



Open Communication Interface for Road Traffic Control Systems

Offene Schnittstellen für die Straßenverkehrstechnik

OCIT-C Center to Center Daten

OCIT-C_Daten_V2.1

OCIT Developer Group (ODG) & Partner

OCIT® ist eine registrierte Marke der Firmen AVT Stoye, Stührenberg, Swarco Traffic Systems und Yunex Traffic

OCIT-C Center to Center Daten

Dokument: OCIT-C Daten V2.1

Herausgeber: ODG & Partner

Kontakt: www.ocit.org

Copyright © 2025 ODG. Änderungen vorbehalten. Dokumente mit Versions- oder Ausgabe-
stände neueren Datums ersetzen alle Inhalte vorhergehender Versionen.

Inhalt

1	Einführung	8
2	Objekte und Schemadefinitionen	11
2.1	Objekt	11
2.2	Objekttypen.....	11
2.3	Objektidentifikation.....	11
2.3.1	Objektidentifikation	12
2.3.2	Objektfilter	13
2.4	Gemeinsame Strukturen (global.xsd).....	13
2.4.1	Element Description	14
2.4.2	Element Rel_Ids.....	15
2.4.3	Element Location	15
2.4.4	State	16
2.4.5	Timeline	17
3	Datenkatalog	18
3.1	Verkehrsmeldungen.....	18
3.2	Verkehrsdaten	19
3.2.1	Unterabschnitt, Abschnitte, Bereiche.....	21
3.2.2	Detektoren und Detektorgruppen	21
3.2.2.1	Objekttyp: TrafficData_detector_Description	21
3.2.2.2	Objekttyp: TrafficData_detector_currentValue	21
3.2.2.3	Objekttyp: TrafficData_detectorGroup_Description.....	23
3.2.2.4	Objekttyp: TrafficData_detectorGroup_currentValue	23
3.2.2.5	Objekttyp: TrafficData_detectorGroup_calculatedValue	23
3.2.2.6	Objekttyp: TrafficData_trafficSubSection_Description.....	23
3.2.2.7	Objekttyp: TrafficData_trafficSection_Description	24
3.2.2.8	Objekttyp: TrafficData_trafficArea_Description	24
3.2.2.9	Objekttyp: TrafficData_trafficSubSection_accumulatedValue	24
3.2.2.10	Objekttyp: TrafficData_trafficSection_accumulatedValue.....	24
3.2.2.11	Objekttyp: TrafficData_trafficArea_accumulatedValue	24
3.3	Parkdaten	25
3.4	Wetter- und Umweltdaten	26
3.5	Kameras	27
3.6	Situationen und Strategien.....	28
3.7	Betriebsmeldungen	30
3.8	Schilder.....	33
3.8.1	Datenmodell.....	33

3.8.2	Komplexe Schildinhalte	36
3.9	ÖV-Fahrgastinformation	37
3.10	Lichtsignalanlagen	38
3.10.1	Vordefinierte Kommandos (Predefined Commands) - Veraltet	39
3.10.2	Schaltbarkeiten (IntersectionFeatures)	40
3.10.3	Ausmodellierte Schaltungen (IntersectionCommand)	41
3.10.4	Steuerungs-Parameter (ControlMethodParameter)	42
3.10.5	LSA Zustand (IntersectionStatus)	42
3.10.6	SGTimes	46
3.10.7	IntersectionGroupDescription	46
3.11	LSA Rohdaten	46
3.11.1	Detektorflanken	47
3.11.2	Signalgruppenzustände	48
3.11.3	Digitale Ausgnge im Lichtsignalsteuergerät	49
3.11.4	Anwenderprogrammwerte (AP-Werte)	51
3.11.4.1	Abfrage beauftragter AP-Werte	53
3.11.4.2	Beauftragung abfragbarer AP-Werte	54
3.11.5	ÖV-Telegramme	55
3.11.5.1	R09-Telegramm	55
3.11.5.2	Erweitertes ÖV-Telegramm	55
3.11.6	Einzeldetektordaten	55
3.11.7	Objektyp: Signalgroup_Description	57
3.12	Car-2-X Kommunikation	57
3.12.1	Statusinformation	57
3.12.2	Priorisierung	57
3.12.2.1	CAM-R09-Telegramm	57
3.12.2.2	CAM-Prio-Telegramm	59
3.12.2.3	Oepnv-Meldepunktliste	60
3.12.2.4	Oepnv-Anforderung	60
3.12.3	Gefahrenmeldungen	61
3.12.3.1	Management_Container	62
3.12.3.2	Situation_Container	63
3.12.3.3	Location_Container	63
3.12.3.4	Alacarte_Container	64
3.12.3.5	TransmissionControl	64
3.12.4	Spat	65
3.12.4.1	Objektyp PredictedSpat	65
3.12.5	MAP Topologiedaten	66
3.13	LSA Versorgungsdaten	66
3.14	Binärcontainer	66

3.15	Projektbezogene Erweiterungen	67
3.15.1	Kompatibilitätsregeln	67
4	Begriffe und Abkürzungen	69

Dokumentenstand

Version Zustand	Datum	Verteiler	Kommentar
V2.0_A01	23.05.2018	PUBLIC	Für OCIT-C V2.0 ODG Homepage
V2.0_A02	09.04.2024	PUBLIC	3.3 Beschreibung ergänzt 3.7 OperatingMessages aktualisiert 3.8 Empfohlene Identifizierung ergänzt 3.10.4 ControlMethodParameter aktualisiert 3.10.5 Beschreibung der LSA-Zustände (1001 bis 1004, 1006, 1007) angepasst 3.10.5 LSA-Zustände 1009 bis 1016 ergänzt 3.11.4 AP-Wert DPV1 ergänzt 3.11.4 Beschreibung der Beauftragung / Filterung von AP Werten angepasst 3.11.5 R09-Telegramm ergänzt 3.12.2 Titel angepasst 3.12.2.3 Oepnv-Meldepunktliste: Textergänzung 3.12.2.4 Oepnv-Anforderung: Textergänzung
V2.1_A01	29.04.2024	PUBLIC	3.10.5 Beschreibung der LSA-Zustände (1005 und 1008) angepasst sowie Ergänzung des LSA-Zustands 1017
V2.1_A02	30.03.2025	PUBLIC	3.15 Vorgaben für „Projektbezogene Erweiterungen“ als nicht verpflichtend formuliert
V2.1_A03	01.10.2025	PUBLIC	3.10.3 Anpassung der Grafik und Ergänzung eines Hinweis-Textes 3.10.5 Anpassung der Tabellenüberschriften

1 Einführung

OCIT-C steht für Open Communication Interface for Road Traffic Control Systems – Center to Center. Mit OCIT-C werden die Funktionen zur Kommunikation zwischen zentralen Verkehrssteuerungs- und Verkehrslenkungssystemen abgedeckt:

- Verkehrsrechnerzentralen und Verkehrsmanagementzentralen (kommunal, regional, überregional)
- Verkehrsingenieurarbeitsplatz mit Verkehrsrechnerzentralen
- Parkleitsysteme, Parkhaussysteme
- Baustellenmanagementsysteme
- Lokale Internetanwender (städtische Informationen im Internet)

Die Definition und Pflege der Schnittstelle OCIT-C wird von der ODG und ihren Partnern durchgeführt.

Mit OCIT-C steht ein Standard zur Verfügung, der OCIT-O passgenau ergänzt. Mit OCIT-C und OCIT-O für die Kommunikation von Zentralen zu Feldgeräten werden alle Anforderungen von der Verkehrssteuerung bis hin zum übergeordneten Verkehrsmanagement abgedeckt.

OCIT-C orientiert sich konsequent an den praktischen Anforderungen. Durch niedrige Implementierungskosten ist der Einsatz auch für Lösungen mit schmalen Budgets geeignet.

Charakteristische Eigenschaften von OCIT-C sind:

- Auf dem Standard SOAP basierendes Austauschprotokoll mit einfachem Request-Response-Kommunikationsmuster (direktes Abfragen von Daten).
- Definition eines umfassenden, alle Teilbereiche der Verkehrssteuerung und Verkehrslenkung enthaltenen Datenmodells im Prozessdatenbereich, Verwendung des OCIT-C Versorgungsdatenmodells für LSA.
- Systemintegration und gewünschte Anpassungen werden vorab über Projektierung geregelt.
- Konformitätstests des Protokolls werden in einer über www.ocit.org bereitgestellten Testumgebung durchgeführt. Tests von gesamten Implementierungen (Protokoll und Dateninhalte) werden projektbezogen durchgeführt.
- Erweiterungen um DATEX II Bestandteile sind je nach Projektanforderungen möglich.

Die OCIT-C Schnittstelle ist offen und kann in diversen Systemen, vorwiegend im Bereich der Straßenverkehrstechnik, eingesetzt werden. Die Aufgabe dieses Dokuments ist es, die standardisierten, über OCIT-C übertragbaren Daten zu beschreiben. Es ist nicht die Aufgabe dieses Dokuments, das Protokoll zu beschreiben. Dieses ist im Dokument „OCIT-C Protokoll“ beschrieben.

Folgende Kommunikationsbausteine sind derzeit standardisiert:

- **Verkehrsmeldungen**
Dieser Kommunikationsbaustein dient zur Übertragung von Meldungen zu Baustellen, Veranstaltungen und Störungen mit den Attributen: Verortung, zeitliche Dauer, tägliche Gültigkeit, Status, Klassifizierung, Auswirkung, z. B. Kapazitätsminderung, Beschränkungen, z. B. Geschwindigkeit oder Gewicht, Referenzierung zu überlagernden Meldungen, Art der Erfassung (manuell oder automatisch) und freier Text.
- **Verkehrsdaten**
Dieser Kommunikationsbaustein dient zur Übertragung von Messwerten oder abgeleiteten Werten verschiedener Detektionseinrichtungen.
- **Parkdaten**
Dieser Kommunikationsbaustein dient zur Übertragung von Daten von Parkeinrichtungen: Bezeichnung, Verortung, Kurz-, Langzeitparker, Belegung, freie Plätze, prognostizierte Werte (Trend) und Status der Parkeinrichtung.
- **Wetter- und Umweltdaten**
Dieser Kommunikationsbaustein dient zur Übertragung von Daten entsprechender Erfassungseinrichtungen:
 - Sensortyp mit den Attributen Verortung, punkt- oder streckenbezogene Messwerte,
 - Daten entsprechend TLS: Niederschlag (Intensität, Wahrscheinlichkeit), Temperatur (Luft, Boden), Wind, Luftfeuchte, Luftdruck, Sichtweite, Zustand Fahrbahnoberfläche, Sonnenaufgang, Sonnenuntergang, Helligkeit und
 - ergänzende Daten: Strahlungsbilanz, Bewölkungsgrad, Ozon, Schwefeldioxid, Benzol, Ruß, Kohlenmonoxid, Staub, Wettervorhersage.
- **Kameras**
Dieser Kommunikationsbaustein dient zur Übertragung von Daten zur Steuerung von Überwachungskameras und zur Übertragung von Bildern.
- **Situationen und Strategien**
Dieser Kommunikationsbaustein dient zur Übertragung von Beschreibungen von verkehrlichen Situationen oder Steuerungsstrategien und Vorgaben von Maßnahmen zur strategischen Verkehrssteuerung.
- **Betriebsmeldungen**
Dieser Kommunikationsbaustein dient zur Übertragung betrieblicher Meldungen von verkehrlichen Einrichtungen, wie z. B. von Lichtsignalanlagen.
- **Schilder**
Dieser Kommunikationsbaustein dient zur Übertragung von Daten zur Steuerung von Segment – und Vollmatrixschildern: Rückmeldungen vom Schild (Status, Zeitstempel, Temperatur und Inhalt), Kommandos an das Schild (Inhalt). Der Inhalt wird unterschieden nach: Prismenwendern, Vollmatrixanzeigen, Anzeigequerschnitte und deren einzelnen Displays. Bei Vollmatrixanzeigen werden übertragen: komplettes Layout (Text- und Bildpositionen, Farben, Schriftart etc.) und der Inhalte der Layoutelemente.
- **ÖV-Fahrgastinformation**
Dieser Kommunikationsbaustein dient zur Übertragung von strecken-, linien- oder haltestellenbezogenen Daten des öffentlichen Nahverkehrs: Verortung, VDV

Fahrgastinformation, Haltestelle und Linie, Typ ÖV-Fahrzeug (Bus, Tram..), Sollankunft, Istankunft an Haltestelle, Status des ÖV-Fahrzeugs.

- **Lichtsignalanlagen (LSA)**

Dieser Kommunikationsbaustein dient zur Übertragung von Daten als Vorgabe für die Steuerung von Lichtsignalanlagen: Beschreibung, Istzustand, vordefinierte Schaltungen, Parameter für verkehrstechnische und umlaufbezogene Signalisierung.

- **LSA-Rohdaten**

Dieser Kommunikationsbaustein dient zur effizienten Übertragung großer Mengen diskreter Werte von Lichtsignalanlagen: Detektorflanken, Signalgruppenzustände, digitale Ausgänge, Anwenderprogrammwerte, ÖV-Daten (erweitertes R09 Telegramm), Einzeldetektordaten.

Hinweis: Alle weiteren für die Lichtsignalsteuerung relevanten Daten, die keine Massendaten / Rohdaten sind, finden sich verteilt in den anderen Schemadefinitionen.

- **Car-2-X Kommunikation**

Dieser Kommunikationsbaustein dient zur Übertragung der Daten für die Car-2-X Kommunikation zwischen den Zentralen.

- **LSA-Versorgungsdaten**

Dieser Kommunikationsbaustein dient zur Datenversorgung von Lichtsignalanlagen, basierend auf den Standards OCIT- Center to Center und OCIT-Outstation Version 2.

- **Binärcontainer**

Dieser Kommunikationsbaustein dient zur Übertragung von binär codierten Daten. Die mit übertragene Beschreibung (Typbezeichnung) identifiziert die Art der Daten im Container und ermöglicht damit dem Empfänger die Verarbeitung.

- **Projektspezifische Erweiterungen**

von Kommunikationsbausteinen sind möglich, müssen aber bestimmten Regeln entsprechen (siehe Kapitel 3.15).

Hinweis: Alle Daten, auch Massendaten, werden mit dem OCIT-C Protokoll übertragen.

2 Objekte und Schemadefinitionen

2.1 Objekt

Ein Objekt ist eine eindeutig zu identifizierende Einheit, auf die auszutauschende Daten bezogen werden.

Beispiel: eine konkrete Einheit, wie LSA, Detektor, Parkhaus, Verkehrsmeldung

2.2 Objekttypen

Ein Objekttyp kennzeichnet eine abfragbare oder konfigurierbare Einheit zwischen den Schnittstellenpartnern. Teilmengen der Objekttypen können nicht abgefragt werden.

Grundsätzlich werden Daten einer Datenart (z.B. Detektorwerte) in folgende Teilaspekte unterteilt. Diese Teilaspekte stellen im Folgenden Objekttypen – also abfragbare Einheiten dar:

- Beschreibende Daten sind ein Objekttyp
- Dynamische Daten sind ein oder mehrere Objekttypen
- Befehle sind ein eigener Objekttyp
- Objekttypen werden in den Schemadefinitionen definiert. Die Annotation der Wurzelemente der zu übertragenden Datenart dokumentieren den Identifizierungsstring des Objekttyps in der Form *objecttype: <Objekttypidentifizierung>*.

2.3 Objektidentifikation

Objektidentifizierung erfolgt in bis zu drei Stufen:

- Objekttyp
- Quellinformation (Source of the object). Diese Stufe ist Optional.
- Identifier (id)

Der eigentliche Identifier muss innerhalb des Objekttyps eindeutig sein (Ausnahme: Betriebsmeldungen, siehe 3.7). Wird die Quelle verwendet gilt, dass der Identifier innerhalb des Objekttyps und der Quelle eindeutig sein muss.

Identifier eines Objektes sind im Allgemeinen ausgehandeltes gemeinsames Wissen zwischen den Schnittstellenpartnern (z.B. Detektoren, Schilder, Parkhäuser).

Andere Objekttypen (z. B. Verkehrsmeldungen) müssen in Bezug über die Schnittstelle dynamisch erzeugt und gelöscht werden. Damit ist es nicht möglich, die Identifier für diese Art von Objekten vorzudefinieren. Für Verkehrsmeldungen werden Identifier und Quellinformation zur eindeutigen Identifizierung verwendet.

2.3.1 Objektidentifikation

Die Objektidentifizierung der OCIT-C ist hinsichtlich der Identifier offen für alle Datenquellen. Die IDs sind daher prinzipiell frei vergebbar.

OCIT I/O sieht für LSA-bezogene Daten bestimmte Ordnungskriterien vor. Um eine sinnvolle Abbildung der Ordnungskriterien für LSA-bezogene Daten zu ermöglichen, bietet sich eine Vorschrift für die Bildung der OCIT-C IDs an, abgeleitet aus den Ordnungskriterien von OCIT I/O. Diese Abbildung bietet sich vor allem bei Verwendung von OCIT-O als Protokoll zur Anbindung der Außenanlagen an.

Die ID eines Messwertes im Feld Identifier/Ident soll bei Objekten deren Messwert durch den Objekttyp eindeutig bestimmt ist, z.B. Verkehrsdaten (→ Kapitel 3.2), Lichtsignalanlagen (→ Kapitel 3.10) und Rohdaten (→ Kapitel 3.11), wie folgt aufgebaut sein:

- J<SystemNr>_<SubsystemNr>_<UnitNr>_<ObjektNr>

Dabei markiert das "J" diesen speziellen Aufbau des Primärschlüssels. Die Marke "J" darf nur im Zusammenhang mit dem hier beschriebenen Format verwendet werden.

Die Adressierung eines Messwertes im Feld Id wird hierbei nicht verwendet und wiederholt gewöhnlich den Wert des Feldes Identifier/Ident.^{1 2}

Bei AP-Werten (→ Kapitel 3.11.4) soll die ID im Feld Identifier/Ident wie folgt aufgebaut sein:

- J<SystemNr>_<SubsystemNr>_<UnitNr>_<OITD-Nummer>_<ObjektNr>

Hinweis: Die Bezeichner <SystemNr>, <SubsystemNr>, <UnitNr> und <ObjektNr> sind dem Standard "OCIT-I" entnommen, ebenso der Begriff der OITD-Nummer. Die OITD-Nummern sind in IP-Schreibweise (57.102) auszuführen.

Ist keine Objektnummer vorhanden oder ist es ein einmaliges Objekt wird „1“ als Objektnummer verwendet.

Für Systeme, welche weder SystemNr noch SubsystemNr im Primärschlüssel verwenden, sondern nur die UnitNr, ist folgender Aufbau ebenfalls zulässig:

- J<UnitNr>_<ObjektNr> für Verkehrsdaten und Lichtsignalanlagen
- J<UnitNr>_<OITD-Nummer>_<ObjektNr> für AP-Werte

Die <OITD-Nummer> soll wie folgt aufgebaut sein:

- <Member-Nummer>.<Unternummer>

Beispiele einer vollständigen Adressierung sind also:

- J1_12_22555_17
für den Detektor mit der Kanalnummer 17 am Knoten 1_12_22555

¹ Dies ermöglicht beispielsweise die Identifizierung eines alleinstehenden Messwertes (ohne den Protokollanteil aus protokoll.xsd) in Logdateien.

² Alternativ kann der Datenlieferant einen lokalen Identifier verwenden.

- J1_12_22555_57.102_3
für den AP-Wert „102“ der Verkehrs-Systeme AG (Member 57) am Knoten
1_12_22555;
der AP-Wert hat die Kanalnummer 3

Dieselben Beispiele sehen mit der kurzen Adressierung wie folgt aus:

- J22555_17
- J22555_57.102_3

2.3.2 Objektfilter

Beim Datenabruf kann eine optionale Filterliste bei den Methoden „inquireAll“ und „get“ des Protokolls angegeben werden. Die Objekte werden in der Filterliste über einen Identifier analog der Objektidentifikation adressiert. Diese Identifier bestehen aus:

- Quellinformation (Source of the object). Diese Stufe ist Optional.
- Identifier (ident)

Für die Identifier im Feld Identifier/Ident der Filterliste ist es zulässig nur einen Teil des verwendeten Primärschlüssels zu verwenden, wobei der Primärschlüssel im Server von links beginnen gefiltert wird.

