

**DIN EN IEC 62656-8
(VDE 0040-8-8)**
DIN

Diese Norm ist zugleich eine **VDE-Bestimmung** im Sinne von VDE 0022. Sie ist nach Durchführung des vom VDE-Präsidium beschlossenen Genehmigungsverfahrens unter der oben angeführten Nummer in das VDE-Vorschriftenwerk aufgenommen und in der „etz Elektrotechnik + Automation“ bekannt gegeben worden.

VDE

ICS 35.240.50

Einsprüche bis 2019-03-18

Entwurf

**Standardisierte Übertragung und Registrierung von Ontologien für
Produkte mittels Datenpaketen –
Teil 8: Webservice-Schnittstelle für Datenpakete
(IEC 3D/317/CDV:2018);
Deutsche und Englische Fassung prEN IEC 62656-8:2018**

Standardized product ontology register and transfer by data parcels –
Part 8: Web service interface for data parcels
(IEC 3D/317/CDV:2018);
German and English version prEN IEC 62656-8:2018

Registre d'ontologie de produit normalisé et transfert par paquets de données –
Partie 8: Interface de service Web pour les paquets de données
(IEC 3D/317/CDV:2018);
Version allemande et anglaise prEN IEC 62656-8:2018

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2019-01-18 wird der Öffentlichkeit zur Prüfung und Stellungnahme vorgelegt.

Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses Entwurfs besonders zu vereinbaren.

Stellungnahmen werden erbeten

- vorzugsweise online im Norm-Entwurfs-Portal von DIN unter www.din.de/go/entwuerfe bzw. für Norm-Entwürfe der DKE auch im Norm-Entwurfs-Portal der DKE unter www.entwuerfe.normenbibliothek.de, sofern dort wiedergegeben;
- oder als Datei per E-Mail an dke@vde.com möglichst in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter www.din.de/go/stellungnahmen-norm-entwuerfe oder für Stellungnahmen zu Norm-Entwürfen der DKE unter www.dke.de/stellungnahme abgerufen werden;
- oder in Papierform an die DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE, Stresemannallee 15, 60596 Frankfurt am Main.

Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevanten Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen.

Gesamtumfang 148 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE

Anwendungsbeginn

Anwendungsbeginn dieser Norm ist

Nationales Vorwort

Die Deutsche Fassung des europäischen Dokuments prEN IEC 62656-8:2018 „Standardisierte Übertragung und Registrierung von Ontologien für Produkte mittels Datenpaketen – Teil 8: Webservice-Schnittstelle für Datenpakete“ (Entwurf in der Umfrage) ist unverändert in diesen Norm-Entwurf übernommen worden.

Die Internationale Elektrotechnische Kommission (IEC) und das Europäische Komitee für Elektrotechnische Normung (CENELEC) haben vereinbart, dass ein auf IEC-Ebene erarbeiteter Entwurf für eine Internationale Norm zeitgleich (parallel) bei IEC und CENELEC zur Umfrage (CDV-Stadium) und Abstimmung als FDIS (en: Final Draft International Standard) bzw. Schluss-Entwurf für eine Europäische Norm gestellt wird, um eine Beschleunigung und Straffung der Normungsarbeit zu erreichen. Dem entsprechend ist das internationale Dokument IEC 3D/317/CDV:2018 „Standardized product ontology register and transfer by data parcels – Part 8: Web service interface for data parcels“ unverändert in den Entwurf prEN IEC 62656-8:2018 übernommen worden.

Da die Deutsche Fassung noch nicht endgültig mit der Englischen und Französischen Fassung abgeglichen ist, ist die englische Originalfassung des IEC-CDV entsprechend der diesbezüglich durch die IEC erteilten Erlaubnis beigefügt. Die Nutzungsbedingungen für den deutschen Text des Norm-Entwurfes gelten gleichermaßen auch für den englischen IEC-Text.

Das internationale Dokument wurde vom SC 3D „Product properties and classes and their identification“ der Internationalen Elektrotechnischen Kommission (IEC) erarbeitet.

Bei der Abstimmung zu dem Europäischen Schluss-Entwurf bei CENELEC und dem Internationalen Schluss-Entwurf bei IEC [Final Draft International Standard (FDIS)] sind jeweils nur „JA/NEIN“-Entscheidungen möglich, wobei „NEIN“-Entscheidungen fundiert begründet werden müssen. Dokumente, die bei CENELEC als Europäische Norm angenommen und ratifiziert werden, sind unverändert als Deutsche Normen zu übernehmen.

Für dieses Dokument ist das nationale Arbeitsgremium K 113 „Informationsstrukturen und Informations-elemente, Grundsätze der Identifikation und Kennzeichnung, Dokumentation und graphische Symbole“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE (www.dke.de) zuständig.

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf ein Dokument ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils aktuellste Ausgabe des in Bezug genommenen Dokuments.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe des Dokuments.

Der Zusammenhang der zitierten Dokumente mit den entsprechenden deutschen Dokumenten ergibt sich, soweit ein Zusammenhang besteht, grundsätzlich über die Nummer der entsprechenden IEC-Publikation. Beispiel: IEC 60068 ist als EN 60068 als Europäische Norm durch CENELEC übernommen und als DIN EN 60068 ins Deutsche Normenwerk aufgenommen.

Es wird auf die Möglichkeit hingewiesen, dass einige Elemente dieses Dokuments Patentrechte berühren können. IEC ist nicht dafür verantwortlich, einige oder alle diesbezüglichen Patentrechte zu identifizieren.

Das Original-Dokument enthält Bilder in Farbe, die in der Papierversion in einer Graustufen-Darstellung wiedergegeben werden. Elektronische Versionen dieses Dokuments enthalten die Bilder in der originalen Farbdarstellung.

**Standardisierte Übertragung und Registrierung von Ontologien für Produkte mittels
Datenpaketen –
Teil 8: Webservice-Schnittstelle für Datenpakete**

Inhalt	Seite
Einleitung	6
1 Anwendungsbereich	7
2 Normative Verweisungen	7
3 Begriffe und Abkürzungen	8
3.1 Begriffe	8
3.2 Abkürzungen	9
4 Nutzungsszenarien	10
4.1 Holistisches Nutzungsszenario	10
4.2 Anwendungsszenario zwischen Server und Client	11
4.3 Anwendungsszenario zwischen Servern	12
5 Paketwebdienst-Spezifikation	13
5.1 Allgemeine Angaben	13
5.2 Ausnahme	14
5.2.1 Allgemeine Angaben	14
5.2.2 Benennungskonvention für die Bezeichnung einer Ausnahme	14
5.2.3 Standarddefinierte Ausnahmen	15
5.3 Suchumfang	15
5.4 Paketregistrierungsdienst	17
5.4.1 Allgemeine Angaben	17
5.4.2 Anforderungsnachricht	17
5.4.3 Antwortnachricht	19
5.4.4 Ausnahme	20
5.5 Paketauflösungsdienst	20
5.5.1 Allgemeine Angaben	20
5.5.2 Anforderungsnachricht	21
5.5.3 Antwortnachricht	23
5.5.4 Ausnahme	24
5.6 Paketabonnementdienst	24
5.6.1 Allgemeine Angaben	24
5.6.2 Anforderungsnachricht	24
5.6.3 Allgemeine Angaben	24
5.6.3 Antwortnachricht	25
5.6.4 Ausnahme	25
5.6.5 Spezifikation einer Änderungsmitteilung	26

6	Spezifikation einer Paketdatendarstellung in einer Webdienstnachricht	26
6.1	Allgemeine Angaben	26
6.2	Grundlegende Datendarstellung	27
6.3	Reservierte Schlüsselwörter	27
6.3.1	Schlüsselwort zur Angabe von verbundenen Datenpaketen	27
6.4	Zusätzliche Anweisungen an Datenpakete für Paketwebdienste	28
6.4.1	Codierungsmodus	28
6.4.2	Vorgesehene Sprache	29
6.4.2	Standardwert	29
6.5	Beschreibung der Anweisungen	30
7	Datendarstellung in JSON	31
7.1	Basisstruktur der Datendarstellung in JSON	31
7.2	Reservierter JSON-Name zur Angabe eines Arrays von Datenpaketen	33
7.3	JSON-Namen für den Klassenkopfbereich	33
7.3.1	JSON-Name zur Angabe der Anweisung "#CLASS_ID"	33
7.3.2	JSON-Name zur Angabe der Anweisung "#PARCEL_MODE"	34
7.3.3	JSON-Name zur Angabe der Anweisung "#PARCEL_ID"	34
7.3.4	JSON-Name zur Angabe der Anweisung "#DEFAULT_SUPPLIER"	34
7.3.5	JSON-Name zur Angabe der Anweisung "#DEFAULT_VERSION"	34
7.3.6	JSON-Name zur Angabe der Anweisung „#OBJECT_ID_NAME“	34
7.3.7	JSON-Name zur Angabe der Anweisung „#ID_ENCODE“	34
7.3.8	JSON-Name zur Angabe der Anweisung „#PWS_CODIFICATION_MODE“	35
7.3.9	JSON-Name zur Angabe der Anweisung „#INTENDED_LANGUAGE“	35
7.4	JSON-Namen für den Schemakopfbereich	35
7.4.1	Basisstruktur der Datendarstellung für den Schemakopfbereich in JSON	35
7.4.2	JSON-Namen für den Schemakopfbereich	35
7.5	Datendarstellung für den Datenbereich in JSON	36
7.5.1	Vertikale JSON-Darstellung für den Datenbereich	36
7.5.2	Laterale JSON-Darstellung für den Datenbereich	37
7.6	Zeichencodierung	37
8	Datendarstellung in XML	37
8.1	Basisstruktur der Datendarstellung in XML	37
8.2	Reserviertes Schlüsselwort zur Angabe eines Datenpakets	38
8.3	XML-Elemente für den Klassenkopfbereich	38
8.3.1	XML-Element zur Angabe der Anweisung „#CLASS_ID“	38
8.3.2	XML-Element zur Angabe der Anweisung „#PARCEL_MODE“	38
8.3.3	XML-Element zur Angabe der Anweisung „#PARCEL_ID“	38
8.3.4	XML-Element zur Angabe der Anweisung „#DEFAULT_SUPPLIER“	39

8.3.5	XML-Element zur Angabe der Anweisung „#DEFAULT_VERSION“	39
8.3.6	XML-Element zur Angabe der Anweisung „#OBJECT_ID_NAME“	39
8.3.7	XML-Element zur Angabe der Anweisung „#ID_ENCODE“	39
8.3.8	XML-Element zur Angabe der Anweisung „#PWS_CODIFICATION_MODE“	39
8.3.9	XML-Element zur Angabe der Anweisung „#INTENDED_LANGUAGE“	39
8.4	XML-Elemente für den Schemakopfbereich	40
8.4.1	Basisstruktur der Datendarstellung für den Schemakopfbereich in XML	40
8.4.2	XML-Elemente des Schemakopfbereichs	40
8.5	XML-Elemente und Attribut für Datenbereich	41
8.5.1	Vertikale XML-Darstellung des Datenbereichs	41
	Laterale XML-Darstellung des Datenbereichs	42
8.6	Zeichencodierung	44
	Anhang A (normativ) Informationsobjektregistrierung	45
	Anhang B (normativ) Schema	46
B.1	JSON-Schema	46
B.1.1	Vertikales JSON-Schema	46
B.1.2	Laterales JSON-Schema	48
B.1.3	Ausnahme JSON-Schema	50
B.2	XML-Schema	51
B.2.1	Vertikales XML-Schema	51
B.2.2	Laterales XML-Schema	54
B.2.3	Ausnahme XML-Schema	57
	Anhang C (normativ) Webdienstdarstellung	58
C.1	Webdienstdarstellung in WADL	58
C.2	Webdienstdarstellung in WSDL	62
	Anhang D (informativ) Beispiele für Datendarstellung	66
D.1	Datenpaket als Beispiel	66
D.2	Beispiel der Datendarstellung in JSON-Notation	67
D.2.1	Beispiel der Datendarstellung in vertikaler JSON-Notation	67
D.2.2	Beispiel der Datendarstellung in lateraler JSON-Notation	68
D.3	Beispiel der Datendarstellung in XML-Notation	69
D.3.1	Beispiel der Datendarstellung in vertikaler XML-Notation	69
D.3.2	Beispiel der Datendarstellung in lateraler XML-Notation	71
	Anhang E (informativ) Beschreibungen der Anweisungen von „optional – informative“	73
	Literaturhinweise	74
Bilder		
	Bild 1 – Holistisches Anwendungsszenario von Paketwebdiensten	11
	Bild 2 – Paketauflösung und Registrierungsdienste zwischen einem Server und einem Client	12

Bild 3 – Paketabonnementdienst zwischen Verzeichnissen	13
Bild 4 – Baumstruktur von Ausnahmen	14
Bild 5 – Beispiel einer Strukturansicht der Verwendung von Suchumfangsergänzungen	16
Bild 6 – Beispiel einer Paketblattansicht der Verwendung von Suchumfangsergänzungen	17
Bild 7 – Übersicht des Paketauflösungsdienstes	21
Bild 8 – Grundstruktur einer Datendarstellung für einen verbundenen Satz von Datenpaketen	27
Bild 9 – Beispiel für die Verwendung von Standardwerten	30
Bild 10 – Basisstruktur der Datendarstellung in JSON	33
Bild 11 – Basisstruktur der Datendarstellung für den Schemakopfbereich in JSON	35
Bild 12 – Basisstruktur der Datendarstellung in XML	38
Bild 13 – Basisstruktur der Datendarstellung für den Schemakopfbereich in XML	40
Bild 14 – Struktur der Datendarstellung für den Datenbereich in der vertikalen XML-Notation	41
Bild 15 – Struktur der Datendarstellung für den Datenbereich in lateraler XML-Notation	43
Bild B.1 – Vertikales XML-Schema	46
Bild B.2 – Laterales JSON-Schema	48
Bild B.3 – Ausnahme JSON-Schema	50
Bild B.4 – Vertikales XML-Schema	51
Bild B.5 – Laterales XML-Schema	54
Bild B.5 – Laterales XML-Schema	56
Bild B.6 – Ausnahme XML-Schema	57
Bild C.1 – Webdienstdarstellung in WADL	58
Bild C.2 – Webdienstdarstellung in WSDL	62
Bild D.1 – Beispiel der Datendarstellung in vertikaler JSON-Notation	67
Bild D.2 – Beispiel der Datendarstellung in lateraler JSON-Notation	68
Bild D.3 – Beispiel der Datendarstellung in vertikaler XML-Notation	69
Bild D.4 – Beispiel der Datendarstellung in lateraler XML-Notation	71

Tabellen

Tabelle 1 – Standarddefinierte Ausnahmen für Paketwebdienste	15
Tabelle 2 – Spezifikation von Suchumfangsergänzungen	16
Tabelle 3 – Die Struktur einer Anforderungsnachricht des Paketregistrierungsdienstes	18
Tabelle 4 – Die Struktur einer Antwortnachricht des Paketregistrierungsdienstes	20
Tabelle 5 – Die Struktur einer Anforderungsnachricht des Paketauflösungsdienstes	22
Tabelle 6 – Die Struktur einer Antwortnachricht des Paketauflösungsdienstes	24
Tabelle 7 – Die Struktur einer Anforderungsnachricht des Paketabonnementdienstes	25
Tabelle 8 – Die Struktur einer Antwortnachricht des Paketabonnementdienstes	25
Tabelle 9 – Die Spezifikation einer Mitteilung	26
Tabelle 10 – Beschreibung der in IEC 62656-1 spezifizierten Anweisungen	31
Tabelle 11 – Beschreibung der in diesem Dokument spezifizierten Anweisungen	31

Tabelle D.1 – Datenpaket als Beispiel	66
Tabelle E.1 – Beschreibungen der Anweisungen von „optional – informative“	73

Einleitung

Für die Beschreibung von Produkten und Dienstleistungen während ihres Lebenszyklus ist eine erweiterte Dateninteroperabilität mit reduzierten menschlichen Eingriffen ein endgültiges Ziel bei der Entwicklung internationaler Normen für intelligente Produktionssysteme. Für die Erreichung dieses Ziels wird einer industriellen Ontologie eine bedeutende Rolle zugeschrieben, da sie Komponenten von Systemen erlaubt, miteinander zu kommunizieren, nämlich durch ein gegenseitiges Verständnis von Maschine zu Maschine über Funktionen, Fähigkeiten, Strukturen und Konfigurationen.

Das in IEC 62656-1 definierte parzellierte Ontologiemodell, auch unter der Abkürzung „POM“ bekannt, ist ein generisches Ontologiemodell mit vierfachen Schichten zur Erfassung unterschiedlicher Ontologiemodelltypen durch Sortierung von Dingen in Kategorien einer homogenen Sammlung ontologischer Entitäten wie Klassen (Konzepte), Eigenschaften, Beziehungen, Aufzählungen, Ausdrücke (Konstanten), Datentypen, usw. Auf der zweiten Schicht von oben mit der Bezeichnung Metaontologieschicht (MO) sind elf Typen von Kategorien definiert. Jede Schicht ist eine Sammlung von Kategorien, während jede Kategorie durch eine tabellenähnliche Matrix namens „Datenpaket“ dargestellt wird, von der Metadaten (Attribute) als eine Auswahl von Instanzen der unmittelbaren höheren Schicht realisiert sind. Die oberste Schicht des POM mit der Bezeichnung Axiomatische Ontologieschicht (AO) umfasst lediglich zwei Datenpakete, die zusammen das „Konzept der Konzepte“ nach Klassen und Eigenschaften definieren, wobei es sich um eine Informationstechnologie (IT)-Darstellung des mathematisch-logischen Begriffs der Klasse (d. h. des „Konzepts“) selbst handelt.

