

OCIT[®]

Open Communication Interface for Road Traffic Control Systems

Offene Schnittstellen für die Straßenverkehrstechnik

OCIT-C Center to Center

Daten

OCIT-C_Daten_V1.2_R2

OCIT Developer Group (ODG)&Partner

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

OCIT-C Center to Center

Daten

Document: OCIT-C_Daten_V1.2_R2

Herausgeber: ODG & Partner

Kontakt: www.ocit.org

Copyright © 2018 ODG. Änderungen vorbehalten. Dokumente mit Versions- oder Ausgabe-
stände neueren Datums ersetzen alle Inhalte vorhergehender Versionen.

Inhalt

Einführung.....	7
1.1 Begriffe und Abkürzungen	10
2 Objekte und Schemadefinitionen.....	12
2.1 Objekt.....	12
2.2 Objekttypen	12
2.3 Objektidentifikation	12
2.3.1 Objektidentifizierung	13
2.3.2 Objektfilter	14
2.4 Gemeinsame Strukturen (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 Datenkatalog.....	19
3.1 Verkehrsmeldungen	19
3.2 Verkehrsdaten	20
3.2.1 Unterabschnitt, Abschnitte, Bereiche	21
3.2.2 Detektoren und Detektorgruppen.....	22
3.2.2.1 Objekttyp: TrafficData_detector_Description	22
3.2.2.2 Objekttyp: TrafficData_detector_currentValue.....	23
3.2.2.3 Objekttyp: TrafficData_detectorGroup_Description	24
3.2.2.4 Objekttyp: TrafficData_detectorGroup_currentValue.....	24
3.2.2.5 Objekttyp: TrafficData_detectorGroup_calculatedValue	24
3.2.2.6 Objekttyp: TrafficData_trafficSubSection_Description	25
3.2.2.7 Objekttyp: TrafficData_trafficSection_Description	25
3.2.2.8 Objekttyp: TrafficData_trafficArea_Description.....	25

3.2.2.9	Objektyp: TrafficData_trafficSubSection_accumulatedValue.....	26
3.2.2.10	Objektyp: TrafficData_trafficSection_accumulatedValue	26
3.2.2.11	Objektyp: TrafficData_trafficArea_accumulatedValue	26
3.3	Parkdaten	27
3.4	Wetter- und Umweltdaten	28
3.5	Kameras	28
3.6	Situationen und Strategien	30
3.7	Betriebsmeldungen.....	31
3.8	Schilder	34
3.8.1	Datenmodell.....	34
3.8.2	Komplexe Schildinhalte	36
3.9	ÖV-Fahrgastinformation	38
3.10	Lichtsignalanlagen	38
3.10.1	Vordefinierte Kommandos (Predefined Commands) - Veraltet.....	39
3.10.2	Schaltbarkeiten (IntersectionFeatures)	40
3.10.3	IntersectionCommand (Ausmodellierete Schaltungen).....	41
3.10.4	Steuerungs-Parameter (ControlMethodParameter).....	43
3.10.5	LSA Zustand (IntersectionStatus)	43
3.10.6	SGTimes.....	44
3.11	LSA Rohdaten.....	44
3.11.1	Detektorflanken.....	45
3.11.2	Signalgruppenzustände	46
3.11.3	Digitale Ausgänge im Lichtsignalsteuergerät	47
3.11.4	Anwenderprogrammwerte (AP-Werte)	49
3.11.4.1	Einstellung abfragbarer AP-Werte	51
3.11.5	Erweitertes ÖV-Telegramm.....	51
3.11.6	Einzeldetektordaten	52
3.12	LSA Versorgungsdaten	53
3.13	Binärcontainer.....	53

3.14	Projektbezogene Erweiterungen	54
3.14.1	Kompatibilitätsregeln.....	54

Dokumentenstand

Version Zustand	Datum	Verteiler	Kommentar
V1.1_R1	30.10.2014	PUBLIC	Version 1.1 Ausgabe 1
V1.2_R1_D1	21.09.2016	ODG Intern	2.3.1: Objektidentifikation aktualisiert 2.3.2: Objektfilter neu
			Referenzierung von OCIT-I auf OCIT-C geändert
V1.2_R1_D2		28.09.2016	2.3.1 Textänderung: OITD-Nummern sind in IP- Schreibweise auszuführen
		14.12.2016	3.11.4: Änderung der Umlaufsekunde (TX)im Beispiel von 1 auf 10
V1.2_R1_D3		22.11.2016	2.3.2 Textergänzung: Objektfilter sind eine Standardfunktion des Servers
		22.01.2018	3.11.4: Anwenderprogrammwerte (AP-Werte), Tabelle der standardisierte AP-Werte eingefügt

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 Info 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 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.
- **LSA-Versorgungsdaten**
Dieser Kommunikationsbaustein dient zur Datenversorgung von Lichtsignalanlagen.
- **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 3.14).

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

1.1 Begriffe und Abkürzungen

Begriff / Abkürzung	Beschreibung
AP	Anwenderprogramm
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.
FTP	File Transfer Protocol, ein Netzwerkprotokoll zur Dateiübertragung
http	Hypertext Transfer Protocol, ein Protokoll zur Übertragung von Daten über ein Netzwerk.
LSA	Lichtsignalanlage
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; englisch 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 ver-

	wendet „Remote Procedure Call“ und ermöglicht dadurch den Aufruf von Funktionen in anderen Computern. Siehe http://www.w3.org/TR/SOAP
SSL	Secure Socket Layer.
SoapServer-Interface	Soap and Protocolmanager auf der Server-Seite
SoapClient-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“.

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 Annotations 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. 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 Objektidentifizierung

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 soll wie folgt aufgebaut sein:

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

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 auszuführen. (57.102)

Ist keine Objektnummer vorhanden oder ist es ein einmaliges Objekt wird „1“ als Objekt-
nummer 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

Objektfilter sind eine Standardfunktion des Servers und müssen wie hier spezifiziert unterstützt werden. Beim Datenabruf durch den Client 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

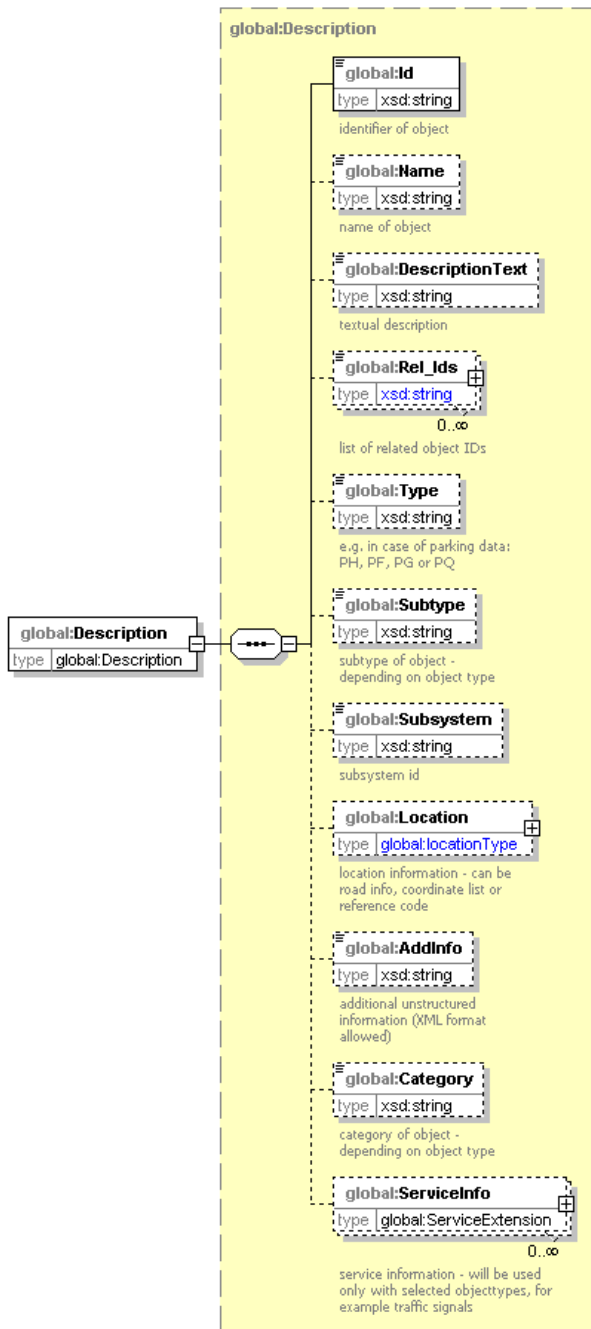
2.4 Gemeinsame Strukturen (global.xsd)

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.

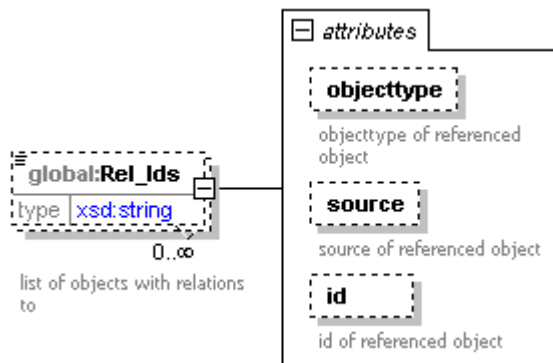
2.4.1 Element Description

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.