Beispiel eines Filter-Identifiers für den Objekttyp TrafficData_detector_currentValue:

- J1_12_22555
für alle Detektoren am Knoten 1_12_22555
- J1_12
für alle Detektoren des Subsystem 12 am System 1
- J1_12_22555_17
Nur die Messwerte des Detektors mit der Kanalnummer 17 am Knoten 1_12_22555

Wird die kurze Adressierung verwendet kann nur auf die Unitnummer gefiltert werden:

- J22555
für alle Detektoren am Knoten 22555
- J22555_17
Nur die Messwerte des Detektors mit der Kanalnummer 17 am Knoten 22555

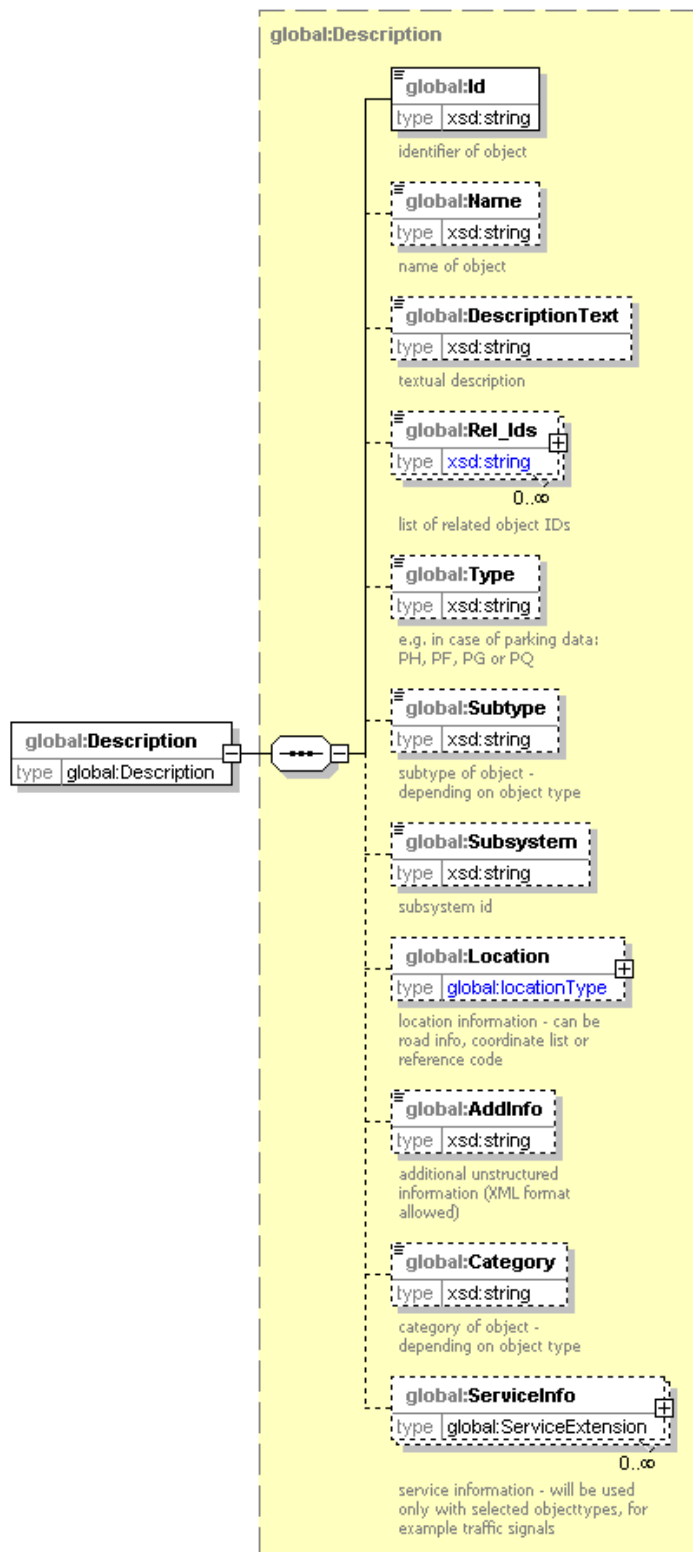
2.4 Gemeinsame Strukturen (global.xsd)

Die genaue Beschreibung des Datenmodells, sowie die elementare Beschreibung der Attribute und Strukturelemente erfolgt vollständig innerhalb der einzelnen Schemadefinitionen in Form von XML Schemadefinitionen (XSD).

Die gemeinsam verwendeten Datenstrukturen werden in *global.xsd* definiert. Die Struktur von *global.xsd* wird hier kurz erläutert.

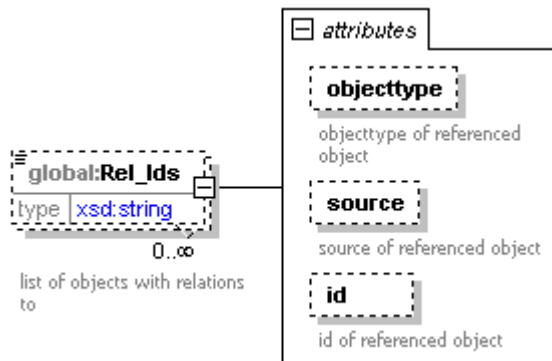
2.4.1 Element Description

In diese Struktur werden alle beschreibenden Daten eingetragen. Diese Datenstruktur wird durch die domänenspezifischen Objekttypen referenziert.



2.4.2 Element Rel_Ids

Das Strukturelement Rel_Ids erlaubt die Referenzierung bzw. die Topologie unter Objekten zu beschreiben. Demnach verweisen diese Strukturelemente auf andere Objekte und umfassen demnach die komplette Identifizierung aus Objekttyp, Quelle und id.

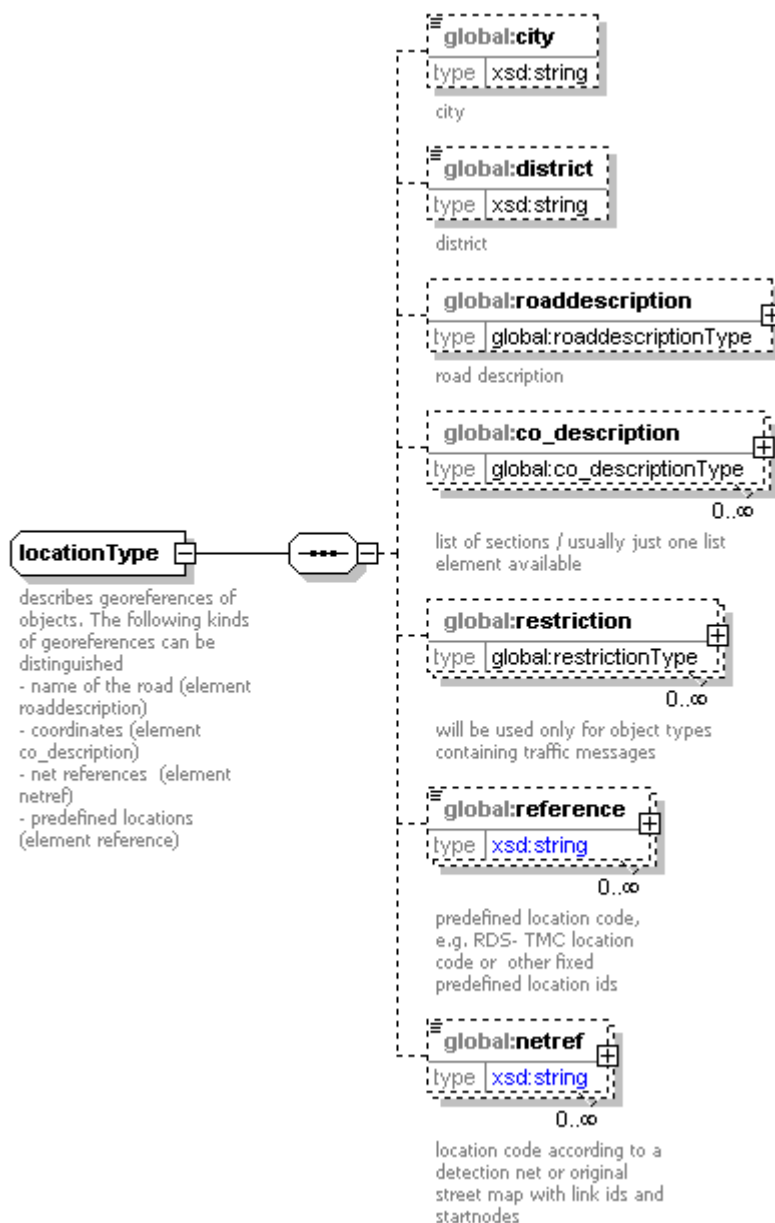


Name	Typ	Verwendung	Beschreibung
objecttype	xsd:string	optional	Objekttyp des referenzierten Objekts
source	xsd:string	optional	Quelle des referenzierten Objekts
id	xsd:string	optional	Identifizierung des referenzierten Objekts

2.4.3 Element Location

Dieses Strukturelement erlaubt die Georeferenzierung des Objektes. Dies kann auf unterschiedliche Art und Weise erfolgen:

- Textuelle Beschreibung über Straßen und Kreuzungsbezeichnung
- Koordinaten
- Vordefinierte Georeferenzen (z.B. TMC-Locations)
- Referenzierung auf Netze (link-ids)



Diese Struktur ist in jeder Objektbeschreibung enthalten. Die Belegung muss in dem jeweiligen Projekt abgestimmt werden.

2.4.4 State

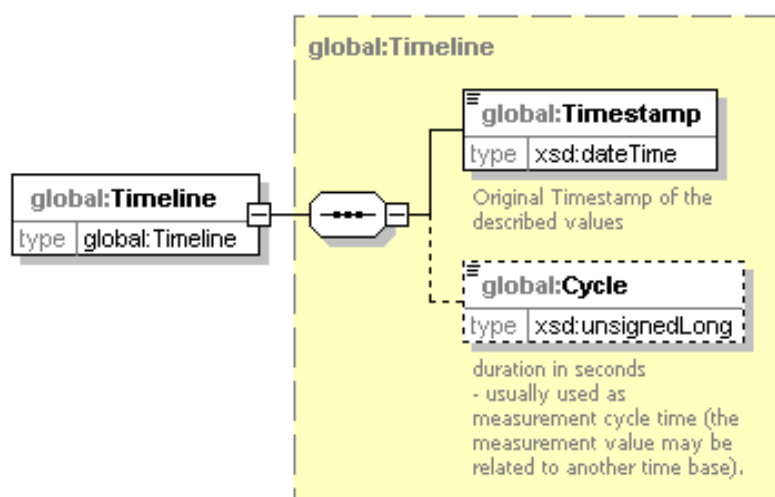
Dynamische Daten haben einen Status. Dieser wird mittels State dargestellt. Alle dynamischen Daten verwenden die global definierte *State*.



Art	Wert	Beschreibung
enumeration	o.k.	Feldgerät o.k.: Werte können verwendet werden
enumeration	n.o.k.	Feldgerät nicht o.k.: Werte können nicht verwendet werden
enumeration	p.o.k.	Feldgerät teilweise o.k.: Es könnte ein Problem bei der Verwendung der Werte geben
enumeration	unknown	Feldgeräatzustand ist unbekannt: Werte sollen nicht verwendet werden
enumeration	offline	Feldgerät ist nicht verbunden.
enumeration	busy	Geforderter Wert ist ungleicher gegenwärtiger Wert (z. B. weil gerade eine Schaltung durchgeführt wird).
enumeration	substitute_value	Gerätesoftware ersetzt einen Wert (z.B. Umschaltung auf ein neues Geräteprofil)

2.4.5 Timeline

Dynamische Daten haben einen zeitlichen Bezug. Dieser wird mittels Zeitstempel hergestellt. Alle dynamischen Daten verwenden die global definierte Timeline.



3 Datenkatalog

Die hier beschriebenen Objekttypen sind die innerhalb OCIT-C definierten Datenarten.

Zur Erläuterung der Datenstrukturen werden in diesem Dokument graphische Darstellungen der Schemadefinitionen verwendet. Detailinformationen dazu finden sich nur in den Schemadefinitionen. Dort werden die hier referenzierten Schemadateien (.xsd) in Textform dargestellt.

Die Schemadefinitionen wurden mit Ausnahme von *intersection_config_data* (LSA-Versorgungsdaten mit dem vorwiegenden Einsatzgebiet im deutschsprachigen Bereich) in englischer Sprache erarbeitet und werden nicht übersetzt.

Alle hier beschriebenen Daten werden beginnend von ihrem Wurzelement in die protocol.xsd innerhalb des Elementes "data" eingefügt. Wurzelement ist jeweils das Element, welches innerhalb der Schemadefinition in der Annotation des Elementes mit dem Kommentar ("objecttype") gekennzeichnet ist. Gleichzeitig dient der darin aufgeführte Objekttyp als Identifizierungsmerkmal dieser Datenart.

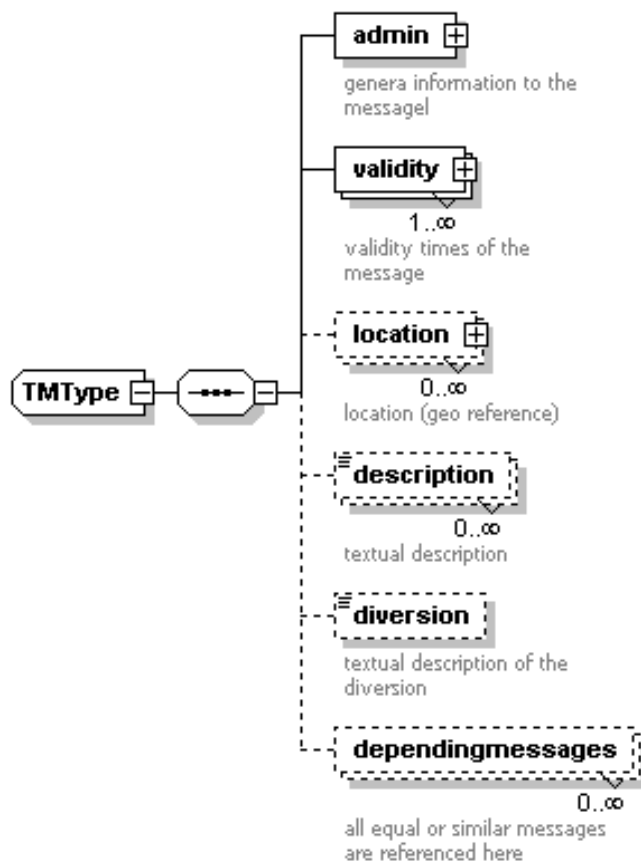
3.1 Verkehrsmeldungen

Dieser Kommunikationsbaustein dient der Übertragung von Meldungen zu Baustellen, Veranstaltungen und Störungen.

Verkehrsmeldungen werden mittels id und Quellinformation identifiziert.

Verfügbare Objekttypen:

Objekttyp	Schemadatei	Kurzbeschreibung
TrafficMessage_RoadWorks	traffic_messages.xsd	Baustellen
TrafficMessage_Events		Veranstaltungen
TrafficMessage_Incidents		Störungen



3.2 Verkehrsdaten

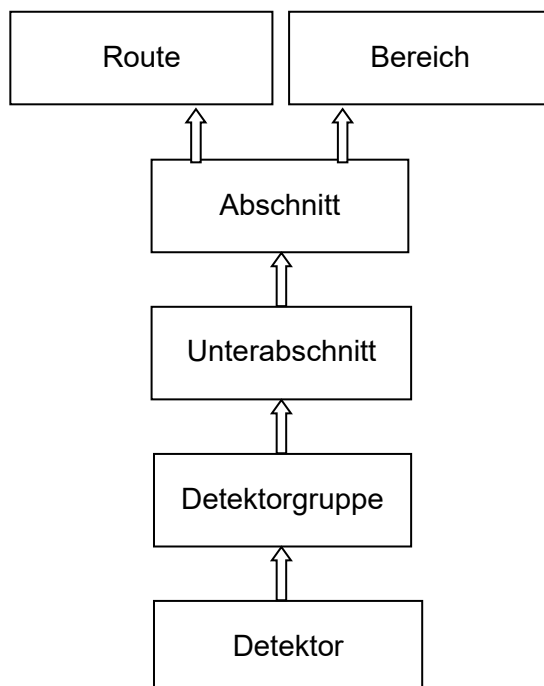
Dieser Kommunikationsbaustein dient zur Übertragung von Messwerten oder abgeleiteten Werten verschiedener Detektionseinrichtungen. Übertragen werden aktuelle und akkumulierte Messergebnisse von Einzeldetektoren oder Detektorgruppen, jedoch keine Detektorrohdaten (Impulse).

Verfügbare Objekttypen:

Objekttyp	Schemadatei	Kurzbeschreibung
TrafficData_detector_Description	Traffic_data.xsd	Beschreibende Daten Detektoren
TrafficData_detector_currentValue		Aktuelle Daten Detektoren
TrafficData_detectorGroup_Description		Beschreibende Daten Messstellen
TrafficData_detectorGroup_currentValue		Aktuelle Daten einer Messstelle
TrafficData_detectorGroup_calculatedValue		Abgeleitete Daten einer Messstelle
TrafficData_trafficSubSection_Description		Beschreibende Daten Abschnitte

Objekttyp	Schemadatei	Kurzbeschreibung
TrafficData_trafficSection_Description		Beschreibende Daten Strecken
TrafficData_trafficArea_Description		Beschreibende Daten Bereiche
TrafficData_trafficRoute_Description		Beschreibende Daten Routen
TrafficData_trafficSubSection_accumulatedValue		Dynamische Werte Abschnitte
TrafficData_trafficSection_accumulatedValue		Dynamische Werte Strecken
TrafficData_trafficArea_accumulatedValue		Dynamische Werte Bereiche
TrafficData_trafficRoute_accumulatedValue		Dynamische Werte Routen

Die folgende Abbildung zeigt die unterschiedlichen Hierarchien, denen die entsprechenden Objekttypen zugeordnet sind.



3.2.1 Unterabschnitt, Abschnitte, Bereiche

Unterabschnitt, Abschnitte und Bereiche (= Subsection, sections und traffic areas) werden mittels vordefinierter Ids referenziert.

Empfohlene Identifizierungen:

SS<primary key>_<secondary_key> für Unterabschnitt
S<primary key>_<secondary_key> for Abschnitt
R<primary key>_<secondary_key> for routes
TA<primary key>_<secondary_key> for traffic areas

Die Verwendung des <primary key> ist nicht zwingend erforderlich.

3.2.2 Detektoren und Detektorgruppen

Detektoren (z.B. Schleifen) und Messstellen (=Detektorgruppen) werden mit vordefinierten Ids identifiziert.

Empfohlene Identifizierungen:

- Für Detektionsmittel, welche über OCIT-O angeschlossen sind;
Objektidentifikation gemäß Abschnitt 2.3.1
- Für beliebige, andere Detektionsmittel:

D<primary key>_<secondary_key> für Detektoren
DG<primary key>_<secondary_key> für Messstellen

Die Verwendung des <primary key> ist nicht zwingend erforderlich.

Hinweis: Zur Identifizierung von Detektoren, die über OCIT-I VD versorgt werden, wird nur der nach dem im Kapitel 3.10.7 beschriebenen System gebildete Primary Key verwendet

3.2.2.1 Objekttyp: TrafficData_detector_Description

Die Datenstruktur wird im Kapitel 2.4.1 erklärt.

Verwendung des Element **type**

- Beschreibung Detektortyp: Schleife, Doppelschleife, Kamera ...

