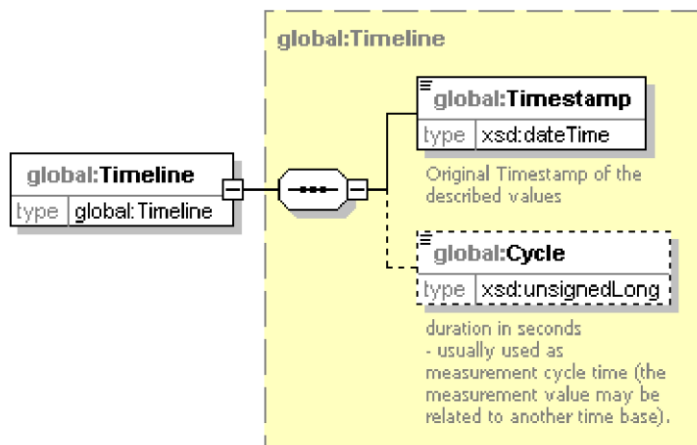
Dynamic data have a status. This is displayed by the state. All dynamic data use the globally defined *State*.



Sort	Value	Description
enumeration	o.k.	Field device o.k.: Values can be used
enumeration	n.o.k.	Field device not o.k.: Values cannot be used
enumeration	p.o.k.	Field device partially o.k. There may be a problem when using the values
enumeration	unknown	Field device status is unknown: Values should not be used
enumeration	offline	Field device is not connected.
enumeration	busy	Required value does not match the present value (e.g. since a switching operation is currently taking place).
enumeration	substitute_value	Device software substitutes a value (e.g. switches to a new device profile)

2.4.5 Timeline

Dynamic data have a time reference. This is displayed by the time stamp. All dynamic data use the globally defined timeline.



3 Data catalogue

The object types described here are defined as data types within OCIT-C.

Graphic portrayals of the schema definitions are used in this document to explain the data structures. Detailed information about this can only be found in the schema definitions. They contain the schema files (.xsd) referenced here in text form.

The schema definitions were written up in English with the exception of intersection_config_data (TSS supply data to be predominantly used in German-speaking areas).

All data described here are entered into the protocol.xsd within the "data" element, starting with their root element. The root element is always the element, which is labelled with the comment ("objecttype") within the schema definition in the annotation of the element. The object type listed within is also used as the identifying characteristic of this data type.

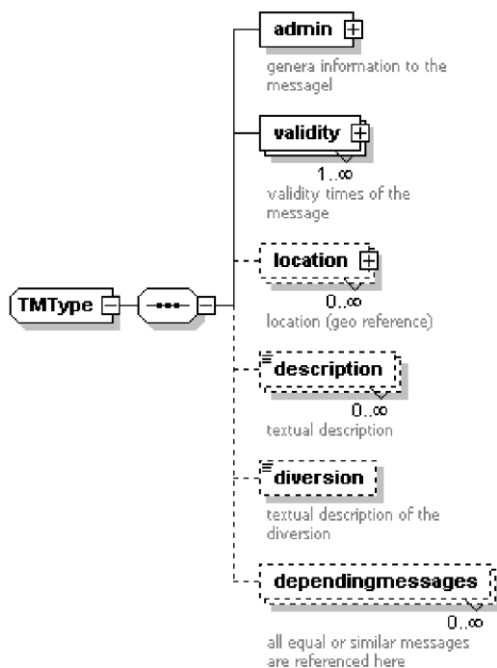
3.1 Traffic messages

This communication module is used to transfer messages regarding roadworks, events, and faults.

Traffic messages are identified using ID and source information.

Available object types:

Object type	Schema file	Short description
TrafficMessage_RoadWorks	traffic_messages.xsd	Roadworks
TrafficMessage_Events		Events
TrafficMessage_Incidents		Faults



3.2 Traffic data

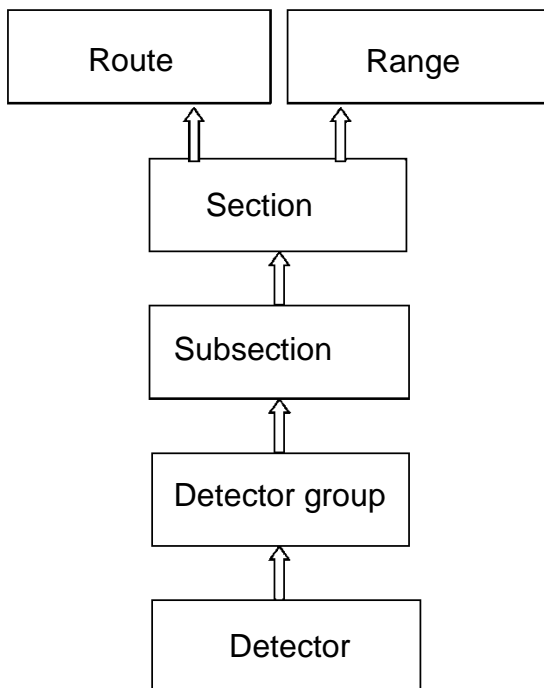
This communication module is used to transfer measurements or derived values from various detection devices. The current and accumulated measurement results from individual detectors or detector groups are transferred, however not detector raw data (impulse).

Available object types:

Object type	Schema file	Short description
TrafficData_detector_Description	Traffic_data.xsd	Data describing detectors
TrafficData_detector_currentValue		Current data from detectors
TrafficData_detectorGroup_Description		Data describing measuring points
TrafficData_detectorGroup_currentValue		Current data of a measuring point
TrafficData_detectorGroup_calculatedValue		Derived data of a measuring point
TrafficData_trafficSubSection_Description		Data describing subsections
TrafficData_trafficSection_Description		Data describing sections
TrafficData_trafficArea_Description		Data describing areas

Object type	Schema file	Short description
TrafficData_trafficRoute_Description		Data describing routes
TrafficData_trafficSubSection_accumulatedValue		Dynamic values of subsections
TrafficData_trafficSection_accumulatedValue		Dynamic values of sections
TrafficData_trafficArea_accumulatedValue		Dynamic values of areas
TrafficData_trafficRoute_accumulatedValue		Dynamic values of routes

The following illustration shows the various hierarchies to which the respective object types are assigned.



3.2.1 Subsection, sections, traffic areas

Subsections, sections and traffic areas are referenced using pre-defined IDs.

Recommended identification:

SS<primary key>_<secondary_key> for subsection

S<primary key>_<secondary_key> for section

R<primary key>_<secondary_key> for routes

TA<primary key>_<secondary_key> for traffic areas

The use of the <primary key> is not absolutely required.

3.2.2 Detectors and detector groups

Detectors (e.g. loops) and measuring points (= detector groups) are identified using pre-defined IDs.

Recommended identifications:

- For detection equipment connected via OCIT-O; the object identification takes place in accordance with section 2.3.1 on page 13
- For any other detection equipment:
D<primary key>_<secondary_key> for detectors
DG<primary key>_<secondary_key> for measuring points

The use of the <primary key> is not absolutely required.

To identify detectors supplied via OCIT-C VD, only the primary key formed from the system described in Chapter 3.11 is used.

3.2.2.1 Object type: TrafficData_detector_Description

The data structure is clarified in Chapter 2.4.1.

Use of the element **type**

- Description of the detector type: Loop, double loop, camera, etc.

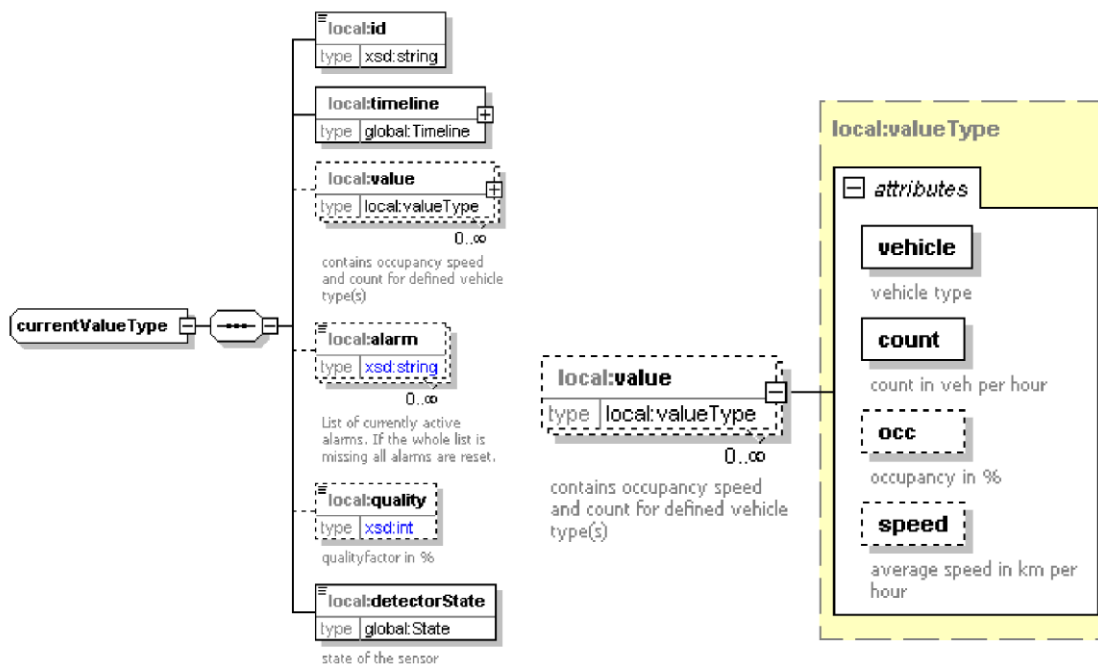
Use element **subsystem**

- Manufacturer/supplier.

