

DIN EN IEC 62714-5

ICS 35.240.50

Einsprüche bis 2021-01-06

Entwurf

**Datenaustauschformat für Planungsdaten industrieller
Automatisierungssysteme –
Automation markup language –
Teil 5: Kommunikation
(IEC 65E/729/CD:2020);
Text Deutsch und Englisch**

Engineering data exchange format for use in industrial automation systems engineering –
Automation markup language –
Part 5: Communication
(IEC 65E/729/CD:2020);
Text in German and English

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2020-11-06 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 oder 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 106 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE

Nationales Vorwort

Das internationale Dokument IEC 65E/729/CD:2020-05 „Engineering data exchange format for use in industrial automation systems engineering – Automation markup language – Part 5: Communication“ (CD, en: Committee Draft) ist unverändert in diesen Norm-Entwurf übernommen worden. Dieser Norm-Entwurf enthält eine noch nicht autorisierte deutsche Übersetzung.

Um Zweifelsfälle in der Übersetzung auszuschließen, ist die englische Originalfassung des CD 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 IEC/SC 65E „Devices and integration in enterprise systems“ der Internationalen Elektrotechnischen Kommission (IEC) erarbeitet und unterliegt dem Copyright der IEC . Das Dokument wurde den nationalen Komitees zur Stellungnahme vorgelegt.

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. Dokumente, die bei CENELEC als Europäische Norm angenommen und ratifiziert werden, sind unverändert als Deutsche Normen zu übernehmen.

Da der Abstimmungszeitraum für einen FDIS bzw. Schluss-Entwurf FprEN nur 6 Wochen beträgt, und dann keine sachlichen Stellungnahmen mehr abgegeben werden können, sondern nur noch eine „JA/NEIN“-Entscheidung möglich ist, wobei eine „NEIN“-Entscheidung fundiert begründet werden muss, wird bereits der CD als DIN-Norm-Entwurf veröffentlicht, um die Stellungnahmen aus der Öffentlichkeit frühzeitig berücksichtigen zu können.

Für dieses Dokument ist das nationale Arbeitsgremium DKE/K 941 „Engineering“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE (www.dke.de) zuständig.

Aktuelle Informationen zu diesem Dokument können über die Internetseiten von DKE (www.dke.de) und DIN (www.din.de) durch eine Suche nach der Dokumentennummer aufgerufen werden.

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf ein Dokument ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnitsnummer, 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.

Inhalt

	Seite
1 Anwendungsbereich.....	7
2 Normative Verweisungen.....	7
3 Begriffe, Abkürzungen und Akronyme.....	8
3.1 Begriffe.....	8
3.2 Abkürzungen und Akronyme.....	8
4 Anwendungsfälle und Netzwerkstrukturen.....	9
4.1 Allgemeines.....	9
4.2 Anwendungsfälle.....	9
4.2.1 Verlustfreie Übertragung der Instanzdaten von Kommunikationsgeräten.....	10
4.2.2 Verlustfreie Übertragung von Kommunikationssystemdaten.....	12
4.3 Abgrenzung des Modellierungsbereiches.....	14
4.3.1 Interaktionsstrukturen und Lebenszyklen.....	14
4.3.2 Netzwerkobjekte.....	15
4.3.3 Netzwerktopologien.....	17
4.3.4 Inhalt der Kommunikation.....	21
4.4 Abgeleitete Modellierungsanforderungen.....	22
5 UML-Modell.....	23
5.1 Überblick.....	23
5.2 Logische Topologie.....	23
5.2.1 Element logicalTopology.....	24
5.2.2 Element logicalConnection.....	24
5.2.3 Element logicalEndPoint.....	24
5.3 Physikalische Topologie.....	25
5.3.1 Element physicalTopology.....	25
5.3.2 Element physicalConnection.....	25
5.3.3 Element physicalEndPoint.....	26
5.4 Gerät.....	26
5.4.1 Allgemeines.....	26
5.4.2 Element physicalDevice.....	26
5.4.3 Element Information.....	27
5.4.4 Element physicalDeviceInformation.....	28
5.4.5 Element logicalDeviceInformation.....	28
5.4.6 Element variable.....	28
5.4.7 Element physicalChannelList.....	28
5.4.8 Element physicalChannel.....	28
5.4.9 Element logicalDevice.....	28
5.4.10 Element networkDataList.....	29
5.4.11 Element networkDataItem.....	29

	Seite
5.4.12 Element pduList.....	29
5.4.13 Element pdu.....	29
5.4.14 Element protocolData.....	29
5.4.15 Element payload.....	29
5.4.16 Element processDataItem.....	29
5.4.17 Element parameterItem.....	29
5.4.18 Element dataItem.....	29
5.4.19 Element processDataItem.....	30
5.4.20 Element logicalEndPointList.....	30
5.4.21 Element deviceResource.....	30
5.4.22 Element variableList.....	30
5.4.23 Element processDataInput.....	31
5.4.24 Element processDataOutput.....	31
5.4.25 Element parameterItem.....	32
6 Darstellung in AutomationML.....	32
6.1 Überblick über die Zuordnung.....	32
6.1.1 Allgemeine Zuordnungsregeln.....	32
6.1.2 Grundlagen.....	33
6.1.3 Modellierung von Beziehungen.....	34
6.2 Basis-Kommunikationsrollenklassenbibliothek.....	37
6.2.1 Allgemeines.....	37
6.2.2 RoleClass PhysicalDevice.....	38
6.2.3 RoleClass PhysicalEndpointList.....	39
6.2.4 RoleClass PhysicalConnection.....	39
6.2.5 RoleClass PhysicalNetwork.....	39
6.2.6 RoleClass LogicalDevice.....	40
6.2.7 RoleClass LogicalEndpointList.....	40
6.2.8 RoleClass LogicalConnection.....	40
6.2.9 RoleClass LogicalNetwork.....	40
6.3 Basis-Kommunikationsschnittstellenklassenbibliothek.....	41
6.3.1 Allgemeines.....	41
6.3.2 InterfaceClass PhysicalEndPoint.....	41
6.3.3 InterfaceClass LogicalEndPoint.....	42
6.4 Schritte zum Modellieren technologiespezifischer Bibliotheken.....	42
6.4.1 Allgemeines.....	42
6.4.2 Schritt 1: Entwicklung technologiespezifischer Rollenklassen.....	42
6.4.3 Schritt 2: Entwicklung technologiespezifischer Schnittstellenklassen.....	43
6.4.4 Schritt 3: Entwicklung von Systemunitklassenbibliotheken.....	43
6.4.5 Schritt 4: Modellierung des Netzwerks.....	44

	Seite
6.4.6 Schritt 5: Modellierung der Verbindungen.....	44
6.5 PDU-Modellierung.....	45
6.5.1 Allgemeines.....	45
6.5.2 RoleClass CommunicationPackage.....	45
6.5.3 InterfaceClass DatagramObject.....	47
6.5.4 Schritte zum Modellieren technologiespezifischer Bibliotheken.....	48
6.6 Verweisungen auf Attribute.....	50
6.7 Verwendung von Metadaten.....	55
Literaturhinweise	56

Bilder

	Seite
Bild 1 – Allgemeine Planungsvorgänge, in denen die Kommunikationssystemplanung integriert ist.....	9
Bild 2 – Datenfluss im Anwendungsfall.....	11
Bild 3 – Alternativer Datenfluss im Anwendungsfall.....	12
Bild 4 – Datenfluss im Anwendungsfall.....	13
Bild 5 – Beispielhafte Ansicht einer logischen Ebene von Kommunikationssystemen.....	15
Bild 6 – beispielhafte Ansicht einer physikalischen Ebene von Kommunikationssystemen.....	16
Bild 7 – Kombinierte Sicht auf Kommunikationssysteme.....	17
Bild 8 – Beispiel für Sterntopologie.....	17
Bild 9 – Beispiel für Ringtopologie.....	18
Bild 10 – Beispiel für Linientopologie.....	18
Bild 11 – Einfaches Netzwerk mit direkter Verdrahtung.....	19
Bild 12 – Netzwerk mit aktiver Infrastruktur.....	19
Bild 13 – Durch Gateways verbundene Netzwerke.....	20
Bild 14 – Hierarchisch strukturierte Netzwerke.....	20
Bild 15 – Netzwerk für mehrere Anwendungen.....	21
Bild 16 – Allgemeine Modellierungsstrategie für PDUs.....	22
Bild 17 – Struktur des Kommunikationsnetzes.....	23
Bild 18 – Sicht auf die logische Topologie.....	24
Bild 19 – Sicht auf die physikalische Topologie.....	25
Bild 20 – Teil 1 des Gerätemodells.....	27
Bild 21 – Teil 2 des Gerätemodells.....	31
Bild 22 – Kommunikation-Rollenklassenbibliothek und Kommunikations-Schnittstellenklassenbibliothek.....	33
Bild 23 – für ein besonderes Beispiel abgeleitete Rollenklassenbibliotheken und Schnittstellenklassenbibliotheken.....	34
Bild 24 – Beispiele für SystemUnitClassLib für die Kommunikationssystemmodellierung.....	36
Bild 25 – Beispiel für finales Netzwerkmodell.....	37
Bild 26 – Basis-Kommunikationsrollenklassenbibliothek.....	38
Bild 27 – CommunicationRoleClassLib.....	38

	Seite
Bild 28 – XML-Text der Kommunikations-Rollenklassenbibliothek.....	38
Bild 29 – Basis-Kommunikationsschnittstellenklassenbibliothek.....	41
Bild 30 – CommunicationInterfaceClassLib.....	41
Bild 31 – XML-Text der Kommunikationsschnittstellenklassenbibliothek.....	41
Bild 32 – Ableitung einer technologiespezifischen Rollenklassenbibliothek von der Basis-Rollenklassenbibliothek.....	42
Bild 33 – Ableitung einer technologiespezifischen Rollenklassenbibliothek von der Basis-Rollenklassenbibliothek.....	43
Bild 34 – Technologiespezifische <SystemUnitClassLib>s.....	44
Bild 35 – Technologiespezifisches Kommunikationsnetz.....	45
Bild 36 – Erweiterte Basis-Kommunikationsrollenklassenbibliothek.....	46
Bild 37 – Erweiterte CommunicationRoleClassLib.....	46
Bild 38 – XML-Text der erweiterten Kommunikations-Rollenklassenbibliothek.....	46
Bild 39 – Erweiterte Kommunikationsschnittstellenklassenbibliothek.....	47
Bild 40 – Erweiterte CommunicationInterfaceClassLib.....	47
Bild 41 – XML-Text der erweiterten Kommunikations-Rollenklassenbibliothek.....	47
Bild 42 – Ableitung einer technologiespezifischen Rollenklassenbibliothek von der erweiterten Rollenklassenbibliothek.....	48
Bild 43 – Ableitung einer technologiespezifischen Schnittstellenklassenbibliothek von der erweiterten Schnittstellenklassenbibliothek.....	49
Bild 44 – Technologiespezifische erweiterte <SystemUnitClassLib>s.....	49
Bild 45 – Technologiespezifisches Kommunikationsnetz mit Kommunikationspaketmodellen.....	50

Tabellen

	Seite
Tabelle 1 – Zuordnungsregeln.....	32
Tabelle 2 – Modellierung von Beziehungen in AutomationML.....	34
Tabelle 3 – RoleClass PhysicalDevice.....	38
Tabelle 4 – RoleClass PhysicalEndpointlist.....	39
Tabelle 5 – RoleClass PhysicalConnection.....	39
Tabelle 6 – RoleClass PhysicalNetwork.....	39
Tabelle 7 – RoleClass LogicalDevice.....	40
Tabelle 8 – RoleClass LogicalEndpointlist.....	40
Tabelle 9 – RoleClass LogicalConnection.....	40
Tabelle 10 – RoleClass LogicalNetwork.....	40
Tabelle 11 – InterfaceClass PhysicalEndPoint.....	41
Tabelle 12 – InterfaceClass LogicalEndPoint.....	42
Tabelle 13 – RoleClass CommunicationPackage.....	47
Tabelle 14 – InterfaceClass DatagrammObject.....	48
Tabelle 15 – Kommunikationsbezogene Attribute.....	51
Tabelle 16 – Feld SourceDocumentInformation nach kommunikationsbezogenen Bibliotheken.....	55

Datenaustauschformat für Planungsdaten industrieller Automatisierungssysteme – Automation markup language – Teil 5: Kommunikation

1 Anwendungsbereich

Die Planungsprozesse technischer Systeme und ihrer integrierten Automatisierungssysteme müssen mit zunehmender Effizienz und Qualität ausgeführt werden. Mit zunehmender Komplexität des geplanten Systems nimmt in den meisten Fällen auch die Projektlaufzeit zu. Um diesem Problem zu begegnen, werden im Planungsprozess vermehrt softwarebasierte technische Arbeitsmittel zum Austausch von Planungsdaten und Artefakten eingesetzt.

Ein integraler Teil moderner technischer Systeme und insbesondere der darin integrierten Automatisierungssysteme sind Kommunikationssysteme. Im Rahmen der zunehmenden Dezentralisierung von Automatisierungssystemen und des vermehrten Einsatzes von Feldbus- und Ethernettechnologien verbinden sie Automatisierungsgeräte und weitere interagierende Einheiten und müssen besondere Anforderungen hinsichtlich Qualität, Sicherheit und Schutz erfüllen. Daher müssen im Rahmen des Planungsprozesses moderner technischer Systeme ebenso Planungsdaten und Artefakte im Zusammenhang mit Kommunikationssystemen ausgetauscht werden.