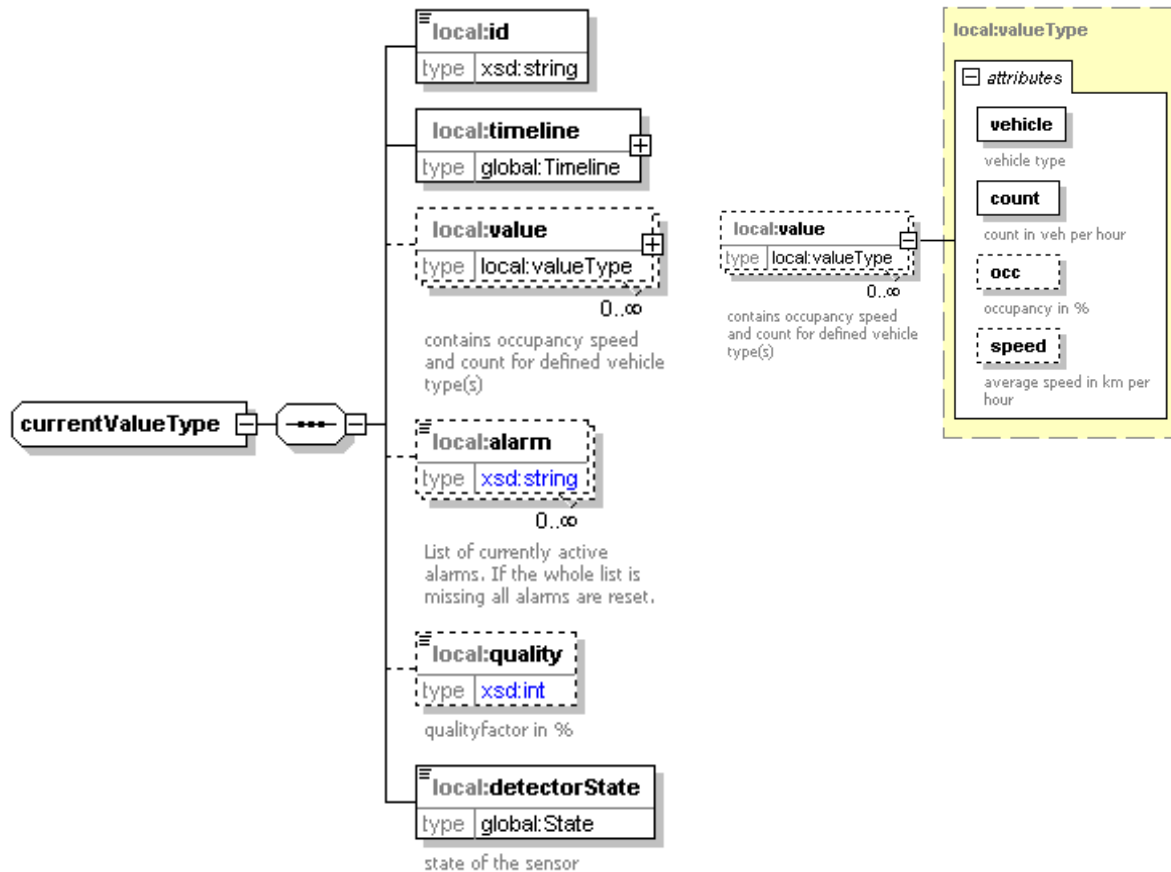
Verwendung Element **subsystem**

- Hersteller/Lieferant

Verwendung Element **Rel_Ids**

- Verweis auf Messstellen
- Verweis auf LSA

3.2.2.2 Objekttyp: TrafficData_detector_currentValue



Für jeden verfügbaren Fahrzeugtyp wird die Struktur „value“ als Feldelement aufgenommen. Nichtvorhandene Fahrzeugtypen sollen nicht als Feldelement aufgenommen werden.

Sobald gültige Werte vorliegen, wird der Fahrzeugtyp „all“ geliefert.

Fahrzeugtypen:

Fahrzeugtyp in OCIT-C	Entsprechung in TLS
<xsd:enumeration value="all"/>	Summe
<xsd:enumeration value="car"/>	Pkw
<xsd:enumeration value="truck"/>	Lkw
<xsd:enumeration value="artic_truck"/>	Sattelkraftfahrzeug
<xsd:enumeration value="motorcycle"/>	Motorrad
<xsd:enumeration value="unclassified"/>	“Phantom”
<xsd:enumeration value="car_with_trailer"/>	Pkw mit Anhänger
<xsd:enumeration value="truck_with_trailer"/>	Lkw mit Anhänger
<xsd:enumeration value="bus"/>	Bus

Fahrzeugtyp in OCIT-C	Entsprechung in TLS
<xsd:enumeration value="delivery_truck"/>	Lieferwagen



Alarmliste:

"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"

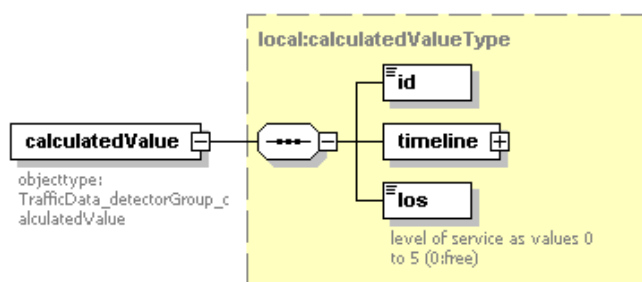
3.2.2.3 Objekttyp: TrafficData_detectorGroup_Description

Verwendung wie TrafficData_detector_Description

3.2.2.4 Objekttyp: TrafficData_detectorGroup_currentValue

Verwendung wie TrafficData_detector_currentValue

3.2.2.5 Objekttyp: TrafficData_detectorGroup_calculatedValue



3.2.2.6 Objekttyp: TrafficData_trafficSubSection_Description

Die Datenstruktur wird im Kapitel 2.4.1 erklärt.

Verwendung Element **subsystem**

- Beschreibung des Datenversorgers

Verwendung Element **Rel_Ids**

- Beschreibung der zugeordneten Detektorgruppen oder anderer Daten Verkehrsdaten

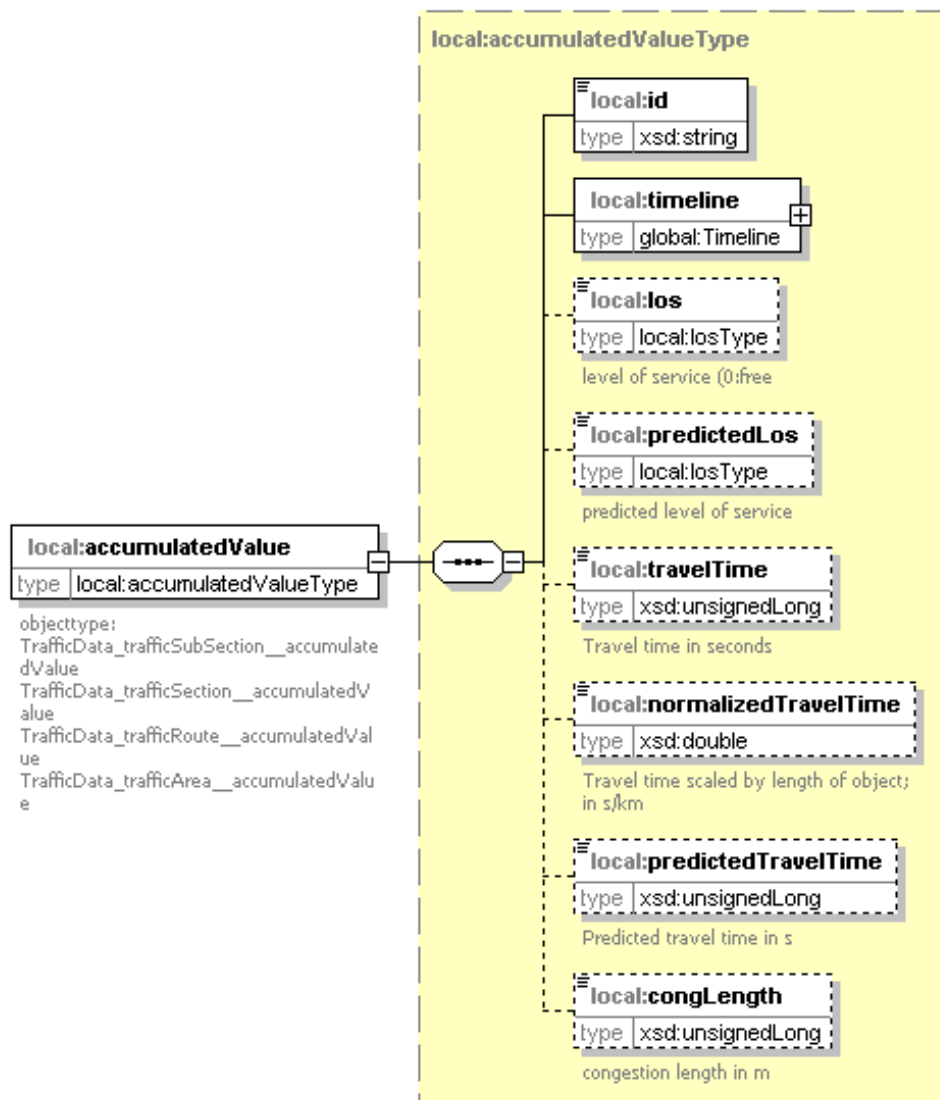
3.2.2.7 Objekttyp: TrafficData_trafficSection_Description

Verwendung wie TrafficData_trafficSubSection_Description

3.2.2.8 Objekttyp: TrafficData_trafficArea_Description

Verwendung wie TrafficData_trafficSubSection_Description

3.2.2.9 Objekttyp: TrafficData_trafficSubSection_accumulatedValue



3.2.2.10 Objekttyp: TrafficData_trafficSection_accumulatedValue

Verwendung wie TrafficData_trafficSubSection_accumulatedValue

3.2.2.11 Objekttyp: TrafficData_trafficArea_accumulatedValue

Verwendung wie TrafficData_trafficSubSection_accumulatedValue

3.3 Parkdaten

Dieser Kommunikationsbaustein dient zur Übertragung von Daten von Parkeinrichtungen: Beschreibung und aktuelle Auslastung von Parkeinrichtungen³, Parkflächen⁴, Parkquartieren⁵.

Parkdaten werden mit vordefinierten Ids referenziert.

Empfohlene Identifizierung:

PF<primary key>_<secondary_key> für Parkeinrichtungen

PD<primary key>_<secondary_key> für Parkbezirke und Parkplätze

Die Verwendung des primary key ist nicht zwingend erforderlich.

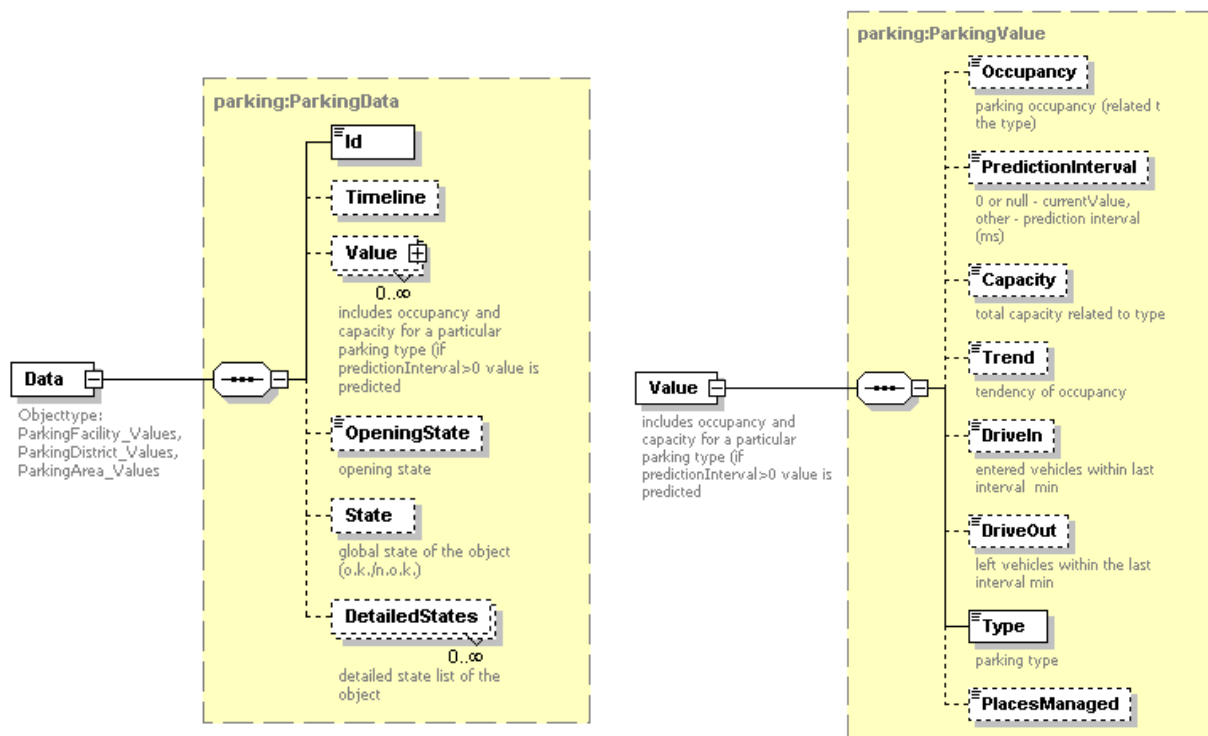
Verfügbare Objekttypen:

Objekttyp	Schemadatei	Kurzbeschreibung
ParkingFacility_Description	parking.xsd	Beschreibende Daten Parkeinrichtung
ParkingArea_Description		Beschreibende Daten Parkfläche
ParkingDistrict_Description		Beschreibende Daten Parkquartier
ParkingFacility_Values		Dynamische Daten Parkeinrichtung
ParkingArea_Values		Dynamische Daten Parkfläche
ParkingDistrict_Values		Dynamische Daten Parkquartier

³ ParkingFacility: Einzelne Parkeinrichtungen wie Parkhäuser, Tiefgaragen, beschränkte Parkflächen

⁴ ParkingArea: Parkflächen im Straßenraum, z.B. Parklizenzbereiche mit Parkscheinautomaten o.ä.

⁵ ParkingDistrict: Aggregation von ParkingFacilities und/oder ParkingAreas, z.B. "Parken in der Altstadt"



3.4 Wetter- und Umweltdaten

Dieser Kommunikationsbaustein dient zur Übertragung von Daten entsprechender Erfassungseinrichtungen.

Umwelt- und Wetterdaten werden mit vordefinierten Ids identifiziert.

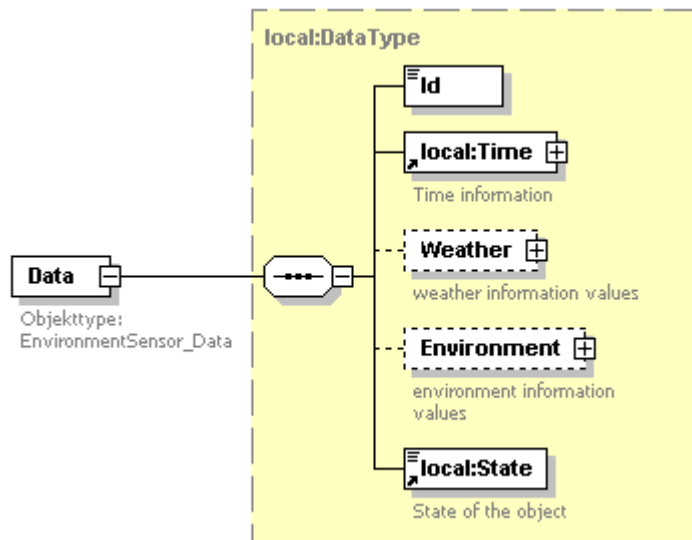
Empfohlene Identifizierung:

W<primary key>_<secondary_key> für Wettersensoren
 E<primary key>_<secondary_key> für Umweltsensoren

Die Verwendung des primary key ist nicht zwingend.

Verfügbare Objekttypen:

Objekttyp	Schemadatei	Kurzbeschreibung
EnvironmentSensor_Descr	environmentsensor.xsd	Beschreibende Daten Wetter oder Umwelt
EnvironmentSensor_Data		Dynamische Daten Wetter oder Umwelt



3.5 Kameras

Dieser Kommunikationsbaustein dient zur Übertragung von Daten zur Steuerung von Überwachungskameras und zur Übertragung von Bildern.

Kameradaten werden mit vordefinierten Ids identifiziert.

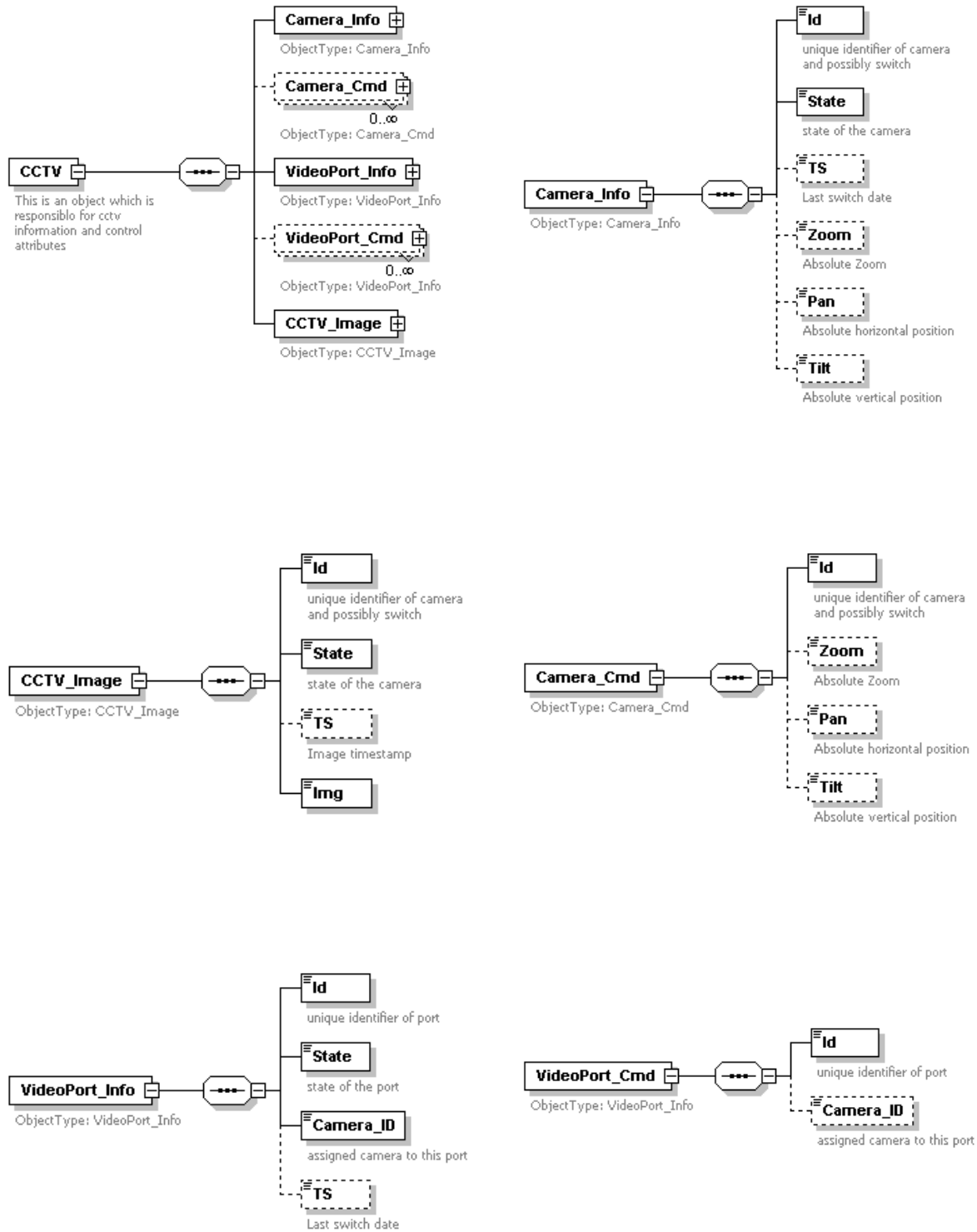
Empfohlene Identifizierung:

Cam<primary key>_<secondary key>

Die Verwendung des primary key ist nicht zwingend.

Verfügbare Objekttypen:

Objekttyp	Schemadatei	Kurzbeschreibung
CCTV_Description	cctv.xsd	Beschreibende Daten Kamera
Camera_Info		Aktueller Zustand Kamera (ohne Bild)
CCTV_Image		Aktuelles Bild einer Kamera
Camera_Cmd		Kommando/Stellbefehl an Kamera
VideoPort_Info		Aktueller Zustand Monitormatrix
VideoPort_Cmd		Kommando/Stellbefehl Monitormatrix



3.6 Situationen und Strategien

Dieser Kommunikationsbaustein dient zur Übertragung von Beschreibungen von verkehrlichen Situationen oder Steuerungsstrategien und Vorgaben von Maßnahmen zur strategischen Verkehrssteuerung.

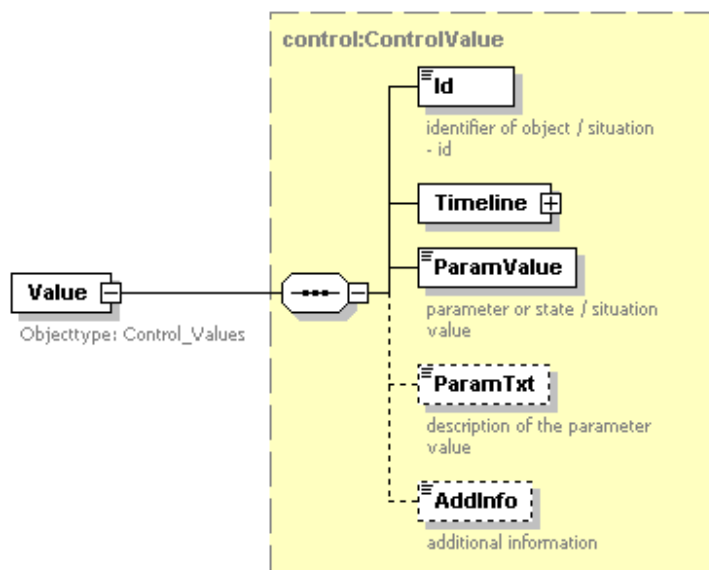
Situationen oder Strategien werden mit vordefinierten Ids identifiziert.