Use element **Rel_Ids**

- Reference to measuring points
- Reference to TSS

3.2.2.2 Object type: TrafficData_detector_currentValue

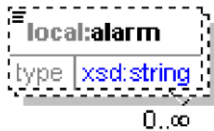


The "value" structure is assumed as the field element for each available vehicle type. Types of vehicle which are not available should not be assumed as a field element.

As soon as there are valid values available, the vehicle type "all" is delivered.

Vehicle types:

Vehicle type in OCIT-C	Equivalence in TLS
<xsd:enumeration value="all"/>	Total
<xsd:enumeration value="car"/>	Car
<xsd:enumeration value="truck"/>	Truck
<xsd:enumeration value="artic_truck"/>	Articulated truck
<xsd:enumeration value="motorcycle"/>	Motorcycle
<xsd:enumeration value="unclassified"/>	"Phantom"
<xsd:enumeration value="car_with_trailer"/>	Car with trailer
<xsd:enumeration value="truck_with_trailer"/>	Truck with trailer
<xsd:enumeration value="bus"/>	Bus
<xsd:enumeration value="delivery_truck"/>	Delivery truck



List of currently active alarms. If the whole list is missing all alarms are reset.

Alarm list:

"NoVideo", "SpeedAlarm", "OccupancyAlarm", "CommunicationError", "StoppedVehicle", "SpeedVariation", "InverseDirection", "SmokeDetected", "BadVideo", "DigitalInput", "UserAlarm", "ConfigChanged", "ZoneActive", "OutputGroupActive", "PhaseActive", "IndiffData", "RebootEvent", "Upgraded", "RecallPressed", "DownloadFailed", "IllegalEvent", "SpeedAlarmLevel1", "SpeedAlarmLevel2", "SpeedAlarmLevel3", "IncidentAlarm", "IncidentWarning"

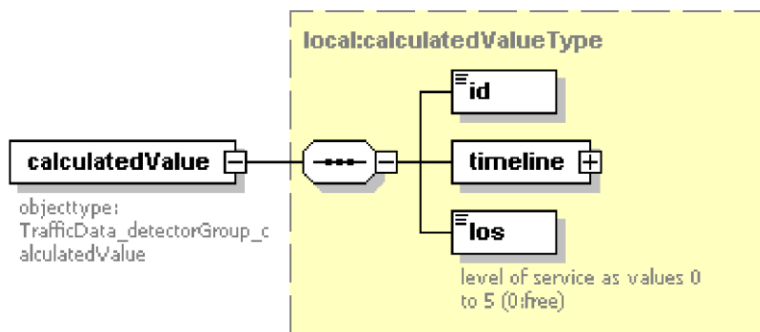
3.2.2.3 Object type: TrafficData_detectorGroup_Description

Use such as TrafficData_detector_Description

3.2.2.4 Object type: TrafficData_detectorGroup_currentValue

Use such as TrafficData_detector_currentValue

3.2.2.5 Object type: TrafficData_detectorGroup_calculatedValue



3.2.2.6 Object type: TrafficData_trafficSubSection_Description

The data structure is clarified in Chapter 2.4.1.

Use element **subsystem**

- Description of the data supplier

Use element **Rel_Ids**

- Description of the assigned detector groups or other data from the traffic data

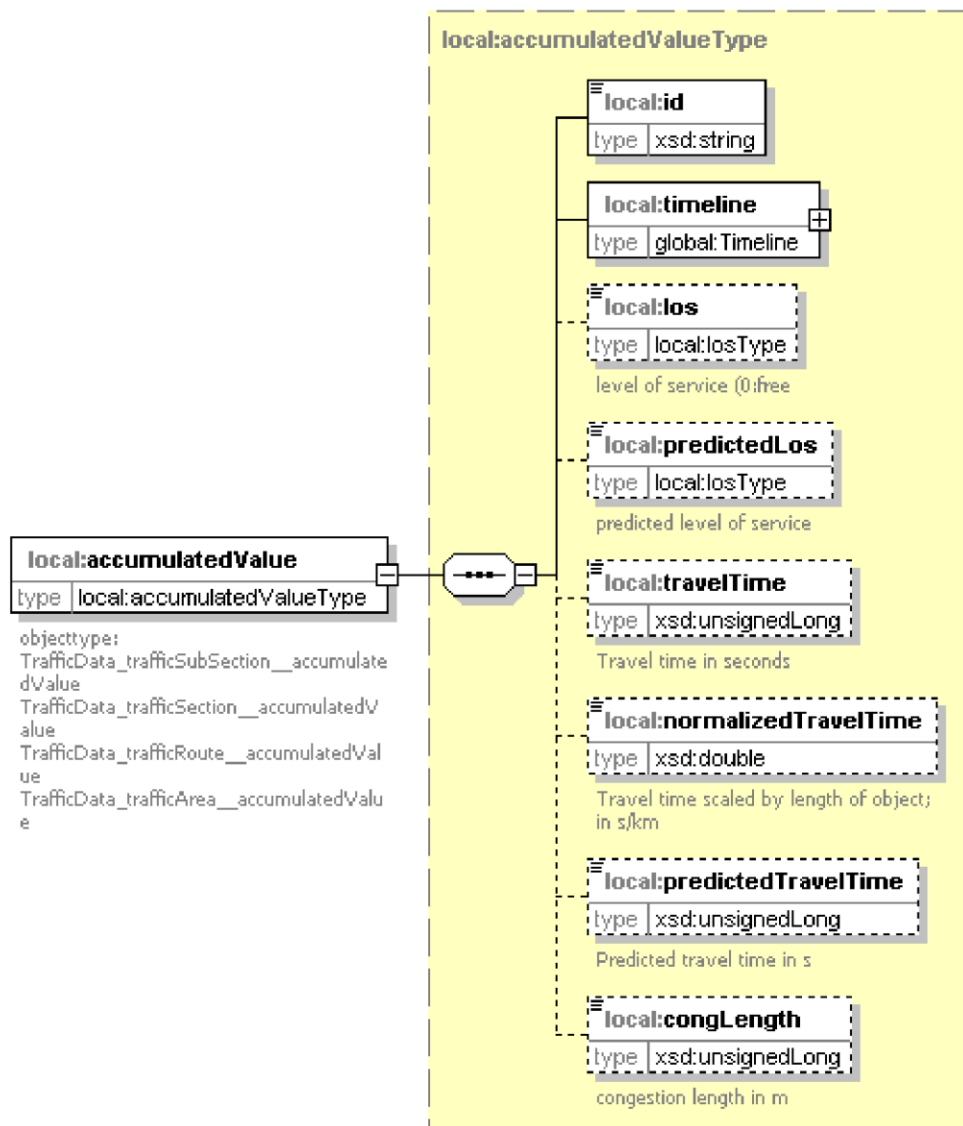
3.2.2.7 Object type: TrafficData_trafficSection_Description

Use such as TrafficData_trafficSubSection_Description

3.2.2.8 Object type: TrafficData_trafficArea_Description

Use such as TrafficData_trafficSubSection_Description

3.2.2.9 Object type: TrafficData_trafficSubSection_accumulatedValue



3.2.2.10 Object type: TrafficData_trafficSection_accumulatedValue

Use such as TrafficData_trafficSubSection_accumulatedValue

3.2.2.11 Object type: TrafficData_trafficArea_accumulatedValue

Use such as TrafficData_trafficSubSection_accumulatedValue

3.3 Parking data

This communication module is used to transfer data from parking facilities: Description and current occupancy of parking facilities, parking lots, parking areas.

Parking data are referenced using pre-defined IDs.

Recommended identification:

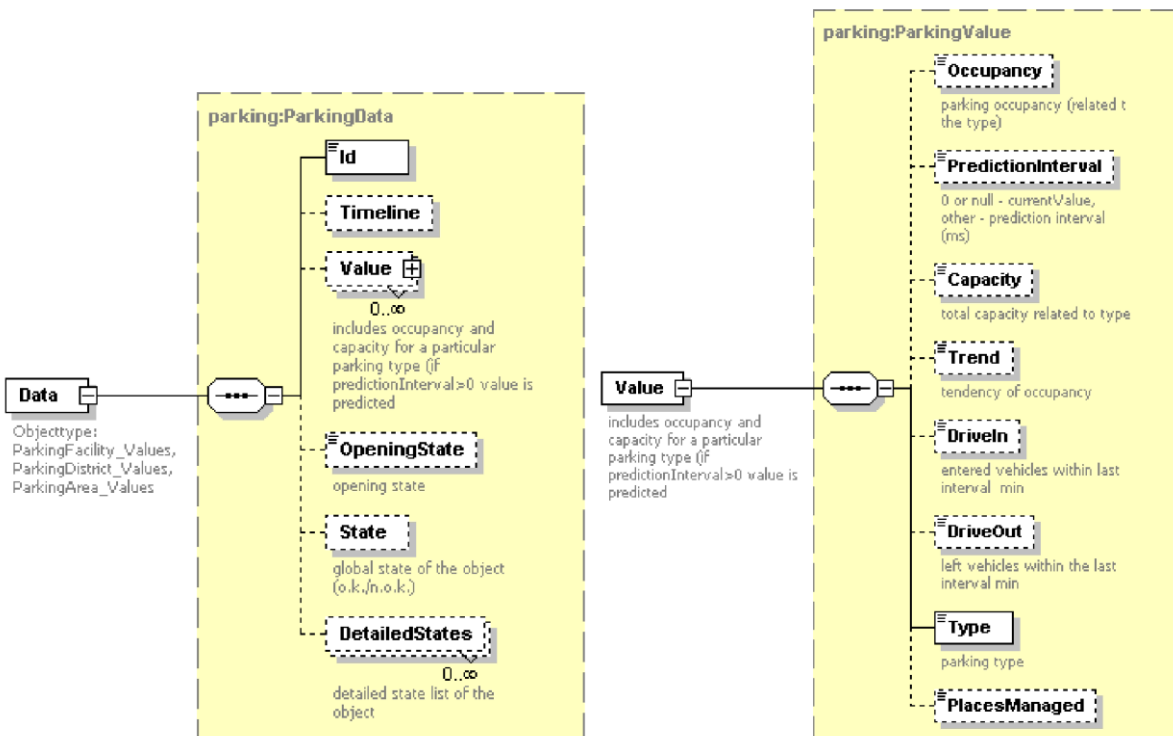
PF<primary key>_<secondary_key> for parking facilities

PD<primary key>_<secondary_key> for parking areas and parking spaces

The use of the primary key is not absolutely required.

