INTERNATIONAL STANDARD

ISO 15745-4

First edition
2003-11-15

AMENDMENT 2
2007-02-01

Industrial automation systems and integration — Open systems application integration framework —

Part 4: Reference description for Ethernet-based control systems

AMENDMENT 2: Profiles for Modbus TCP, EtherCAT and ETHERNET Powerlink

Systèmes d'automatisation industrielle et intégration — Cadres d'intégration d'application pour les systèmes ouverts —

Partie 4: Description de référence pour les systèmes de contrôle fondés sur Ethernet

AMENDEMENT 2: Profils pour Modbus TCP, EtherCAT et ETHERNET Powerlink

Reference number
ISO 15745-4:2003/Amd.2:2007(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

Amendment 2 to ISO 15745-4:2003 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 5, *Architecture, communications and integration frameworks*. Amendment 2 to ISO 15745-4:2003 specifies profiles for Modbus TCP[1)], EtherCAT[2)], and ETHERNET Powerlink[3)], and, as such, adds to the number of technology-specific elements and rules in ISO 15745-4 for describing both communication network profiles and communication-related aspects of device profiles, thus further extending the Application Integration Framework in ISO 15745-1.

---

1) Modbus® and Modbus TCP™ are the registered trademarks of Schneider Automation Inc. This information is given for the convenience of users of ISO15745 and does not constitute an endorsement by ISO of the trademark holder or any of its products. Compliance to ISO15745 does not require use of the trade name Modbus TCP or Modbus. Use of the trademark Modbus TCP or Modbus requires permission of the trade name holder.

2) EtherCAT™ is the registered trademark of Beckhoff, Verl. This information is given for the convenience of users of ISO15745 and does not constitute an endorsement by ISO/IEC of the trademark holder or any of its products. Compliance to ISO15745 does not require use of the trade name EtherCAT™. Use of the trademark EtherCAT requires permission of the trade name holder.

3) ETHERNET Powerlink™ is the registered trademark of ETHERNET Powerlink Standardization Group. This information is given for the convenience of users of ISO15745 and does not constitute an endorsement by ISO/IEC of the trademark holder or any of its products. Compliance to ISO15745 does not require use of the trade name ETHERNET Powerlink™. Use of the trademark ETHERNET Powerlink requires permission of the trade name holder.

Industrial automation systems and integration — Open systems application integration framework —

Part 4:
Reference description for Ethernet-based control systems

AMENDMENT 2: Profiles for Modbus TCP, EtherCAT and ETHERNET Powerlink

Page 1, Clause 2

Add the following normative references:

"ISO 1000*, SI units and recommendations for the use of their multiples and of certain other units*

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

IEC/PAS 62030, *Digital data communications for measurement and control - Fieldbus for use in industrial control systems - Section 1: MODBUS® Application Protocol Specification V1.1a - Section 2: Real-Time Publish-Subscribe (RTPS) Wire Protocol Specification Version 1.0*

IEC/PAS 62407, *Real-time Ethernet control automation technology (EtherCATTM)*

IEC/PAS 62408, *Real-time Ethernet Powerlink (EPL)*

RFC 1157 SNMP, S*imple Network Management Protocol (SNMP) Management Frameworks"*

Page 2, Clause 4

Add the following abbreviated terms:

"DDXML Device Description eXtensible Markup Language

EPL ETHERNET Powerlink

FMMU Fieldbus Memory Management Unit

MIB Management Information Base

SNMP Simple Network Management Protocol (RFC 1157)"

Page 4, Table 1

Add a row with the entries "DDXML" under the "ProfileTechnology name" column and "Modbus TCP" under the "Technology" column.

Add a row with the entries "EtherCAT" under the "ProfileTechnology name" column and "EtherCAT" under the "Technology" column.

Add a row with the entries "EPL" under the "ProfileTechnology name" column and "ETHERNET Powerlink" under the "Technology" column.

Page 4, Subclause 5.3

Add the following items to the list in the first paragraph:

"— Modbus TCP (see 6.5)

— EtherCAT (see 6.6)

— ETHERNET Powerlink (see 6.7)".

Page 18

Insert the following new subclauses 6.5, 6.6 and 6.7 after subclause 6.4 inserted from Amendment 1 to ISO 15745-4:2003, and before Annex A.

6.5 Modbus TCP

6.5.1 Device profile

6.5.1.1 General

Figure 20 shows the class structure of a Modbus TCP device profile.

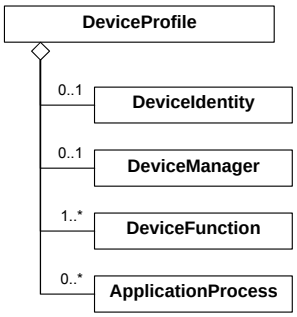


Figure 20 — Modbus TCP device profile class diagram

NOTE The Modbus TCP device profile class diagram shown in Figure 20 defines the main classes. These classes are further decomposed and detailed in Annex E.

The XML schemas representing the Modbus TCP device profile template are defined in E.4.6. The template is based on two parts:

— the DDXML profile header defined in E.3, and

— the DDXML device profile defined in E.4.

6.5.1.2 Device identity

The DeviceIdentity class contains attributes that are independent of the network, of the process and uniquely identify the device.

Figure 21 shows the structure of the Modbus TCP DeviceIdentity class.

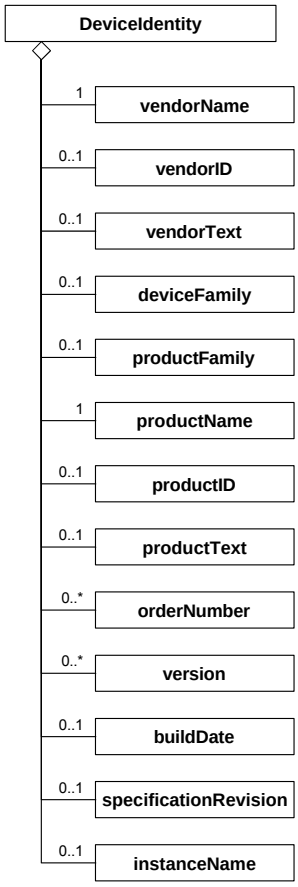


Figure 21 — Modbus TCP DeviceIdentity class diagram

Further details are given in E.4.2.

6.5.1.3 Device manager

The DeviceManager class contains attributes and supports services that enable the monitoring of the device. Communication specific configuration data and mapping information is defined in the communication network specific part structured according to the schema specified in E.5.

Figure 22 shows the structure of the Modbus TCP DeviceManager class.

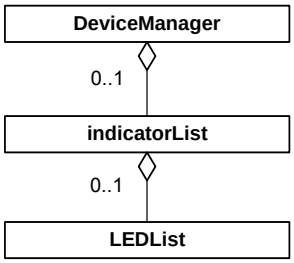


Figure 22 — Modbus TCP DeviceManager class diagram

Further details are given in E.4.3.

6.5.1.4 Device function

The DeviceFunction class describes the intrinsic function of a device in terms of its technology. It contains network independent descriptions/definitions of the technological device functionality.

Figure 23 shows the structure of the Modbus TCP DeviceFunction class.

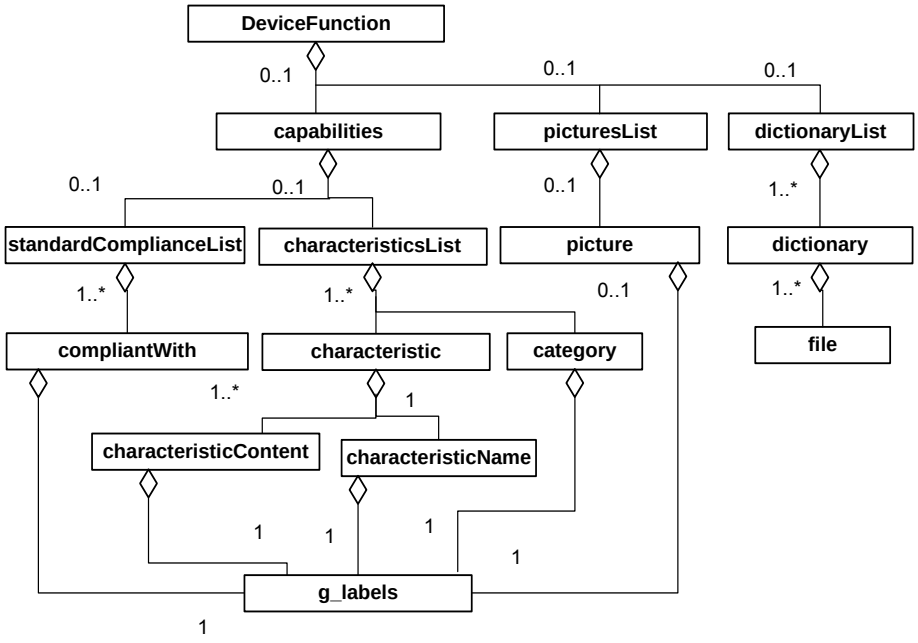


Figure 23 — Modbus TCP DeviceFunction class diagram

Further details are given in E.4.4.

6.5.1.5 Application process

The ApplicationProcess class represents the set of services and parameters, which constitute the behaviour, and the interfaces of the device in terms of the application, independent of the device technology and the underlying communication networks and communication protocols.

Figure 24 shows the structure of the Modbus TCP ApplicationProcess class.

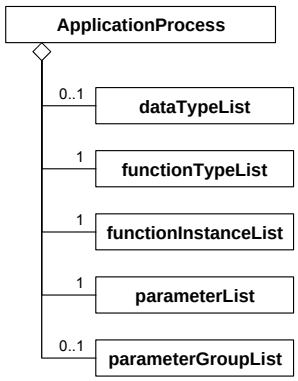


Figure 24 — Modbus TCP ApplicationProcess class diagram

Further details are given in E.4.5.

6.5.2 Communication network profile

6.5.2.1 General

Figure 25 shows the class structure of the Modbus TCP communication network profile. These classes are further decomposed and detailed in Annex E.

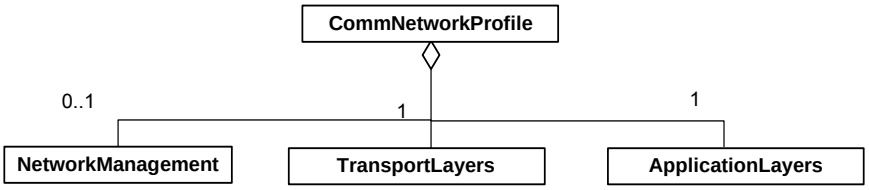


Figure 25 — Modbus TCP communication network profile class diagram

The XML schemas representing the Modbus TCP communication network profile template are defined in E.5.5. Like for the device profile, the template is also based on two parts:

— the DDXML profile header defined in E.3, and

— the DDXML communication network profile defined in E.5.

6.5.2.2 Application layers

The Modbus TCP ApplicationLayers class represents the combined profiles for the upper 3 OSI layers of the Modbus TCP communication network integration model.

Further details are given in E.5.2.

6.5.2.3 Transport layers

The Modbus TCP TransportLayers class represents the combined profiles for the lower 4 OSI layers of the Modbus TCP communication network integration model.

Further details are given in E.5.3.

6.5.2.4 Network management

The Modbus TCP NetworkManagement class represents the network configuration and performance adjustment capabilities of the Modbus TCP communication network integration model.

Further details are given in E.5.4.

6.6 EtherCAT

6.6.1 Device profile

6.6.1.1 General

Figure 26 shows the class structure of an EtherCAT device profile.

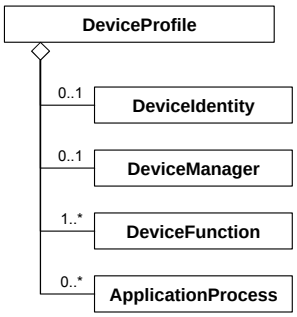


Figure 26 — EtherCAT device profile class diagram

NOTE The EtherCAT device profile class diagram shown in Figure 26 defines the main classes. These classes are further decomposed and detailed in Annex F.

The XML schema representing the EtherCAT device profile template is defined in F.4.6. The template is based on two parts:

— the EtherCAT profile header defined in F.3, and

— the EtherCAT device profile defined in F.4.

6.6.1.2 Device identity

The DeviceIdentity class contains attributes that are independent of the network, of the process and uniquely identify the device.

Figure 27 shows the structure of the EtherCAT DeviceIdentity class.

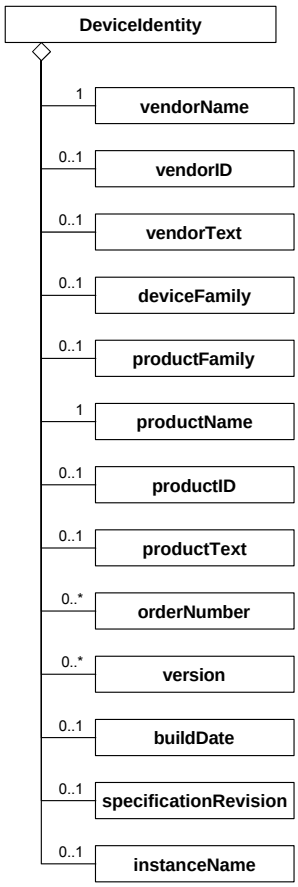


Figure 27 — EtherCAT DeviceIdentity class diagram

Further details are given in F.4.2.

6.6.1.3 Device manager

The DeviceManager class contains attributes and supports services that enable the monitoring of the device. Communication specific configuration data and mapping information is defined in the communication network specific part structured according to the schema specified in F.5.

Figure 28 shows the structure of the EtherCAT DeviceManager class.

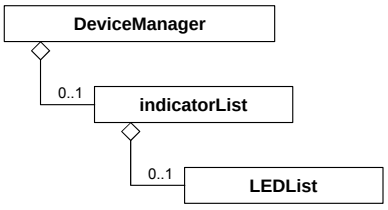


Figure 28 — EtherCAT DeviceManager class diagram

Further details are given in F.4.3.

6.6.1.4 Device function

The DeviceFunction class describes the intrinsic function of a device in terms of its technology. It contains network independent descriptions/definitions of the technological device functionality.

Figure 29 shows the structure of the EtherCAT DeviceFunction class.

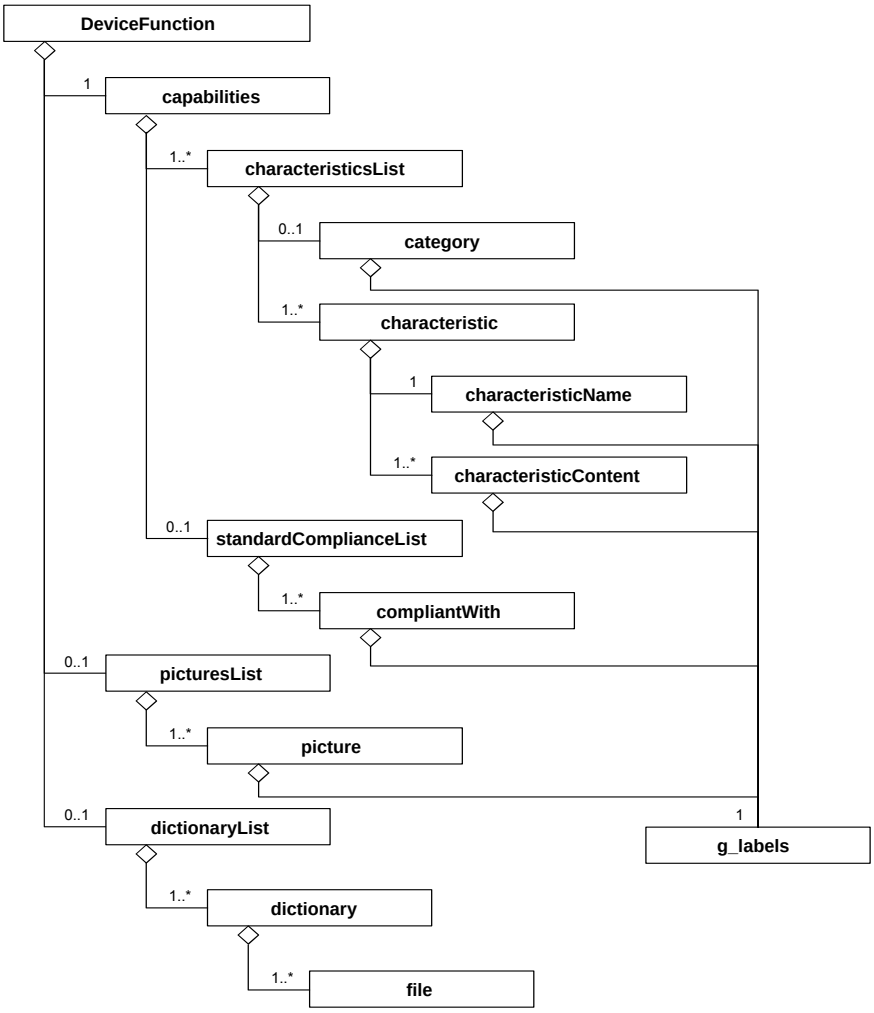


Figure 29 — EtherCAT DeviceFunction class diagram

Further details are given in F.4.4.

6.6.1.5 Application process

The ApplicationProcess class represents the set of services and parameters, which constitute the behaviour, and the interfaces of the device in terms of the application, independent of the device technology and the underlying communication networks and communication protocols.

Figure 30 shows the structure of the EtherCAT ApplicationProcess class.

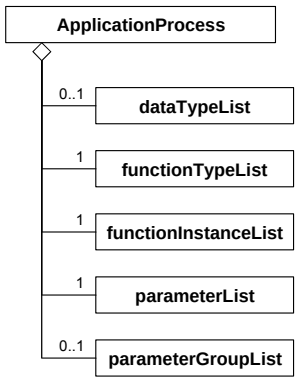


Figure 30 — EtherCAT ApplicationProcess class diagram

Further details are given in F.4.5.

6.6.2 Communication network profile

6.6.2.1 General

Figure 31 shows the class structure of the EtherCAT communication network profile. These classes are further decomposed and detailed in Annex F.

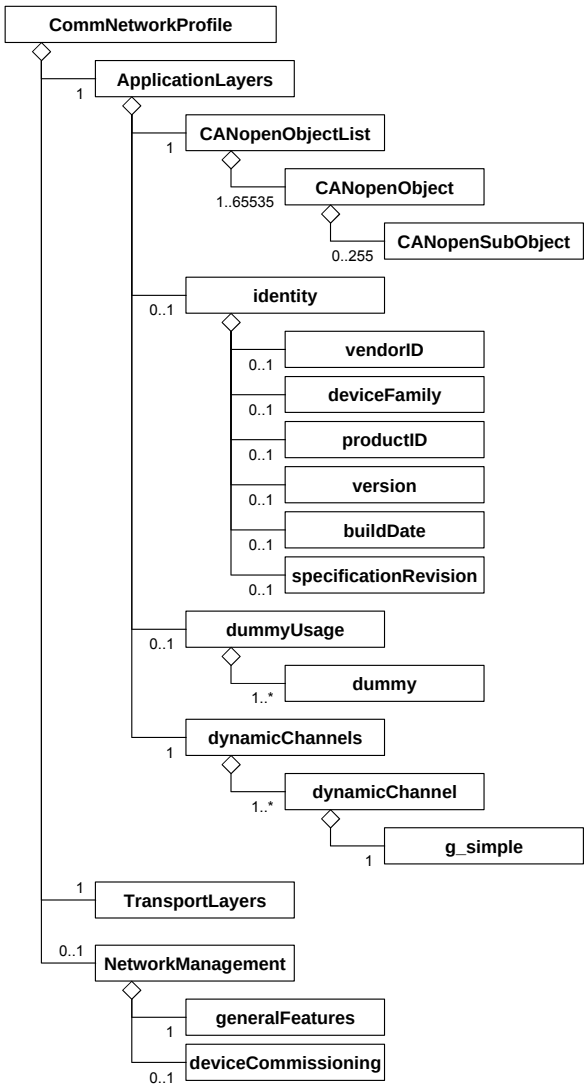


Figure 31 — EtherCAT communication network profile class diagram

The XML schema representing the EtherCAT communication network profile is defined in F.5.5.

6.6.2.2 Application layers

The EtherCAT ApplicationLayers class represents the combined profiles for the upper 3 OSI layers of the EtherCAT communication network integration model.

Further details are given in F.5.2.

6.6.2.3 Transport layers

The EtherCAT TransportLayers class represents the combined profiles for the lower 4 OSI layers of the EtherCAT communication network integration model.

Further details are given in F.5.3.

6.6.2.4 Network management

The EtherCAT NetworkManagement class represents the network configuration and performance adjustment capabilities of the EtherCAT communication network integration model.

Further details are given in F.5.4.

6.7 ETHERNET Powerlink

6.7.1 Device profile

6.7.1.1 General

Figure 32 shows the class structure of an ETHERNET Powerlink device profile.

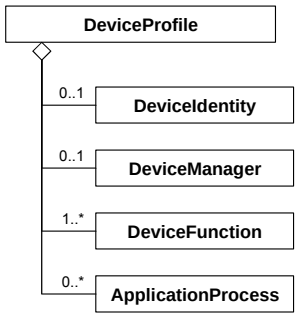


Figure 32 — ETHERNET Powerlink device profile class diagram

NOTE The ETHERNET Powerlink device profile class diagram shown in Figure 32 defines the main classes. These classes are further decomposed and detailed in Annex G.

The XML schema representing the ETHERNET Powerlink device profile template is defined in G.4.6. The template is based on two parts:

— the EPL profile header defined in G.3, and

— the EPL device profile defined in G.4.

6.7.1.2 Device identity

The DeviceIdentity class contains attributes that are independent of the network, of the process and uniquely identify the device.

Figure 33 shows the structure of the ETHERNET Powerlink DeviceIdentity class.

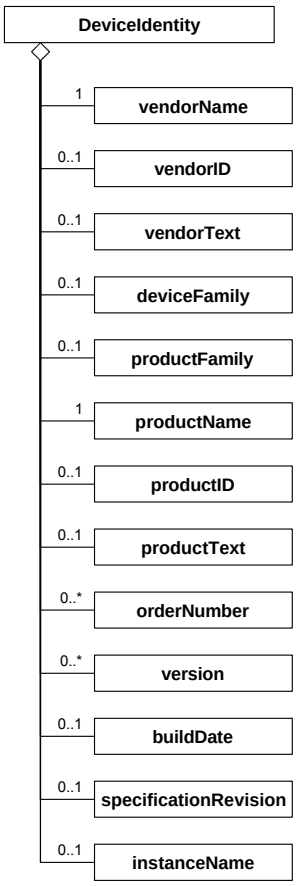


Figure 33 — ETHERNET Powerlink DeviceIdentity class diagram

Further details are given in G.4.2.

6.7.1.3 Device manager

The DeviceManager class contains attributes and supports services that enable the monitoring of the device. Communication specific configuration data and mapping information is defined in the communication network specific part structured according to the schema specified in G.5.

Figure 34 shows the structure of the ETHERNET Powerlink DeviceManager class.

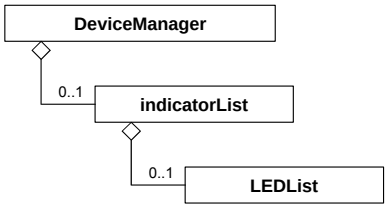


Figure 34 — ETHERNET Powerlink DeviceManager class diagram

Further details are given in G.4.3.

6.7.1.4 Device function

The DeviceFunction class describes the intrinsic function of a device in terms of its technology. It contains network independent descriptions/definitions of the technological device functionality.

Figure 35 shows the structure of the ETHERNET Powerlink DeviceFunction class.

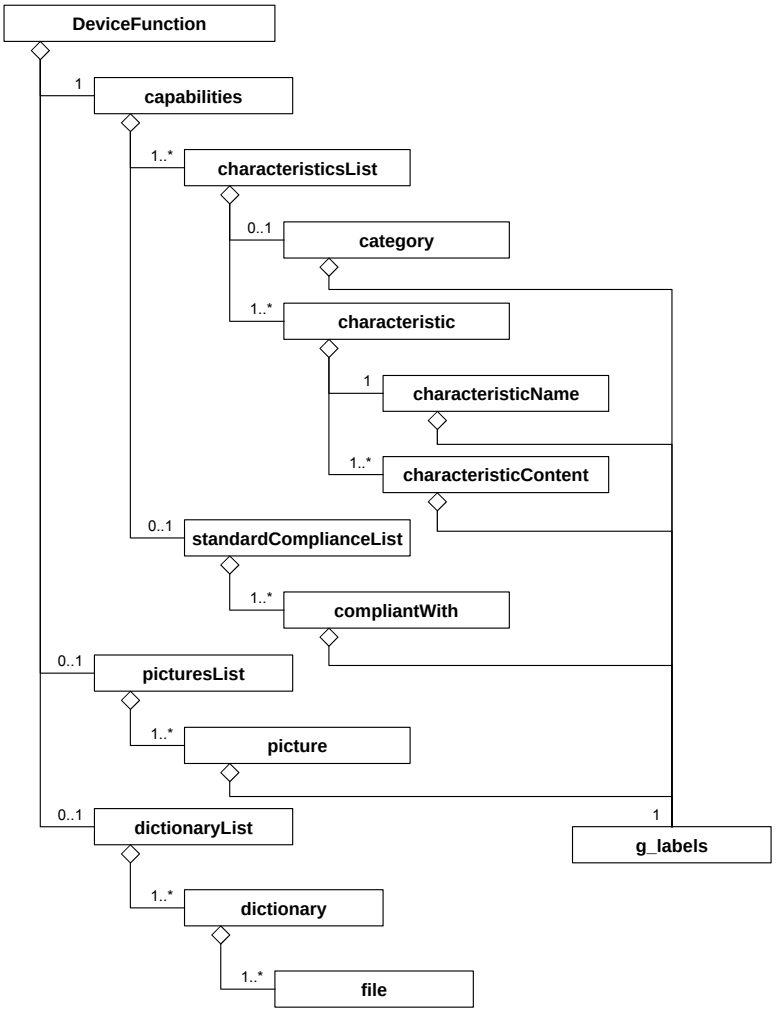


Figure 35 — ETHERNET Powerlink DeviceFunction class diagram

Further details are given in G.4.4.

6.7.1.5 Application process

The ApplicationProcess class represents the set of services and parameters, which constitute the behaviour, and the interfaces of the device in terms of the application, independent of the device technology and the underlying communication networks and communication protocols.

Figure 36 shows the structure of the ETHERNET Powerlink ApplicationProcess class.

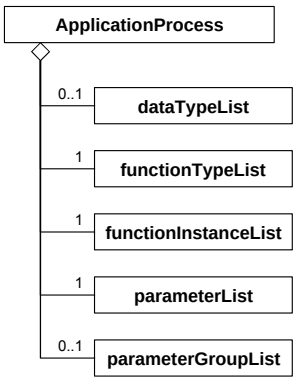


Figure 36 — ETHERNET Powerlink ApplicationProcess class diagram

Further details are given in G.4.5.

6.7.2 Communication network profile

6.7.2.1 General

Figure 37 shows the class structure of the ETHERNET Powerlink communication network profile. These classes are further decomposed and detailed in Annex G.