In jeder Phase des Planungsprozesses technischer Systeme können kommunikationssystembezogene Daten erzeugt werden, die in späteren Planungsphasen verwendet werden können. Ein typischer Anwendungsfall ist die Erstellung der Konfigurationsdaten von Kommunikationskomponenten von Automatisierungsgeräten einschließlich Kommunikationsadressen und die Strukturierung von Kommunikationspaketen in Geräten zur PLC-Programmierung während der Steuerungsprogrammierung und ihre Verwendung in einem Werkzeug zur Bauteilkonfiguration. Eine weitere typische Anwendung ist die Übertragung der Konfigurationen von Kommunikationsgeräten an Werkzeuge zur virtuellen Inbetriebnahme, zur Dokumentation oder zur Diagnose.

Jedoch ist die einheitliche und verlustfreie Übertragung der Planungsdaten von Kommunikationssystemen im Planungsprozess von technischen Systemen eine bis heute ungelöste Aufgabe. Zwar wurden von Nutzerorganisationen und Anwenderunternehmen für Teile der relevanten Daten Austauschformate wie FDCML, EDDL oder GSD entwickelt. Diese können die zuvor angeführten Anwendungsbeispiele jedoch nicht abdecken. Insbesondere die Daten mit Bezug zur Netztechnologie, mit denen Kommunikationsbezüge, deren Eigenschaften und Qualitäten beschrieben werden, können nicht durch ein Datenaustauschformat modelliert werden.

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).

IEC 60027 (all parts), *Letter symbols to be used in electrical technology*

IEC 60050, *International Electrotechnical Vocabulary*

IEC 62424:2008, *Representation of process control engineering - Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools*

ISO/IEC 9834-8, *Information technology - Open Systems Interconnection - Procedures for the operation of OSI Registration Authorities: Generation and registration of Universally Unique Identifiers (UUIDs) and their use as ASN.1 Object Identifier components*

IEC 62714-1, *Engineering data exchange format for use in industrial systems engineering – AutomationML, Architecture and general requirements*

E DIN EN IEC 62714-5:2020-12

IEC 62714-2, *Engineering data exchange format for use in industrial systems engineering – AutomationML, Role Libraries*

IEC 62714-4, *Engineering data exchange format for use in industrial systems engineering – AutomationML, Logic*

ISO 80000-1, *Quantities and units – Part 1: General*

Extensible Markup Language (XML) 1.0:2004, W3C Recommendation (verfügbar unter: <http://www.w3.org/TR/2004/REC-xml-20040204/>)

IEC 61158, *Industrial communication networks – Fieldbus specifications*

IEC 61784, *Industrial communication networks – Profiles*

IEC 61346, *Industrial systems, Installations and Equipment and Industrial Products — Structuring Principles and Reference Designations*

IEC 61131-10, *Programmable controllers – Part 10: PLC open XML exchange format, 2019*

3 Begriffe, Abkürzungen und Akronyme

3.1 Begriffe

Für die Anwendung dieses Dokuments gelten die Begriffe nach IEC 62714-1 und die folgenden Begriffe.

3.1.1

AutomationML/AML

XML-basiertes Datenaustauschformat für Anlagenplanungsdaten

3.1.2

Automatisierungsobjekt

Einheit in einem automatisierten System

Anmerkung 1 zum Begriff: Automatisierungsobjekte sind z. B. eine Automatisierungskomponente, ein Ventil und/oder ein Signal.

3.2 Abkürzungen und Akronyme

AML	Automation Markup Language
AutomationML	Automation Markup Language
CAEX	Datenformat für den rechnergestützten Datenaustausch (en: Computer Aided Engineering Exchange)
ECAD	Rechnergestützte Planung in der Elektrotechnik (en: Computer Aided Engineering for Electrical Engineering)
EDDL	elektronische Gerätebeschreibungssprache (en: Electronic Device Description Language)
FDCML	Auszeichnungssprache für Feldgerätkonfiguration (en: Field Device Configuration Markup Language)
GUID	globaler eindeutiger Bezeichner
GSD	Allgemeine Stationsbeschreibung
ID	Bezeichner (en: identifier)
MCAD	Rechnergestützte mechanische Planung (en: Computer Aided Engineering for Mechanical Engineering)

PDU	Protokolldateneinheit (en: Protocol Data Unit)
UML	vereinheitlichte Modellierungssprache (en: Unified Modelling Language)
UUID	universeller eindeutiger Bezeichner
XML	Extensible Markup Language (erweiterbare Auszeichnungssprache)

4 Anwendungsfälle und Netzwerkstrukturen

4.1 Allgemeines

Die Modellierung von Kommunikationssystemen basierend auf AutomationML zielt auf die Modellierung großer Datenmengen ab, die im Planungsprozess von Produktionssystemen erstellt, ausgetauscht und angewendet werden. Trotzdem werden nicht alle möglichen kommunikationssystembezogenen Daten modelliert. Im folgenden Abschnitt werden Beispiele für die Anwendung der Kommunikationssystemmodellierung sowie die darin enthaltenen relevanten Datensätze genannt.

4.2 Anwendungsfälle

Informationen im Zusammenhang mit der Netzwerktopologie und dem Kommunikationssystem sind für verschiedene Vorgänge in der Planungskette von Produktionssystemen von Bedeutung. Innerhalb des Planungsprozesses von Produktionssystemen kann das Kommunikationssystem in der Detailplanungsphase unter Verwendung verschiedener Arbeitsmittel konzipiert werden. Hierbei werden kommunikationssystembezogene Daten erstellt, die anschließend in der Detailplanung und der Inbetriebnahme von Geräten angewendet werden sollten. Bild 1 zeigt ein Beispiel für relevante Planungsvorgänge im allgemeinen Planungsprozess von Produktionssystemen und die darin integrierte Planung von Kommunikationssystemen.

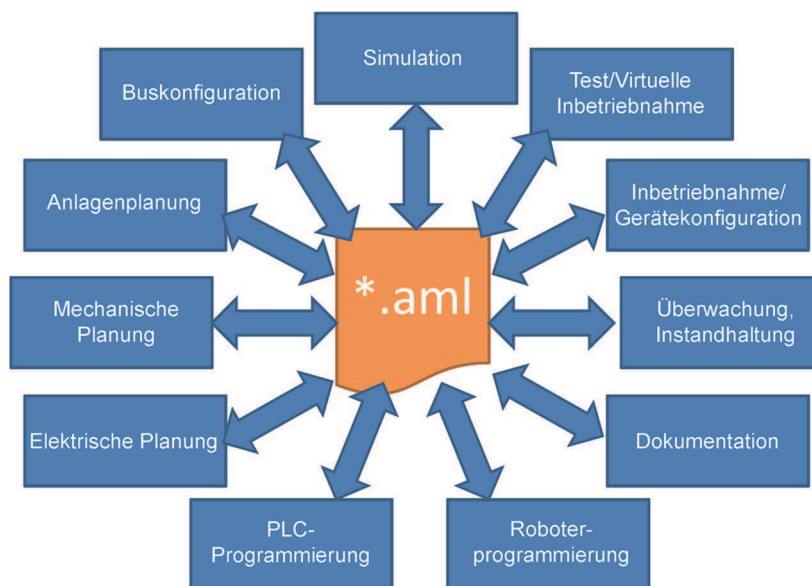


Bild 1 – Allgemeine Planungsvorgänge, in denen die Kommunikationssystemplanung integriert ist

Innerhalb der aufgeführten Planungsvorgänge sind unter anderem die folgenden technischen Werkzeuge von wesentlicher Bedeutung:

- Werkzeug für die Anlagenplanung
- Technisches Werkzeug für die mechanische Planung (MCAD)
- Technisches Werkzeug für die elektrische Planung (ECAD)

E DIN EN IEC 62714-5:2020-12

- Werkzeug für die PLC-Programmierung
- Werkzeug für die Roboterprogrammierung
- Werkzeug für die HMI-Programmierung
- OPC-Systemkonfigurationsmittel
- Hilfsmittel zur Bauteilkonfiguration
- Buskonfigurationstools
- Simulationstools
- SCADA-Systeme
- Virtuelle Inbetriebnahmetools
- Dokumentationsmittel
- Hilfsmittel für die Sicherheit von Kommunikationssystemen
- Hilfsmittel für das Management von Kommunikationssystemen
- Diagnosetool für Kommunikationssysteme

Diese Arbeitsmittel dienen je nach Anwendungsfall in der Planungskette dem Erstellen und/oder Verwenden kommunikationssystembezogener Planungsdaten.

Dennoch bestehen in der Planungskette (unter anderem) zwei Hauptanwendungen, bei denen kommunikationssystembezogene Daten zwischen technischen Arbeitsmitteln ausgetauscht werden sollten. Diese beiden Anwendungsfälle sind das wesentliche Ziel der Modellierung von Kommunikationssystemen basierend auf AutomationML.

4.2.1 Verlustfreie Übertragung der Instanzdaten von Kommunikationsgeräten

Im allgemeinen Planungsprozess werden in diesem Anwendungsfall kommunikationsrelevante Daten für die Konfiguration der Kommunikationsbauteile von Sensoren und Aktoren vom Hilfsmittel für die PLC-Programmierung und ähnlichen Arbeitsmitteln an Hilfsmittel für die Konfiguration/Programmierung von Sensoren und Aktoren übertragen. Dazu zählen relevante Daten für die korrekte Kommunikation (wie Adressen und Kanäle) und die richtige Strukturierung von im Rahmen der Kommunikation übertragenen Kommunikationsdatenpaketen (wie übertragene Sensor-Datenpunkte).

An den dazugehörigen Planungsvorgängen können Techniker wie PLC-Programmierer, HMI-Programmierer, elektrischer Konstrukteur, Inbetriebsetzer von Kommunikation und PLC sowie Roboterprogrammierer beteiligt sein. Es werden die folgenden Planungs- und Datenaustauschvorgänge ausgeführt, deren Abfolge als Beispiel anzusehen ist.

Schritt 1. Gestaltung der Systeminstrumentation, Festlegung der verwendeten/verbundenen Geräte

Schritt 2. Export der Geräte-Informationen aus dem Hilfsmittel für die Systeminstrumentation

Schritt 3. Import der Geräte-Informationen in das Hilfsmittel für PLC-Programmierung und Gerätekonfiguration

Schritt 4. Integration von Gerätebeschreibungen (wie GSD) in Hilfsmittel für die PLC-Programmierung

Schritt 5. Entwurf von PLC-Programmen und -Konfigurationen im Hilfsmittel für die PLC-Programmierung

Schritt 6. Export der für das Kommunikationsgerät relevanten Daten aus dem Hilfsmittel für die PLC-Programmierung

Schritt 7. Import der für das Kommunikationsgerät relevanten Daten in das Hilfsmittel zur Bauteilkonfiguration

Schritt 8. Verwendung der Daten zur Parametrierung des Kommunikationsbauteils des Gerätes (Adressen usw.) und Strukturierung der Kommunikationspakete (Datenpunkte senden)

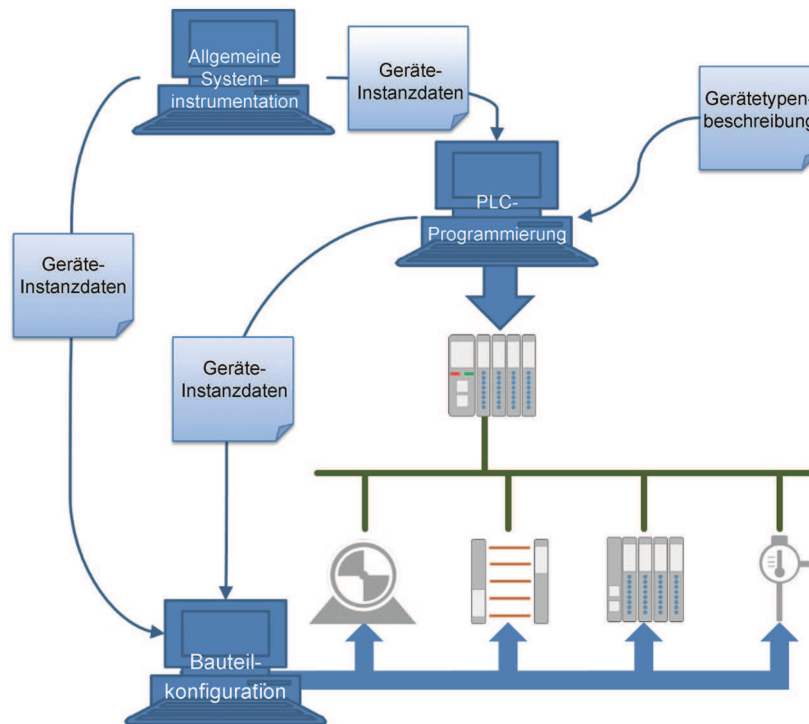


Bild 2 – Datenfluss im Anwendungsfall

Es gibt verschiedene mögliche Alternativen, die aber in der Regel nicht zur Anwendung kommen. Die folgenden Abfolgen sind denkbar und stehen im Mittelpunkt des Anwendungsfalls.