Available object types:

Object type	Schema file	Short description
ParkingFacility_Description	parking.xsd	Description of the parking facility data
ParkingDistrict_Description		Description of the parking district
ParkingArea_Description		Description of the parking area
ParkingFacility_Values		Dynamic data of the parking facility
ParkingArea_Values		Dynamic data of the parking area
ParkingDistrict_Values		Dynamic data of the parking district



3.4 Weather and environmental data

This communication module is used to transfer data from respective detection facilities.

Environmental and weather data are identified with pre-defined id's.

Recommended identification:

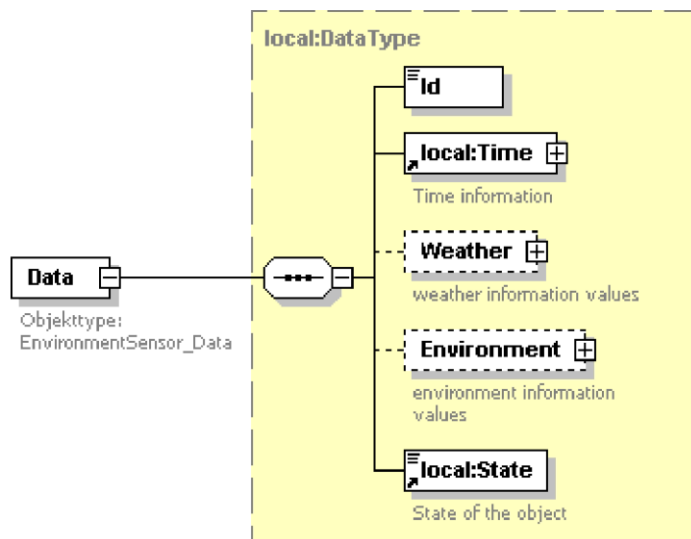
W<primary key>-<secondary_key> for weather sensors

E<primary key>-<secondary_key> for environmental sensors

The use of the primary key is not obligatory.

Available object types:

Object type	Schema file	Short description
EnvironmentSensor_Descr	environmentsensor.xsd	Description of weather or environmental data
EnvironmentSensor_Data		Dynamic weather or environmental data



3.5 Cameras

This communication module is used to transfer data to control monitoring cameras and to transfer images.

Camera data are identified with pre-defined id's.

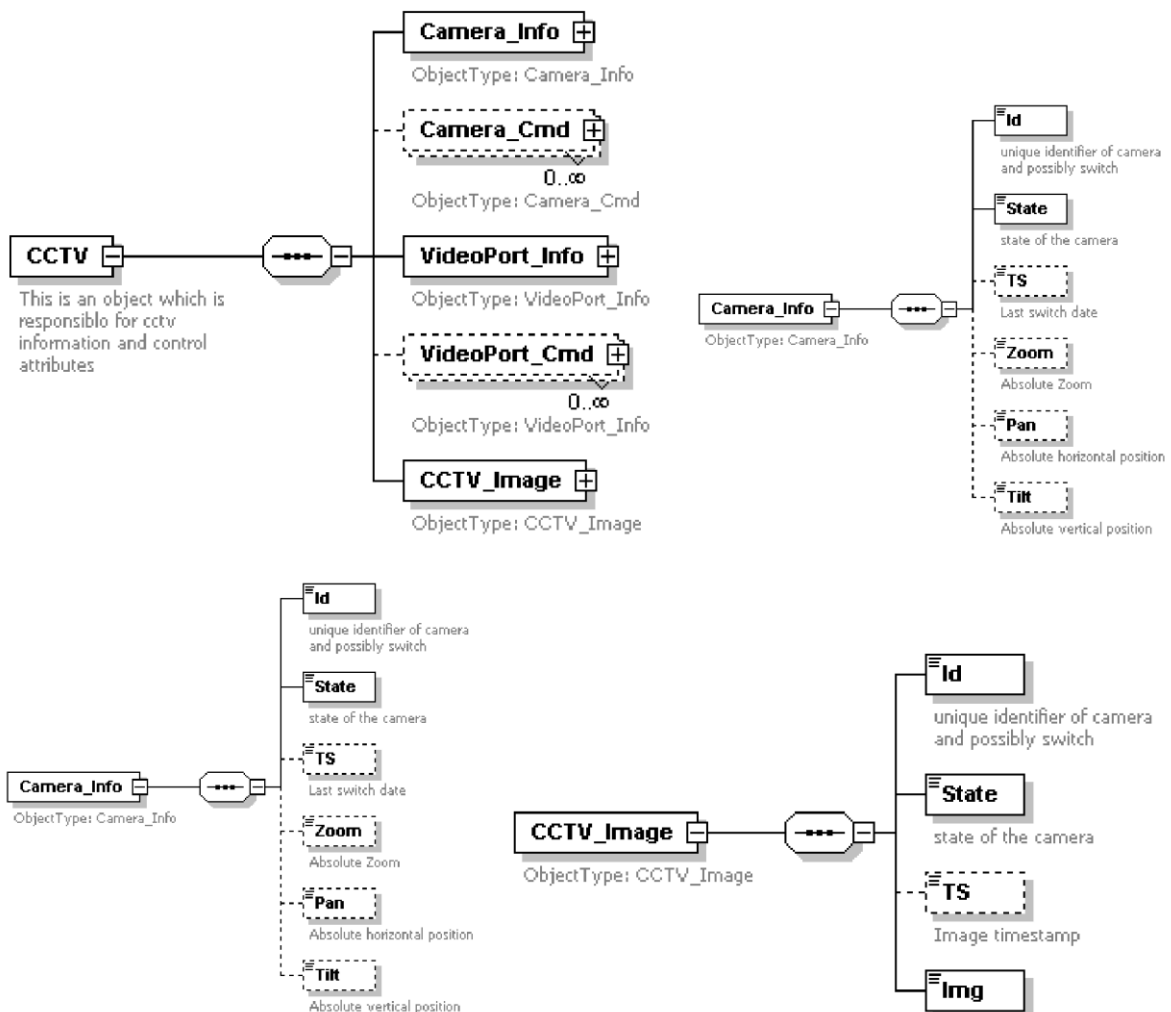
Recommended identification:

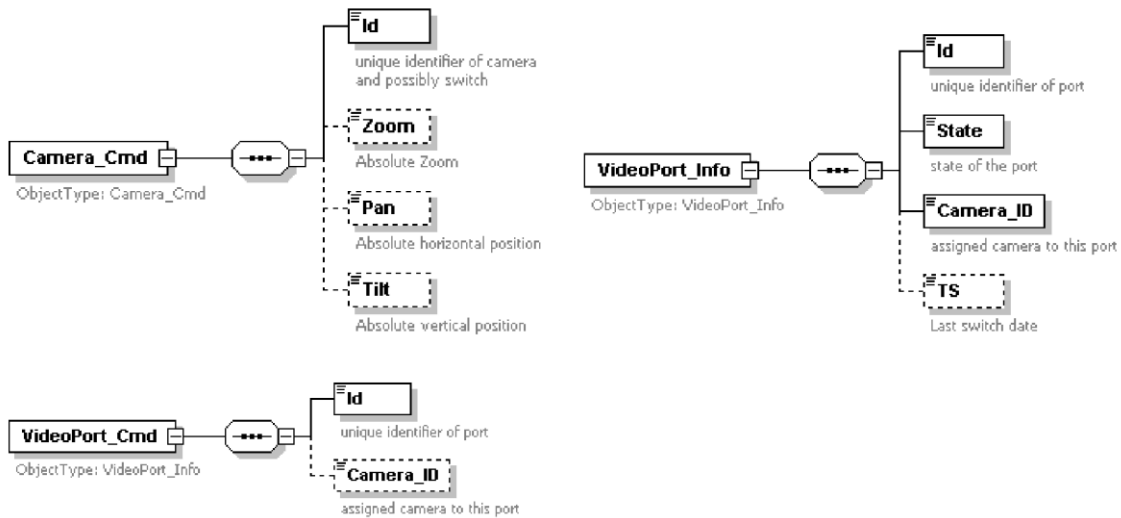
Cam<primary key>_<secondary key>

The use of the primary key is not obligatory.

Available object types:

Object type	Schema file	Short description
CCTV_Description	cctv.xsd	Description of the camera data
Camera_Info		Current status of the camera (without image)
CCTV_Image		Current image of a camera
Camera_Cmd		Command/adjusting command to a camera
VideoPort_Info		Current status of the monitor matrix
VideoPort_Cmd		Command/adjusting command for the monitor matrix
VideoPort_Cmd		Command/adjusting command for the monitor matrix





3.6 Situations and strategies

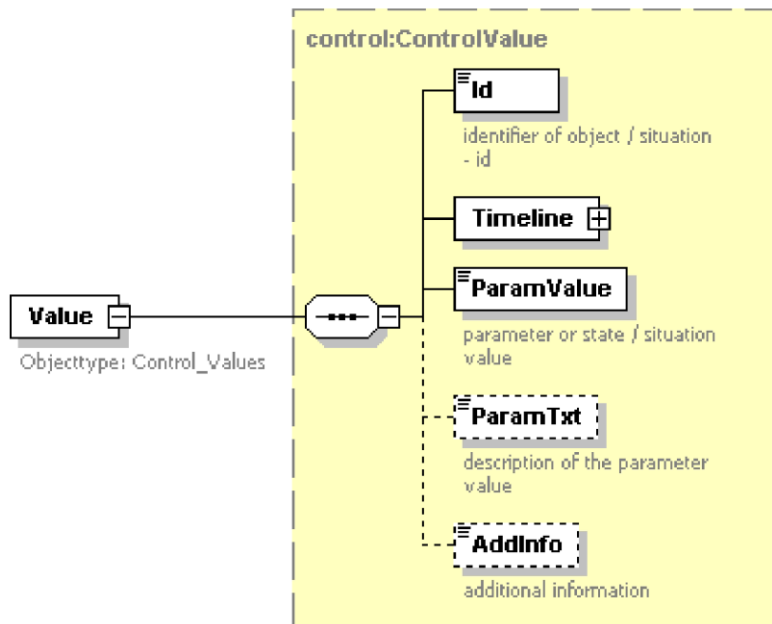
This communication module is used to transfer the descriptions of traffic situations or control strategies and input for strategic traffic control measures.