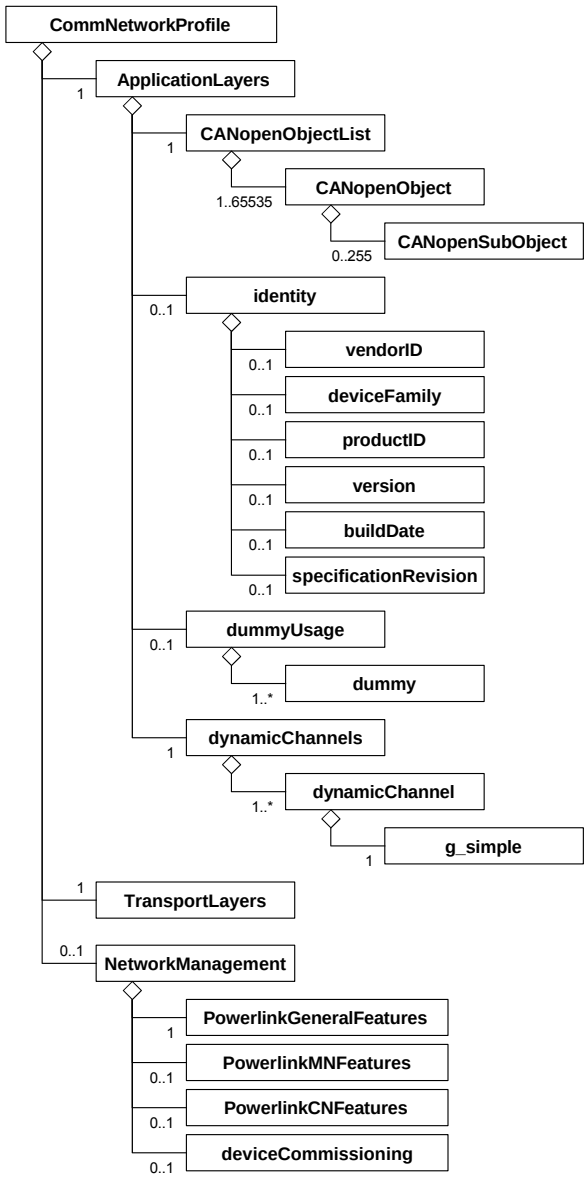


Figure 37 — ETHERNET Powerlink communication network profile class diagram

The XML schema representing the ETHERNET Powerlink communication network profile is defined in G.5.5.

6.7.2.2 Application layers

The ETHERNET Powerlink ApplicationLayers class represents the combined profiles for the upper 3 OSI layers of the ETHERNET Powerlink communication network integration model.

Further details are given in G.5.2.

6.7.2.3 Transport layers

The ETHERNET Powerlink TransportLayers class represents the combined profiles for the lower 4 OSI layers of the ETHERNET Powerlink communication network integration model.

Further details are given in G.5.3.

6.7.2.4 Network management

The ETHERNET Powerlink NetworkManagement class represents the network configuration and performance adjustment capabilities of the ETHERNET Powerlink communication network integration model.

Further details are given in G.5.4.

Page 125

Insert the following new Annex E, Annex F and Annex G after Annex D inserted from Amendment 1 to ISO 15745-4:2003, and before the Bibliography.

Annex E
(normative)

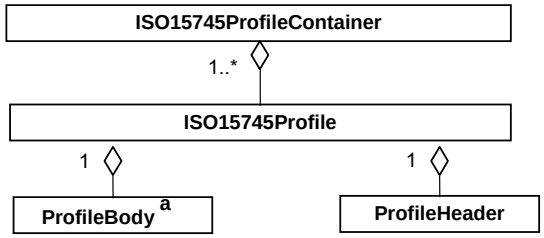
Modbus TCP profile templates

E.1 Overview

Modbus TCP is an Ethernet based communication system specified in IEC/PAS 62030.

Modbus TCP uses the concept of the multi-profile container specified in Amendment 1 to ISO 15745-4:2003 for XML profile files. Therefore, Modbus TCP profile templates are based on the alternate ISO15745ProfileContainer template specified in amendment 1[6)] of ISO 15745-1.

Figure E.1 shows the structure of a Modbus TCP XML profile.



Two types of ProfileBody are used:
ProfileBody_Device_ModbusTCP or ProfileBody_CommunicationNetwork_ModbusTCP
a

Figure E.1 — Modbus TCP profile template

The ProfileTechnology name is DDXML (Device Description eXtensible Markup Language).

E.2 General rules

E.2.1 Using unique IDs

An element can have the attribute uniqueID of type xsd:ID. The unique identifier therefore is forced to be unique in the whole XML file. An element that references the unique identifier contains a named attribute of type xsd:IDREF.

Unique identifiers may be generated in two ways. One possibility is to build a string out of the element name and an up-counting number. A second way may be the concatenation of strings of parent elements. Both methods guarantee the uniqueness of the string.

E.2.2 Language support

E.2.2.1 General

Device profiles complying with the XML schema described in this annex need a support of different languages, since tools are then able to use names out of the XML file in order to display them in their user interface. Communication parameters for example may be presented in the user interface of a tool.

The language support is implemented via the label group g_labels. Each name of an element, which would possibly be displayed and is therefore language dependent, is provided inside the schema as a g_labels element. Optionally, a URI may be added as an attribute to the label element.

EXAMPLE

For a given parameter name:

— German: Baudrate

— English: Baud rate

— French: Vitesse de transmission

E.2.2.2 Element g_labels

The group g_labels supports the introduction of a label (name) and a description in the context of the parent element (see Figure E.2).

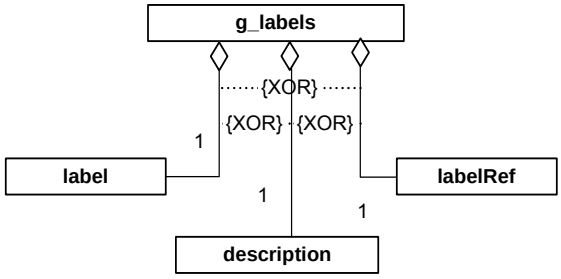


Figure E.2 — Group g_labels

Every element, which needs a name or a description, shall select one and only one of the three elements to perform this task: the label, the description and the labelRef element.

1) The label element allows storage of the identifying name and the descriptive text inside the XML file itself. The label element has the attributes given in Table E.1.

Table E.1 — Attributes of element label

Attribute	Data type	Use	Description
lang	xsd:language	required	Language used for the name or the description
URI	xsd:anyURI	optional	Optional link to further descriptive information

The element may appear n times, once for each language. For identifying the language, the lang attribute is used.

2) The description element allows storage of textual descriptions inside the XML file itself. The element may appear several times, once for each language. The description element has the same attributes as the label element.

3) The labelRef element allows storage of reference descriptive texts inside an external text resource file.

   The labelRef element provides a pointer via its attributes dictID and textID to a text entry in a separate text source file. These text source files are referenced in the dictionary sub-elements of the DeviceFunction element. Text source files may be any files containing character sequences and other information, for example figures.

   The labelRef element also may appear n times, to allow references to several dictionary entries, which contain links to files in different languages. The respective language is defined in the lang attribute of the dictionary element.

   The labelRef element contains the attributes given in Table E.2.

Table E.2 — Attributes of element labelRef

Attribute	Data type	Use	Description
dictID	xsd:IDREF	required	References a single dictionary element inside the dictionaryList element; the dictionary element contains a link to the external text source file
textID	xsd:string	optional	References a character sequence inside the external text source file by pattern matching

E.2.2.3 Language identifier

For the multi-language support each label gets an attribute with the content of the language code. The language code corresponds to the content of the label element.

In order to verify which languages are supported in the XML file, the attribute supportedLanguages in the ProfileBody element lists the supported languages.

E.2.2.4 Attribute lang

The language identifier lang consists of a combination of a language code (as defined in ISO 639-1) plus an optional dash character plus an optional country code (as defined in ISO 3166-1). The attribute lang is an attribute of the label element.

Some of the values for lang are given in Table E.3.

Table E.3 — Values of attribute lang

Language	value of lang
English (United States)	en-us
German (Standard)	de
French (Standard)	fr
Spanish (Standard)	es
Italian (Standard)	It
Portuguese (Brazil)	pt-br

E.2.2.5 SupportedLanguages attribute

The supportedLanguages attribute identifies supported languages and consists of a list of language codes plus optional country codes.

EXAMPLE

supportedLanguages=”en-us de fr es”

E.2.2.6 URIs

A general mechanism allows of describing a URI in the context of a label element. The URI is implemented via an optional attribute URI.

EXAMPLE: This is used in the context of a vendor label, parameter label, or services label.

E.3 ProfileHeader description

To facilitate the identification of a profile, the profile header of the device profile as well as the communication network profile shall comply with the model shown in Figure E.3, which is directly inherited from ISO 15745-1.

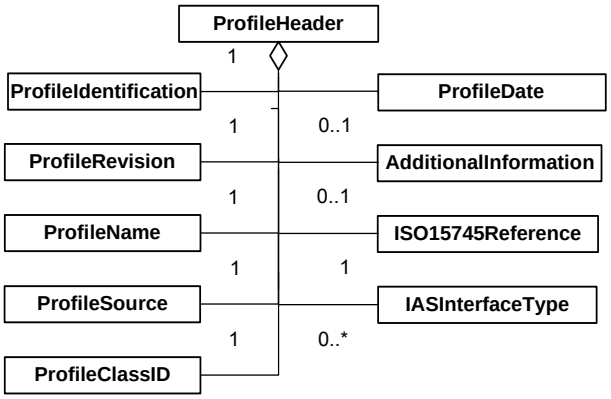


Figure E.3 — Profile header class diagram

The ProfileHeader element is composed of the following elements:

— the ProfileIdentification element identifies the current profile,

— the ProfileRevision element identifies the current profile revision,

— the ProfileName element contains a descriptive English name of the current profile. In case that more than one ProfileBody element are present within a device profile, it is suggested that the value of the ProfileName element should be the concatenation of the values of the productName elements inside the respective DeviceIdentity elements,

— the ProfileSource element identifies the validator of the current profile,

— the ProfileClassID element identifies the class of the current profile according to ISO 15745-1,

— the ISO15745Reference element states the ISO 15745 part, edition and technology, to which the description conforms.

E.4 Device profile template description

E.4.1 ProfileBody_Device_ModbusTCP

This part defines the Modbus TCP device profile.

The ProfileBody_Device_ModbusTCP contains the DeviceIdentity, the DeviceManager, the DeviceFunction and the ApplicationProcess elements shown in Figure 20.

The ProfileBody element contains the description of

— a single device (for example a proximity sensor or an electromechanical limit switch), or of a more complex one (for example a circuit breaker with up to 2500 parameters, more than 100 functions), or

— a part of a device also called "module" in the PLC world (for example a part of an I/O controller or an electrical protection unit).

The ProfileBody element contains the attributes given in Table E.4.

Table E.4 — Attributes of element ProfileBody

Attribute	Data type	Use	Description
formatName	xsd:string	fixed	Format identifier
formatVersion	xsd:string	fixed	Format version identifier
fileName	xsd:string	required	Name of the file with extension without path
fileCreator	xsd:string	required	Person creating the file
fileCreationDate	xsd:date	required	Date of file creation
fileCreationTime	xsd:time	optional	Time of file creation
fileModifiedBy	xsd:string	optional	Person modifying the file
fileModificationDate	xsd.date	optional	Date of last file change
fileModificationTime	xsd:time	optional	Time of last file change
fileVersion	xsd:string	required	Vendor specific version of the file
supportedLanguages	xsd:NMTOKENS	optional	List of supported languages

E.4.2 DeviceIdentity

E.4.2.1 General

The DeviceIdentity class (see Figure 21) contains elements, which are independent of the network and of the process. It describes the identity of a single device or of a group of devices.

Table E.5 specifies the attribute readOnly, which is attached to the vendorName, vendorID, vendorText, deviceFamily, productFamily, productName, productID, productText, orderNumber, version, specificationRevision, and instanceName elements.

Table E.5 — Attribute of element vendorName

Attribute	Data type	Use	Description
readOnly	xsd:boolean	default	Indicates whether the value is read-only for a user: false, true (default)

E.4.2.2 Element vendorName

The vendorName element identifies the name or the brand name of the vendor of the device.

E.4.2.3 Element vendorID

The vendorID element identifies the vendor. This information has to be filled in when the product described is recognized and validated by a consortium.

NOTE Consortia specific product families and vendor identifiers are linked.

E.4.2.4 Element vendorText

The vendorText element allows the vendor to provide additional company information, like address or hotline number. The g_labels group offers the possibility to include a vendor URI inside the vendorText element.

E.4.2.5 Element deviceFamily

The deviceFamily element states the family of the device.

EXAMPLE

Examples for device families are:

— Variable speed drive

— Circuit breaker

— Pressure sensor

E.4.2.6 Element productFamily

The productFamily element states a vendor specific affiliation of the device type to a certain set of devices inside a family. The list of valid productFamily values is system, tool or consortia specific.

NOTE Consortia specific product families and vendor identifiers are linked.

E.4.2.7 Element productName

The productName element states a vendor specific designation or name of the device type.

E.4.2.8 Element productID

The productID element states a vendor specific unique identification for the device type described.

E.4.2.9 Element productText

The productText element allows the vendor to provide a short textual description of the device type.

E.4.2.10 Element orderNumber

The orderNumber element is used to store the single order number of a given product or the set of different order numbers of the products of a product family, depending upon whether the device profile describes a product or a product family.

E.4.2.11 Element version

The version element is used to store different types of version information. Multiple version elements are possible.

The version element has the attributes given in Table E.6.

Table E.6 — Attributes of element version

Attribute	Data type	Use	Description
versionType	xsd:NMTOKEN	required	Type of version: — SW – Software — FW – Firmware — HW – Hardware
readOnly	xsd:boolean	default	Indicates whether the value is read-only for a user: false, true (default)

E.4.2.12 Element buildDate

The buildDate element specifies the build date of the software unit.

E.4.2.13 Element specificationRevision

The specificationRevision element contains the revision of the specification, to which this device conforms.

E.4.2.14 Element instanceName

This element contains the instance name of the device.

E.4.3 DeviceManager

E.4.3.1 General

The DeviceManager element defines the list of indicators provided by the device type, if any.

E.4.3.2 Elements indicatorList / LEDList

E.4.3.2.1 General

Figure E.4 specifies the number and type of indicators, which are provided by a device type.

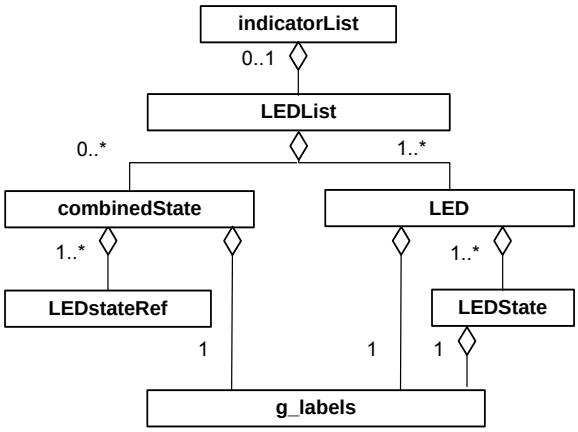


Figure E.4 — indicatorList / LEDList

E.4.3.2.2 LED

The LED element describes the features of a single LED of the device type. A detailed feature description may be provided through the g_labels group.

Further properties of the LED are represented as attributes of the LED element given in Table E.7.

Table E.7 — Attributes of element LED

Attribute	Data type	Use	Description
LEDcolors	xsd:string	required	Colours of the LED; valid values are monocolor and bicolor
LEDtype	xsd:string	optional	Rough classification of the supervised item or functionality; valid values are IO, device and communication

In addition to the descriptive parts introduced above, the LED element contains one to many LEDstate elements, which define the device states signalled by the LED and the visual behaviour used for signalling the states.

The visual behaviour used for signalling the state is encoded as attribute values of the LEDstate element, as given in Table E.8. Additionally a unique ID is allocated for the LED state.

Table E.8 — Attributes of element LEDstate

Attribute	Data type	Use	Description
uniqueID	xsd:ID	required	Unique ID for the LED state; may be referenced from an LEDstateRef element
state	xsd:string	required	State of the LED; possible attribute values: On, off, flashing
LEDcolor	xsd:string	required	Colour of the LED state; valid values: green, amber, red
flashingPeriod	xsd:unsignedInt	optional	If state is flashing: flashing period of the LED in milliseconds
impulsWidth	xsd:unsignedByte	default	Width of the flashing impulse given in percent (%) of the flashing period; if the attribute impulsWidth is missing, the default value is 50 (%)
numberOfImpulses	xsd:unsignedByte	default	Number of impulses in case that more than one flashing impulse is inside one flashing period; if the attribute is present, the attribute impulsWidth shall be present also if the attribute numberOfImpulses is missing, the default value is 1

E.4.3.2.3 Element combinedState

The combinedState element allows the indication of device states which are signaled by more than one LED.

The description of the combined state is provided through the g_labels group.

The LED states participating in the signalling of the combined state are referenced by means of at least two LEDstateRef sub-elements of the combinedState element.

The reference to a LEDstate element is encoded as the attribute value of the single attribute of the LEDstateRef element (see Table E.9).

Table E.9 — Attributes of element LEDstateRef

Attribute	Data type	Use	Description
stateIDRef	xsd:IDREF	required	Unique ID of the referenced LEDstate element

E.4.4 DeviceFunction

E.4.4.1 General

The DeviceFunction element shown in Figure 23 defines the catalogue view of the device, presented as a set of capabilities listing both device characteristics and compliance with various standards.

E.4.4.2 Element capabilities

E.4.4.2.1 General

The mandatory capabilities element describes all functionalities, their characteristics, and the important parameters of the device, that need to be known by tools which use the device profile to select products with the same or similar properties.

The capabilities element describes device features in a purely textual form. It contains a sequence of one to many characteristicsList elements and an optional standardComplianceList element.

E.4.4.2.2 Element characteristicsList

E.4.4.2.2.1 General

The characteristicsList element is a collection of characteristics. The element shall contain at least one characteristic sub-element. The characteristics inside a list may be associated with a category, which can be expressed as textual content of the g_labels sub-element of the optional category sub-element of the characteristicsList element.

E.4.4.2.2.2 Element characteristic

The characteristic element describes a single characteristic of a device. It contains a mandatory characteristicName element and one to many characteristicContent elements.

E.4.4.2.2.3 Element characteristicName

The mandatory characteristicName element denotes a major technical characteristic of the device. The vocabulary used in the product data sheet is recommended for the names of characteristics.

EXAMPLE

"Maximum operational voltage", "Overload protection", "Electrical durability".

E.4.4.2.2.4 Element characteristicContent

This mandatory element contains a value for the characteristic. Multiple values may be expressed by using multiple characteristicContent elements.

EXAMPLE

An example of a single value for "Maximum operational voltage" is 680V.

E.4.4.2.3 Element standardComplianceList

The standardComplianceList element is a collection of compliantWith elements. The element itself is optional; if it exists, it shall contain at least one compliantWith sub-element.

The compliantWith sub-element has attributes, which state the compliance of the device with an international or company internal standard. The content of type g_labels of this element may contain remarks concerning that standard.

The name or number of the standard is provided through the required name attribute of the compliantWith element. The second, default valued range attribute of the compliantWith element defines the range of applicability of the standard as given in Table E.10.

Table E.10 — Attributes of element compliantWith

Attribute	Data type	Use	Description
name	xsd:string	required	Name or number of the standard
range	xsd:NMTOKEN	default	The two possible enumerated values of the attribute are international (default) or internal

E.4.4.3 Element picturesList

The picturesList element offers the possibility to link pictures to the device profile. It contains one to many picture sub-elements, whose caption is provided via a g_labels sub-element.

Table E.11 defines attributes of the picture sub-element: an optional number of the picture, and the mandatory link to an external resource containing the graphical information.

Table E.11 — Attributes of element picture

Attribute	Data type	Use	Description
URI	xsd:anyURI	required	Link to the external resource
number	xsd:unsignedInt	optional	Number of the picture

E.4.4.4 Element dictionaryList

The optional dictionaryList element offers the possibility to include links to external text source files to the device profile. It contains one to many dictionary elements, where each one contains one to many file sub-elements. Several files are necessary if different file formats are needed within a dictionary.

A mandatory lang attribute of type xsd:language defines the language used in the files which are linked to the dictionary element (see Table E.12). A mandatory uniqueID attribute of type xsd:ID holds the unique identification of the dictionary entry which is referenced from the attribute dictID of a labelRef element as given in Table E.2.

Table E.12 — Attributes of element dictionary

Attribute	Data type	Use	Description
lang	xsd:language	required	Language used for files belonging to a dictionary entry
uniqueID	xsd:ID	required	Unique ID of the dictionary entry

A file sub-element contains a single mandatory attribute given in Table E.13.

Table E.13 — Attributes of element file

Attribute	Data type	Use	Description
URI	xsd:anyURI	required	Link to the respective file

E.4.5 ApplicationProcess

E.4.5.1 General

The ApplicationProcess element represents the set of services and parameters which constitute the behavior and the interfaces of the device in terms of the application, independent of the device technology and the underlying communication networks and communication protocols.

The sub-elements of the ApplicationProcess element in Figure 24 provide a generic approach for the description of arbitrary, flat or hierarchically structured functions of a device.

Functions are modeled as function types, which are instantiated within the device or - if hierarchical structures are needed - inside function types. The interface variables of these function instances, which may be of simple or complex data type, are associated with the parameters of the device by building a reference from the parameter to the respective interface variable of the function instance, in flat as well as in hierarchical structures.

The ApplicationProcess element contains up to five lists of items (see Figure 24):

— two lists which define data types (optional) and function types (required),

— one required list which defines the function instances on device level (possibly including connections between instances),

— one required list which defines the device parameters, and

— one optional list which defines parameter groups (combinations of parameters for specific purposes).

E.4.5.2 Element dataTypeList

E.4.5.2.1 General

The optional dataTypeList element is present if complex data types like arrays or data structures are needed inside variable declarations of the device profile.

If present, the dataTypeList element shown in Figure E.5 contains a sequence of one to many elements out of the choice of:

— an array element,

— a struct element,

— an enum element, or

— a derived element.

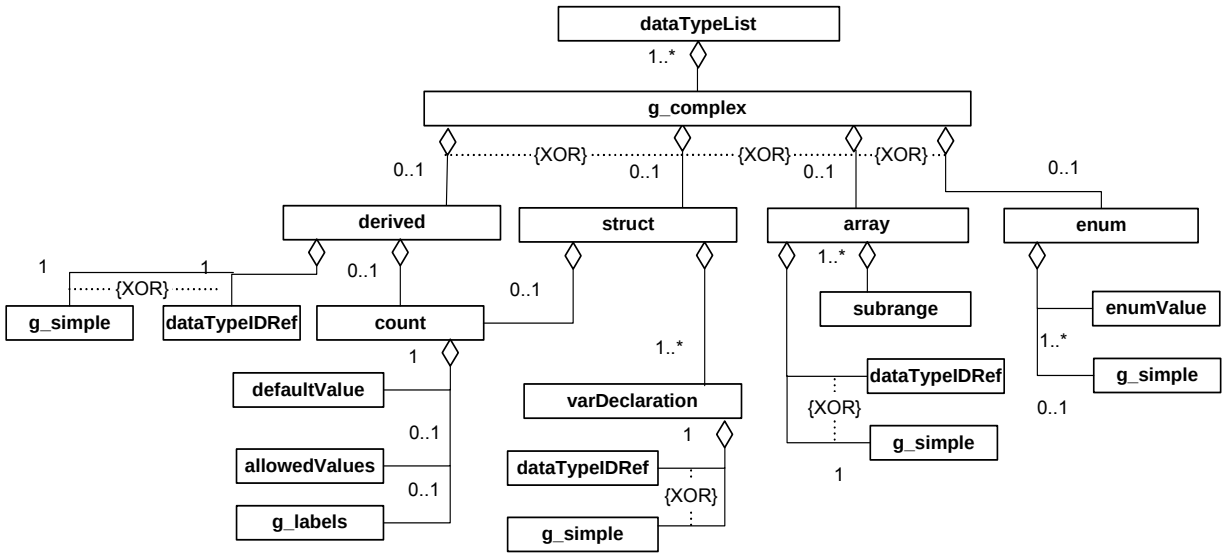


Figure E.5 — dataTypeList

E.4.5.2.2 Common elements

E.4.5.2.2.1 Group g_simple

The group g_simple contains a choice of elements, whose names represent the names of all simple data types allowed in the definition of variables inside a device profile. The simple data types conform to the elementary data types defined in IEC 61131-3; the data types BITSTRING and CHAR (=STRING[1]) are added.

These elements are introduced inside a group to allow their placement directly as a sub-element of the array element (or of the varDeclaration element, see E.4.5.4.3.2).

E.4.5.2.2.2 Element count

The count element defines the number of used units of the base type of the derived type. Multilingual names and/or descriptions for the count element are provided through the group g_labels. See E.2.2.2 for the description of the group g_labels.

The count is described by:

— its attributes,

— the mandatory sub-element defaultValue and a possibly empty set of sub-elements g_labels and allowedValues.

The number of units is expressed as the value of the defaultValue attribute of the count element. The allowedValue attribute defines the range of values for the default value.

The sub-elements defaultValue and allowedValues are described in E.4.5.6.2.5 and in E.4.5.6.2.7.

The count element shall contain the attributes given in Table E.14.

Table E.14 — Attributes of element count

Attribute	Data type	Use	Description
uniqueID	xsd:ID	required	Unique ID of the count
access	xsd:NMTOKEN	default	Defines which operations are valid for the count: — read - read access only (default value) — write - write access only — readWrite - both read and write access — noAccess - access denied

E.4.5.2.3 Element array

E.4.5.2.3.1 General

The array element serves to describe an array data type, which may be referenced from an interface variable of a function type, from another array type definition, or from a component variable inside the definition of a structured data type.

The array element contains at least one subrange element and either an element describing a simple data type out of the group g_simple, or an element dataTypeIDRef, which references one of the defined complex data types within the dataTypeList element.

For multi-dimensional arrays, several subrange elements will be present. In this case, the first subrange element in the sequence defines the subrange for the leftmost array index, and the last subrange element in the sequence defines the subrange for the rightmost array index.

The array element contains the attributes given in Table E.15.

Table E.15 — Attributes of element array

Attribute	Data type	Use	Description
name	xsd:string	required	Data type name of the array type
uniqueID	xsd:ID	required	Unique ID of the array type
description	xsd:string	optional	Optional textual description of the array type

E.4.5.2.3.2 Element subrange

The subrange element defines the lower and the upper limit of an array index for one dimension of the array. This element has no sub-elements.

The limit values of type xsd:long are contained in the two attributes of the subrange element given Table E.16.

Table E.16 — Attributes of element subRange

Attribute	Data type	Use	Description
lowerLimit	xsd:long	required	Lower limit of the subrange
upperLimit	xsd:long	required	Upper limit of the subrange

E.4.5.2.4 Element struct

E.4.5.2.4.1 General

The struct element serves to describe a structured data type, which may be referenced from an interface variable of a function type, from an array type definition, or from a component variable inside the definition of another structured data type.

The struct element contains a sequence of one to many varDeclaration elements, which define the components of the structured data type.

The struct element shall contain the attributes given in Table E.17.

Table E.17 — Attributes of element struct

Attribute	Data type	Use	Description
name	xsd:string	required	Data type name of the structured data type
uniqueID	xsd:ID	required	Unique ID of the structured data type
description	xsd:string	optional	Optional textual description of the structured data type

E.4.5.2.4.2 Element varDeclaration

In the context of the definition of a structured data type, the varDeclaration element describes a single component variable (member) of the structure.

In the context of the definition of the interface of a function, the varDeclaration element describes a single interface variable of the function type.

The data type of the component variable or interface variable is either defined by an element describing a simple data type out of the group g_simple, or by an element dataTypeIDRef, which references one of the defined complex data types within the dataTypeList element.

All further properties of the variable are contained in the attributes of the varDeclaration element, as given in Table E.18.