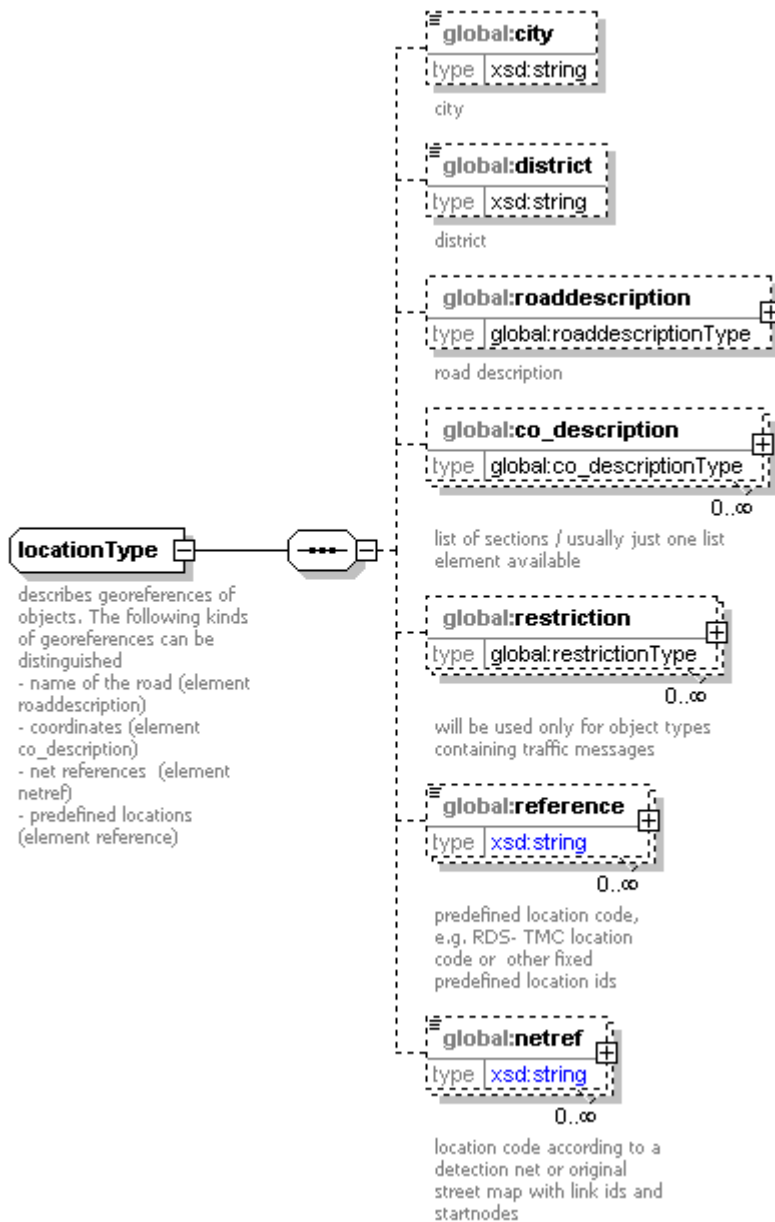


Name	Typ	Verwendung	Beschreibung
objecttype	xsd:string	optional	Objektyp des referenzierten Objekts
source	xsd:string	optional	Quelle des referenzierten Objekts
id	xsd:string	optional	Identifizierung des referenzierten Objekts

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

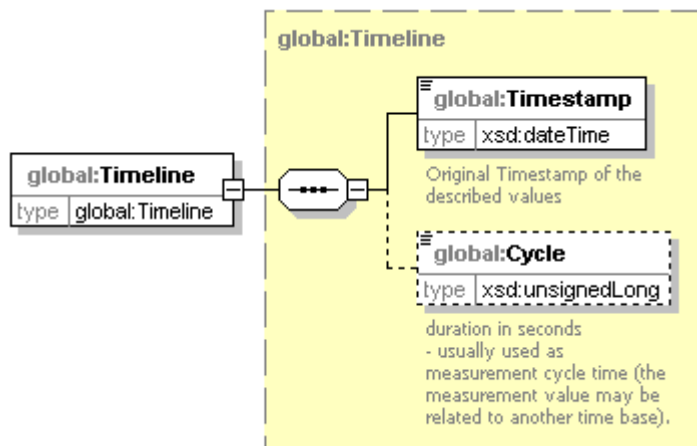
Dynamische Daten haben einen Status. Dieser wird mittels State dargestellt. Alle dynamischen Daten verwenden die global definierte *State*.



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.

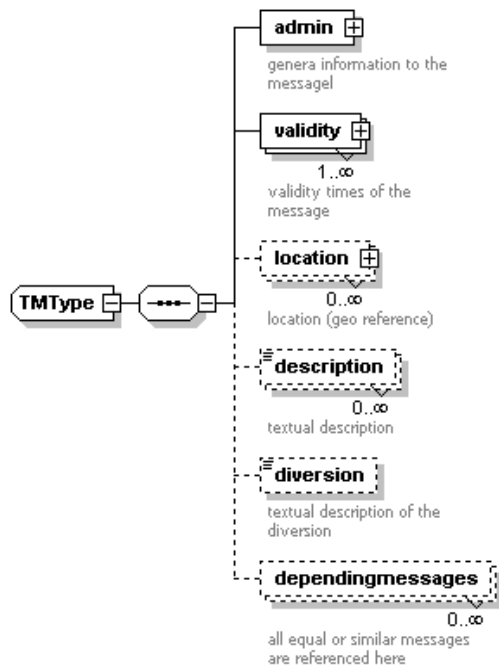
3.1 Verkehrsmeldungen

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

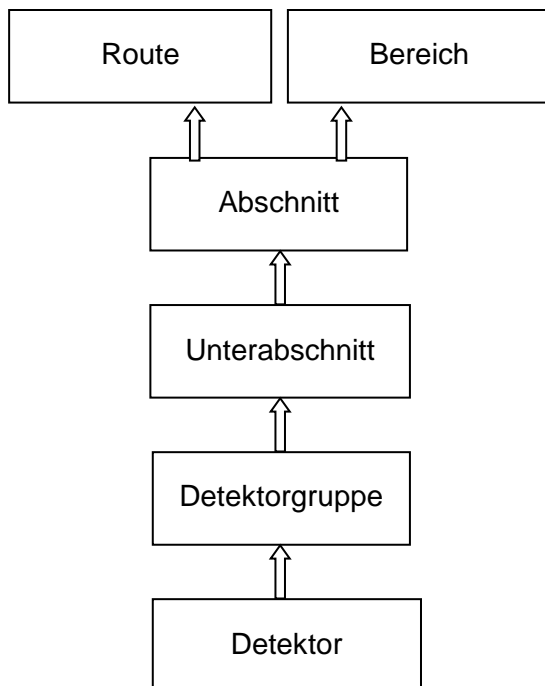
Dieser Kommunikationsbaustein dient zur Übertragung von Messwerten oder abgeleiteten Werten verschiedener Detektionseinrichtungen. Übertragen werden aktuelle und akkumulierte Messergebnisse von Einzeldetektoren oder Detektorgruppen, jedoch keine Detektorrohdaten (Impulse).

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
TrafficData_trafficSection_Description		Beschreibende Daten Strecken
TrafficData_trafficArea_Description		Beschreibende Daten Bereiche

Objektyp	Schemadatei	Kurzbeschreibung
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 Identifizierung:

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 auf Seite 13
- 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.

Zur Identifizierung von Detektoren, die über OCIT-C VD versorgt werden, wird nur der nach dem im Kapitel 3.11 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**

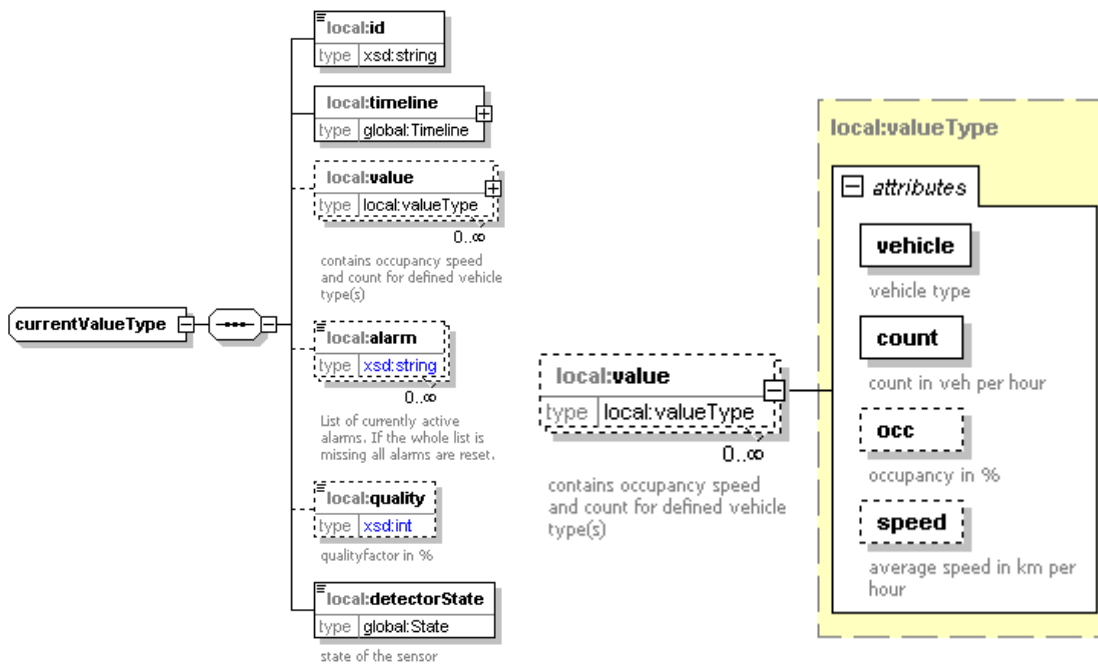
- Hersteller/Lieferant.