Situations or strategies are identified using pre-defined IDs.

Recommended identification:
 Sit<primary key>_<secondary_key>

Available object types:

Object type	Schema file	Short description
Control_Description	Control.xsd	Data describing the situation or strategy
Control_Values		Current strategy/situation



AddInfo can be used for additional information relating to the project.

Application examples:

- Strategy
The ID identifies the strategy, the variant of the strategy can be modelled using ParamValue.
- Situation.
The ID identifies the situation (e.g. the status of a set of doors), the status of this set of doors can be modelled using ParamValue (e.g. door open or door closed).
- Digital contact
The ID identifies the digital contact, the status of this contact can be modelled using ParamValue (e.g. high level or low level).
- Groups
The ID identifies the group (e.g. the TSS group), the status of this group can be modelled using ParamValue (e.g. group off, group in SP1, or activate group with specific figure).

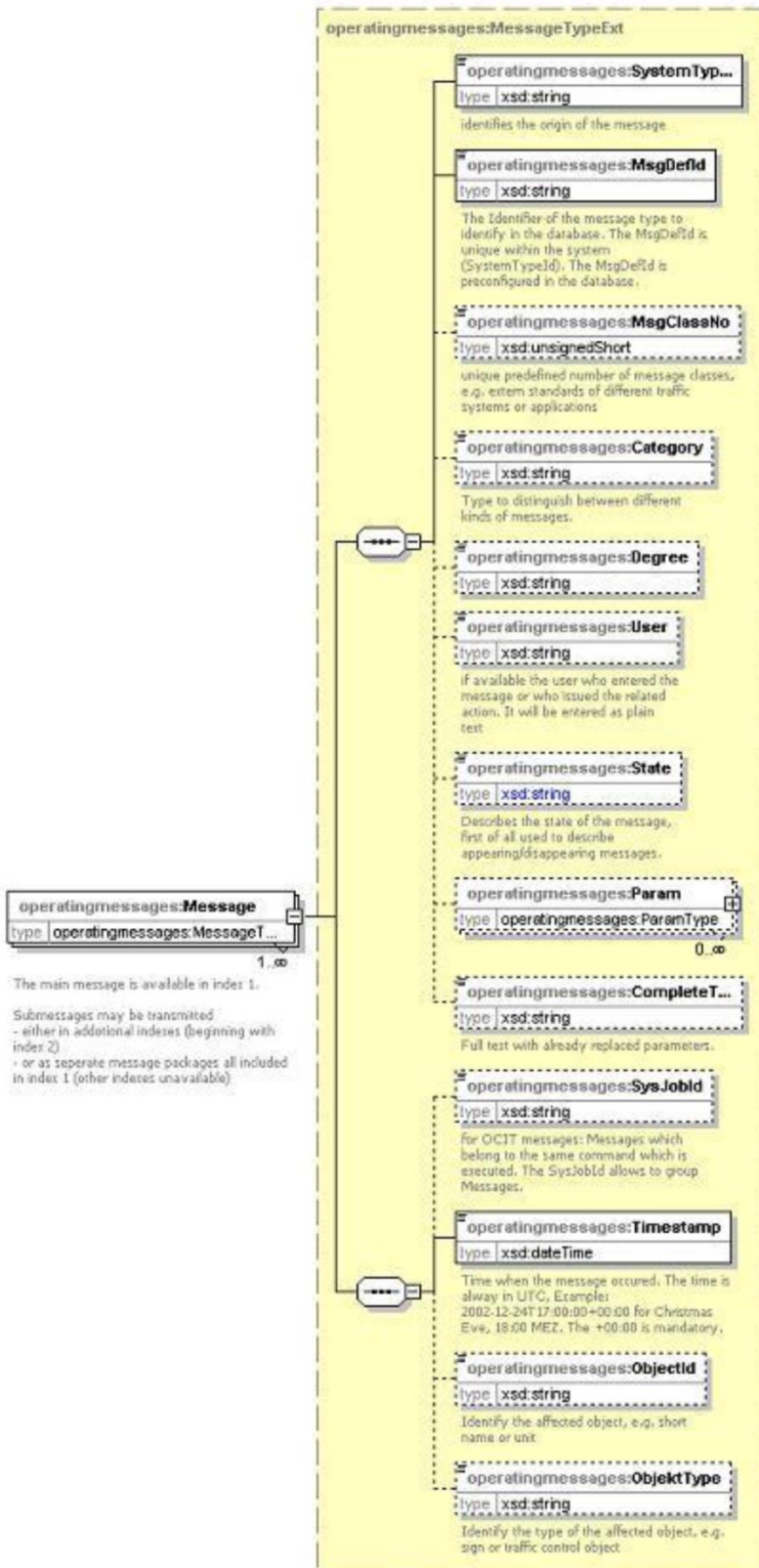
3.7 Operating messages

This communication module is used to transfer operating messages from traffic equipment, such as from traffic signal systems.

Available object types:

Object type	Schema file	Short description
OperatingMessages_Operatingmessage	operating_messages.xsd	Operating message

Operating messages have a main message and the option of secondary messages. The main message ranks first. Then the secondary messages follow. The elements of a main or a secondary message will be described below



Day	Description
SystemTypeID	Origin of the operating message (in OCIT-O, this corresponds to the member number)
MsgDefID	Message type for identification of the operating message. The message type is pre-configured uniquely within the system (SystemTypeField) and within the database (in OCIT-O, this corresponds to the otype).
MsgClassNo	Message class for differentiating between message groups from a manufacturer
Category	Characteristic for determining between various types of messages
Degree	Severity of the message
User	If available, enter the user who entered the message or performed the corresponding action.
State	Describes the status of the message, especially the depiction of the messages.
Param	Parameter for specifying the operating message. The parameter types are pre-defined within the database.
CompleteText	Complete text of the operating message including parameter
SysJobId	If messages are the cause of a certain action, then the action as well as the messages carry the same SysJobId. The SysJobId groups the messages by specific causes / triggers.
Timestamp	Time stamp of when the message was issued
ObjectId	Identification of the object in question, e.g. abbreviation
Object type	Identification of the object type in question, e.g. sign or traffic control object

You can specify switched and array parameters by fulfilling the corresponding ParamId: "x1", "x2", etc., "y1a", "y1b", "y2a", etc. To do this, a rule must be defined that states how class or field names and array indices shall be defined accordingly, in order to generate a unique ParamId.

3.8 Signs

This communication module is used to transfer data for controlling segment and full matrix signs.

Available object types:

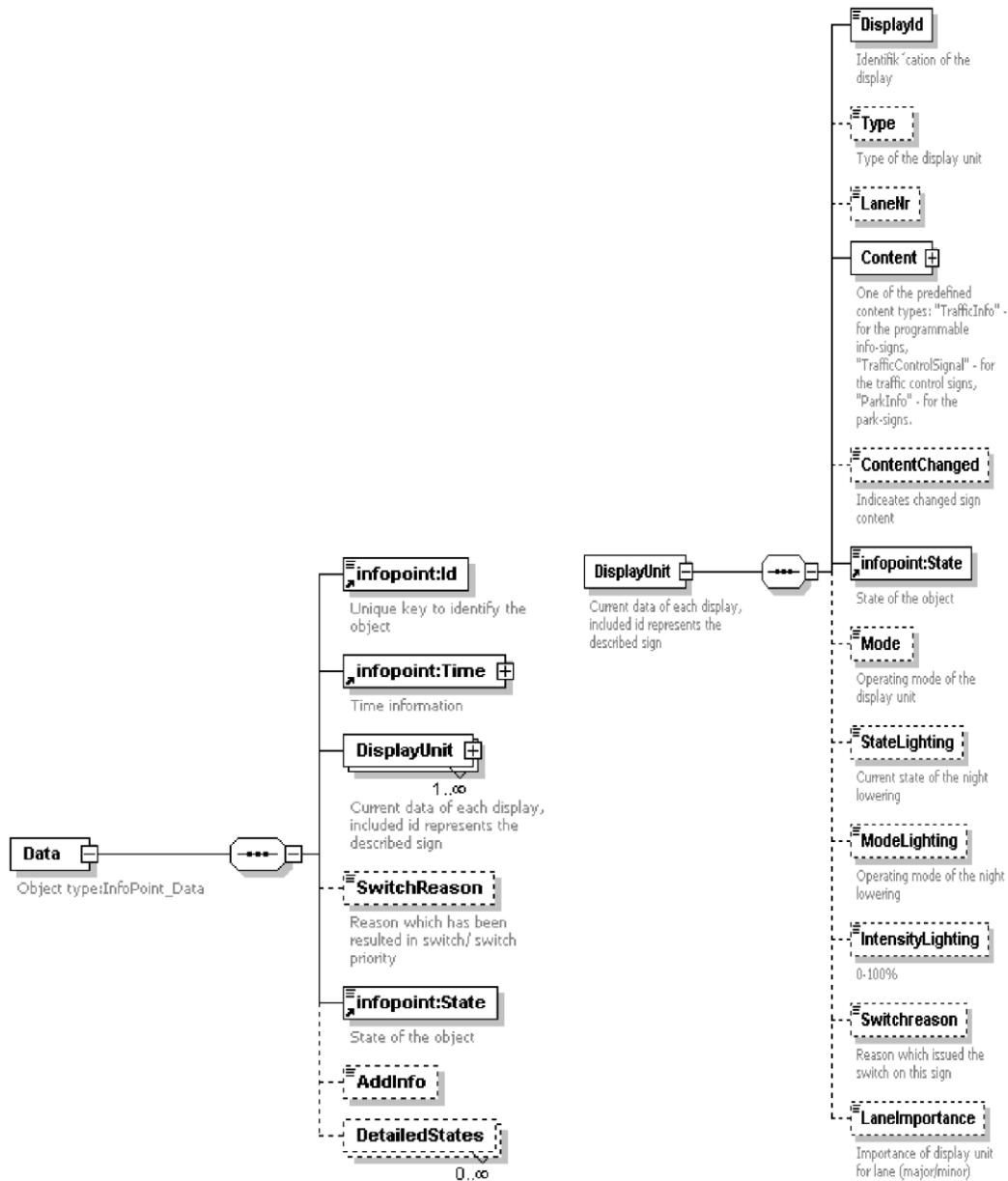
Object type	Schema file	Short description
Infopoint_Description	infopoint_data.xsd (imported sign.xsd)	Data describing signs
InfoPoint_Data		Sign contents (current status)
InfoPoint_Data		Sign contents (target status)
Infopoint_Cmd		Alternative adjusting command for signs