Table E.18 — Attributes of element varDeclaration

Attribute	Data type	Use	Description
name	xsd:string	required	Name of the interface variable or structure component
uniqueID	xsd:ID	required	Unique ID of the interface variable or structure component (see NOTE 1)
size	xsd:string	optional	Number of elements, if the interface variable or structure component is of type anonymous ARRAY, BITSTRING, STRING or WSTRING (see NOTE 2)
initialValue	xsd:string	optional	Initial value of the interface variable or structure component (see NOTE 3)
description	xsd:string	optional	Optional textual description of the interface variable or structure component

NOTE 1 When creating the unique ID for a variable, it is essential that the ID is unique over all created IDs inside the XML source file. To allow equal names for component variables of different data structures and equal names for interface variables of function types, the ID of a variable should generally concatenate the type name of the structured data type or the function type with the variable name, to guarantee uniqueness.

NOTE 2 Anonymous types define the size of an array, bitstring or string directly in the variable declaration, and not through the reference to a named complex data type. In the case of an array, the type of a single array element is given by the data type of the variable. In the case of a bitstring, the single array element is a single bit. In the case of a string, the single array element is a single-byte resp. double-byte character.

NOTE 3 If present, this attribute defines the initial (default) value of the interface variable of the function type. It is overwritten by a given default value of a parameter associated with the interface variable of the function instance.

E.4.5.2.5 Element enum

E.4.5.2.5.1 General

The enum element serves to describe an enumerated data type, which may be referenced from an interface variable of a function type, from an array type definition, or from a component variable inside the definition of a structured data type.

According to Figure E.5, it contains a sequence of one to many enumValue elements, which define the enumeration constants of the enumerated data type. The data type of the enumeration constants is optionally defined by an element describing a simple data type out of the group g_simple.

The enum element contains the attributes given in Table E.19.

Table E.19 — Attributes of element enum

Attribute	Data type	Use	Description
name	xsd:string	required	Type name of the enumerated data type
uniqueID	xsd:ID	required	Unique ID of the enumerated data type
size	xsd:string	optional	Optional number of enumerated values of the enumerated data type
description	xsd:string	optional	Optional textual description of the enumerated data type

E.4.5.2.5.2 Element enumValue

The enumValue element defines the name(s) and optionally a numerical value of a single enumeration constant. The name(s) are specified through the g_labels group, whereas the value is contained in the single value attribute of the enumValue element, as given in Table E.20.

Table E.20 — Attributes of element enumValue

Attribute	Data type	Use	Description
value	xsd:string	optional	Optional attribute: fixed numerical value for the enumeration constant, represented as a string of characters

E.4.5.2.6 Element derived

The derived element serves to derive a new data type from a given base type.

The derived element contains an optional count element and either an element describing a simple data type out of the group g_simple, or an element dataTypeIDRef, which references one of the defined complex data types within the dataTypeList element.

If the count element is missing, the derived type definition just introduces a new type name for the respective base type. If the count element is present, it defines the number of units of the respective base type used to build the derived type (e.g base type BITSTRING, count = 4 defines a derived type of size 4 bit).

The derived element contains the attributes given in Table E.21.

Table E.21 — Attributes of element derived

Attribute	Data type	Use	Description
name	xsd:string	required	Data type name of the derived type
uniqueID	xsd:ID	required	Unique ID of the derived type
description	xsd:string	optional	Optional textual description of the derived type

E.4.5.3 Element functionTypeList

If the optional ApplicationProcess element is present in the device profile, it contains a mandatory functionTypeList element shown in Figure E.6.

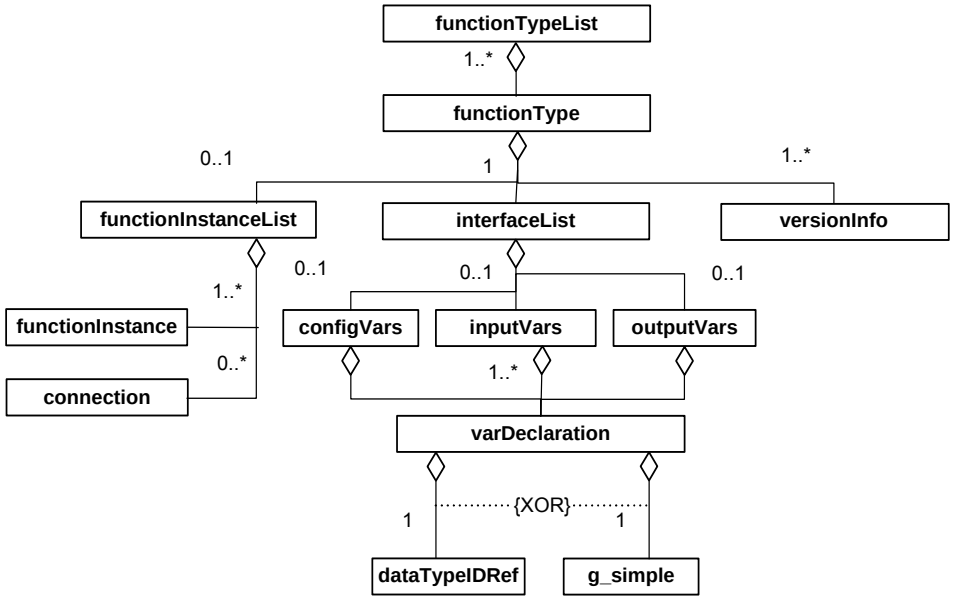


Figure E.6 — functionTypeList

The functionTypeList represents a sequence of one to many functionType elements.

Each of the functionType elements represents the type description of a device function, which is referenced from at least one instance of that function type inside a functionInstanceList element. References from more than one instance of the same function type are also possible.

The description of a function type contains all those objects and data which are common for all instances of a given function type.

EXAMPLE 1

Examples are the variable - or function parameter - objects that constitute the interface of the function (type respectively instance).

EXAMPLE 2

Other examples are instances contained within the body of a function in a hierarchically structured functional description. These instances, which are located within a functionInstanceList element inside the function type, reference other function types in the list of function types.

E.4.5.4 Element functionType

E.4.5.4.1 General

The functionType element contains one to many versionInfo elements, a mandatory interfaceList element and an optional functionInstanceList element. The functionInstanceList element is only present within a functionType element, if the function is hierarchically structured.

Additionally, the functionType element shall contain the attributes given in Table E.22.

Table E.22 — Attributes of element functionType

Attribute	Data type	Use	Description
name	xsd:string	required	Type name of the function type
uniqueID	xsd:ID	required	Unique ID of the function type
description	xsd:string	optional	Optional textual description of the function type
package	xsd:string	optional	Optional textual association of the function type with a "package" or similar classification scheme - the usage of this attribute is left to the profile validator

E.4.5.4.2 Element versionInfo

The mandatory versionInfo element within the functionType element provides information on the versioning history of a function type (concerning the definition of the interface).

To keep track of the versioning history, the versionInfo element may be entered multiple times. The multiple entries shall be arranged within the functionType element in the following sequence:

a) the first entry represents the most recent version,

b) the second entry represents the immediately preceding version,

c) the last entry represents the first released version.

This element will be provided once at the creation of the description of the function type. New elements will only be added, if modifications of a function type are introduced, which lead to a modified version of the device profile.

The versionInfo element shall contain the attributes given in Table E.23.

Table E.23 — Attributes of element versionInfo

Attribute	Data type	Use	Description
organization	xsd:string	required	Name of the organisation maintaining the function type
version	xsd:string	required	Version identification in the versioning history; suggested format: "xx.yy" (xx,yy = 0..255)
author	xsd:string	required	Name of the person maintaining the function type
date	xsd:date	required	Date of this version
remarks	xsd:string	optional	Descriptive information concerning the specific step in the versioning history

E.4.5.4.3 Element interfaceList

E.4.5.4.3.1 General

The mandatory interfaceList element within the functionType element provides the definition of the interface of the function type. Elements of the interface are:

— the input variables, and/or

— the output variables, and/or

— the configuration variables

of the function type.

Consequently the interfaceList element contains a sequence of three elements, where each element represents lists of one to many variable declarations encoded as varDeclaration elements:

— one optional element inputVars,

— one optional element outputVars, and

— one optional element configVars.

Neither the interfaceList nor the inputVars, outputVars or configVars elements have any attributes.

E.4.5.4.3.2 Element varDeclaration

In the context of the definition of a structured data type, the varDeclaration element describes a single component variable (member) of the structure.

In the context of the definition of the interface of a function type, the varDeclaration element describes a single interface variable of the function type.

The data type of the component variable or interface variable is either defined by an element describing a simple data type out of the group g_simple, or by an element dataTypeIDRef, which references one of the defined complex data types within the dataTypeList element.

E.4.5.2.2.1 describes the group g_simple and E.4.5.4.3.3 describes the dataTypeIDRef element.

All further properties of the variable are contained in the attributes of the varDeclaration element, as given in Table E.24.

Table E.24 — Attributes of element varDeclaration

Attribute	Data type	Use	Description
name	xsd:string	required	Name of the interface variable or structure component
uniqueID	xsd:ID	required	Unique ID of the interface variable or structure component
size	xsd:string	optional	Number of elements, if the interface variable or structure component is of type anonymous ARRAY, BITSTRING, STRING or WSTRING
initialValue	xsd:string	optional	Initial value of the interface variable or structure component
description	xsd:string	optional	Optional textual description of the interface variable or structure component

E.4.5.4.3.3 Element dataTypeIDRef

The dataTypeIDRef element serves to reference a complex data type inside the dataTypeList element (see E.4.5.2), either from an interface variable of a function type, or from an array type definition, or from a component variable inside the definition of a structured data type.

The reference of type xsd:IDREF is provided as an attribute of the dataTypeIDRef element, as given in Table E.25.

Table E.25 — Attributes of element dataTypeIDRef

Attribute	Data type	Use	Description
uniqueIDRef	xsd:IDREF	required	Unique ID of the referenced data type

E.4.5.5 Element functionInstanceList

E.4.5.5.1 General

If the optional ApplicationProcess element is present in the device profile, it contains a mandatory functionInstanceList element, which contains a sequence of one to many functionInstance elements and zero to many connection elements.

At the application process level, the functionInstance elements represent the accessible application functions of the device type, independent of the network type or protocol. The connection elements represent connections - if any - between specific output and input variables of those function instances.

The functionInstanceList element also appears as an optional sub-element of the functionType element (see E.4.5.4). Like at the application process level, the functionInstanceList element in that case contains a sequence of one to many functionInstance elements and zero to many connection elements.

The functionInstanceList element is only present within a functionType element, if a function is hierarchically structured. In this case the functionInstance elements represent the internal functions contained in the function type, and the connection elements the optional internal connections. These functions and their optional connections would be instantiated together with the instantiation of the containing function type.

The functionInstanceList element does not have any attributes.

E.4.5.5.2 Element functionInstance

The mandatory functionInstance element does not contain any sub-elements.

The functionInstance element shall contain the attributes given in Table E.26.

Table E.26 — Attributes of element functionInstance

Attribute	Data type	Use	Description
name	xsd:string	required	Name of the function instance
uniqueID	xsd:ID	required	Unique ID of the function instance (see NOTE)
typeIDRef	xsd:IDREF	required	Unique ID of the referenced function type
description	xsd:string	optional	Optional textual description of the function instance
NOTE When creating the unique ID for a function instance, it is essential that the ID is unique over all created IDs inside the XML source file. To allow equal names for function instances inside different function types, the ID of a function instance should generally concatenate the name of the containing function type with the instance name, to guarantee uniqueness.			

E.4.5.5.3 Element connection

The optional connection element defines a connection between an output variable of a function instance and an input variable of another function instance. Inside function types, the connection may also be drawn between an input variable of the function type and an input variable of a contained function instance, or between an output variable of a contained function instance and an output variable of the function type. The connection element may appear zero to many times.

The connection element contains the attributes given in Table E.27.

Table E.27 — Attributes of element connection

Attribute	Data type	Use	Description
source	xsd:string	required	Starting point of connection
destination	xsd:string	required	Endpoint of connection
description	xsd:string	optional	Optional textual description of the connection

EXAMPLE

The values of the source and the destination attributes may be used to encode the starting point and the endpoint of a connection using the syntax <function_instance_name>'.'<variable_name>; example for the value of a source attribute: 'PowerMeasures.Frequency'. Connections to interface variables of a function type use the names of the interface variables only.

E.4.5.6 Element parameterList

E.4.5.6.1 General

If the optional ApplicationProcess element is present in the device profile, it contains a mandatory parameterList element shown in Figure E.7, which represents a sequence of one to many parameter elements.

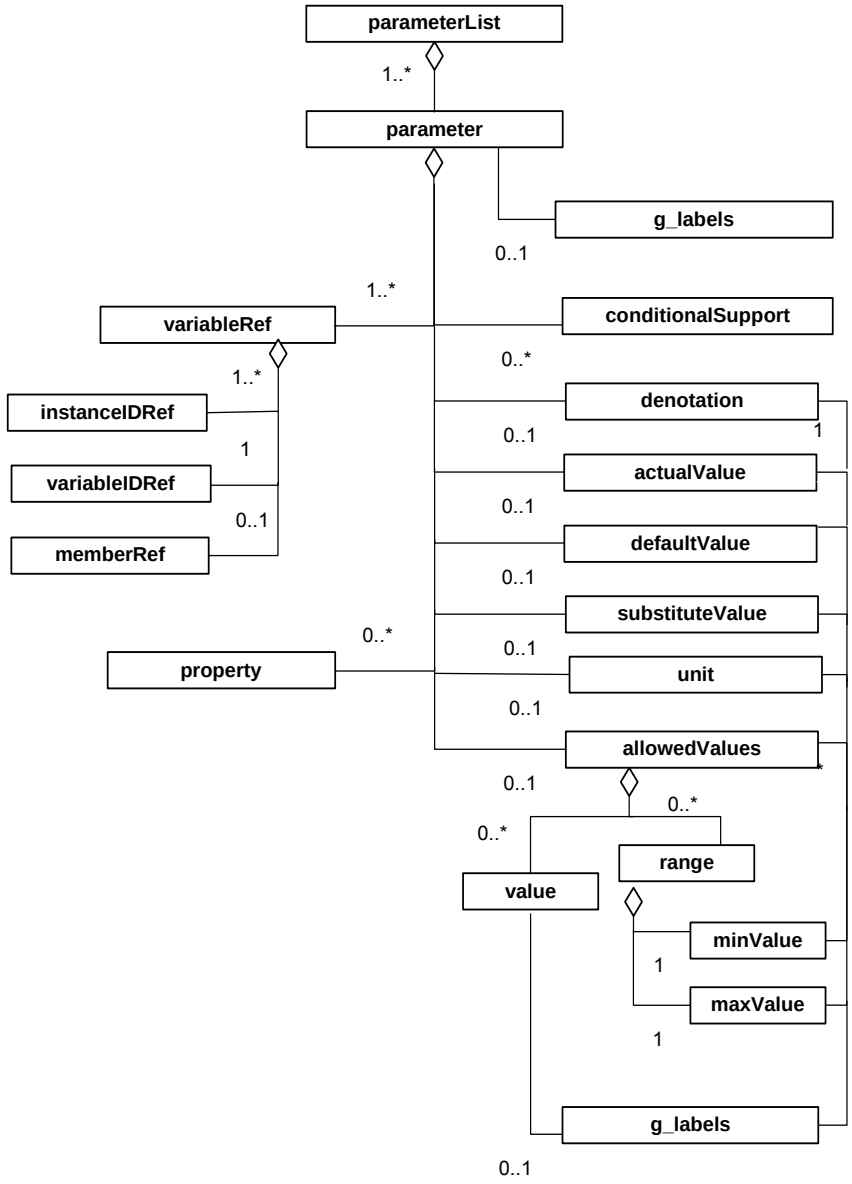


Figure E.7 — parameterList

Each of the parameter elements represents a parameter of the device profile. Multilingual names and/or descriptions for the parameters are provided through the group g_labels. E.2.2.2 describes the group g_labels.

The parameter is described by:

— its attributes,

— a reference to one (or more) interface variable(s) of one (or more) function instance(s) (mandatory element variableRef),

— a possibly empty set of sub-elements (conditionalSupport, denotation, actualValue, defaultValue, substituteValue, allowedValues, unit, property and g_labels).

NOTE References to multiple variables are a special case: specific parameters may reference an output variable of one function instance and an input variable of another function instance at the same time. In this case the data types of

the two variables shall be the same. The XML parser cannot check the equality of data types. This can only be checked by a supporting tool.

E.4.5.6.2 Element parameter

E.4.5.6.2.1 General

The parameter element shall contain the attributes given in Table E.28.

Table E.28 — Attributes of element parameter

Attribute	Data type	Use	Description
uniqueID	xsd:ID	required	Unique ID of the parameter
access	xsd:NMTOKEN	default	Defines which operations are valid for the parameter: — read - read access only (default value) — write - write access only — readWrite - both read and write access — noAccess - access denied
support	xsd:NMTOKEN	optional	Defines whether or not the parameter has to be implemented in the device; valid values: — mandatory - parameter implementation is required — optional - parameter implementation is possible but not required — conditional - parameter implementation is required if one or more other optional parameter(s) is (are) implemented; these parameters are specified using the sub-element conditionalSupport
persistence	xsd:boolean	default	Defines the behaviour after power failure; valid values are false (default) and true
offset	xsd:string	optional	Offset which is added to an actual value to form a scaled value: EngineeringValue = (ParameterValue + offset) * multiplier; if not present, offset = 0 is assumed
multiplier	xsd:string	optional	Scaling factor by which an actual value is multiplied to form a scaled value: EngineeringValue = (ParameterValue + offset) * multiplier; if not present, multiplier = 1 is assumed

E.4.5.6.2.2 Element conditionalSupport

One or more conditionalSupport elements are present only if the value of the support attribute of the parameter element is conditional. Each element refers to a single optional parameter. If at least one of those optional parameters is implemented, the conditional parameter has also to be implemented.

The element conditionalSupport shall contain the single attribute given in Table E.29.

Table E.29 — Attributes of element conditionalSupport

Attribute	Data type	Use	Description
paramIDRef	xsd:IDREF	required	Unique ID of the referenced optional parameter

E.4.5.6.2.3 Element denotation

The denotation element serves to hold application-specific, multilingual names of the parameter. The names are provided through the mandatory g_labels sub-element. It is also possible to add multilingual descriptive information. The element denotation does not have any attributes.

E.4.5.6.2.4 Element actualValue

The actualValue element serves to hold the actual value of the parameter. An optional g_labels sub-element may provide multilingual descriptive information for this value. The value itself is provided in the value attribute of the element actualValue. An offset and multiplier may also be provided.

The attributes of the element actualValue shall be as given in Table E.30.

Table E.30 — Attributes of element actualValue

Attribute	Data type	Use	Description
value	xsd:string	required	Actual value
offset	xsd:string	optional	Offset which is added to an actual value to form a scaled value: EngineeringValue = (value + offset) * multiplier; if not present, the respective value of the parameter element shall be used
multiplier	xsd:string	optional	Scaling factor by which an actual value is multiplied to form a scaled value: EngineeringValue = (value + offset) * multiplier; if not present, the respective value of the parameter element shall be used

E.4.5.6.2.5 Element defaultValue

The defaultValue element serves to hold the default value of the parameter. This value overwrites the initial value of the interface variable of the function type associated with the parameter.

An optional g_labels sub-element may provide multilingual names and/or descriptive information for this value. The value itself is provided by the value attribute of the element defaultValue. An offset and multiplier may also be provided.

The attributes of the element defaultValue shall be as given in Table E.30.

E.4.5.6.2.6 Element substituteValue

The substituteValue element defines a specific value of the parameter that is provided to the device application in certain device operating states (for example on device fault).

An optional g_labels sub-element may provide multilingual names and/or descriptive information for this value. The value itself is provided by the value attribute of the element substituteValue. An offset and multiplier may also be provided.

The attributes of the element substituteValue shall be as given in Table E.30.

E.4.5.6.2.7 Element allowedValues

The allowedValues element defines a list of supported values and/or a single range or several ranges of supported values for the parameter.

The list of supported values is represented by zero to many value sub-elements of the allowedValues element, whereas the ranges are represented by zero to many range sub-elements of the allowedValues element.

The value sub-element holds a single allowed value of the parameter. An optional g_labels sub-element may provide multilingual names and/or descriptive information for this value. The value itself is provided by the value attribute of the element value. An offset and multiplier may also be provided.

The attributes of the element value shall be as given in Table E.30.

The range sub-element contains two required sub-elements, namely the element minValue and the element maxValue, which define the limits of that range of allowed values. The minValue and the maxValue elements have the same structure and attributes as the value sub-element of the allowedValues element. Therefore the description of the value sub-element and Table E.30 are also valid for these sub-elements.

E.4.5.6.2.8 Element unit

The unit element defines the engineering unit of a parameter (for example time, temperature, pressure, flow, acceleration, current, energy), as specified in ISO 1000. An optional g_labels sub-element may provide multilingual names and/or descriptive information for the engineering unit.

The attributes of the element unit shall be as given in Table E.31.

Table E.31 — Attributes of element unit

Attribute	Data type	Use	Description
multiplier	xsd:string	required	Multiplier for engineering units of analog parameters
unitURI	xsd:anyURI	optional	Link to the respective unit definition in a file containing all engineering units (for example time, temperature, pressure, flow, acceleration, current, energy…), as standardized by ISO 1000

E.4.5.6.2.9 Element variableRef

The variableRef element builds a reference to an interface variable of a function instance, or, if the variable is an array or a structure, possibly a reference to a member of the variable (array element or structure component).

In a hierarchically structured ApplicationProcess element, function instances can be located inside function instances of other function types. Therefore a specific instance in the functional tree can only be accessed by stepping through the tree, i.e. the specific instance shall be addressed through a concatenation of instance names. To map this concatenation and allow the referencing of a member, the variableRef element contains:

— a sequence of one to many instanceIDRef elements, followed by

— a single mandatory variableIDRef element, and

— an optional memberRef element.

The variableRef element has the attribute given in Table E.33.

Table E.32 — Attribute of element variableRef

Attribute	Data type	Use	Description
position	xsd:unsignedByte	default	Defines the sequence of multiple mapped data items into a single parameter object; position=1 means starting the mapping at the lowest bit position; the number of bits is defined by the data type of the data item; subsequent data items are packed without gaps; default value: 1 (see Note)
NOTE Attribute can be omitted for a single mapped data item.			

E.4.5.6.2.10 Element instanceIDRef

The instanceIDRef element serves to reference a function instance inside a functionInstanceList element, which may reside either on the level of the ApplicationProcess element or on the level of a functionType element.

The reference of type xsd:IDREF is provided as an attribute of the instanceIDRef element, as given in Table E.33.

Table E.33 — Attributes of element instanceIDRef

Attribute	Data type	Use	Description
uniqueIDRef	xsd:IDREF	required	Unique ID of the referenced function instance

E.4.5.6.2.11 Element variableIDRef

The variableIDRef element serves to reference an interface variable of a function type inside the functionTypeList element.

In a given variableRef element the instance of that function type is defined by the functionInstance element referenced by the instanceIDRef element, which is just preceding the variableIDRef element.

The reference of type xsd:IDREF is provided as an attribute of the variableIDRef element, as given in Table E.34.

Table E.34 — Attributes of element variableIDRef

Attribute	Data type	Use	Description
uniqueIDRef	xsd:IDREF	required	Unique ID of the referenced interface variable of a function type

E.4.5.6.2.12 Element memberRef

The optional memberRef element either references the respective component of an interface variable of structured data type (attribute uniqueIDRef is used), or the respective array element of an interface variable of array data type (attribute index is used). One of these two attributes shall be present if the memberRef element is present.

The memberRef element shall contain the attributes given in Table E.35.

Table E.35 — Attributes of element memberRef

Attribute	Data type	Use	Description
uniqueIDRef	xsd:IDREF	optional	Unique ID of the referenced component of a structured data type
index	xsd:long	optional	Index of the referenced array element

E.4.5.6.3 Element property

The property element is introduced as a generic element to allow the inclusion of values for additional specific properties into the description of a parameter.

The property element shall contain the attributes given in Table E.36.

Table E.36 — Attributes of element property

Attribute	Data type	Use	Description
name	xsd:string	required	Name of the property
value	xsd:string	required	Value of the property

E.4.5.7 Element parameterGroupList

E.4.5.7.1 General

The optional parameterGroupList element, if present, contains a sequence of one to many parameterGroup elements, as shown in Figure E.8. Multilingual names and/or descriptions for the parameter groups are provided through the group g_labels. See E.2.2.2 for the description of the group g_labels.

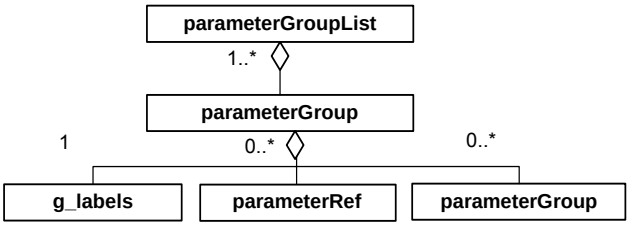


Figure E.8 — parameterGroupList

E.4.5.7.2 Element parameterGroup

Each of the parameterGroup elements combines a set of parameters out of the parameterList element to build a group of parameters which serve a specific purpose, for example to prepare HMI views. This purpose is indicated by the value of the kindOfAccess attribute of the parameterGroup element. It is possible to define a hierarchy of parameter groups.

The respective parameters in the set are referenced through the corresponding number of parameterRef elements.

The parameterGroup element contains the attributes given in Table E.37.

Table E.37 — Attributes of element parameterGroup

Attribute	Data type	Use	Description
uniqueID	xsd:ID	required	Unique ID of the parameter group
kindOfAccess	xsd:string	optional	Classifies the parameters of the parameter group

E.4.5.7.3 Element parameterRef

The parameterRef element serves to reference a parameter element inside the parameterList element of the ApplicationProcess element.

The reference of type xsd:IDREF is provided as an attribute of the parameterRef element, as given in Table E.38.