Nachfolgende Teile der Normen IEC 62656, die in ihrer Gesamtheit als „Paketnormen“ bekannt sind, dienen der Spezialisierung des POM für einen bestimmten Zweck.

IEC 62656-2 ist ein Leitfaden für Fachexperten zur Anwendung des POM zur Aufzeichnung eines Datenwörterbuchs aus in Produktnormen verfügbaren Definitionen, in einer dem Wörterbuchschemata von IEC 61360-2/ISO 13584-42 entsprechenden Form (d. h., allgemeines Datenwörterbuchmodell (en: common data dictionary model, CDDM)) und unter Verwendung der Spezifikation des Teils als offizieller Datenschnittstelle für IEC 61360-4 DB, bekannt als IEC CDD (Common Data Dictionary), durch Ermöglichung des Uploads und Downloads des Wörterbuchs in das und aus dem IEC CDD. Eine Referenzimplementierung dieses Teils ist verfügbar als kostenloses Tool für Standardisierungszwecke.

IEC 62656-3 ist vorgesehen als Abbildungsspezifikation zwischen einem Standarddatenmodell der „Smart-Grid“-Domäne mit der Abkürzung CIM (Common Information Model) und einem erweiterten oder eher verallgemeinerten Datenmodell des IEC CDD, nämlich dem POM. Das CIM umfasst die Normenreihe IEC 61968/61970/62325. Das IEC CDD kann somit das CIM aufnehmen, sofern das IEC CDD das POM hinreichend als Datenschnittstelle oder Datenbank umsetzt. Alternativ bringt diese Abbildung zwangsläufig eine kleine aber wesentliche Erweiterung des IEC CDD mit sich, ohne die das CIM nicht im IEC CDD untergebracht werden kann. Dennoch gibt es nichts zum Tool hinzuzufügen oder davon abzuziehen, das aktuell als Datenschnittstelle für das IEC CDD genutzt wird und die IEC 62656-1 vollständig darstellt.

IEC 62656-5 dient als Schnittstelle zur Beschreibung von Aktivitäten als eine Ontologie gemäß IEC 62656-1 und ermöglicht damit das Speichern von Definitionen, die in tätigkeitszentrierten internationalen Normen wie IEC 62224-3 als Ontologie verfügbar sind. Dieser Teil kann auch auf die Beschreibung von nicht fertigungsbezogenen Nutzungsszenarien angewendet werden, wie für die Beschreibung von Aktivitäten des Naturgefahrenmanagements oder der elektronischen Touristenführung oder Navigation, mit einer harmonischen Integration von Aktivitäten mit verwandten Produkten und Dienstleistungen.

Dies bedeutet, dass ein gemeinsames Ontologie-Repository (en: common ontology repository oder kurz „COR“), das auf dem POM basiert, sowohl das IEC CDD als auch CIM-Typen von Datenwörterbüchern oder Ontologien speichern kann. Des Weiteren kann es reibungslos die Unterschiede überbrücken und die Lücken auffüllen und so Ontologien unterschiedlicher Herkunft abdecken.

Von den übrigen Teilen der Reihe IEC 62656 wird erwartet, dass sie Licht auf ein neues Spektrum von Anwendungen für das COR auf Basis des POM werfen.

Bei allen ist IEC 62656-8 für die Beschreibung grundlegender Webdienste für auf dem POM basierende semantische Repositories vorgesehen, während ein erweiterter Typ von Webschnittstelle einschließlich komplexer Produkthanfrage sowie Abfrageweiterleitung an ein anderes Repository einem zukünftig zu entwickelnden Teil der Reihe vorbehalten bleibt.

1 Anwendungsbereich

Dieses Dokument legt eine auf JSON [1] und XML [2] basierende Webdienstschnittstelle zum Transport eines ontologischen Datensatzes gemäß IEC 62656-1 über das Internet oder ein Extranet-Netzwerk fest. Dieser Webdienst erlaubt einen Austausch von Datenpaketen zwischen einem Paket-Server und einem Paketclient oder zwischen Paket-Servern. Die dabei verwendeten Datenpakete umfassen eine der Paketontologieschichten, d. h. Axiomatische Ontologieschicht (AO), Metaontologieschicht (MO), Domänenontologieschicht (DO) oder Domänenbibliotheksschicht (en: Domain Library layer (DL)), deren Details in IEC 62656-1 definiert sind. Diese Schnittstelle umfasst drei Basisdienste, Registrierungsdienst, Auflösungsdienst und Subskriptionsdienst. Hinsichtlich Abfragen wird die Abfragesprache für ein Datenpaket in einem anderen Teil von IEC 62656 behandelt.

Dementsprechend gehören zur in diesem Dokument festgelegten Webdienstschnittstelle folgende Punkte:

- detaillierte Spezifikation der drei Grunddienste,
- Transportmethoden von Datenpaketen für die Ontologiedatenkommunikation.

Die folgenden Punkte gehören nicht zum Anwendungsbereich dieses Dokuments:

- Benutzeridentifizierung und -autorisierung,
- Abfragesprache,
- Daten- und Kommunikationssicherheitstechniken.

2 Normative Verweisungen

Die folgenden Dokumente werden im Text in solcher Weise in Bezug genommen, dass einige Teile davon oder ihr gesamter Inhalt Anforderungen des vorliegenden Dokuments darstellen. Bei datierten Verweisungen gilt nur die in Bezug genommene Ausgabe. Bei undatierten Verweisungen gilt die letzte Ausgabe des in Bezug genommenen Dokuments (einschließlich aller Änderungen).

ISO 639-1:2002, *Codes for the representation of names of languages – Part 1: Alpha-2 code*

ISO 3166-1:2013, *Codes for the representation of names of countries and their subdivisions – Part 1: Country codes*

ISO 13584-24:2003, *Industrial automation systems and integration – Parts library – Part 24: Logical resource: Logical model of supplier library*

ISO 13584-32:2010, *Industrial automation systems and integration – Parts library – Part 32: Implementation resources: OntoML: Product ontology markup language*

ISO 13584-42:2010, *Industrial automation systems and integration – Parts library – Part 42: Description methodology: Methodology for structuring parts families*

ISO/TS 29002-20:2010, *Industrial automation systems and integration – Exchange of characteristic data – Part 20: Concept dictionary resolution services*

IEC 61360-1:2017(to be), *Standard data element types with associated classification scheme for electric components – Part 1: Definitions – Principles and methods*

IEC 61360-2:2012, *Standard data element types with associated classification scheme for electric components – Part 2: EXPRESS dictionary schema*

IEC 62656-1:2014, *Standardized product ontology register and transfer by spreadsheets – Part 1: Logical structure for data parcels*

IEC 62656-2:2013, *Standardized product ontology register and transfer by spreadsheets – Part 2: Application guide for use with the IEC common data dictionary (CDD)*

ISO/IEC 21778 (to be), *The JSON data interchange format*

3 Begriffe und Abkürzungen

3.1 Begriffe

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe.

3.1.1

Anwendungsontologie-Repository

AOR

Ontologie-Repository, in dem proprietäre Ontologien und ihre Instanzen gespeichert werden können

Anmerkung 1 zum Begriff: Eine proprietäre Ontologie kann eine Veränderung und/oder eine Erweiterung einer standardisierten Ontologie sein.

Anmerkung 2 zum Begriff: Standardontologien können ebenfalls im AOR gespeichert werden.

3.1.2

gemeinsames Ontologie-Repository

COR

gemeinsam genutztes Ontologie-Repository, in dem standardisierte Ontologien unterschiedlicher Basismodelle gespeichert werden können

3.1.3

verbundene Datenpakete

Pakete von Tabellenblättern, die gemeinschaftlich benutzt werden, um eine Bibliothek, ein Referenzlexikon oder ein Meta-Referenzlexikon zu definieren

[QUELLE: IEC 62656-1:2014, 3.6]

3.1.4

Datenpaket

Paket

Informationsstruktur in Form von zwei zusammengehörenden Ebenen, die einen Satz Merkmale und einen Satz von Tupeln von Werten für die Sätze der Merkmale umfasst, mit dem Ziel, ein bereichsbezogenes Referenzlexikon, eine bereichsbezogene Bibliothek oder ein ontologisches Modell zu beschreiben

Anmerkung 1 zum Begriff: Ein Datenpaket wird üblicherweise als Satz von Arbeitsblättern implementiert und ausgetauscht, aber das Implementierungs- oder Austauschmedium ist nicht auf Arbeitsblätter begrenzt, es kann eine beliebige andere Form annehmen.

[QUELLE: IEC 62656-1:2014, 3.8]

3.1.5

Wörterbuch

Datenwörterbuch

Menge von Begriffen mit entsprechenden Identifikationen, formuliert in einer kanonischen Syntax und mit allgemein akzeptierten Definitionen, dazu bestimmt, ein lexikalisches oder taxonomisches Rahmenwerk zur Repräsentation von Wissen in einer von Computern verarbeitbaren Form zu errichten, die von verschiedenen informationsverarbeitenden Systemen und Gemeinschaften benutzt werden kann

[QUELLE: IEC 62656-1:2014, 3.10]

3.1.6

JSON-Daten

Daten, die in Übereinstimmung mit der Spezifikation von ISO/IEC 21778 dargestellt sind

3.1.7

JSON-Name

Serie von Zeichen, die einem Wert oder Objekt für die Verweisung darauf in JSON-Daten zugewiesen sind

3.1.8

ontologische Entität

Artefakt zur Darstellung einer Kategorie von Dingen oder der Beziehung zwischen ihnen

[QUELLE: IEC 62656-1:2014, 3.36]

3.1.9

Ontologie-Repository

Datenspeicher für Ontologien oder ontologische Entitäten

3.1.10

Paketclient

Client-System oder sonstige Anwendung, die Tabellenblätter aus Datenpaketen lesen oder schreiben und die wahlweise eine Fähigkeit besitzen kann, die Tabellenblätter an ein Serversystem zu senden oder von dort zu empfangen

[QUELLE: IEC 62656-1:2014, 3.37]

3.1.11

Paketontologieschicht

Abstraktionsschicht, die als Satz von Datenpaketen auf der gleichen Ebene ausgestaltet ist

Anmerkung 1 zum Begriff: IEC 62656-1:2014 klassifiziert Paketontologieschichten als axiomatische Ontologie (AO), Metaontologie (MO), Domänenontologie (DO) und Domänenbibliothek (DL).

3.1.12

Paket-Server

Serversystem oder sonstige Anwendung, die Tabellen bereitstellen kann, im Allgemeinen über das Internet

[ERGÄNZT: IEC 62656-1:2014, 3.41]

3.1.13

Paketregistrierung

Vorgang zur Eingabe einer neuen Informationsaufzeichnung durch Datenpakete auf einem Paket-Server

Anmerkung 1 zum Begriff: Registrierungsverfahren sind klassifiziert als Vorgänge zur Hinzufügung, Änderung und Löschung.

3.1.14

Paketauflösung

Vorgang zum Erhalt von Informationen aus Datenpaketen durch Verwendung des/der spezifischen Parameter(s) als Suchbedingung(en)

3.1.15

Paketabonnement

Vereinbarung zum Erhalt des Status oder der Änderung der Information durch Datenpakete

3.2 Abkürzungen

AOR Anwendungsontologie-Repository

CDD Gemeinsames Referenzlexikon (en: Common Data Dictionary)

COR	Gemeinsames Ontologie-Repository (en: Common Ontology Repository)
ICID	Internationaler Konzept-Identifikator (International Concept Identifier)
JSON	JavaScript Object Notation
RAI	Bezeichner der Registrierungsstelle (en: Registration Authority Identifier)
SOAP	Einfaches Objektzugriffsprotokoll (Simple Object Access Protocol)
VI	Versionsbezeichner (Version Identifier)
XML	erweiterbare Auszeichnungssprache (en: eXtensible Markup Language)
WADL	Webanwendungs-Beschreibungssprache (Web Application Description Language)
WSDL	Webdienst-Beschreibungssprache (Web Services Description Language)

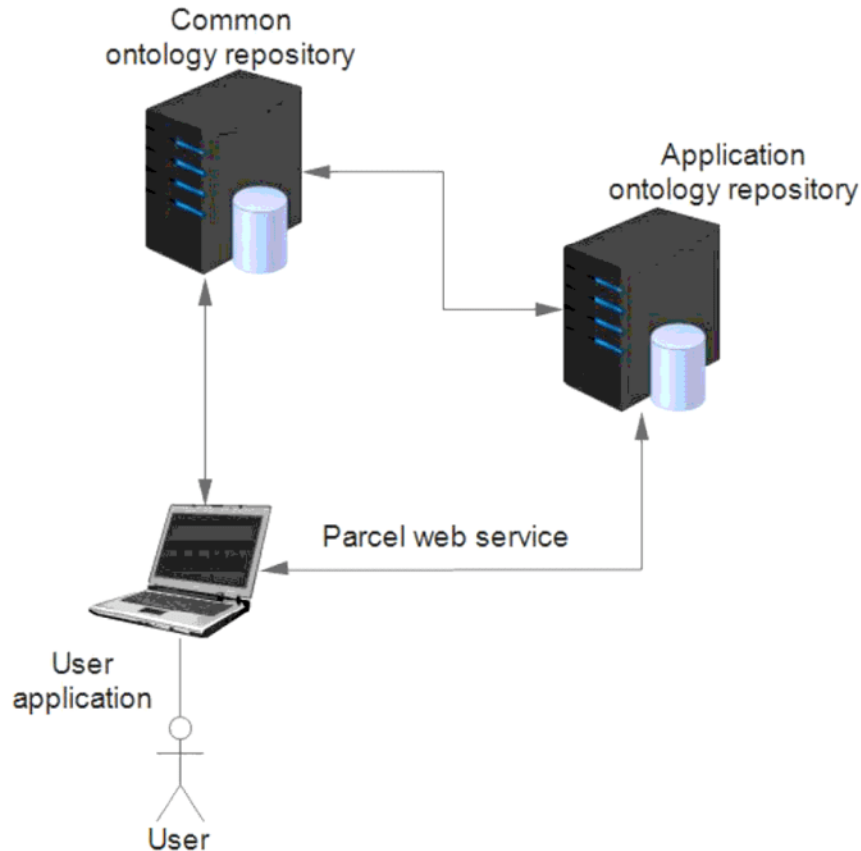
4 Nutzungsszenarien

4.1 Holistisches Nutzungsszenario

Eine Standardontologie wie IEC 61360-4 besteht aus einem standardisierten Satz von Klassen und Eigenschaften, der für ein gemeinsames Verständnis der Bedeutung von Daten unter Anwendern verwendet werden sollte. Im Allgemeinen enthält eine solche Standardontologie einen vereinbarten Mindestsatz von Klassen und Eigenschaften mit offiziellen Sprachen, weshalb er in der Regel für die tatsächliche Nutzung modifiziert oder erweitert werden muss. Eine Sprachübersetzung ist beispielsweise wichtig zur Verwendung einer Ontologie für nicht einheimische Anwender in einem regionalen Unternehmen. Oder, um als weiteres Beispiel anzuführen, eine Erweiterung von Klassen von Eigenschaften wie einem Preis für einen Kunden und eine Art Verwaltungsinformation sind wesentlich für die Beschreibung von Produktdaten in jedem Unternehmen. Die vier Modellierungsschichten, ein in [IEC 62656-1](#) definierter Ansatz, ermöglichen die Darstellung einer Erweiterung und Änderung in Datenpaketen.

Eine solche individuell angepasste Ontologie wird im Allgemeinen auf einem Server unterhalten (z. B. Anwendungsontologie-Repository) wie etwa einem nationalen und Unternehmensserver, getrennt von einem Server (z. B. gemeinsames Ontologie-Repository), auf dem eine Standardontologie gespeichert ist. Wenn es einen Mechanismus nicht nur für das Hochladen und Herunterladen von Datenpaketen, sondern auch für das Abonnement einer Ontologie mithilfe der Webtechnik gibt, dann wird die Verwendung einer Standardontologie wachsen und ihre Erweiterung einfacher. Die im vorliegenden Dokument genannten Webdienste werden eingebracht, um den Austausch von Datenpaketen einer Ontologie zwischen einem Server und einem Client oder zwischen Servern zu ermöglichen.

Die in diesem Dokument definierten Webdienste für Datenpakete gehen von den folgenden in [Bild 1](#) beschriebenen Szenarien aus, ohne sich jedoch darauf zu beschränken.