The sign.xsd is applied in the case of complex matrix signs.

3.8.1 Data model

The data model displayed below is used for exchanging sign contents as well as for activating signs.

The "state" XML element is only used for exchanging sign contents. This element does not play any role in activating signs.

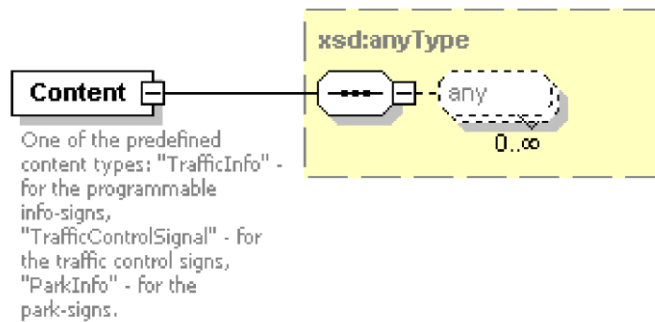


Day	Contents	Description
ID		Identification of the sign
Time	Timestamp	In UTC
DisplayUnit	sub elements	Only occurs - once on full-matrix displays - multiple times on segment displays
DisplayId	-	Labelled segment displays
Type	-	Unused

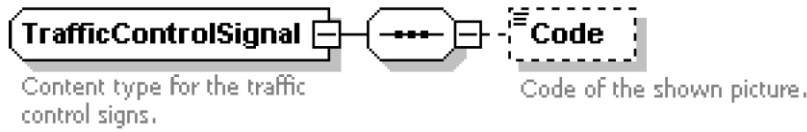
Day	Contents	Description
Content	As defined in sign.xsd	See sign.xsd
ContentChanged	-	Unused
State	-	Unused here
Mode	-	Determination of whether the part of the sign will be controlled centrally or locally
StateLighting		Unused
SwitchReason	-	Unused
State	o.k. n.o.k. busy	Status information from the sign server
	switch	Command display from the sign to the sign server
AddInfo		Additional information (optional).
On_off	on off	

3.8.2 Complex sign contents

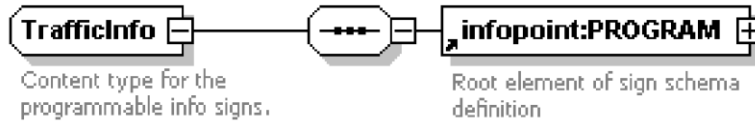
The sign.xsd models complex sign contents. The connection to the infopoint_data.xsd schema is realised using the "any" type:



The "any" field can identify the sign content using simple predefined code selection



- a programmable matrix definition or base 64 encoded picture stream

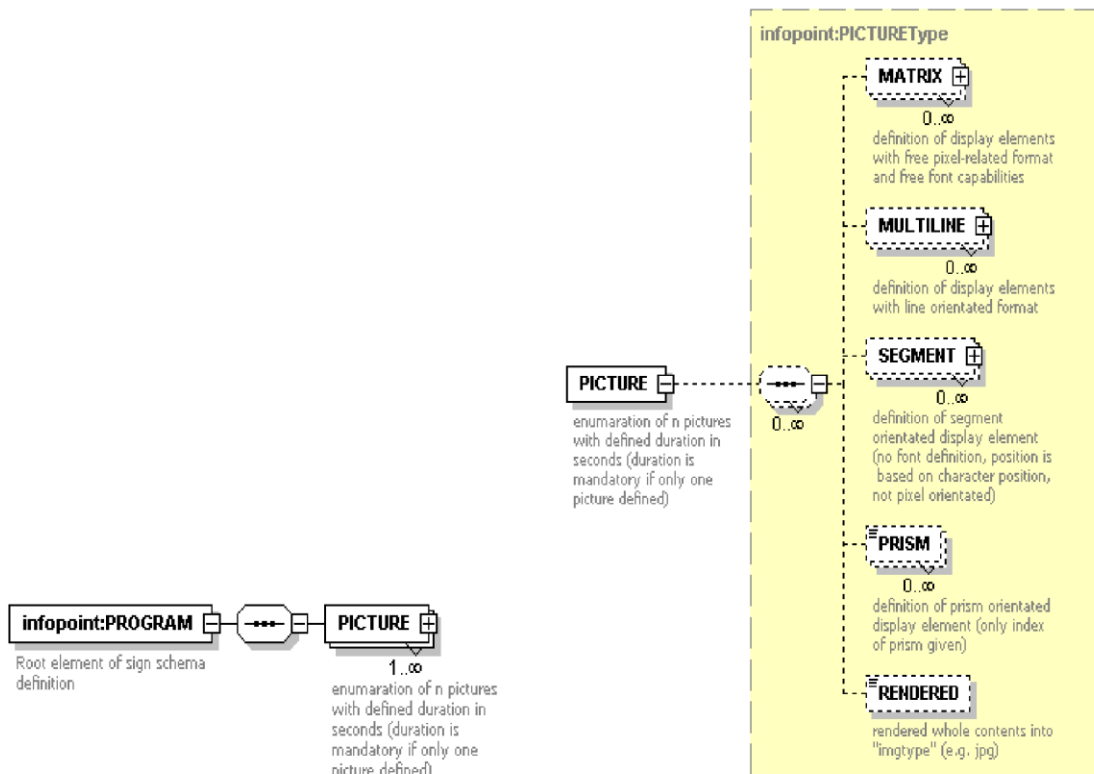


- describe parking information.

The content is a chain of characters e.g. a text, a place holder value or a side of a prism. The ContentType attribute is able to describe the use of the chain of characters in greater detail.

- Programme Definition

allows the definition of frame sequences:



3.9 Public transport passenger information

This communication module is used to transfer road, line, or public transport stop related data of the public transport network.

Publictransport_data are referenced using pre-defined IDs.

Recommended identification:

PT<primary key>_<secondary_key> for junctions

The use of the primary key is not absolutely required.

Available object types:

Object type	Schema file	Short description
PT_Description	publictransport_data.xsd	
PT_Link		Passenger information relating to the road: Number, speed and delay
PT_LinePerLinkLink		Passenger information relating to the line: Number, speed and delay

3.10 Traffic signal systems

This communication module is used to transfer data as input for controlling traffic signal systems.

Intersection_data are referenced using pre-defined IDs.

Recommended identification:

J<primary key>_<secondary_key> for junctions

The use of the primary key is not absolutely required.

Available object types:

Object type	Schema file	Short description
IntersectionDescription	intersection_data.xsd	Data describing TSS
IntersectionStatus		Current TSS data (actual status)
PredefCommand		Pre-defined TSS switching operations
IntersectionFeature		Ability of a TSS to perform a switching operation

Object type	Schema file	Short description
IntersectionCommand		TSS switching operation (modelled out as a replacement of the PredefCommand)
ControlMethodParameters		Dynamic parameters for traffic-related processes
SGTimes		Summarised red and green times for a cycle.

3.10.1 Pre-defined commands - obsolete

With the introduction of modelled commands, pre-defined commands have become superfluous. They will still be used during the transition period for compatibility reasons.