Table E.38 — Attributes of element parameterRef

Attribute	Data type	Use	Description
uniqueIDRef	xsd:IDREF	required	Unique ID of the referenced parameter

E.4.6 DDXML device profile template schemas

E.4.6.1 XML schema: ISO15745ProfileContainer.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="ISO15745ProfileContainer">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="ISO15745Profile" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="ISO15745Profile">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="ProfileHeader" type="ProfileHeader_DataType"/>
        <xsd:element name="ProfileBody" type="ProfileBody_DataType"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:annotation>
    <xsd:documentation>* HEADER SECTION *</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType name="ProfileHeader_DataType">
    <xsd:sequence>
      <xsd:element name="ProfileIdentification" type="xsd:string"/>
      <xsd:element name="ProfileRevision" type="xsd:string"/>
      <xsd:element name="ProfileName" type="xsd:string"/>
      <xsd:element name="ProfileSource" type="xsd:string"/>
      <xsd:element name="ProfileClassID" type="ProfileClassID_DataType"/>
      <xsd:element name="ProfileDate" type="xsd:date" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="AdditionalInformation" type="xsd:anyURI" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="ISO15745Reference" type="ISO15745Reference_DataType"/>
      <xsd:element name="IASInterfaceType" type="IASInterface_DataType" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:annotation>
    <xsd:documentation>* BODY SECTION *</xsd:documentation>
  </xsd:annotation>
```

```
  <xsd:complexType name="ProfileBody_DataType" abstract="true">
  </xsd:complexType>

  <xsd:annotation>
    <xsd:documentation>* HEADER DATA TYPES *</xsd:documentation>
  </xsd:annotation>
  <xsd:simpleType name="ProfileClassID_DataType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="AIP"/>
      <xsd:enumeration value="Process"/>
      <xsd:enumeration value="InformationExchange"/>
      <xsd:enumeration value="Resource"/>
      <xsd:enumeration value="Device"/>
      <xsd:enumeration value="CommunicationNetwork"/>
      <xsd:enumeration value="Equipment"/>
      <xsd:enumeration value="Human"/>
      <xsd:enumeration value="Material"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:complexType name="ISO15745Reference_DataType">
    <xsd:sequence>
      <xsd:element name="ISO15745Part" type="xsd:positiveInteger"/>
      <xsd:element name="ISO15745Edition" type="xsd:positiveInteger"/>
      <xsd:element name="ProfileTechnology" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:simpleType name="IASInterface_DataType">
    <xsd:union>
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="CSI"/>
          <xsd:enumeration value="HCI"/>
          <xsd:enumeration value="ISI"/>
          <xsd:enumeration value="API"/>
          <xsd:enumeration value="CMI"/>
          <xsd:enumeration value="ESI"/>
          <xsd:enumeration value="FSI"/>
          <xsd:enumeration value="MTI"/>
          <xsd:enumeration value="SEI"/>
          <xsd:enumeration value="USI"/>
        </xsd:restriction>
      </xsd:simpleType>
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:length value="4"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:union>
  </xsd:simpleType>

  <xsd:annotation>
    <xsd:documentation>* ISO 15745 DEFINED DATA TYPES *</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType name="ProfileHandle_DataType">
    <xsd:sequence>
      <xsd:element name="ProfileIdentification" type="xsd:string"/>
      <xsd:element name="ProfileRevision" type="xsd:string"/>
      <xsd:element name="ProfileLocation" type="xsd:anyURI" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>

</xsd:schema>
```

E.4.6.2 XML schema: CommonElements.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <!--################ common attribute group-->
  <xsd:attributeGroup name="ag_formatAndFile">
    <xsd:attribute name="formatName" type="xsd:string" fixed="DDXML" form="unqualified"/>
    <xsd:attribute name="formatVersion" type="xsd:string" fixed="2.0" form="unqualified"/>
    <xsd:attribute name="fileName" type="xsd:string" use="required" form="unqualified"/>
    <xsd:attribute name="fileCreator" type="xsd:string" use="required" form="unqualified"/>
    <xsd:attribute name="fileCreationDate" type="xsd:date" use="required" form="unqualified"/>
```

```
  <xsd:attribute name="fileCreationTime" type="xsd:time" use="optional"/>
  <xsd:attribute name="fileModificationDate" type="xsd:date" use="optional" form="unqualified"/>
  <xsd:attribute name="fileModificationTime" type="xsd:time" use="optional"/>
  <xsd:attribute name="fileModifiedBy" type="xsd:string" use="optional"/>
  <xsd:attribute name="fileVersion" type="xsd:string" use="required" form="unqualified"/>
</xsd:attributeGroup>
<!--################ common groups-->
<xsd:group name="g_labels">
  <xsd:sequence>
    <xsd:choice maxOccurs="unbounded">
      <xsd:element name="label">
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base="xsd:string">
              <xsd:attribute name="lang" type="xsd:language" use="required"/>
              <xsd:attribute name="URI" type="xsd:anyURI" use="optional"/>
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="description">
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base="xsd:string">
              <xsd:attribute name="lang" type="xsd:language" use="required"/>
              <xsd:attribute name="URI" type="xsd:anyURI" use="optional"/>
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="labelRef">
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base="xsd:anyURI">
              <xsd:attribute name="dictID" type="xsd:IDREF" use="required"/>
              <xsd:attribute name="textID" type="xsd:string" use="optional"/>
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
      </xsd:element>
    </xsd:choice>
  </xsd:sequence>
</xsd:group>
<xsd:group name="g_simple">
  <xsd:choice>
    <xsd:element name="BOOL"/>
    <xsd:element name="BITSTRING"/>
    <xsd:element name="BYTE"/>
    <xsd:element name="CHAR"/>
    <xsd:element name="WORD"/>
    <xsd:element name="DWORD"/>
    <xsd:element name="LWORD"/>
    <xsd:element name="SINT"/>
    <xsd:element name="INT"/>
    <xsd:element name="DINT"/>
    <xsd:element name="LINT"/>
    <xsd:element name="USINT"/>
    <xsd:element name="UINT"/>
    <xsd:element name="UDINT"/>
    <xsd:element name="ULINT"/>
    <xsd:element name="REAL"/>
    <xsd:element name="LREAL"/>
    <xsd:element name="TIME"/>
    <xsd:element name="DATE"/>
    <xsd:element name="DT"/>
    <xsd:element name="TOD"/>
    <xsd:element name="STRING"/>
    <xsd:element name="WSTRING"/>
  </xsd:choice>
</xsd:group>
<!--################ common elements-->
<xsd:element name="vendorID">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
```

```
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
<xsd:element name="deviceFamily">
  <xsd:complexType>
    <xsd:group ref="g_labels"/>
    <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="productID">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
<xsd:element name="version">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="versionType" use="required">
          <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
              <xsd:enumeration value="SW"/>
              <xsd:enumeration value="FW"/>
              <xsd:enumeration value="HW"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
<xsd:element name="buildDate" type="xsd:date"/>
<xsd:element name="specificationRevision">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
</xsd:schema>
```

E.4.6.3 XML schema: ProfileBody_Device_ModbusTCP.xsd

The XML schema ProfileBody_Device_ModbusTCP.xsd includes the schema ISO15745ProfileContainer.xsd in E.4.6.1 and the schema CommonElements.xsd in E.4.6.2.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="ISO15745ProfileContainer.xsd"/>
  <xsd:include schemaLocation="CommonElements.xsd"/>

  <!--################ profile body device -->
  <xsd:complexType name="ProfileBody_Device_ModbusTCP">
    <xsd:complexContent>
      <xsd:extension base="ProfileBody_DataType">
        <xsd:sequence>
          <xsd:element ref="DeviceIdentity" minOccurs="0"/>
          <xsd:element ref="DeviceManager" minOccurs="0"/>
          <xsd:element ref="DeviceFunction" maxOccurs="unbounded"/>
          <xsd:element ref="ApplicationProcess" minOccurs="0" maxOccurs="unbounded"/>
          <xsd:element name="ExternalProfileHandle" type="ProfileHandle_DataType" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="ag_formatAndFile"/>
```

```
      <xsd:attribute name="supportedLanguages" type="xsd:NMTOKENS" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<!--################ device identity elements -->
<xsd:element name="DeviceIdentity">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="vendorName"/>
      <xsd:element ref="vendorID" minOccurs="0"/>
      <xsd:element ref="vendorText" minOccurs="0"/>
      <xsd:element ref="deviceFamily" minOccurs="0"/>
      <xsd:element ref="productFamily" minOccurs="0"/>
      <xsd:element ref="productName"/>
      <xsd:element ref="productID" minOccurs="0"/>
      <xsd:element ref="productText" minOccurs="0"/>
      <xsd:element ref="orderNumber" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="version" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="buildDate" minOccurs="0"/>
      <xsd:element ref="specificationRevision" minOccurs="0"/>
      <xsd:element ref="instanceName" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="productFamily">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
<xsd:element name="instanceName">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="readOnly" type="xsd:boolean" default="false"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
<xsd:element name="orderNumber">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
<xsd:element name="productName">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
<xsd:element name="productText">
  <xsd:complexType>
    <xsd:group ref="g_labels"/>
    <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="vendorName">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
<xsd:element name="vendorText">
```

```
  <xsd:complexType>
    <xsd:group ref="g_labels"/>
    <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
  </xsd:complexType>
</xsd:element>
<!--################ device manager elements -->
<xsd:element name="DeviceManager">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="indicatorList" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="indicatorList">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="LEDList" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="LEDList">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="LED" maxOccurs="unbounded"/>
      <xsd:element ref="combinedState" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="LED">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:group ref="g_labels"/>
      <xsd:element ref="LEDstate" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="LEDcolors" use="required">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="monocolor"/>
          <xsd:enumeration value="bicolor"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name="LEDtype" use="optional">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="IO"/>
          <xsd:enumeration value="device"/>
          <xsd:enumeration value="communication"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
<xsd:element name="LEDstate">
  <xsd:complexType>
    <xsd:group ref="g_labels"/>
    <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
    <xsd:attribute name="state" use="required">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="on"/>
          <xsd:enumeration value="off"/>
          <xsd:enumeration value="flashing"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name="LEDcolor" use="required">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="green"/>
          <xsd:enumeration value="amber"/>
          <xsd:enumeration value="red"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name="flashingPeriod" type="xsd:unsignedInt" use="optional"/>
```

```
    <xsd:attribute name="impulsWidth" type="xsd:unsignedByte" default="50"/>
    <xsd:attribute name="numberOfImpulses" type="xsd:unsignedByte" default="1"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="combinedState">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:group ref="g_labels"/>
      <xsd:element name="LEDstateRef" minOccurs="2" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:attribute name="stateIDRef" type="xsd:IDREF" use="required"/>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--################ device function elements -->
<xsd:element name="DeviceFunction">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="capabilities"/>
      <xsd:element ref="picturesList" minOccurs="0"/>
      <xsd:element ref="dictionaryList" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="capabilities">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="characteristicsList" maxOccurs="unbounded"/>
      <xsd:element ref="standardComplianceList" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="characteristicsList">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="category" minOccurs="0">
        <xsd:complexType>
          <xsd:group ref="g_labels"/>
        </xsd:complexType>
      </xsd:element>
      <xsd:element ref="characteristic" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="characteristic">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="characteristicName"/>
      <xsd:element ref="characteristicContent" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="characteristicContent">
  <xsd:complexType>
    <xsd:group ref="g_labels"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="characteristicName">
  <xsd:complexType>
    <xsd:group ref="g_labels"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="standardComplianceList">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="compliantWith" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="compliantWith">
  <xsd:complexType>
    <xsd:group ref="g_labels"/>
    <xsd:attribute name="name" type="xsd:string" use="required"/>
    <xsd:attribute name="range" default="international">
```

```
      <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
          <xsd:enumeration value="international"/>
          <xsd:enumeration value="internal"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
<xsd:element name="picturesList">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="picture" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="picture">
  <xsd:complexType>
    <xsd:group ref="g_labels"/>
    <xsd:attribute name="URI" type="xsd:anyURI" use="required"/>
    <xsd:attribute name="number" type="xsd:unsignedInt" use="optional"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="dictionaryList">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="dictionary" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="dictionary">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="file" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="lang" type="xsd:language" use="required"/>
    <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="file">
  <xsd:complexType>
    <xsd:attribute name="URI" type="xsd:anyURI" use="required"/>
  </xsd:complexType>
</xsd:element>
<!--################ application process elements -->
<xsd:element name="ApplicationProcess">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="dataTypeList" minOccurs="0"/>
      <xsd:element ref="functionTypeList"/>
      <xsd:element ref="functionInstanceList"/>
      <xsd:element ref="parameterList"/>
      <xsd:element ref="parameterGroupList" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="dataTypeList">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:group ref="g_complex" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="functionTypeList">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="functionType" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="functionType">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="versionInfo" maxOccurs="unbounded"/>
      <xsd:element ref="interfaceList"/>
      <xsd:element ref="functionInstanceList" minOccurs="0"/>
```

```
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" use="required"/>
    <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
    <xsd:attribute name="description" type="xsd:string" use="optional"/>
    <xsd:attribute name="package" type="xsd:string" use="optional"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="versionInfo">
  <xsd:complexType>
    <xsd:attribute name="organization" type="xsd:string" use="required"/>
    <xsd:attribute name="version" type="xsd:string" use="required"/>
    <xsd:attribute name="author" type="xsd:string" use="required"/>
    <xsd:attribute name="date" type="xsd:date" use="required"/>
    <xsd:attribute name="remarks" type="xsd:string" use="optional"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="interfaceList">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="inputVars" minOccurs="0"/>
      <xsd:element ref="outputVars" minOccurs="0"/>
      <xsd:element ref="configVars" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="inputVars">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="varDeclaration" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="outputVars">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="varDeclaration" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="configVars">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="varDeclaration" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="varDeclaration">
  <xsd:complexType>
    <xsd:choice>
      <xsd:group ref="g_simple"/>
      <xsd:element ref="dataTypeIDRef"/>
    </xsd:choice>
    <xsd:attribute name="name" type="xsd:string" use="required"/>
    <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
    <xsd:attribute name="size" type="xsd:string" use="optional"/>
    <xsd:attribute name="initialValue" type="xsd:string" use="optional"/>
    <xsd:attribute name="description" type="xsd:string" use="optional"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="dataTypeIDRef">
  <xsd:complexType>
    <xsd:attribute name="uniqueIDRef" type="xsd:IDREF" use="required"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="functionInstanceList">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="functionInstance" maxOccurs="unbounded"/>
      <xsd:element ref="connection" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="functionInstance">
  <xsd:complexType>
    <xsd:attribute name="name" type="xsd:string" use="required"/>
    <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
```

```
    <xsd:attribute name="typeIDRef" type="xsd:IDREF" use="required"/>
    <xsd:attribute name="description" type="xsd:string" use="optional"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="connection">
  <xsd:complexType>
    <xsd:attribute name="source" type="xsd:string" use="required"/>
    <xsd:attribute name="destination" type="xsd:string" use="required"/>
    <xsd:attribute name="description" type="xsd:string" use="optional"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="parameterList">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="parameter" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="parameter">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:group ref="g_labels"/>
      <xsd:element ref="variableRef" maxOccurs="unbounded"/>
      <xsd:element ref="conditionalSupport" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="denotation" minOccurs="0"/>
      <xsd:element ref="actualValue" minOccurs="0"/>
      <xsd:element ref="defaultValue" minOccurs="0"/>
      <xsd:element ref="substituteValue" minOccurs="0"/>
      <xsd:element ref="allowedValues" minOccurs="0"/>
      <xsd:element ref="unit" minOccurs="0"/>
      <xsd:element ref="property" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="ag_parameter"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="variableRef">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="instanceIDRef" maxOccurs="unbounded"/>
      <xsd:element ref="variableIDRef"/>
      <xsd:element ref="memberRef" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="position" type="xsd:unsignedByte" default="1"/>
 </xsd:complexType>
</xsd:element>
<xsd:element name="instanceIDRef">
  <xsd:complexType>
    <xsd:attribute name="uniqueIDRef" type="xsd:IDREF"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="variableIDRef">
  <xsd:complexType>
    <xsd:attribute name="uniqueIDRef" type="xsd:IDREF"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="memberRef">
  <xsd:complexType>
    <xsd:attribute name="uniqueIDRef" type="xsd:IDREF" use="optional"/>
    <xsd:attribute name="index" type="xsd:long" use="optional"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="actualValue">
  <xsd:complexType>
    <xsd:group ref="g_labels" minOccurs="0"/>
    <xsd:attributeGroup ref="ag_value"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="conditionalSupport">
  <xsd:complexType>
    <xsd:attribute name="paramIDRef" type="xsd:IDREF" use="required"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="denotation">
  <xsd:complexType>
    <xsd:group ref="g_labels"/>
  </xsd:complexType>
```

```
</xsd:element>
<xsd:element name="defaultValue">
  <xsd:complexType>
    <xsd:group ref="g_labels" minOccurs="0"/>
    <xsd:attributeGroup ref="ag_value"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="substituteValue">
  <xsd:complexType>
    <xsd:group ref="g_labels" minOccurs="0"/>
    <xsd:attributeGroup ref="ag_value"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="allowedValues">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="value" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="range" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="value">
  <xsd:complexType>
    <xsd:group ref="g_labels" minOccurs="0"/>
    <xsd:attributeGroup ref="ag_value"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="range">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="minValue">
        <xsd:complexType>
          <xsd:group ref="g_labels" minOccurs="0"/>
          <xsd:attributeGroup ref="ag_value"/>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="maxValue">
        <xsd:complexType>
          <xsd:group ref="g_labels" minOccurs="0"/>
          <xsd:attributeGroup ref="ag_value"/>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="unit">
  <xsd:complexType>
    <xsd:group ref="g_labels"/>
    <xsd:attribute name="multiplier" type="xsd:string" use="required"/>
    <xsd:attribute name="unitURI" type="xsd:anyURI" use="optional"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="property">
  <xsd:complexType>
    <xsd:attribute name="name" type="xsd:string" use="required"/>
    <xsd:attribute name="value" type="xsd:string" use="required"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="parameterGroupList">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="parameterGroup" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="parameterGroup">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:group ref="g_labels"/>
      <xsd:element ref="parameterGroup" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="parameterRef" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
    <xsd:attribute name="kindOfAccess" type="xsd:string" use="optional"/>
  </xsd:complexType>
</xsd:element>
```

```
<xsd:element name="parameterRef">
  <xsd:complexType>
    <xsd:attribute name="uniqueIDRef" type="xsd:IDREF" use="required"/>
  </xsd:complexType>
</xsd:element>
<!--################ complex types -->
<xsd:element name="array">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="subrange" maxOccurs="unbounded"/>
      <xsd:choice>
        <xsd:group ref="g_simple"/>
        <xsd:element ref="dataTypeIDRef"/>
      </xsd:choice>
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" use="required"/>
    <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
    <xsd:attribute name="description" type="xsd:string" use="optional"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="subrange">
  <xsd:complexType>
    <xsd:attribute name="lowerLimit" type="xsd:long" use="required"/>
    <xsd:attribute name="upperLimit" type="xsd:long" use="required"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="struct">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="count" minOccurs="0"/>
      <xsd:element ref="varDeclaration" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" use="required"/>
    <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
    <xsd:attribute name="description" type="xsd:string" use="optional"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="enum">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="enumValue" maxOccurs="unbounded"/>
      <xsd:group ref="g_simple" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" use="required"/>
    <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
    <xsd:attribute name="size" type="xsd:string" use="optional"/>
    <xsd:attribute name="description" type="xsd:string" use="optional"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="enumValue">
  <xsd:complexType>
    <xsd:group ref="g_labels"/>
    <xsd:attribute name="value" type="xsd:string" use="optional"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="derived">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="count" minOccurs="0"/>
      <xsd:choice>
        <xsd:group ref="g_simple"/>
        <xsd:element ref="dataTypeIDRef"/>
      </xsd:choice>
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" use="required"/>
    <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
    <xsd:attribute name="description" type="xsd:string" use="optional"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="count">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:group ref="g_labels" minOccurs="0"/>
      <xsd:element ref="defaultValue"/>
      <xsd:element ref="allowedValues" minOccurs="0"/>
    </xsd:sequence>
```