Text im Bild:

Englisch

Common ontology repository
Application ontology repository
Parcel web service
User application

Deutsch

Gemeinsames Ontologie-Repository
Anwendungsontologie-Repository
Paketwebdienst
Benutzeranwendung

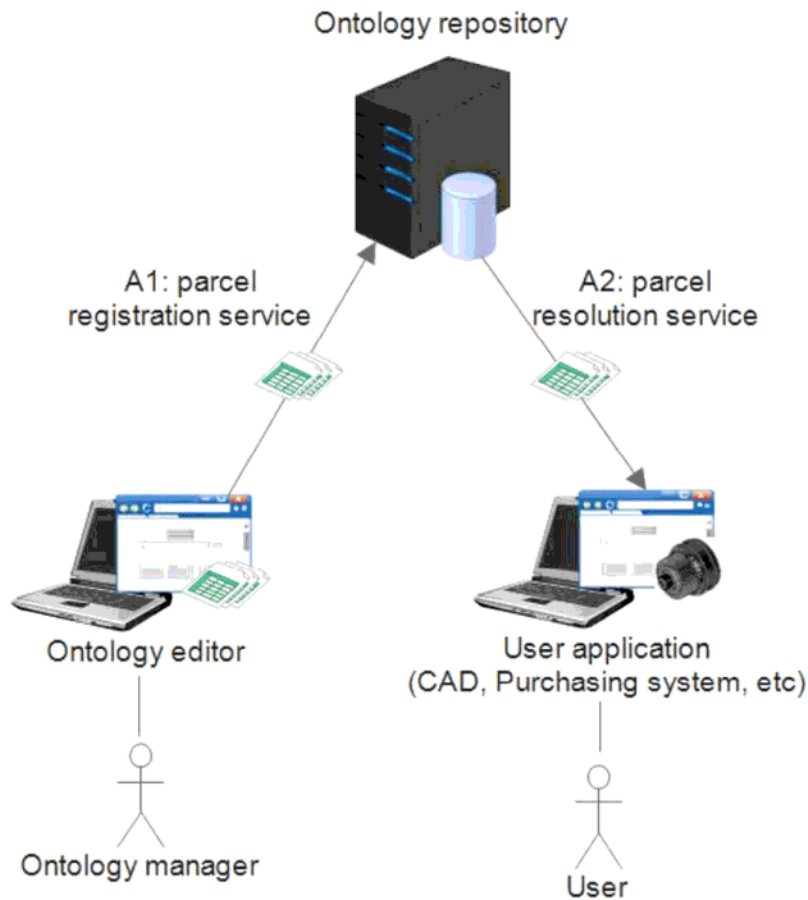
Bild 1 – Holistisches Anwendungsszenario von Paketwebdiensten

4.2 Anwendungsszenario zwischen Server und Client

Ein detailliertes Anwendungsszenario zwischen einem Ontologie-Repository und einem Paket-Client ist in [Bild 2](#) mit den folgenden Aktivitäten und zugehörigen Informationsflüssen beschrieben:

- A1: Eine Client-Anwendung wie ein Ontologie-Editor registriert den Inhalt eines Satzes von Datenpaketen in ein Ontologie-Repository ohne einen Betriebsvorgang auf der Webseite des Ontologie-Repositorys zu durchlaufen.
- A2: Eine Client-Anwendung wie ein Controller, eine CAD-Anwendung oder ein Einkaufssystem fragen eine Auflösung des Bezeichners eines Wörterbuchelements an und rufen einen Informationssatz über ein Wörterbuchelement teilweise oder vollständig ab, das mit einem Bezeichner versehen ist, ohne einen Betriebsvorgang auf der Webseite eines Ontologie-Repositorys zu durchlaufen.

ANMERKUNG Von einem Satz von Datenpaketen bereitgestellter Inhalt stellt ein Datenwörterbuch oder eine Aktualisierung eines Datenwörterbuchs dar.



Text im Bild:

Englisch

Ontology repository
A1: parcel registration service
A2: parcel resolution service
Ontology editor
User application (CAD, Purchasing system, etc)
Ontology manager
User

Deutsch

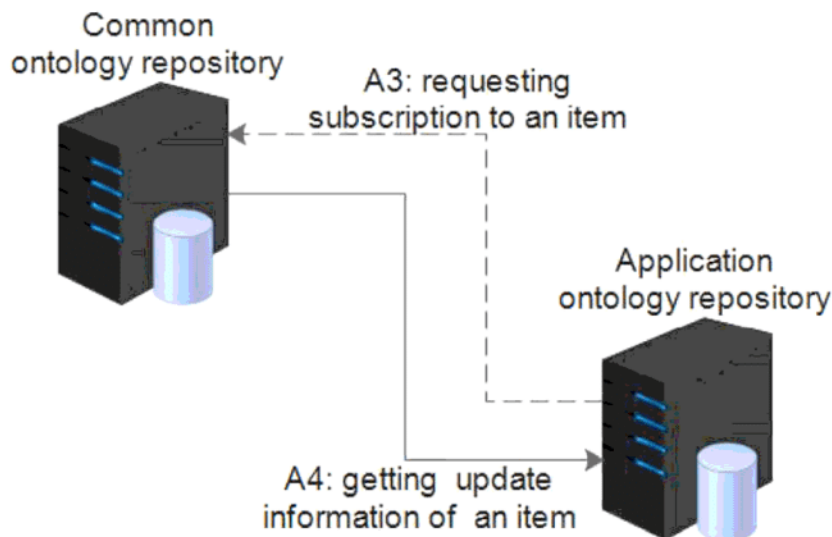
Ontologie-Repository
A1: Paketregistrierungsdienst
A2: Paketauflösungsdienst
Ontologie-Editor
Benutzeranwendung (CAD, Einkaufssystem, usw.)
Ontologie-Manager
Benutzer

Bild 2 – Paketauflösung und Registrierungsdienste zwischen einem Server und einem Client

4.3 Anwendungsszenario zwischen Servern

Ein detailliertes Anwendungsszenario zwischen Servern ist in [Bild 3](#) mit den folgenden Aktivitäten und zugehörigen Informationsflüssen beschrieben:

- A3: Ein Anwendungsontologie-Repository fordert das Abonnement eines Wörterbuchelements an, das in einem gemeinsamen Ontologie-Repository oder einem anderen Anwendungsontologie-Repository gespeichert ist, um eine Aktualisierung des im letzteren gespeicherten Inhalts zu erhalten, wenn eine neue Instanz darin registriert wird.
- A4: Ein Anwendungsontologie-Repository, das ein im gemeinsamen Ontologie-Repository gespeichertes Wörterbuchelement abonniert, erhält eine Aktualisierung des Inhalts auf regelmäßiger definierter Zeitspannenbasis oder auf ereignisgesteuerter Basis, d. h., wenn eine neue Instanz im zuletzt genannten Repository registriert wird. Ontologie-Repository-Server können eine Schalkaskade erstellen, um den Inhalt von einem vorgeschalteten Server an einen nachgeschalteten zu liefern.



Text im Bild:

Englisch

Common ontology repository
A3: requesting subscription to an item
Application ontology repository
A4: getting update information of an item

Deutsch

Gemeinsames Ontologie-Repository
A3: Abonnementanforderung für ein Element
Anwendungsontologie-Repository
A4: Erhalt von aktualisierten Informationen eines Elements

Bild 3 – Paketabonnementdienst zwischen Verzeichnissen

5 Paketwebdienst-Spezifikation

5.1 Allgemeine Angaben

Dieses Dokument legt die folgenden drei grundlegenden Webdienste fest, auf deren Grundlage komplexere oder erweiterte Benutzeranwendungen aufgebaut werden können:

- Paketregistrierungsdienst (siehe 0),
- Paketauflösungsdienst (siehe 5.5),
- Paketabonnementdienst (siehe 5.6).

Wie im Allgemeinen auch andere Webdienste hat jeder in diesem Dokument definierte Basis-Webdienst Behandlungen für Anforderungen, Antworten und Ausnahmen.

Eine Anforderung ist eine Nachricht, die von einem Paketclient an einen Paket-Server gesendet wird. Die Nachricht kann einen oder mehrere Parameter enthalten, die jeweils zur Ausführung eines Dienstes auf einem Paket-Server dienen.

Eine Antwort ist eine Nachricht, die von einem Paket-Server an einen Paketclient als Ergebnis der Ausführung eines Dienstes gesendet wird. Die Nachricht kann das Ergebnis der Anfrage von einem Paketclient enthalten.

Eine Ausnahme ist ein unerwartetes Ereignis, das bei der Ausführung eines Paketwebdienstes auftritt. Wenn ein solches Ereignis auf einem Paket-Server auftritt und wenn Informationen über das Ereignis hilfreich oder bedeutsam für einen Paketclient sind, gibt der Paket-Server eine Ausnahmemeldung an den Paket-Server aus. Auf diese Weise kann ein Paketclient seinen Funktionsfluss fortsetzen, wenn ein Dienst ausfällt. Wenn bei der Verarbeitung einer Funktion, deren Ausführung ein Paketclient anfordert, eine Ausnahme auftritt, wird die Ausführung beendet, und der Paket-Server gibt die Ausnahme an den Paketclient aus.

Als maschinenlesbare Beschreibung von Webdiensten sind WADL (Web Application Description Language)[3] und WSDL (Web Service Description Language)[4] weit verbreitet. Eine WADL-Beschreibung für die in 5 definierten Dienste wird in C.1 bereitgestellt und eine WSDL-Beschreibung dafür in C.2.

5.2 Ausnahme

5.2.1 Allgemeine Angaben

Wie in 5.1 erwähnt, ist eine Ausnahme ein unerwartetes Ereignis, das bei der Ausführung eines Paketwebdienstes auftritt.

Das vorliegende Dokument spezifiziert die folgenden Ausnahmetypen für ein unerwartetes Ereignis.

- Standarddefinierte Ausnahme
- Benutzerdefinierte Ausnahme

„Standarddefinierte Ausnahme“ ist der Supertyp von Ausnahmen, die in diesen Dokumenten definiert sind. Die Spezifikation dieser Ausnahmen findet sich in 5.2.3.

„Benutzerdefinierte Ausnahme“ ist der Supertyp von Ausnahmen zur Festlegung von Ausnahmen, die in Benutzeranwendungen für einen bestimmten Zweck definiert sind. Für einen Paketwebdienst zwischen Paket-Servern oder zwischen einem Paket-Server und einem Paketclient ist die Verwendung einer benutzerdefinierten Ausnahme zulässig, sie darf jedoch nicht von einem Empfänger einer solchen Ausnahme behandelt werden. In anderen Worten, eine solche Ausnahme darf von einem Empfänger nur als Textinformation erkannt werden.

Bild 4 zeigt eine Ausnahmenstruktur, die aus den allgemeinen Ausnahmen und in Tabelle 1 definierten Ausnahmen besteht.

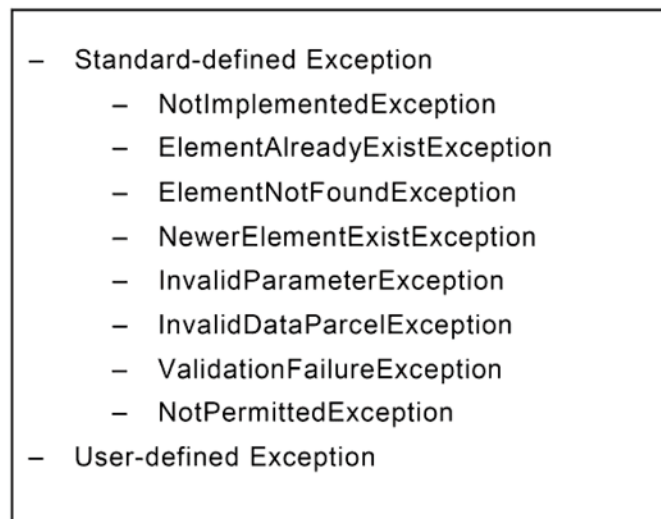


Bild 4 – Baumstruktur von Ausnahmen

5.2.2 Benennungskonvention für die Bezeichnung einer Ausnahme

Zur robusten Implementierung einer benutzerdefinierten Ausnahme sowohl auf einem Paket-Server als auch auf einem Paketclient muss festgelegt werden, welche Zeichen zur Benennung einer Ausnahme verwendet werden können. Als Benennungskonventionen müssen die folgenden Regeln zur Bezeichnung benutzerdefinierter Ausnahmen gelten.

- Jeder Wert darf nur alphabetische oder numerische Zeichen enthalten.
- Jeder Wert muss mit einem Großbuchstaben beginnen.
- Jeder Wert muss mit „Exception“ enden.

5.2.3 Standarddefinierte Ausnahmen

Standarddefinierte Ausnahmen, die von einem Paket-Server an einen Paketclient über einen Paketwebdienst ausgegeben oder zurückgegeben werden können, sind in Tabelle 1 beschrieben.

Name und Definition jeder Ausnahme werden zum menschlichen Verständnis angegeben. Das bevorzugte Buchstabensymbol jeder Ausnahme, das der in 5.2.1 definierten Benennungskonvention folgt, wird für eine Implementierung angegeben, die als Kennzeichnung der Ausnahme in einer Programmiersprache verwendet werden kann.

Tabelle 1 – Standarddefinierte Ausnahmen für Paketwebdienste

Bevorzugter Name	Definition	Bevorzugtes Buchstabensymbol
no implementation	eine die definierte Funktion angegebende standarddefinierte Ausnahme ist auf dem Ontologie-Repository-Server nicht implementiert	NotImplementedException
element already existing	eine die definierte Funktion angegebende standarddefinierte Ausnahme wird nicht ausgeführt, weil das gleiche Element bereits im Ontologie-Repository vorhanden ist	ElementAlreadyExistException
no element found	eine die definierte Anforderung angegebende standarddefinierte Ausnahme passt zu keinem Element im Ontologie-Repository	ElementNotFoundException
newer element existing	eine eine neuere Version eines Elements angegebende standarddefinierte Ausnahme findet sich im Ontologie-Repository	NewerElementExistException
invalid parameter	eine einen unerwarteten Parameter angegebende standarddefinierte Ausnahme ist in der Anforderungsnachricht spezifiziert	InvalidParameterException
unexpected value	eine einen unerwarteten Wert angegebende standarddefinierte Ausnahme wird in einen Parameter der Anforderungsnachricht eingegeben	InvalidDataParcelException
validation failure	eine einen oder mehrere Fehler angegebende standarddefinierte Ausnahme tritt während eines Validierungsprozesses in den spezifizierten Datenpaketen auf	ValidationFailureException
no permission	eine die Ausführung der spezifizierten Funktion angegebende standarddefinierte Ausnahme ist für den Anwender nicht zulässig	NotPermittedException

5.3 Suchumfang

Nach dem Prinzip des in 5.5 definierten Paketauflösungsdienstes wird, wenn es eine Zeile gibt, die einem für den Parameter „keyword“ einer Anforderungsnachricht spezifizierten Bezeichner oder Namen entspricht, diese Zeile zurückgegeben. Wenn beispielsweise der Bezeichner einer Klasse in einer Anforderung angegeben ist, dann wird die Definitionszeile dieser Klasse zurückgegeben.

Ähnlich wie beim Paketauflösungsdienst wird nach dem Prinzip des in 5.6 definierten Paketauflösungsdienstes, wenn es ein Element gibt, das einem für den Parameter „identifiers“ einer Anforderungsnachricht spezifizierten Bezeichner entspricht, dieses Element abonniert oder abgemeldet. Wenn beispielsweise der Bezeichner einer Klasse in einer Anforderung angegeben ist, dann wird die Klasse abonniert oder abgemeldet. Wenn eine Unterklasse neu zu einer Klasse hinzugefügt wird, d. h. die Verzweigungsstruktur geändert wird, wird die Klasse niemals verändert, weil die Informationen der „ist-ein“-Beziehung im Allgemeinen auf einer Unterklassenseite durch Verwendung des Attributs MDC_P010 (Superklasse) beschrieben wird. Dies führt dazu, dass die Änderung möglicherweise nicht an Teilnehmer der Klasse kommuniziert wird.

Um gleichzeitig zu abonnieren bzw. sich abzumelden oder sowohl Informationen einer Klasse als auch Informationen ihrer übergeordneten Klassen oder unteren Klassen durch eine Anforderung zu erhalten, defi-

niert dieses Dokument eine Suchumfangsergänzung für den Anforderungsparameter „Schlüsselwort“ des Paketauflösungsdienstes. Tabelle 2 zeigt die Spezifikation dieser Ergänzungen.

Tabelle 2 – Spezifikation von Suchumfangsergänzungen

Suchumfangsergänzung	Beschreibung	Beispiel
*	Diese Ergänzung erweitert den Umfang auf alle Unterklassen einer bestimmten Klasse, einschließlich von Klassen, die daraus Merkmale importieren.	AAA004* 0112/2///61360_4#AAA004##001*
\$	Diese Ergänzung erweitert den Umfang auf alle direkten Unterklassen. Der Beziehungsfall wird nicht berücksichtigt.	AAA004\$ 0112/2///61360_4#AAA004##001\$
%	Diese Ergänzung erweitert den Umfang auf alle Superklassen, einschließlich von Klassen, aus denen Merkmale importiert werden.	AAA004% 0112/2///61360_4#AAA004##001%
!	Diese Ergänzung erweitert den Umfang nur auf alle direkten Superklassen.	AAA004! 0112/2///61360_4#AAA004##001!

Bild 5 zeigt, wie jede Suchumfangsergänzung die Erweiterung eines Suchumfangs anstrebt, indem als Beispiel ein Teil des Klassenbaums von IEC 61360-4 mit Stand vom 10.06.2018 verwendet wird. Wenn AAA004 des Parameters „keyword“ in einer Anforderungsnachricht eines Paketauflösungsdienstes definiert ist, dann wird die zu AAA004 passende Zeile zurückgegeben. Wenn AAA004% oder AAA004! des Parameters „keyword“ definiert sind, dann werden zu AAA004 passende Zeilen oder Bezeichner der entsprechenden Superklasse zurückgegeben, d. h., AAA001, AAA002 und AAA003. Wenn AAA004* oder AAA004\$ des Parameters „keyword“ definiert sind, dann werden zu AAA004 passende Zeilen oder Bezeichner der entsprechenden Superklasse zurückgegeben, d. h., AAA005, AAA006, AAA007, AAA008, AAA009 und AAA010.