The following pre-defined commands are available:

Command	Configuration of the XML element
Intersection on/off	on off local
SPx (intersection is implicitly switched on) x is the program number	plan1 plan2 (planx) local plan
Individual transport on/off	it_on it_off it_local
PT on/off	pt_on pt_off pt_local
Local TA on/off	localta_on localta_off localta_local
Partial intersection on/off	Subintersection0_on; subintersection0_off; subintersection0_local subintersection1_on; subintersection1_off; subintersection1_local subintersection2_on; subintersection2_off; subintersection2_local subintersection3_on; subintersection3_off; subintersection3_local
Special commands	specialcommand<number>

Command	Configuration of the XML element
	localspecialcommand<number>
Project-specific modification	projectspecificcommand_<name or number> localprojectspecificcommand_<name or number>
Release	release

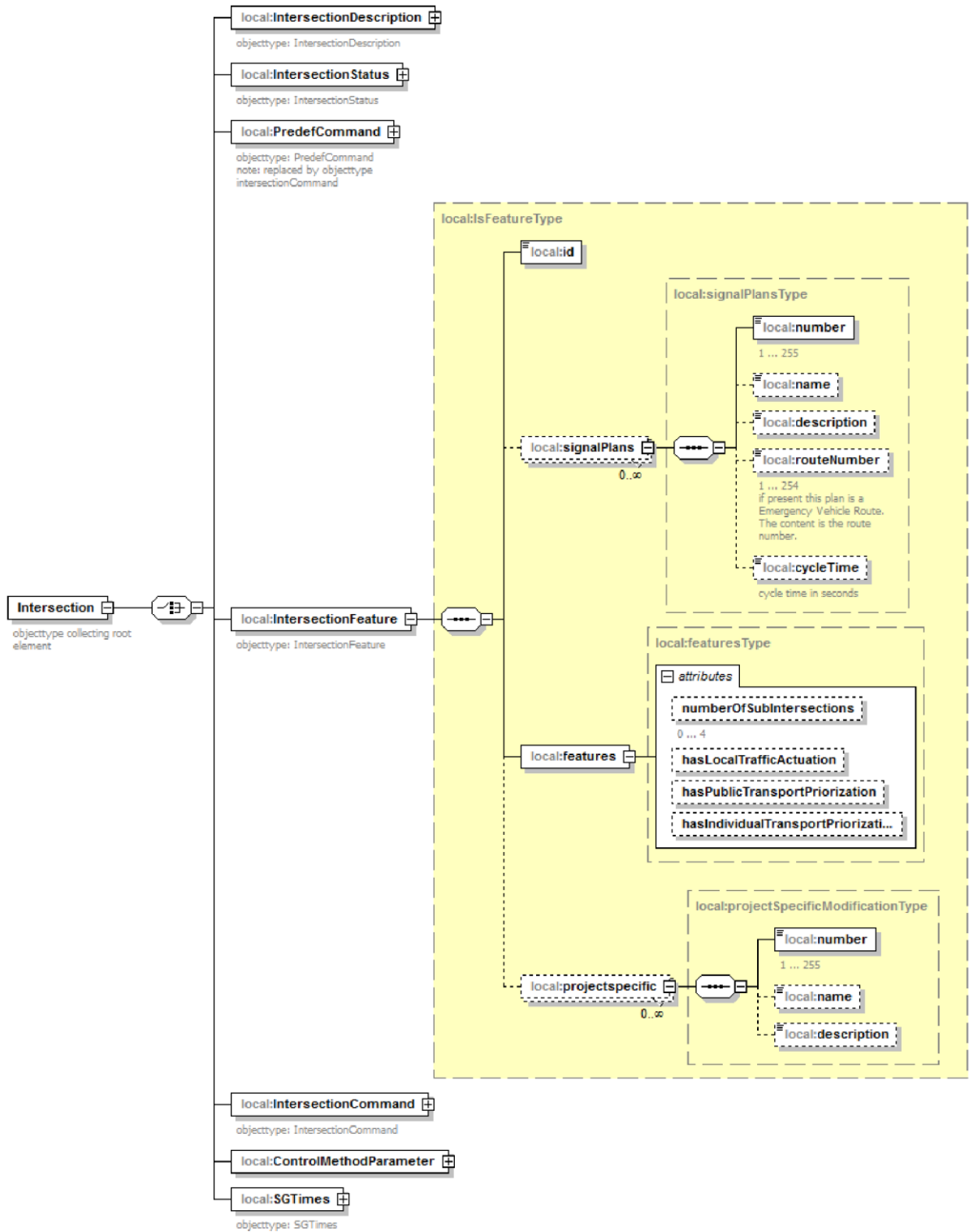
Further commands relating to the project can be added as part of the project.

Should multiple pre-defined commands be executed simultaneously at the same point in time, these must be executed within a put (within the put-List under the same object ID). The server interprets these so that these can only lead to one switchover in the controller.

3.10.2 Intersection Features

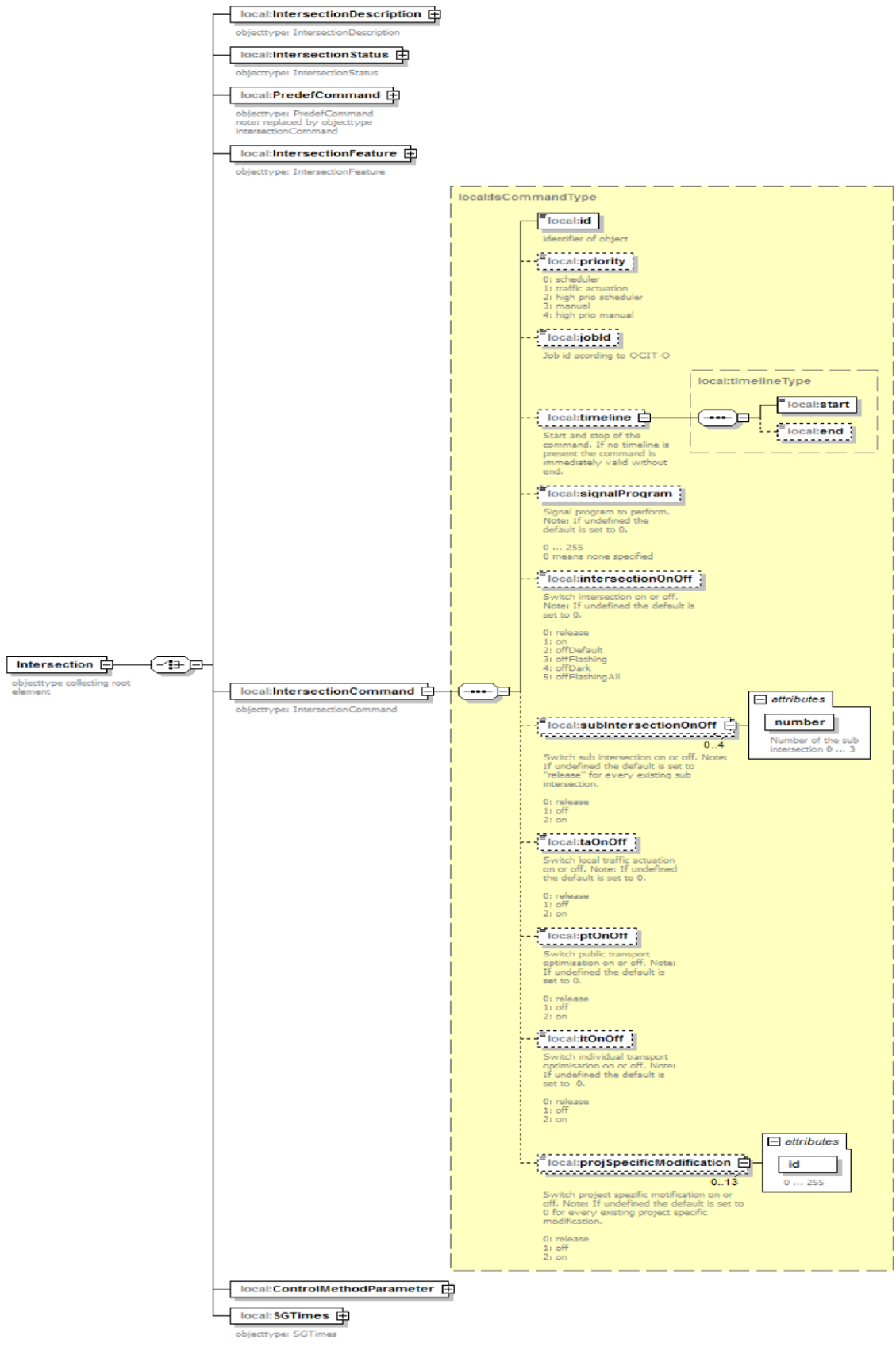
The IntersectionFeatures object type allows you to see the switching options of a TSS (e.g. which plans can be switched to).

This makes it possible to limit the selection of switching operations, before they are sent via the Intersection-Command (and in this regard declined).



3.10.3 IntersectionCommand (modelled switching operations)

The IntersectionCommand object type allows you to trigger switching operations.



3.10.4 ControlMethodParameter

The recipient of the dynamic parameters is a traffic-procedure within the traffic signal system.

Parameter	Procedure
path	Path for the ApValueName in OCIT-O, e.g. VS-PLUS.NETZSTEUERUNG
methodname	Name of the control procedure which should receive the data, e.g. VS-PLUS
interfacename	Name of the interface in the controller e.g. NETZSTEUERUNG
data	Binary data contents for APValueBlock (1:508)

Feedback can be given via suitable operating messages, that can be agreed upon on a project or process-specific basis.

3.10.5 TSS status (IntersectionStatus)

Detailed information about the status of the TSS can be found under "faultStatus":

TextNo: A number which labels the content

Text: Optional text entry

Params: Parameters for further details

- Time source (information about the time source of the device time)
TextNo: 1,001
Text: "Time source"
Params: 0 (=line voltage), 1 (=Quartz clock), 2 (=LAN), 3 (=DCF radio), 4 (=GPS), 5 (=Unknown)
- EmergencyOff (information about whether the EMERGENCY-OFF switch was actuated)
TextNo: 1,002
Text "EmergencyOff"
- DoorOpen (information about whether the door-closing switch was actuated)
TextNo: 1,003
Text: "DoorOpen"
- LinevoltageOK (information about whether there is line voltage present)
TextNo: 1,004
Text "LineOk"
- PersistenceStorageOK (information about whether the persistent storage is in order)
TextNo: 1,005
Text: "PersistenceStorageFaulty"
- Lamp fault (information about faulty lamps)
TextNo: 1,006
Text: "LampFault"

Params: One parameter per faulty lamp with contents: "<RelIntersection>-<SignalGroup>-<SignalHead>-<Chamber>", e.g. "1-8-2-3"

- Detector fault (information about faulty detectors)
TextNo: 1,007
Text: "DetectorFault"
Params: One parameter per faulty detector with content channel number, e.g. "5"

Collective fault from OCIT-O is listed under faultState.