Schritt 1. Gestaltung der Kommunikationssystem-Konfigurationen in einem Arbeitsmittel eines Drittanbieters

Schritt 2. Integration der Gerätebeschreibungen

Schritt 3. Export der für das Kommunikationsgerät relevanten Daten

Schritt 4. Import der für das Kommunikationsgerät relevanten Daten in Hilfsmittel für Bauteilkonfiguration/ Programmierung

Schritt 5. Verwendung der Daten zur Parametrierung des Kommunikationsbauteils des Gerätes (Adressen usw.) und Strukturierung der Kommunikationspakete (Datenpunkte senden)

Schritt 1. Drittanbieterprogramm, z. B. das Hilfsmittel zur Bauteilkonfiguration liefert Daten wie Signale, Datenmenge, mit denen die Instanzdaten beschrieben werden

Schritt 2. Gerätehersteller liefert Gerätebeschreibungen zur Beschreibung des Datentyps

Schritt 3. Buskonfigurationstool verwendet diese Datenfragmente und Gerätebeschreibungen

Schritt 4. Buskonfigurator erstellt die Buskonfiguration

Schritt 5. Import der Buskonfiguration in das Hilfsmittel zur PLC-Programmierung

Schritt 6. Verwendung der Daten durch externe Geräte im PLC-Programm

Beide Abfolgen sind in Bild 3 dargestellt.

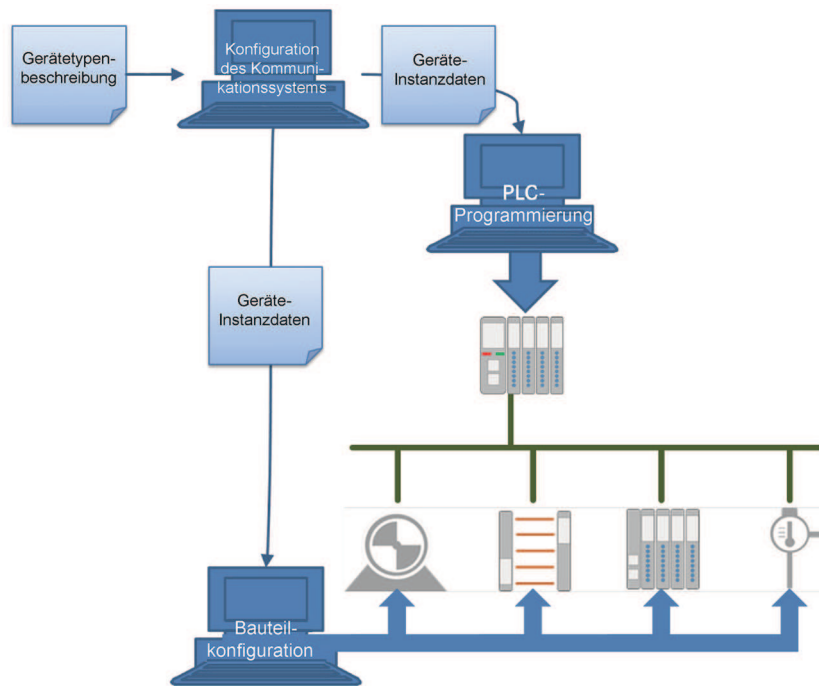


Bild 3 – Alternativer Datenfluss im Anwendungsfall

Mögliche Arbeitsmittel zum Export von Kommunikationssystemdaten können beispielsweise Hilfsmittel für die PLC-Programmierung sein. Diese bieten Möglichkeiten zur PLC-Programmierung mit Datenpunkten (Variablen), Bauteilkonfiguration und indirekte Kommunikationsnetzbeschreibungen. Zusätzlich können technische Arbeitsmittel des Kommunikationssystems ebenso Datenquellen sein wie Instrumentationsanwendungen wie beispielsweise ECAD-Tools.

Datensenken des Datenaustauschs können für die Konfiguration der Sensorkommunikation, die HMI-Programmierung, Roboterprogrammierung, OPC-Systemkonfiguration oder die Konfiguration der Kommunikation zwischen Aktoren genutzt werden. Diese bieten vornehmlich Möglichkeiten zur Programmierung mit Datenpunkten (Variablen), Bauteilkonfiguration und indirekte Kommunikationsnetzbeschreibungen. Zutreffende Arbeitsmittel können FDT-Tools, Hilfsmittel für die HMI-Programmierung und Hilfsmittel für die Roboterprogrammierung sein.

Auf der Grundlage dieses Anwendungsfalls müssen bei der Modellierung von Kommunikationssystemen basierend auf AutomationML Daten zu Eingängen/Ausgängen, Zuordnung von Variablen (E/A) zu Kommunikationsdatenpaketen und Geräteparametern für die Parametrierung von Kommunikationsgeräten wie Adressen (z. B. IP-Adresse), Medienzugriff (z. B. MAC-Adresse), Subnetzmasken, Gateway-Adressen und verwendeten Kommunikationsobjekten wie Profildaten beinhaltet sein.

Zusätzlich sollten durch die Modellierung auch die folgenden nicht-funktionalen Anforderungen erfüllt werden. Die Auflistung der Geräteparameter muss durch Nutzer erweiterbar sein, um zukünftige Technologien einzubinden. Zur Identifizierung der Objektsemantik müssen geeignete Rollen- und/oder Systemunitklassenbibliotheken definiert werden.

4.2.2 Verlustfreie Übertragung von Kommunikationssystemdaten

Neben der Konfiguration von Kommunikationsgeräten werden Kommunikationssystemdaten von verschiedenen technischen Anwendungen für Planung, Überwachung, Instandhaltung usw. genutzt. Aus diesem Grund befasst sich dieser Anwendungsfall mit der Übertragung von Konfigurations- und Strukturdaten des Kommunikationsnetzes einschließlich der Infrastruktur-Bauteilkonfiguration, Konfiguration des Endgerätes hinsichtlich Kommunikationssystemparameter, Netzwerkstruktur und -verdrahtung, Dienstqualität usw. Diese Daten müssen Hilfsmitteln zur Bauteilkonfiguration, Dokumentations- und Instandhaltungsmitteln sowie Netz-Management-Tools bereitgestellt werden. Sie müssen die Kombination physikalischer Verdrahtung mit logischen Kommunikationsverbindungen zur Fehlererkennung ermöglichen.

Bei diesem Anwendungsfall sind Techniker wie PLC-Programmierer, HMI-Programmierer, elektrischer Konstrukteur, Inbetriebsetzer von Kommunikation und PLC sowie Roboterprogrammierer und Bediener beteiligt. Es wird der folgende Planungsprozess ausgeführt, dessen Abfolge als Beispiel anzusehen ist.

Schritt 1. Planung der im Kommunikationssystem, das sich innerhalb der Hilfsmittel für die PLC-Programmierung und der Hilfsmittel für die Programmierung/Konfiguration von Sensoren/Aktoren befindet, zu übertragenden Variablen

Schritt 2. Export dieser Variablen aus den Hilfsmitteln für die PLC-Programmierung und den Hilfsmitteln für die Programmierung/Konfiguration von Sensoren/Aktoren

Schritt 3. Import der Variablen in die technischen Arbeitsmittel des Kommunikationsnetzes

Schritt 4. Planung der Zuordnung von Variablen und Planung der physikalischen Verdrahtung

Schritt 5. Export des Kommunikationssystemdesigns aus den technischen Arbeitsmitteln des Kommunikationsnetzes

Schritt 6. Import des Kommunikationssystemdesigns in nachfolgende technische Arbeits- oder Ausführungsmittel

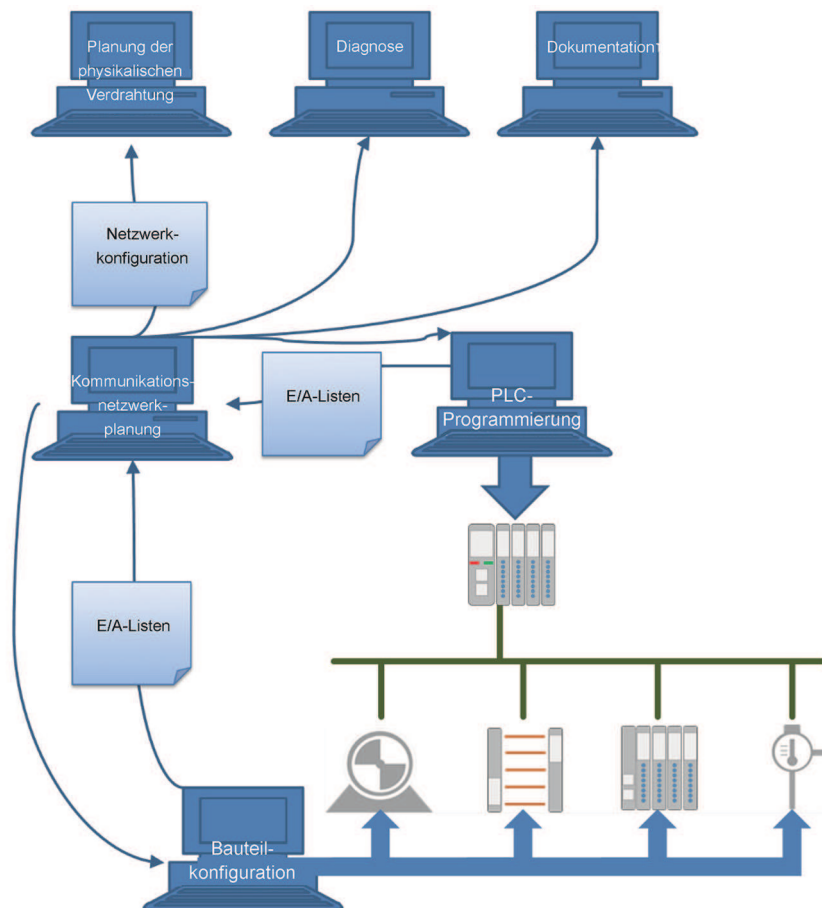


Bild 4 – Datenfluss im Anwendungsfall

Quelle der ausgetauschten Planungsdaten können PLC-Programmierungshilfsmittel für PLC-Programmierungsprojekte mit Datenpunkten (Variablen), Bauteilkonfiguration und indirekten Kommunikationsnetzbeschreibung sein, technische Arbeitsmittel des Kommunikationssystems einschließlich solcher für Bauteilkonfiguration und Kommunikationsverbindungseigenschaften, Konfigurationsmittel für die Sensorkommunikation, Hilfsmittel für die HMI-Programmierung, Hilfsmittel für die Roboterprogrammierung, OPC-Systemkonfigurationsmittel oder Konfigurationsmittel für die Aktorkommunikation, die alle

gemeinsam der Programmierung mit Datenpunkten (Variablen), Bauteilkonfiguration und indirekten Kommunikationsnetzbeschreibung dienen.

Senken des Datenaustauschs können für die Sicherheitskontrolle, Konfiguration, Verwaltung und Diagnose des Kommunikationssystems genutzt werden.

Auf der Grundlage dieses Anwendungsfalls müssen bei der Modellierung von Kommunikationssystemen basierend auf AutomationML Daten zu Netzwerktopologie, Verbindungslisten von Geräten samt Eigenschaften, physikalische Verbindungen zwischen Geräteschnittstellen samt Eigenschaften, logische Verbindungen zwischen gerätebasierten Anwendungen samt Eigenschaften, Zuordnung logischer Verknüpfungen zu physikalischen Verbindungen, Zuordnung von Verknüpfungs- und/oder Verbindungsattributen zu Verknüpfungen oder Verbindungen wie Sender- und Empfängeradressen, Dienstqualität, verwendete Protokolle usw. beinhaltet sein.

Zusätzlich lassen sich nicht-funktionelle Anforderungen ableiten. Die Auflistung der Verknüpfungs- und Verbindungsattribute muss durch Nutzer erweiterbar sein, um zukünftige Technologien einzubinden. Zur Identifizierung der Objektsemantik müssen geeignete Rollen- und/oder Systemunitklassenbibliotheken definiert werden.

4.3 Abgrenzung des Modellierungsbereiches

Im Anwendungsbereich der oben angeführten Anwendungsfälle können unterschiedliche Kommunikationstechnologien, Nutzungsstrategien und Lebenszyklen des Kommunikationssystems sowie Architekturen des Kommunikationsnetzes erwogen werden. Im Folgenden wird die große theoretische Bandbreite auf die am häufigsten angewendeten Fälle eingegrenzt.

4.3.1 Interaktionsstrukturen und Lebenszyklen

Den zuvor beschriebenen Anwendungsfällen folgend, sollte sich die Modellierung des Kommunikationssystems basierend auf AutomationML auf die Beschreibung einer zyklischen Kommunikation auf der Grundlage einer festen Zuweisung der Kommunikationspartner konzentrieren. Diese Kommunikationspartner betreiben (mehr oder weniger komplexe) Teile der Kontrolllogik von Steuerungsanwendungen unterschiedlicher kommunizierender Steuergeräte.