Verwendung Element **Rel_Ids**

- Verweis auf Messstellen

- Verweis auf LSA

3.2.2.2 Objekttyp: TrafficData_detector_currentValue

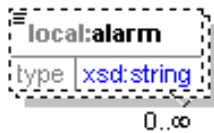


Für jeden verfügbaren Fahrzeugtyp wird die Struktur „value“ als Feldelement aufgenommen. Nichtvorhandene Fahrzeugtypen sollen nicht als Feldelement aufgenommen werden.

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
<xsd:enumeration value="delivery_truck"/>	Lieferwagen



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

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"

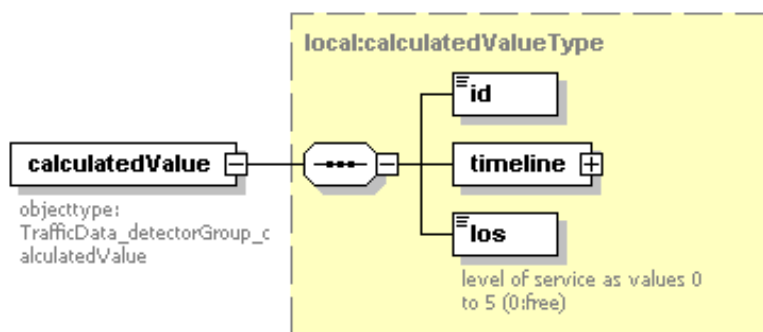
3.2.2.3 Objekttyp: TrafficData_detectorGroup_Description

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

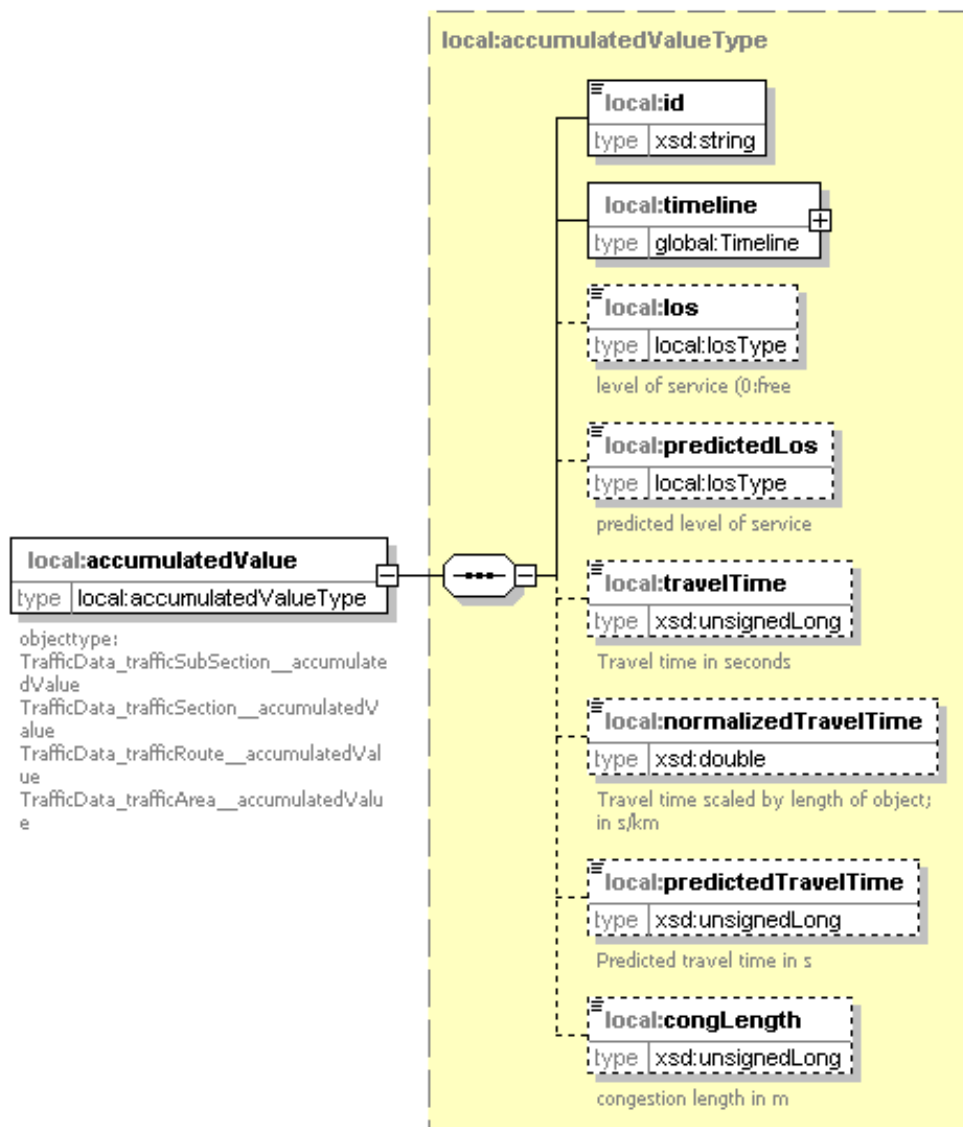
3.2.2.7 Objekttyp: TrafficData_trafficSection_Description

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 Parkeinrichtungen, Parkquartieren, Parkflächen.

Parkdaten werden mit vordefinierten Ids referenziert.

Empfohlene Identifizierung:

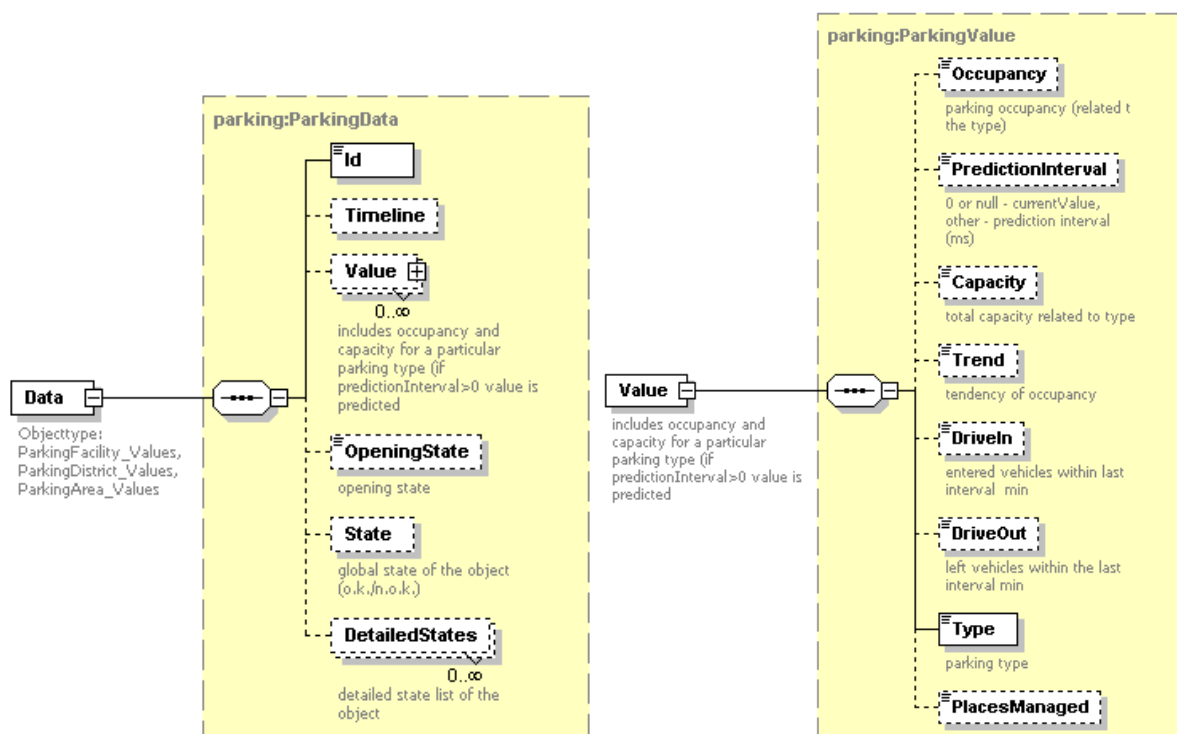
PF<primary key>_<secondary_key> für Parkeinrichtungen

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 Parkiereinrichtung
ParkingDistrict_Description		Beschreibende Daten Parkquartier
ParkingArea_Description		Beschreibende Daten Parkfläche
ParkingFacility_Values		Dynamische Daten Parkiereinrichtung
ParkingArea_Values		Dynamische Daten Parkquartier
ParkingDistrict_Values		Dynamische Daten Parkfläche



3.4 Wetter- und Umweltdaten

Dieser Kommunikationsbaustein dient zur Übertragung von Daten entsprechender Erfassungseinrichtungen.