Empfohlene Identifizierung:

Sit<primary key>_<secondary_key>

Verfügbare Objekttypen:

Objekttyp	Schemadatei	Kurzbeschreibung
Control_Description	Control.xsd	Beschreibende Daten Situation oder Strategie
Control_Values		Aktuelle Strategie/Situation



AddInfo kann für projektbezogene Zusatzinformationen belegt werden.

Anwendungsbeispiele:

- **Strategie**
Die Id identifiziert die Strategie, über ParamValue kann die Variante der Strategie modelliert werden.
- **Situation.**
Die Id identifiziert die Situation (z.B. den Zustand einer Türe), über ParamValue kann der Zustand dieser Türe modelliert werden (z.B. Tür auf oder Tür zu).
- **Digitalkontakt**
Die Id identifiziert den Digitalkontakt, über ParamValue kann der Zustand dieses Kontaktes modelliert werden (z.B. high level oder low level).
- **Gruppen**
Die Id identifiziert die Gruppe (z.B. die LSA Gruppe), über ParamValue kann der Zustand dieser Gruppe modelliert werden (z.B. Gruppe aus oder Gruppe in SP1 oder Gruppe mit bestimmten Bild beschalten).

3.7 Betriebsmeldungen

Dieser Kommunikationsbaustein dient zur Übertragung betrieblicher Meldungen von verkehrlichen Einrichtungen, wie z. B. von Lichtsignalanlagen.

Verfügbare Objekttypen:

Objekttyp	Schemadatei	Kurzbeschreibung
OperatingMessages_Operatingmessage	operating_messages.xsd	Betriebsmeldung

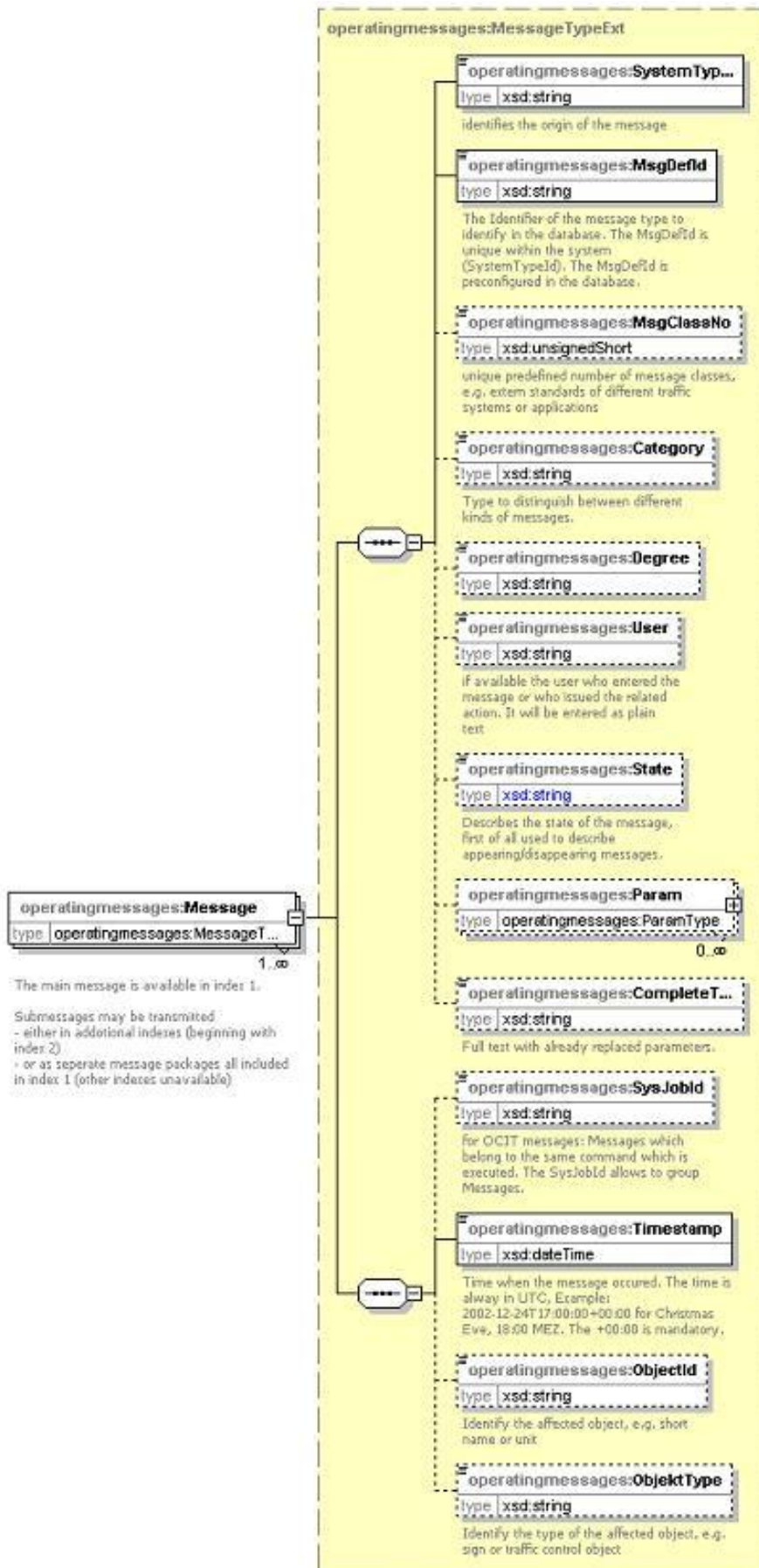
Betriebsmeldungen verfügen über eine Hauptmeldung und optional über mehrere Nebemeldungen. Die Hauptmeldung steht an erster Stelle. Anschließend folgen die Nebemeldungen. Die Meldungsbestandteile einer Haupt- bzw. Nebemeldung werden im Folgenden beschrieben

Die Identifizierung erfolgt über den Objektidentifizier des Objekts, zu dem diese Meldung gehört. Eine Abfrage identifiziert daher ein Objekt bzw. ein Gerät (nicht eine Meldung) und kann in der Response mehrere verschiedene Meldungsarten (MsgDefId) liefern.

Beispiele für LSA-bezogene Meldungen (siehe auch: 2.3.1): „J1_12_22555_1“

Beispiel für Meldungen zu einem Schild: „Schild_Bahnhofstr._Nord“

Der Objektidentifizier der Meldung muss identisch sein mit dem Objektidentifizier, mit dem auch die dynamischen Daten des Objekts adressiert werden.



Tag	Beschreibung
SystemTypeID	Ursprung der Betriebsmeldung (bei OCIT-O entspricht dies der Membernummer)
MsgDefID	Meldungstyp zur Identifizierung der Betriebsmeldung. Der Meldungstyp ist eindeutig innerhalb des Systems (SystemTypfeld) und innerhalb der Datenbank vorkonfiguriert (bei OCIT-O entspricht dies dem OType).
MsgClassNo	Meldungsklasse zur Unterscheidung verschiedener Meldungsgruppen eines Herstellers
Category	Unterscheidungsmerkmal zwischen verschiedenen Meldungsarten
Degree	Schweregrad der Meldung
User	Wenn verfügbar, der Benutzer eingetragen, der die Meldung eingegeben hat oder die zugehörige Aktion ausgeführt hat.
State	Beschreibt den Zustand der Nachricht, vor allem zur Darstellung der Meldungen.
Param	Parameter zur Präzisierung der Betriebsmeldung. Die Parameterarten sind innerhalb der Datenbank vordefiniert.
CompleteText	Vollständiger Text der Betriebsmeldung inklusive der Parameter
SysJobld	Bei Meldungen die Ursache einer bestimmten Aktion sind, tragen sowohl die Aktion als auch die Meldung die gleiche SysJobld. Die SysJobld gruppiert die Meldungen nach bestimmten Ursachen / Verursachern.
Timestamp	Zeitstempel vom Entstehungszeitpunkt der Meldung
Objectld	Identifizierung des betreffenden Objekts, z. B. Kurzbezeichnung
Objektyp	Identifizierung des betroffenen Objekttyps, z. B. Schild oder Verkehrssteuerungsobjekt

Man kann verschachtelte und Array-Parameter dadurch angeben, dass man die Paramld entsprechend füllt: "x1", "x2", ..., "y1a", "y1b", "y2a", ... Dazu muss eine Vorschrift definiert werden, wie Klassen- oder Feldnamen und Array-Indizes entsprechend kombiniert werden, um eine eindeutige Paramld zu erzeugen.

3.8 Schilder

Dieser Kommunikationsbaustein dient zur Übertragung von Daten zur Steuerung von Segment – und Vollmatrixschildern.

Empfohlene Identifizierung:

IP<SystemNr>_<SubsystemNr>_<UnitNr>_<ObjektNr>

Verfügbare Objekttypen:

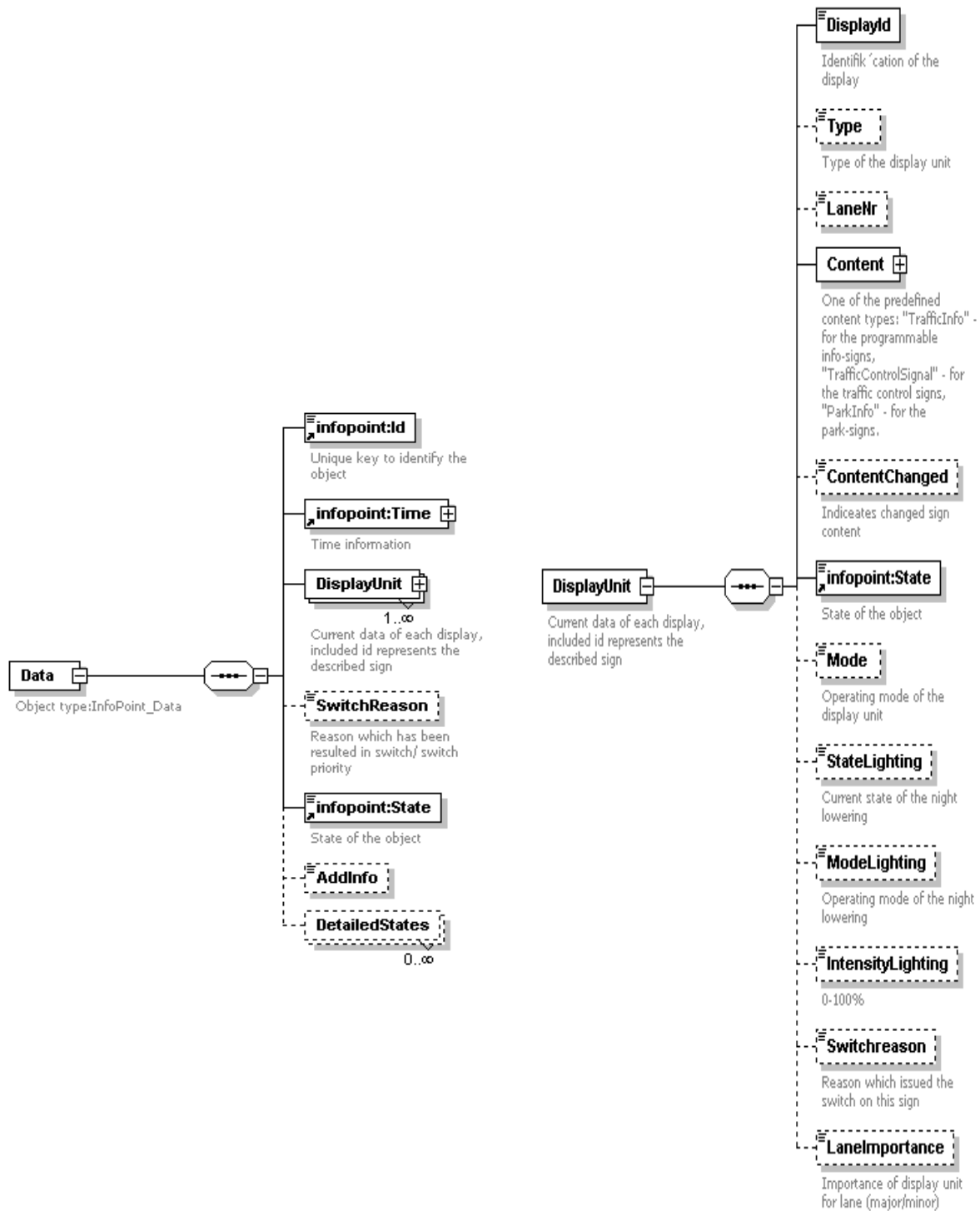
Objekttyp	Schemadatei	Kurzbeschreibung
Infopoint_Description	infopoint_data.xsd	Beschreibende Daten Schilder
InfoPoint_Data	(importiert sign.xsd)	Schilderinhalt (Iststand)
InfoPoint_Data		Schilderinhalt (Sollzustand)
Infopoint_Cmd		Alternative Stellbefehl für Schilder

Die *sign.xsd* findet im Falle komplexer Matrixschilder Anwendung.

3.8.1 Datenmodell

Das im Folgenden dargestellte Datenmodell wird zum Austausch von Schildinhalten wie auch für das Schalten von Schildern verwendet.

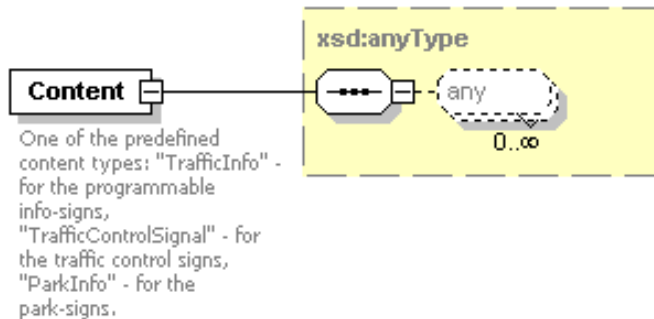
Das XML Element "state" wird nur zum Austausch von Schildinhalten verwendet. Zum Schalten von Schildern spielt dieses Element keine Rolle.



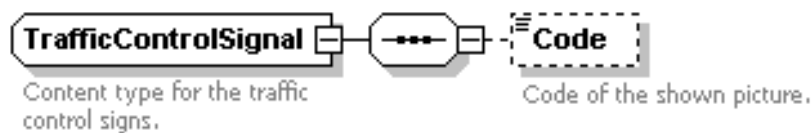
Tag	Inhalt	Beschreibung
Id		Identifizierung des Schildes
Time	Timestamp	In UTC
DisplayUnit	sub elements	Tritt nur auf - einmal bei Vollmatrix-Displays - mehrmals bei Segment-Displays
DisplayId	-	Kennzeichnet Segment-Displays
Type	-	Unbenutzt
Content	Wie in sign.xsd definiert	Siehe sign.xsd
ContentChanged	-	Unbenutzt
State	-	Unbenutzt an dieser Stelle
Mode	-	Unterscheidung, ob der Anteil vom Schild von der Zentrale oder lokal angesteuert wird
StateLighting		Unbenutzt
SwitchReason	-	Unbenutzt
State	o.k. n.o.k. busy	Statusinformation vom Schilder-Server
	switch	Kommandoanzeige vom Schild zum Schilder-Server
AddInfo		Zusätzliche Information (optional).
On_off	on off	

3.8.2 Komplexe Schildinhalte

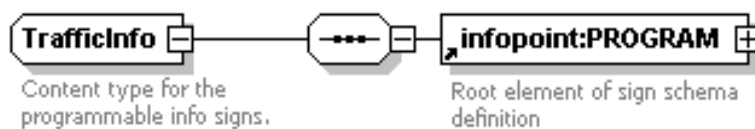
Die sign.xsd modelliert komplexe Schildinhalte. Die Verbindung zum Schema infopoint_data.xsd wird mittels anyType realisiert:



Das Feld "any" kann mittels vordefinierter Codes welche den Schildinhalt identifizieren (simple predefined code selection)



- eine programmierbare Matrixdefinition oder base 64 kodierter picture stream

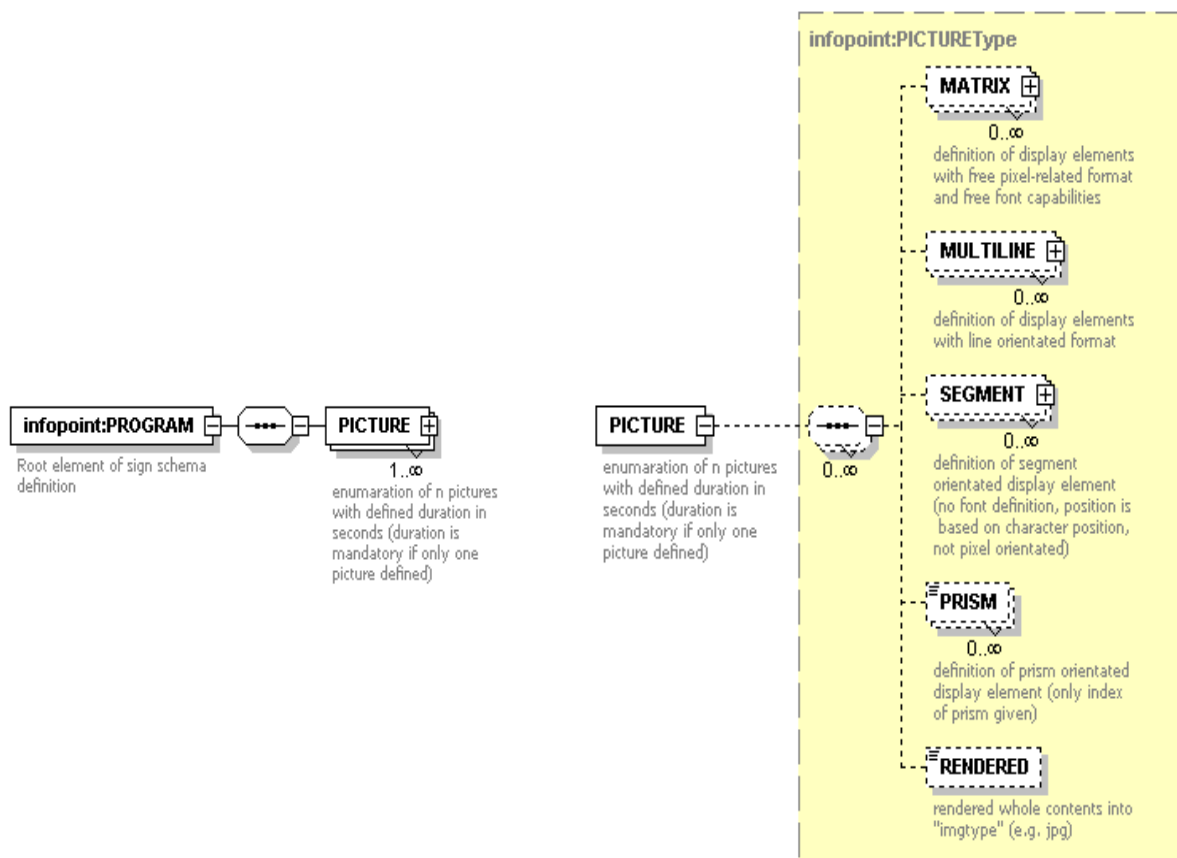


- eine Parkinformation beschreiben.

Der Inhalt ist eine Zeichenkette, z.B. ein Text oder ein Restplatzwert oder eine Prismenseite. Das Attribut ContentType kann die Verwendung der Zeichenkette näher beschreiben.

- Programm Definition

erlaubt die Definition von Bildsequenzen:



3.9 ÖV-Fahrgastinformation

Dieser Kommunikationsbaustein dient zur Übertragung von strecken-, linien- oder haltestellenbezogenen Daten des öffentlichen Nahverkehrs.

Publictransport_data wird mit vordefinierten Ids referenziert.

Empfohlene Identifizierung:

PT<primary key>_<secondary_key> for junctions

Die Verwendung des primary key ist nicht zwingend erforderlich.

Verfügbare Objekttypen:

Objekttyp	Schemadatei	Kurzbeschreibung
PT_Description	publictransport_data.xsd	
PT_Link		Streckenbezogene Fahrgastinformation: Anzahl, Geschwindigkeit und Verspätung
PT_LinePerLinkLink		Linienbezogene Fahrgastinformation: Anzahl, Geschwindigkeit und Verspätung

3.10 Lichtsignalanlagen

Dieser Kommunikationsbaustein dient zur Übertragung von Daten als Vorgabe für die Steuerung von Lichtsignalanlagen.

Intersection_data wird mit vordefinierten Ids referenziert.

Empfohlene Identifizierung:

J<primary key>_<secondary_key> for junctions

Die Verwendung des primary key ist nicht zwingend erforderlich.