Aus diesem Grund sind mindestens zwei unterschiedliche Sichten auf das Kommunikationssystem von Bedeutung: die logische und die physikalische Sicht. Auf einer hohen logischen Ebene erfolgt ein logischer Austausch von Anwendungsdaten (Variablen) von datenverarbeitenden Anwendungen. Für die physikalische Umsetzung der datenverarbeitenden Anwendungen werden Steuergeräte verwendet. Diese Geräte kommunizieren physikalisch über Kommunikationssysteme. Beide Ausgangspunkte sehen für die modellierten Objekte (Anwendungsteile, Geräte, Verbindungen usw.) besondere Eigenschaften vor, die für den korrekten Gebrauch/das entsprechende Verhalten des Kommunikationssystems relevant sind. Derartige Systeme und deren beschreibende Informationen stehen im Fokus der folgenden Präsentationen.

Außerhalb des Anwendungsbereichs der Modellierung von Kommunikationssystemen basierend auf AutomationML liegen unter anderem

- Kommunikationsstrukturen, die auf azyklischen Diensten basieren und nur zeitlich befristet verwendet werden, wie sie üblicherweise im World Wide Web präsent sind,
- Aspekte in Bezug auf Sicherheit oder Schutz und
- komplexe Daten zu Geräteverhalten und -struktur, die üblicherweise in Sprachen zur Gerätebeschreibung enthalten sind.

Die Verwendung von AutomationML für die Modellierung von Kommunikationssystemen mit diesen Eigenschaften ist nicht ausgeschlossen, wenn auch nicht vorgesehen.

Kommunikationssystembeschreibungen können unterschiedliche Detailstufen umfassen. So ist von sehr abstrakten Beschreibungen mit lediglich der Bezeichnung der Kommunikationspartner samt Abhängigkeiten bis zur ausführlichen Beschreibung der Kommunikationsverbindungen samt Eigenschaften sowie Geräte- und

Anwendungsteilen samt Eigenschaften alles denkbar. Diese unterschiedlichen Detailstufen liegen im Fokus der vorgesehenen Kommunikationssystemmodellierung.

Kommunikationssysteme sind komplexe Systeme, die innerhalb eines Kommunikationssystems verschiedene Technologien umfassen. Als Beispiel werden PLC-gestützte Steuersysteme genannt. Diese beinhalten für gewöhnlich verschiedene Feldbussysteme, eine ethernetbasierte Kommunikation, Kupfer- und Glasfaserkabel, Funkkommunikation, unterschiedliche aktive Infrastruktur usw. Auf derselben physikalischen Infrastruktur können unterschiedliche Kommunikationsprotokolle (verdrahtet, drahtlos usw.) betrieben werden. Ein einzelnes Gerät kann verschiedene Teile einer Steuerungsanwendung ausführen. Daher sind die unterschiedlichen Kommunikationstechnologien miteinander verbunden und voneinander abhängig. Diese Komplexität und gegenseitige Verbindung sollte durch AutomationML ausgedrückt werden können und steht im Fokus der folgenden Beschreibungen.

Kommunikationssysteme werden in verschiedenen Phasen der Projektierung eines Produktionssystems wie der Detailplanung und Inbetriebnahme betrachtet und kommen in unterschiedlichen Phasen des Lebenszyklus eines Produktionssystems wie der Bauteilkonfiguration, Produktionssteigerung, normaler Produktion, Diagnose und Instandhaltung zum Einsatz. Diese unterschiedlichen Phasen bei der Projektierung und Verwendung von Produktionssystemen und ihre Bedeutung auf Inhalt und Verwendung des Modells sollten bei der Modellierung von Kommunikationssystemen basierend auf AutomationML berücksichtigt werden.

4.3.2 Netzwerkobjekte

Ein Kommunikationsnetz besteht aus zwei Ebenen: einer logischen und einer physikalischen Ebene. Bei der Zuordnung dieser Ebenen zum häufig herangezogenen ISO/OSI-Referenzmodell, das aus sieben Schichten besteht [2], entspricht die physikalische Ebene den Schichten 1 und 2 des OSI-Modells und die logische Ebene den Schichten 3 bis 7, mit Steuerungsanwendungen oberhalb von Schicht 7.

Auf der logischen Ebene können die folgenden Einheiten modelliert werden. Steuerungsanwendungen bestehen normalerweise aus mehrfach verzweigten Anwendungsteilen, die im Steuerungsprozess verschiedene Funktionen bereitstellen und logische Geräte bilden. Ein Beispiel (siehe Bild 5) kann eine Hauptsteuerungsanwendung sein, bei der Sensor- und Aktordaten mit zwei E/A-Funktionen ausgetauscht werden. Im Allgemeinen müssen diese logischen Geräte (Teile der Steuerungsanwendung) verschiedenartige Daten austauschen, im angeführten Beispiel sind das unterschiedliche Sensorsignale und Aktorzustände, die als Verbindungspunkte zu den logischen Geräten und Endpunkten des Informationsaustauschs zwischen logischen Geräten angesehen werden können. Der Informationsaustausch selbst erfolgt über verschiedene logische Verbindungen.

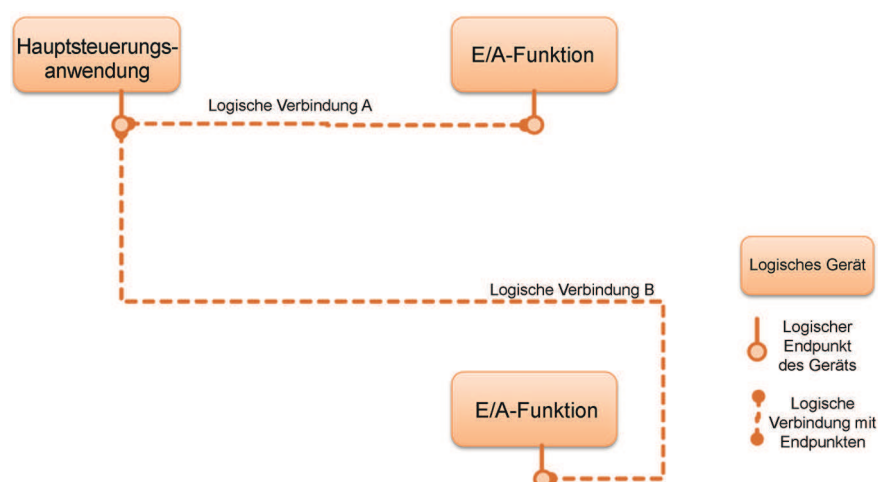


Bild 5 – Beispielhafte Ansicht einer logischen Ebene von Kommunikationssystemen

Das logische Netzwerk kann aus verschiedenen Objekten mit unterschiedlichen beschreibenden Eigenschaften bestehen. In logischen Steuergeräten finden sich beispielsweise eindeutige Bezeichner, Zykluszeiten und Fußabdruck (um nur einige zu nennen), während logische Endpunkte durch Datentypen und logische

Verbindungen durch erforderliche Übertragungsraten gekennzeichnet sein können. In jedem Fall können diese beschreibenden Eigenschaften als Attribute der untersuchten Objekte betrachtet werden.

Auf der physikalischen Ebene finden sich physikalische Geräte mit physikalischen Schnittstellen. In dem in Bild 6 dargestellten Beispiel ist eine PLC über eine aktive Infrastruktur mit zwei E/A-Geräten verbunden. Physikalische Geräte verfügen über physikalische Endpunkte, die für Netzwerkschnittstellen wie Stecker und Buchsen stehen und über physikalische Verbindungen mit einem Kommunikationssystem verbunden sind. Im Vergleich zur logischen Schicht gibt es hier zusätzliche physikalische Einheiten, die für Bauteile der Infrastruktur des Netzes stehen (Switches usw.).

Wie im Fall des logischen Netzwerks können auch die physikalischen Netzwerkobjekte über unterschiedliche beschreibende Eigenschaften verfügen. Physikalische Geräte können über Bezeichner und Verarbeitungskompetenzen verfügen, wohingegen sich physikalische Endpunkte durch Adresse, maximale Datenrate und Verbindungstyp und physikalische Verbindungen durch mögliche Datenrate, Kabeltyp, Antennentyp, Steckertyp oder Dokumentationsnummer kennzeichnen. Diese beschreibenden Eigenschaften können als Attribute der untersuchten Objekte angesehen werden.

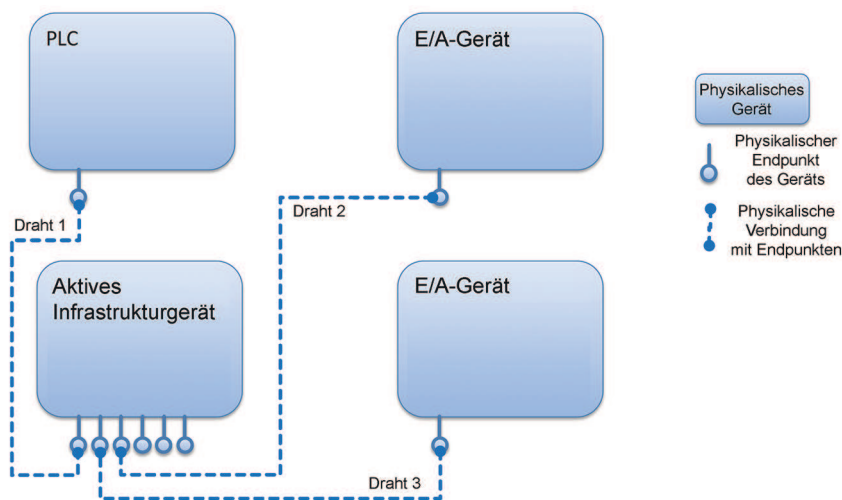


Bild 6 – beispielhafte Ansicht einer physikalischen Ebene von Kommunikationssystemen

Für eine vollständige Beschreibung des Kommunikationssystems müssen beide Ebenen kombiniert werden. Hierbei werden logische Geräte von physikalischen Geräten betrieben, die in einer physikalischen Netzwerktopologie miteinander verbunden sind. Im Beispiel in Bild 7 betreibt eine PLC eine Hauptsteuerungsanwendung, während die E/A-Geräte E/A-Funktionen bereitstellen. Zusätzlich werden physikalische Schnittstellen und logische Schnittstellen zugeordnet. So wird jeder logischen Verbindung virtuell ein Satz physikalischer Verbindungen zugeordnet, die diese implementieren. Die Implementierung durch die physikalischen Verbindungen muss nicht unbedingt einer einmaligen Kette entsprechen, da dies nicht von allen Kommunikationssystemen unterstützt wird.

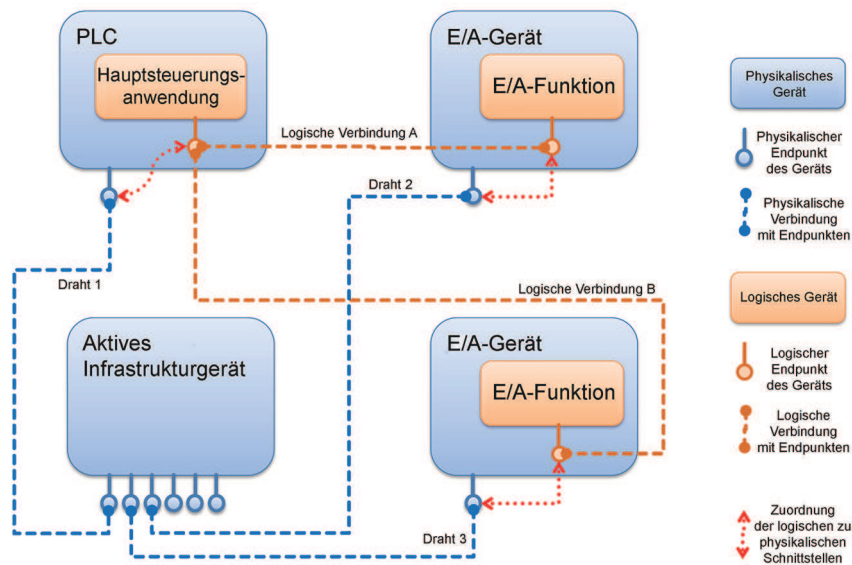


Bild 7 – Kombinierte Sicht auf Kommunikationssysteme

4.3.3 Netzwerktopologien

Bei industriellen Kommunikationssystemen müssen unterschiedliche Netzwerkarchitekturen berücksichtigt werden. Einerseits lassen sich verschiedene Verdrahtungstopologien unterscheiden. Andererseits sind verschiedene physikalische Netzwerktopologien möglich, die häufig von der Kabelführung oder den Einstellungen der Funkkanäle abhängig sind. Die hier vorgestellte Kommunikationssystemmodellierung sollte auf alle anwendbar sein.

Im Hinblick auf unterschiedliche Verdrahtungstopologien sollte die Modellierung Stern-, Linien- und Ringtopologien samt Kombinationen abdecken. Einige Beispiele finden sich in Bild 8, Bild 9 und Bild 10.