Umwelt- und Wetterdaten werden mit vordefinierten Ids identifiziert.

Empfohlene Identifizierung:

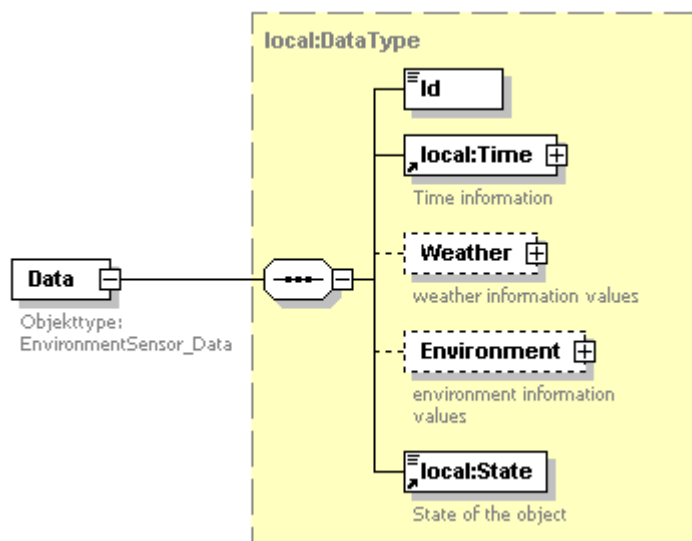
W<primary key>_<secondary_key> für Wettersensoren

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.

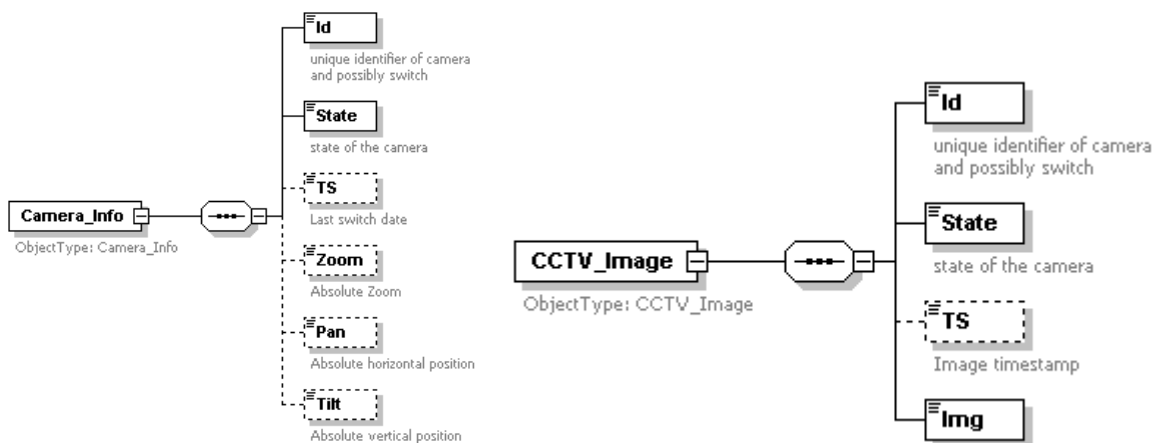
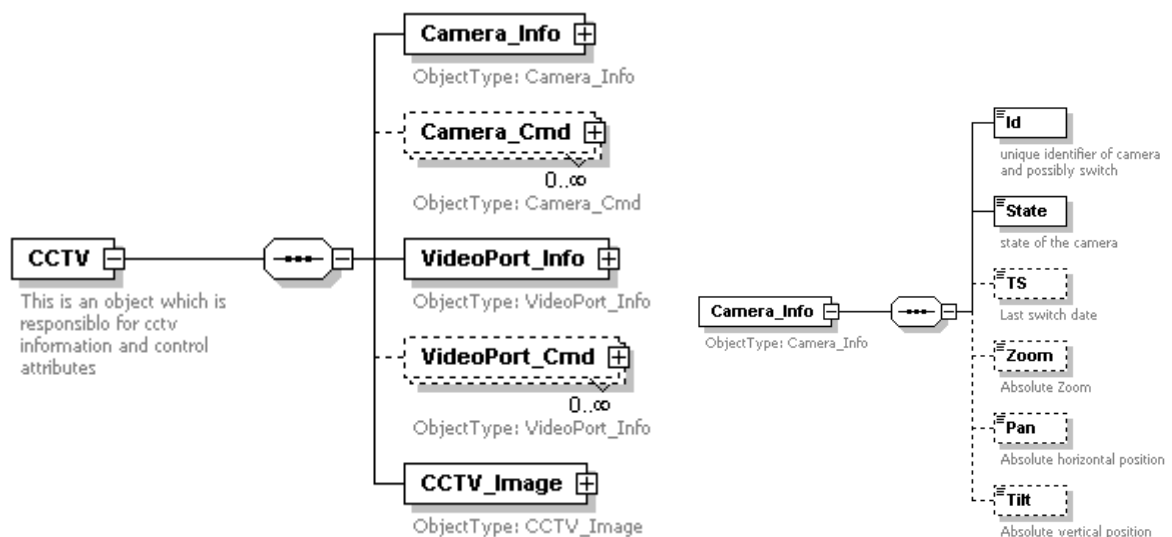
Empfohlene Identifizierung:

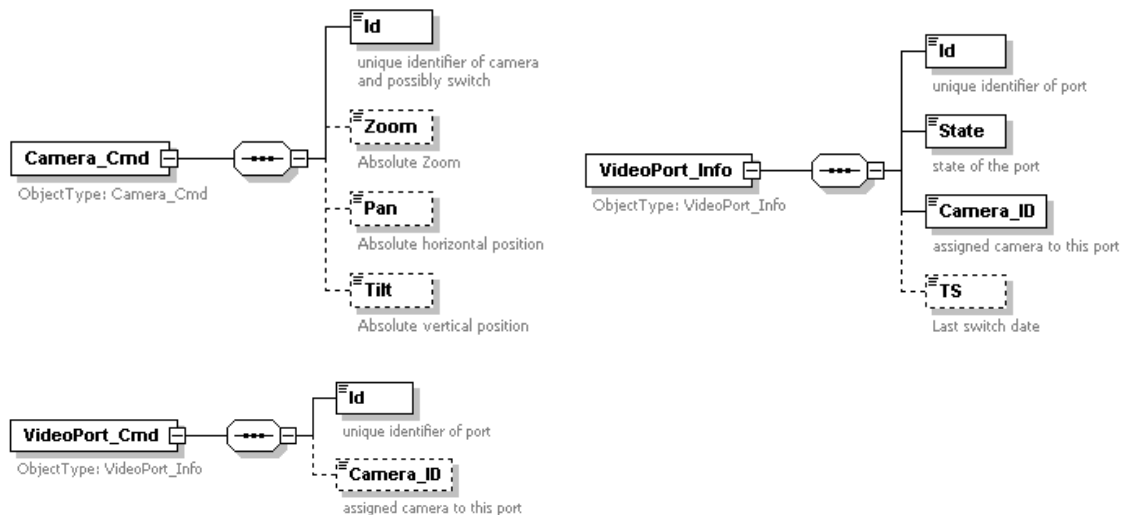
Cam<primary key>_<secondary key>

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.

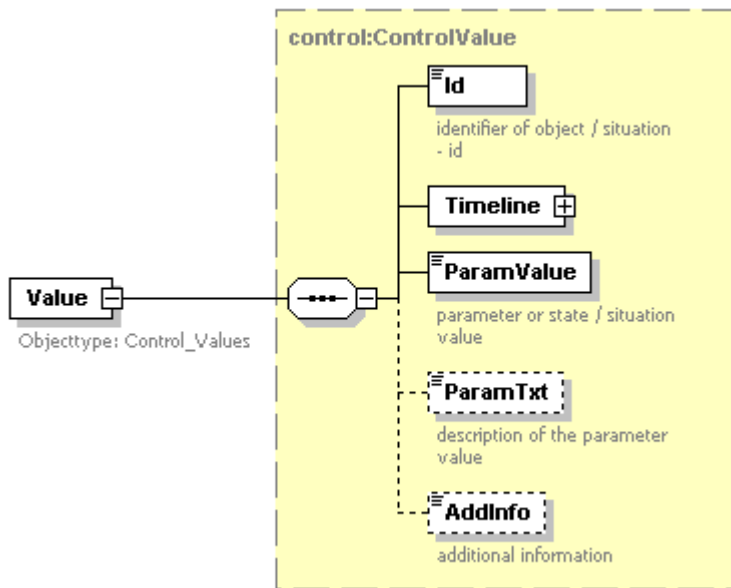
Situationen oder Strategien werden mit vordefinierten Ids identifiziert.