```
      <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
      <xsd:attribute name="access" default="read">
        <xsd:simpleType>
          <xsd:restriction base="xsd:NMTOKEN">
            <xsd:enumeration value="read"/>
            <xsd:enumeration value="write"/>
            <xsd:enumeration value="readWrite"/>
            <xsd:enumeration value="noAccess"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
    </xsd:complexType>
  </xsd:element>
  <!--################ group-->
  <xsd:group name="g_complex">
    <xsd:choice>
      <xsd:element ref="array"/>
      <xsd:element ref="struct"/>
      <xsd:element ref="enum"/>
      <xsd:element ref="derived"/>
    </xsd:choice>
  </xsd:group>
  <!--################ attribute groups-->
  <xsd:attributeGroup name="ag_parameter">
    <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
    <xsd:attribute name="access" default="read">
      <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
          <xsd:enumeration value="read"/>
          <xsd:enumeration value="write"/>
          <xsd:enumeration value="readWrite"/>
          <xsd:enumeration value="noAccess"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name="support" use="optional">
      <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
          <xsd:enumeration value="mandatory"/>
          <xsd:enumeration value="optional"/>
          <xsd:enumeration value="conditional"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name="persistence" type="xsd:boolean" default="false"/>
    <xsd:attribute name="offset" type="xsd:string" use="optional"/>
    <xsd:attribute name="multiplier" type="xsd:string" use="optional"/>
  </xsd:attributeGroup>
  <xsd:attributeGroup name="ag_value">
    <xsd:attribute name="value" type="xsd:string" use="required"/>
    <xsd:attribute name="offset" type="xsd:string" use="optional"/>
    <xsd:attribute name="multiplier" type="xsd:string" use="optional"/>
  </xsd:attributeGroup>
</xsd:schema>
```

E.5 Communication network profile template description

E.5.1 ProfileBody_CommunicationNetwork_ModbusTCP

This part defines the Modbus TCP communication network profile.

The ProfileBody_CommunicationNetwork_ModbusTCP contains the ApplicationLayers, the TransportLayers and the NetworkManagement elements shown in Figure 25.

E.5.2 ApplicationLayers

E.5.2.1 General

Figure E.9 shows the structure of the Modbus TCP ApplicationLayers class.

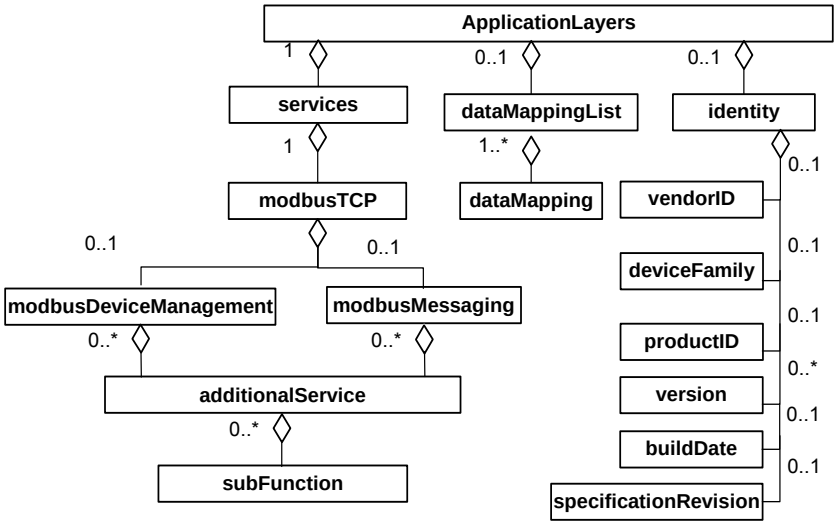


Figure E.9 — Modbus TCP ApplicationLayers class

The ApplicationLayers element has the attributes given in Table E.39.

Table E.39 — Attributes of element ApplicationLayers

Attribute	Data type	Use	Description
conformanceClass	xsd:string	required	Conformance class of the device type (see NOTE)
communicationEntityType	xsd:NMTOKENS	default	Type of the communication entity - if several types are supported, their names shall be separated by a blank character, for example "master slave"; the type names shall be chosen out of the following enumeration of names: — slave — master — server — client — interconnection (for example interconnection is gateway functionality) — peer- communication entity which acts as a client and a server
NOTE Using that attribute it is possible to classify the device according to the supported services of the communication protocol.			

E.5.2.2 Element identity

Since different communication profiles may require different identity informations, an optional local identity sub-element may be used within an ApplicationLayers element. This identity element may contain a subset of the sub–elements of the DeviceIdentity element described in 6.5.1.2. All sub–element descriptions given there also apply for the sub-elements of this identity element.

E.5.2.3 Elements services / ModbusTCP

E.5.2.3.1 General

A single services element describes the communication profile specific features of a device type. This annex only specifies the modbusTCP element corresponding to Modbus TCP service.

The modbusTCP element describes the additional Modbus services supported by the device, which are not defined through the conformanceClass attribute of the ApplicationLayers element. It has the optional sub-elements modbusMessaging and modbusDeviceManagement, as shown in Figure E.9.

The modbusTCP element contains the attributes given in Table E.40.

Table E.40 — Attributes of element modbusTCP

Attribute	Data type	Use	Description
type	xsd:NMTOKENS	required	Type of the messaging endpoint - if combinations of types are supported, their names shall be separated by a blank character, for example "server gateway"; valid type names: — client — server — gateway
accessControl	xsd:boolean	default	Specifies whether access control for remote IP addresses is supported by the device type: false (default), true
maxNbOfTotalConnection	xsd:unsignedInt	optional	Maximum number of concurrent Modbus TCP connections supported by the device type; requirement: TotalConn = ServerConn + ClientConn
maxNbOfServerConnection	xsd:unsignedInt	optional	Maximum number of concurrent Modbus TCP server connections supported by the device type
maxNbOfClientConnection	xsd:unsignedInt	optional	Maximum number of concurrent Modbus TCP client connections supported by the device type
maxSizeOfGatewayRoutingTable	xsd:unsignedInt	optional	If applicable: maximum size of the gateway routing table supported by the device type

E.5.2.3.2 Element modbusMessaging

If present, the optional modbusMessaging element contains a sequence of one to many additionalService sub-elements, which enumerate the additional Modbus services provided by the device. The modbusMessaging element has no attributes.

E.5.2.3.3 Element modbusDeviceManagement

If present, the optional modbusDeviceManagement element contains a sequence of one to many additionalService sub-elements, which enumerate the additional Modbus device management services provided by the device. The modbusDeviceManagement element has no attributes.

E.5.2.3.4 Element additionalService

The additionalService sub-element of the modbusMessaging element shown in Figure E.9 allows to specify further Modbus services supported by the device type. It contains the attributes given in Table E.41.

Table E.41 — Attributes of element additionalService

Attribute	Data type	Use	Description
name	xsd:string	required	Name of the additionally supported Modbus service
functionCode	xsd:unsignedByte	required	Function code of the additionally supported Modbus service
subFunctionCode	xsd:unsignedByte	optional	Sub-function code of the additionally supported Modbus service, if any (see NOTE)
description	xsd:string	optional	Optional textual description of the additionally supported Modbus service
NOTE This attribute is omited if no sub-function code is supported, or if more than one sub-function code is supported by the service. If more than one sub-function code is supported then the sub-element subFunction is used instead.			

E.5.2.3.5 Element subFunction

This subFunction element is used in case multiple sub-function codes are associated with the Modbus communication service.

The subFunction element contains the attributes given in Table E.42.

Table E.42 — Attributes of element subFunction

Attribute	Data type	Use	Description
code	xsd:unsignedByte	required	Sub-function code of the supported Modbus service
description	xsd:string	optional	Textual description of the sub-function code

E.5.2.4 Element dataMappingList

If present, the dataMappingList element shown in Figure E.9 contains a sequence of one to many dataMapping sub-elements, which define the associations between the protocol-specific addresses of communicated data and the respective parameter elements within the ApplicationProcess element.

The dataMappingList element has the attribute given in Table E.43.

Table E.43 — Attribute of element dataMappingList

Attribute	Data type	Use	Description
description	xsd:string	optional	Optional textual description of the data mapping list

The dataMapping element has an empty content part. It contains the three attributes given in Table E.44.

Table E.44 — Attributes of element dataMapping

Attribute	Data type	Use	Description
parameterIDRef	xsd:IDREF	required	Unique ID of the referenced parameter element
accessPath	xsd:string	required	Protocol-specific address value
services	xsd:NMTOKENS	optional	Names of the services allowed to this parameter

E.5.3 TransportLayers

E.5.3.1 General

The TransportLayers element contains the PhysicalLayer and the MacLinkLayer elements shown in Figure E.10 .

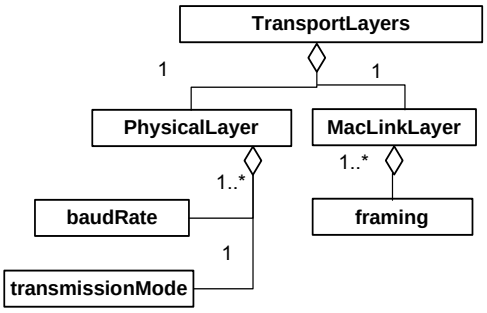


Figure E.10 — Modbus TCP TransportLayers class

E.5.3.2 PhysicalLayer

E.5.3.2.1 General

The PhysicalLayer element shown in Figure E.10 contains a sequence of mandatory sub-elements defining the properties of the physical layers of the communication entity. The sub-element baudRate shall appear one to many times, whereas there is a single mandatory transmissionMode element. All sub-elements define valid values for their different attributes.

E.5.3.2.2 Element baudRate

The baudRate element provides a single baud rate value supported by the Ethernet connection. If several baud rates are supported, the element appears several times. The baudRate element has an empty content and the single attribute given in Table E.45.

Table E.45 — Attribute of element baudRate

Attribute	Data type	Use	Description
value	xsd:string	required	Supported baud rate value; valid values are 10 Mbps and 100 Mbps

E.5.3.2.3 Element transmissionMode

The transmissionMode element provides the transmission mode supported by the Ethernet connection. The transmissionMode element has an empty content and the single attribute given in Table E.46.

Table E.46 — Attribute of element transmissionMode

Attribute	Data type	Use	Description
mode	xsd:string	required	Supported transmission mode; valid values are half-duplex, full-duplex and auto-negotiation

E.5.3.3 MacLinkLayer / framing

The MacLinkLayer element contains only the framing element shown in Figure E.10.

The framing element provides a single framing type supported by the Ethernet connection. If several framing types are supported, the element appears several times. The framing element has an empty content and the single attribute given in Table E.47.

Table E.47 — Attribute of element framing

Attribute	Data type	Use	Description
type	xsd:string	required	Supported framing type; valid values are Ethernet II, IEEE 802.3 sender and IEEE 802.3 receiver

E.5.4 NetworkManagement

E.5.4.1 General

The NetworkManagement element contains the SNMP sub-element shown in Figure E.11.

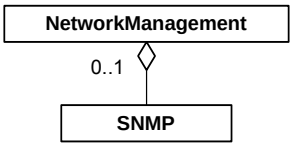


Figure E.11 — Modbus TCP NetworkManagement class

E.5.4.2 SNMP

The SNMP element in Figure E.11, if present, indicates that the service "Simple Network Management Protocol" (see RFC 1157) is supported by the device type.

The SNMP element contains the attribute given in Table E.48.

Table E.48 — Attributes of element SNMP

Attribute	Data type	Use	Description
version	xsd:unsignedByte	default	Version of the supported network management protocol; valid values are snmpV1, snmpV2 and snmpV3

E.5.5 DDXML communication network profile template schemas

The XML schema ProfileBody_CommunicationNetwork_ModbusTCP.xsd includes the schema ISO15745ProfileContainer.xsd in E.4.6.1 and the schema CommonElements.xsd in E.4.6.2.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="ISO15745ProfileContainer.xsd"/>
  <xsd:include schemaLocation="CommonElements.xsd"/>
```

```
  <!--################ profile body communication network -->
  <xsd:complexType name="ProfileBody_CommunicationNetwork_ModbusTCP">
    <xsd:complexContent>
      <xsd:extension base="ProfileBody_DataType">
        <xsd:choice>
          <xsd:sequence>
            <!--################ application layers elements -->
            <xsd:element name="ApplicationLayers">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="identity" minOccurs="0">
                    <xsd:complexType>
                      <xsd:sequence>
                        <xsd:element ref="vendorID" minOccurs="0"/>
                        <xsd:element ref="deviceFamily" minOccurs="0"/>
                        <xsd:element ref="productID" minOccurs="0"/>
                        <xsd:element ref="version" minOccurs="0" maxOccurs="unbounded"/>
                        <xsd:element ref="buildDate" minOccurs="0"/>
                        <xsd:element ref="specificationRevision" minOccurs="0"/>
                      </xsd:sequence>
                    </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="dataMappingList" minOccurs="0">
                    <xsd:complexType>
                      <xsd:sequence>
                        <xsd:element name="dataMapping" maxOccurs="unbounded">
                          <xsd:complexType>
                            <xsd:attribute name="parameterIDRef" type="xsd:IDREF" use="required"/>
                            <xsd:attribute name="accessPath" type="xsd:string" use="required"/>
                            <xsd:attribute name="services" type="xsd:NMTOKENS" use="optional"/>
                          </xsd:complexType>
                        </xsd:element>
                      </xsd:sequence>
                      <xsd:attribute name="description" type="xsd:string" use="optional"/>
                    </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="services">
                    <xsd:annotation>
                      <xsd:documentation>Ethernet TCP/IP services</xsd:documentation>
                    </xsd:annotation>
                    <xsd:complexType>
                      <xsd:sequence>
                        <xsd:element name="modbusTCP">
                          <xsd:complexType>
                            <xsd:sequence>
                              <xsd:element name="modbusMessaging" minOccurs="0">
                                <xsd:complexType>
                                  <xsd:sequence>
                                    <xsd:element name="additionalService" maxOccurs="unbounded">
                                      <xsd:complexType>
                                        <xsd:sequence minOccurs="0">
                                          <xsd:element name="subFunction" maxOccurs="unbounded">
                                            <xsd:complexType>
                                              <xsd:attribute name="code" type="xsd:unsignedByte"
use="required"/>
                                              <xsd:attribute name="description" type="xsd:string"
use="optional"/>
                                            </xsd:complexType>
                                          </xsd:element>
                                        </xsd:sequence>
                                        <xsd:attribute name="name" type="xsd:string" use="required"/>
                                        <xsd:attribute name="functionCode" type="xsd:unsignedByte"
use="required"/>
                                        <xsd:attribute name="subFunctionCode" type="xsd:unsignedByte"
use="optional"/>
                                        <xsd:attribute name="description" type="xsd:string"
use="optional"/>
                                      </xsd:complexType>
                                    </xsd:element>
                                  </xsd:sequence>
                                </xsd:complexType>
                              </xsd:element>
                              <xsd:element name="modbusDeviceManagement" minOccurs="0">
                                <xsd:complexType>
                                  <xsd:sequence>
                                    <xsd:element name="additionalService" maxOccurs="unbounded">
```

```
                                        <xsd:complexType>
                                          <xsd:sequence minOccurs="0">
                                            <xsd:element name="subFunction" maxOccurs="unbounded">
                                              <xsd:complexType>
                                                <xsd:attribute name="code" type="xsd:unsignedByte"
use="required"/>
                                                <xsd:attribute name="description" type="xsd:string"
use="optional"/>
                                              </xsd:complexType>
                                            </xsd:element>
                                          </xsd:sequence>
                                          <xsd:attribute name="name" type="xsd:string" use="required"/>
                                          <xsd:attribute name="functionCode" type="xsd:unsignedByte"
use="required"/>
                                          <xsd:attribute name="subFunctionCode" type="xsd:unsignedByte"
use="optional"/>
                                          <xsd:attribute name="description" type="xsd:string"
use="optional"/>
                                        </xsd:complexType>
                                      </xsd:element>
                                    </xsd:sequence>
                                  </xsd:complexType>
                                </xsd:element>
                              </xsd:sequence>
                              <xsd:attribute name="type" type="xsd:NMTOKENS" use="required"/>
                              <xsd:attribute name="accessControl" type="xsd:boolean" default="false"/>
                              <xsd:attribute name="maxNbOfTotalConnection" type="xsd:unsignedInt"
use="optional"/>
                              <xsd:attribute name="maxNbOfServerConnection" type="xsd:unsignedInt"
use="optional"/>
                              <xsd:attribute name="maxNbOfClientConnection" type="xsd:unsignedInt"
use="optional"/>
                              <xsd:attribute name="maxSizeOfGatewayRoutingTable" type="xsd:unsignedInt"
use="optional"/>
                            </xsd:complexType>
                          </xsd:element>
                        </xsd:sequence>
                      </xsd:complexType>
                    </xsd:element>
                  </xsd:sequence>
                  <xsd:attribute name="conformanceClass" type="xsd:string" use="required"/>
                  <xsd:attribute name="communicationEntityType" type="xsd:NMTOKENS" default="slave">
                    <xsd:annotation>
                      <xsd:documentation>Defines the entity type: slave (default), master, client,
server, interconnection (example: gateway), peer (acts as client and server)</xsd:documentation>
                    </xsd:annotation>
                  </xsd:attribute>
                </xsd:complexType>
              </xsd:element>
              <!--################ transport layers elements -->
              <xsd:element name="TransportLayers">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="MacLinkLayer">
                      <xsd:complexType>
                        <xsd:sequence>
                          <xsd:element name="framing" maxOccurs="unbounded">
                            <xsd:complexType>
                              <xsd:attribute name="type" use="required">
                                <xsd:simpleType>
                                  <xsd:restriction base="xsd:string">
                                    <xsd:enumeration value="Ethernet II"/>
                                    <xsd:enumeration value="IEEE 802.3 sender"/>
                                    <xsd:enumeration value="IEEE 802.3 receiver"/>
                                  </xsd:restriction>
                                </xsd:simpleType>
                              </xsd:attribute>
                            </xsd:complexType>
                          </xsd:element>
                        </xsd:sequence>
                      </xsd:complexType>
                    </xsd:element>
                    <xsd:element name="PhysicalLayer">
                      <xsd:complexType>
                        <xsd:sequence>
                          <xsd:element name="baudRate" maxOccurs="unbounded">
```

```
                        <xsd:complexType>
                          <xsd:attribute name="value" use="required">
                            <xsd:simpleType>
                              <xsd:restriction base="xsd:string">
                                <xsd:enumeration value="10 Mbps"/>
                                <xsd:enumeration value="100 Mbps"/>
                              </xsd:restriction>
                            </xsd:simpleType>
                          </xsd:attribute>
                        </xsd:complexType>
                      </xsd:element>
                      <xsd:element name="transmissionMode">
                        <xsd:complexType>
                          <xsd:attribute name="mode" use="required">
                            <xsd:simpleType>
                              <xsd:restriction base="xsd:string">
                                <xsd:enumeration value="half-duplex"/>
                                <xsd:enumeration value="full-duplex"/>
                                <xsd:enumeration value="auto-negotiation"/>
                              </xsd:restriction>
                            </xsd:simpleType>
                          </xsd:attribute>
                        </xsd:complexType>
                      </xsd:element>
                    </xsd:sequence>
                  </xsd:complexType>
                </xsd:element>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
          <!--################ network management elements -->
          <xsd:element name="NetworkManagement" minOccurs="0">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="SNMP" minOccurs="0">
                  <xsd:complexType>
                    <xsd:attribute name="version" default="snmpV1">
                      <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                          <xsd:enumeration value="snmpV1"/>
                          <xsd:enumeration value="snmpV2"/>
                          <xsd:enumeration value="snmpV3"/>
                        </xsd:restriction>
                      </xsd:simpleType>
                    </xsd:attribute>
                  </xsd:complexType>
                </xsd:element>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
        <xsd:element name="ExternalProfileHandle" type="ProfileHandle_DataType"/>
      </xsd:choice>
      <xsd:attributeGroup ref="ag_formatAndFile"/>
      <xsd:attribute name="supportedLanguages" use="optional">
        <xsd:simpleType>
          <xsd:restriction base="xsd:NMTOKENS"/>
        </xsd:simpleType>
      </xsd:attribute>
    </xsd:extension>
  </xsd:complexContent>
 </xsd:complexType>
</xsd:schema>
```

Annex F
(normative)

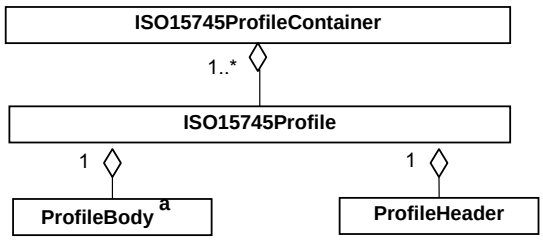
EtherCAT profile templates

F.1 Overview

EtherCAT is an Ethernet based communication system specified in IEC/PAS 62407.

EtherCAT uses the concept of the multi-profile container specified in Amendment 1 to ISO 15745-1:2003 for XML profile files. Therefore, EtherCAT profile templates are based on the alternate ISO15745ProfileContainer master profile template specified in amendment 1[7)] of ISO 15745-1:2003.

Figure F.1 shows the structure of an EtherCAT XML profile.



[a] Two types of ProfileBody are used: ProfileBody_Device_EtherCAT or ProfileBody_CommunicationNetwork_EtherCAT

Figure F.1 — EtherCAT profile template

The ProfileTechnology name is EtherCAT.

F.2 General rules

F.2.1 Using unique IDs

An element can have the attribute uniqueID of type xsd:ID. The unique identifier therefore is forced to be unique in the whole XML file. An element that references the unique identifier contains a named attribute of type xsd:IDREF.

Unique identifiers may be generated in two ways. One possibility is to build a string out of the element name and an up-counting number. A second way may be the concatenation of strings of parent elements. Both methods guarantee the uniqueness of the string.

F.2.2 Language support

F.2.2.1 General

Device profiles complying with the XML schema described in this annex need a support of different languages, since tools are then able to use names out of the XML file in order to display them in their user interface. Communication parameters for example may be presented in the user interface of a tool.

The language support is implemented via the label group g_labels. Each name of an element, which would possibly be displayed and is therefore language dependent, is provided inside the schema as a g_labels element. Optionally, a URI may be added as an attribute to the label element.

EXAMPLE

For a given parameter name:

— German: Baudrate

— English: Baud rate

— French: Vitesse de transmission

F.2.2.2 Element g_labels

The group g_labels supports the introduction of a label (name) and a description in the context of the parent element (see Figure F.2).

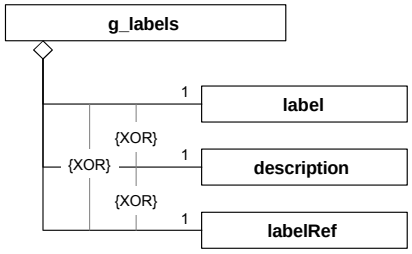


Figure F.2 — Group g_labels

Every element, which needs a name or a description, shall select one and only one of the three elements to perform this task: the label, the description and the labelRef element.

1) The label element allows storage of the identifying name and the descriptive text inside the XML file itself. The label element has the attributes given in Table F.1.

Table F.1 — Attributes of element label

Attribute	Data type	Use	Description
lang	xsd:language	required	Language used for the name or the description
URI	xsd:anyURI	optional	Optional link to further descriptive information

The element may appear n times, once for each language. For identifying the language, the lang attribute is used.

2) The description element allows storage of textual descriptions inside the XML file itself. The element may appear several times, once for each language. The description element has the same attributes as the label element.

3) The labelRef element allows referencing of descriptive texts inside an external text source file.

   The labelRef element provides a pointer (via its attributes dictID and textID) to a text entry in a separate text source file. These text source files are referenced in the dictionary sub-elements of the DeviceFunction element. Text source files may be any files containing character sequences and other information, for example figures.

   The labelRef element also may appear n times, to allow references to several dictionary entries, which contain links to files in different languages. The respective language is defined in the lang attribute of the dictionary element.

   The labelRef element contains the attributes given in Table F.2.

Table F.2 — Attributes of element labelRef

Attribute	Data type	Use	Description
dictID	xsd:IDREF	required	References a single dictionary element inside the dictionaryList element; the dictionary element contains a link to the external text source file
textID	xsd:string	optional	References a character sequence inside the external text source file by pattern matching

F.2.2.3 Language identifier

For the multi-language support each label gets an attribute with the content of the language code. The language code corresponds to the content of the label element.

In order to verify which languages are supported in the XML file the attribute supportedLanguages in the ProfileBody element lists the supported languages.

F.2.2.4 Attribute lang

The language identifier lang consists of a combination of a language code (as defined in ISO 639-1) plus an optional dash character plus an optional country code (as defined in ISO 3166-1). The attribute lang is an attribute of the label element.

Some of the values for lang are given in Table F.3.

Table F.3 — Values of attribute lang

Language	value of lang
English (United States)	en-us
German (Standard)	de
French (Standard)	fr
Spanish (Standard)	es
Italian (Standard)	It
Portuguese (Brazil)	pt-br

F.2.2.5 SupportedLanguages attribute

The supportedLanguages attribute identifies supported languages and consists of a list of language codes plus optional country codes.

EXAMPLE

supportedLanguages="en-us de fr es"

F.2.2.6 URIs

A general mechanism allows describing a URI in the context of a label element. The URI is implemented via an optional attribute URI.

EXAMPLE

For example this is used in the context of a vendor label, parameter label, or services label.

F.3 ProfileHeader

To facilitate the identification of a profile, the profile header of the device profile shall comply with the model shown in Figure F.3, which is directly inherited from ISO 15745-1.

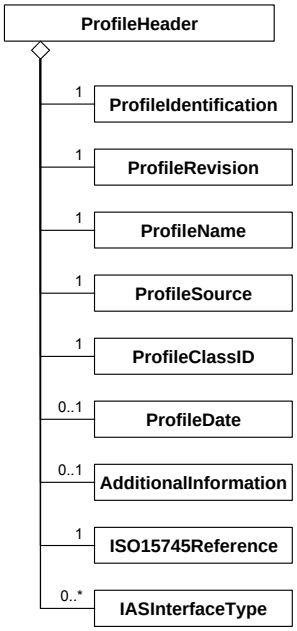


Figure F.3 — Profile header class diagram

The ProfileHeader element is composed of the following elements:

— the ProfileIdentification element identifies the current profile,

— the ProfileRevision element identifies the current profile revision,

— the ProfileName element contains a descriptive English name of the current profile. In case that more than one ProfileBody element are present within a device profile, it is suggested that the value of the

ProfileName element should be the concatenation of the values of the productName elements inside the respective DeviceIdentity elements,

— the ProfileSource element identifies the validator of the current profile,

— the ProfileClassID element identifies the class of the current profile according to ISO 15745-1,

— the ISO15745Reference element states the ISO 15745 part, edition and technology, to which this description conforms.

F.4 Device profile template description

F.4.1 ProfileBody_Device_EtherCAT

This part defines the EtherCAT device profile.

The ProfileBody_Device_EtherCAT contains the DeviceIdentity, the DeviceManager, the DeviceFunction and the ApplicationProcess elements shown in Figure 26.

The ProfileBody element contains the description of

— a single device (for example a proximity sensor or an electromechanical limit switch), or of a more complex one (for example a circuit breaker with up to 2500 parameters, more than 100 functions), or

— a part of a device also called "module" in the PLC world (for example part of an I/O controller or an electrical protection unit).

The ProfileBody element contains the attributes given in Table F.4.

Table F.4 — Attributes of element ProfileBody

Attribute	Data type	Use	Description
formatName	xsd:string	fixed	Format identifier
formatVersion	xsd:string	fixed	Format version identifier
fileName	xsd:string	required	Name of the file with extension without path
fileCreator	xsd:string	required	Person creating the file
fileCreationDate	xsd:date	required	Date of file creation
fileCreationTime	xsd:time	optional	Time of file creation
fileModifiedBy	xsd:string	optional	Person modifying the file
fileModificationDate	xsd.date	optional	Date of last file change
fileModificationTime	xsd:time	optional	Time of last file change
fileVersion	xsd:string	required	Vendor specific version of the file
supportedLanguages	xsd:NMTOKENS	optional	List of supported languages

F.4.2 DeviceIdentity

F.4.2.1 General

The DeviceIdentity class (see Figure 27) contains elements, which are independent of the network and of the process. It describes the identity of a single device or of a group of devices.

Table F.5 specifies the attribute readOnly, which is attached to the vendorName, vendorID, vendorText, deviceFamily, productFamily, productName, productID, productText, orderNumber, version, specificationRevision, and instanceName elements.

Table F.5 — Attribute of element vendorName

Attribute	Data type	Use	Description
readOnly	xsd:boolean	default	Indicates whether the value is read-only for a user: false, true (default)

F.4.2.2 Element vendorName

The vendorName element identifies the name or the brand name of the vendor of the device.

F.4.2.3 Element vendorID

The vendorID element identifies the vendor. This information has to be filled in when the product described is recognized and validated by a consortium.

NOTE Consortia specific product families and vendor identifiers are linked.

F.4.2.4 Element vendorText

The vendorText element allows the vendor to provide additional company information, like address or hotline number. The g_labels group offers the possibility to include a vendor URI inside the vendorText element.

F.4.2.5 Element deviceFamily

The deviceFamily element states the family of the device.

EXAMPLE

Examples for device families are:

— Variable speed drive,

— Circuit breaker,

— Pressure sensor.

F.4.2.6 Element productFamily

The productFamily element states a vendor specific affiliation of the device type to a certain set of devices inside a family. The list of valid productFamily values is system, tool or consortia specific.

NOTE Consortia specific product families and vendor identifiers are linked.

F.4.2.7 Element productName

The productName element states a vendor specific designation or name of the device type.

F.4.2.8 Element productID

The productID element states a vendor specific unique identification for the device type described.

F.4.2.9 Element productText

The productText element allows the vendor to provide a short textual description of the device type.

F.4.2.10 Element orderNumber

The orderNumber element is used to store the single order number of a given product or the set of different order numbers of the products of a product family, depending upon whether the device profile describes a product or a product family.

F.4.2.11 Element version

The version element is used to store different types of version information. Multiple version elements are possible.

The version element has the attributes given in Table F.6.

Table F.6 — Attributes of element version

Attribute	Data type	Use	Description
versionType	xsd:NMTOKEN	required	Type of version: — SW – Software — FW – Firmware — HW – Hardware
readOnly	xsd:boolean	default	Indicates whether the value is read-only for a user: false, true (default)

F.4.2.12 Element buildDate

The buildDate element specifies the build date of the software unit.

F.4.2.13 Element specificationRevision

The specificationRevision element contains the revision of the specification, to which this device conforms.

F.4.2.14 Element instanceName

This element contains the instance name of the device.

F.4.3 DeviceManager

F.4.3.1 General

The DeviceManager element defines the list of indicators provided by the device type, if any.

F.4.3.2 LEDList

F.4.3.2.1 General

The LEDList element shown in Figure F.4 specifies the number and type of indicators, which are provided by a device type.

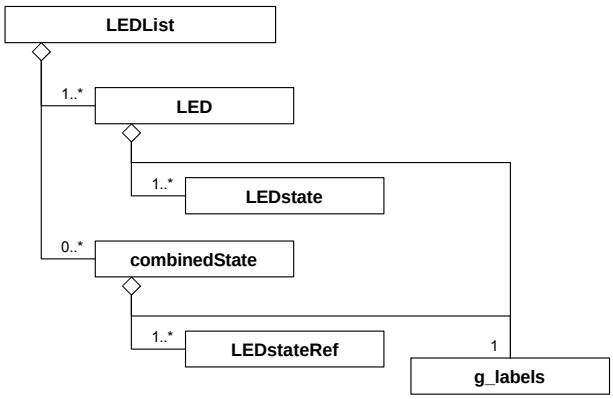


Figure F.4 —LEDList

F.4.3.2.2 LED

The LED element describes the features of a single LED of the device type. A detailed feature description may be provided through the g_labels group.

Further properties of the LED are represented as attributes of the LED element given in Table F.7.

Table F.7 — Attributes of element LED

Attribute	Data type	Use	Description
LEDcolors	xsd:string	required	Colours of the LED; valid values are monocolor and bicolor
LEDtype	xsd:string	optional	Rough classification of the supervised item or functionality; valid values are IO, device and communication

In addition to the descriptive parts introduced above, the LED element contains one to many LEDstate elements, which define the device states signalled by the LED and the visual behaviour used for signalling the states.

The visual behaviour used for signalling the state is encoded as attribute values of the LEDstate element, as given in Table F.8. Additionally a unique ID is allocated for the LED state.

Table F.8 — Attributes of element LEDstate

Attribute	Data type	Use	Description
uniqueID	xsd:ID	required	Unique ID for the LED state; may be referenced from an LEDstateRef element
state	xsd:string	required	State of the LED; possible attribute values: On, off, flashing
LEDcolor	xsd:string	required	Colour of the LED state; valid values: green, amber, red
flashingPeriod	xsd:unsignedInt	optional	If state is flashing: flashing period of the LED in milliseconds
impulsWidth	xsd:unsignedByte	default	Width of the flashing impulse given in percent (%) of the flashing period; if the attribute impulsWidth is missing, the default value is 50 (%)
numberOfImpulses	xsd:unsignedByte	default	Number of impulses in case that more than one flashing impulse is inside one flashing period; if the attribute is present, the attribute impulsWidth shall be present also if the attribute numberOfImpulses is missing, the default value is 1

F.4.3.2.3 Element combinedState

The combinedState element allows the indication of device states, which are signalled by more than one LED.

The description of the combined state is provided through the g_labels group.

The LED states participating in the signalling of the combined state are referenced by means of at least two LEDstateRef sub-elements of the combinedState element.

The reference to a LEDstate element is encoded as the attribute value of the single attribute of the LEDstateRef element (see Table F.9).

Table F.9 — Attributes of element LEDstateRef

Attribute	Data type	Use	Description
stateIDRef	xsd:IDREF	required	Unique ID of the referenced LEDstate element

F.4.4 DeviceFunction

F.4.4.1 General

The DeviceFunction element shown in Figure 29 defines the catalogue view of the device, presented as a set of capabilities listing both device characteristics and compliance with various standards.

F.4.4.2 Element capabilities

F.4.4.2.1 General

The mandatory capabilities element describes all functionalities, their characteristics, and the important parameters of the device, that need to be known by tools which use the device profile to select products with the same or similar properties.

The capabilities element describes device features in a purely textual form. It contains a sequence of one to many characteristicsList elements and an optional standardComplianceList element.

F.4.4.2.2 Element characteristicsList

F.4.4.2.2.1 General

The characteristicsList element is a collection of characteristics. The element shall contain at least one characteristic sub-element. The characteristics inside a list may be associated with a category, which can be expressed as textual content of the g_labels sub-element of the optional category sub-element of the characteristicsList element.

F.4.4.2.2.2 Element characteristic

The characteristic element describes a single characteristic of a device. It contains a mandatory characteristicName element and one to many characteristicContent elements.

F.4.4.2.2.3 Element characteristicName

The mandatory characteristicName element denotes a major technical characteristic of the device. The vocabulary used in the product data sheet is recommended for the names of characteristics.

EXAMPLE

"Maximum operational voltage", "Overload protection", "Electrical durability".

F.4.4.2.2.4 Element characteristicContent

This mandatory element contains a value for the characteristic. Multiple values may be expressed by using multiple characteristicContent elements.

EXAMPLE

An example of a single value for "Maximum operational voltage" is 680V.

F.4.4.2.3 Element standardComplianceList

The standardComplianceList element is a collection of compliantWith elements. The element itself is optional; if it exists, it shall contain at least one compliantWith sub-element.

The compliantWith sub-element has attributes, which state the compliance of the device with an international or company internal standard. The content of type g_labels of this element may contain remarks concerning that standard.

The name or number of the standard is provided through the required name attribute of the compliantWith element. The second, default valued range attribute of the compliantWith element defines the range of applicability of the standard as given in Table F.10.

Table F.10 — Attributes of element compliantWith

Attribute	Data type	Use	Description
name	xsd:string	required	Name or number of the standard
range	xsd:NMTOKEN	default	The two possible enumerated values of the attribute are international (default) or internal

F.4.4.3 Element picturesList

The picturesList element offers the possibility to link pictures to the device profile. It contains one to many picture sub-elements, whose caption is provided via a g_labels sub-element.

Table F.11 defines attributes of the picture sub-element: an optional number of the picture, and the mandatory link to an external resource containing the graphical information.