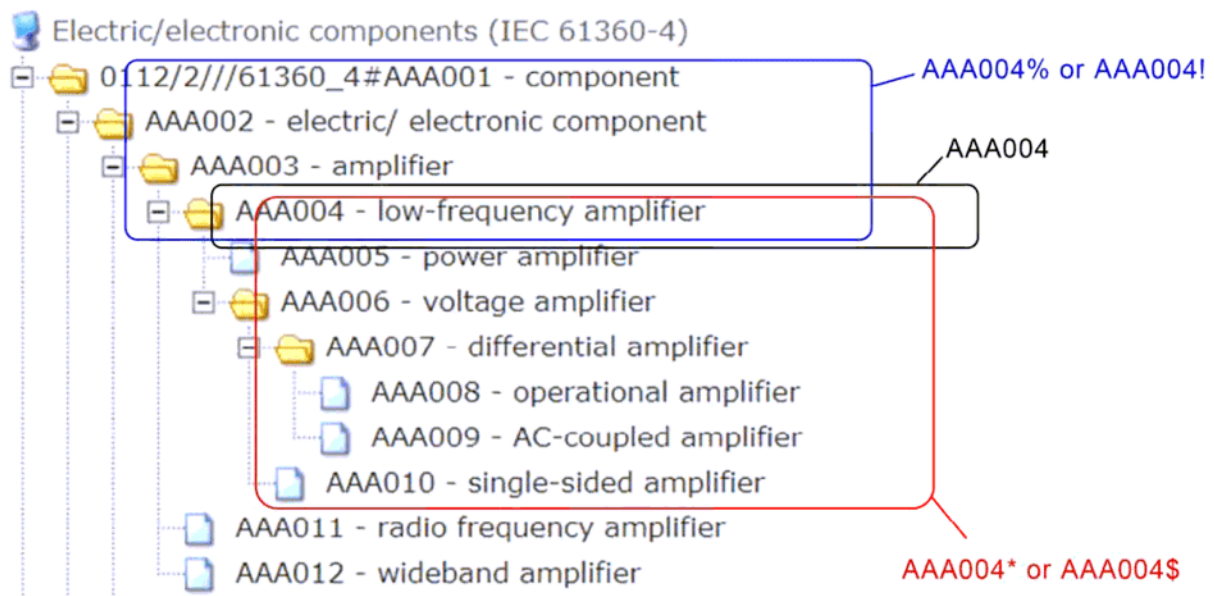


Bild 5 – Beispiel einer Strukturansicht der Verwendung von Suchumfangsergänzungen

Bild 6 zeigt ein Beispiel eines Klassenblatts des in Bild 5 beschriebenen Klassenbaums. Jedes ausgefüllte Feld mit einer Kennzeichnung eines Bezeichners mit/ohne einen Suchumfang gibt an, welche Zeilen zurückgegeben werden, wenn der Bezeichner mit/ohne die Suchumfangsergänzung beim Parameter „Schlüsselwort“ in einer Anforderungsnachricht eines Paketauflösungsdienstes angegeben wird.

#CLASS_ID:=MDC_C002			
#CLASS_NAME.en:=Class meta-class			
#PROPERTY_ID	MDC_P001_5	MDC_P010	MDC_P014
#PROPERTY_NAME.en	Code	Superclass	Applicable properties
	AAA001	UNIVERSE	{AAE001, AAE022, ...}
	AAA002	AAA001	{AAE002, AAE007, ...}
	AAA003	AAA002	{AAE697, AAE969, ...}
	AAA004	AAA003	{AAF169}
	AAA005	AAA004	
	AAA006	AAA004	{AAF158, AAF159, ...}
	AAA007	AAA006	{AAF157, AAF160, ...}
	AAA008	AAA007	{AAF152, AAF153, ...}
	AAA009	AAA007	
	AAA010	AAA006	
	AAA011	AAA003	
	AAA012	AAA003	{AA424, AAE698, ...}

AAA004% or AAA004!

AAA004

AAA004* or AAA004\$

Bild 6 – Beispiel einer Paketblattansicht der Verwendung von Suchumfangsergänzungen

5.4 Paketregistrierungsdienst

5.4.1 Allgemeine Angaben

Der Paketregistrierungsdienst ist ein Service, der einem Paketclient ermöglicht, Artikel in Datenpaketen auf einen Paket-Server zu registrieren. Wenn ein Paket-Server eine Anforderung von einem Paketclient erhält, der eine Registrierungsbefugnis hat, prüft der Paket-Server, ob alle Elemente in Datenpaketen in der Anforderung gültig sind oder nicht und aktualisiert dann den Inhalt in seiner Datenbank. Wenn während der Ausführung einer Anforderung von einem Client ein Fehler auftritt, wenn z. B. der Paket-Server einen oder mehrere ungültige Elemente entdeckt, sollte der Paket-Server die Ausführung der Anforderung beenden und dann eine Ausnahme an den Client ausgeben, mit einigen detaillierten Informationen zu den Fehlern. In einem solchen Fall wird kein Element in den Datenpaketen in der Anforderung hinzugefügt, aktualisiert oder auf dem Paket-Server gelöscht.

Der Paketregistrierungsdienst dient zum Registrieren einer ganzen Ontologie von Anfang an auf einen Paket-Server. Er wird auch verwendet, um den vorhandenen Inhalt einer auf einem Paket-Server registrierten Ontologie zu verändern. Die Änderung des Inhalts kann in der Hinzufügung eines neuen Elements oder der Modifikation oder Löschung des vorhandenen Elements bestehen.

5.4.2 Anforderungsnachricht

5.4.2.1 Allgemeines

Die Anforderungsnachricht des Paketregistrierungsdienstes muss „conjunctiveParcels“ als Parameter enthalten. Der Parameter hat einen Zeichenkettenwert, der aus einem verbundenen Satz von Datenpaketen in JSON oder XML besteht, wovon jede Instanz für die Registrierung auf einen Paket-Server angefordert wird. Jedes Datenpaket in einer Anforderungsnachricht muss die folgenden Informationen enthalten. [Tabelle 3](#) zeigt die Zusammenfassung der zugehörigen Parameter.

Tabelle 3 – Die Struktur einer Anforderungsnachricht des Paketregistrierungsdienstes

Name	Datentyp	Verpflichtung	Beispiel
conjunctiveParcels	Zeichenkette	Verbindlich	<pre><?XML-Version="1.0" Codierung="utf-8"?> <conjunctiveParcels ontoLayer="DO"> <parcel> ... </parcel> </conjunctiveParcels></pre>

5.4.2.2 Anforderung von Eingangsdatenpaketen

5.4.2.2.1 Allgemeine Angaben

Die in IEC 62656-1 definierte Anweisung #PARCEL_MODE muss in allen Datenpaketen spezifiziert werden, die von einem Paketclient über den Paketregistrierungsdienst an einen Paket-Server gesendet werden.

In Bezug auf die Spezifikation von #PARCEL_MODE, FULL und UPDATE wird angenommen, dass das Ergebnis des Vorgangs einen kompletten Satz von Datenpaketen zur Validierung als Ontologie enthält, während PARTIAL in der Voreinstellung davon ausgeht, dass das Ergebnis des Vorgangs ein unvollständiger Satz von Datenpaketen ist. Wenn eine Validierung stattfindet ist das Ergebnis somit im Allgemeinen ein Fehler.

Die Anweisungen werden in den Paketwebdiensten wie folgt genutzt:

- Entweder FULL, UPDATE oder PARTIAL können als Wert für diese Anweisung zugewiesen werden.
- Innerhalb eines verbundenen Satzes von Datenpaketen muss der Wert der Anweisung #PARCEL_MODE für jedes Paket gleichbleibend sein.
- Sind FULL oder UPDATE spezifiziert, sollte auf dem Server eine Validierung durchgeführt werden.

Sind UPDATE oder PARTIAL für #PARCEL_MODE spezifiziert, muss einer der folgenden drei Vorgänge für jede Instanz in einem Datenpaket verwendet werden.

- #ADD (siehe 5.4.2.2.2)
- #MOD (siehe 5.4.2.2.3)
- #DEL (siehe 5.4.2.2.4)

Ist FULL für #PARCEL_MODE spezifiziert, ist es nicht erforderlich, die drei oben bezeichneten Vorgänge anzuwenden. Wenn diese Operationen in einem Datenpaket des FULL-Modus erscheinen, müssen sie ignoriert werden.

5.4.2.2.2 Hinzufüfungsvorgang

Schlüsselwort: #ADD

Bezeichnung: Addition operation

Definition: Vorgang für die Modi UPDATE und PARTIAL, der angibt, dass eine Instanz zum auf einem Paket-Server gespeicherten Inhalt neu hinzugefügt werden soll

Beschreibung: Im Modus UPDATE muss ein Paket-Server die Vervielfältigung der Instanz mit dem/den Wert(en) des (zusammengesetzten) Schlüssels oder des mit MDC_P066 identifizierten Datenobjektbezeichners überprüfen, bevor der Inhalt aktualisiert wird. Wird die Vervielfältigung gefunden, lehnt der Paket-Server die Anforderung ab und gibt dann die standarddefinierte Ausnahme „validation_failed_exception“ an den

Paketclient zurück. Wenn es eine Unstimmigkeit in der Konfiguration des (zusammengesetzten) Schlüssels zwischen den über den Dienst hochgeladenen Datenpaketen und dem auf dem Paket-Server gespeicherten Inhalt gibt, muss der Paket-Server die standarddefinierte Ausnahme „validation_failed_exception“ zurückgeben, die eine Liste von Bezeichnern für Elemente enthält, für die die Validierung in Bezug auf den Paketclient fehlgeschlagen ist.

Modifikationsvorgang

Schlüsselwort: #MOD

Bezeichnung: Modification operation

Definition: Vorgang für die Modi UPDATE und PARTIAL, der angibt, dass die Instanz des auf einem Paket-Server gespeicherten Inhalts, die dem/den Wert(en) des (zusammengesetzten) Schlüssels oder dem mit MDC_P066 identifizierten Datenobjektbezeichner entspricht, modifiziert werden soll.

Beschreibung: Im Modus UPDATE muss ein Paket-Server das Vorhandensein der Instanz überprüfen, bevor der Aktualisierungsprozess ausgeführt wird. Wird sie nicht gefunden, lehnt der Paket-Server die Anforderung ab und gibt dann die standarddefinierte Ausnahme „validation_failed_exception“ an den Paketclient zurück. Wenn es eine Unstimmigkeit in der Konfiguration des (zusammengesetzten) Schlüssels zwischen den über den Dienst hochgeladenen Datenpaketen und dem auf dem Paket-Server gespeicherten Inhalt gibt, muss der Paket-Server die standarddefinierte Ausnahme „validationFailureException“ zurückgeben, die eine Liste von Bezeichnern für Elemente enthält, für die die Validierung in Bezug auf den Paketclient fehlgeschlagen ist.

Löschvorgang

Schlüsselwort: #DEL

Bezeichnung: Deletion operation

Definition: Vorgang für die Modi FULL und PARTIAL, der angibt, dass die Instanz des auf einem Paket-Server gespeicherten Inhalts, die dem/den Wert(en) des (zusammengesetzten) Schlüssels oder dem mit MDC_P066 identifizierten Datenobjektbezeichner entspricht, gelöscht werden soll.

Beschreibung: Im Modus UPDATE muss ein Paket-Server das Vorhandensein der Instanz überprüfen, bevor sie gelöscht wird. Wird sie nicht gefunden, lehnt der Paket-Server die Anforderung ab und gibt dann die standarddefinierte Ausnahme „validation_failed_exception“ an den Paketclient zurück. Eine Instanz mit diesem Vorgang braucht weder Werte des (zusammengesetzten) Schlüssels noch des mit MDC_P066 identifizierten Datenobjektbezeichners. Wenn eine zu löschende Instanz weder über den/die Wert(e) des (zusammengesetzten) Schlüssels noch über den Datenobjektbezeichner verfügt, wird sie vom Ausführungsprozess des Paketauflösungsdienstes ignoriert. Wenn es eine Unstimmigkeit in der Konfiguration des (zusammengesetzten) Schlüssels zwischen den über den Dienst hochgeladenen Datenpaketen und dem auf dem Paket-Server gespeicherten Inhalt gibt, muss der Paket-Server die standarddefinierte Ausnahme „validationFailureException“ zurückgeben, die eine Liste von Bezeichnern für Elemente enthält, für die die Validierung in Bezug auf den Paketclient fehlgeschlagen ist.

5.4.3 Antwortnachricht

Die Antwortnachricht des Paketregistrierungsdienstes muss „operationResult“ als Parameter enthalten. Der Parameter hat einen Boole'schen Wert, der angibt, ob die Ausführung einer Registrierungsanforderung er-

folgreich verläuft oder nicht. Bei erfolgreicher Ausführung gibt das Ontologie-Repository eine „true“-Anzeige an den Paketclient zurück. Schlägt eine Ausführung fehl, wird eine der Ausnahmen an den Paketclient zurückgegeben, zusätzlich zur Rückgabe einer „false“-Anzeige. Tabelle 4 zeigt die Zusammenfassung der Struktur einer Antwortnachricht.

Tabelle 4 – Die Struktur einer Antwortnachricht des Paketregistrierungsdienstes

Name	Datentyp	Verpflichtung	Beispiel
operationResult	Boole'scher Wert	Verbindlich	true

5.4.4 Ausnahme

Die folgenden in 5.2 spezifizierten standarddefinierten Ausnahmen gelten für den Paketregistrierungsdienst.

- no implementation
- element already existing for the #ADD operation
- no element found for the #MOD and #DEL operations
- newer version element existing
- unexpected parameter
- unexpected value
- validation failure
- no permission

5.5 Paketauflösungsdienst

5.5.1 Allgemeine Angaben

Der Paketauflösungsdienst ist ein Service, der einem Paketclient ermöglicht, auf einem Paket-Server gespeicherte ontologische Entitäten aufzulösen. Eine Auflösung auf einem Paket-Server erfolgt mithilfe von Bezeichnern oder Namen, die in einer Anforderungsnachricht von einem Paketclient angegeben sind. Wenn es ontologische Entitäten gibt, welche diesen Bezeichnern oder Namen auf einem Paket-Server entsprechen, gibt der Paket-Server Informationen dieser ontologischen Entitäten in einem XML- oder JSON-Format in Übereinstimmung mit diesem Dokument zurück. Wenn während der Ausführung einer Anforderung von einem Client ein Fehler auftritt, sollte der Paket-Server die Ausführung der Anforderung beenden und dann eine Ausnahme an den Client ausgeben, mit einigen detaillierten Informationen zu den Fehlern. In einem solchen Fall wird kein Inhalt an den Paketclient zurückgegeben.

Der Paketauflösungsdienst stellt zwei Arten von Auflösungen zur Angabe eines Bezeichners einer ontologischen Identität wie folgt bereit:

- Definitionsauflösung zum Auflösen von Definitionen ontologischer Entitäten, die zu Bezeichnern oder Namen passen, die in einer Anforderungsnachricht spezifiziert sind.
- Instanzauflösung zum Auflösen von Instanzen ontologischer Entitäten, die zu Bezeichnern oder Namen passen, die in einer Anforderungsnachricht definiert sind.

Ein Parameterwert der Anforderungsnachricht bestimmt, welche Auflösungsart gewählt wird.

BEISPIEL 1 In einer Definitionsauflösung werden bei Angabe des Bezeichners einer Klasse in einer Anforderungsnachricht alle Informationen der Klasse wie Code, Name und Definition zurückgegeben. Solche Informationen bestehen aus Paaren eines in IEC 62656-1 definierten Klassenattributs und seines Werts.

BEISPIEL 2 In einer Instanzauflösung werden bei Angabe des Bezeichners einer Klasse in einer Anforderungsnachricht alle Instanzen der Klasse zurückgegeben.

Wenn viele Bezeichner gleichzeitig in einer Anforderungsnachricht angegeben sind, kann das Ergebnis einen verbundenen Satz von Datenpaketen enthalten. Sind beispielsweise die Bezeichner der Klassenmetaklasse und der Eigenschaftsmetaklasse in einer Anforderungsnachricht spezifiziert, enthält die Antwortnachricht einen verbundenen Satz von Datenpaketen, der Instanzen dieser Metaklassen umfasst.

Hinweis Ein komplexer Abfragemechanismus wie der Abruf von Klassen über eine Klassenhierarchie wird entweder in einer anderen Norm oder in einem anderen Teil der IEC 62656 definiert.

Wenn ein Bezeichner einer Elementklasse in einer Anforderungsnachricht angegeben ist, beinhaltet eine in einer Antwortnachricht zu der Anforderung enthaltene Instanz Werte aller Eigenschaften der Elementklasse. Wenn der Paket-Server eine Klasse als Ansicht für eine Elementklasse vorbereitet, ist der Bezeichner der Klasse als Ansicht verfügbar, anstatt den Bezeichner der Elementklasse anzugeben. Das heißt, dass eine solche Klasse als Ansicht zur Festlegung eines vordefinierten Satzes nützlicher oder sinnvoller Anfragen zu den Eigenschaften eines Produkts verwendet wird. Eine solche Klasse als Ansicht kann auf Basis des in ISO 13584-24:2003 definierten Wörterbucheintragmechanismus modelliert werden. Sie kann verstanden werden als ein Satz typischer Anfragen zu einer Produktklasse wie etwa FAQ (häufig gestellte Fragen).