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).

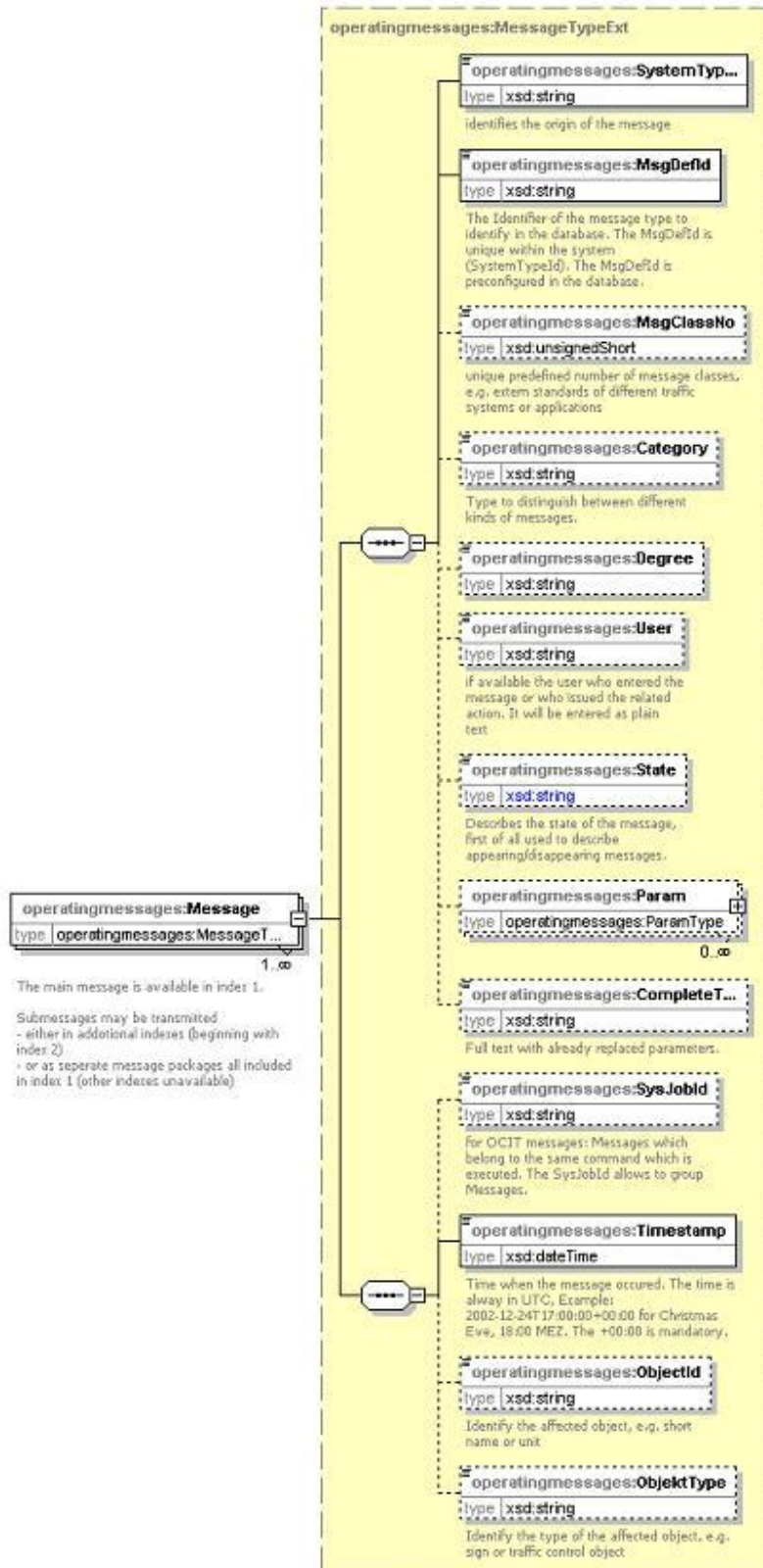
3.7 Betriebsmeldungen

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 Nebenmeldungen. Die Hauptmeldung steht an erster Stelle. Anschließend folgen die Nebenmeldungen. Die Meldungsbestandteile einer Haupt- bzw. Nebenmeldung werden im Folgenden beschrieben



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
SysJobId	Bei Meldungen die Ursache einer bestimmten Aktion sind, tragen sowohl die Aktion als auch die Meldung die gleiche SysJobId. Die SysJobId gruppiert die Meldungen nach bestimmten Ursachen / Verursachern.
Timestamp	Zeitstempel vom Entstehungszeitpunkt der Meldung
ObjectId	Identifizierung des betreffenden Objekts, z. B. Kurzbezeichnung
Objekttyp	Identifizierung des betroffenen Objekttyps, z. B. Schild oder Verkehrssteuerungsobjekt

Man kann verschachtelte und Array-Parameter dadurch angeben, dass man die ParamId 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 ParamId zu erzeugen.

3.8 Schilder

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

Verfügbare Objekttypen:

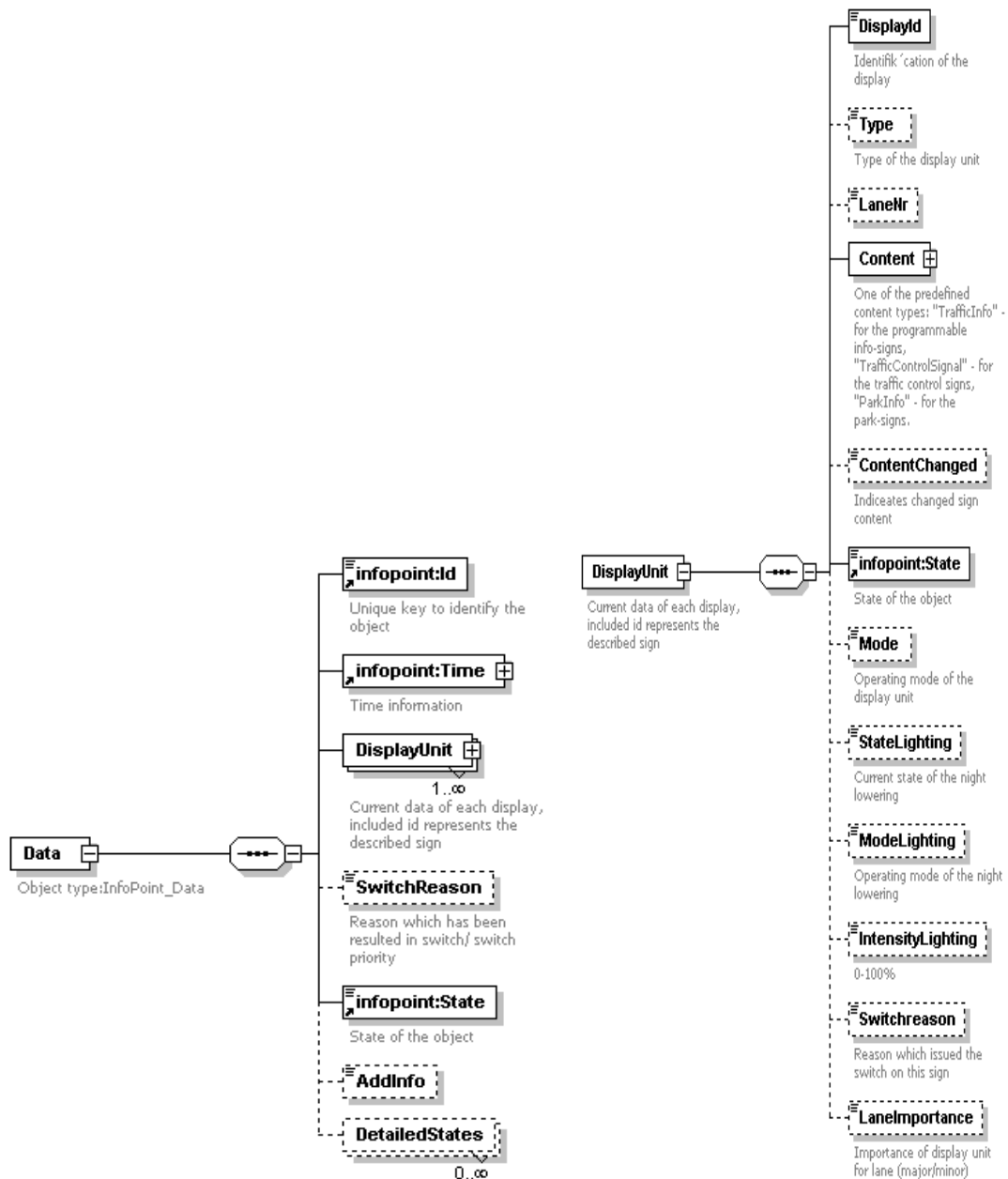
Objekttyp	Schemadatei	Kurzbeschreibung
Infopoint_Description	infopoint_data.xsd	Beschreibende Daten Schilder
InfoPoint_Data	(importiert sign.xsd)	Schilderinhalt (Istzustand)
InfoPoint_Data		Schilderinhalt (Sollzustand)
Infopoint_Cmd		Alternative Stellbefehl für Schilder

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.