Table F.11 — Attributes of element picture

Attribute	Data type	Use	Description
URI	xsd:anyURI	required	Link to the external resource
number	xsd:unsignedInt	optional	Number of the picture

F.4.4.4 Element dictionaryList

The optional dictionaryList element offers the possibility to include links to external text source files to the device profile. It contains one to many dictionary elements, where each one contains one to many file sub-elements. Several files are necessary if different file formats are needed within a dictionary.

A mandatory lang attribute of type xsd:language defines the language used in the files which are linked to the dictionary element (see Table F.12). A mandatory uniqueID attribute of type xsd:ID holds the unique identification of the dictionary entry which is referenced from the attribute dictID of a labelRef element as given in Table F.2.

Table F.12 — Attributes of element dictionary

Attribute	Data type	Use	Description
lang	xsd:language	required	Language used for files belonging to a dictionary entry
uniqueID	xsd:ID	required	Unique ID of the dictionary entry

A file sub-element contains a single mandatory attribute given in Table F.13.

Table F.13 — Attributes of element file

Attribute	Data type	Use	Description
URI	xsd:anyURI	required	Link to the respective file

F.4.5 ApplicationProcess

F.4.5.1 General

The ApplicationProcess element represents the set of services and parameters, which constitute the behaviour and the interfaces of the device in terms of the application, independent of the device technology and the underlying communication networks and communication protocols.

The sub-elements of the ApplicationProcess element in Figure 30 provide a generic approach for the description of arbitrary, flat or hierarchically structured functions of a device.

Functions are modelled as function types, which are instantiated within the device or - if hierarchical structures are needed - inside function types. The interface variables of these function instances, which may be of simple or complex data type, are associated with the parameters of the device by building a reference from the parameter to the respective interface variable of the function instance, in flat as well as in hierarchical structures.

The ApplicationProcess element contains up to five lists of items (see Figure 30):

— two lists which define data types (optional) and function types (required),

— one required list which defines the function instances on device level (possibly including connections between instances),

— one required list which defines the device parameters, and

— one optional list which defines parameter groups (combinations of parameters for specific purposes).

F.4.5.2 Element dataTypeList

F.4.5.2.1 General

The optional dataTypeList element is present if complex data types like arrays or data structures are needed inside variable declarations of the device profile.

If present, the dataTypeList element shown in Figure F.5 contains a sequence of one to many elements out of the choice of:

— an array element,

— a struct element,

— an enum element, or

— a derived element.

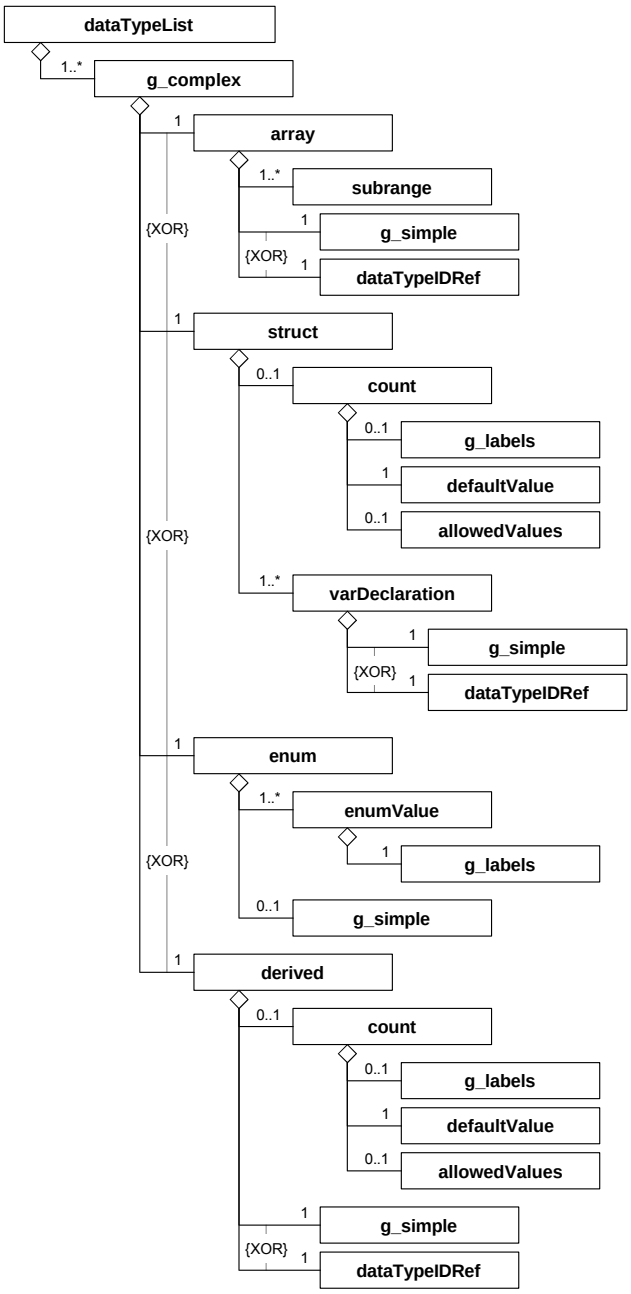


Figure F.5 — dataTypeList

F.4.5.2.2 Common elements

F.4.5.2.2.1 Group g_simple

The group g_simple contains a choice of elements, whose names represent the names of all simple data types allowed in the definition of variables inside a device profile. The simple data types conform to the elementary data types defined in IEC 61131-3; the data types BITSTRING and CHAR (=STRING[1]) are added.

These elements are introduced inside a group to allow their placement directly as a sub-element of the array element (or of the varDeclaration element, see F.4.5.4.3.2).

F.4.5.2.2.2 Element count

The count element defines the number of used units of the base type of the derived type. Multilingual names and/or descriptions for the count element are provided through the group g_labels. See F.2.2.2 for the description of the group g_labels.

The count is described by:

— its attributes,

— the mandatory sub-element defaultValue and a possibly empty set of sub-elements g_labels and allowedValues.

The number of units is expressed as the value of the defaultValue attribute of the count element. The allowedValue attribute defines the range of values for the default value.

The sub-elements defaultValue and allowedValues are described in F.4.5.6.2.5 and in F.4.5.6.2.7.

The count element shall contain the attributes given in Table F.14.

Table F.14 — Attributes of element count

Attribute	Data type	Use	Description
uniqueID	xsd:ID	required	Unique ID of the count
access	xsd:NMTOKEN	default	Defines which operations are valid for the count: — read - read access only (default value) — write - write access only — readWrite - both read and write access — noAccess - access denied

F.4.5.2.3 Element array

F.4.5.2.3.1 General

The array element serves to describe an array data type, which may be referenced from an interface variable of a function type, from another array type definition, or from a component variable inside the definition of a structured data type.

The array element contains at least one subrange element and either an element describing a simple data type out of the group g_simple, or an element dataTypeIDRef, which references one of the defined complex data types within the dataTypeList element.

For multi-dimensional arrays, several subrange elements will be present. In this case, the first subrange element in the sequence defines the subrange for the leftmost array index, and the last subrange element in the sequence defines the subrange for the rightmost array index.

The array element contains the attributes given in Table F.15.

Table F.15 — Attributes of element array

Attribute	Data type	Use	Description
name	xsd:string	required	Data type name of the array type
uniqueID	xsd:ID	required	Unique ID of the array type
description	xsd:string	optional	Optional textual description of the array type

F.4.5.2.3.2 Element subrange

The subrange element defines the lower and the upper limit of an array index for one dimension of the array. This element has no sub-elements.

The limit values of type xsd:long are contained in the two attributes of the subrange element given Table F.16.

Table F.16 — Attributes of element subRange

Attribute	Data type	Use	Description
lowerLimit	xsd:long	required	Lower limit of the subrange
upperLimit	xsd:long	required	Upper limit of the subrange

F.4.5.2.4 Element struct

F.4.5.2.4.1 General

The struct element serves to describe a structured data type, which may be referenced from an interface variable of a function type, from an array type definition, or from a component variable inside the definition of another structured data type.

The struct element contains a sequence of one to many varDeclaration elements, which define the components of the structured data type.

The struct element shall contain the attributes given in Table F.17.

Table F.17 — Attributes of element struct

Attribute	Data type	Use	Description
name	xsd:string	required	Data type name of the structured data type
uniqueID	xsd:ID	required	Unique ID of the structured data type
description	xsd:string	optional	Optional textual description of the structured data type

F.4.5.2.4.2 Element varDeclaration

In the context of the definition of a structured data type, the varDeclaration element describes a single component variable (member) of the structure.

In the context of the definition of the interface of a function, the varDeclaration element describes a single interface variable of the function type.

The data type of the component variable or interface variable is either defined by an element describing a simple data type out of the group g_simple, or by an element dataTypeIDRef, which references one of the defined complex data types within the dataTypeList element.

All further properties of the variable are contained in the attributes of the varDeclaration element, as given in Table F.18.

Table F.18 — Attributes of element varDeclaration

Attribute	Data type	Use	Description
name	xsd:string	required	Name of the interface variable or structure component
uniqueID	xsd:ID	required	Unique ID of the interface variable or structure component (see NOTE 1)
size	xsd:string	optional	Number of elements, if the interface variable or structure component is of type anonymous ARRAY, BITSTRING, STRING or WSTRING (see NOTE 2)
initialValue	xsd:string	optional	Initial value of the interface variable or structure component (see NOTE 3)
description	xsd:string	optional	Optional textual description of the interface variable or structure component

NOTE 1 When creating the unique ID for a variable, it is essential that the ID is unique over all created IDs inside the XML source file. To allow equal names for component variables of different data structures and equal names for interface variables of function types, the ID of a variable should generally concatenate the type name of the structured data type or the function type with the variable name, to guarantee uniqueness.

NOTE 2 Anonymous types define the size of an array, bitstring or string directly in the variable declaration, and not through the reference to a named complex data type. In the case of an array, the type of a single array element is given by the data type of the variable. In the case of a bitstring, the single array element is a single bit. In the case of a string, the single array element is a single-byte resp. double-byte character.

NOTE 3 If present, this attribute defines the initial (default) value of the interface variable of the function type. It is overwritten by a given default value of a parameter associated with the interface variable of the function instance.

F.4.5.2.5 Element enum

F.4.5.2.5.1 General

The enum element serves to describe an enumerated data type, which may be referenced from an interface variable of a function type, from an array type definition, or from a component variable inside the definition of a structured data type.

According to Figure F.5, it contains a sequence of one to many enumValue elements, which define the enumeration constants of the enumerated data type. The data type of the enumeration constants is optionally defined by an element describing a simple data type out of the group g_simple.

The enum element contains the attributes given in Table F.19.

Table F.19 — Attributes of element enum

Attribute	Data type	Use	Description
name	xsd:string	required	Type name of the enumerated data type
uniqueID	xsd:ID	required	Unique ID of the enumerated data type
size	xsd:string	optional	Optional number of enumerated values of the enumerated data type
description	xsd:string	optional	Optional textual description of the enumerated data type

F.4.5.2.5.2 Element enumValue

The enumValue element defines the name(s) and optionally a numerical value of a single enumeration constant. The name(s) are specified through the g_labels group, whereas the value is contained in the single value attribute of the enumValue element, as given in Table F.20.

Table F.20 — Attributes of element enumValue

Attribute	Data type	Use	Description
value	xsd:string	optional	Optional attribute: fixed numerical value for the enumeration constant, represented as a string of characters

F.4.5.2.6 Element derived

The derived element serves to derive a new data type from a given base type.

The derived element contains an optional count element and either an element describing a simple data type out of the group g_simple, or an element dataTypeIDRef, which references one of the defined complex data types within the dataTypeList element.

If the count element is missing, the derived type definition just introduces a new type name for the respective base type. If the count element is present, it defines the number of units of the respective base type used to build the derived type (for example base type BITSTRING, count = 4 defines a derived type of size 4 bit).

The derived element contains the attributes given in Table F.21.

Table F.21 — Attributes of element derived

Attribute	Data type	Use	Description
name	xsd:string	required	Data type name of the derived type
uniqueID	xsd:ID	required	Unique ID of the derived type
description	xsd:string	optional	Optional textual description of the derived type

F.4.5.3 Element functionTypeList

If the optional ApplicationProcess element is present in the device profile, it contains a mandatory functionTypeList element shown in Figure F.6.

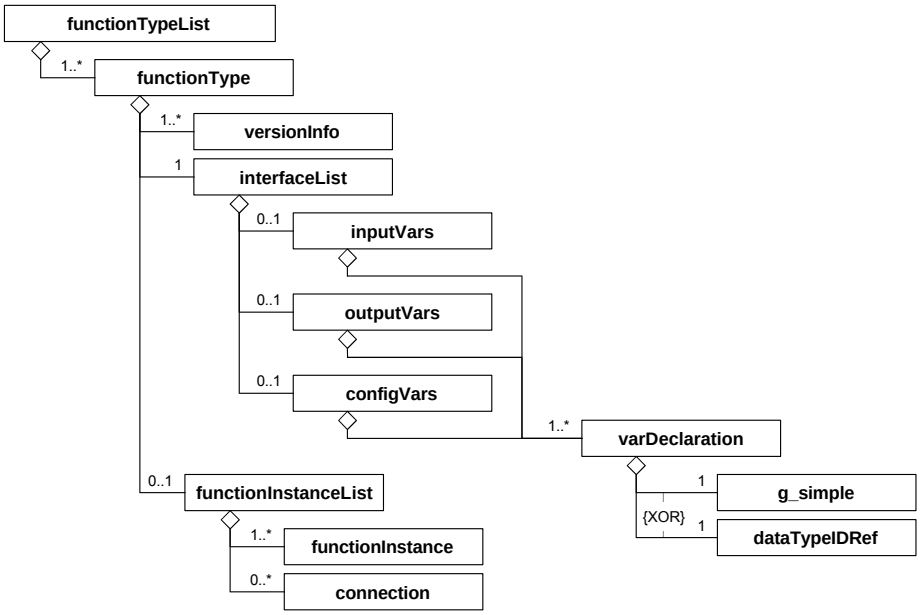


Figure F.6 — functionTypeList

The functionTypeList represents a sequence of one to many functionType elements.

Each of the functionType elements represents the type description of a device function, which is referenced from at least one instance of that function type inside a functionInstanceList element. References from more than one instance of the same function type are also possible.

The description of a function type contains all those objects and data which are common for all instances of a given function type.

EXAMPLE 1

Examples are the variable - or function parameter - objects that constitute the interface of the function (type respectively . instance).

EXAMPLE 2

Other examples are instances contained within the body of a function in a hierarchically structured functional description. These instances, which are located within a functionInstanceList element inside the function type, reference other function types in the list of function types.

F.4.5.4 Element functionType

F.4.5.4.1 General

The functionType element contains one to many versionInfo elements, a mandatory interfaceList element and an optional functionInstanceList element. The functionInstanceList element is only present within a functionType element, if the function is hierarchically structured.

Additionally, the functionType element shall contain the attributes given in Table F.22.

Table F.22 — Attributes of element functionType

Attribute	Data type	Use	Description
name	xsd:string	required	Type name of the function type
uniqueID	xsd:ID	required	Unique ID of the function type
description	xsd:string	optional	Optional textual description of the function type
package	xsd:string	optional	Optional textual association of the function type with a "package" or similar classification scheme - the usage of this attribute is left to the profile validator

F.4.5.4.2 Element versionInfo

The mandatory versionInfo element within the functionType element provides information on the versioning history of a function type (concerning the definition of the interface).

To keep track of the versioning history, the versionInfo element may be entered multiple times. The multiple entries shall be arranged within the functionType element in the following sequence:

a) the first entry represents the most recent version,

b) the second entry represents the immediately preceding version,

c) the last entry represents the first released version.

This element will be provided once at the creation of the description of the function type. New elements will only be added, if modifications of a function type are introduced, which lead to a modified version of the device profile.

The versionInfo element shall contain the attributes given in Table F.23.

Table F.23 — Attributes of element versionInfo

Attribute	Data type	Use	Description
organization	xsd:string	required	Name of the organisation maintaining the function type
version	xsd:string	required	Version identification in the versioning history; suggested format: "xx.yy" (xx,yy = 0..255)
author	xsd:string	required	Name of the person maintaining the function type
date	xsd:date	required	Date of this version
remarks	xsd:string	optional	Descriptive information concerning the specific step in the versioning history

F.4.5.4.3 Element interfaceList

F.4.5.4.3.1 General

The mandatory interfaceList element within the functionType element provides the definition of the interface of the function type. Elements of the interface are:

— the input variables, and/or

— the output variables, and/or

— the configuration variables

of the function type.

Consequently the interfaceList element contains a sequence of three elements, where each element represents lists of one to many variable declarations encoded as varDeclaration elements:

— one optional element inputVars,

— one optional element outputVars, and

— one optional element configVars.

Neither the interfaceList nor the inputVars, outputVars or configVars elements have any attributes.

F.4.5.4.3.2 Element varDeclaration

In the context of the definition of a structured data type, the varDeclaration element describes a single component variable (member) of the structure.

In the context of the definition of the interface of a function type, the varDeclaration element describes a single interface variable of the function type.

The data type of the component variable or interface variable is either defined by an element describing a simple data type out of the group g_simple, or by an element dataTypeIDRef, which references one of the defined complex data types within the dataTypeList element.

F.4.5.2.2.1 describes the group g_simple and F.4.5.4.3.3 describes the dataTypeIDRef element.

All further properties of the variable are contained in the attributes of the varDeclaration element, as given in Table F.24.

Table F.24 — Attributes of element varDeclaration

Attribute	Data type	Use	Description
name	xsd:string	required	Name of the interface variable or structure component
uniqueID	xsd:ID	required	Unique ID of the interface variable or structure component
size	xsd:string	optional	Number of elements, if the interface variable or structure component is of type anonymous ARRAY, BITSTRING, STRING or WSTRING
initialValue	xsd:string	optional	Initial value of the interface variable or structure component
description	xsd:string	optional	Optional textual description of the interface variable or structure component

F.4.5.4.3.3 Element dataTypeIDRef

The dataTypeIDRef element serves to reference a complex data type inside the dataTypeList element (see F.4.5.2), either from an interface variable of a function type, or from an array type definition, or from a component variable inside the definition of a structured data type.

The reference of type xsd:IDREF is provided as an attribute of the dataTypeIDRef element, as given in Table F.25.

Table F.25 — Attributes of element dataTypeIDRef

Attribute	Data type	Use	Description
uniqueIDRef	xsd:IDREF	required	Unique ID of the referenced data type

F.4.5.5 Element functionInstanceList

F.4.5.5.1 General

If the optional ApplicationProcess element is present in the device profile, it contains a mandatory functionInstanceList element, which contains a sequence of one to many functionInstance elements and zero to many connection elements.

At the application process level, the functionInstance elements represent the accessible application functions of the device type, independent of the network type or protocol. The connection elements represent connections - if any - between specific output and input variables of those function instances.

The functionInstanceList element also appears as an optional sub-element of the functionType element (see F.4.5.4). Like at the application process level, the functionInstanceList element in that case contains a sequence of one to many functionInstance elements and zero to many connection elements.

The functionInstanceList element is only present within a functionType element, if a function is hierarchically structured. In this case the functionInstance elements represent the internal functions contained in the function type, and the connection elements the optional internal connections. These functions and their optional connections would be instantiated together with the instantiation of the containing function type.

The functionInstanceList element does not have any attributes.

F.4.5.5.2 Element functionInstance

The mandatory functionInstance element does not contain any sub-elements.

The functionInstance element shall contain the attributes given in Table F.26.

Table F.26 — Attributes of element functionInstance

Attribute	Data type	Use	Description
name	xsd:string	required	Name of the function instance
uniqueID	xsd:ID	required	Unique ID of the function instance (see NOTE)
typeIDRef	xsd:IDREF	required	Unique ID of the referenced function type
description	xsd:string	optional	Optional textual description of the function instance
NOTE When creating the unique ID for a function instance, it is essential that the ID is unique over all created IDs inside the XML source file. To allow equal names for function instances inside different function types, the ID of a function instance should generally concatenate the name of the containing function type with the instance name, to guarantee uniqueness.			

F.4.5.5.3 Element connection

The optional connection element defines a connection between an output variable of a function instance and an input variable of another function instance. Inside function types, the connection may also be drawn between an input variable of the function type and an input variable of a contained function instance, or between an output variable of a contained function instance and an output variable of the function type. The connection element may appear zero to many times.

The connection element contains the attributes given in Table F.27.

Table F.27 — Attributes of element connection

Attribute	Data type	Use	Description
source	xsd:string	required	Starting point of connection
destination	xsd:string	required	Endpoint of connection
description	xsd:string	optional	Optional textual description of the connection

EXAMPLE

The values of the source and the destination attributes may be used to encode the starting point and the endpoint of a connection using the syntax <function_instance_name>'.'<variable_name>; example for the value of a source attribute: 'PowerMeasures.Frequency'. Connections to interface variables of a function type use the names of the interface variables only.

F.4.5.6 Element parameterList

F.4.5.6.1 General

If the optional ApplicationProcess element is present in the device profile, it contains a mandatory parameterList element shown in Figure F.7, which represents a sequence of one to many parameter elements.

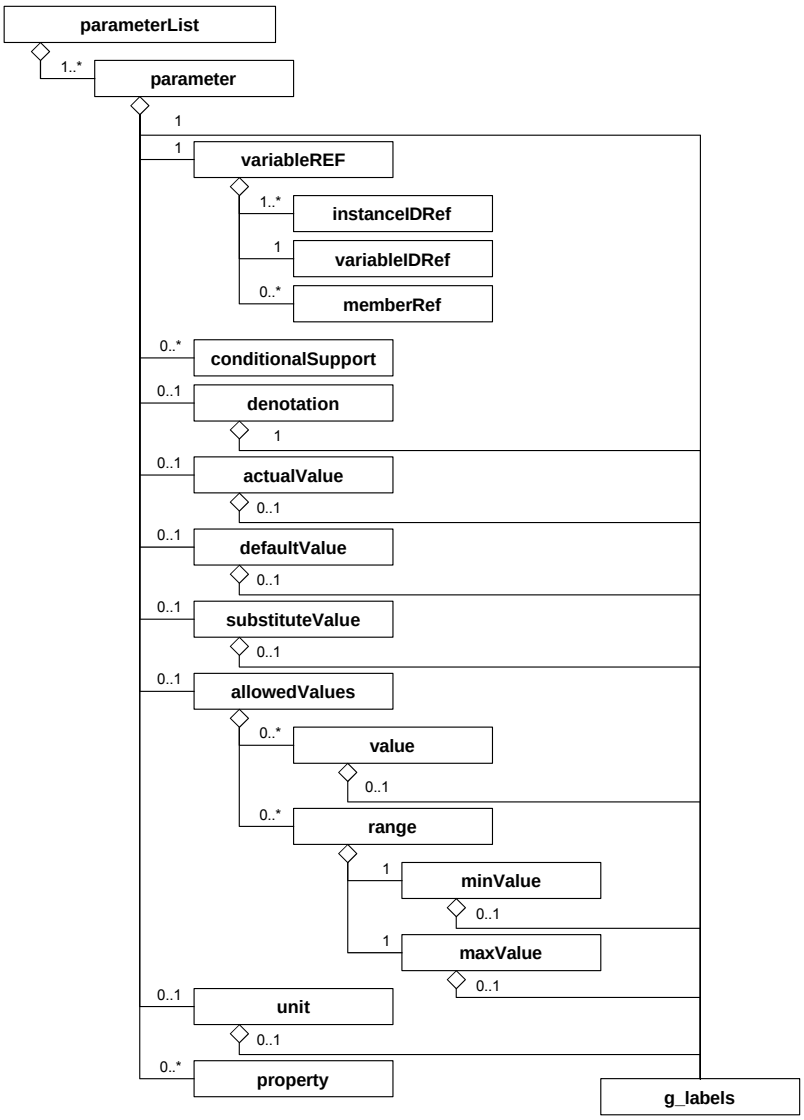


Figure F.7 — parameterList

Each of the parameter elements represents a parameter of the device profile. Multilingual names and/or descriptions for the parameters are provided through the group g_labels. F.2.2.2 describes the group g_labels.

The parameter is described by:

— its attributes,

— a reference to one (or more) interface variable(s) of one (or more) function instance(s) (mandatory element variableRef),

— a possibly empty set of sub-elements (conditionalSupport, denotation, actualValue, defaultValue, substituteValue, allowedValues, unit, property and g_labels).

NOTE References to multiple variables are a special case: specific parameters may reference an output variable of one function instance and an input variable of another function instance at the same time. In this case the data types of the two variables shall be the same. The XML parser cannot check the equality of data types. This can only be checked by a supporting tool.

F.4.5.6.2 Element parameter

F.4.5.6.2.1 General

The parameter element shall contain the attributes given in Table F.28.

Table F.28 — Attributes of element parameter

Attribute	Data type	Use	Description
uniqueID	xsd:ID	required	Unique ID of the parameter
access	xsd:NMTOKEN	default	Defines which operations are valid for the parameter: — read - read access only (default value) — write - write access only — readWrite - both read and write access — noAccess - access denied
support	xsd:NMTOKEN	optional	Defines whether or not the parameter has to be implemented in the device; valid values: — mandatory - parameter implementation is required — optional - parameter implementation is possible but not required — conditional - parameter implementation is required if one or more other optional parameter(s) is (are) implemented; these parameters are specified using the sub-element conditionalSupport
persistence	xsd:boolean	default	Defines the behaviour after power failure; valid values are false (default) and true
offset	xsd:string	optional	Offset which is added to an actual value to form a scaled value: EngineeringValue = (ParameterValue + offset) * multiplier; if not present, offset = 0 is assumed
multiplier	xsd:string	optional	Scaling factor by which an actual value is multiplied to form a scaled value: EngineeringValue = (ParameterValue + offset) * multiplier; if not present, multiplier = 1 is assumed

F.4.5.6.2.2 Element conditionalSupport

One or more conditionalSupport elements are present only if the value of the support attribute of the parameter element is conditional. Each element refers to a single optional parameter. If at least one of those optional parameters is implemented, the conditional parameter has also to be implemented.

The element conditionalSupport shall contain the single attribute given in Table F.29.

Table F.29 — Attributes of element conditionalSupport

Attribute	Data type	Use	Description
paramIDRef	xsd:IDREF	required	Unique ID of the referenced optional parameter

F.4.5.6.2.3 Element denotation

The denotation element serves to hold application-specific, multilingual names of the parameter. The names are provided through the mandatory g_labels sub-element. It is also possible to add multilingual descriptive information. The element denotation does not have any attributes.

F.4.5.6.2.4 Element actualValue

The actualValue element serves to hold the actual value of the parameter. An optional g_labels sub-element may provide multilingual descriptive information for this value. The value itself is provided in the value attribute of the element actualValue. An offset and multiplier may also be provided.

The attributes of the element actualValue shall be as given in Table F.30.

Table F.30 — Attributes of element actualValue

Attribute	Data type	Use	Description
value	xsd:string	required	Actual value
offset	xsd:string	optional	Offset which is added to an actual value to form a scaled value: EngineeringValue = (value + offset) * multiplier; if not present, the respective value of the parameter element shall be used
multiplier	xsd:string	optional	Scaling factor by which an actual value is multiplied to form a scaled value: EngineeringValue = (value + offset) * multiplier; if not present, the respective value of the parameter element shall be used

F.4.5.6.2.5 Element defaultValue

The defaultValue element serves to hold the default value of the parameter. This value overwrites the initial value of the interface variable of the function type associated with the parameter.

An optional g_labels sub-element may provide multilingual names and/or descriptive information for this value. The value itself is provided by the value attribute of the element defaultValue. An offset and multiplier may also be provided.

The attributes of the element defaultValue shall be as given in Table F.30.