Bild 7 zeigt ein Beispiel für den Instanzauflösungsdienst. Die Tabelle in Bild 7 enthält Instanzen einer Elementklasse C001 in einem Arbeitsblattformular der IEC 62656-1. Eine Klasse QCA001 ist eine Klasse als Ansicht für die Elementklasse V001, die Eigenschaften P001, P002 und P003 aus der Elementklasse C001 importiert. Wenn Anwender A eine Anforderung durch Angabe von C001 sendet, erhält er alle Instanzen der Klasse. Zwischenzeitlich sendet Anwender B eine Anforderung durch Angabe von QCA001 anstatt von C001, um Instanzen der Elementklasse C001 zu erhalten, von der jede nur Werte von P001, P002 und P003 hat.

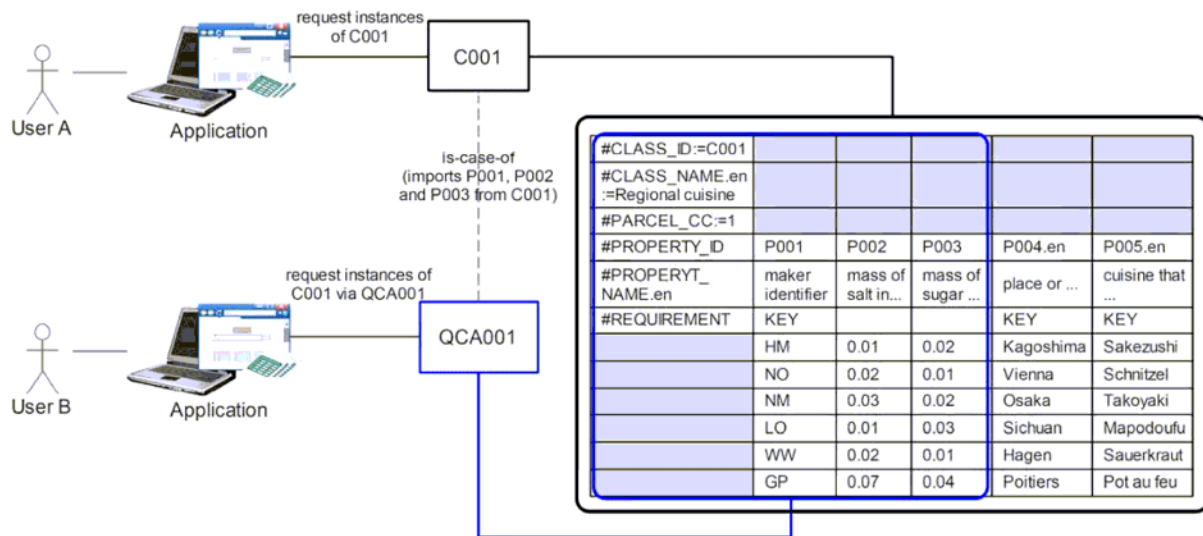


Bild 7 – Übersicht des Paketauflösungsdienstes

5.5.2 Anforderungsnachricht

5.5.2.1 Allgemeine Angaben

Die Anforderungsnachricht des Paketauflösungsdienstes muss als Parameter „requestKind“, „keywordKind“, „keyword“, „language“, „pwsCodificationMode“, „startPoint“ und „endPoint“ enthalten. Tabelle 5 zeigt die Zusammenfassung dieser Parameter.

Tabelle 5 – Die Struktur einer Anforderungsnachricht des Paketauflösungsdienstes

Name	Datentyp	Verpflichtung	Beispiel
requestKind	Zeichenkette	Verbindlich	DEFINITION
keywordKind	Zeichenkette	Verbindlich	ID
keyword	Zeichenkette	Verbindlich	0112/2///61360_4#AAA013 0112/2///61360_4#AAA013* Bemessungsspannung
language	Zeichenkette	Optional	en en, fr
pwsCodificationMode	Zeichenkette	Verbindlich	VERTIKAL
startPoint	Ganze Zahl	Optional	10
endPoint	Ganze Zahl	Optional	101

5.5.2.2 requestKind

Der Parameter „requestKind“ hat einen Zeichenkettenwert, entweder von „DEFINITION“ oder „INSTANCE“.

Ist „DEFINITION“ spezifiziert, wird davon ausgegangen, dass Informationen einer ontologischen Entität selbst aufgelöst werden, die zu dem in einer Anforderungsnachricht angegebenen Bezeichner oder Namen passen. Dies dient dem Abruf aller Informationen einer Klasse durch Angabe des entsprechenden Bezeichners, einschließlich Code, bevorzugter Name und Definitionen.

BEISPIEL 1 Im Modus „DEFINITION“ werden, wenn der Bezeichner einer Klasse angegeben ist, Informationen der Klasse abgerufen, einschließlich Code, Name und Definition.

Ist „INSTANCE“ spezifiziert, wird davon ausgegangen, dass Instanzen ontologischer Entitäten aufgelöst werden, die zu in einer Anforderungsnachricht angegebenen Bezeichnern oder Namen passen. Dies dient beispielsweise dem Erhalt von Codes aller Klassen in einer Ontologie durch Angabe des Bezeichners der Klassenmetaklasse.

BEISPIEL 2 Im Modus „INSTANCE“ werden, wenn der Bezeichner einer Klassenmetaklasse angegeben ist, Informationen aller Klassen einer Ontologie abgerufen, einschließlich Code, Name und Definition.

5.5.2.3 keywordKind

Der Parameter „keywordKind“ hat einen Zeichenkettenwert, entweder von „ID“ oder „NAME“. Ist „ID“ spezifiziert, muss ein Wert des Parameters „keyword“ eine Liste von Bezeichnern ontologischer Entitäten haben. Ist „NAME“ spezifiziert, muss ein Wert des Parameters „keyword“ eine Liste von Namen ontologischer Entitäten haben.

5.5.2.4 keyword

Der Parameter „keyword“ hat einen Zeichenkettenwert, der aus einem der folgenden Elemente besteht:

- einer Liste von Bezeichnern ontologischer Entitäten mit/ohne eine Suchumfangsergänzung, wie in 5.3 angegeben,
- einer Liste von Namen ontologischer Entitäten.

Wenn ein Paket-Server Schlüsselwörter über einen Paketwebdienst erhält, gibt er an seinen Client ein Ergebnis zurück, das zu einem der Schlüsselwörter passt.

Hinweis 1 Ist mehr als ein Element angegeben, werden die einzelnen Elemente durch ein Komma getrennt „“.

Hinweis 2 Wenn ein Name selbst ein doppeltes Anführungszeichen enthält, muss dieses von einem weiteren doppelten Anführungszeichen codiert werden.

Hinweis 3 Wenn ein Name selbst ein Komma oder doppeltes Anführungszeichen enthält, muss der Name zwischen einem doppelten Anfangs- und Endanführungszeichen eingeschlossen sein.

5.5.2.5 language

Der Parameter „language“ hat einen Zeichenkettenwert, der aus einer Liste von Sprachen besteht, aus der jede Sprache eines der folgenden Dinge ist gemäß dem in IEC 62656-1 definierten ICID:

- Zwei-Buchstaben-Code basierend auf ISO 639-1, wie etwa „en“,
- die Kombination eines Zwei-Buchstaben-Sprachcodes basierend auf ISO 639-1 und eines Zwei-Buchstaben-Landescodes basierend auf ISO 3166-1, verbunden mit einem Bindestrich, z. B. „en-US“.

Wenn eine oder mehr Sprachen spezifiziert sind, gibt ein Paket-Server den Inhalt der angegebenen Sprachen an seinen Client zurück.

Hinweis 1 Ist mehr als eine Sprache angegeben, werden die einzelnen Sprache durch ein Komma getrennt „“.

Hinweis 2 Ist in einer Anforderung keine Sprache angegeben, wird der Inhalt einer Standardsprache zurückgegeben.

5.5.2.6 pwsCodificationMode

Der Parameter „pwsCodificationMode“ hat einen Zeichenkettenwert, entweder von „VERTICAL“ oder „LATERAL“. Ist „VERTICAL“ spezifiziert, muss ein als Ergebnis des Paketauflösungsdienstes zurückgegebener verbundener Satz von Datenpaketen in der in 6 definierten vertikalen Codierung strukturiert sein. Ist „LATERAL“ spezifiziert, muss ein als Ergebnis des Paketauflösungsdienstes zurückgegebener verbundener Satz von Datenpaketen in der in 6 definierten lateralen Codierung strukturiert sein.

5.5.2.7 startPoint

Der Parameter „startPoint“ gibt einen Startpunkt oder Fortsetzungspunkt eines (verbundenen Satzes von) Datenpaketen als Ergebnis der Ausführung einer Auflösungsanforderung an. „startPoint“ hat einen positiven ganzzahligen Wert. Sein Standardwert ist 1. Ist dieser Parameter nicht spezifiziert, wird der Standardwert angewendet.

Für jede Instanz, die in einem (verbundenen Satz von) Datenpaketen als Ergebnis der Ausführung einer Auflösungsanforderung enthalten ist, muss die Zahl virtuell zugewiesen werden.

5.5.2.8 endPoint

Der Parameter „endPoint“ gibt einen Endpunkt eines Ergebnisses eines Paketauflösungsdienstes an. „endPoint“ hat einen positiven ganzzahligen Wert, der mindestens so groß wie der Wert des Parameters „startPoint“ sein muss. Ist dieser Parameter nicht spezifiziert, werden alle Elemente nach dem Startpunkt als Ergebnis der Anforderung zurückgegeben.

Für jede Instanz, die in einem (verbundenen Satz von) Datenpaketen als Ergebnis der Ausführung einer Auflösungsanforderung enthalten ist, muss die Zahl virtuell zugewiesen werden.

5.5.3 Antwortnachricht

Die Antwortnachricht des Paketauflösungsdienstes muss „conjunctiveParcels“ als Parameter enthalten. Der Parameter hat einen Zeichenkettenwert, der aus einem verbundenen Satz von Datenpaketen besteht, der den in einer Anforderungsnachricht angegebenen Bedingungen entspricht. [Tabelle 6](#) zeigt die Zusammenfassung der Struktur einer Antwortnachricht.

Tabelle 6 – Die Struktur einer Antwortnachricht des Paketauflösungsdienstes

Name	Datentyp	Verpflichtung	Beispiel
conjunctiveParcels	Zeichenkette	Verbindlich	<pre><?xml version="1.0" encoding="utf-8"?> <conjunctiveParcels ontoLayer="DO"> <parcel> ... </parcel> </conjunctiveParcels></pre>

5.5.4 Ausnahme

Die folgenden in 5.2 spezifizierten standarddefinierten Ausnahmen gelten für den Paketauflösungsdienst.

- no implementation
- no element found
- unexpected parameter
- unexpected value
- no permission

5.6 Paketabonnementsdienst

5.6.1 Allgemeine Angaben

Der Paketabonnementsdienst ist ein Service, der einem Paketclient ermöglicht, den Empfang einer Mitteilung zu beginnen oder zu beenden, die den Paketclient darüber informiert, dass der Status, die Informationen oder ein Wert des angegebenen Elements verändert wurde. In diesem Fall ist mit der Änderung eines Elements eine Hinzufügung, Modifikation oder Löschung gemeint. Wenn während der Ausführung einer Anforderung von einem Client ein Fehler auftritt, wenn z. B. der Paket-Server einen oder mehrere ungültige Bezeichner entdeckt, sollte der Paket-Server die Ausführung der Anforderung beenden und dann eine Ausnahme an den Client ausgeben, mit einigen detaillierten Informationen zu den Fehlern. In einem solchen Fall wird der Abonnementstatus auf dem Paket-Server nicht verändert.

Der Paketabonnementsdienst ermöglicht Push-Abonnements. D. h., wenn der Status oder ein Wert eines Elements verändert wird, sendet der Paket-Server eine Mitteilung an seine Abonnenten.

5.6 Anforderungsnachricht

5.6.1 Allgemeine Angaben

Die Anforderungsnachricht des Paketabonnementsdienstes muss „requestKind“, „identifiers“ und „subscriptionOperationKind“ als Parameter enthalten. [Tabelle 7](#) zeigt die Zusammenfassung der Struktur einer Anforderungsnachricht.

Tabelle 7 – Die Struktur einer Anforderungsnachricht des Paketabonnementdienstes

Name	Datentyp	Verpflichtung	Beispiel
requestKind	Zeichenkette	Verbindlich	DEFINITION
identifiers	Zeichenkette	Verbindlich	0112/2///61360_4#AAA013 0112/2///61360_4#AAA013* AAA013
subscriptionOperationKind	Zeichenkette	Verbindlich	BEGINN

5.6.2.2 requestKind

Der Parameter „requestKind“ hat einen Zeichenkettenwert, entweder von „DEFINITION“ oder „INSTANCE“.

Ist „DEFINITION“ spezifiziert, wird davon ausgegangen, dass ein Abonnement der Informationen einer ontologischen Entität selbst, die zu dem in einer Anforderungsnachricht angegebenen Bezeichner passen, begonnen oder beendet wird.

Ist „INSTANCE“ spezifiziert, wird davon ausgegangen, dass ein Abonnement von Instanzen ontologischer Entitäten, die zu in einer Anforderungsnachricht angegebenen Bezeichnern passen, begonnen oder beendet wird.

5.6.2.3 identifiers

Der Parameter „identifiers“ hat einen Zeichenkettenwert, der aus einer Liste von Bezeichnern ontologischer Entitäten mit/ohne Suchumfangergänzungen, wie in 5.3 angegeben, besteht, die ein Paketclient oder Anwender anfordert, um ein entsprechendes Abonnement zu starten.

5.6.2.4 subscriptionOperationKind

Der Parameter „subscriptionOperationKind“ hat einen Zeichenkettenwert, entweder „START“ oder „END“. Ist „START“ spezifiziert, wird ein Abonnement ontologischer Entitäten von Bezeichnern gestartet, die im Parameter „identifiers“ angegeben sind. Ist „END“ spezifiziert, wird ein Abonnement ontologischer Entitäten von Bezeichnern beendet, die im Parameter „identifiers“ angegeben sind.

5.6.3 Antwortnachricht

Die Antwortnachricht muss „operationResult“ als Parameter enthalten. Der Parameter hat einen Boole'schen Wert, der angibt, ob die Ausführung einer Abonnementsanforderung erfolgreich verläuft oder nicht. Bei erfolgreicher Ausführung gibt der Ontologie-Server eine „true“-Anzeige an den Paketclient zurück. Schlägt eine Ausführung fehl, wird eine der Ausnahmen ausgegeben oder an einen Paketclient zurückgegeben, zusätzlich zur Rückgabe einer „false“-Anzeige. Tabelle 8 zeigt die Zusammenfassung der Struktur einer Antwortnachricht.

Tabelle 8 – Die Struktur einer Antwortnachricht des Paketabonnementdienstes

Name	Datentyp	Verpflichtung	Beispiel
operationResult	Boole'scher Wert	Verbindlich	true

5.6.4 Ausnahme

Die folgenden in 5.2 spezifizierten standarddefinierten Ausnahmen gelten für den Paketabonnementdienst.

- no implementation
- no element found

- unexpected parameter
- unexpected value
- no permission

5.6.5 Spezifikation einer Änderungsmitteilung

Eine Benachrichtigung, die an einen Paketclient gesendet wird, besteht aus einem obligatorischen Parameter „identifiers“ und zwei optionalen Parametern „defaultDataSupplier“ und „defaultDataVersion“. Tabelle 9 zeigt die Zusammenfassung der Spezifikation einer Mitteilung.

Der Parameter „identifiers“ dient zur Spezifizierung einer Liste von Bezeichnern von Elementen, an denen der Paketclient interessiert ist. Diese Bezeichner werden beschrieben unter Verwendung eines Kommas als Separator.

Die optionalen Parameter können verwendet werden, um die Wiederholung des gleichen RAI oder VI zu verhindern und somit eine reduzierte Datengröße der Mitteilung zu ermöglichen. Wenn „defaultDataSupplier“ in einer Mitteilung vorhanden ist, kann ein RAI aus einigen Bezeichnern in der Mitteilung ausgelassen werden. In diesem Fall sollte der Wert von „defaultDataSupplier“ auf diese Bezeichner angewendet werden. Wenn gleichfalls „defaultDataVersion“ in einer Mitteilung vorhanden ist, kann ein VI aus einigen Bezeichnern in der Mitteilung ausgelassen werden. In diesem Fall sollte der Wert von „defaultDataVersion“ auf diese Bezeichner angewendet werden.

Tabelle 9 – Die Spezifikation einer Mitteilung

Name	Datentyp	Verpflichtung	Beispiel
identifiers	Zeichenkette	Verbindlich	
defaultDataSupplier	Zeichenkette	Optional	
defaultDataVersion	Zeichenkette	Optional	

6 Spezifikation einer Paketdatendarstellung in einer Webdienstnachricht

6.1 Allgemeine Angaben

Ein Format von über einen Webdienst ausgetauschten Daten sollte die folgenden Kriterien erfüllen.

- Spezifikation ist öffentlich zugänglich,
- das Format ist weit verbreitet in Unternehmenssystemen und -geräten einschließlich mobiler Geräte und Anwendungen,
- Software-Bibliotheken für verschiedene Programmiersprachen stehen bereit.

In Übereinstimmung mit diesen Kriterien werden XML und JSON in diesem Dokument übernommen.