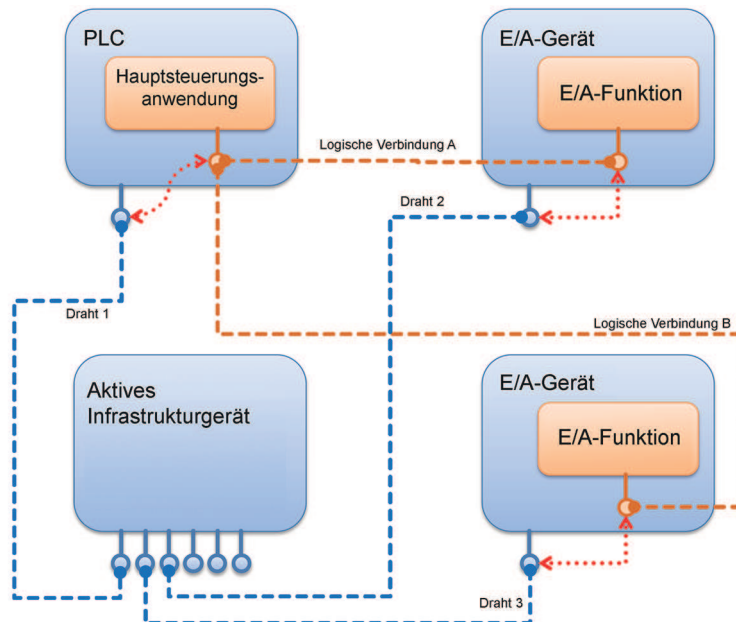


Bild 8 – Beispiel für Sterntopologie

In Bild 8 ist eine typische Sterntopologie dargestellt, bei der das aktive Infrastrukturgerät Verbindungen zwischen den verschiedenen physikalischen Geräten herstellt, während die logischen Verbindungen Zweipunktverbindungen zwischen logischen Geräten bleiben.

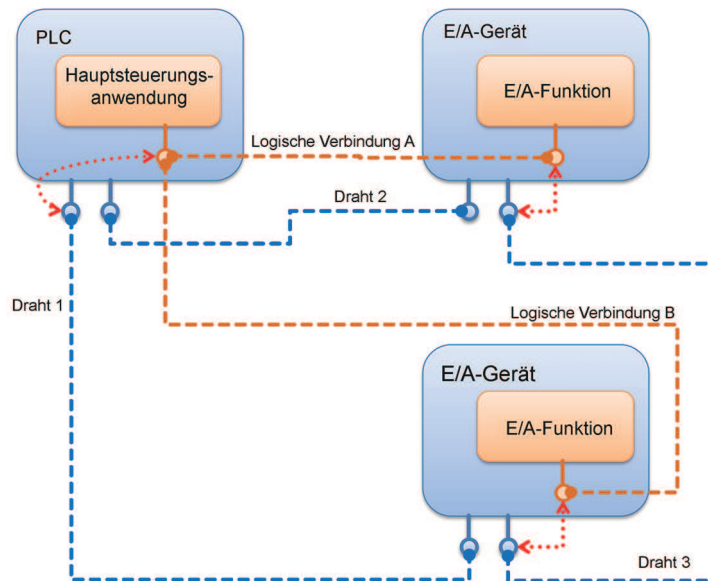


Bild 9 – Beispiel für Ringtopologie

In Bild 9 ist eine typische Ringtopologie dargestellt, bei der jedes physikalische Gerät über zwei physikalische Schnittstellen zur Verbindung der Geräte in einem physikalischen Ring verfügt. Auch hier sind die logischen Verbindungen weiterhin Zweipunktverbindungen zwischen logischen Geräten.

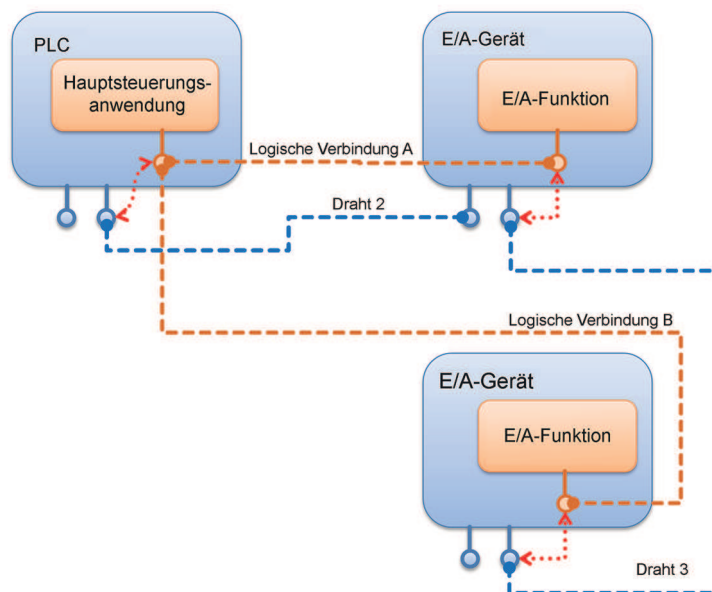


Bild 10 – Beispiel für Linientopologie

Bild 10 zeigt eine typische Linientopologie. Auch hier verfügt jedes physikalische Gerät über zwei physikalische Schnittstellen zur Verbindung der Geräte in einer physikalischen Linie. Daher kann auch von einer offenen Ringtopologie gesprochen werden. Die logischen Verbindungen sind weiterhin Zweipunktverbindungen zwischen logischen Geräten.

Mit Blick auf Netzwerktopologien sollte die Modellierung die Darstellung einfacher Netze mit direkter Verdrahtung der Bauteile, Netze mit aktiver Infrastruktur, durch Gateways verbundene Netze und hierarchische Netze (d. h. Netze, in denen ein Netzbauteil ein vollständiges weiteres Netzwerk enthält, z. B. ein Rückwandbus in einer Modulsteuerung) ermöglichen. Beispiele für solche Netzstrukturen werden in Bild 11, Bild 12, Bild 13 und Bild 14 gezeigt.

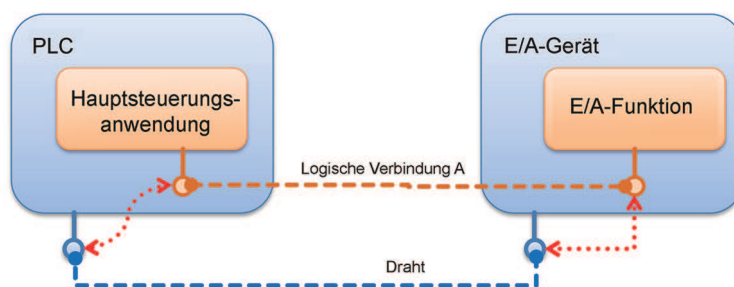


Bild 11 – Einfaches Netzwerk mit direkter Verdrahtung

In Bild 11 ist ein sehr einfaches Netz dargestellt, bei der zwei physikalische Geräte über direkte Verdrahtung eine logische Verbindung zwischen den relevanten logischen Geräten herstellen. Es kann als repräsentatives minimales Netzwerk angesehen werden.

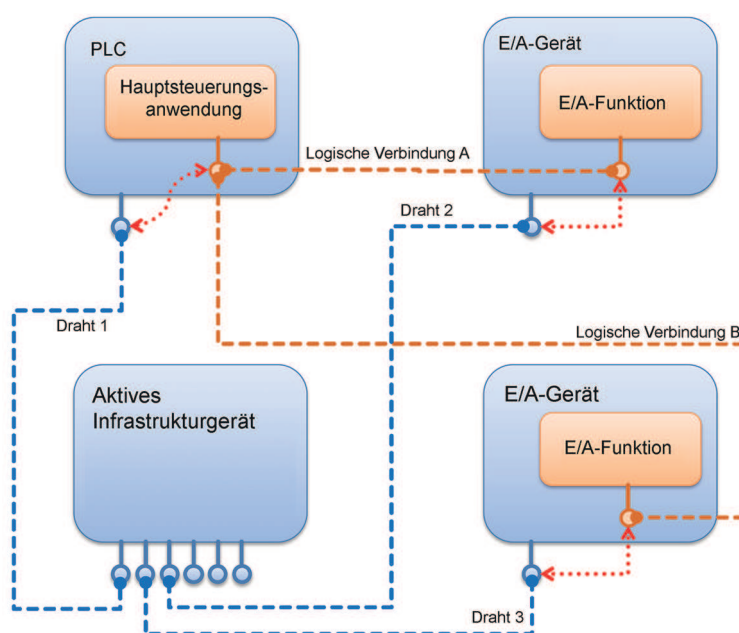


Bild 12 – Netzwerk mit aktiver Infrastruktur

In Bild 12 ist ein Netzwerk mit aktiven Infrastrukturgeräten dargestellt, wie es in ethernetbasierten Netzen üblich ist, in den Switches, Hubs und Router zur Kommunikationsweiterleitung eingesetzt werden.

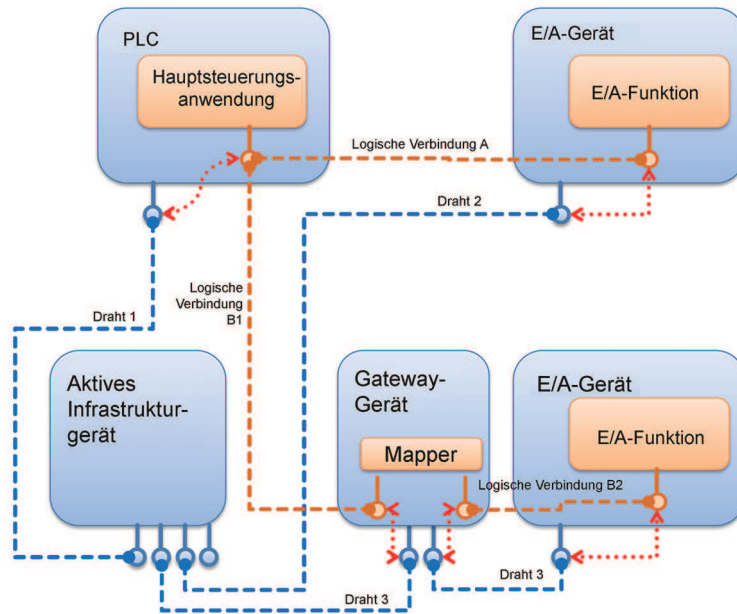


Bild 13 – Durch Gateways verbundene Netzwerke

In Bild 13 ist ein Netzwerk dargestellt, in dem eine logische Verbindung zwei unterschiedliche Netzwerke umfasst, die für die Zuweisung der logischen Verbindungen ein Anwendungs-Gateway verwenden. In diesem Gateway werden die Dateneinheiten der beiden relevanten Verbindungen einander zugeordnet.

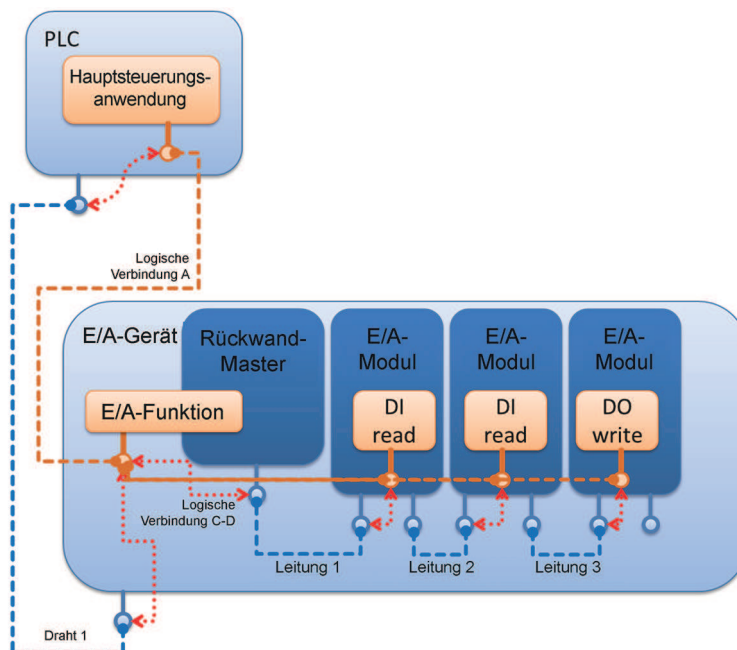


Bild 14 – Hierarchisch strukturierte Netzwerke

Bild 14 stellt eine hierarchische Netzstruktur dar. Diese findet sich üblicherweise in einem Kommunikationssystem, das ein physikalisches Gerät mit einem Rückwand-Netz beinhaltet, das die unterschiedlichen inneren Bauteile des physikalischen Geräts verbindet. Dieser Fall trifft bei Verwendung modularer Feldbuskoppler in einem Netzwerk zu.

Neben der Möglichkeit unterschiedlicher Topologien und Netzwerk-Interaktionsstrukturen finden sich in industriellen Kommunikationssystemen in der Regel verschiedene Anwendungen, die über ein Kommunikationssystem verbunden sind, wie Steuerung und Systemkonfiguration/Programmierung. Netzwerke für mehrere Anwendungen sollten ebenfalls, wie in Bild 15 dargestellt, modelliert werden können.