F.4.5.6.2.6 Element substituteValue

The substituteValue element defines a specific value of the parameter that is provided to the device application in certain device operating states (for example on device fault).

An optional g_labels sub-element may provide multilingual names and/or descriptive information for this value. The value itself is provided by the value attribute of the element substituteValue. An offset and multiplier may also be provided.

The attributes of the element substituteValue shall be as given in Table F.30.

F.4.5.6.2.7 Element allowedValues

The allowedValues element defines a list of supported values and/or a single range or several ranges of supported values for the parameter.

The list of supported values is represented by zero to many value sub-elements of the allowedValues element, whereas the ranges are represented by zero to many range sub-elements of the allowedValues element.

The value sub-element holds a single allowed value of the parameter. An optional g_labels sub-element may provide multilingual names and/or descriptive information for this value. The value itself is provided by the value attribute of the element value. An offset and multiplier may also be provided.

The attributes of the element value shall be as given in Table F.30.

The range sub-element contains two required sub-elements, namely the element minValue and the element maxValue, which define the limits of that range of allowed values. The minValue and the maxValue elements have the same structure and attributes as the value sub-element of the allowedValues element. Therefore the description of the value sub-element and Table F.30 are also valid for these sub-elements.

F.4.5.6.2.8 Element unit

The unit element defines the engineering unit of a parameter (fro example time, temperature, pressure, flow, acceleration, current, energy), as specified in ISO 1000. An optional g_labels sub-element may provide multilingual names and/or descriptive information for the engineering unit.

The attributes of the element unit shall be as given in Table F.31.

Table F.31 — Attributes of element unit

Attribute	Data type	Use	Description
multiplier	xsd:string	required	Multiplier for engineering units of analog parameters
unitURI	xsd:anyURI	optional	Link to the respective unit definition in a file containing all engineering units (for example time, temperature, pressure, flow, acceleration, current, energy…), as standardized by ISO 1000

F.4.5.6.2.9 Element variableRef

The variableRef element builds a reference to an interface variable of a function instance, or, if the variable is an array or a structure, possibly a reference to a member of the variable (array element or structure component).

In a hierarchically structured ApplicationProcess element, function instances can be located inside function instances of other function types. Therefore a specific instance in the functional tree can only be accessed by stepping through the tree, i.e. the specific instance shall be addressed through a concatenation of instance names. To map this concatenation and allow the referencing of a member, the variableRef element contains:

— a sequence of one to many instanceIDRef elements, followed by

— a single mandatory variableIDRef element, and

— an optional memberRef element.

The variableRef element has the attribute given in Table F.32.

Table F.32 — Attribute of element variableRef

Attribute	Data type	Use	Description
position	xsd:unsignedByte	default	Defines the sequence of multiple mapped data items into a single parameter object; position=1 means starting the mapping at the lowest bit position; the number of bits is defined by the data type of the data item; subsequent data items are packed without gaps; default value: 1 (see Note)
NOTE Attribute can be omitted for a single mapped data item.			

F.4.5.6.2.10 Element instanceIDRef

The instanceIDRef element serves to reference a function instance inside a functionInstanceList element, which may reside either on the level of the ApplicationProcess element or on the level of a functionType element.

The reference of type xsd:IDREF is provided as an attribute of the instanceIDRef element, as given in Table F.33.

Table F.33 — Attributes of element instanceIDRef

Attribute	Data type	Use	Description
uniqueIDRef	xsd:IDREF	required	Unique ID of the referenced function instance

F.4.5.6.2.11 Element variableIDRef

The variableIDRef element serves to reference an interface variable of a function type inside the functionTypeList element.

In a given variableRef element the instance of that function type is defined by the functionInstance element referenced by the instanceIDRef element, which is just preceding the variableIDRef element.

The reference of type xsd:IDREF is provided as an attribute of the variableIDRef element, as given in Table F.34.

Table F.34 — Attributes of element variableIDRef

Attribute	Data type	Use	Description
uniqueIDRef	xsd:IDREF	required	Unique ID of the referenced interface variable of a function type

F.4.5.6.2.12 Element memberRef

The optional memberRef element either references the respective component of an interface variable of structured data type (attribute uniqueIDRef is used), or the respective array element of an interface variable of array data type (attribute index is used). One of these two attributes shall be present if the memberRef element is present.

The memberRef element shall contain the attributes given in Table F.35.

Table F.35 — Attributes of element memberRef

Attribute	Data type	Use	Description
uniqueIDRef	xsd:IDREF	optional	Unique ID of the referenced component of a structured data type
index	xsd:long	optional	Index of the referenced array element

F.4.5.6.3 Element property

The property element is introduced as a generic element to allow the inclusion of values for additional specific properties into the description of a parameter.

The property element shall contain the attributes given in Table F.36.

Table F.36 — Attributes of element property

Attribute	Data type	Use	Description
name	xsd:string	required	Name of the property
value	xsd:string	required	Value of the property

F.4.5.7 Element parameterGroupList

F.4.5.7.1 General

The optional parameterGroupList element, if present, contains a sequence of one to many parameterGroup elements, as shown in Figure F.8. Multilingual names and/or descriptions for the parameter groups are provided through the group g_labels. See F.2.2.2 for the description of the group g_labels.

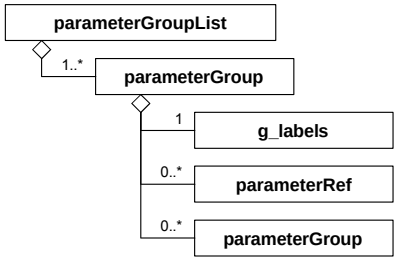


Figure F.8 — parameterGroupList

F.4.5.7.2 Element parameterGroup

Each of the parameterGroup elements combines a set of parameters out of the parameterList element to build a group of parameters which serve a specific purpose, for example to prepare HMI views. This purpose is indicated by the value of the kindOfAccess attribute of the parameterGroup element. It is possible to define a hierarchy of parameter groups.

The respective parameters in the set are referenced through the corresponding number of parameterRef elements.

The parameterGroup element contains the attributes given in Table F.37.

Table F.37 — Attributes of element parameterGroup

Attribute	Data type	Use	Description
uniqueID	xsd:ID	required	Unique ID of the parameter group
kindOfAccess	xsd:string	optional	Classifies the parameters of the parameter group

F.4.5.7.3 Element parameterRef

The parameterRef element serves to reference a parameter element inside the parameterList element of the ApplicationProcess element.

The reference of type xsd:IDREF is provided as an attribute of the parameterRef element, as given in Table F.38.

Table F.38 — Attributes of element parameterRef

Attribute	Data type	Use	Description
uniqueIDRef	xsd:IDREF	required	Unique ID of the referenced parameter

F.4.6 EtherCAT device profile template schemas

F.4.6.1 XML schema: ISO15745ProfileContainer.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <xsd:element name="ISO15745ProfileContainer">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ISO15745Profile" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="ISO15745Profile">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ProfileHeader" type="ProfileHeader_DataType"/>
            <xsd:element name="ProfileBody" type="ProfileBody_DataType"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:annotation>
      <xsd:documentation>* HEADER SECTION *</xsd:documentation>
   </xsd:annotation>
   <xsd:complexType name="ProfileHeader_DataType">
      <xsd:sequence>
         <xsd:element name="ProfileIdentification" type="xsd:string"/>
         <xsd:element name="ProfileRevision" type="xsd:string"/>
         <xsd:element name="ProfileName" type="xsd:string"/>
         <xsd:element name="ProfileSource" type="xsd:string"/>
         <xsd:element name="ProfileClassID" type="ProfileClassID_DataType"/>
         <xsd:element name="ProfileDate" type="xsd:date" minOccurs="0"/>
         <xsd:element name="AdditionalInformation" type="xsd:anyURI" minOccurs="0"/>
         <xsd:element name="ISO15745Reference" type="ISO15745Reference_DataType"/>
         <xsd:element name="IASInterfaceType" type="IASInterface_DataType" minOccurs="0"
maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:annotation>
      <xsd:documentation>* BODY SECTION *</xsd:documentation>
   </xsd:annotation>
   <xsd:complexType name="ProfileBody_DataType" abstract="true"/>
   <xsd:annotation>
      <xsd:documentation>* HEADER DATA TYPES *</xsd:documentation>
   </xsd:annotation>
   <xsd:simpleType name="ProfileClassID_DataType">
      <xsd:restriction base="xsd:string">
         <xsd:enumeration value="AIP"/>
         <xsd:enumeration value="Process"/>
         <xsd:enumeration value="InformationExchange"/>
         <xsd:enumeration value="Resource"/>
         <xsd:enumeration value="Device"/>
         <xsd:enumeration value="CommunicationNetwork"/>
         <xsd:enumeration value="Equipment"/>
         <xsd:enumeration value="Human"/>
         <xsd:enumeration value="Material"/>
      </xsd:restriction>
   </xsd:simpleType>
   <xsd:complexType name="ISO15745Reference_DataType">
      <xsd:sequence>
         <xsd:element name="ISO15745Part" type="xsd:positiveInteger"/>
         <xsd:element name="ISO15745Edition" type="xsd:positiveInteger"/>
         <xsd:element name="ProfileTechnology" type="xsd:string"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:simpleType name="IASInterface_DataType">
```

```
      <xsd:union>
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="CSI"/>
               <xsd:enumeration value="HCI"/>
               <xsd:enumeration value="ISI"/>
               <xsd:enumeration value="API"/>
               <xsd:enumeration value="CMI"/>
               <xsd:enumeration value="ESI"/>
               <xsd:enumeration value="FSI"/>
               <xsd:enumeration value="MTI"/>
               <xsd:enumeration value="SEI"/>
               <xsd:enumeration value="USI"/>
            </xsd:restriction>
         </xsd:simpleType>
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:length value="4"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:union>
   </xsd:simpleType>
   <xsd:annotation>
      <xsd:documentation>* ISO 15745 DEFINED DATA TYPES *</xsd:documentation>
   </xsd:annotation>
   <xsd:complexType name="ProfileHandle_DataType">
      <xsd:sequence>
         <xsd:element name="ProfileIdentification" type="xsd:string"/>
         <xsd:element name="ProfileRevision" type="xsd:string"/>
         <xsd:element name="ProfileLocation" type="xsd:anyURI" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:schema>
```

F.4.6.2 XML schema: CommonElements.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <!--################ common attribute group-->
   <xsd:attributeGroup name="ag_formatAndFile">
      <xsd:attribute name="formatName" type="xsd:string" fixed="DDXML" form="unqualified"/>
      <xsd:attribute name="formatVersion" type="xsd:string" fixed="2.0" form="unqualified"/>
      <xsd:attribute name="fileName" type="xsd:string" use="required" form="unqualified"/>
      <xsd:attribute name="fileCreator" type="xsd:string" use="required" form="unqualified"/>
      <xsd:attribute name="fileCreationDate" type="xsd:date" use="required" form="unqualified"/>
      <xsd:attribute name="fileCreationTime" type="xsd:time" use="optional"/>
      <xsd:attribute name="fileModificationDate" type="xsd:date" use="optional" form="unqualified"/>
      <xsd:attribute name="fileModificationTime" type="xsd:time" use="optional"/>
      <xsd:attribute name="fileModifiedBy" type="xsd:string" use="optional"/>
      <xsd:attribute name="fileVersion" type="xsd:string" use="required" form="unqualified"/>
   </xsd:attributeGroup>
   <!--################ common groups-->
   <xsd:group name="g_labels">
      <xsd:sequence>
         <xsd:choice maxOccurs="unbounded">
            <xsd:element name="label">
               <xsd:complexType>
                  <xsd:simpleContent>
                     <xsd:extension base="xsd:string">
                        <xsd:attribute name="lang" type="xsd:language" use="required"/>
                        <xsd:attribute name="URI" type="xsd:anyURI" use="optional"/>
                     </xsd:extension>
                  </xsd:simpleContent>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="description">
               <xsd:complexType>
                  <xsd:simpleContent>
                     <xsd:extension base="xsd:string">
                        <xsd:attribute name="lang" type="xsd:language" use="required"/>
                        <xsd:attribute name="URI" type="xsd:anyURI" use="optional"/>
                     </xsd:extension>
                  </xsd:simpleContent>
               </xsd:complexType>
            </xsd:element>
```

```
            <xsd:element name="labelRef">
               <xsd:complexType>
                  <xsd:simpleContent>
                     <xsd:extension base="xsd:anyURI">
                        <xsd:attribute name="dictID" type="xsd:IDREF" use="required"/>
                        <xsd:attribute name="textID" type="xsd:string" use="optional"/>
                     </xsd:extension>
                  </xsd:simpleContent>
               </xsd:complexType>
            </xsd:element>
         </xsd:choice>
      </xsd:sequence>
   </xsd:group>
   <xsd:group name="g_simple">
      <xsd:choice>
         <xsd:element name="BOOL"/>
         <xsd:element name="BITSTRING"/>
         <xsd:element name="BYTE"/>
         <xsd:element name="CHAR"/>
         <xsd:element name="WORD"/>
         <xsd:element name="DWORD"/>
         <xsd:element name="LWORD"/>
         <xsd:element name="SINT"/>
         <xsd:element name="INT"/>
         <xsd:element name="DINT"/>
         <xsd:element name="LINT"/>
         <xsd:element name="USINT"/>
         <xsd:element name="UINT"/>
         <xsd:element name="UDINT"/>
         <xsd:element name="ULINT"/>
         <xsd:element name="REAL"/>
         <xsd:element name="LREAL"/>
         <xsd:element name="TIME"/>
         <xsd:element name="DATE"/>
         <xsd:element name="DT"/>
         <xsd:element name="TOD"/>
         <xsd:element name="STRING"/>
         <xsd:element name="WSTRING"/>
      </xsd:choice>
   </xsd:group>
   <!--################ common elements-->
   <xsd:element name="vendorID">
      <xsd:complexType>
         <xsd:simpleContent>
            <xsd:extension base="xsd:string">
               <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
            </xsd:extension>
         </xsd:simpleContent>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="deviceFamily">
      <xsd:complexType>
         <xsd:group ref="g_labels"/>
         <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="productID">
      <xsd:complexType>
         <xsd:simpleContent>
            <xsd:extension base="xsd:string">
               <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
            </xsd:extension>
         </xsd:simpleContent>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="version">
      <xsd:complexType>
         <xsd:simpleContent>
            <xsd:extension base="xsd:string">
               <xsd:attribute name="versionType" use="required">
                  <xsd:simpleType>
                     <xsd:restriction base="xsd:NMTOKEN">
                        <xsd:enumeration value="SW"/>
                        <xsd:enumeration value="FW"/>
                        <xsd:enumeration value="HW"/>
                     </xsd:restriction>
```

```
                  </xsd:simpleType>
               </xsd:attribute>
               <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
            </xsd:extension>
         </xsd:simpleContent>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="buildDate" type="xsd:date"/>
   <xsd:element name="specificationRevision">
      <xsd:complexType>
         <xsd:simpleContent>
            <xsd:extension base="xsd:string">
               <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
            </xsd:extension>
         </xsd:simpleContent>
      </xsd:complexType>
   </xsd:element>
</xsd:schema>
```

F.4.6.3 XML schema: ProfileBody_Device_EtherCAT.xsd

The XML schema ProfileBody_Device_EtherCAT.xsd includes the schema ISO15745ProfileContainer.xsd in F.4.6.1 and the schema CommonElements.xsd in F.4.6.2.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <xsd:include schemaLocation="ISO15745ProfileContainer.xsd"/>
   <xsd:include schemaLocation="CommonElements.xsd"/>
   <!--################ profile body device -->
   <xsd:complexType name="ProfileBody_Device_EtherCAT">
      <xsd:complexContent>
         <xsd:extension base="ProfileBody_DataType">
            <xsd:sequence>
               <xsd:element ref="DeviceIdentity" minOccurs="0"/>
               <xsd:element ref="DeviceManager" minOccurs="0"/>
               <xsd:element ref="DeviceFunction" maxOccurs="unbounded"/>
               <xsd:element ref="ApplicationProcess" minOccurs="0" maxOccurs="unbounded"/>
               <xsd:element name="ExternalProfileHandle" type="ProfileHandle_DataType" minOccurs="0"
maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attributeGroup ref="ag_formatAndFile"/>
            <xsd:attribute name="supportedLanguages" type="xsd:NMTOKENS" use="optional"/>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
   <!--################ device identity elements -->
   <xsd:element name="DeviceIdentity">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="vendorName"/>
            <xsd:element ref="vendorID" minOccurs="0"/>
            <xsd:element ref="vendorText" minOccurs="0"/>
            <xsd:element ref="deviceFamily" minOccurs="0"/>
            <xsd:element ref="productFamily" minOccurs="0"/>
            <xsd:element ref="productName"/>
            <xsd:element ref="productID" minOccurs="0"/>
            <xsd:element ref="productText" minOccurs="0"/>
            <xsd:element ref="orderNumber" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element ref="version" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element ref="buildDate" minOccurs="0"/>
            <xsd:element ref="specificationRevision" minOccurs="0"/>
            <xsd:element ref="instanceName" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="productFamily">
      <xsd:complexType>
         <xsd:simpleContent>
            <xsd:extension base="xsd:string">
               <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
            </xsd:extension>
         </xsd:simpleContent>
      </xsd:complexType>
   </xsd:element>
```

```
<xsd:element name="instanceName">
   <xsd:complexType>
      <xsd:simpleContent>
         <xsd:extension base="xsd:string">
            <xsd:attribute name="readOnly" type="xsd:boolean" default="false"/>
         </xsd:extension>
      </xsd:simpleContent>
   </xsd:complexType>
</xsd:element>
<xsd:element name="orderNumber">
   <xsd:complexType>
      <xsd:simpleContent>
         <xsd:extension base="xsd:string">
            <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
         </xsd:extension>
      </xsd:simpleContent>
   </xsd:complexType>
</xsd:element>
<xsd:element name="productName">
   <xsd:complexType>
      <xsd:simpleContent>
         <xsd:extension base="xsd:string">
            <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
         </xsd:extension>
      </xsd:simpleContent>
   </xsd:complexType>
</xsd:element>
<xsd:element name="productText">
   <xsd:complexType>
      <xsd:group ref="g_labels"/>
      <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="vendorName">
   <xsd:complexType>
      <xsd:simpleContent>
         <xsd:extension base="xsd:string">
            <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
         </xsd:extension>
      </xsd:simpleContent>
   </xsd:complexType>
</xsd:element>
<xsd:element name="vendorText">
   <xsd:complexType>
      <xsd:group ref="g_labels"/>
      <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
   </xsd:complexType>
</xsd:element>
<!--################ device manager elements -->
<xsd:element name="DeviceManager">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="indicatorList" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="indicatorList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="LEDList" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="LEDList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="LED" maxOccurs="unbounded"/>
         <xsd:element ref="combinedState" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="LED">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:group ref="g_labels"/>
         <xsd:element ref="LEDstate" maxOccurs="unbounded"/>
```

```
      </xsd:sequence>
      <xsd:attribute name="LEDcolors" use="required">
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="monocolor"/>
               <xsd:enumeration value="bicolor"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="LEDtype" use="optional">
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="IO"/>
               <xsd:enumeration value="device"/>
               <xsd:enumeration value="communication"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:attribute>
   </xsd:complexType>
</xsd:element>
<xsd:element name="LEDstate">
   <xsd:complexType>
      <xsd:group ref="g_labels"/>
      <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
      <xsd:attribute name="state" use="required">
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="on"/>
               <xsd:enumeration value="off"/>
               <xsd:enumeration value="flashing"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="LEDcolor" use="required">
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="green"/>
               <xsd:enumeration value="amber"/>
               <xsd:enumeration value="red"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="flashingPeriod" type="xsd:unsignedInt" use="optional"/>
      <xsd:attribute name="impulsWidth" type="xsd:unsignedByte" default="50"/>
      <xsd:attribute name="numberOfImpulses" type="xsd:unsignedByte" default="1"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="combinedState">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:group ref="g_labels"/>
         <xsd:element name="LEDstateRef" minOccurs="2" maxOccurs="unbounded">
            <xsd:complexType>
               <xsd:attribute name="stateIDRef" type="xsd:IDREF" use="required"/>
            </xsd:complexType>
         </xsd:element>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<!--################ device function elements -->
<xsd:element name="DeviceFunction">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="capabilities"/>
         <xsd:element ref="picturesList" minOccurs="0"/>
         <xsd:element ref="dictionaryList" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="capabilities">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="characteristicsList" maxOccurs="unbounded"/>
         <xsd:element ref="standardComplianceList" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
```

```
</xsd:element>
<xsd:element name="characteristicsList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="category" minOccurs="0">
            <xsd:complexType>
               <xsd:group ref="g_labels"/>
            </xsd:complexType>
         </xsd:element>
         <xsd:element ref="characteristic" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="characteristic">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="characteristicName"/>
         <xsd:element ref="characteristicContent" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="characteristicContent">
   <xsd:complexType>
      <xsd:group ref="g_labels"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="characteristicName">
   <xsd:complexType>
      <xsd:group ref="g_labels"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="standardComplianceList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="compliantWith" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="compliantWith">
   <xsd:complexType>
      <xsd:group ref="g_labels"/>
      <xsd:attribute name="name" type="xsd:string" use="required"/>
      <xsd:attribute name="range" default="international">
         <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
               <xsd:enumeration value="international"/>
               <xsd:enumeration value="internal"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:attribute>
   </xsd:complexType>
</xsd:element>
<xsd:element name="picturesList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="picture" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="picture">
   <xsd:complexType>
      <xsd:group ref="g_labels"/>
      <xsd:attribute name="URI" type="xsd:anyURI" use="required"/>
      <xsd:attribute name="number" type="xsd:unsignedInt" use="optional"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="dictionaryList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="dictionary" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="dictionary">
   <xsd:complexType>
      <xsd:sequence>
```

```
         <xsd:element ref="file" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="lang" type="xsd:language" use="required"/>
      <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="file">
   <xsd:complexType>
      <xsd:attribute name="URI" type="xsd:anyURI" use="required"/>
   </xsd:complexType>
</xsd:element>
<!--################ application process elements -->
<xsd:element name="ApplicationProcess">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="dataTypeList" minOccurs="0"/>
         <xsd:element ref="functionTypeList"/>
         <xsd:element ref="functionInstanceList"/>
         <xsd:element ref="parameterList"/>
         <xsd:element ref="parameterGroupList" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="dataTypeList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:group ref="g_complex" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="functionTypeList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="functionType" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="functionType">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="versionInfo" maxOccurs="unbounded"/>
         <xsd:element ref="interfaceList"/>
         <xsd:element ref="functionInstanceList" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="name" type="xsd:string" use="required"/>
      <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
      <xsd:attribute name="description" type="xsd:string" use="optional"/>
      <xsd:attribute name="package" type="xsd:string" use="optional"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="versionInfo">
   <xsd:complexType>
      <xsd:attribute name="organization" type="xsd:string" use="required"/>
      <xsd:attribute name="version" type="xsd:string" use="required"/>
      <xsd:attribute name="author" type="xsd:string" use="required"/>
      <xsd:attribute name="date" type="xsd:date" use="required"/>
      <xsd:attribute name="remarks" type="xsd:string" use="optional"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="interfaceList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="inputVars" minOccurs="0"/>
         <xsd:element ref="outputVars" minOccurs="0"/>
         <xsd:element ref="configVars" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="inputVars">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="varDeclaration" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="outputVars">
```

```
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="varDeclaration" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="configVars">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="varDeclaration" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="varDeclaration">
   <xsd:complexType>
      <xsd:choice>
         <xsd:group ref="g_simple"/>
         <xsd:element ref="dataTypeIDRef"/>
      </xsd:choice>
      <xsd:attribute name="name" type="xsd:string" use="required"/>
      <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
      <xsd:attribute name="size" type="xsd:string" use="optional"/>
      <xsd:attribute name="initialValue" type="xsd:string" use="optional"/>
      <xsd:attribute name="description" type="xsd:string" use="optional"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="dataTypeIDRef">
   <xsd:complexType>
      <xsd:attribute name="uniqueIDRef" type="xsd:IDREF" use="required"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="functionInstanceList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="functionInstance" maxOccurs="unbounded"/>
         <xsd:element ref="connection" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="functionInstance">
   <xsd:complexType>
      <xsd:attribute name="name" type="xsd:string" use="required"/>
      <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
      <xsd:attribute name="typeIDRef" type="xsd:IDREF" use="required"/>
      <xsd:attribute name="description" type="xsd:string" use="optional"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="connection">
   <xsd:complexType>
      <xsd:attribute name="source" type="xsd:string" use="required"/>
      <xsd:attribute name="destination" type="xsd:string" use="required"/>
      <xsd:attribute name="description" type="xsd:string" use="optional"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="parameterList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="parameter" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="parameter">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:group ref="g_labels"/>
         <xsd:element ref="variableRef" maxOccurs="unbounded"/>
         <xsd:element ref="conditionalSupport" minOccurs="0" maxOccurs="unbounded"/>
         <xsd:element ref="denotation" minOccurs="0"/>
         <xsd:element ref="actualValue" minOccurs="0"/>
         <xsd:element ref="defaultValue" minOccurs="0"/>
         <xsd:element ref="substituteValue" minOccurs="0"/>
         <xsd:element ref="allowedValues" minOccurs="0"/>
         <xsd:element ref="unit" minOccurs="0"/>
         <xsd:element ref="property" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="ag_parameter"/>
```

```
   </xsd:complexType>
</xsd:element>
<xsd:element name="variableRef">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="instanceIDRef" maxOccurs="unbounded"/>
         <xsd:element ref="variableIDRef"/>
         <xsd:element ref="memberRef" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="position" type="xsd:unsignedByte" default="1"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="instanceIDRef">
   <xsd:complexType>
      <xsd:attribute name="uniqueIDRef" type="xsd:IDREF"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="variableIDRef">
   <xsd:complexType>
      <xsd:attribute name="uniqueIDRef" type="xsd:IDREF"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="memberRef">
   <xsd:complexType>
      <xsd:attribute name="uniqueIDRef" type="xsd:IDREF" use="optional"/>
      <xsd:attribute name="index" type="xsd:long" use="optional"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="actualValue">
   <xsd:complexType>
      <xsd:group ref="g_labels" minOccurs="0"/>
      <xsd:attributeGroup ref="ag_value"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="conditionalSupport">
   <xsd:complexType>
      <xsd:attribute name="paramIDRef" type="xsd:IDREF" use="required"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="denotation">
   <xsd:complexType>
      <xsd:group ref="g_labels"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="defaultValue">
   <xsd:complexType>
      <xsd:group ref="g_labels" minOccurs="0"/>
      <xsd:attributeGroup ref="ag_value"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="substituteValue">
   <xsd:complexType>
      <xsd:group ref="g_labels" minOccurs="0"/>
      <xsd:attributeGroup ref="ag_value"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="allowedValues">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="value" minOccurs="0" maxOccurs="unbounded"/>
         <xsd:element ref="range" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="value">
   <xsd:complexType>
      <xsd:group ref="g_labels" minOccurs="0"/>
      <xsd:attributeGroup ref="ag_value"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="range">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="minValue">
            <xsd:complexType>
               <xsd:group ref="g_labels" minOccurs="0"/>
```