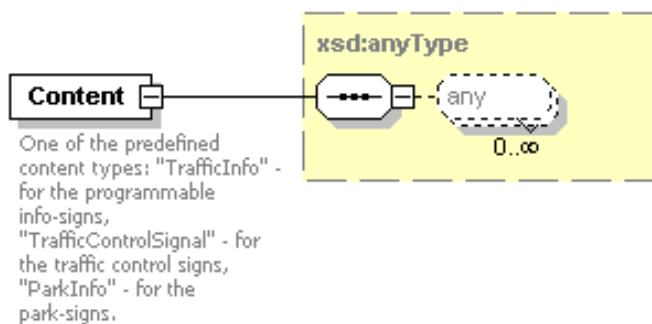


Tag	Inhalt	Beschreibung
Id		Identifizierung des Schildes
Time	Timestamp	In UTC
DisplayUnit	sub elements	Tritt nur auf - einmal bei Vollmatrix-Displays - mehrmals bei Segment-Displays
DisplayId	-	Kennzeichnet Segment-Displays
Type	-	Unbenutzt

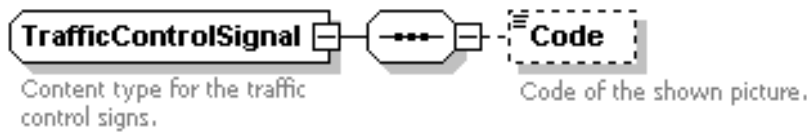
Tag	Inhalt	Beschreibung
Content	Wie in sign.xsd definiert	Siehe sign.xsd
ContentChanged	-	Unbenutzt
State	-	Unbenutzt an dieser Stelle
Mode	-	Unterscheidung, ob der Anteil vom Schild von der Zentrale oder lokal angesteuert wird
StateLighting		Unbenutzt
SwitchReason	-	Unbenutzt
State	o.k. n.o.k. busy	Statusinformation vom Schilder-Server
	switch	Kommandoanzeige vom Schild zum Schilder-Server
AddInfo		Zusätzliche Information (optional).
On_off	on off	

3.8.2 Komplexe Schildinhalte

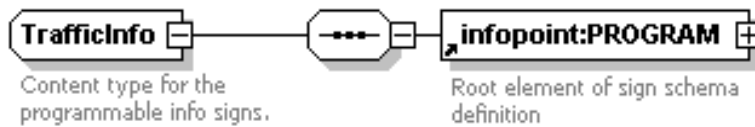
Die sign.xsd modelliert komplexe Schildinhalte. Die Verbindung zum Schema in-fopoint_data.xsd wird mittels any-type realisiert:



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



- eine programmierbare Matrixdefinition oder base 64 kodierter picture stream

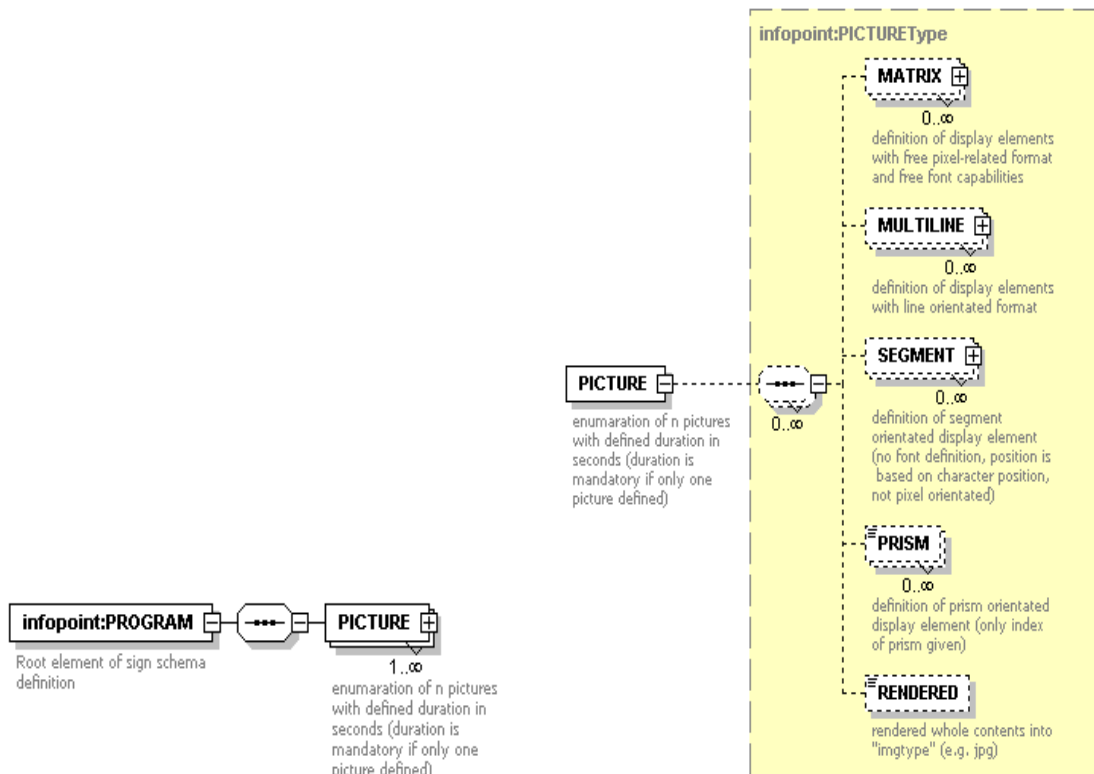


- eine Parkinformation beschreiben.

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)

Objekttyp	Schemadatei	Kurzbeschreibung
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

Kommando	Belegung des XML Elements
	subintersection3_on; subintersection3_off; subintersection3_local
Sonderkommandos	specialcommand<number> localspecialcommand<number>
Projektspezifische Modifikation	projectspecificcommand_<name or number> localprojectspecificcommand_<name or number>
Freigabe	release

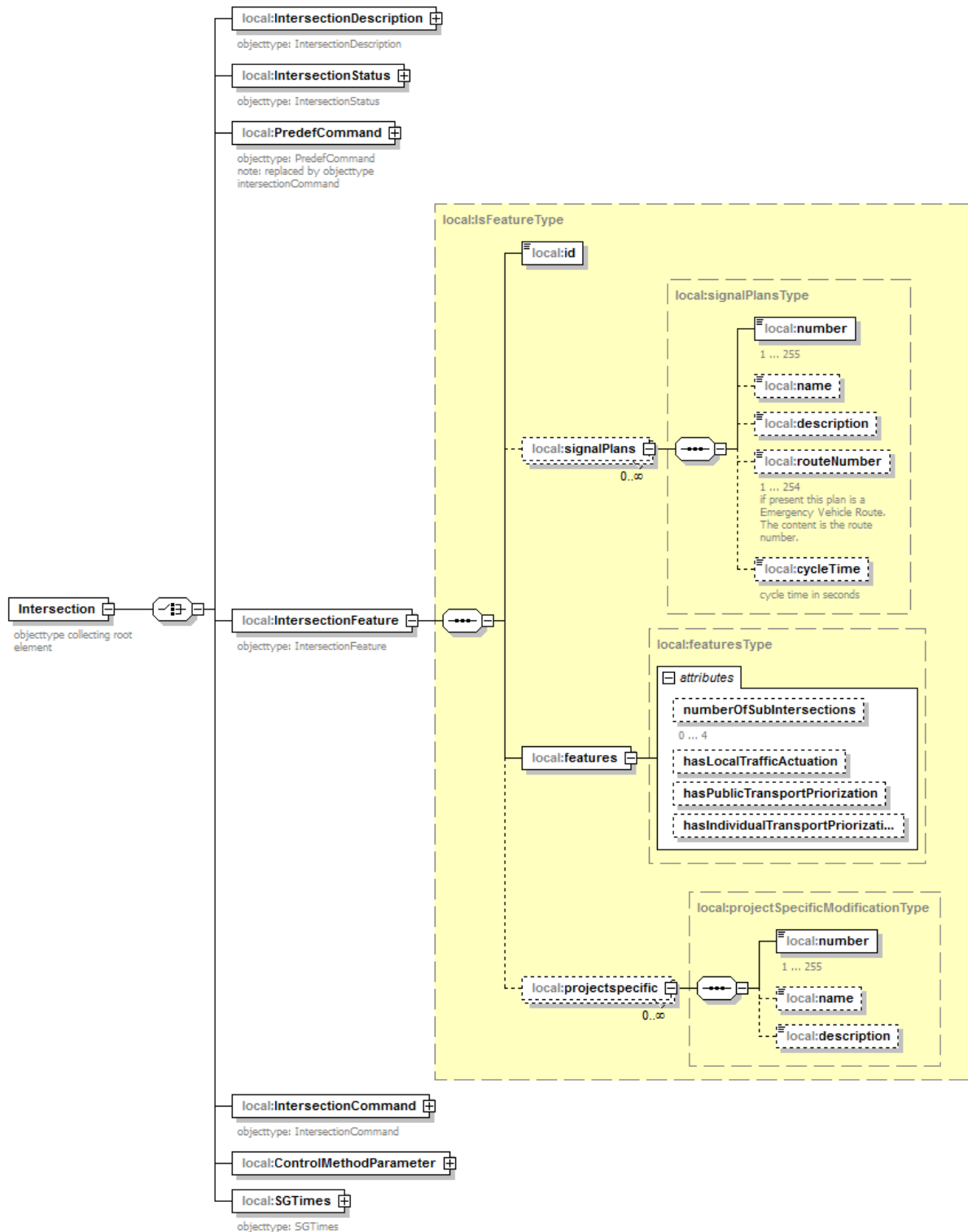
Projektbezogen können weitere Kommandos im Rahmen des Projektes hinzugefügt werden.