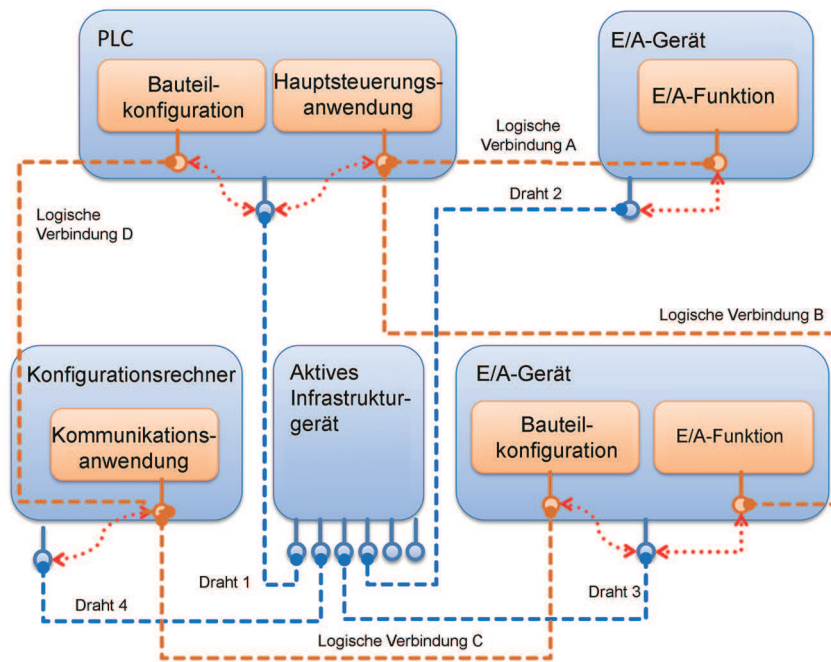


Bild 15 – Netzwerk für mehrere Anwendungen

4.3.4 Inhalt der Kommunikation

In Kommunikationssystemen werden Kommunikationsdatagramme (Protokolldateneinheit – en: Protocol Data Unit, PDU –) unter Kommunikationspartnern, d. h. Teilen der Steuerungsanwendung ausgetauscht. Daher muss jede logische Verbindung PDU-Objekte enthalten, die über diese Verbindung ausgetauscht werden. Im Fall eines PDU-Objektes, das Steuerungsdaten (Signale von Sensoren und Aktoren, Status, Alarme usw.) enthält, kann dieses mit einem PLCopenXMLInterface (siehe IEC 612714-4) oder SignalInterface verknüpft werden, das die ausgetauschten Daten modelliert.

Diese Struktur ist in Bild 16 dargestellt.

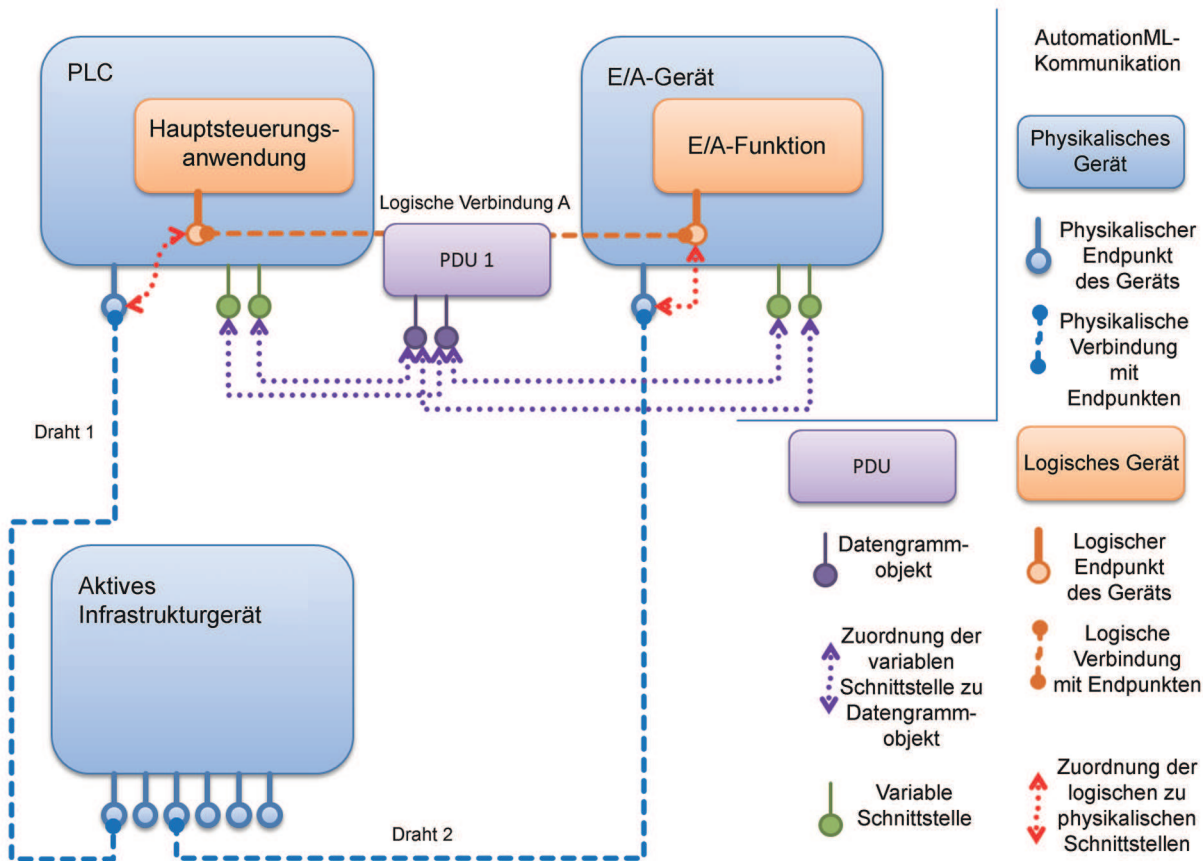


Bild 16 – Allgemeine Modellierungsstrategie für PDUs

4.4 Abgeleitete Modellierungsanforderungen

Auf Grundlage der in diesem Abschnitt angeführten Beobachtungen werden die folgenden allgemeinen Informationen modelliert:

- Informationen auf physikalischer Ebene
 - Physikalische Geräte als Darstellung der physikalisch kommunizierenden Einheiten samt Eigenschaften
 - Endpunkte physikalischer Verbindungen, die an physikalische Geräte angeschlossen sind, als Darstellung der physikalischen Schnittstelle der Geräte (Buchsen) samt beschreibender Eigenschaften
 - Physikalische Verbindungen als Darstellung der physikalischen Verdrahtung oder verwendeten Funkkanäle samt Eigenschaften
- Informationen auf logischer Ebene
 - Logische Geräte als Darstellung von Anwendungsteilen samt Eigenschaften
 - Endpunkte logischer Verbindungen, die an logische Geräte angeschlossen sind, als Darstellung der Schnittstelle des Anwendungsteils samt beschreibender Eigenschaften
 - Logische Verbindungen als Darstellung des Informationsaustauschs zwischen Anwendungsteilen samt beschreibender Eigenschaften
 - Prozessdateneinheiten modellieren die ausgetauschten Datenpakete.

- c) Beziehung zwischen den logischen und physikalischen Ebenen
 - a. Zuordnung der Endpunkte als Darstellung der Implementierung logischer Verbindungen durch physikalische Verbindungen
 - b. Beziehungen zwischen den ausgetauschten Variablen und der/dem angewendeten Kommunikationstechnologie/-protokoll insbesondere durch Zuordnung der Variablen zu Kommunikationspaketstrukturen
- d) Kommunikationsnetze
 - a. Eindeutige Zuordnung von Geräten und Verbindungen zu Netzwerken
 - b. Eindeutig hierarchische Netzstrukturierung
 - c. Abhängigkeiten zwischen Netzwerken
 - d. Abhängigkeiten zwischen Geräten wie z. B. Redundanzeigenschaften
 - e. Abhängigkeiten zwischen Verbindungen wie z. B. Redundanzeigenschaften

5 UML-Modell

5.1 Überblick

Im UML-Modell werden statische Beziehungen zwischen Netzeinheiten durch Klassendiagramme dargestellt. Die Einheiten werden stets als Typen beschrieben. Die reale Implementierung des Modells hängt vom bevorzugten Tool ab. Eine Aggregation stellt hier eine „verfügt-über“-Zuordnung oder Beziehung zwischen den Klassen dar und die Aggregation ist spezifischer als die Zuordnung [1].

Noch wichtiger in den UML-Klassendiagrammen sind die definierten Kardinalitäten. Eine nicht definierte Kardinalität ist als 0...n zu lesen.

Bild 17 zeigt die Fokuspunkte der modellierten Kommunikationsstruktur eines einzelnen Kommunikationsnetzes. Das physikalische Gerät ist das zentrale Objekt der weiteren Beschreibungen. Im Einzelnen wird die modellierte Struktur in logische und physikalische Topologie aufgeteilt.

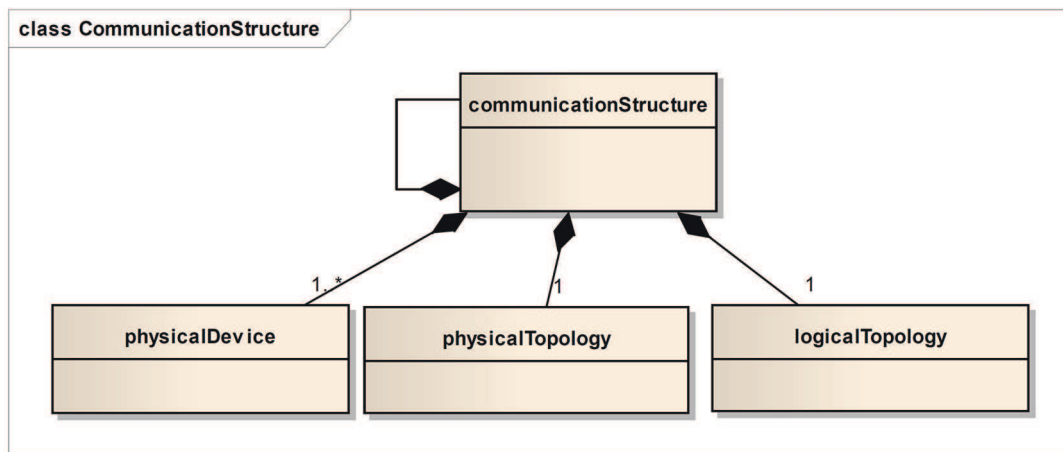


Bild 17 – Struktur des Kommunikationsnetzes

Die Beziehung der Kommunikationsstruktur zu sich selbst (reflexive Zuordnung) ermöglicht die Darstellung verschachtelter Kommunikationsnetze oder mehrerer untereinander verbundener Netzsegmente.

5.2 Logische Topologie

Die logische Topologie beschreibt die logische Sicht auf die Kommunikationspartner. Sie stellt die Kommunikationsbeziehungen aus der Sicht von Anwenderprogrammen dar. Bild 18 zeigt die drei beteiligten (abstrakten) Elemente und ihre Beziehungen:

- logicalTopology
- logicalConnection
- logicalEndPoint

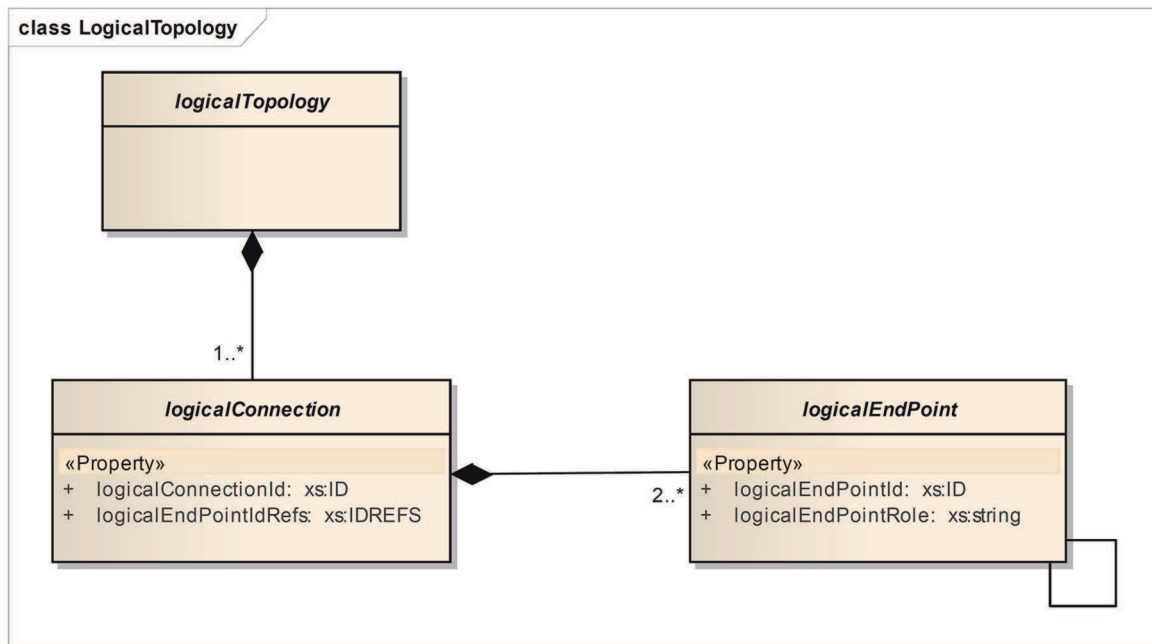


Bild 18 – Sicht auf die logische Topologie

5.2.1 Element logicalTopology

Die Klasse logicalTopology ermöglicht die Modellierung eines Automatisierungs- oder Kommunikationsnetzes auf einem physikalischen Netzwerk. Sie aggregiert sämtliche Kommunikationsbeziehungen zwischen logischen Endpunkten, die wiederum physikalischen Geräten zugeordnet werden.

Als reale Ableitung von logicalTopology ist die Darstellung eines Steuerungsnetzes in einer Anlage zulässig.