Verfügbare Objekttypen:

Objekttyp	Schemadatei	Kurzbeschreibung
IntersectionDescription	intersection_data.xsd	Beschreibende Daten LSA
IntersectionStatus		Aktuelle LSA Daten (Iststand)
PredefCommand		Vordefinierte LSA Schaltungen
IntersectionFeature		Schaltbarkeiten einer LSA
IntersectionCommand		LSA Schaltung (ausmodelliert als Ersatz der PredefCommand)
ControlMethodParameters		Dynamische Parameter für verkehrstechnische Verfahren
SGTimes		Auf einen Umlauf zusammengefasste Rot- und Grünzeiten

3.10.1 Vordefinierte Kommandos (Predefined Commands) - Veraltet

Mit Einführung ausmodellierter Kommandos werden vordefinierte Kommandos überflüssig. Sie werden Übergangsweise aus Gründen der Kompatibilität hier weiter aufgeführt.

Folgende vordefinierten Kommandos stehen zur Verfügung:

Kommando	Belegung des XML Elements
Knoten ein/aus	on off local
SPx (Knoten wird implizit eingeschaltet) x ist die Programmnummer	plan1 plan2 (planx) local plan
IV ein/aus	it_on it_off it_local
ÖV ein/aus	pt_on pt_off pt_local
Lokale VA ein/aus	localta_on localta_off localta_local
Teilknoten ein/aus	Subintersection0_on; subintersection0_off; subintersection0_local subintersection1_on; subintersection1_off; subintersection1_local subintersection2_on; subintersection2_off; subintersection2_local subintersection3_on; subintersection3_off; subintersection3_local
Sonderkommandos	specialcommand<number> localspecialcommand<number>
Projektspezifische Modifikation	projectspecificcommand_<name or number> localprojectspecificcommand_<name or number>
Freigabe	release

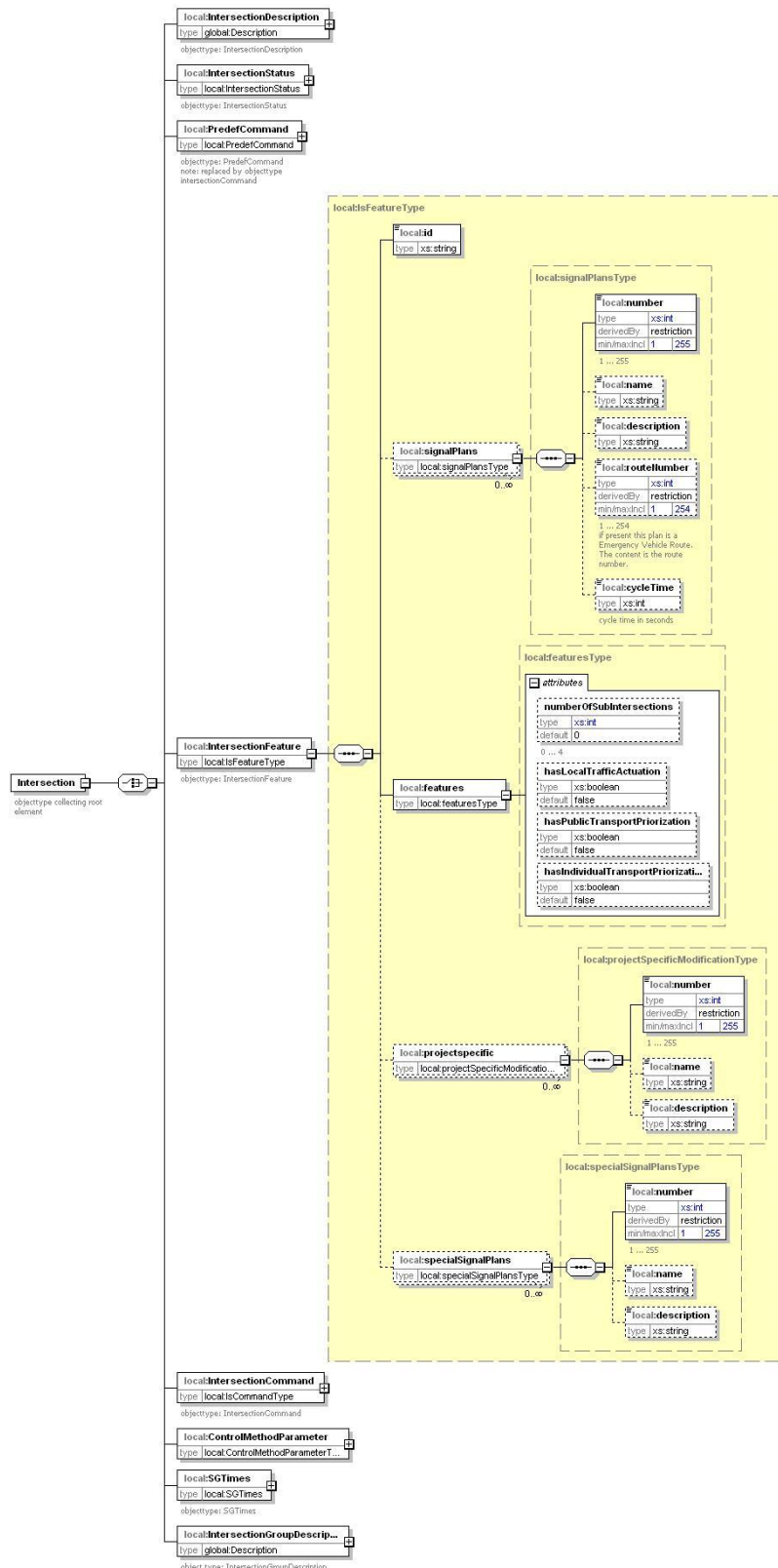
Projektbezogen können weitere Kommandos im Rahmen des Projektes hinzugefügt werden.

Sollten zu einem Zeitpunkt gleichzeitig mehrere vordefinierten Kommandos ausgeführt werden, sind diese innerhalb eines puts (innerhalb der put-Liste unter gleicher Objekt-Id) aufzuführen. Der Server interpretiert diese so, dass diese nur zu einer Umschaltung im Steuergerät führen.

3.10.2 Schaltbarkeiten (IntersectionFeatures)

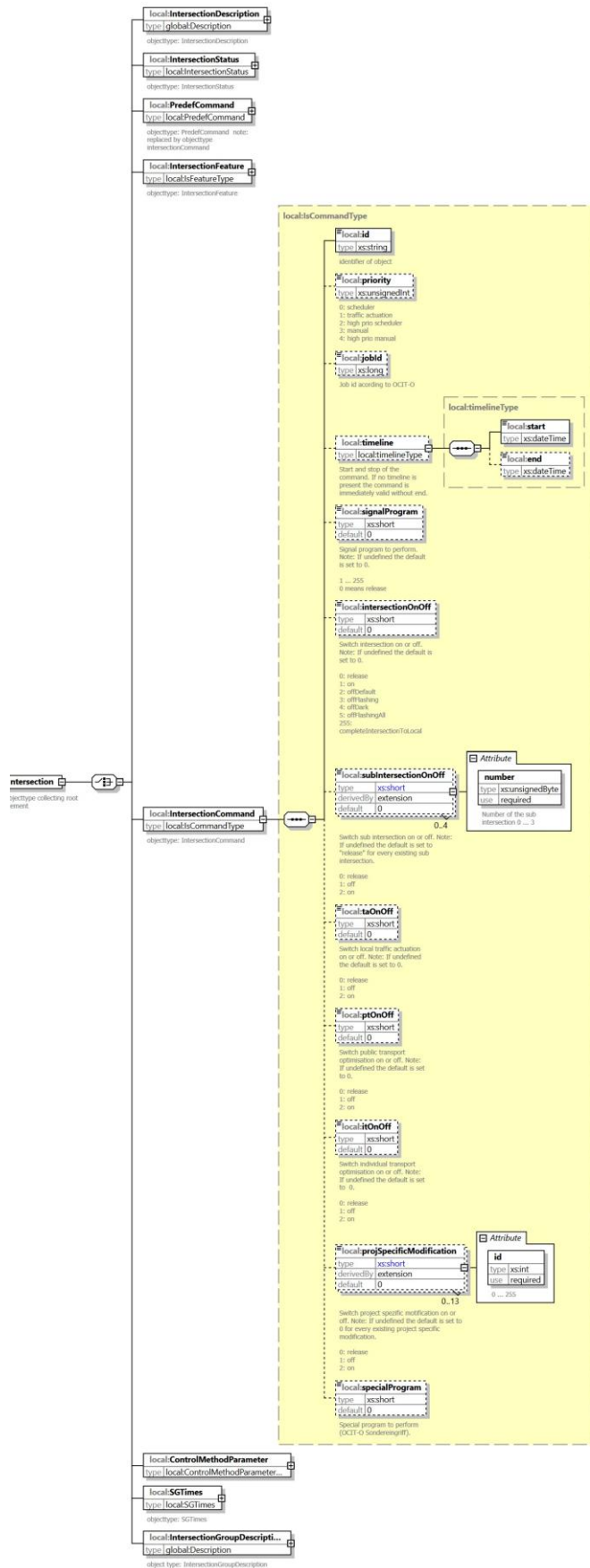
Mittels dem Objekttype IntersectionFeatures können Schaltmöglichkeiten einer LSA weitergegeben werden (z.B. welche Pläne sind schaltbar).

Damit lässt sich die Auswahl von Schaltungen einschränken, bevor diese über INtersection-Command abgesetzt und in diesem Zuge abgelehnt werden.



3.10.3 Ausmodellierte Schaltungen (IntersectionCommand)

Mittels dem Objekttyp IntersectionCommand lassen sich Schaltungen auslösen.



Hinweis: Die Elemente signalProgram, intersectionOnOff, subIntersectionOnOff, taOnOff, ptOnOff, itOnOff, projSpecificModification und specialProgram werden mit dem Default-Wert (0 – Release) belegt, wenn sie nicht angegeben werden.

3.10.4 Steuerungs-Parameter (ControlMethodParameter)

Adressat der dynamischen Parameter ist ein verkehrstechnisches Verfahren innerhalb der Lichtsignalanlage.

Der Inhalt wird daher vollständig durch die Gerätefunktion, z.B. das VA-Verfahren, bestimmt und muss zwischen dem Geräte-Hersteller und dem sendenden OCIT-C-Client abgestimmt werden.

Identifizierung: J <SystemNr> <SubsystemNr> <UnitNr> <ObjektNr>

Beispiel: „J1_12_22555_1“ (identifiziert ein Gerät, Nicht einen ApWert)

Siehe: 2.3.1

Parameter	Verfahren
path	Name des Parametersatzes für die transparente Weitergabe an das Gerät, z.B. per OCIT-O, unter dem das Gerät den APWertBlock (BLOB) erwartet. Beispiele: „61.200“, „VDK“, „DPV1“, „„VS-PLUS.NETZSTEUERUNG“
data	Binärer Inhalt des ApWert Blocks, transparent weitergeleitet an das Gerät, z.B. per OCIT-O.
id	Optionale Bezeichnung des Gerätes
methodname	Optionaler Name des Steuerungsverfahrens, das die Daten erhalten soll, z.B. VS-PLUS
interfacename	Optionaler Name der Schnittstelle im Steuergerät, z.B. NETZSTEUERUNG

Eine Rückmeldung kann über geeignete Betriebsmeldungen erfolgen, die projekt- oder verfahrensspezifisch vereinbart werden.

3.10.5 LSA Zustand (IntersectionStatus)

Detailinformationen über den LSA Zustand finden sich im „faultStatus“, sofern vom Gerät unterstützt:

TextNr: Eine Nummer die den Inhalt kennzeichnet

Text: Optionaler Texteintrag

Params: Parameter zur weiteren Detaillierung

Default: Vom Client verwendeter Wert, wenn die TextNr nicht vom Gerät gesendet wurde.

LSA Zustand (Intersection Status mit RSU Device Status)			
TextNr	Daten		Beschreibung
1001	Zeitquelle		Gibt die aktuelle Zeitquelle der Gerätezeit an. Default: 5
	Params		
	0	Netzspannung	
	1	Quarzuhr	
	2	LAN	
	3	DCF	
	4	GPS	
	5	Unbekannt	
1002	NotAus		Der NOTAUS-Schalter des Gerätes wurde betätigt.
1003	TuerAuf		Die Tür ist geöffnet.
1004	NetzOK		Die Netzspannung im Gerät ist OK.
1005	PersistenzspeicherOK		Darf ab v2.1 nicht mehr verwendet werden. Wird ersetzt durch TextNr 1017.
1006	Lampenstoerung		Die angegebenen Lampen sind gestört.
	Params	Ein Parameter pro Lampenstörung <Relknoten><Signalgruppe><Signalgeber><Kammer> z.B „0-8-2-3“	
1007	Detektorstoerung		Die angegebenen Detektoren sind gestört.
	Params	Kanalnummer Ein Parameter pro gestörtem Detektor z.B. "5"	
1008	Sammelstoerung		Sammelstörung aus OCIT-O wird unter faultState aufgeführt. Zur Abwärtskompatibilität muss ein Client sowohl Param (bevorzugt) als auch den Störungstext unterstützen. Default: 0
	Params	Störungstext	
	0	„KeineStoerung“	
	1	„Stoerung“	
	2	„StoerungMitAbschaltung“	
	3	„StoerungMitTeilabschaltung“	
	4	„InterneStoerung“	
1009	Verbindungsverlust		Information, dass die Verbindung zwischen Zentrale und LSA unterbrochen ist

1010	OevEmpfang		Gibt an, ob der ÖV-Empfänger funktioniert. Default: 255
	Params		
	0	OevEmpfang konfiguriert und OK	
	1	Zu viele CRC-Fehler bei den empfangenen Telegrammen	
	2	Empfangseinheit gestört bzw. keine Kommunikation	
	3	Funkempfangseinheit gestört bzw. kein Rauschen	
	4	Seit längerer Zeit keine gültigen ÖV-Telegramme mehr empfangen (Timeoutzeit gerätespezifisch parametrierbar)	
	254	Kein OevEmpfang konfiguriert	
	255	Zustand unbekannt oder nicht lieferbar	
1011	Referenzzeit		Gibt an, ob das Gerät synchron zur Referenzzeit ist. Default: 255
	Params		
	0	Geräte-TX läuft synchron zur Referenzzeit	
	1	Gerät läuft nicht synchron zur Referenzzeit	
	2	Gerät läuft nicht in synchronisierter Betriebsart	
	254	Kein Rückrechnungsverfahren konfiguriert, d.h. Gerät muss nicht zeitsynchronisiert laufen	
	255	Zustand unbekannt oder nicht lieferbar	
1012	Zentralenbefehle		Gibt an, ob das Gerät aktuell auf Zentralen-Schaltbefehle reagieren würde. Default: 255
	Params		
	0	Gerät akzeptiert Schaltwünsche von der Zentrale	
	1	Gerät akzeptiert keine Schaltwünsche von der Zentrale (z.B. Betriebsart lokal fix)	
	255	Zustand unbekannt oder nicht lieferbar	
1013	USV		Gibt an, ob die USV im Gerät installiert ist und funktioniert. Default: 255
	Params		
	0	USV ist OK, Batterie hat ausreichend Spannung	

	1	USV ist defekt bzw. Batterie leer oder defekt	
	254	Keine USV im Gerät konfiguriert	
	255	Zustand unbekannt oder nicht lieferbar	
1014	Zeitquelle / Zeitsynchronisation		Gibt an, ob mindestens eine der konfigurierten Zeitquellen zur Zeitsynchronisation verfügbar ist. Default: 255
	Params		
	0	Mindestens eine der definierten Zeitquellen zur Zeitsynchronisation ist erreichbar und funktioniert	
	1	Keine der konfigurierten Zeitquellen ist erreichbar bzw. funktioniert	
	254	Keine externen Zeitquellen im Gerät konfiguriert	
	255	Zustand unbekannt oder nicht lieferbar	
1015	RSUZustand		Gibt den Zustand der RSU an. Default: 255
	Params		
	0	RSU konfiguriert und OK	
	1	Kommunikationseinheit gestört bzw. keine Kommunikation	
	2	Zertifizierungssystem gestört bzw. keine gültige Zertifikatskette	
	3	Interne Störung, z.B. Versorgungsproblem	
	4	Interne Kommunikationsstörung zwischen RSU und Steuergerät/OCIT-Vorsatz	
	254	Keine RSU konfiguriert	
	255	Zustand unbekannt oder nicht lieferbar	
1016	Wartungszustand		Gibt an, ob sich das Gerät im Wartungszustand befindet Default: 255
	Params		
	0	Gerät befindet sich nicht im Wartungszustand	
	1	Gerät befindet sich im Wartungszustand	
	254	Wartungszustand wird vom Gerät nicht unterstützt	
	255	Wartungszustand unbekannt oder nicht lieferbar	
1017	Persistenzspeicher		Gibt den Zustand des persistenten

	Params		Gerätespeichers an.
	0	Persistenzspeicher ist ok	Default: 0
	1	Persistenzspeicher ist gestört	

3.10.6 SGTimes

Signalgruppenbezogene Rot- und Grünzeiten werden im Objekttyp SGTimes umlaufbezogen zusammengefasst. Die Adressierung erfolgt auf Ebene der LSA, d.h. innerhalb dieses Objekttyps ist es nicht möglich Signalgruppen einzeln anzusprechen.

3.10.7 IntersectionGroupDescription

Die LSA-Gruppenzuordnung wird im Objekttyp IntersectionGroupDescription definiert.

Empfohlene Identifizierungen:

JG<SystemNr>_<SubsystemNr>_<GroupNr>

JG<GroupNr>

Verwendung des Element **type**

- Beschreibung LSA-Gruppentyp: z.B. Gruppe, Bezirk

Verwendung Element **Rel_Ids**

- Verweis auf LSA
- Verweis auf LSA-Gruppe

3.11 LSA Rohdaten

Dieser Kommunikationsbaustein dient zur effizienten Übertragung großer Mengen diskreter Werte von Lichtsignalanlagen.

Die zu übertragenden Daten werden nach Zuständen gruppiert. Jeder angenommene Zustand wird einmal aufgeführt und ihm die Zeitpunkte seiner Annahme zugeordnet. Alle Zeitstempel aus einem bestimmten Zeitintervall beinhalten eine Startzeit (xsd:element name="timeline") für das Intervall und den Versatz zur Startzeit. Der Versatz lässt sich als Produkt einer Anzahl und einer Zeiteinheit(xsd:element name="intervallLength") darstellen. Diese Anzahl wird als vorzeichenloser 16-Bit-Wert abgespeichert. Daraus ergibt sich einerseits eine maximale Intervalllänge, andererseits ist es möglich, alle 16-Bit-Werte nacheinander ohne eigene XML-Elemente aufzuführen. Die XML-Repräsentation für das Feld von 16-Bit-Werten ist dann dessen Base64-Codierung (xsd:element name="Events").

Empfohlene Identifizierung:

- Objektidentifikation gemäß Abschnitt 2.3.1

Verfügbare Objekttypen:

Objekttyp	Schemadateien	Kurzbeschreibung
RawTrafficDataBlock_Detectoredge	intersection_raw_data.xsd	Detektorflanken
DigOut_Raw_Values		Digitale Ausgänge
RawTrafficDataBlock_Signalgroupvalue		Signalgruppenwerte
NamedValue_Raw_Values		AP – Werte
NamedValuesOrderList	intersection_raw_data.xsd	AP-Werte Bestellung
PublicTransportTelegramExtended	Public_Transport_ExtendedTelegram.xsd (importiert Public_Transport_Telegram.xsd)	Erweitertes ÖV- Telegramm
DetectorExtValueType	detector_ext.xsd	Einzeldetektordaten

3.11.1 Detektorflanken

Das Objekt detectorEdges (Objekttyp RawTrafficDataBlock_Detectoredge) wurde für die Übertragung der Detektorflanken definiert.

Der Value (xsd:string) wird mit ‚0‘ für fallende Flanke und ‚1‘ für steigende Flanke gesetzt.

Beispiel:

Detektor hat den Zustand „belegt“ zu folgenden Zeiten erreicht (steigend):

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.

Detektor hat den Zustand „nicht belegt“ zu folgenden Zeiten erreicht:

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.

Die Daten werden wie folgt eingetragen:

Id des Detektors: id: Det_1

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

Zeiteinheit in Millisekunden: intervalLength: 100

Steigende Flanken: data/Value: 1

Events:

Berechnungsvorschrift: (Eventszeit – Startzeit) / Zeiteinheit == Anzahl

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

Anzahl wird als vorzeichenloser 16-Bit-Wert abgespeichert.

1: $100 / 100 == 0x0001$

2: $1200 / 100 == 0x000c$

3: $2000 / 100 == 0x0014$

Alle 16-Bit-Werte sollen nacheinander eingetragen werden.