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 werden (und in diesem Zuge abgelehnt werden).

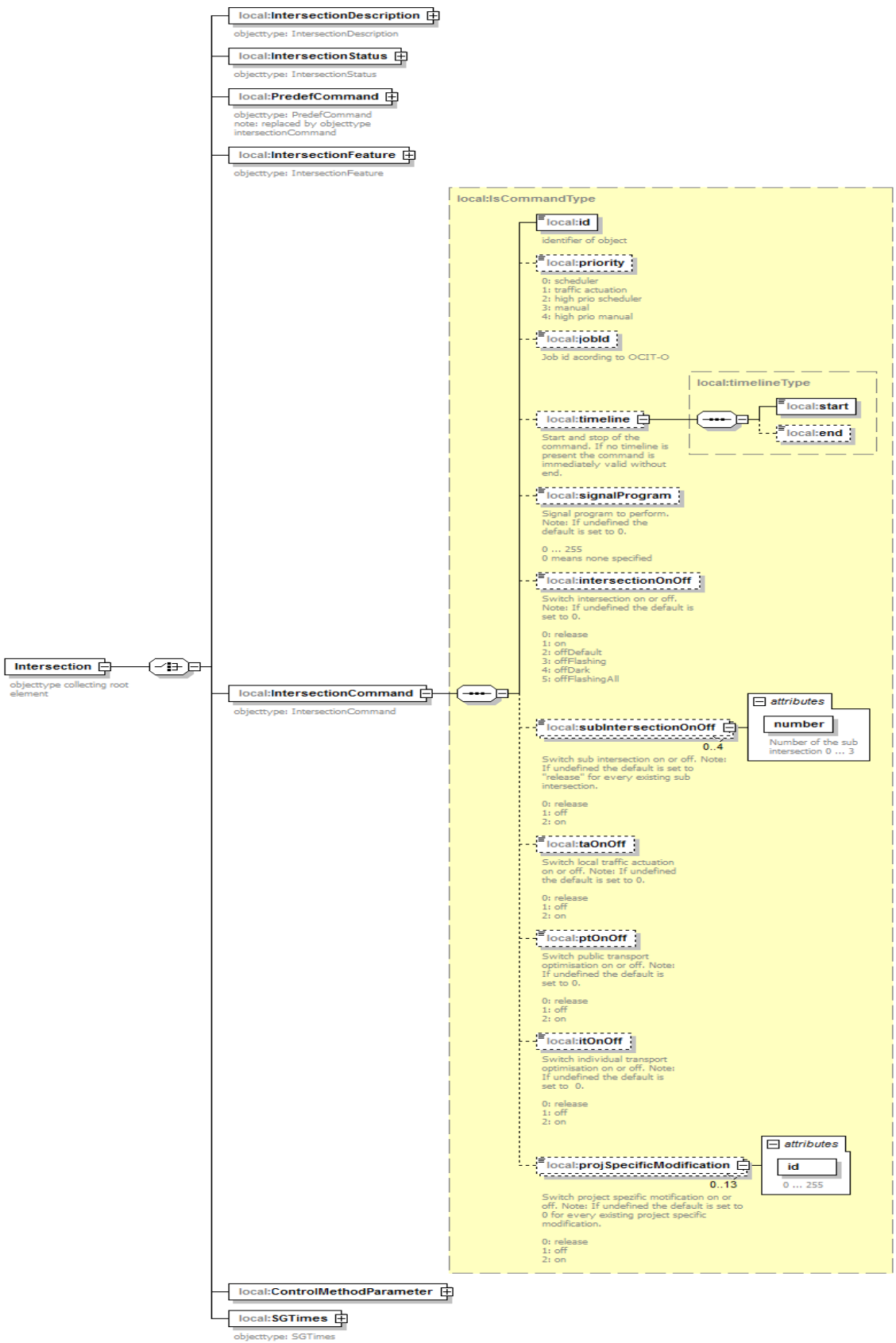


Generated by XMLSpy

www.altova.com

3.10.3 IntersectionCommand (Ausmodellerte Schaltungen)

Mittels dem Objekttyp IntersectionCommand lassen sich Schaltungen auslösen.



3.10.4 Steuerungs-Parameter (ControlMethodParameter)

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

Parameter	Verfahren
path	Pfad für ApWertName in OCIT-O, z.B. VS-PLUS.NETZSTEUERUNG
methodname	Name des Steuerungsverfahrens, das die Daten erhalten soll, z.B. VS-PLUS
interfacename	Name der Schnittstelle im Steuergerät, z.B. NETZSTEUERUNG
data	Binärer Dateninhalt für APWertBlock (1:508)

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“:

TextNr: Eine Nummer die den Inhalt kennzeichnet

Text: Optionaler Texteintrag

Params: Parameter zur weiteren Detaillierung

- Zeitquelle (Information über die Zeitquelle der Gerätezeit)
TextNr: 1001
Text: "Zeitquelle"
Params: 0 (=Netzspannung), 1 (=Quarzuhr), 2 (=LAN), 3 (=DCF), 4 (=GPS), 5 (=Unbekannt)
- NotAus (Information ob der NOTAUS-Schalter betätigt wurde)
TextNr: 1002
Text "NotAus"
- TuerAuf (Information ob der Türschließkontakt betätigt wurde)
TextNr: 1003
Text: "TuerAuf"
- NetzspannungOK (Information ob die Netzspannung vorhanden ist)
TextNr: 1004
Text „NetzOK“
- PersistenzspeicherOK (Information ob der persistente Speicher in Ordnung ist)
TextNr: 1005
Text: "PersistenzspeicherGestoert"
- Lampenstoerung (Information über gestörte Lampen)
TextNr: 1006
Text: "Lampenstoerung"

Params: Ein Parameter pro gestörter Lampe mit Inhalt "<Relknoten>-<Signalgruppe>-<Signalgeber>-<Kammer>", z.B. "1-8-2-3"

- Detektorstörung (Information über gestörte Detektoren)
TextNr: 1007
Text: "Detektorstoerung",
Params: Ein Parameter pro gestörtem Detektor mit Inhalt Kanalnummer, z.B. "5"

Sammelstörung aus OCIT-O wird unter faultState aufgeführt.

TextNr: 1008
Text: Sammelstoerung
Params: Ein Parameter. Die Werte: KeineStoerung, Stoerung, StoerungMitAbschaltung, StoerungMitTeilabschaltung, InterneStoerung

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.