5.2.2 Element logicalConnection

Das Element logicalConnection beschreibt die Kommunikationsbeziehung zwischen zwei logischen Endpunkten, die Kommunikationspartner sind. Ein allgemeinerer Begriff ist „logische Verbindung“ (en: logical link), der die Beziehung zwischen mindestens zwei logicalEndPoints angibt.

Die Eigenschaft logicalConnectionId ist ein GUID innerhalb der communicationStructure und bezeichnet die logicalConnection.

Die Eigenschaft logicalEndPointIdRefs ist eine Anordnung mit allen GUIDs von logicalEndPoints mit Bezug zur logicalConnection. Daher können m:n-Beziehungen in einer Kommunikationsbeziehung modelliert werden.

Als reale Ableitung von logicalConnention ist die Verbindungsbeziehung in einer Peer-to-peer-Beziehung zulässig.

5.2.3 Element logicalEndPoint

Das Element logicalEndPoint stellt einen Kommunikationspartner in einer solchen Beziehung dar.

Die Eigenschaft logicalEndPointId ist ein GUID innerhalb der communicationStructure und bezeichnet den logicalEndPoint.

Die Eigenschaft `logicalEndPointRole` entstammt der Datentyp-Zeichenkette und kann zur Spezifizierung der Rolle von `logicalEndPointId` in der Kommunikationsbeziehung verwendet werden. Paare wie „Quelle“ – „Ziel“, „Quelle“ – „Senke“, „Publisher“ – „Abonnent“ oder „Master“ – „Slave“ sind denkbare Werte.

Als Ableitungen eines solchen Kommunikationsendpunktes sind ein IP-Port oder ein Peer zulässig.

5.3 Physikalische Topologie

Die physikalische Topologie beschreibt die physikalische Sicht auf die Kommunikationspartner. Sie stellt die Kommunikationsbeziehungen aus der Sicht von Verdrahtungs- oder Kanaleinstellungen dar. Bild 19 zeigt die drei beteiligten (abstrakten) Elementen und ihre Beziehungen:

- `physicalTopology`
- `physicalConnection`
- `physicalEndPoint`

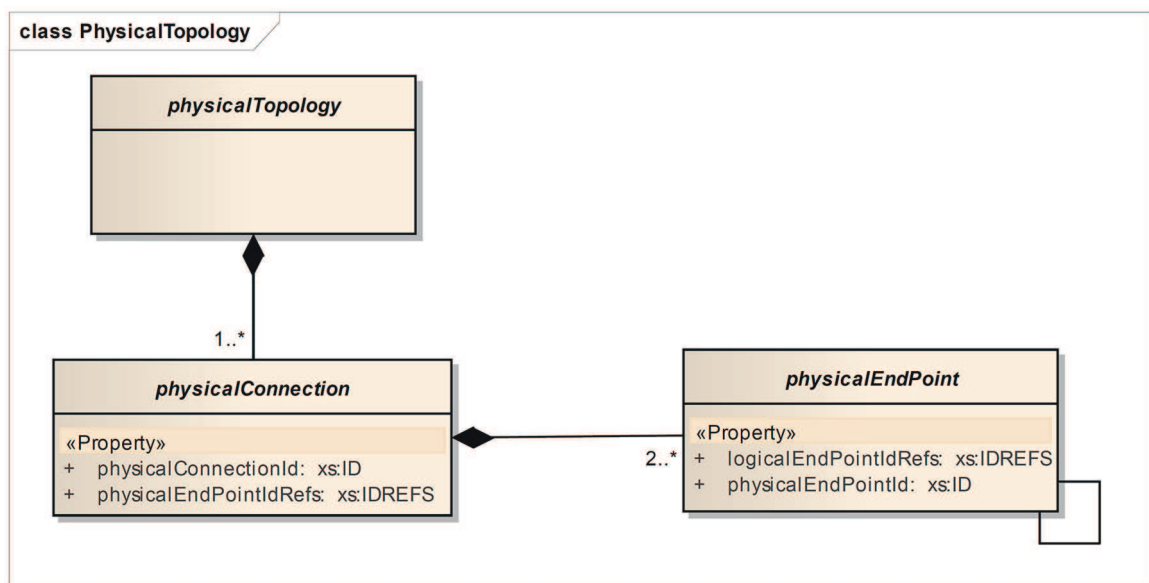


Bild 19 – Sicht auf die physikalische Topologie

5.3.1 Element `physicalTopology`

Die Klasse `physicalTopology` ist der Einstieg für die Modellierung der physikalischen Topologie eines Automatisierungsnetzes. Sie aggregiert sämtliche Kommunikationsbeziehungen zwischen physikalischen Endpunkten, die Automatisierungsgeräten zugeordnet werden.

Als reale Ableitung von `physicalTopology` ist die Darstellung physikalischer Netzwerke in einer Anlage zulässig.

5.3.2 Element `physicalConnection`

Das Element `physicalConnection` beschreibt die Kommunikationsbeziehung zwischen zwei physikalischen Endpunkten. Die `physicalEndPoints` können auch über ein Funknetz verbunden sein. In diesem Fall erfolgt die Verknüpfung über den Äther.

Die Eigenschaft `physicalConnectionId` ist ein GUID innerhalb der `communicationStructure` und bezeichnet die `physicalConnection`.

Die Eigenschaft `physicalEndPointIdRefs` ist eine Anordnung mit allen GUIDs von `physicalEndPoints` mit Bezug zur `physicalConnection`. Daher können m:n-Verbindungen in einem Kommunikationsnetz modelliert werden.

Als reale Ableitung von `physicalConnention` sind Kabelbuchsen in einer Peer-to-peer-Beziehung zulässig.

5.3.3 Element `physicalEndPoint`

Das Element `physicalEndPoint` stellt einen realen Kommunikationsendpunkt in einer solchen Beziehung dar.

Die Eigenschaft `physicalEndPointId` ist ein GUID innerhalb der `communicationStructure` und bezeichnet den `physicalEndPoint`.

Die Eigenschaft `logicalEndPointIdRefs` ist eine Anordnung mit GUIDs von `logicalEndpoints` mit Bezug zu diesem `physicalEndPoint`.

Als Ableitung eines solchen Endpunktes ist ein Ethernetanschluss zulässig.

5.4 Gerät

5.4.1 Allgemeines

Das Modell des Gerätes ist das komplexeste innerhalb des UML-Modells. Bild 20 zeigt den ersten und Bild 21 den zweiten Teil davon.

5.4.2 Element `physicalDevice`

Das `physicalDevice` ist das zentrale Bauteil. Die Attribute ermöglichen eine eindeutige Bezeichnung seiner Instanzen und Subinstanzen. Zudem besteht eine Aggregationsbeziehung zu sich selbst. Anhand dieser Beziehung können modulare Geräte modelliert werden. Die Aggregation der Informationseinheit fasst alle wichtigen Produktdatenbeschreibungen, Bezeichner, Dokumentation usw. als zentrales Objekt zusammen, das auch von Instanzen eines `logicalDevice` verwendet werden darf. Ein `physicalDevice` oder seine Submodule aggregieren die `logicalDevices`. Zudem werden die physikalischen Endpunkte, z. B. Ports, dem `physicalDevice` über die `physicalEndPointList` zugewiesen. Dasselbe Muster wird für die physikalischen Kanäle verwendet. Im vorliegenden Fall ist das `physicalDevice` eine PLC und darf somit mehrere `deviceResources` enthalten.

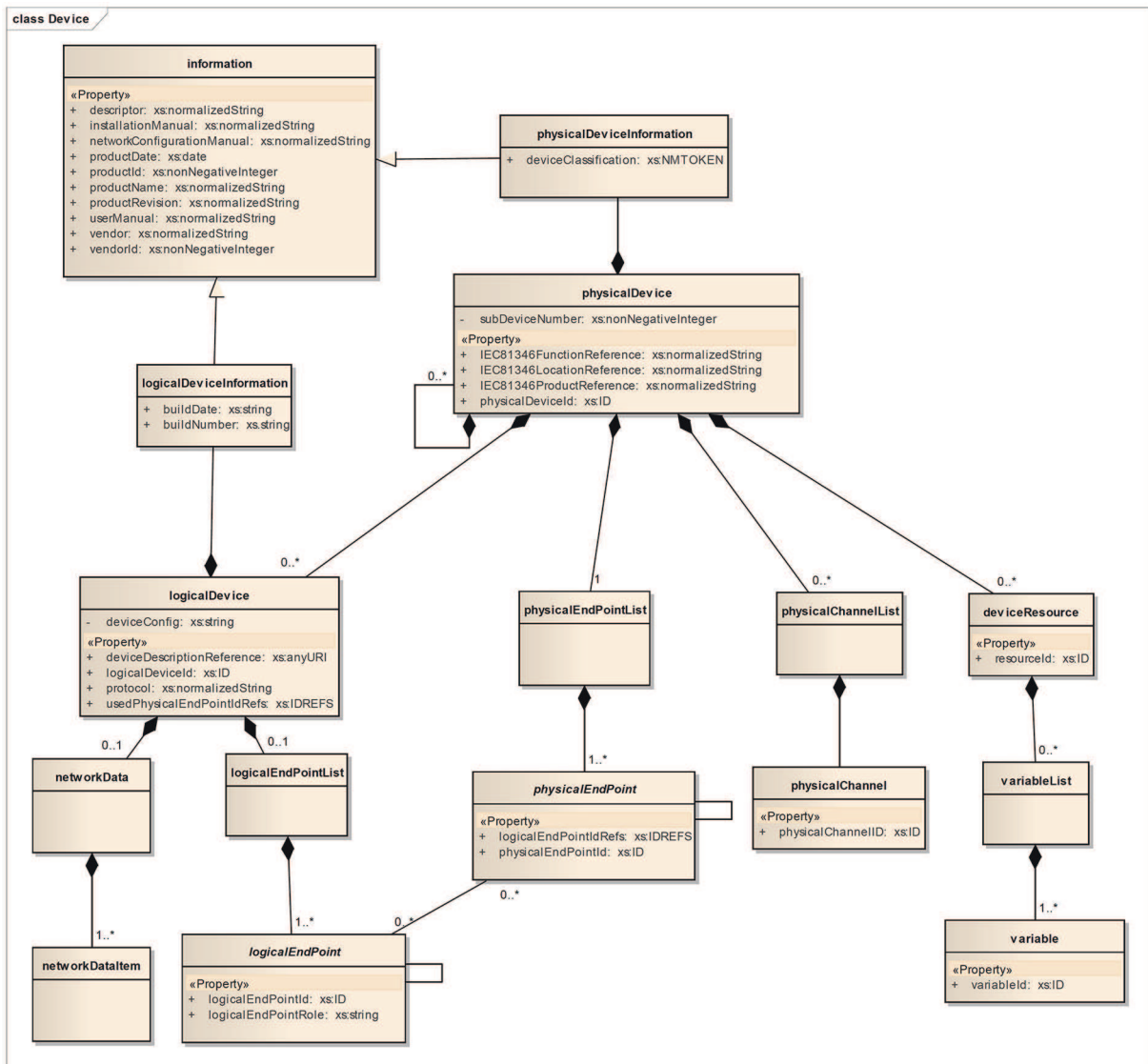


Bild 20 – Teil 1 des Gerätemodells

Die Eigenschaft `physicalDeviceId` ist ein GUID innerhalb der `communicationStructure` und bezeichnet das `physicalDevice`. Wenn es sich um ein Untermodul eines Geräts handelt, definiert die Eigenschaft `subDeviceNumber` die herstellerspezifische Reihenfolge des Untermoduls im Gerät.

Die Eigenschaft `IEC81346FunctionReference` beschreibt den funktionalen Aspekt des `physicalDevice` nach den Regeln von IEC 81346.

Die Eigenschaft `IEC81346LocationReference` beschreibt die Position des `physicalDevice` in der Anlage nach den Regeln von IEC 81346.

Die Eigenschaft `IEC81346ProductReference` beschreibt die Produktinformation des `physicalDevice` nach den Regeln von IEC 81346.

5.4.3 Element Information

Das Element `Information` ist ein Platzhalter für jegliche Information, die dem physikalischen und logischen Gerätetyp zugeordnet ist. Sie dient der ausführlichen Beschreibung der Informationsart einer Geräteinstanz. Die vorgeschlagenen Eigenschaften dürfen im Bedarfsfall ausgeweitet werden.

5.4.4 Element physicalDeviceInformation

Das Element physicalDeviceInformation erweitert den allgemeinen Platzhalter Information, um Geräte-Informationen mit Bezug zum physicalDevice darzustellen.

5.4.5 Element logicalDeviceInformation

Das Element logicalDeviceInformation erweitert den allgemeinen Platzhalter Information, um Geräte-Informationen mit Bezug zum logicalDevice darzustellen.

5.4.6 Element variable

Das Element variable stellt eine Variable, z. B. eines PLC-Programms oder in eine andere Steuereinheit dar. Der Datentyp wird hier nicht ausdrücklich modelliert. Die Einheit variable stellt eine VARIABLE in einer programmierbaren Komponente dar, z. B. eine PLC nach Definition in IEC 61131-3. Die Einheit variable wurde bereits in IEC 62714-4 als Schnittstelle definiert und verweist mittels GUID auf ein variables Element von PLCopenXML.

Die Eigenschaft variableId ist ein GUID innerhalb der communicationStructure und bezeichnet die variable.