In der Industrie wurde XML traditionell als Datenformat für den Datenaustausch zwischen Industriesystemen verwendet. XML hat einige Vorteile, wie die klar definierte Schema- und Sicherheitstechnik im Netz. ISO 13584-32, als OntoML bezeichnet, definiert das XML-Schema [3] zur Darstellung von ISO 13584-konformen Daten in einem XML-Formular. Andererseits gibt es einige Nachteile bei XML wie etwa die Datengröße, da alle Daten in einem Paar von XML-Tags eingeschlossen sein sollten.

Eine weitere Datennotation, JSON, wird ebenfalls genutzt, insbesondere in mobilen Anwendungen. JSON ermöglicht eine einfache und leichte Datenbeschreibung auf Basis der JavaScript-Notation. Wegen der Benutzerfreundlichkeit und leichten Verständlichkeit wird JSON nicht nur in JavaScript verwendet, sondern auch in anderen beliebten Programmiersprachen wie Java, C#, PHP und Python, als ein Datenformat, das zwischen Systemen, (mobilen) Geräten und Anwendungen ausgetauscht wird.

6.2 Grundlegende Datendarstellung

Datenpakete werden zwischen einem Paket-Server und einem Paketclient ausgetauscht, die in eine Anforderung oder Antwortnachricht eingebunden sind. Dieses Dokument legt die Datendarstellung von Datenpaketen in einer Nachricht fest.

Bild 8 zeigt die konzeptionelle Datenstruktur eines verbundenen Satzes von Datenpaketen, wie in IEC 62656-1 definiert. Ein verbundener Satz von Datenpaketen mit der Bezeichnung „conjunctive parcels“ umfasst ein oder mehrere Datenpakete, von denen jedes einzelne als „parcel“ bezeichnet wird. Jedes Datenpaket besteht aus zwei Bereichen, der Reihe nach aus einem Kopfbereich und einem Datenbereich. Der Kopfbereich umfasst seinerseits zwei Unterabschnitte, die einen Klassenkopfbereich und einen Schemakopfbereich darstellen. Der Klassenkopfbereich beschreibt die Informationen zum Datenpaket, während der Schemakopfbereich das Datenschema der Daten im Paket beschreibt. Der Datenbereich beschreibt Instanzen in Übereinstimmung mit dem im Schemakopfbereich erläuterten Schema.

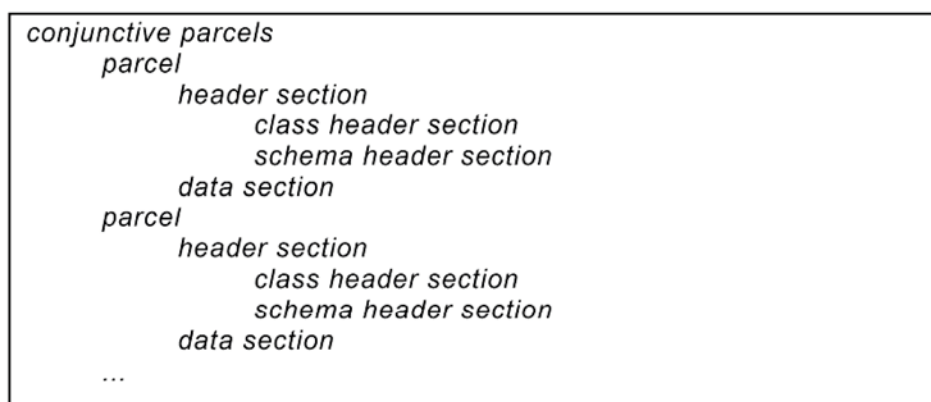


Bild 8 – Grundstruktur einer Datendarstellung für einen verbundenen Satz von Datenpaketen

Die folgende 6.3 legt reservierte Schlüsselwörter für die Datendarstellung sowohl in JSON als auch in XML fest. Für die Datendarstellung in JSON wird jedes einzelne Schlüsselwort als JSON-Name verwendet. Für die Datendarstellung in XML wird jedes einzelne Schlüsselwort als XML-Elementname oder Attributname verwendet.

6.3 Reservierte Schlüsselwörter

6.3.1 Schlüsselwort zur Angabe von verbundenen Datenpaketen

Schlüsselwort: conjunctiveParcels
Bezeichnung: conjunctive parcels
Definition: reserviertes Schlüsselwort zur Sammlung von einem oder mehreren Datenpaketen, von denen jedes im Schlüsselwort „parcel“ eingeschlossen ist
Auftreten: 1 für eine Anforderung oder Antwortnachricht

Schlüsselwort zur Angabe einer Paketontologieschicht eines Satzes von Datenpaketen

Schlüsselwort: ontoSchicht
Bezeichnung: parcel ontology layer
Definition: reserviertes Schlüsselwort zur Beschreibung einer Paketontologieschicht eines Satzes von Datenpaketen, gesammelt durch das Schlüsselwort „conjunctiveParcels“
Beschreibung: Der Wert muss entweder „AO“, „MO“, „DO“ oder „DL“ lauten
Auftreten: 1 für ein Auftreten des Schlüsselworts „conjunctiveParcels“

Schlüsselwort zur Angabe eines Kopfbereichs

Schlüsselwort: header
Bezeichnung: header section
Definition: reserviertes Schlüsselwort zur Sammlung des in das Schlüsselwort „classHeader“

eingeschlossenen Klassenkopfbereichs und des in das Schlüsselwort „schemaHeader“ eingeschlossenen Schemakopfbereichs
Auftreten: 1 für ein Auftreten des Schlüsselworts „parcel“

Schlüsselwort zur Angabe eines Klassenkopfbereichs

Schlüsselwort: classHeader
Bezeichnung: class header section
Definition: reserviertes Schlüsselwort zur Sammlung von Anweisungen des Klassenkopfbereichs
Auftreten: 1 für ein Auftreten des Schlüsselworts „header“

Schlüsselwort zur Angabe eines Schemakopfbereichs

Schlüsselwort: schemaHeader
Bezeichnung: schema header section
Definition: reserviertes Schlüsselwort zur Sammlung von Anweisungen des Schemakopfbereichs
Auftreten: 1 für ein Auftreten des Schlüsselworts „header“

Schlüsselwort zur Angabe eines Datenbereichs

Schlüsselwort: data
Bezeichnung: data section
Definition: reserviertes Schlüsselwort zur Sammlung von Instanzen
Auftreten: 1 für ein Auftreten des Schlüsselworts „conjunctiveParcels“

Schlüsselwort zur Angabe eines Standardlieferanten im Datenbereich

Schlüsselwort: defaultSupplier
Bezeichnung: default supplier
Definition: reserviertes Schlüsselwort zur Angabe eines Standardlieferanten von Eigenschaftsbezeichnern im Datenbereich, von denen jeder als Schlüssel für seine Werte verwendet wird
Auftreten: 1 für ein Auftreten des Schlüsselworts „data“

Schlüsselwort zur Angabe einer Standardversion im Datenbereich

Schlüsselwort: defaultVersion
Bezeichnung: default version
Definition: reserviertes Schlüsselwort zur Angabe einer Standardversion von Eigenschaftsbezeichnern im Datenbereich, von denen jeder als Schlüssel für seine Werte verwendet wird
Auftreten: 1 für ein Auftreten des Schlüsselworts „data“

6.4 Zusätzliche Anweisungen an Datenpakete für Paketwebdienste

6.4.1 Codierungsmodus

Sowohl in JSON als auch in XML stehen zwei Codierungsmodi bereit. Eine ist die vertikale Codierung und die andere die laterale.

Vertikale Codierung ist ein Modus zum Sammeln von Werten je Eigenschaft. Die vertikale Codierung ist geeignet zum Erhalt von Werten einer Eigenschaft, z. B. um zu prüfen, ob ein festgelegter Wert in einer Liste von Werten einer Eigenschaft existiert oder nicht und zur Aggregation und Berechnung von Werten einer Eigenschaft.

Laterale Codierung ist ein Modus zum Sammeln von Werten je Instanz. Die laterale Codierung ist geeignet zum Erhalt von Eigenschafts-Wert-Paaren einer Instanz.

Ein Vorteil der Bereitstellung dieser beiden Modi ist die Kontinuität eines Prozesses. Tritt ein Fehler auf während der Kommunikation zwischen einem Paket-Server und einem Paketclient, darf der Paket-Server (oder der Paketclient) im Ergebnis nur eine Teilmenge der Datenpakete erhalten. Im Fall der vertikalen Codierung kann der Paket-Server (oder Paketclient), wenn er (oder der Paketclient) einen Wertesatz einer Eigenschaft benötigt und dieser in der Teilmenge der Datenpakete enthalten ist, mit der Prozessausführung fortfahren. Im Fall der lateralen Codierung ist ein ähnliches Szenario anwendbar.

Das vorliegende Dokument definiert die folgende Anweisung #PWS_CODIFICATION_MODE für die Angabe, welcher Modus zur Darstellung eines verbundenen Satzes von über einen Paketwebdienst ausgetauschten Datenpaketen verwendet wird.

Schlüsselwort: #PWS_CODIFICATION_MODE

Bezeichnung: codification mode

Definition: Anweisung für ein Datenpaket zur Angabe, welche Codierung für die Darstellung des Datenpakets gewählt wird

Beschreibung: Ist „LATERAL“ spezifiziert, wird das Datenpaket in der lateralen Codierung dargestellt. Ist „VERTICAL“ spezifiziert, wird das Datenpaket in der vertikalen Codierung dargestellt.

Kategorie: optional – funktional

Format: Entweder LATERAL oder VERTICAL kann als Wert zugewiesen werden.

6.4.2 Vorgesehene Sprache

Es ist zulässig, dass ein Datenpaket Definitionen in mehreren Sprachen enthält. Über einen Paketregistrierungsdienst wird der in einem verbundenen Satz von Datenpaketen enthaltene Inhalt aller Sprachen auf der Empfangsseite (im Allgemeinen ein Paket-Server) verarbeitet, dies ist jedoch manchmal gefährlich, da er unbeabsichtigte Änderungen in nicht vorgesehenen Sprachen enthalten kann. Von einer Übersetzung aus dem Englischen in eine andere Sprache kann beispielsweise der englische Inhalt nur als Referenz für die Übersetzung in eine als „language variant“ bezeichnete andere Sprache verwendet werden. In diesem Fall muss die englische Definition selbst dann auf dem Paket-Server ignoriert werden, wenn sie versehentlich verändert wird. Daher ist/sind die wirklich für eine Änderung vorgesehene(n) Sprachdefinition(en) deutlich zu kennzeichnen. Um eine unbeabsichtigte Änderung des Inhalts zu vermeiden, definiert dieses Dokument die folgende Anweisung „#INTENDED_LANGUAGE“ für den Paketregistrierungsdienst.

Schlüsselwort: #INTENDED_LANGUAGE

Bezeichnung: intended language(s)

Definition: Bezeichnung der Sprache(n) entsprechend ISO 639, welche vorgesehenen Definitionsänderungen enthält/enthalten

Beschreibung: Wenn es eine Sprache in einem Datenpaket gibt, die nicht als vorgesehene Sprache angegeben ist, muss deren Definition auf der Empfangsseite (im Allgemeinen ein Paket-Server) ignoriert werden. Ist diese Anweisung nicht in einem Datenpaket spezifiziert, müssen alle Sprachen im Datenpaket einschließlich der Quellsprache als beabsichtigt betrachtet werden. Die Quellsprache kann wie auch andere Sprachen als ein Element des Werts dieses Befehls definiert werden. Ist die Quellsprache festgelegt, enthält die Beschreibung in der Quellsprache einige vorgesehene Definitionsänderungen.

Kategorie: optional – funktional

Format: Das Schlüsselwort „#INTENDED_LANGUAGE“ muss in der ersten Spalte beschrieben werden, und Sprachcode(s) muss/müssen nach dem durch das Symbol „:=“ (Ergebnzeichen) getrennten Schlüsselwort beschrieben werden. Der Sprachcode gemäß ISO 639-1, möglicherweise gefolgt von einem auf ISO 3166-1 basierenden zweistelligen Ländercode, ermöglicht die Identifizierung der Sprache. Sind mehrfache Sprachen spezifiziert, müssen diese Sprachcodes durch Kommata abgetrennt und in geschweifte Klammern „{“ und „}“ gesetzt werden. Die Zellen in der zweiten Spalte und danach müssen ignoriert werden.

6.4.2 Standardwert

Bei einigen Computernetzwerkumgebungen wie in einem mobilen Netzwerk und einem Netzwerk mit geringer Bandbreite sollte ein Verfahren zur Reduzierung der zwischen einem Server und einem Client auszu-

tauschenden Gesamtdatenmenge in Betracht gezogen werden, um eine effiziente und zuverlässige Netzwerkkommunikation zu erreichen.

IEC 62656-1 spezifiziert zwei Anweisungen `#DEFAULT_DATA_SUPPLIER` und `#DEFAULT_DATA_VERSION` zur Vermeidung von Wiederholungen des gleichen RAL bzw. VL von ICIDs im Datenbereich. Auf ähnliche Weise wäre auch ein Verfahren zum Festlegen eines Standardwerts einer Eigenschaft hilfreich.

Das vorliegende Dokument definiert die folgende Anweisung `#DEFAULT_VALUE` zur Angabe eines Standardwerts einer Eigenschaft. Ein Standardwert, der auf eine Eigenschaftsspalte im Blatt eines Datenpakets eingestellt ist, ist nur in der Spalte wirksam.

Schlüsselwort: `#DEFAULT_VALUE`

Bezeichnung: default value

Definition: durch die Eigenschafts-ID spezifizierte Voreinstellung der Eigenschaft

Beschreibung: Der Standardwert für die Eigenschaft muss auf alle leeren Felder der Eigenschaft im Datenbereich angewendet werden, wo die Werte nicht festgelegt sind. Um einen Nullwert in einem Feld beizubehalten, muss ein spezieller Wert „(null)“, der aus einer in Klammern gesetzten Zeichenkette „null“ besteht, im Feld ausdrücklich beschrieben werden. Wenn ein Datenpaket, das einen Standardwert enthält, an ein externes System zu senden ist, das den Standardwert nicht verarbeiten kann, sollte das Datenpaket vor dem eigentlichen Versand an das externe System vorverarbeitet werden, um leere Felder mit dem Standardwert auszufüllen, außer wenn das Feld ausdrücklich als „(null)“ gekennzeichnet ist.

Kategorie: optional – funktional

Format: Das Schlüsselwort `#DEFAULT_VALUE` wird in der ersten Spalte beschrieben. Die Standardwerte werden in den Zellen beschrieben, die den zweiten und folgenden Spalten entsprechen. Jeder solche Standardwert entspricht der Eigenschaft, die in der Zeile `#PROPERTY_ID` beschrieben ist.

Bild 9 zeigt, wie Standardwerte beschrieben und auf Zellen im Datenbereich angewendet werden. Von dicken Linien umkreiste leere Zellen weisen darauf hin, dass Werte in diesen Zellen durch ihre entsprechenden Standardwerte ersetzt werden, wenn die Datenpakete verarbeitet werden.

#PROPERTY_ID	P001	P002	P003	P004	P005
#DATATYPE	STRING_TYPE	STRING_TYPE	LEVEL(MAX) OF INT_TYPE	BOOLEAN_TYPE	ENUM_REAL_TYPE(E001 (1.0, 2.0, 3.0))
#DEFAULT_VALUE	New York		2	TRUE	1.0
			1.	FALSE	2.0
	Paris				3.0
			3		2.0

Bild 9 – Beispiel für die Verwendung von Standardwerten

6.5 Beschreibung der Anweisungen

IEC 62656-1 definiert die im Kopfbereich jedes Datenpakets zu verwendenden Anweisungen. Diese Anweisungen sind in vier Typkategorien aufgeteilt, „mandatory“, „optional – functional“, „optional – informative“ und „comment“. Gemäß **IEC 62656-1** ist „mandatory“ eine Kategorie, die angibt, dass eine zugehörige Anweisung in einem Datenpaket vorhanden sein muss. „optional – functional“ ist eine Kategorie, die angibt, dass wenn eine zugehörige Anweisung in einem Datenpaket vorhanden ist, ihr Wert entsprechend der durch das Schlüsselwort implizierten Funktion verarbeitet werden muss. „optional – informative“ ist eine Kategorie, die angibt, dass ein Wert einer zugehörigen Anweisung den Anwendern eines Datenpakets nur als Informationsmeldung bereitgestellt wird. „comment“ schließlich ist eine Kategorie, die angibt, dass eine Zeile, in der eine entsprechende Anweisung beschrieben ist, als Kommentar interpretiert werden muss. Der in diesem Dokument spezifizierte Webdienst soll für den Datenaustausch zwischen einem Paket-Server und einem

Paketclient implementiert werden, weshalb sowohl Anweisungen des Typs „mandatory“ als auch des Typs „optional – functional“ in einem zwischen ihnen ausgetauschten Datenpaket behandelt werden müssen.

Die Zeichenkette jeder Anweisung beginnt mit dem Rautenzeichen „#“, z. B. #CLASS_ID zur Beschreibung des Klassenbezeichners für ein Datenpaket. Die XML verbietet jedoch die Verwendung eines Rautenzeichens „#“ als Anfangszeichen eines Elements und Attributs. Daher definiert dieses Dokument Benennungskonventionen entsprechend den in IEC 62656-1 festgelegten Anweisungen, wodurch diese für die Datendarstellung in XML oder JSON sicher verwendet werden können.