Byte1 == 0x00

Byte2 == 0x01

Byte3 == 0x00

Byte4 == 0x0c

Byte5 == 0x00

Byte6 == 0x14

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

Fallende Flanken: data/Value: 0

Events:

4: $300 / 100 == 0x0003$

5: $1800 / 100 == 0x0012$

6: $2300 / 100 == 0x0017$

data/Events: AAMAEgAX (000300120017 in Base64 kodiert)

3.11.2 Signalgruppenzustände

Das Objekt sgValues (Objekttyp RawTrafficDataBlock_Signalgroupvalue) wurde für die Übertragung der Signalgruppenzustände definiert.

Der Value(xsd:string) wird als integer belegt. Der Wert soll identisch zu OCIT-O Objekt Signalbilder 1:611 verwendet werden. (z.B. 3 für rot, 48 für grün.)

Falls Value auf „3“ gesetzt wird, werden die Zeitpunkte eingetragen, zu denen die Signalgruppe in den Zustand „rot“ wechselt.

Falls Value auf „48“ gesetzt wird, werden die Zeitpunkte eingetragen, zu denen die Signalgruppe in den Zustand „grün“ wechselt.

Beispiel:

Signalgruppe hat den Zustand „rot“ um:

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.

Die Daten werden wie folgt eingetragen:

```
id: Sg_1

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

intervalLength: 1000

data/Value: 3

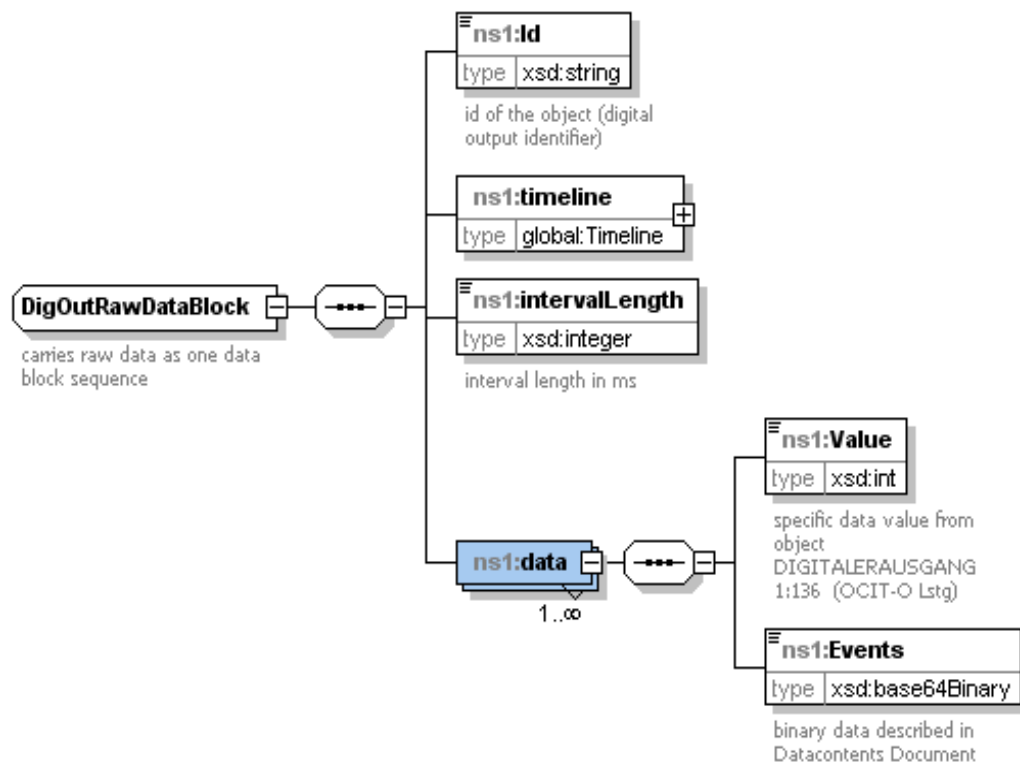
Events:

1: 10000 / 1000 == 0x000a
2: 70000 / 1000 == 0x0046
3: 130000 / 1000 == 0x0082

data/Events: AAoARgCC (000a00460082 in Base64 kodiert)
```

3.11.3 Digitale Ausgng im Lichtsignalsteuergerät

Das Objekt DigOutRawType (Objektyp DigOut_Raw_Values) wurde für die Übertragung der Zustände des digitalen Ausgangs definiert.



Der Zustand wurde als Typ integer definiert. Der Wert wird aus OCIT-O Objekt DIGITALERAUSGANG 1:136 übernommen, z.B. 3 für Ein, 1 für Blinken Anfang Aus.

Falls Value auf „3“ gesetzt wird, werden die Zeitpunkte eingetragen, zu denen der digitale Ausgang in den Zustand „Ein“ wechselt.

Falls Value auf „1“ gesetzt wird, werden die Events eingetragen, zu denen der digitale Ausgang in den Zustand „Blinken Anfang Aus“ wechselt.

Beispiel:

Digitaler Ausgang hat den Zustand „Ein“ um:

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

Die Daten werden wie folgt eingetragen:

```

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

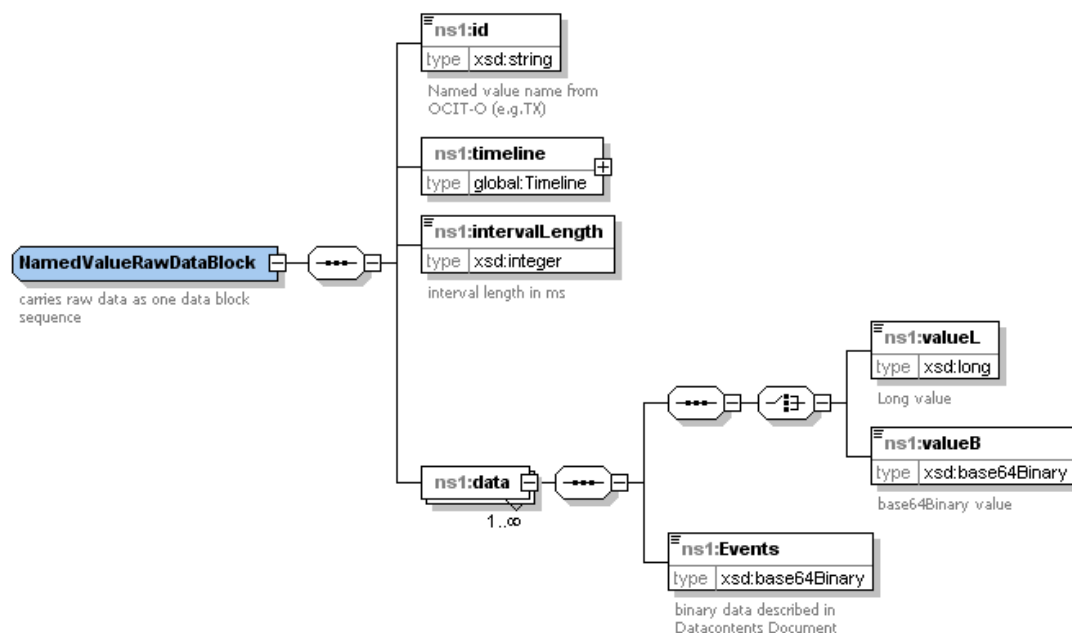
3: 2000 / 100 == 0x0014

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

3.11.4 Anwenderprogrammwerte (AP-Werte)

Das Objekt NamedValueRawType wurde für die Übertragung der Zustände des AP-Wertes definiert.

AP-Werte werden über einen Identifikationsstring identifiziert. Der String ist im XML-Schema intersection_config_data_ap_values.xsd (siehe OCIT-C_LSA_Versorgungsdaten) unter OITDdef/OITDdef/Verfahren/OITD/Bezeichner/KurzBez definiert.



Der Zustand wurde als Auswahl zwischen den Typen long und base64Binary definiert. Falls AP-Werte der Typen:

long, integer, short, byte übertragen werden sollen, wird valueL gesetzt.

Falls ein AP-Wert des Typs BLOB (s. OCIT-O Objekt APWertBlock 1:508) übertragen werden soll, wird valueB gesetzt. Es werden die binären Daten aus dem Typ BLOB übertragen (BYTE data[]).

OCIT-C definiert folgende standardisierte AP-Werte:

OITD IP- Schreibweise	OITD4- Nr.	OITD2- Nr.	Name	Beschreibung
41.94	2687070	42078	TX	Die Umlaufsekunde des laufenden Signalprogramms Beginnend mit Sekunde 0 in 100ms-Schritten gezählt. 65535: Wert undefiniert
41.96	2687072	42080	PH	Aktuelle Phase im Signalprogramm 0: Phase nicht aktiv oder im Verfahren nicht definiert 1...65534: Aktuelle Phasennummer
41.97	2687073	42081	UE	Gewünschte Phase 0: Kein Phasenübergang aktiv 1...65534: Übergang von Phase PH nach Phase UE aktiv
41.98	2687074	42082	DPV1	Dynamische Parameter für ein verkehrstechnisches Verfahren

Andere Namen von AP-Werten sind projektspezifisch zu vereinbaren.

Beispiele:

- Umlaufsekunde (TX) hat den Zustand „10“ um:

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.

Die Daten werden wie folgt eingetragen:

id: TX

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

intervalLength: 1000

data/Value: 1

Events:

1: 10000 / 1000 == 0x000a

2: 70000 / 1000 == 0x0046

3: 130000 / 1000 == 0x0082

data/Events: AAOARgCC (000a00460082 in Base64 kodiert)

- AP-Wert Block (APWertB) hat den Zustand „BYTE data[01, 05, 0c, a2]“ um:
 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.

Die Daten werden wie folgt eingetragen:

```
id: APWertB_1

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

intervalLength: 1000

data/Value: AQUmog == (01050ca2 in Base64 kodiert)

Events:

1: 10000 / 1000 == 0x000a
2: 70000 / 1000 == 0x0046
3: 130000 / 1000 == 0x0082

data/Events: AAOARgCC (000a00460082 in Base64 kodiert)
```

3.11.4.1 Abfrage beauftragter AP-Werte

Die zuvor bei einem Knoten beauftragten AP-Werte werden zurückgeliefert:

Objekttyp: „NamedValue_Raw_Values“

Falls nur bestimmte AP-Werte zurückgeliefert werden sollen (z. B. nur TX), muss in der Methode „get“ des Protokolls das Element „data“ mit dem Datentyp „NamedValueFilterType“ gesetzt werden. Anderenfalls werden alle AP-Werte zurückgeliefert, die der Server für diese Anlage gesammelt hat.

Als Filtermöglichkeiten stehen für eine Abfrage drei Varianten zur Verfügung:

Variante 1 (knotenbezogene Filterung):

- „filterList“ (ohne NamedValueFilterType)
Schränkt die gelieferten AP-Werte auf die angegebenen AP-Werte ein (Standard-Verhalten der filterList). Knoten- und (optional) OITD-Nummer dürfen nicht abgekürzt werden.
 Beispiel 1: J1_12_4711_41.94_1 nur dieser AP-Wert dieses Knotens
 Beispiel 2: J1_12_4711_61.111 nur dieses AP-Wert-Array dieses Knotens
 Beispiel 3: J1_12_4711 alle AP-Werte dieses Knotens

Variante 2 (AP-Wert-bezogene Filterung):

- „data“ (Datentyp NamedValueFilterType)
Einschränkung auf eine Liste von AP-Werten mit Angabe der OITD-Nummern in IP-Schreibweise (wie bei Beauftragung) ohne Knotenbezug.

Beispiel 1:	41.94_1	definiert einen einzelnen AP-Wert (nur eine Instanz)
Beispiel 2:	61.111	definiert alle Instanzen einer AP-Wertliste
Beispiel 3:	61.111_213	definiert eine AP-Wert-Instanz einer Liste

Variante 3 (kombinierte Knoten- und AP-Wert-Filterung):

Ermöglicht eine orthogonale Filterung durch unabhängige Kombination von Knoten- und AP-Wert-Filterung.

- „filterList“
Schränkt die gelieferten AP-Werte auf die angegebenen Knoten ein. Hier ist nur die Angabe von Knoten erlaubt.
Beispiel: J1_12_4711 alle AP-Werte dieses Knotens
- „data“ (Datentyp NamedValueFilterType)
Schränkt die per „filterlist“ auf Knoten vorgefilterten AP-Werte auf die angegebenen AP-Werte ein.
Siehe Variante 2.

Beispiel für einen kombinierten XML-Request:

```
<ocitc:data xmlns:ns20="http://odg_and_partner/external/intersection_rawData"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ns20:NamedValueFilterType">
  <ns20:NamedValueId>41.94</ns20:NamedValueId>
  <ns20:NamedValueId>41.96</ns20:NamedValueId>
  <ns20:NamedValueId>41.97</ns20:NamedValueId>
</ocitc:data>
<ocitc:filterList>
  <ocitc:filter>
    <ocitc:identifier>
      <ocitc:ident>J1_13_466</ocitc:ident>
    </ocitc:identifier>
  </ocitc:filter>
  <ocitc:filter>
    <ocitc:identifier>
      <ocitc:ident>J1_13_8161</ocitc:ident>
    </ocitc:identifier>
  </ocitc:filter>
</ocitc:filterList>
```

3.11.4.2 Beauftragung abfragbarer AP-Werte

Zur Initialisierung extern abfragbarer AP-Werte wird ein Objekttyp eingeführt, der diese Liste „abfragbarer AP-Werte“ durch genau einen Schnittstellenpartner ermöglicht. D.h. im Kundenprojekt wird festgelegt, ob die Konfiguration dieser Liste überhaupt, und wenn ja, dann durch welchen Kommunikationspartner die Liste gesetzt werden kann.

Die externe Beauftragung ist optional, eine projektspezifische Abstimmung ist in jedem Fall erforderlich.

Eine externe Beauftragung ist nur in geringem Umfang möglich, da die maximale Anzahl an insgesamt beauftragbaren AP_Werten durch das Gerät begrenzt ist (Geräte-Typ, VA, Detektoren, ...).

Die Abfrage der AP-Werte durch mehrere Kommunikationspartner ist jedoch möglich, auch wenn die Konfiguration der Liste nur von einem Partner gesetzt werden kann.

Objekttyp: „NamedValuesOrderList“

Die Liste besteht aus den vollständigen Bezeichnern inkl. ObjektNr eines oder mehrerer AP-Werte für jeden Knoten. AP-Werte, auch Listenelemente, müssen einzeln aufgeführt werden.

Folgende Schreibweisen sind möglich:

- J1_12_4711_41.49_1 definiert einen einzelnen AP-Wert (nur eine Instanz)
- J1_12_4711_61.111_213 definiert eine einzelne AP-Wert-Instanz einer Liste

Die Funktion ist abhängig von der verwendeten Methode:

- put, delete:
Alle noch nicht beauftragten AP-Werte werden beauftragt bzw. entfernt. Durch Verwendung der putList können AP-Werte in einem Schritt bei mehreren Knoten beauftragt bzw. entfernt werden.
Wegen direkter Beeinflussung der Steuergeräte muss diese evtl. kritische Funktion projektspezifisch vereinbart werden.
- get:
Abfrage der aktuell beauftragten und lieferbaren AP-Werte von allen Knoten (ohne filterList) oder nur der in filterList angegebenen Knoten.

3.11.5 ÖV-Telegramme

3.11.5.1 R09-Telegramm

Das Objekt PTTelegramType (Objekttyp PublicTransportTelegramm) wurde für die Übertragung des R09-Telegramms für ÖV-Daten definiert. Für die Übertragung im Protokoll wird als Container das Objekt rawDC unter Ergänzung des Zeitstempels verwendet.

3.11.5.2 Erweitertes ÖV-Telegramm

Das Objekt PTExtendedTelegramType (Objekttyp PublicTransportTelegrammExtended) wurde für die Übertragung von erweiterten ÖV Telegrammen definiert. Für die Übertragung im Protokoll wird als Container das Objekt rawDC unter Ergänzung des Zeitstempels verwendet.

3.11.6 Einzeldetektordaten

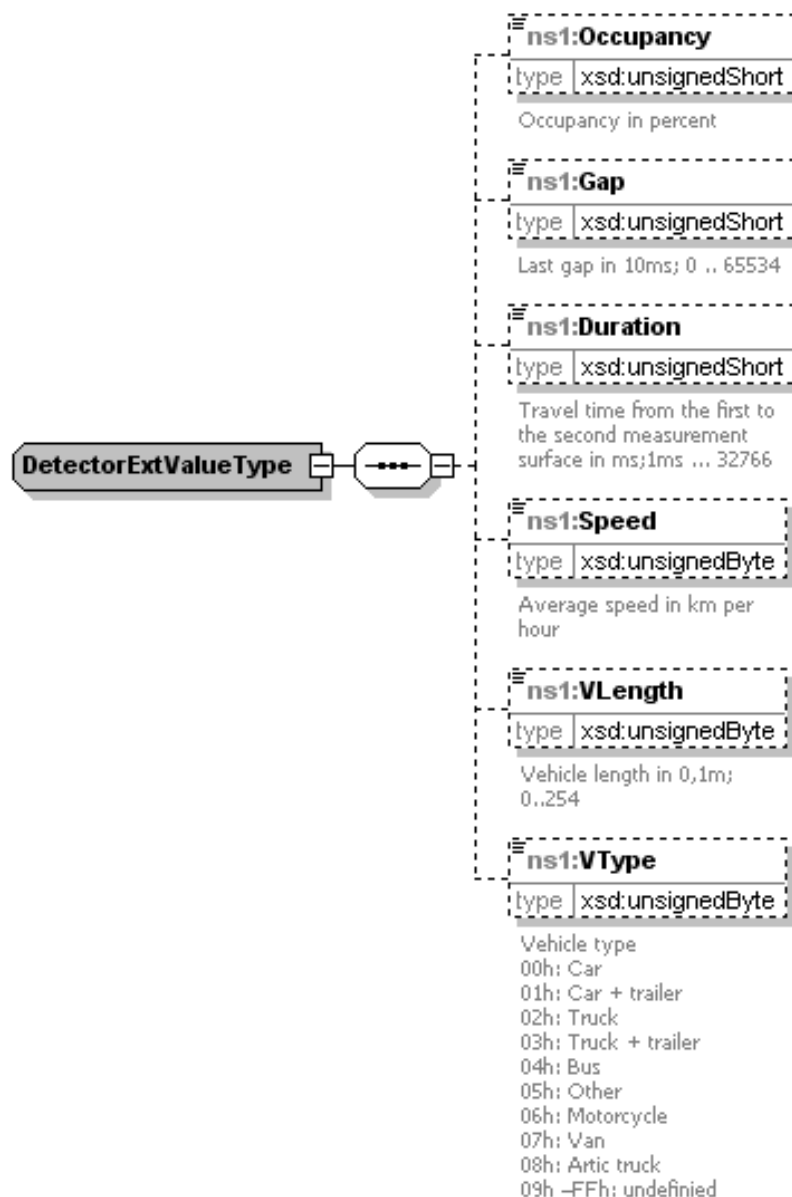
Das Objekt DetectorExtValueType (detector_ext.xsd) wurde für die Übertragung der Zusatzinformationen des Detektors definiert. Der Messwert bezieht sich auf das Detektorereignis eines einzelnen Fahrzeugs.

Hinweis: Da bei der Übertragung von Einzelwerten große Datenmengen entstehen, sollen die Daten nur für wenige Detektoren verfügbar werden.

Empfohlene Identifizierung:

DExt<primary key>_<secondary key>

<primary key> ist optional.



Der erweiterte Detektormesswert beinhaltet folgende Elemente:

Occupancy – Belegungsdauer der Messstelle in 10ms

Gap – Die letzte Lücke der Schleife in 10ms

Duration – Fahrdauer von der ersten bis zur zweiten Messstelle in ms

Speed – Gemessene Geschwindigkeit in km/h

VLength – Länge des Fahrzeugs

VType – Art des Fahrzeugs (s. OCIT-O Objekt FahrzeugArt 1:3128)

3.11.7 Objekttyp: Signalgroup_Description

Die Datenstruktur wird im Kapitel 2.4.1 erklärt.

Verwendung Element **Rel_Ids**

- Verweis auf LSA

3.12 Car-2-X Kommunikation

In diesem Abschnitt wird eine kurze Übersicht alle Objekte der Car-2-X Kommunikation für die OCIT-C V2.0 Schnittstelle zwischen Zentrale beschrieben.

3.12.1 Statusinformation

Für eine Qualitätsanalyse und Qualitätssicherung von Knotenpunkten sollen Daten aus der C2X – Kommunikation verwendet werden.

Die Fahrzeuge senden die Statusinformationen vieler Sensoren mittels der CAM Nachrichten an die RSU. Viele Daten, wie Außentemperatur, Status des Bremskraftverstärkers oder die Winkelstellung des Lenkrads, sind dabei uninteressant. Die für die Verkehrstechnik interessanten Daten sind folgende:

- Datum und Zeit
- Position
- Geschwindigkeit
- Richtung des Fahrzeugs
- Fahrzeugtyp

Mit Hilfe der Kreuzungstopologie aus der MAP Nachricht lassen sich die Positionen der Fahrzeuge einer Spur oder einer Signalgruppe zuordnen. Für eine zentrale Analyse der Fahrzeugdaten werden man folgende Werte bereitgestellt (jeweils im Bereich der RSU):

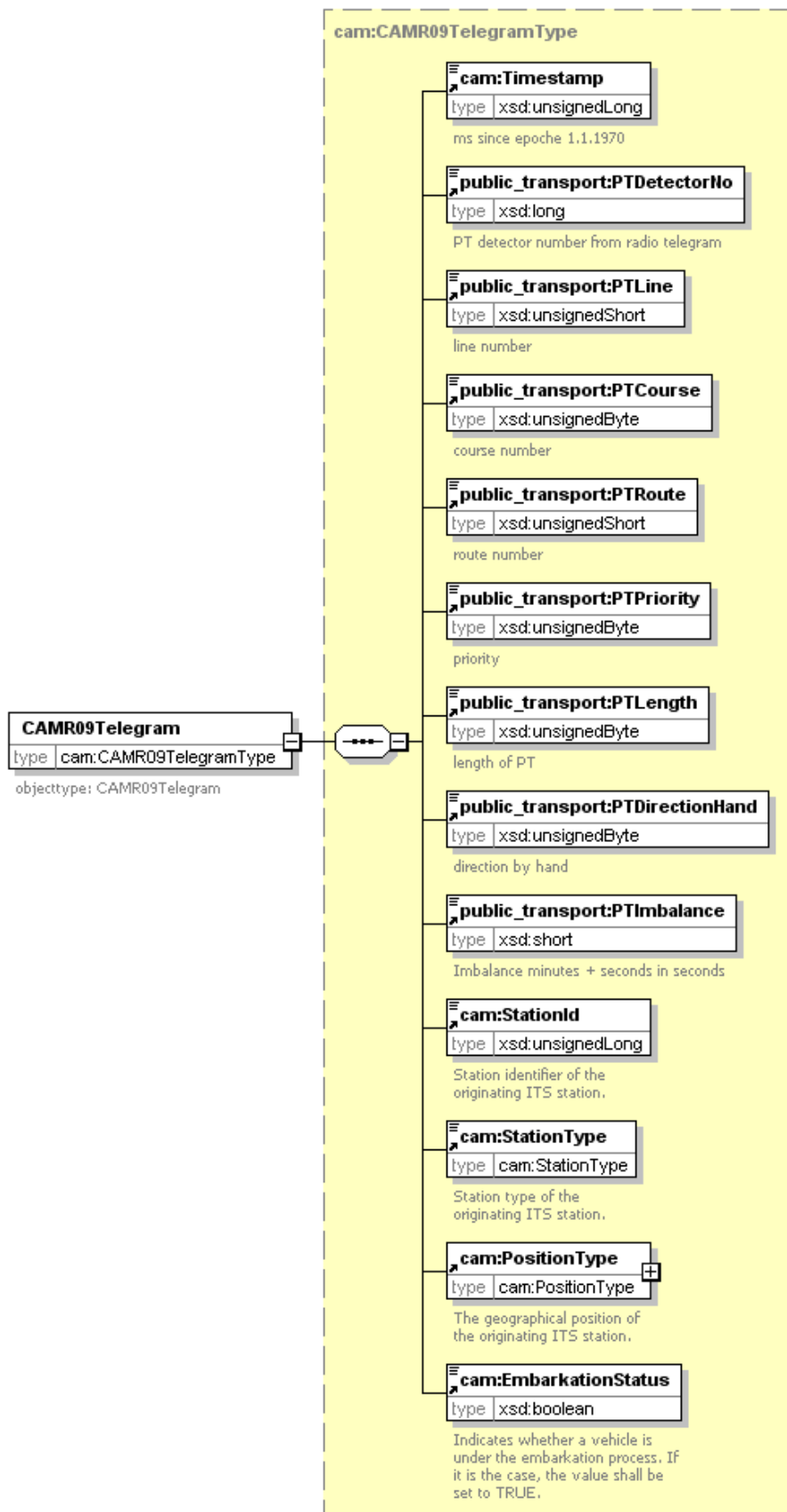
- Durchschnittsgeschwindigkeit
- Wartezeit
- Anzahl der Halte
- verbindungsbezogene Durchschnittswerte

3.12.2 Priorisierung

Über die CAM Telegramme der Car2X Kommunikation können öffentliche Verkehrsmittel und Sondereinsatzfahrzeuge eine Priorisierung anfordern. Mit OCIT-C Car können diese Anforderungstelegramme aufgezeichnet werden.

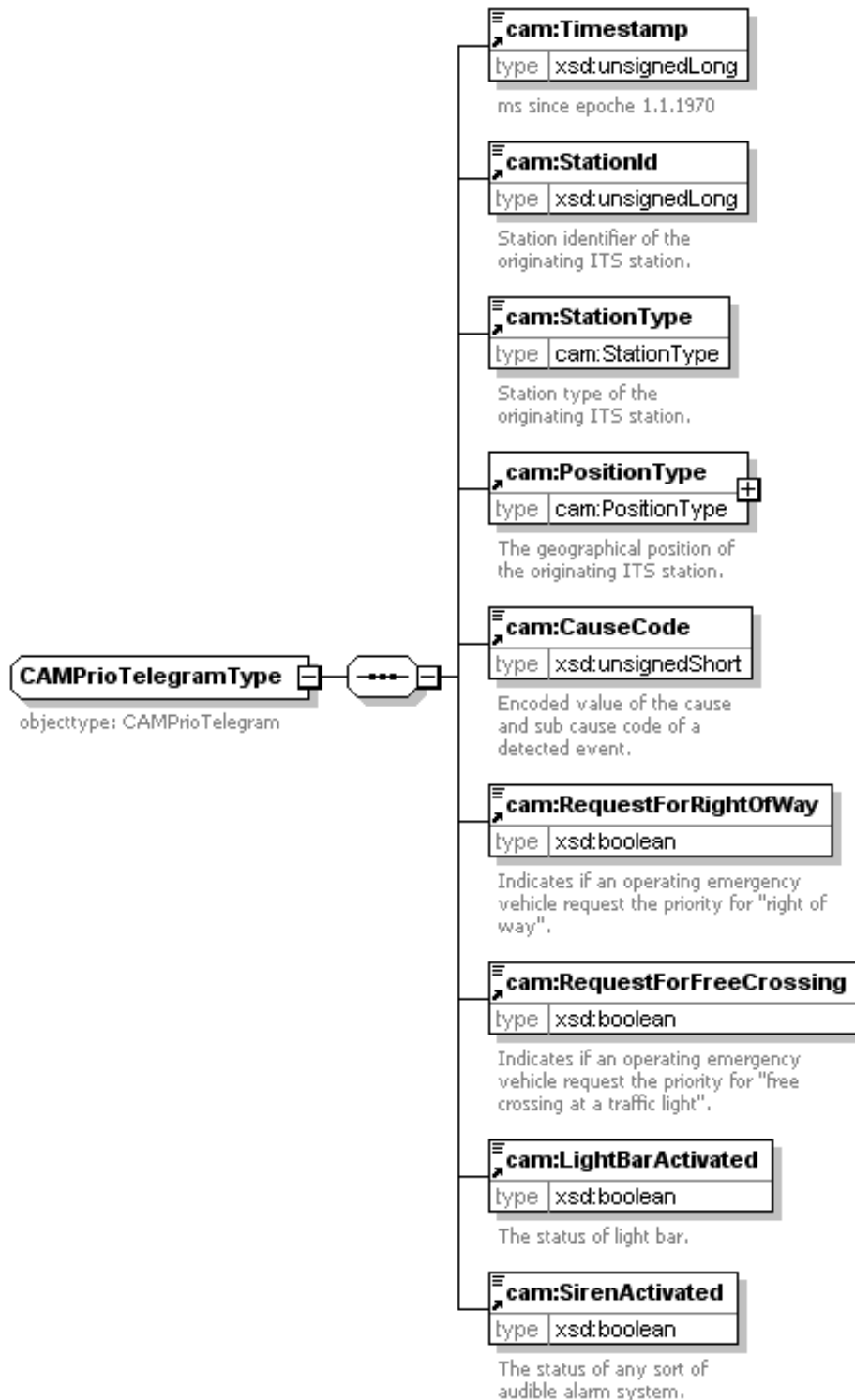
3.12.2.1 CAM-R09-Telegramm

Das Objekt CAMR09TelegramType (Objekttyp CAMR09Telegram) wurde für die Übertragung von CAM Meldungen mit R09-Container definiert.



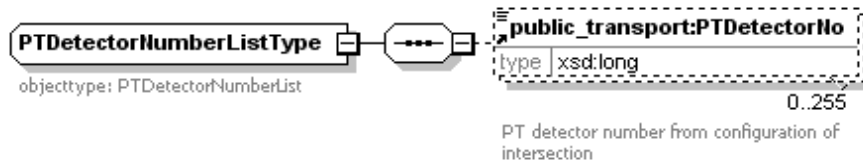
3.12.2.2 CAM-Prio-Telegramm

Das Objekt CAMPrioTelegramType (Objekttyp CAMPrioTelegram) wurde für die Übertragung von CAM Meldungen mit Rescue-, Emergency- oder SafetyCar-Container definiert.



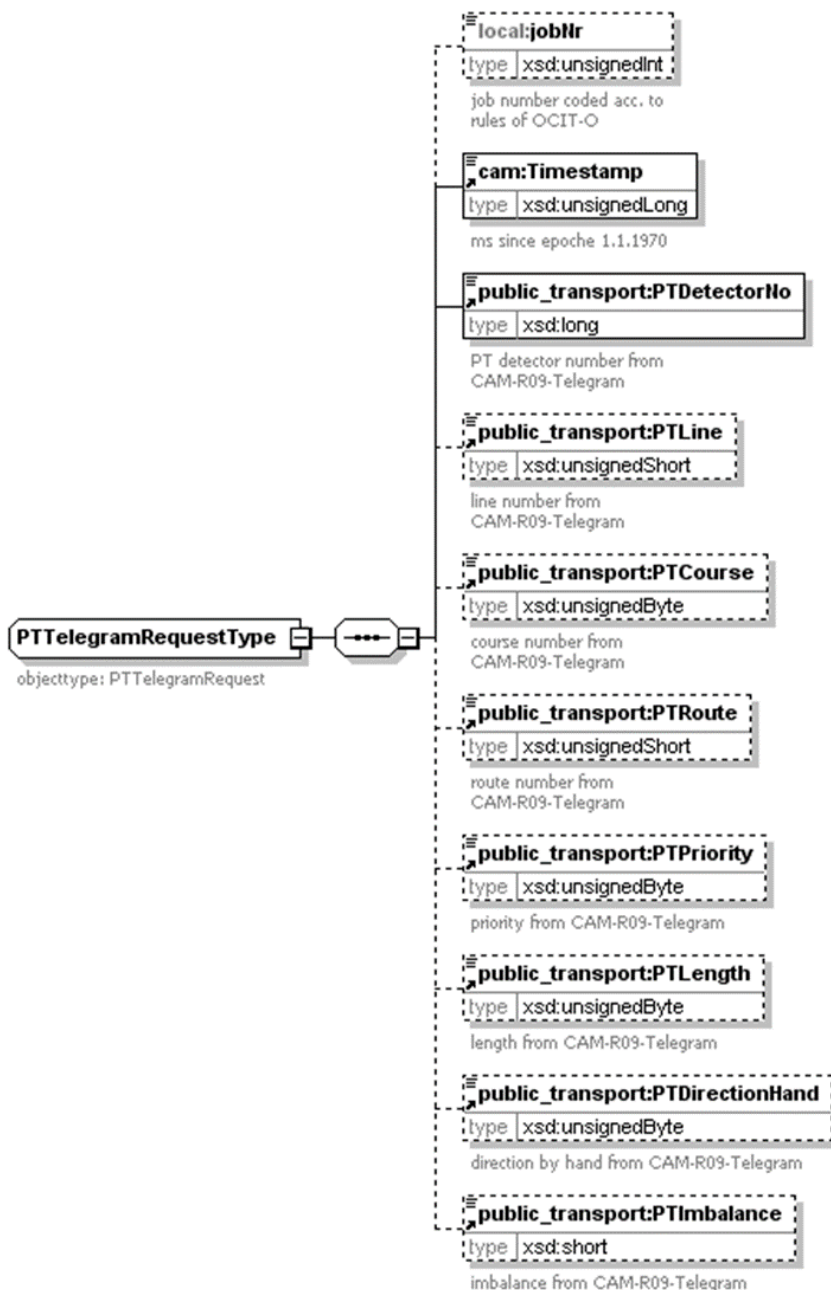
3.12.2.3 Oepnv-Meldepunktliste

Das Objekt PTDetectorNumberListType (Objekttyp PTDetectorNumberList) wurde für die Übertragung der in einer LSA konfigurieren OEPNV-Meldepunkte definiert. Damit kann man eine Abfrage der OEPNV Meldepunktliste der LSA durchführen.



3.12.2.4 Oepnv-Anforderung

Das Objekt PTTelegramRequestType (Objekttyp PTTelegramRequest) wurde für die Weiterleitung von OEPNV Anforderungstelegrammen (R09-Telegramme) definiert.



3.12.3 Gefahrenmeldungen

Verfügbare Objekttypen:

Objekttyp	Schemadatei	Kurzbeschreibung
DENMessage	Denm.xsd	Decentralized Environmental Notification Message

Gefahrenmeldungen können auf der Zentralen- sowie auf der Feldebene ausgelöst und im gesamten System verteilt werden.

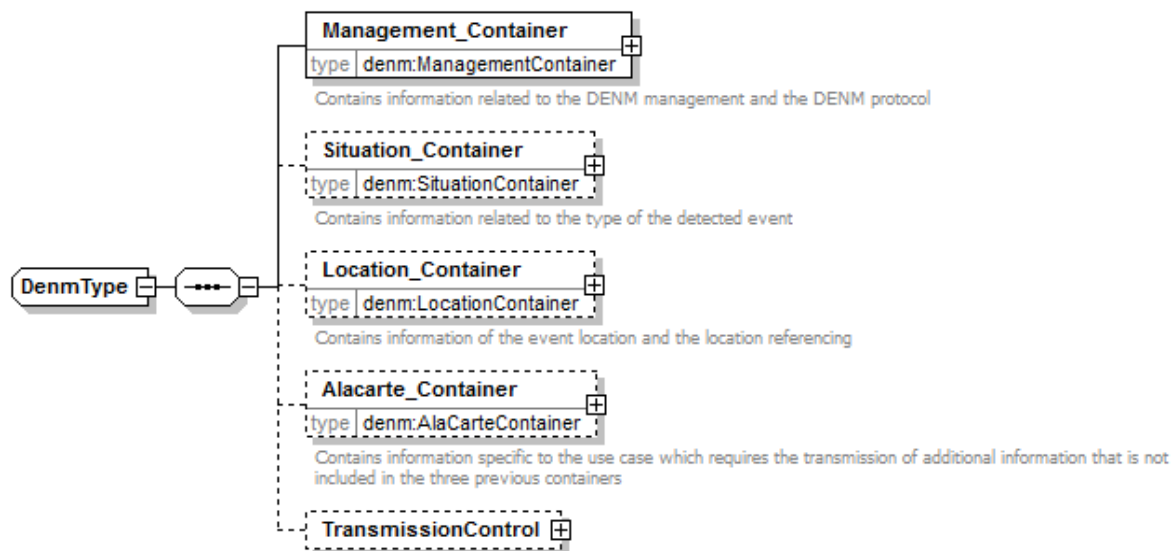
Gefahrenmeldungen umfassen folgende Bereiche:

- Straßenarbeiten
- Unfall und Unfallmeldung
- Verkehrslage
- Straßenzustand
- Metrologische Daten (Wind, Niederschlag, Sicht)

Gefahrenmeldungen werden mit dem DENMessage Objekt übertragen.

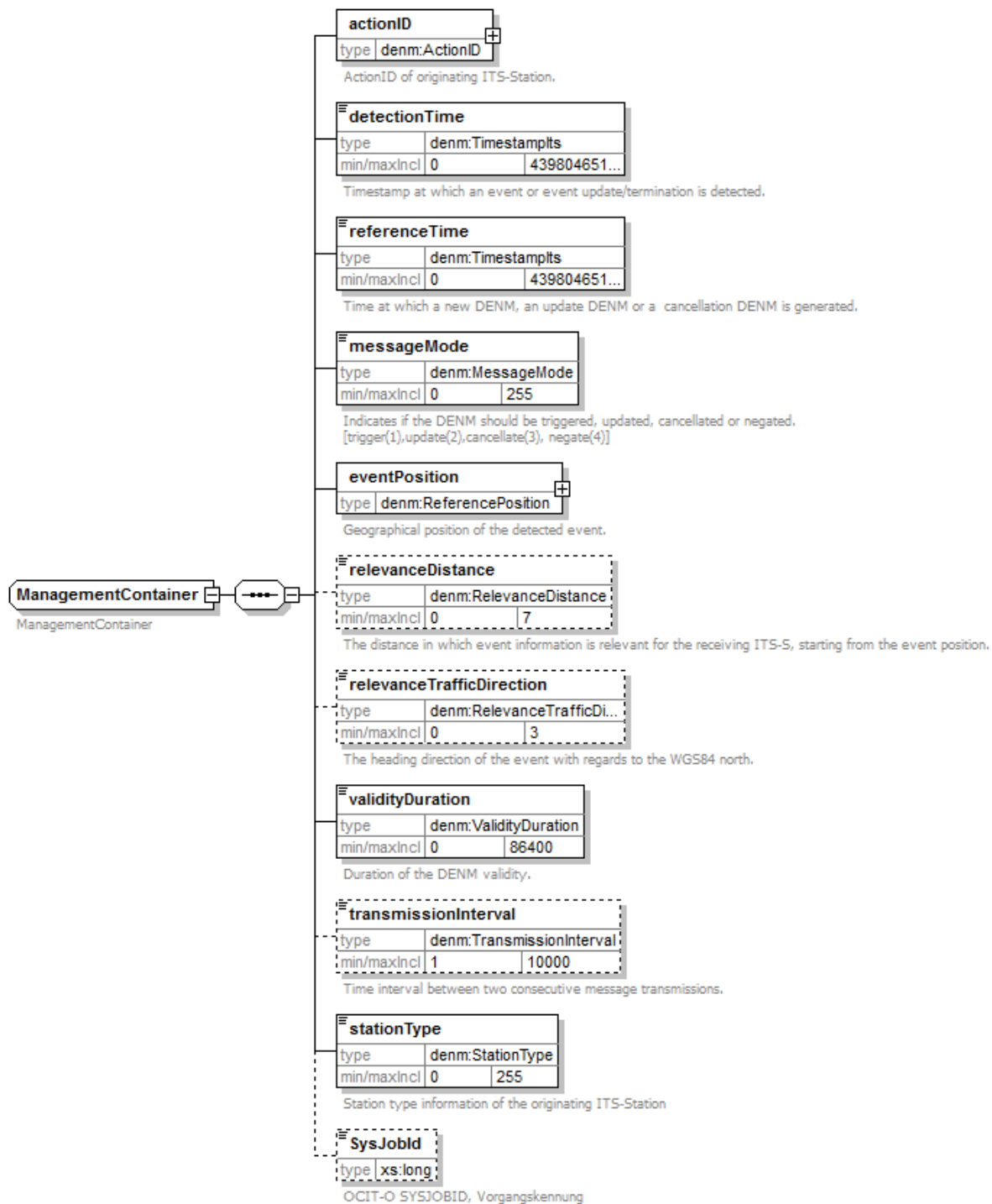
Ein DENMessage Objekt setzt sich zusammen aus den 5 Blöcken:

- Management_Container
- Situation_Container
- Location_Container
- Alacarte_Container
- TransmissionControl



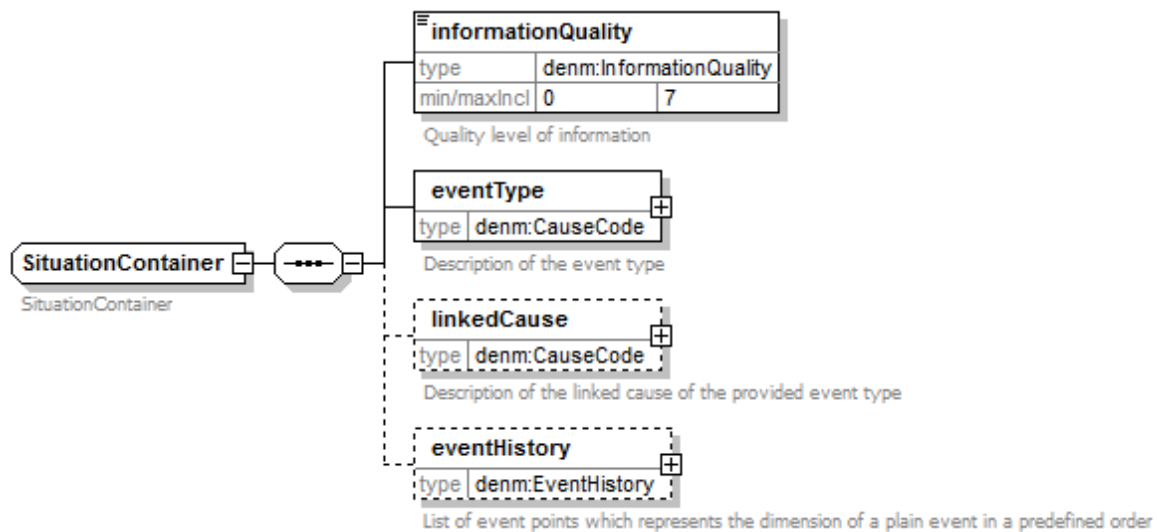
3.12.3.1 Management_Container

Contains information related to the DENM management and the DENM protocol.



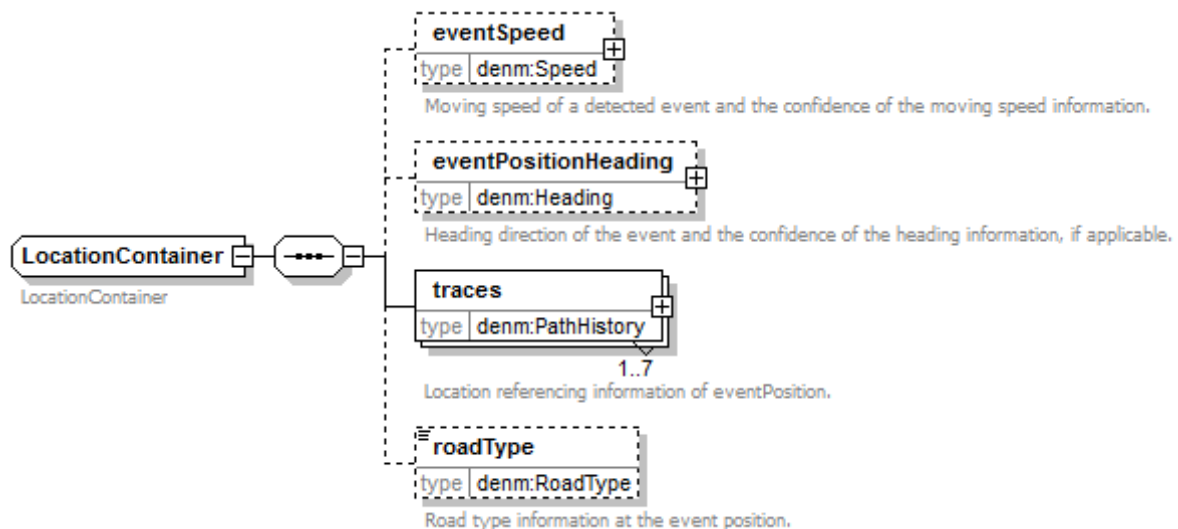
3.12.3.2 Situation_Container

Contains information related to the type of the detected event.



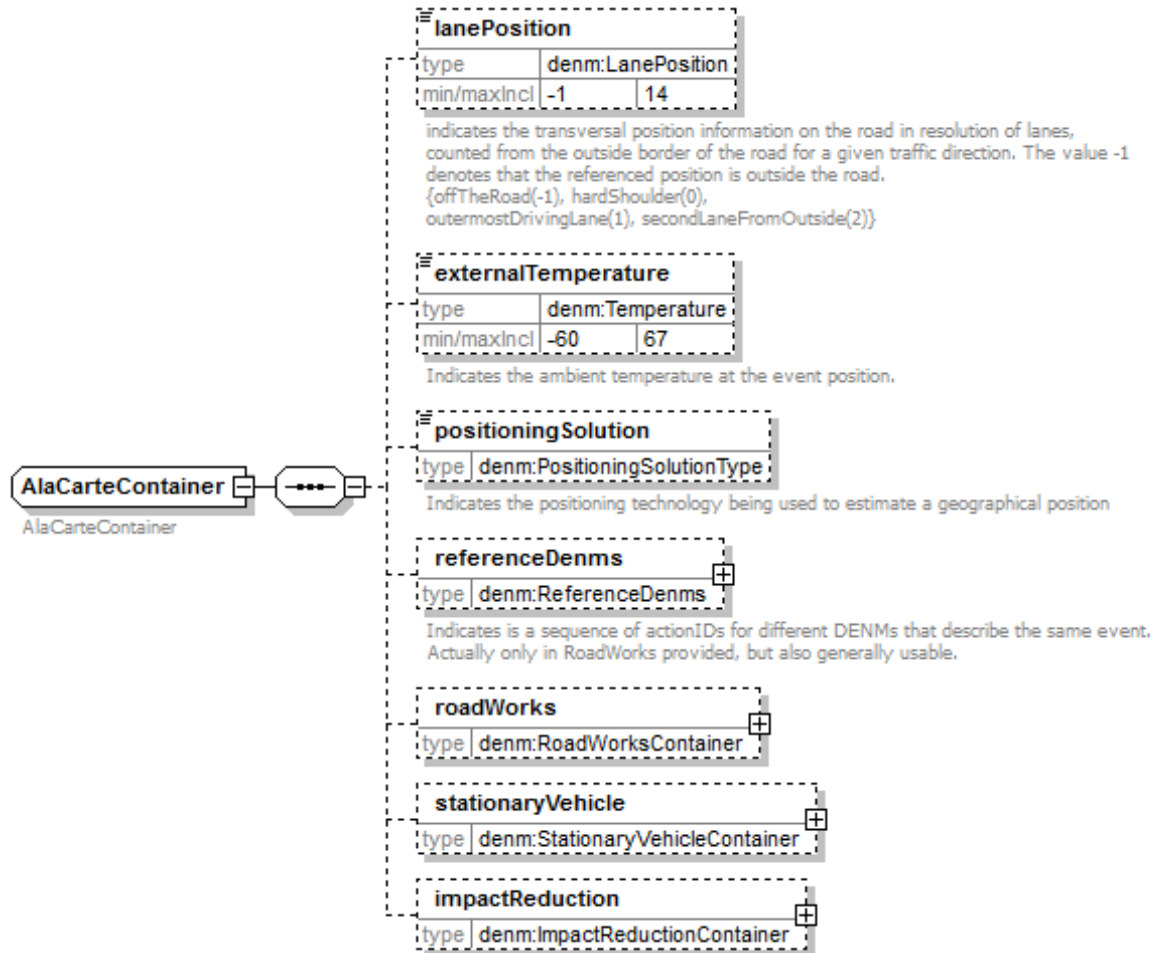
3.12.3.3 Location_Container

Contains information of the event location and the location referencing.



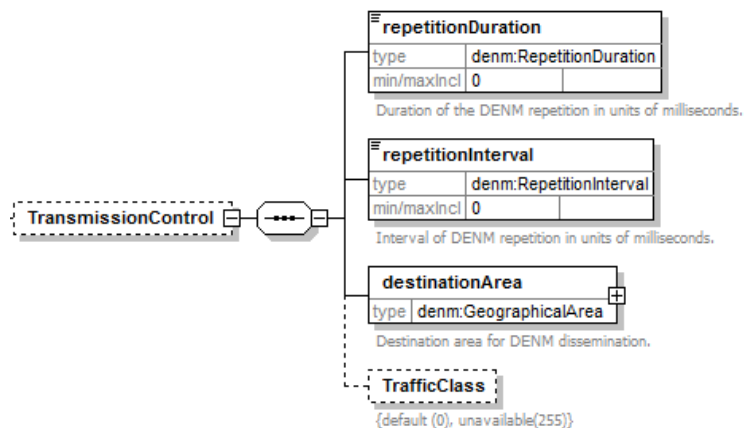
3.12.3.4 Alacarte_Container

Contains information specific to the use case which requires the transmission of additional information that is not included in the three previous containers.



3.12.3.5 TransmissionControl

Contains information about transmission features.



3.12.4 Spat

Die Spat bietet die Möglichkeit die prognostizierte Dauer für den aktuellen Zustand einer Signalgruppe abzufragen bzw. diese auch zwischen Zentralen zu auszutauschen. Die Übermittlung wird erforderlich, da es vorauszusehen ist, dass es zukünftig zentrale Prognoseverfahren geben wird, die Prognosen auf Basis von historischen Daten erstellen.

Der im Folgenden vorgestellte Objekttyp kann zur Einzelabfrage bzw. für das Übermitteln des prognostizierten Signalzustands von Signalgruppen benutzt werden.

3.12.4.1 Objekttyp PredictedSpat

Der Objekttyp PredictedSpat wird je Lichtsignalanlage einmal adressierbar. Die Identifizierung erfolgt in Anlehnung an die in OCIT-C verwendete Adressierung der Lichtsignalanlagen.

ObjekttypPredictedSpat			
Id		Identifizierung der Lichtsignalanlage	
predictionSource		Namen des Prognoseverfahrens (Herkunft)	
timeline		Enthält UTC Zeitstempel aus global.xsd auf den sich die nachfolgenden Werte beziehen.	
sg (Array)		Anzahl der nachfolgenden Datensätze entspricht maximal der Anzahl von Signalgruppen oder der Anzahl der Signalgruppen für die eine Vorhersage vorliegt.	
	nr		Nummer der Signalgruppe
	forecast (Array)		Anzahl der Feldeinträge entspricht der Anzahl der für diese Signalgruppe verfügbarer Signalgruppenfarbwechsel.
		offset	Zeitoffset zum Timestamp in 100ms Einheiten Zukunft > 0 Historie (wenn erforderlich): <0
		quality	Qualitätsaussage des Prognosewerts für die betreffende Signalgruppe in %