TextNo: 1,008
Text: CollectiveFault
Params: A parameter. The values: NoFault, Fault, FaultWithShutoff, FaultWithPartialShutoff, InternalFault

3.10.6 SGTimes

Red and green times for signal groups are summarised relating to the cycle in the SGTimes object type. The addressing is done at the TSS level, i.e. it is not possible to activate signal groups individually within this object type.

3.11 TSS raw data

This communication module is used to efficiently transfer a large amount of confidential values from traffic signal light systems.

The data to be transferred are grouped by statuses. Each status assumed is listed once and assigned to the point in time at which it was assumed. All time stamps from a certain time interval contain a start time (xsd:element name="timeline") for the interval and the offset until the start time. The offset can be depicted as a product made up of a number and a time unit (xsd:element name="intervalLength"). This number is saved as an unsigned 16 bit value. On the one hand, this results in a maximum interval length, on the other hand, it is possible to list all 16 bit values successively without separate XML elements. The XML representation for the field of 16 bit values is then its Base64 coding (xsd:element name="Events").

Recommended identification:

- Object identification in accordance with section 2.3.1 on page 13

Available object types:

Object type	Schema files	Short description
RawTrafficDataBlock_Detectoredge		Detector edges
DigOut_Raw_Values		Digital outputs

Object type	Schema files	Short description
RawTrafficDataBlock_Signalgroupvalue	intersection_raw_data.xsd	Signal group values
NamedValue_Raw_Values		AP - values
NamedValuesOrderList	intersection_raw_data.xsd	AP value order
PublicTransportTelegramExtended	Public_Transport_ExtendedTelegram.xsd (importiertPublic_Transport_Telegram.xsd)	Extended PT telegram
DetectorExtValueType	detector_ext.xsd	Data from individual detectors

3.11.1 Detector edges

The detectorEdges object (object type RawTrafficDataBlock_Detectedge) has been defined for transferring the detector edges.

The value ("xsd:string) is set with '0' for falling edges and '1' for rising edges.

Example:

Detector had the "occupied" status at the following times (rising):

1. 2011-03-23T14:20:00.100+01:00,
2. 2011-03-23T14:20:01.200+01:00,
3. 2011-03-23T14:20:02.000+01:00.

Detector had the "not occupied" status at the following times:

4. 2011-03-23T14:20:00.300+01:00,
5. 2011-03-23T14:20:01.800+01:00,
6. 2011-03-23T14:20:02.300+01:00.

The data are to be entered as follows:

ID of the detector: id: Det_1

Start time: timeline/TimeStamp: 2011-03-23T14:20:00+01:00

Time unit in milliseconds: intervalLength: 100

Rising edges: data/Value: 1

Events:

Calculation provision: (Event time - start time)/time unit == number

(2011-03-23T14:20:00.100 - 2011-03-23T14:20:00.000) / 100 == 1

Number is saved as an unsigned 16 bit value.

1: 100/100 == 0x0001

2: 1200/100 == 0x000c

3: 2000/100 == 0x0014

All 16 bit values should be entered successively.

Byte1 == 0x00

Byte2 == 0x01

Byte3 == 0x00

Byte4 == 0x0c

Byte5 == 0x00

Byte6 == 0x14

data/Events: AAEADAAU (0001000c0014 coded in Base64)

Falling edges: data/Value: 0

Events:

4: 300/100 == 0x0003

5: 1800/100 == 0x0012

6: 2300/100 == 0x0017

data/Events: AAMAEgAX (000300120017 coded in Base64)

3.11.2 Signal group states

The sgValues object (object type RawTrafficDataBlock_Signalgroupvalue) has been defined for transferring the signal group statuses.

The value (xsd:string) is defined as an integer. The value should be used identically to OCIT-O object signal aspects 1:611 (e.g. 3 for red, 48 for green).

If the value is set to "3", the point in time at which the signal group switches to "red" is entered.

If the value is set to "48", the point in time at which the signal group switches to "green" is entered.

Example:

Signal group had the "red" status at:

1. 2011-03-23T14:20:10.000+01:00,
2. 2011-03-23T14:21:10.000+01:00,
3. 2011-03-23T14:22:10.000+01:00.

The data are to be entered as follows:

id: Sg_1

timeline/Timestamp: 2011-03-23T14:20:00+01:00

intervalLength: 1,000

data/Value: 3

Events:

1:10000 / 1000 == 0x000a

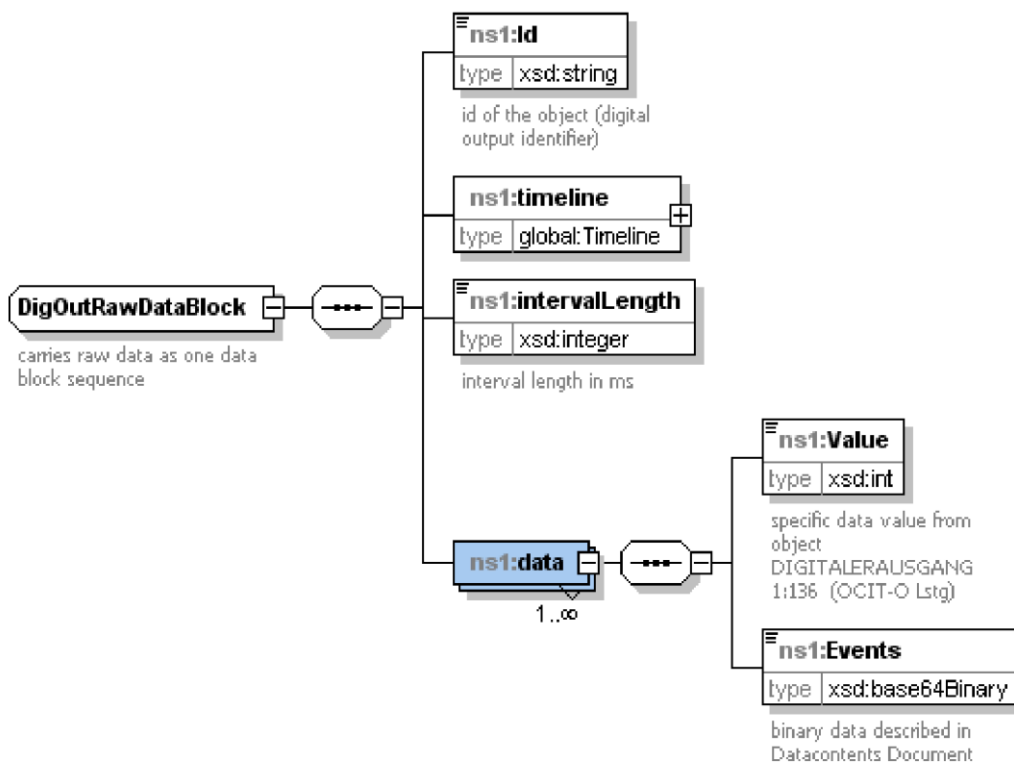
2:70000 / 1000 == 0x0046

3:130000 / 1000 == 0x0082

data/Events: AAoARgCC (000a00460082 coded in Base64)

3.11.3 Digital outputs in the traffic signal controller

The object DigOutRawType (object type DigOut_Raw_Values) has been defined for transferring the statuses of digital outputs.



The status has been defined as an integer type. The value has been adopted from the OCIT-O object DIGITALERAUSGANG 1:136, e.g. 3 for one, 1 for flashing start off.

If the value is set to "3", the point in time at which the digital output switches to the "on" state is entered.

If the value is set to "1", the events for which the digital output switches to the "Flashing start off" state is entered.

Example:

Digital output has the "on" status at:

1. 2011-03-23T14:20:00.100+01:00,
2. 2011-03-23T14:20:01.200+01:00,
3. 2011-03-23T14:20:02.000+01:00

The data are to be entered as follows:

id: Dout_1

timeline/TimeStamp: 2011-03-23T14:20:00+01:00

intervalLength: 100

data/Value: 3

Events:

1:100/100 == 0x0001

2:1200/100 == 0x000c

3:2000/100 == 0x0014

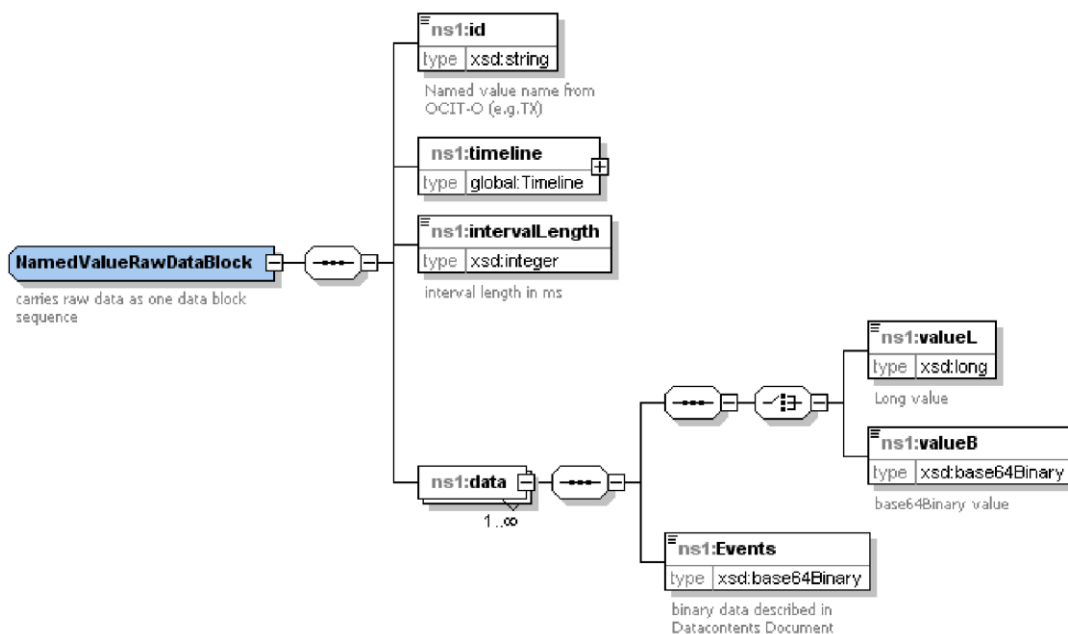
data/Events: AAEADAAU (0001000c0014 coded in Base64)

3.11.4 User program values (AP-values)

The NamedValueRawType object has been defined for transferring the status of AP values.