Die Benennungskonventionen sind folgende:

- Ein Anfangsrautenzeichen „#“ muss entfernt werden.
- Ein Unterstrich „_“ muss entfernt werden.
- Jeder Buchstabe direkt nach dem Unterstrich „_“ muss groß geschrieben werden.
- Die übrigen Buchstaben müssen klein geschrieben werden.

So wird beispielsweise die Anweisung „#SUPER_ALT_CLASSID“ geändert zu "superAltClassID".

Die Beschreibungen der in diesem Dokument spezifizierten Anweisungen sind in Tabelle 10 aufgelistet.

Tabelle 10 – Beschreibung der in IEC 62656-1 spezifizierten Anweisungen

In IEC 62656-1 definiertes Anweisungsschlüsselwort	Beschreibung in XML und JSON
#CLASS_ID	classID
#PARCEL_MODE	parcelMode
#PARCEL_ID	parcelID
#DEFAULT_SUPPLIER	defaultSupplier
#DEFAULT_VERSION	defaultVersion
#DEFAULT_DATA_SUPPLIER	defaultDataSupplier
#DEFAULT_DATA_VERSION	defaultDataVersion
#OBJECT_ID_NAME	objectIdName
#PROPERTY_ID	propertyID
#REQUIREMENT	requirement
#ID_ENCODE	idEncoding

Hinweis Das vorliegende Dokument beschränkt nicht die Verwendung von Anweisungsschlüsselwörtern von „optional - informative“ in einem Webdienst. [Tabelle E.1](#) liefert die Beschreibung für diese Schlüsselwörter.

Darüber hinaus ist die Übersetzung der in diesem Dokument definierten Anweisung in Tabelle 11 aufgeführt.

Tabelle 11 – Beschreibung der in diesem Dokument spezifizierten Anweisungen

In diesem Dokument festgelegtes Schlüsselwort	Beschreibung in XML und JSON
#PWS_CODIFICATION_MODE	pwsCodificationMode
#INTENDED_LANGUAGE	intendedLanguage
#DEFAULT_VALUE	defaultValue

7 Datendarstellung in JSON

7.1 Basisstruktur der Datendarstellung in JSON

In Übereinstimmung mit ISO/IEC 21778 ist ein JSON-Wert entweder ein Objekt, ein Array, eine Zahl, eine Zeichenkette, wahr, falsch oder null. Jedes JSON-Objekt ist dargestellt als ein Paar geschweiften Klammern „{“ und „}“, das eine Null oder mehrere Namens-/Wertpaare umgibt. Jedes JSON-Array ist dargestellt als ein Paar eckiger Klammern „[“ und „]“, das eine Null oder mehrere Objekte und/oder Werte umgibt.

Sowohl ein JSON-Objekt als auch ein JSON-Array sind strukturierte Daten. Der Unterschied besteht darin, dass es sich bei ersterem um eine ungeordnete Sammlung von Namens-/Wertpaaren handelt und beim letzterem um eine geordnete Sammlung von Objekten und/oder Werten.

Die Basisstruktur der Datendarstellung in JSON ist in [Bild 10](#) beschrieben. ISO/IEC 21778 empfiehlt, dass JSON-Namen innerhalb eines JSON-Objekts eindeutig sein sollten. In der Basisstruktur eines verbundenen Satzes von Datenpaketen wie in [Bild 8](#) beschrieben wird das Element „parcel“ im Element „conjunctiveParcels“ vervielfältigt, weshalb das Element „Parcel“ nicht mit einem JSON-Objekt dargestellt werden kann. Statt des Elements „parcel“ legt dieses Dokument einen speziellen JSON-Namen „parcels“ fest, der ein Array von JSON-Objekten enthält, von denen jedes ein Datenpaket angibt. Der JSON-Name ist in [7.2](#) definiert. Die übrigen Elemente in der Basisstruktur eines verbundenen Satzes von Datenpaketen sind mithilfe von JSON-Objekten dargestellt.

Der JSON-Name „classHeader“ im JSON-Name „header“ dient zur Einbindung der Informationen des Klassenkopfbereichs eines Datenpakets. Die Datendarstellung des Klassenkopfbereichs in JSON besteht aus JSON-Namens-/Wertpaaren, von denen jedes eine Anweisung und ihren Wert angibt. Die Spezifikation von JSON-Namen zur Beschreibung des Klassenkopfbereichs findet sich in [7.3](#).

Der JSON-Name „schemaHeader“ im JSON-Namen „header“ dient zur Einbindung einer Liste der Eigenschaftsinformationen eines Datenpakets, die das Schema für den Datenbereich sind. Die Spezifikation von JSON-Namen zur Beschreibung des Schemakopfbereichs findet sich in [7.3.9](#).

Der JSON-Name „data“ dient der Einbindung von Daten des Datenbereichs in einer vertikalen JSON-Notation oder einer lateralen JSON-Notation. Diese JSON-Notationen sind definiert in [7.5](#).

```
{
  "conjunctiveParcels": {
    "ontoLayer": <any of AO, MO, DO or DL>,
    "parcels": [
      {
        "header": {
          "classHeader": {
            <Data representation of the class header section in JSON>
          },
          "schemaHeader": [
            <Data representaiton of the schema header section in JSON>
          ]
        },
        "data": {
          <Data representation of the data section in JSON>
        }
      },
      {
        "header": {
          "classHeader": {
            <Data representation of the class header section in JSON>
          },
          "schemaHeader": [
            <Data representation of the schema header section in JSON>
          ]
        },
        "data": {
          <Data representation of the data section in JSON>
        }
      },
      ...
    ]
  }
}
```

Bild 10 – Basisstruktur der Datendarstellung in JSON

Das JSON-Schema zur Beschreibung von JSON-Daten in der vertikalen JSON-Notation findet sich in [B.1.1](#) und dasjenige zur Beschreibung von JSON-Daten in der lateralen JSON-Notation in [B.1.2](#).

7.2 Reservierter JSON-Name zur Angabe eines Arrays von Datenpaketen

JSON-Name: parcels
Bezeichnung: data parcel
Definition: reserviertes Schlüsselwort zur Beschreibung eines Arrays von Datenpaketen, von denen jedes als JSON-Objekt beschrieben ist
Auftreten: 1 für ein Auftreten des Schlüsselworts „conjunctiveParcels“

7.3 JSON-Namen für den Klassenkopfbereich

7.3.1 JSON-Name zur Angabe der Anweisung "#CLASS_ID"

JSON-Name: classID
Definition: reservierter JSON-Name zur Beschreibung eines Werts der Anweisung mit der Bezeichnung "#CLASS_ID" in einem Datenpaket
Datentyp: Zeichenkette
Verpflichtung: Verbindlich
Beispiel: "classID": "MDC_C002"

7.3.2 JSON-Name zur Angabe der Anweisung "#PARCEL_MODE"

JSON-Name: parcelMode
Definition: reservierter JSON-Name zur Beschreibung eines Werts der Anweisung mit der Bezeichnung "#PARCEL_MODE" in einem Datenpaket
Beschreibung: Wenn dieses Schlüsselwort vorhanden ist, kann sein Wert FULL, UPDATE oder PARTIAL lauten. Wird ein Datenpaket in einem Paketregistrierungsdienst verwendet, muss dieses Schlüsselwort vorhanden sein.
Datentyp: Zeichenkette
Verpflichtung: Optional
Beispiel: "parcelMode": "PARTIAL"

7.3.3 JSON-Name zur Angabe der Anweisung "#PARCEL_ID"

JSON-Name: parcelID
Definition: reservierter JSON-Name zur Beschreibung eines Werts der Anweisung mit der Bezeichnung "#PARCEL_ID" in einem Datenpaket
Datentyp: Zeichenkette
Verpflichtung: Optional
Beispiel: "parcelID": „2006-06-25 08:19:49“

7.3.4 JSON-Name zur Angabe der Anweisung "#DEFAULT_SUPPLIER"

JSON-Name: defaultSupplier
Definition: reservierter JSON-Name zur Beschreibung eines Werts der Anweisung mit der Bezeichnung "#DEFAULT_SUPPLIER" in einem Datenpaket
Datentyp: Zeichenkette
Verpflichtung: Optional
Beispiel: "defaultSupplier": „0112/2///62656_1“

7.3.5 JSON-Name zur Angabe der Anweisung "#DEFAULT_VERSION"

JSON-Name: defaultVersion
Definition: reservierter JSON-Name zur Beschreibung eines Werts der Anweisung mit der Bezeichnung "#DEFAULT_VERSION" in einem Datenpaket
Datentyp: Zeichenkette
Verpflichtung: Optional
Beispiel: "defaultVersion": „1“

7.3.6 JSON-Name zur Angabe der Anweisung „#OBJECT_ID_NAME“

JSON-Name: objectIdName
Definition: reservierter JSON-Name zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#OBJECT_ID_NAME“ in einem Datenpaket
Beschreibung: Wenn dieses Schlüsselwort vorhanden ist, kann sein Wert „GUID“ oder „UUID“ lauten.
Datentyp: Zeichenkette
Verpflichtung: Optional
Beispiel: „objectIdName“: „GUID“

7.3.7 JSON-Name zur Angabe der Anweisung „#ID_ENCODE“

JSON-Name: idEncode
Definition: reservierter JSON-Name zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#ID_ENCODE“ in einem Datenpaket
Datentyp: Zeichenkette
Verpflichtung: Optional
Beispiel: „idEncode“: „ICID“

7.3.8 JSON-Name zur Angabe der Anweisung „#PWS_CODIFICATION_MODE“

JSON-Name: pwsCodificationMode
Definition: reservierter JSON-Name zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#PWS_CODIFICATION_MODE“ in einem Datenpaket
Beschreibung: Wenn dieses Schlüsselwort vorhanden ist, muss sein Wert „VERTICAL“ oder „LATERAL“ lauten.
Datentyp: Zeichenkette
Verpflichtung: Verbindlich
Beispiel: „pwsCodificationMode“ „LATERAL“

7.3.9 JSON-Name zur Angabe der Anweisung „#INTENDED_LANGUAGE“

JSON-Name: intendedLanguage
Definition: reservierter JSON-Name zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#INTENDED_LANGUAGE“ in einem Datenpaket
Beschreibung: Ist mehr als eine Sprache angegeben, werden die einzelnen Sprache durch ein Komma getrennt „，“.
Datentyp: Zeichenkette
Verpflichtung: Optional
Beispiel: „intendedLanguage“: „en,fr“

7.4 JSON-Namen für den Schemakopfbereich

7.4.1 Basisstruktur der Datendarstellung für den Schemakopfbereich in JSON

Der JSON-Name „Schemakopf“ hat JSON-Objekte, von denen jeder Informationen einer Eigenschaft angibt, die als Schema einer Zellenspalte verwendet wird. Die Struktur der Datendarstellung für den Schemakopfbereich in JSON ist in Bild 11 beschrieben. Die Informationen einer in jedem JSON-Objekt enthaltenen Eigenschaft sind in 7.4.2 angegeben.

```
"schemaHeader": [  
  {  
    <information of a property as schema of a cell column>  
  },  
  {  
    <information of a property as schema of a cell column>  
  },  
  ...  
]
```

Bild 11 – Basisstruktur der Datendarstellung für den Schemakopfbereich in JSON

Ein Beispiel der Datendarstellung für den Schemakopfbereich in JSON findet sich in [D.2](#).

7.4.2 JSON-Namen für den Schemakopfbereich

7.4.2.1 JSON-Name zur Angabe eines Werts der Anweisung „#PROPERTY_ID“

JSON-Name: propertyID
Definition: reservierter JSON-Name zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#PROPERTY_ID“ in einem Datenpaket
Datentyp: Zeichenkette
Verpflichtung: Verbindlich
Beispiel: „propertyID“ „MDC_P001_5“

7.4.2.2 JSON-Name zur Angabe eines Werts der Anweisung „#DEFAULT_DATA_SUPPLIER“

JSON-Name: defaultDataSupplier

Definition: reservierter JSON-Name zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#DEFAULT_DATA_SUPPLIER“ in einem Datenpaket
Datentyp: Zeichenkette
Verpflichtung: Optional
Beispiel: "defaultDataSupplier": "0112/2///61360_4"

7.4.2.3 JSON-Name zur Angabe eines Werts der Anweisung „#DEFAULT_DATA_VERSION“

JSON-Name: defaultDataVersion
Definition: reservierter JSON-Name zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#DEFAULT_DATA_VERSION“ in einem Datenpaket
Datentyp: Zeichenkette
Verpflichtung: Optional
Beispiel: "defaultDataVersion": "1"

7.4.2.4 JSON-Name zur Angabe der Anweisung „#DEFAULT_VALUE“

JSON-Name: defaultValue
Definition: reservierter JSON-Name zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#DEFAULT_VALUE“ in einem Datenpaket
Datentyp: Zeichenkette
Verpflichtung: Optional
Beispiel: "defaultValue": "New York"

7.4.2.5 JSON-Name zur Angabe eines Werts der Anweisung „#REQUIREMENT“

JSON-Name: Requirement
Definition: reservierter JSON-Name zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#REQUIREMENT“ in einem Datenpaket
Beschreibung: Wenn dieses Schlüsselwort vorhanden ist, muss sein Wert entweder leer sein oder wie folgt lauten: „CONST“, „KEY“, „NOT_NULL“, „MANDATORY“, „OPTIONAL“ oder „OBSOLETE“. Ein leerer Wert entspricht der Bezeichnung „OPTIONAL“, während die Werte „MANDATORY“, „OPTIONAL“ und „OBSOLETE“ als „MAND“, „OPT“ und „OBS“ abgekürzt werden können. Die Details jedes Werts sind in [5.11.24](#) von [IEC 62656-1:2014](#) angegeben.
Datentyp: Zeichenkette
Verpflichtung: Optional
Beispiel: "requirement": "KEY"

7.5 Datendarstellung für den Datenbereich in JSON

7.5.1 Vertikale JSON-Darstellung für den Datenbereich

In der vertikalen JSON-Notation enthält der JSON-Name „data“ ein Paar aus dem JSON-Namen „values“ und einem JSON-Objekt als Wert, das eine ungeordnete Sammlung von JSON-Namens-/Wertpaaren enthält, von denen jedes einen Eigenschaftsbezeichner und seine Werte angibt. Der JSON-Name „data“ weist auch die optionalen JSON-Namen „defaultSupplier“ und „defaultVersion“ und ihre Werte auf. Wenn der JSON-Name „defaultSupplier“ und sein Wert im Datenbereich vorhanden sind, kann der RAI jedes Eigenschaftsbezeichners, der als ein JSON-Name im Wert des JSON-Namens „values“ verwendet wird, ausgelassen werden und muss dann um einen Datenempfänger erweitert werden. Wenn auf ähnliche Weise der JSON-Name „defaultVersion“ und sein Wert im Datenbereich vorhanden sind, kann der VI jedes Eigenschaftsbezeichners, der als ein JSON-Name im Wert des JSON-Namens „values“ verwendet wird, ausgelassen werden und muss dann um einen Datenempfänger erweitert werden.

Ist ein Wert leer, muss „null“ als ein JSON-Wert in einem JSON-Array beschrieben werden.

Ein Beispiel für die vertikale JSON-Notation des Datenbereichs findet sich in [D.2.1](#).

7.5.2 Laterale JSON-Darstellung für den Datenbereich

In der lateralen JSON-Notation enthält der JSON-Name „data“ ein Paar aus dem JSON-Namen „instances“ und einem JSON-Array als Wert, wobei jedes Element ein JSON-Array von Eigenschaftswerten einer Instanz darstellt. Die Reihenfolge und Anzahl der Elemente jedes JSON-Arrays von Eigenschaftswerten muss der Reihenfolge und Anzahl der Eigenschaften im Schemakopfbereich entsprechen.

Ein Beispiel der lateralen JSON-Notation des Datenbereichs findet sich in [D.2.2](#).

7.6 Zeichencodierung

ISO/IEC 21778 empfiehlt die Verwendung von UTF-8, UTF-16 oder UTF-32 als Zeichencodierung. In Übereinstimmung mit diesen Spezifikationen sollten alle im Paketwebdienst ausgetauschten Textdaten in JSON in UTF-8, UTF-16 oder UTF-32 codiert sein.

8 Datendarstellung in XML

8.1 Basisstruktur der Datendarstellung in XML

In Übereinstimmung mit der vom World Wide Web Consortium (W3C) bereitgestellten XML-Spezifikation gibt es zwei Möglichkeiten zur Beschreibung eines Wertes, eine ist das XML-Element und die andere das XML-Attribut. Es gibt keine Regel oder Empfehlung in der Spezifikation, welche angewendet werden sollte, aber ein XML-Element eignet sich für die Darstellung strukturierter Daten. Daher spezifiziert dieses Dokument als Grundlage der Nutzung der zwei Verfahren die Verwendung eines XML-Elements zur Beschreibung der Elemente in der Basisstruktur eines verbundenen Satzes von Datenpaketen, wie in [Bild 2](#) beschrieben. Ein XML-Attribut wird zur Darstellung eines ergänzenden Wertes für ein Element verwendet.

Die Basisstruktur der Datendarstellung in XML ist in [Bild 12](#) beschrieben. In der Basisstruktur eines verbundenen Satzes von Datenpaketen wie in [Bild 8](#) beschrieben werden alle Elemente mithilfe von XML-Elementen dargestellt. Eine als Wert von „ontoLayer“ beschriebene Paketontologieschicht ist eine ergänzende Information für einen verbundenen Satz von Paketen und wird daher als ein XML-Attribut dargestellt.