5.4.7 Element physicalChannelList

Das Element physicalChannelList stellt einen Platzhalter für Elemente von physicalChannel dar.

5.4.8 Element physicalChannel

Das Element physicalChannel stellt einen Verbindungspunkt zu einem Signal dar, z. B. einen analogen oder binären Ein- oder Ausgang. Der Datentyp wird hier nicht ausdrücklich modelliert. Der physicalChannel wird über die physicalChannelList aufgelistet.

Die Eigenschaft physicalChannelId ist ein GUID innerhalb der communicationStructure und bezeichnet den Kanal oder das Signal.

5.4.9 Element logicalDevice

Das Element logicalDevice stellt nicht physikalische Aspekte eines realen Geräts dar. So beinhalten komplexere Feldgeräte häufig mehrere logicalDevices mit jeweils eigenen Adresseninformationen und spezifischen Aufgaben.

Das logicalDevice aggregiert die networkDetailItems über die networkDataList, die für Konfigurationsaufgaben verwendet wird. Zudem werden die logicalEndPoints über die logicalEndPointList aggregiert. Zusätzlich zur Darstellung der logischen Topologie in 4.2 aggregiert der logicalEndPoint zudem die Protokolldateneinheiten, die über die logische Verbindung ausgetauscht werden.

Die Eigenschaft deviceConfig beschreibt die während des Startens des logicalDevice verwendeten Informationen. Diese dürfen während der Laufzeit je nach Kommunikationssystem unter verschiedenen Kommunikationspartnern ausgetauscht werden.

Die Eigenschaft deviceDescriptionReference verweist auf eine externe Gerätebeschreibungsdatei.

Die Eigenschaft logicalDeviceId ist ein GUID innerhalb des Automatisierungssystems und bezeichnet das logicalDevice.

Die Eigenschaft protocol beschreibt das verwendete Protokoll, z. B. PROFINET oder PROFIBUS usw.

Die Eigenschaft usedPhysicalEndPointIdRefs stellt eine Liste von GUID mit zugeordneten physicalEndPoints dar. Das bedeutet, dass diese Endpunkte vom logicalDevice verwendet werden.

5.4.10 Element networkDataList

Das Element networkDataList stellt die dem logicalDevice zugeordneten networkDataItems dar.

5.4.11 Element networkDataItem

Ein networkDataItem ist eine spezifische Konfigurationseinheit, die zu dem logicalDevice gehört. Typische networkDataItems können MAC-Adresse, IP-Adresse, Subnetzmaske, Stationsname usw. sein.

5.4.12 Element pduList

Das Element pduList stellt die über den logicalEndPoint ausgetauschten Protokolladateneinheiten dar.

5.4.13 Element pdu

Das Element pdu stellt eine spezifische Protokolladateneinheit mit Bezug zu dieser logicalConnection dar. Sie aggregiert protocolData und payload. payload lässt sich in processDataItem oder parameterItemList spezialisieren.

5.4.14 Element protocolData

Das Element protocolData kann zum Beschreiben des Protokoll-Overhead verwendet werden, der in der Regel durch processDataItem oder parameterItem nicht modelliert wird. Für reale Kommunikationsprotokolle können Spezialisierungen dieser Einheiten definiert werden.

5.4.15 Element payload

Das Element payload stellt processDataItem und parameterItem als generalisiertes abstraktes Element dar. Die optionale Eigenschaft address darf zur Adressierung oder Bezeichnung von dataItem verwendet werden.

5.4.16 Element processDataItemList

Diese Element ist ein Muster zur Aufzählung sämtlicher processDataItems.

5.4.17 Element parameterItemList

Dieses Element ist ein Muster zur Aufzählung sämtlicher parameterItems. Zusätzlich verfügt eine pdu dieses Typs über eine kommunikationsprotokollspezifische Adresse.

5.4.18 Element dataItem

Das Element dataItem stellt ein über die logicalConnection kommuniziertes Datenobjekt dar.

Die Eigenschaft octetOffset definiert den Byte-Offset des Datenobjekts in den Nutzdaten der PDU. Es wird ab dem ersten Byte der Nutzdaten gezählt, beginnend mit dem Index „0“ für das erste zu sendende/empfangende Byte.

Die Eigenschaft bitOffset definiert den Bit-Offset des Datenobjekts innerhalb eines adressierten Bytes der PDU-Nutzdaten.

Die Eigenschaften octetOffset und bitOffset dürfen alternativ verwendet werden. Bei Verwendung beider Eigenschaften müssen diese addiert werden, um die Bitposition zu berechnen.

Die Eigenschaft numberOfBits definiert die Länge des Datenobjekts in Bits.

5.4.19 Element processDataltem

Das Element processDataltem stellt ein über die logicalConnection kommuniziertes Prozessdatenobjekt dar. Solche Datenobjekte werden häufig über zyklische Dienstelemente des Kommunikationssystems übertragen.

Die Eigenschaft processDataltemId ist ein GUID innerhalb des Automatisierungssystems und bezeichnet das processDataltem.

Die Eigenschaft variableIdRef ist die Referenz des processDataltem im Prozessdatenbild, das einer Variablen des Programms von PLC oder Steuereinheit zugeordnet ist. Die Referenz wird eingestellt, wenn das physicalDevice eine PLC oder Steuereinheit ist.

Die Eigenschaft processDataltemIdRef ist die Referenz des processDataltem im Prozessdatenbild, das einer Variablen des Geräteprogramms zugeordnet ist. Die Referenz wird eingestellt, wenn das physicalDevice keine PLC oder Steuereinheit ist.

Die Eigenschaft physicalChannelIdRef ist die Referenz des processDataltem auf einen physicalChannel eines Geräts.

5.4.20 Element logicalEndPointList

Dieses Element ist ein Muster zur Aufzählung aller verfügbaren logicalEndPoints.

5.4.21 Element deviceResource

Das Element deviceResource stellt eine Laufzeitumgebung (z. B. CPU oder virtuelle Maschine mit spezifischer Software) in einem Gerät dar, die die Ausführung von Programmen oder Programmteilen über ihre Mensch-Maschine-Schnittstelle oder Schnittstellenfunktionen von Sensor und Aktor ermöglicht. Diese Einheit ist eine Art Darstellung des Begriffs RESSOURCE, der in IEC 61131-3 verwendet wird.

5.4.22 Element variableList

Das Element ist ein Muster zur Aufzählung aller verfügbaren Variablen in einer deviceResource.

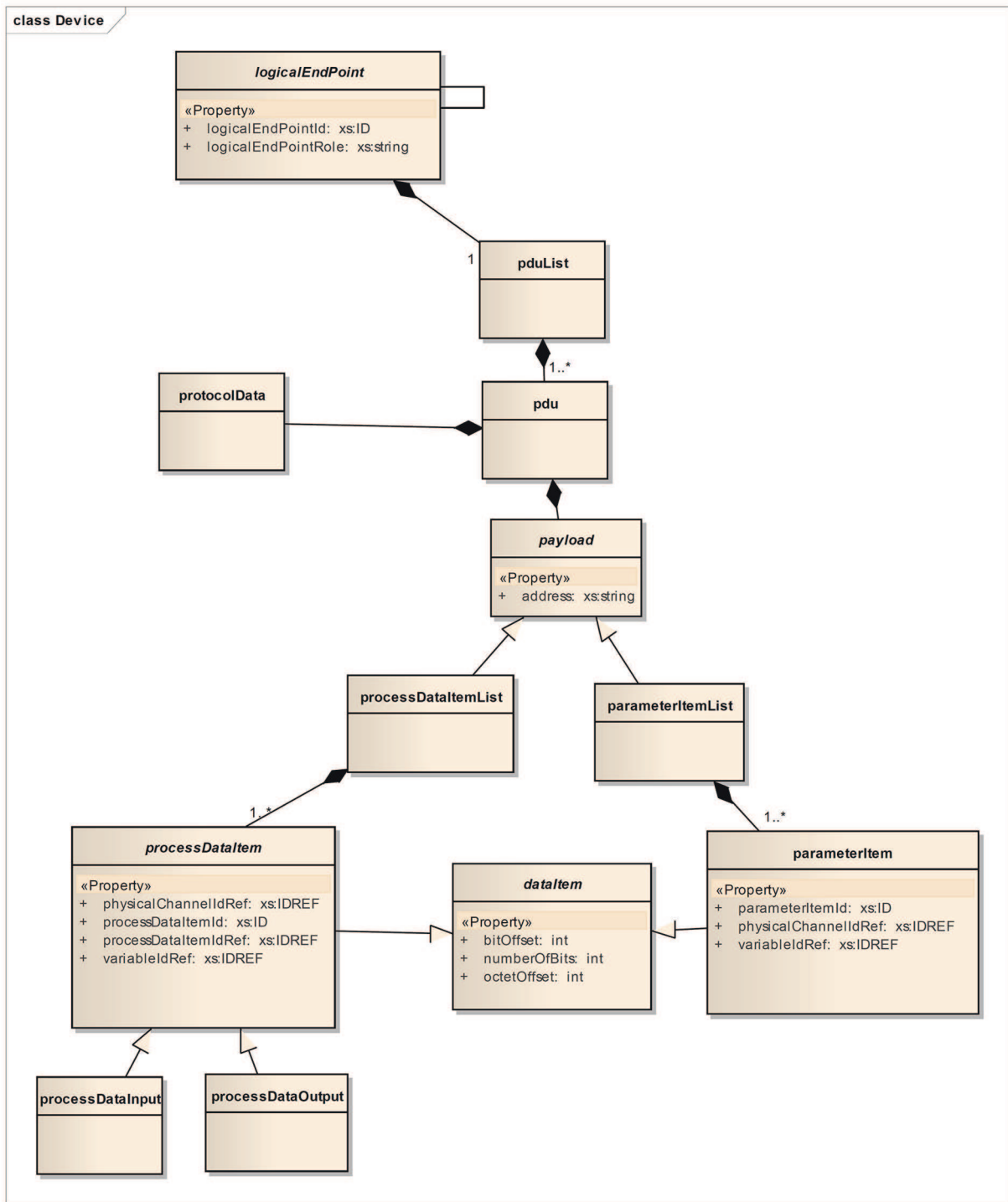


Bild 21 – Teil 2 des Gerätemodells

5.4.23 Element processDataInput

Das Element processDataInput ist eine Spezialisierung von processDataItem und stellt den Datenfluss der Prozessdaten als einen Eingang des gesteuerten Prozesses dar.

5.4.24 Element processDataOutput

Die Element processDataOutput ist eine Spezialisierung von processDataItem und stellt den Datenfluss der Prozessdaten als einen Ausgang des gesteuerten Prozesses dar.

5.4.25 Element parameterItem

Das Element parameterItem stellt ein über die logicalConnection kommuniziertes Parameterobjekt dar. Solche Datenobjekte werden häufig über azyklische Dienstelemente des Kommunikationssystems übertragen.

Die Eigenschaft parameterItemId ist ein GUID innerhalb des Automatisierungssystems und bezeichnet das parameterItem.

Die Eigenschaft variableItemIdRef verweist auf eine Variable des PLC-Programms. Die Referenz wird eingestellt, wenn das physicalDevice eine PLC oder Steuereinheit ist.

Die Eigenschaft physicalChannelIdRef ist die Referenz des processDataItem auf einen physicalChannel eines Geräts.

6 Darstellung in AutomationML

6.1 Überblick über die Zuordnung

Die Methode zur Modellierung eines Kommunikationssystems in AutomationML basiert auf der Verwendung eigener Rollenklassen (<RoleClass>) und Schnittstellenklassen (<InterfaceClass>) sowie einem besonderen Anwendungsprozess. Diese Methode greift die Unterscheidung zwischen Anwendungs- und physikalischen Netzwerken sowie deren Verbindung untereinander auf, wie in 4 und 5 beschrieben.

6.1.1 Allgemeine Zuordnungsregeln

Zur Darstellung der in 4 und 5 genannten unterschiedlichen Elementen werden die in Tabelle 1 aufgeführten folgenden Zuordnungsregeln angewendet. Hierbei wird die Art und Weise der Darstellung der Semantik eines Objekts und Modellierung eines Elementes mit Bezug dazu unterschieden.

Tabelle 1 – Zuordnungsregeln

Informationseinheit des UML-Modells	Semantische Modellierung	Einheit-Modellierung
logicalTopology	RoleClass LogicalNetwork	InternalElement
logicalConnection	RoleClass LogicalConnection	InternalElement
logicalEndpoint	InterfaceClass LogicalEndPoint	Schnittstelle
physicalTopology	RoleClass PhysicalNetwork	InternalElement
physicalConnection	RoleClass PhysicalConnection	InternalElement
physicalEndpoint	InterfaceClass PhysicalEndPoint	Schnittstelle
physicalDevice	RoleClass PhysicalDevice	InternalElement
variable	InterfaceClass VariableInterface nach Definition in IEC 62714-4	Schnittstelle
physicalChannel	InterfaceClass SignalInterface nach Definition in IEC 62714-1	Schnittstelle
logicalDevice	RoleClass LogicalDevice	InternalElement
pduList	Implizit modelliert durch Hosting von InternalElement	
pdu	RoleClass CommunicationPackage	InternalElement
protocolData	Attribut oder InterfaceClass DatagrammObject	Attribut oder Schnittstelle
payload	InterfaceClass DatagrammObject	Schnittstelle