AP values are identified via an identification string. The string is defined in the XML schema intersection_config_data_ap_values.xsd (see OCIT-C_LSA_Versorgungsdaten) under OI-TDdef/OITDdef/Verfahren/OITD/Bezeichner/KurzBez.

If only certain AP values should be delivered, (e.g. only TX), in the protocol's "get" method, the "data" element with the data type "NamedValueFilterType" needs to be set. Otherwise all AP values which the server has collected for this system are delivered.



The state has been defined as a selection between the long and the base64Binary types. If AP values of the types:

long, integer, short, byte should be transferred, valueL is set.

If a BLOB type AP value should be transferred (cf. OCIT-O object APValueBlock 1:508), valueB is set. The binary data from the BLOB type are transferred (BYTE data[]).

OCIT-C defines the following standardised AP values:

OITD IP notation	OITD4-No.	OITD2-No.	Name	Description
41.94	2687070	42078	TX	The cycle second of the signal program running is counted beginning with second 0 in 100ms increments. 65535: Value undefined
41.96	2687072	42080	PH	Current stage in the signal program 0: Stage not active or not defined in the process 1 to 65534: Current stage number
41.97	2687073	42081	UE	Requested stage 0: No stage transition active 1... 65534: Transition active from stage PH to stage UE

Other names for the AP values are to be agreed upon on a project-specific basis.

Examples:

- Cycle second (TX) has the state "10" at:

1. 2011-03-23T 14:20:10.000+01:00,
2. 2011-03-23T 14:21:10.000+01:00,
3. 2011-03-23T 14:22:10.000+01:00.

The data are to be entered as follows:

id: TX

timeline/TimeStamp: 2011-03-23T14:20:00+01:00

intervalLength: 1,000

data/Value: 10

Events:

1:10000 / 1000 == 0x000a

2:70000 / 1000 == 0x0046

3:130000 / 1000 == 0x0082

data/Events: AAoARgCC (000a00460082 coded in Base64)

- AP value block (ABValueB) has the state "BYTE data[01, 05, 0c, a2]" at:

1. 2011-03-23T14:20:10.000+01:00,

2. 2011-03-23T14:21:10.000+01:00,

3. 2011-03-23T14:22:10.000+01:00.

The data are to be entered as follows:

id: APValueB_1

timeline/Timestamp: 2011-03-23T14:20:00+01:00

intervalLength: 1,000

data/Value: AQUMog == (01050ca2 coded in Base64)

Events:

1:10000 / 1000 == 0x000a

2:70000 / 1000 == 0x0046

3:130000 / 1000 == 0x0082

data/Events: AAoARgCC (000a00460082 coded in Base64)

3.11.4.1 Setting of requestable AP values

To start externally requestable AP values, an object type has been introduced which makes a list of "requestable AP values" through precisely one interface partner possible. This means, the specification is made in the customer project as to whether the configuration of this list can even be set, and if yes, by which communication partners.

The request of the AP values by multiple communication partners is however possible, even if the configuration of the list can only be set by one partner.

Object type: "NamedValuesOrderList"

3.11.5 Extended PT telegram

The object PTExtendedTelegramType (object typePublicTransportTelegrammExtended) has been defined for transferring extended public transport telegrams. For the transfer in the protocol, the rawDC object is used together with the time stamp as the container.

3.11.6 Data from individual detectors

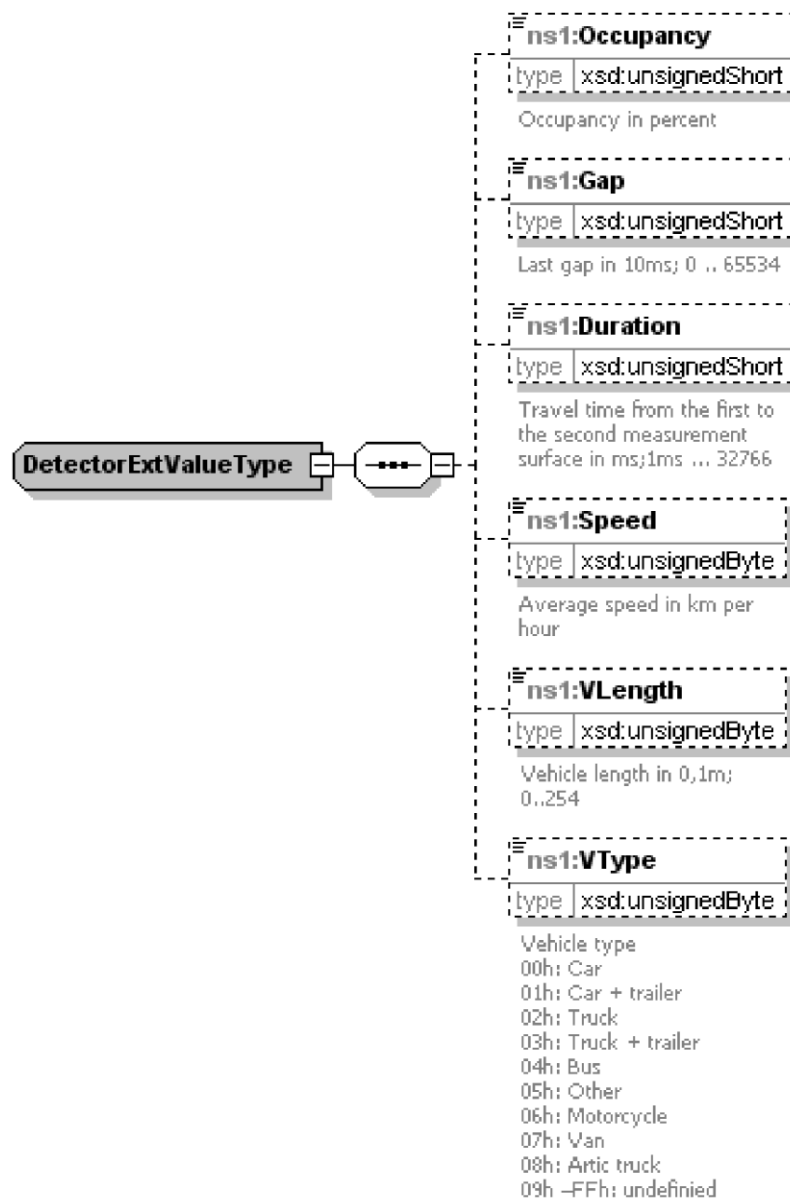
The DetectorExtValueType (detector_ext.xsd) object has been defined for transferring additional information from detectors. The measurement refers to the detector event of an individual vehicle.

Note: Since large data quantities are involved with the transfer of individual values, the data should only be available for a few detectors.

Recommended identification:

DExt<primary key>_<secondary key>

<primary key> is optional.



The extended detector measurements include the following elements:

Occupancy - occupancy duration of the measuring point in 10ms

Gap - last gap time before vehicle in 10ms

Duration - journey time from the first to the second measuring point in ms

Speed - measured speed in km/h VLength - length of the vehicle

VType - type of vehicle (cf. OCIT-O object VehicleType 1:3128)

3.12 TSS supply data

This communication module is used for supplying data to traffic signal systems. The methods and data for supply data from OCIT-C and OCIT outstation version 2 have been adopted in full and supplemented with:

- Fault messages for the planning tool
- Asynchronous feedback if there is a change of supply
- List of the AP values which can be delivered for a TA process
- Request of the field device information

For a detailed description, see document OCIT-C_TSS_Supply data.

3.13 Binary container

This communication module is used to transfer binary encoded data. The data are assigned to a point in time (or time period) and an object. The name type identifies the type of data in the container and thus makes it possible for the recipient to process.

Binary data are identified using pre-defined IDs.

Example: The so-called "Compromised messages" from a TSS should be transferred to the container, the name type in this case would be "KPM", the object ID would be specified as the ID of the TSS and the data container would contain a complete compromised message, as well as the one transferred from the TSS.

Recommended identification:

BC<primary key>_<secondary key>

The use of the primary key is not obligatory.

Available object types:

Object type	Schema file	Short description
BinaryContainer	binary_container.xsd	Binary container, the contents of which can be specified on a project-specific basis.

3.14 Project-specific extensions

Project-specific extensions of communication modules are possible. The following must be taken into consideration:

- It must be possible to disengage project-specific data structures using uniquely identifiable project or customer-specific object types.

These newly introduced object types must be stored in separate schema definitions and relate to the standard XML data types.

To prevent ambiguousness, separate manufacturer-specific prefixes are placed before the object types used. In addition, the groups to which the data type belong are indicated. An underscore follows the prefix ("_")

Example:

Manufacturer "abc"
New object type: "xyz"
New resulting identifying object type: "abc_xyz".

- Extensions as part of protocol functions must be avoided under any circumstances.
- Extensions to existing data types must be avoided. Should it not be possible to avoid this, it is necessary to take into account the compatibility rules set out in the next chapter.

3.14.1 Compatibility rules

Extension rules for public data models (schema definitions) and the assigned object types:

- Elements and attributes may be added to existing schema definitions,
- New attributes and elements must be optional in order to ensure backwards compatibility.
- The consuming interface must respond flexibly to the availability of the newly entered elements/attributes.
- The consuming interface must be respond flexibly to any unknown elements/attributes and skip these upon encountering without any further assessment. This means unknown elements may not influence the behaviour of the decoding of the received XML structure.

- It is not permitted to delete attributes or elements from schema definitions.
- It is not permitted to modify the semantics of existing attributes or elements.

OCIT-C_Daten_V1.2_R1

Copyright © 2016 ODG & Partner