Das XML-Element „classHeader“ im XML-Element „header“ dient zur Einbindung der Informationen des Klassenkopfbereichs eines Datenpakets. Das XML-Element des Klassenkopfbereichs besteht aus XML-Elementen, von denen jedes eine Anweisung und ihren Wert angibt. Die Spezifikation der Datendarstellung für den Klassenkopfbereich in XML ist in [8.3](#) angegeben.

Das XML-Element „schemaHeader“ im XML-Element „header“ dient zur Einbindung einer Liste der Eigenschaftsinformationen eines Datenpakets, die das Schema für den Datenbereich sind. Die Spezifikation der Datendarstellung für den Schemakopfbereich in XML ist in [8.3.9](#) angegeben.

Das XML-Element „data“ dient der Einbindung von Daten des Datenbereichs in einer vertikalen XML-Notation oder einer lateralen XML-Notation. Diese XML-Notationen sind definiert in [8.5](#).

```
<conjunctiveParcels ontLayer="<any of AO, MO, DO or DL>">
  <parcel>
    <header>
      <classHeader>
        <Data representation of class header section in XML>
      </classHeader>
      <schemaHeader>
        <Data representation of schema header section in XML>
      </schemaHeader>
    </header>
    <data>
      <Data representation of data section in XML>
    </data>
  </parcel>
</conjunctiveParcels>
```

Bild 12 – Basisstruktur der Datendarstellung in XML

Das XML-Schema zur Darstellung von Daten in der vertikalen XML-Notation findet sich in [B.2.1](#) und dasjenige zur Darstellung von Daten in der lateralen XML-Notation in [B.2.2](#).

8.2 Reserviertes Schlüsselwort zur Angabe eines Datenpakets

Schlüsselwort: parcel
Bezeichnung: data parcel
Definition: reserviertes XML-Element zur Beschreibung eines Datenpakets
Auftreten: 1..n für ein Auftreten des Schlüsselworts „conjunctiveParcels“

8.3 XML-Elemente für den Klassenkopfbereich

8.3.1 XML-Element zur Angabe der Anweisung „#CLASS_ID“

Schlüsselwort: classID
Definition: reserviertes XML-Element zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#CLASS_ID“ in einem Datenpaket
Datentyp: Zeichenkette
Verpflichtung: Verbindlich
Beispiel: <KlassenID>MDC_C002</KlassenID>

8.3.2 XML-Element zur Angabe der Anweisung „#PARCEL_MODE“

Schlüsselwort: parcelMode
Definition: reserviertes XML-Element zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#PARCEL_MODE“ in einem Datenpaket
Beschreibung: Wenn dieses XML-Element vorhanden ist, kann sein Wert VOLL, AKTUALISIEREN oder PARTIELL lauten. Wird ein Datenpaket in einem Paketregistrierungsdienst verwendet, muss dieses Element vorhanden sein.
Datentyp: Zeichenkette
Verpflichtung: Optional
Beispiel: <parcelMode>PARTIAL</parcelMode>

8.3.3 XML-Element zur Angabe der Anweisung „#PARCEL_ID“

Schlüsselwort: parcelID
Definition: reserviertes XML-Element zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#PARCEL_ID“ in einem Datenpaket
Datentyp: Zeichenkette
Verpflichtung: Optional
Beispiel: <parcelID>2006-06-25 08:19:49</parcelID>

8.3.4 XML-Element zur Angabe der Anweisung „#DEFAULT_SUPPLIER“

Schlüsselwort: defaultSupplier

Definition: reserviertes XML-Element zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#DEFAULT_SUPPLIER“ in einem Datenpaket

Datentyp: Zeichenkette

Verpflichtung: Optional

Beispiel: <defaultSupplier>0112/2///62656_1</defaultSupplier>

8.3.5 XML-Element zur Angabe der Anweisung „#DEFAULT_VERSION“

Schlüsselwort: defaultVersion

Definition: reserviertes XML-Element zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#DEFAULT_VERSION“ in einem Datenpaket

Datentyp: Zeichenkette

Verpflichtung: Optional

Beispiel: <defaultVersion>1</defaultVersion>

8.3.6 XML-Element zur Angabe der Anweisung „#OBJECT_ID_NAME“

Schlüsselwort: objectIdName

Definition: reserviertes XML-Element zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#OBJECT_ID_NAME“ in einem Datenpaket

Beschreibung: Wenn dieses Element vorhanden ist, kann sein Wert „GUID“ oder „UUID“ lauten.

Datentyp: Zeichenkette

Verpflichtung: Optional

Beispiel: <objectIdName>GUID</objectIdName>

8.3.7 XML-Element zur Angabe der Anweisung „#ID_ENCODE“

Schlüsselwort: idEncode

Definition: reserviertes XML-Element zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#ID_ENCODE“ in einem Datenpaket

Datentyp: Zeichenkette

Verpflichtung: Optional

Beispiel: <idEncode>ICID</idEncode>

8.3.8 XML-Element zur Angabe der Anweisung „#PWS_CODIFICATION_MODE“

Schlüsselwort: pwsCodificationMode

Definition: reserviertes XML-Element zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#PWS_CODIFICATION_MODE“ in einem Datenpaket

Beschreibung: Wenn dieses Element vorhanden ist, muss sein Wert „VERTICAL“ oder „LATERAL“ lauten.

Datentyp: Zeichenkette

Verpflichtung: Verbindlich

Beispiel: <pwsCodificationMode>LATERAL</pwsCodificationMode>

8.3.9 XML-Element zur Angabe der Anweisung „#INTENDED_LANGUAGE“

Schlüsselwort: intendedLanguage

Definition: reserviertes XML-Element zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#INTENDED_LANGUAGE“ in einem Datenpaket

Beschreibung: Ist mehr als eine Sprache angegeben, werden die einzelnen Sprache durch ein Komma getrennt „，“.

Datentyp: Zeichenkette

Verpflichtung: Optional

Beispiel: <intendedLanguage>en,fr</intendedLanguage>

8.4 XML-Elemente für den Schemakopfbereich

8.4.1 Basisstruktur der Datendarstellung für den Schemakopfbereich in XML

Das XML-Element „schemaHeader“ hat XML-Elemente „property“, von denen jedes Informationen einer Eigenschaft angibt, die als Schema einer Zellenspalte verwendet wird. Die Struktur der Datendarstellung für den Schemakopfbereich in XML ist in Bild 13 beschrieben. Die Informationen einer in jedem XML-Element „property“ enthaltenen Eigenschaft sind in 8.4.2 angegeben.

```
<schemaHeader>
  <property>
    <information of a property as schema of a cell column>
  </property>
  <property>
    <information of a property as schema of a cell column>
  </property>
  ...
</schemaHeader>
```

Bild 13 – Basisstruktur der Datendarstellung für den Schemakopfbereich in XML

8.4.2 XML-Elemente des Schemakopfbereichs

8.4.2.1 XML-Element zur Angabe der Anweisung „#PROPERTY_ID“

Schlüsselwort: propertyID
Definition: reserviertes XML-Element zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#PROPERTY_ID“ in einem Datenpaket
Datentyp: Zeichenkette
Verpflichtung: Verbindlich
Beispiel: <propertyID>MDC_P001_5</propertyID>

8.4.2.2 XML-Element zur Angabe der Anweisung „#DEFAULT_DATA_SUPPLIER“

Schlüsselwort: defaultDataSupplier
Definition: reserviertes XML-Element zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#DEFAULT_DATA_SUPPLIER“ in einem Datenpaket
Datentyp: Zeichenkette
Verpflichtung: Optional
Beispiel: <defaultDataSupplier>MDC_P001_5</defaultDataSupplier>

8.4.2.3 XML-Element zur Angabe der Anweisung „#DEFAULT_DATA_VERSION“

Schlüsselwort: defaultDataVersion
Definition: reserviertes XML-Element zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#DEFAULT_DATA_VERSION“ in einem Datenpaket
Datentyp: Zeichenkette
Verpflichtung: Optional
Beispiel: <defaultDataVersion>1</defaultDataVersion>

8.4.2.4 XML-Element zur Angabe der Anweisung „#DEFAULT_VALUE“

Schlüsselwort: defaultValue
Definition: reserviertes XML-Element zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#DEFAULT_VALUE“ in einem Datenpaket
Datentyp: Zeichenkette
Verpflichtung: Optional
Beispiel: <defaultValue>New York</defaultValue>

8.4.2.5 XML-Element zur Angabe der Anweisung „#REQUIREMENT“

Schlüsselwort: requirement

Definition: reserviertes XML-Element zur Beschreibung eines Werts der Anweisung mit der Bezeichnung „#REQUIREMENT“ in einem Datenpaket

Beschreibung: Wenn dieses Schlüsselwort vorhanden ist, muss sein Wert entweder leer sein oder wie folgt lauten: „CONST“, „KEY“, „NOT_NULL“, „MANDATORY“, „OPTIONAL“ oder „OBSOLETE“. Ein leerer Wert entspricht der Bezeichnung „OPTIONAL“, während die Werte „MANDATORY“, „OPTIONAL“ und „OBSOLETE“ als „MAND“, „OPT“ und „OBS“ abgekürzt werden können. Die Details jedes Werts sind in 5.11.24 von IEC 62656-1:2014 angegeben.

Datentyp: Zeichenkette

Verpflichtung: Optional

Beispiel: <requirement>KEY</requirement>

8.5 XML-Elemente und Attribut für Datenbereich

8.5.1 Vertikale XML-Darstellung des Datenbereichs

8.5.1.1 Basisstruktur der Datendarstellung für den Datenbereich in vertikaler XML-Notation

In der vertikalen XML-Notation enthält das XML-Element „data“ XML-Elemente „values“, die jeweils mit ihrem Attribut „propertyID“ identifiziert werden, wobei der Eigenschaftsbezeichner angegeben ist. Jedes XML-Element „values“ enthält auch eine geordnete Sammlung von XML-Elementen „value“. Jedes XML-Element „value“ enthält einen Eigenschaftswert.

Das XML-Element „data“ hat die XML-Attribute „defaultSupplier“ und „defaultVersion“. Wenn das XML-Attribut „defaultSupplier“ im Datenbereich vorhanden ist, kann der RAI jedes Eigenschaftsbezeichners, der im XML-Attribut „propertyID“ des XML-Elements „values“ angegeben ist, ausgelassen werden und muss dann um einen Datenempfänger erweitert werden. Wenn auf ähnliche Weise das XML-Attribut „defaultVersion“ im Datenbereich vorhanden ist, kann der VI jedes im XML-Attribut „propertyID“ des XML-Elements „values“ spezifizierten Eigenschaftsbezeichners ausgelassen werden und muss dann um einen Datenempfänger erweitert werden.

Demzufolge ist die Datendarstellung für den Datenbereich in der vertikalen XML-Notation strukturiert wie in Bild 14 beschrieben.

```
<data defaultSupplier="<RA/>" defaultVersion="<VI/>">
  <values propertyID="<property ID>">
    <a value of the specified property in the vertical XML notation>
    <a value of the specified property in the vertical XML notation>
    ...
  </values>
  <values propertyID="<property ID>">
    <a value of the specified property in the vertical XML notation>
    <a value of the specified property in the vertical XML notation>
    ...
  </values>
  ...
</data>
```

Bild 14 – Struktur der Datendarstellung für den Datenbereich in der vertikalen XML-Notation

Ein Beispiel der Datendarstellung in der vertikalen XML-Notation findet sich in D.3.1.

8.5.1.2 XML-Element zur Angabe von Werten einer Eigenschaft in vertikaler Codierung

Schlüsselwort: values

Definition: reserviertes XML-Element im vertikalen Codierungsmodus zur Beschreibung einer

Liste von Werten einer Eigenschaft

Beschreibung: Der Bezeichner einer Eigenschaft ist im Attribut mit der Bezeichnung „propertyID“ dieses Elements definiert.

Verpflichtung: Verbindlich

Beispiel:

```
<values propertyID="MDC_P001_5">
  <value>AAA001</value>
  <value>AAA002</value>
  <value>AAA003</value>
  <value>AAA013</value>
</values>
```

8.5.1.3 XML-Attribut zur Angabe eines Eigenschaftsbezeichners für einen Wert in vertikaler Codierung

Schlüsselwort: propertyID

Definition: reserviertes XML-Attribut des XML-Elements mit der Bezeichnung „values“ im vertikalen Codierungsmodus, zur Beschreibung des Bezeichners einer Eigenschaft, deren Werte im Element aufgeführt sind.

Beschreibung: Wird der RAI des Bezeichners ausgelassen, muss er um den Wert des Attributs mit dem Namen „defaultSupplier“ des Elements mit der Bezeichnung „data“ erweitert werden. Wird der VI des Bezeichners ausgelassen, muss er um den Wert des Attributs mit dem Namen „defaultVersion“ des Elements mit der Bezeichnung „data“ erweitert werden.

Datentyp: Zeichenkette

Verpflichtung: Verbindlich

Beispiel:

```
<values propertyID="MDC_P001_5">
  <value>AAA001</value>
  <value>AAA002</value>
  <value>AAA003</value>
  <value>AAA013</value>
</values>
```

XML-Element zur Angabe eines Eigenschaftswerts in vertikaler Codierung

Schlüsselwort: value

Definition: reserviertes XML-Element im vertikalen Codierungsmodus zur Beschreibung eines Werts der Eigenschaft, die im Attribut mit der Bezeichnung „propertyID“ des Elternelements mit dem Name „values“ dieses Elements definiert ist

Beschreibung: Ist ein Wert leer oder null, sollte das Element als "<value></value>" oder "<value/>" beschrieben werden.

Datentyp: Beliebige

Verpflichtung: Verbindlich

Beispiel:

```
<value>component</value>
```

Laterale XML-Darstellung des Datenbereichs

Basisstruktur der Datendarstellung für den Datenbereich in lateraler XML-Notation

In der lateralen XML-Notation enthält das XML-Element „data“ XML-Elemente „instance“, von denen jedes XML-Elemente „value“ enthält. Jedes XML-Element „value“ enthält einen Eigenschaftswert und hat ein XML-Attribut „propertyID“, das den Eigenschaftsbezeichner für den Wert angibt.

Auf gleiche Weise wie bei der vertikalen XML-Notation für den Datenbereich verfügt das XML-Element „data“ über XML-Attribute „defaultSupplier“ und „defaultVersion“. Wenn das XML-Attribut „defaultSupplier“ im Datenbereich vorhanden ist, kann der RAI des Eigenschaftsbezeichners, der im XML-Attribut „propertyID“ jedes XML-Elements „value“ angegeben ist, ausgelassen werden und muss dann um einen Datenempfänger erweitert werden. Wenn auf ähnliche Weise das XML-Attribut „defaultVersion“ im Datenbereich vorhanden sind,

kann der VI des Eigenschaftsbezeichners, der im XML-Attribut „propertyID“ jedes XML-Elements „value“ angegeben ist, ausgelassen werden und muss dann um einen Datenempfänger erweitert werden.

Demzufolge ist die Datendarstellung für den Datenbereich in der lateralen XML-Notation strukturiert wie in Bild 15 beschrieben.

```
<data defaultSupplier="<RA/>" defaultVersion="<VI/>">
  <instance>
    <a value of an instance in the lateral XML notation>
    <a value of an instance in the lateral XML notation>
    ...
  </instance>
  <instance>
    <a value of an instance in the lateral XML notation>
    <a value of an instance in the lateral XML notation>
    ...
  </instance>
  ...
</data>
```

Bild 15 – Struktur der Datendarstellung für den Datenbereich in lateraler XML-Notation

Ein Beispiel der Datendarstellung in der lateralen XML-Notation findet sich in [D.3.2](#).

8.5.2.2 XML-Element zur Angabe einer Instanz in lateraler Codierung

Schlüsselwort: instance

Definition: reserviertes XML-Element im lateralen Codierungsmodus zur Beschreibung von Eigenschafts-Wertpaaren einer Instanz

Beschreibung: Jedes Eigenschafts-Wertpaar ist in einem Kindelement mit der Bezeichnung „value“ dieses Elements beschrieben.

Verpflichtung: Verbindlich

Beispiel:

```
<instance>
  <value propertyID="MDC_P001_5">AAA001</value>
  <value propertyID="MDC_P004_1.en">component</value>
  <value propertyID="MDC_P010">UNIVERSE</value>
  <value propertyID="MDC_P014">{AAE001,AAE006}</value>
</Instanz>
```

8.5.2.3 XML-Element zur Angabe eines Eigenschaftswerts in lateraler Codierung

Schlüsselwort: value

Definition: reserviertes XML-Element im lateralen Codierungsmodus zur Beschreibung eines Eigenschaftswertpaars

Beschreibung: Ein Eigenschaftsbezeichner wird im Attribut mit der Bezeichnung „propertyID“ dieses Elements beschrieben, während sein Wert als Wert dieses Elements beschrieben wird. Ist ein Wert leer oder null, sollte das Element als „<value></value>“ oder „<value/>“ beschrieben werden.

Datentyp: Beliebige

Verpflichtung: Verbindlich

Beispiel:

```
<value propertyID="MDC_P004_1.en">component</value>
```

8.5.2.4 XML-Attribut zur Angabe eines Eigenschaftsbezeichners für einen Wert in lateraler Codierung

Schlüsselwort: propertyID

Definition: reserviertes XML-Attribut des XML-Elements mit der Bezeichnung „value“ im lateralen Codierungsmodus zur Beschreibung eines Eigenschaftsbezeichners