3.11 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 auf Seite 13

Verfügbare Objekttypen:

Objekttyp	Schemadateien	Kurzbeschreibung
RawTrafficDataBlock_Detectoredge		Detektorflanken
DigOut_Raw_Values		Digitale Ausgänge

Objektyp	Schemadateien	Kurzbeschreibung
RawTrafficDataBlock_Signalgroupvalue	intersection_raw_data.xsd	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	Einzeldetektor- daten

3.11.1 Detektorflanken

Das Objekt detectorEdges (Objektyp 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

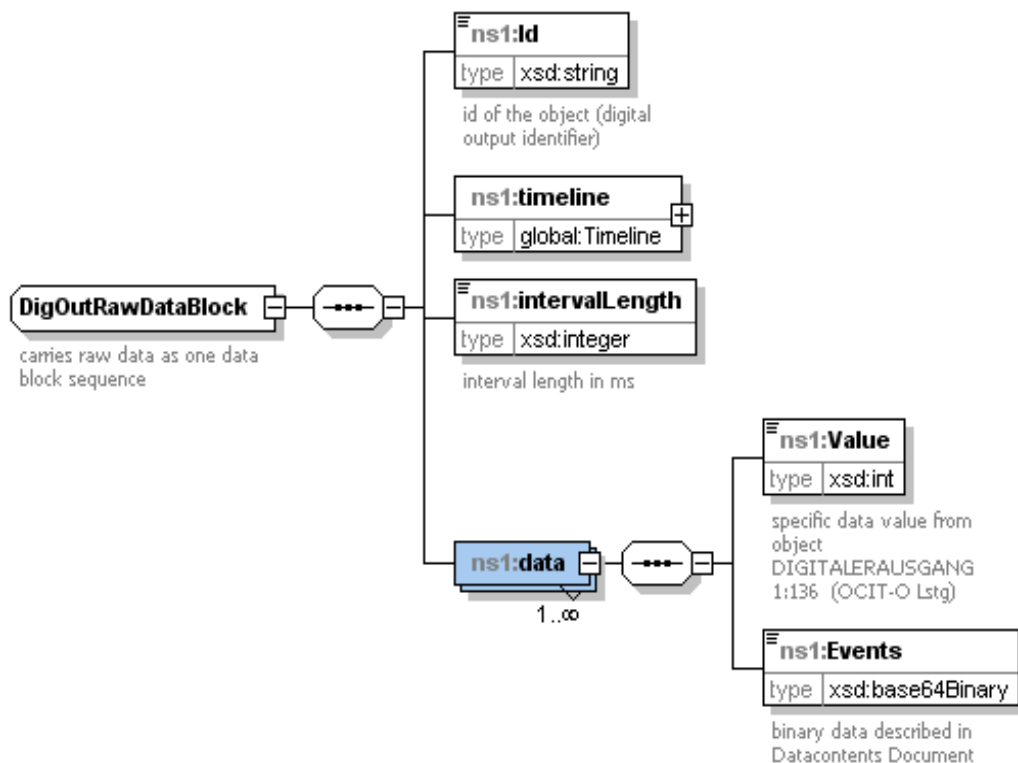
2: 70000 / 1000 == 0x0046

3: 130000 / 1000 == 0x0082

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

3.11.3 Digitale Ausgänge im Lichtsignalsteuergerät

Das Objekt DigOutRawType (Objektyp DigOut_Raw_Values) wurde für die Übertragung der Zustände des digitalen Ausganges 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
intervallLength: 100
data/Value: 3
```


Events:

1: 100 / 100 == 0x0001

2: 1200 / 100 == 0x000c

3: 2000 / 100 == 0x0014

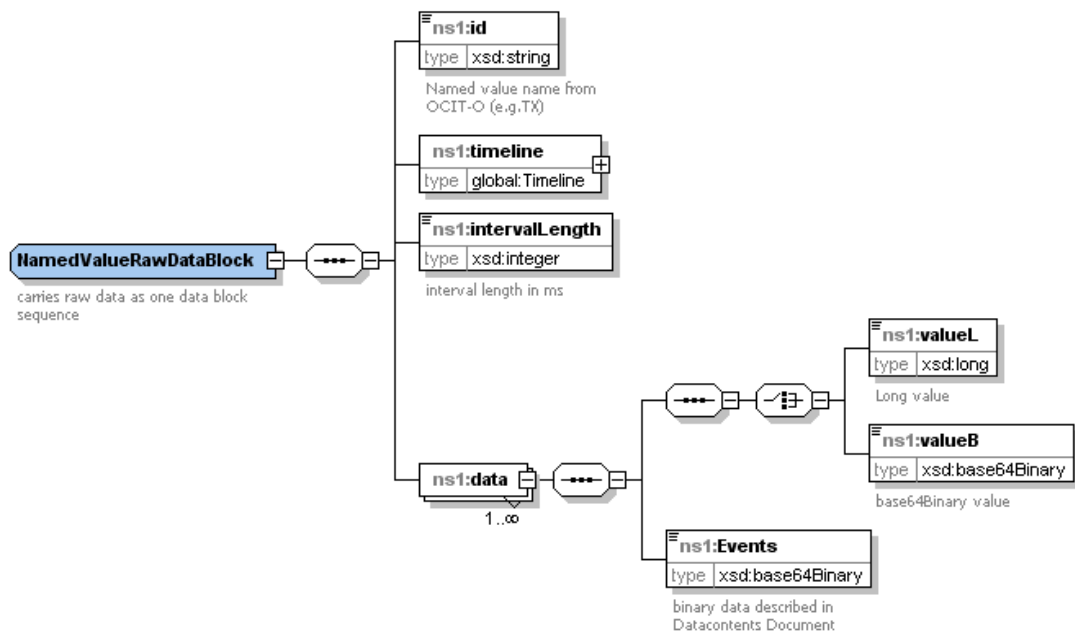
data/Events: AAEADAAU (0001000c0014 in Base64 kodiert)

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 OI-TDdef/OITDdef/Verfahren/OITD/Bezeichner/KurzBez definiert.

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.



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

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: 10

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 Einstellung 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 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“

3.11.5 Erweitertes ÖV-Telegramm

Das Objekt PTExtendedTelegramType (Objekttyp PublicTransportTelegramExtended) 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

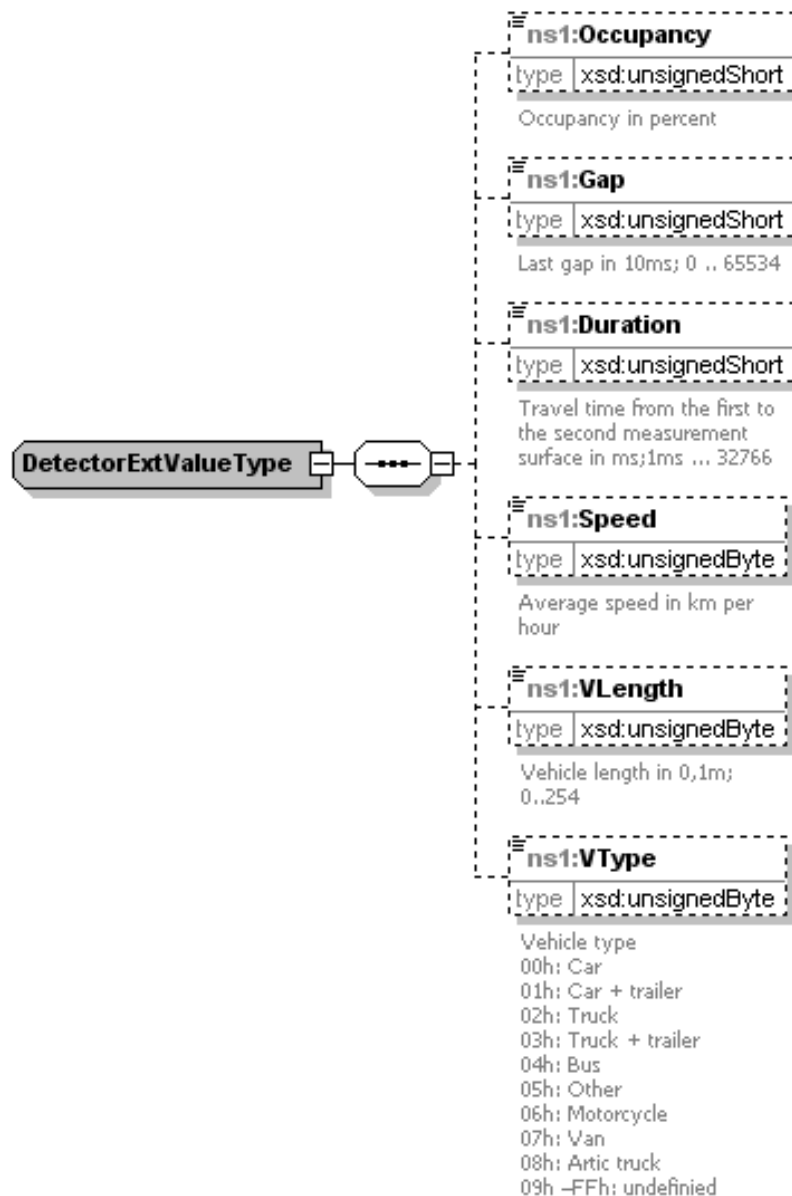
Das Objekt DetectorExtValueType (detector_ext.xsd) wurde für die Übertragung der Zusatzinformationen des Detektors definiert. Der Messwert bezieht sich auf das Detektorereignis eines einzelnen Fahrzeugs.

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)

3.12 LSA Versorgungsdaten

Dieser Kommunikationsbaustein dient zur Datenversorgung von Lichtsignalanlagen. Die Methoden und Daten für Versorgungsdaten von OCIT- C und OCIT-Outstation Version 2 wurden vollständig übernommen und ergänzt mit:

- 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.

3.13 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
-----------	-------------	------------------

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

3.14 Projektbezogene Erweiterungen

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

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

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

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.

3.14.1 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.

OCIT-C_Daten_V1.2_R2

Copyright © 2018 ODG & Partner