		trafficState	Aktueller bzw. zukünftiger Zustand der Signalgruppe (0=FREI, 1=GESPERRT)
		color	Aktueller bzw. zukünftiger Farbzustand der Signalgruppe (OCIT-Farbcode)
		duration	Prognostizierte Restdauer für den aktuellen bzw. zukünftigen Signalzustand (in 100ms Einheiten)
		min	Minimale Restdauer für den aktuellen bzw. zukünftigen Signalzustand
		max	Maximale Restdauer für den aktuellen bzw. zukünftigen Signalzustand. (Bei Festzeit sind ProgDuration, MinDuration und MaxDuration identisch.)

		transition	Dauer des Übergangs zwischen dem beschriebenen und dem folgenden Zustand (0 = kein Übergang vorhanden)
--	--	------------	--

Eine nicht verfügbare Spat wird über die entsprechenden ErrorCodes der OCIT-C gekennzeichnet.

Anmerkung: Die Einzelabfrage liefert immer den aktuellen Zustand und optional die nächsten n folgenden Signal-Endzustände. Der Zeitstempel in Verbindung mit dem Zeitoffset bezeichnet den Beginn des aktuellen Zustands, d.h. je nach Zeitpunkt der Abfrage kann dieser auch in der Vergangenheit liegen. Der Offset ist dann 0 und der Zustand liegt aktuell bereits an, es ist aber keine Aussage möglich, wie lange der Zustand bereits anliegt. Die übergebenen Duration-Werte beziehen sich auf den Zeitpunkt timestamp + offset.

3.12.5 MAP Topologiedaten

Aktuell fehlt es noch an konkreten Erfahrungen aus praktischen Anwendungen. Daher werden die MAP-Daten so wie sie aus der Spezifikation der ETSI (ASN.1 Format) kommen übernommen. Die Daten werden als Datenblock vom Verkehrsingenieurarbeitsplatz zum Steuergerät übertragen.

Eventuelle Anpassungen getrieben durch Erfahrungen aus der praktischen Anwendung sind zu erwarten.

Da die Daten nur als Datenblock durchgereicht werden, findet in OCIT-C keine geänderte oder detaillierte Checksummenverwaltung für den Block der MAP-Daten statt. Es wird nur nach dem bekannten Algorithmus eine Checksumme über den Block 5 gebildet.

3.13 LSA Versorgungsdaten

Dieser Kommunikationsbaustein dient zur Datenversorgung von Lichtsignalanlagen. Fehlermeldungen für das Planungstool

- asynchrone Rückmeldung bei Versorgungsänderung
- Liste der für ein VA-Verfahren lieferbaren AP-Werte
- Abfrage der Feldgeräteinformation

Detaillierte Beschreibung siehe Dokument „OCIT-C_LSA_Versorgungsdaten“.

3.14 Binärcontainer

Dieser Kommunikationsbaustein dient zur Übertragung von binär codierten Daten. Die Daten sind einem Zeitpunkt (oder Zeitbereich) und einem Objekt zugeordnet. Die Typbezeichnung identifiziert die Art der Daten im Container und ermöglicht damit dem Empfänger die Verarbeitung.

Binärdaten werden mittels vordefinierten Ids identifiziert.

Beispiel: Es sollen sogenannte "Komprimierte Meldungen" einer LSA in dem Container übertragen werden, die Typbezeichnung wäre in diesem Fall "KPM", als Objekt-Id würde die ID der LSA angegeben und der Datencontainer enthält eine komplette komprimierte Meldung, so wie sie von der LSA übertragen wurde.

Empfohlene Identifizierung:

BC<primary key>_<secondary key>

Die Verwendung des primary key ist nicht zwingend.

Verfügbare Objekttypen:

Objekttyp	Schemadatei	Kurzbeschreibung
BinaryContainer	binary_container.xsd	Binärcontainer, dessen Inhalt projektbezogen festgelegt werden kann.

3.15 Projektbezogene Erweiterungen

Projektspezifische Erweiterungen von Kommunikationsbausteinen sind möglich. Folgendes sollte dabei beachtet werden:

- Projektbezogene Datenstrukturen werden mittels eindeutig identifizierbarer projekt- bzw. kundenspezifischer Objekttypen gelöst.

Diese neu eingeführten Objekttypen werden in eigenen Schemadefinitionen abgelegt und beziehen sich auf Standard XML-Datentypen.

Zur Vermeidung von Doppeldeutigkeiten werden vor den verwendeten Objekttypen eigene herstellerspezifische Präfixe eingebaut. Zudem wird damit die Zugehörigkeit des Datentyps gekennzeichnet. Dem Präfix folgt ein underscore (" _ ")

Beispiel:

Hersteller "abc"

Neuer Objekttyp: "xyz"

Neuer resultierender identifizierender Objekttyp: "abc_xyz".

- Erweiterungen im Rahmen der Protokollfunktionen sind unbedingt zu vermeiden.
- Erweiterungen zu existierenden Datentypen sind zu vermeiden. Sollte dies nicht zu vermeiden sein, sind die Kompatibilitätsregeln des nachfolgenden Kapitels zu berücksichtigen.

3.15.1 Kompatibilitätsregeln

Erweiterungsregeln für veröffentlichte Datenmodelle (Schemadefinitionen) und die zugeordneten Objekttypen:

- Elemente und Attribute dürfen in bestehenden Schemadefinitionen hinzugefügt werden.

- Neue Attribute und Elemente müssen optional sein um Rückwärtskompatibilität zu gewährleisten.
- Die konsumierende Schnittstelle muss flexibel auf die Verfügbarkeit der neu eingeführten Elemente/Attribute reagieren.
- Die konsumierende Schnittstelle muss flexibel auf etwaig unbekannte Elemente Attribute reagieren und diese bei Eintreffen ohne weitere Bewertung überlesen. Das bedeutet unbekannte Elemente dürfen das Verhalten der Dekodierung der empfangenen XML Struktur nicht beeinflussen.
- Es ist nicht erlaubt Attribute oder Elemente aus Schemadefinitionen zu löschen.
- Es ist nicht erlaubt die Semantik bestehender Attribute und Elemente zu verändern.

4 Begriffe und Abkürzungen

Begriff / Abkürzung	Beschreibung
AP	Anwenderprogramm
CAM	Cooperative Awareness Message
Car2X	Kommunikation zwischen Fahrzeugen sowie Fahrzeugen mit der Umgebung.
Client	Ein Programm, das von anderen (Servern) angebotene Dienste in Anspruch nehmen will und diese dazu aktiv aufruft.
DATEX II	Spezifikationen des Technischen Komitees 278 des Europäischen Komitees für Normung (CEN) zum Austausch von Verkehrsinformationen zwischen Verkehrszentralen.
DENM	Dencentralized Environmental Notification Message
FTP	File Transfer Protocol, ein Netzwerkprotokoll zur Dateiübertragung
http	Hypertext Transfer Protocol, ein Protokoll zur Übertragung von Daten über ein Netzwerk.
LSA	Lichtsignalanlage
MAP	Topology Info for Intersection and Road Segments
Methode	Die einer Klasse von Objekten zugeordneten Algorithmen. Auch synonym gebraucht als Funktion, Prozedur, Befehl, Aktion.
ÖV	Öffentlicher Nahverkehr
OCIT	Open Communication Interface for Road Traffic Control Systems / Offene Schnittstellen für die Straßenverkehrstechnik.
OCIT-C	Open Communication Interface for Road Traffic Control Systems – Center to Center. Mit OCIT-C werden die Funktionen zur Kommunikation zwischen zentralen Verkehrssteuerungs- und -Verkehrslenkungssystemen abgedeckt.
OCIT-O	OCIT-Outstations Schnittstelle zwischen Verkehrssteuerungszentralen und Lichtsignalsteuergeräten zur Steuerung und Versorgung der Lichtsignalsteuergeräte
ODG	OCIT Developer Group
OSI	OSI-Schichtenmodell (auch OSI-Referenzmodell; englische Open Systems Interconnection Reference Model), ein Kommunikationsmodell der Internationalen Organisation für Normung (ISO) für Kommunikationsprotokolle in Rechnernetzen.

OTS 2	Open Traffic Systems, Version 2
Server	Ein Programm, das bestimmte Dienste anbietet und dazu passiv auf eingehende Aufrufe (von Clients) wartet.
SOAP	SOAP (Simple Object Access Protocol), ist ein Protokoll mit dessen Hilfe Daten zwischen Systemen ausgetauscht werden können. SOAP verwendet „Remote Procedure Call“ und ermöglicht dadurch den Aufruf von Funktionen in anderen Computern. Siehe http://www.w3.org/TR/SOAP
SSL	Secure Socket Layer.
Soap-Server-Interface	Soap and Protocolmanager auf der Server-Seite
Soap-Client-Interface	Soap and Protocolmanager auf der Client-Seite
Protocolmanager	Protokollschicht zur Implementierung von Kommandos im Puffer
TLS	Technische Lieferbedingungen für Streckenstationen. Die TLS sind ein Standard für den Aufbau von Verkehrsbeeinflussungsanlagen an Bundesfernstraßen. Herausgeber: Bundesanstalt für Straßenwesen.
TCP / IP	Transmission Control Protocol / Internet Protocol, eine Familie von Netzwerkprotokollen für das Internet.
VDV	Verband Deutscher Verkehrsunternehmen
WSDL	Web Services Description Language, eine plattform-, programmiersprachen- und protokollunabhängige_Beschreibungssprache für Netzwerkdienste (Webservices) zum Austausch von Nachrichten auf Basis von XML.
XML	Extensible Markup Language, eine Auszeichnungssprache zur Darstellung strukturierter Daten in Form von Text. XML wird u. a. für den plattform- und implementationsunabhängigen Austausch von Daten zwischen Computersystemen eingesetzt. Ein XML-Dokument besteht aus Textzeichen, im einfachsten Fall in ASCII-Kodierung, und ist damit maschinenlesbar. Binärdaten enthält es nicht. Die XML-Spezifikation wird vom World Wide Web Consortium (W3C) als Empfehlung (Recommendation) herausgegeben.
XSD	XML Schema, eine Empfehlung des World Wide Web Consortium (W3C) zum Definieren von Strukturen für XML-Dokumente. Die Struktur wird in Form eines XML-Dokuments beschrieben. Darüber hinaus wird eine große Anzahl von Datentypen unterstützt. In der XSD Schemasprache werden Datentypen, einzelne XML-Schema-Instanzen (Dokumente) und Gruppen solcher Instanzen beschrieben. Ein konkretes XML Schema wird auch als eine XSD (XML Schema Definition) bezeichnet und hat als Datei üblicherweise die Endung „.xsd“.

Weitere Erklärungen zu fachtechnischen Begriffen und Abkürzungen die in diesem Dokument verwendet werden, finden Sie im Dokument „OCIT – O Glossar V3.0“.

OCIT-C_Daten_V2.1

Copyright © 2025 ODG & Partner