```
               <xsd:attributeGroup ref="ag_value"/>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="maxValue">
            <xsd:complexType>
               <xsd:group ref="g_labels" minOccurs="0"/>
               <xsd:attributeGroup ref="ag_value"/>
            </xsd:complexType>
         </xsd:element>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="unit">
   <xsd:complexType>
      <xsd:group ref="g_labels"/>
      <xsd:attribute name="multiplier" type="xsd:string" use="required"/>
      <xsd:attribute name="unitURI" type="xsd:anyURI" use="optional"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="property">
   <xsd:complexType>
      <xsd:attribute name="name" type="xsd:string" use="required"/>
      <xsd:attribute name="value" type="xsd:string" use="required"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="parameterGroupList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="parameterGroup" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="parameterGroup">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:group ref="g_labels"/>
         <xsd:element ref="parameterGroup" minOccurs="0" maxOccurs="unbounded"/>
         <xsd:element ref="parameterRef" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
      <xsd:attribute name="kindOfAccess" type="xsd:string" use="optional"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="parameterRef">
   <xsd:complexType>
      <xsd:attribute name="uniqueIDRef" type="xsd:IDREF" use="required"/>
   </xsd:complexType>
</xsd:element>
<!--################ complex types -->
<xsd:element name="array">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="subrange" maxOccurs="unbounded"/>
         <xsd:choice>
            <xsd:group ref="g_simple"/>
            <xsd:element ref="dataTypeIDRef"/>
         </xsd:choice>
      </xsd:sequence>
      <xsd:attribute name="name" type="xsd:string" use="required"/>
      <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
      <xsd:attribute name="description" type="xsd:string" use="optional"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="subrange">
   <xsd:complexType>
      <xsd:attribute name="lowerLimit" type="xsd:long" use="required"/>
      <xsd:attribute name="upperLimit" type="xsd:long" use="required"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="struct">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="count" minOccurs="0"/>
         <xsd:element ref="varDeclaration" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="name" type="xsd:string" use="required"/>
```

```
      <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
      <xsd:attribute name="description" type="xsd:string" use="optional"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="enum">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="enumValue" maxOccurs="unbounded"/>
         <xsd:group ref="g_simple" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="name" type="xsd:string" use="required"/>
      <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
      <xsd:attribute name="size" type="xsd:string" use="optional"/>
      <xsd:attribute name="description" type="xsd:string" use="optional"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="enumValue">
   <xsd:complexType>
      <xsd:group ref="g_labels"/>
      <xsd:attribute name="value" type="xsd:string" use="optional"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="derived">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="count" minOccurs="0"/>
         <xsd:choice>
            <xsd:group ref="g_simple"/>
            <xsd:element ref="dataTypeIDRef"/>
         </xsd:choice>
      </xsd:sequence>
      <xsd:attribute name="name" type="xsd:string" use="required"/>
      <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
      <xsd:attribute name="description" type="xsd:string" use="optional"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="count">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:group ref="g_labels" minOccurs="0"/>
         <xsd:element ref="defaultValue"/>
         <xsd:element ref="allowedValues" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
      <xsd:attribute name="access" default="read">
         <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
               <xsd:enumeration value="read"/>
               <xsd:enumeration value="write"/>
               <xsd:enumeration value="readWrite"/>
               <xsd:enumeration value="noAccess"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:attribute>
   </xsd:complexType>
</xsd:element>
<!--################ group-->
<xsd:group name="g_complex">
   <xsd:choice>
      <xsd:element ref="array"/>
      <xsd:element ref="struct"/>
      <xsd:element ref="enum"/>
      <xsd:element ref="derived"/>
   </xsd:choice>
</xsd:group>
<!--################ attribute groups-->
<xsd:attributeGroup name="ag_parameter">
   <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
   <xsd:attribute name="access" default="read">
      <xsd:simpleType>
         <xsd:restriction base="xsd:NMTOKEN">
            <xsd:enumeration value="read"/>
            <xsd:enumeration value="write"/>
            <xsd:enumeration value="readWrite"/>
            <xsd:enumeration value="noAccess"/>
         </xsd:restriction>
```

```
         </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="support" use="optional">
         <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
               <xsd:enumeration value="mandatory"/>
               <xsd:enumeration value="optional"/>
               <xsd:enumeration value="conditional"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="persistence" type="xsd:boolean" default="false"/>
      <xsd:attribute name="offset" type="xsd:string" use="optional"/>
      <xsd:attribute name="multiplier" type="xsd:string" use="optional"/>
   </xsd:attributeGroup>
   <xsd:attributeGroup name="ag_value">
      <xsd:attribute name="value" type="xsd:string" use="required"/>
      <xsd:attribute name="offset" type="xsd:string" use="optional"/>
      <xsd:attribute name="multiplier" type="xsd:string" use="optional"/>
   </xsd:attributeGroup>
</xsd:schema>
```

F.5 Communication network profile template description

F.5.1 ProfileBody

For the communication network profile the ProfileBody contains the ApplicationLayers, the TransportLayers and the NetworkManagement elements shown in Figure 31.

The ProfileBody element contains the attributes given in Table F.39.

Table F.39 — Attributes of element ProfileBody

Attribute	Data type	Use	Description
formatName	xsd:string	fixed	Format identifier
formatVersion	xsd:string	fixed	Format version identifier
fileName	xsd:string	required	Name of the file with extension without path
fileCreator	xsd:string	required	Person creating the file
fileCreationDate	xsd:date	required	Date of file creation
fileCreationTime	xsd:time	optional	Time of file creation
fileModifiedBy	xsd:string	optional	Person modifying the file
fileModificationDate	xsd.date	optional	Date of last file change
fileModificationTime	xsd:time	optional	Time of last file change
fileVersion	xsd:string	required	Vendor specific version of the file
supportedLanguages	xsd:NMTOKENS	optional	List of supported languages

F.5.2 ApplicationLayers

F.5.2.1 General

Figure 31 shows the structure of the EtherCAT ApplicationLayers class CANopenObjectList.

F.5.2.1.1 General

Figure 31 shows the structure of the CANopenObjectList element. The element contains one to many CANopenObject elements.

NOTE EtherCAT has adopted CANopen (EN 50325-4) terminology and object structure (CANopen over EtherCAT). Therefore CANopen terms are used here.

F.5.2.1.2 CANopenObject

F.5.2.1.2.1 General

Figure 31 shows the structure of the CANopenObject element. The element contains zero to many CANopenSubObject elements. The CANopenObject element and the CANopenSubObject element map the functional part of the EtherCAT device profile to the CANopen over EtherCAT communication network profile.

The CANopenObject element contains the attributes given in Table F.40.

Table F.40 — Attributes of element CANopenObject

Attribute	Data type	Use	Description
index	xsd:hexBinary	required	Index of the object (four hex digits)
name	xsd:string	required	Name of the object
objectType	xsd:unsignedByte	required	CANopen object type
dataType	xsd:hexBinary	optional	CANopen data type (two hex digits)
lowLimit	xsd:string	optional	Low limit of the parameter value
highLimit	xsd:string	optional	High limit of the parameter value
accessType	xsd:string	optional	Access type of the object; valid values: — ro - read only access — wo - write only access — rw - both read and write access — rwr - both read and write access; prefered read access — rww - both read and write access; prefered write access — const - read only access; the value is not changing
defaultValue	xsd:string	optional	Default value of the object
pdoMapping	xsd:boolean	optional	PDO mapping of the object; valid values: — true - mapped — false - not mapped
objFlags	xsd:hexBinary	optional	Controls the behaviour of tools (four hex digits)
uniqueIDRef	xsd:IDREF	optional	Unique ID of an appropriate element in the application process part referenced from this object. If the attribute is given the attributes dataType, lowLimit, highLimit, accessType, and defaultValue are given by the referenced element from the application process part.
subNumber	xsd:unsignedByte	optional	Number of sub-objects of the object

F.5.2.1.2.2 CANopenSubObject

The CANopenSubObject element has an empty content.

The CANopenSubObject element contains the attributes given in Table F.41.

Table F.41 — Attributes of element CANopenSubObject

Attribute	Data type	Use	Description
subIndex	xsd:hexBinary	required	Subindex of the object (two hex digits)
name	xsd:string	required	Name of the object
objectType	xsd:unsignedByte	required	CANopen object type
dataType	xsd:hexBinary	optional	CANopen data type (two hex digits)
lowLimit	xsd:string	optional	Low limit of the parameter value
highLimit	xsd:string	optional	High limit of the parameter value
accessType	xsd:string	optional	Access type of the object; valid values: — ro – read only access — wo – write only access — rw – both read and write access — rwr – both read and write access; prefered read access — rww – both read and write access; prefered write access — const – read only access; the value is not changing
defaultValue	xsd:string	optional	Default value of the object
pdoMapping	xsd:boolean	optional	PDO mapping of the object; valid values: — true – mapped — false – not mapped
objFlags	xsd:hexBinary	optional	Controls the behaviour of tools (four hex digits)
uniqueIDRef	xsd:IDREF	optional	Unique ID of an appropriate element in the application process part referenced from this object. If the attribute is given the attributes dataType, lowLimit, highLimit, accessType, and defaultValue are given by the referenced element from the application process part.

F.5.2.2 Identity

Since different communication profiles may require different identity information, an optional local identity element may be used within an ApplicationLayers element. This identity element may contain a subset of the sub-elements of the DeviceIdentity element described in F.4.2. All sub-element descriptions given there also apply for the sub-elements of this identity element.

F.5.2.3 Element dummyUsage

F.5.2.3.1 General

Figure 31 shows the structure of the dummyUsage element. The element contains one to many dummy elements.

F.5.2.3.2 Element dummy

The dummy element has no content. The element is used to enable and disable certain dummy entries for the dummy mapping.

The dummy element contains the attributes given in Table F.42.

Table F.42 — Attributes of element dummy

Attribute	Data type	Use	Description
entry	xsd:string	required	The string is constructed using the name of the dummy object, followed by the equal sign and then the value of either 0 for disabled mapping or 1 for enabled mapping. The allowed values are as follows: — Dummy0001=0 — Dummy0002=0 — Dummy0003=0 — Dummy0004=0 — Dummy0005=0 — Dummy0006=0 — Dummy0007=0 — Dummy0001=1 — Dummy0002=1 — Dummy0003=1 — Dummy0004=1 — Dummy0005=1 — Dummy0006=1 — Dummy0007=1

F.5.2.4 Element dynamicChannels

F.5.2.4.1 General

Figure 31 shows the structure of the dynamicChannels element. The element contains one to many dynamicChannel elements.

F.5.2.4.2 Element dynamicChannel

The dynamicChannel element contains an element describing a simple data type out of the group g_simple. The element is used to mark available channels that can be used to create a link between the data to be communicated in the EtherCAT network and the application programme running on the device.

The dynamicChannel element contains the attributes given in Table F.43.

Table F.43 — Attributes of element dynamicChannel

Attribute	Data type	Use	Description
accessType	xsd:NMTOKEN	required	Access type of the object; valid values: — readOnly – read only access — writeOnly – write only access — readWriteWrite – both read and write access; preferred write access
startIndex	xsd:hexBinary	required	Start index of the object
endIndex	xsd:hexBinary	required	End index of the object
maxNumber	xsd:unsignedInt	required	Maximum number of links to application programme
addressOffset	xsd:hexBinary	required	Address offset within application programme memory
bitAlignment	xsd:unsignedByte	optional	Bit alingment of data within the object counted from the least significant bit.
manufacturerSpecific	xsd:string	optional	Available for manufacturer specific use

F.5.3 TransportLayers

The TransportLayers element has no content.

F.5.4 NetworkManagement

F.5.4.1 General

Figure 31 shows the structure of the EtherCAT NetworkManagement class.

F.5.4.2 Element generalFeatures

The generalFeatures element has an empty content.

The generalFeatures element contains the attributes given in Table F.44.

Table F.44 — Attributes of element generalFeatures

Attribute	Data type	Use	Description
dynamicChannels	xsd:unsignedByte	required	States the support of dynamic network variable generation. The value of 0 means that dynamic channels are not supported. Any value different from 0 means that the corresponding number of channels is supported
granularity	xsd:unsignedByte	required	States the granularity supported by the device type. The value shall be different from 0 and defines the bit size of the smallest unit of data that is supported by the device
FMMUChannels	xsd:byte	required	Number of supported FMMU channels
FMMUBitOperationsNotSupported	xsd:boolean	required	FMMU supports bit operation — false – bit operation is supported — true – bit operation is not supported
SyncManagerChannels	xsd:byte	required	Number of supported Sync manager channels
RAMSize	xsd:byte	required	Size of RAM accessible from application in KBytes (1 to 64)

F.5.4.3 Element deviceCommissioning

The deviceCommissioning element has an empty content.

The deviceCommissioning element contains the attributes given in Table F.45.

Table F.45 — Attributes of element deviceCommissioning

Attribute	Data type	Use	Description
nodeID	xsd:unsignedByte	required	The unique ID of the device.
nodeName	xsd:string	required	The name of the device.
networkNumber	xsd:unsignedLong	required	The unique number of the network segment where the device is connected.
networkName	xsd:string	required	The name of the network segment where the device is connected.

F.5.5 EtherCAT communication network profile template schema

The XML schema ProfileBody_CommunicationNetwork_EtherCAT.xsd includes the schema ISO15745ProfileContainer.xsd in F.4.6.1 and the schema CommonElements.xsd in F.4.6.2.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <xsd:include schemaLocation="ISO15745ProfileContainer.xsd"/>
   <xsd:include schemaLocation="CommonElements.xsd"/>
   <!-- profile body -->
   <xsd:complexType name="ProfileBody_CommunicationNetwork_EtherCAT">
      <xsd:complexContent>
         <xsd:extension base="ProfileBody_DataType">
            <xsd:choice>
               <xsd:sequence>
                  <xsd:element name="ApplicationLayers">
                     <xsd:complexType>
                        <xsd:sequence>
                           <xsd:element ref="CANopenObjectList"/>
                           <xsd:element name="identity" minOccurs="0">
```

```
                                    <xsd:complexType>
                                       <xsd:sequence>
                                          <xsd:element ref="vendorID" minOccurs="0"/>
                                          <xsd:element ref="deviceFamily" minOccurs="0"/>
                                          <xsd:element ref="productID" minOccurs="0"/>
                                          <xsd:element ref="version" minOccurs="0" maxOccurs="unbounded"/>
                                          <xsd:element ref="buildDate" minOccurs="0"/>
                                          <xsd:element ref="specificationRevision" minOccurs="0"/>
                                       </xsd:sequence>
                                    </xsd:complexType>
                                 </xsd:element>
                                 <xsd:element name="dummyUsage" minOccurs="0">
                                    <xsd:complexType>
                                       <xsd:sequence>
                                          <xsd:element name="dummy" maxOccurs="unbounded">
                                             <xsd:complexType>
                                                <xsd:attribute name="entry" use="required">
                                                   <xsd:simpleType>
                                                      <xsd:restriction base="xsd:string">
                                                         <xsd:enumeration value="Dummy0001=0"/>
                                                         <xsd:enumeration value="Dummy0002=0"/>
                                                         <xsd:enumeration value="Dummy0003=0"/>
                                                         <xsd:enumeration value="Dummy0004=0"/>
                                                         <xsd:enumeration value="Dummy0005=0"/>
                                                         <xsd:enumeration value="Dummy0006=0"/>
                                                         <xsd:enumeration value="Dummy0007=0"/>
                                                         <xsd:enumeration value="Dummy0001=1"/>
                                                         <xsd:enumeration value="Dummy0002=1"/>
                                                         <xsd:enumeration value="Dummy0003=1"/>
                                                         <xsd:enumeration value="Dummy0004=1"/>
                                                         <xsd:enumeration value="Dummy0005=1"/>
                                                         <xsd:enumeration value="Dummy0006=1"/>
                                                         <xsd:enumeration value="Dummy0007=1"/>
                                                      </xsd:restriction>
                                                   </xsd:simpleType>
                                                </xsd:attribute>
                                             </xsd:complexType>
                                          </xsd:element>
                                       </xsd:sequence>
                                    </xsd:complexType>
                                 </xsd:element>
                                 <xsd:element name="dynamicChannels" minOccurs="0">
                                    <xsd:complexType>
                                       <xsd:sequence>
                                          <xsd:element name="dynamicChannel" maxOccurs="unbounded">
                                             <xsd:complexType>
                                                <xsd:sequence>
                                                   <xsd:group ref="g_simple"/>
                                                </xsd:sequence>
                                                <xsd:attribute name="accessType" use="required">
                                                   <xsd:simpleType>
                                                      <xsd:restriction base="xsd:NMTOKEN">
                                                         <xsd:enumeration value="readOnly"/>
                                                         <xsd:enumeration value="writeOnly"/>
                                                         <xsd:enumeration value="readWriteWrite"/>
                                                      </xsd:restriction>
                                                   </xsd:simpleType>
                                                </xsd:attribute>
                                                <xsd:attribute name="startIndex" type="xsd:hexBinary"
use="required"/>
                                                <xsd:attribute name="endIndex" type="xsd:hexBinary"
use="required"/>
                                                <xsd:attribute name="maxNumber" type="xsd:unsignedInt"
use="required"/>
                                                <xsd:attribute name="addressOffset" type="xsd:hexBinary"
use="required"/>
                                                <xsd:attribute name="bitAlignment"
type="xsd:unsignedByte" use="optional"/>
                                                <xsd:attribute name="manufacturerSpecific"
type="xsd:string" use="optional"/>
                                             </xsd:complexType>
                                          </xsd:element>
                                       </xsd:sequence>
                                    </xsd:complexType>
                                 </xsd:element>
                              </xsd:sequence>
```

```
                  </xsd:complexType>
               </xsd:element>
               <xsd:element name="TransportLayers">
                  <xsd:complexType/>
               </xsd:element>
               <xsd:element name="NetworkManagement" minOccurs="0">
                  <xsd:complexType>
                     <xsd:sequence>
                        <xsd:element name="generalFeatures">
                           <xsd:complexType>
                              <xsd:attribute name="dynamicChannels" type="xsd:unsignedByte"
use="required"/>
                              <xsd:attribute name="granularity" type="xsd:unsignedByte"
use="required"/>
                              <xsd:attribute name="FMMUChannels" type="xsd:byte" use="required"/>
                              <xsd:attribute name="FMMUBitOperationNotSupported"
type="xsd:boolean" use="required"/>
                              <xsd:attribute name="SyncManagerChannels" type="xsd:byte"
use="required"/>
                              <xsd:attribute name="RAMSize" type="xsd:byte" use="required"/>
                           </xsd:complexType>
                        </xsd:element>
                        <xsd:element name="deviceCommissioning" minOccurs="0">
                           <xsd:complexType>
                              <xsd:attribute name="nodeID" type="xsd:unsignedByte"
use="required"/>
                              <xsd:attribute name="nodeName" type="xsd:string" use="required"/>
                              <xsd:attribute name="networkNumber" type="xsd:unsignedLong"
use="required"/>
                              <xsd:attribute name="networkName" type="xsd:string"
use="required"/>
                           </xsd:complexType>
                        </xsd:element>
                     </xsd:sequence>
                  </xsd:complexType>
               </xsd:element>
            </xsd:sequence>
            <xsd:element name="ExternalProfileHandle" type="ProfileHandle_DataType"/>
         </xsd:choice>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
<!-- CANopen object dictionary-->
<xsd:element name="CANopenObjectList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="CANopenObject" maxOccurs="65535">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="CANopenSubObject" minOccurs="0" maxOccurs="255">
                     <xsd:complexType>
                        <xsd:attribute name="subIndex" type="xsd:unsignedByte" use="required"/>
                        <xsd:attribute name="name" type="xsd:string" use="required"/>
                        <xsd:attribute name="objectType" type="xsd:unsignedByte" use="required"/>
                        <xsd:attribute name="dataType" type="xsd:hexBinary" use="optional"/>
                        <xsd:attribute name="lowLimit" type="xsd:string" use="optional"/>
                        <xsd:attribute name="highLimit" type="xsd:string" use="optional"/>
                        <xsd:attribute name="accessType" use="optional">
                           <xsd:simpleType>
                              <xsd:restriction base="xsd:string">
                                 <xsd:enumeration value="ro"/>
                                 <xsd:enumeration value="wo"/>
                                 <xsd:enumeration value="rw"/>
                                 <xsd:enumeration value="rwr"/>
                                 <xsd:enumeration value="rww"/>
                                 <xsd:enumeration value="const"/>
                              </xsd:restriction>
                           </xsd:simpleType>
                        </xsd:attribute>
                        <xsd:attribute name="defaultValue" type="xsd:string" use="optional"/>
                        <xsd:attribute name="actualValue" type="xsd:string" use="optional"/>
                        <xsd:attribute name="denotation" type="xsd:string" use="optional"/>
                        <xsd:attribute name="PDOmapping" type="xsd:boolean" use="optional"/>
                        <xsd:attribute name="objFlags" type="xsd:unsignedInt" use="optional"/>
                        <xsd:attribute name="uniqueIDRef" type="xsd:IDREF" use="optional"/>
                     </xsd:complexType>
```

```
                  </xsd:element>
               </xsd:sequence>
               <xsd:attribute name="index" type="xsd:hexBinary" use="required"/>
               <xsd:attribute name="name" type="xsd:string" use="required"/>
               <xsd:attribute name="objectType" type="xsd:unsignedByte" use="required"/>
               <xsd:attribute name="dataType" type="xsd:hexBinary" use="optional"/>
               <xsd:attribute name="lowLimit" type="xsd:string" use="optional"/>
               <xsd:attribute name="highLimit" type="xsd:string" use="optional"/>
               <xsd:attribute name="accessType" use="optional">
                  <xsd:simpleType>
                     <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="ro"/>
                        <xsd:enumeration value="wo"/>
                        <xsd:enumeration value="rw"/>
                        <xsd:enumeration value="rwr"/>
                        <xsd:enumeration value="rww"/>
                        <xsd:enumeration value="const"/>
                     </xsd:restriction>
                  </xsd:simpleType>
               </xsd:attribute>
               <xsd:attribute name="defaultValue" type="xsd:string" use="optional"/>
               <xsd:attribute name="actualValue" type="xsd:string" use="optional"/>
               <xsd:attribute name="denotation" type="xsd:string" use="optional"/>
               <xsd:attribute name="PDOmapping" type="xsd:boolean" use="optional"/>
               <xsd:attribute name="objFlags" type="xsd:hexBinary" use="optional"/>
               <xsd:attribute name="uniqueIDRef" type="xsd:IDREF" use="optional"/>
               <xsd:attribute name="subNumber" type="xsd:unsignedByte" use="optional"/>
            </xsd:complexType>
         </xsd:element>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
</xsd:schema>
```

Annex G
(normative)

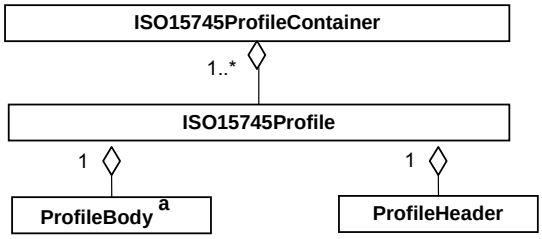
ETHERNET Powerlink profile templates

G.1 Overview

ETHERNET Powerlink is an Ethernet based communication system specified in IEC/PAS 62408.

ETHERNET Powerlink uses the concept of the multi-profile container specified in Amendment 1 to ISO 15745-1:2003 for XML profile files. Therefore, ETHERNET Powerlink profile templates are based on the alternate ISO15745ProfileContainer master profile template specified in Amendment 1 to ISO 15745-1:2003.

Figure G.1 shows the structure of an ETHERNET Powerlink XML profile.



[a] Two types of ProfileBody are used: ProfileBody_Device_EPL or ProfileBody_CommunicationNetwork_EPL

Figure G.1 — ETHERNET Powerlink profile template

The ProfileTechnology name is EPL.

G.2 General rules

G.2.1 Using unique IDs

An element can have the attribute uniqueID of type xsd:ID. The unique identifier therefore is forced to be unique in the whole XML file. An element that references the unique identifier contains a named attribute of type xsd:IDREF.

Unique identifiers may be generated in two ways. One possibility is to build a string out of the element name and an up-counting number. A second way may be the concatenation of strings of parent elements. Both methods guarantee the uniqueness of the string.

G.2.2 Language support

G.2.2.1 General

Device profiles complying with the XML schema described in this annex need a support of different languages, since tools are then able to use names out of the XML file in order to display them in their user interface. Communication parameters for example may be presented in the user interface of a tool.

The language support is implemented via the group g_labels. Each name of an element, which would possibly be displayed and is therefore language dependent, is provided inside the schema as a g_labels element. Optionally, a URI may be added as an attribute to the label element.

EXAMPLE

For a given parameter name:

— German: Baudrate

— English: Baud rate

— French: Vitesse de transmission

G.2.2.2 Element g_labels

The group g_labels supports the introduction of a label (name) and a description in the context of the parent element (see Figure G.2).

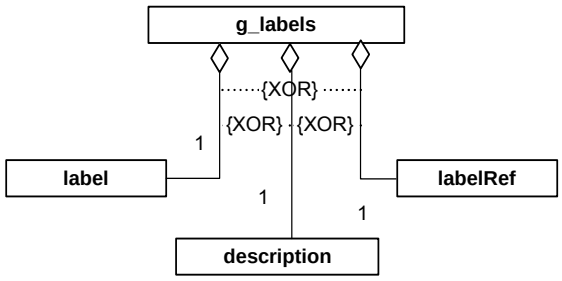


Figure G.2 — Group g_labels

Every element, which needs a name or a description, shall select one and only one of the three elements to perform this task: the label, the description and the labelRef element.

1) The label element allows storage of the identifying name and the descriptive text inside the XML file itself. The label element has the attributes given in Table G.1.

Table G.1 — Attributes of element label

Attribute	Data type	Use	Description
lang	xsd:language	required	Language used for the name or the description
URI	xsd:anyURI	optional	Optional link to further descriptive information

The element may appear n times, once for each language. For identifying the language, the lang attribute is used.

2) The description element allows storage of textual descriptions inside the XML file itself. The element may appear several times, once for each language. The description element has the same attributes as the label element.

3) The labelRef element allows referencing of descriptive texts inside an external text source file.

   The labelRef element provides a pointer (via its attributes dictID and textID) to a text entry in a separate text source file. These text source files are referenced in the dictionary sub-elements of the DeviceFunction element. Text source files may be any files containing character sequences and other information, for example figures.

   The labelRef element also may appear n times, to allow references to several dictionary entries, which contain links to files in different languages. The respective language is defined in the lang attribute of the dictionary element.

   The labelRef element contains the attributes given in Table G.2.

Table G.2 — Attributes of element labelRef

Attribute	Data type	Use	Description
dictID	xsd:IDREF	required	References a single dictionary element inside the dictionaryList element; the dictionary element contains a link to the external text source file
textID	xsd:string	optional	References a character sequence inside the external text source file by pattern matching

G.2.2.3 Language identifier

For the multi-language support each label gets an attribute with the content of the language code. The language code corresponds to the content of the label element.

In order to verify which languages are supported in the XML file the attribute supportedLanguages in the ProfileBody element lists the supported languages.

G.2.2.4 Attribute lang

The language identifier lang consists of a combination of a language code (as defined in ISO 639-1) plus an optional dash character plus an optional country code (as defined in ISO 3166-1). The attribute lang is an attribute of the label element.

Some of the values for lang are given in Table G.3.

Table G.3 — Values of attribute lang

Language	value of lang
English (United States)	en-us
German (Standard)	de
French (Standard)	fr
Spanish (Standard)	es
Italian (Standard)	It
Portuguese (Brazil)	pt-br

G.2.2.5 SupportedLanguages attribute

The supportedLanguages attribute identifies supported languages and consists of a list of language codes plus optional country codes.

EXAMPLE

supportedLanguages=”en-us de fr es”

G.2.2.6 URIs

A general mechanism allows describing a URI in the context of a label element. The URI is implemented via an optional attribute URI.

EXAMPLE

This is used in the context of a vendor label, parameter label, or services label.

G.3 ProfileHeader

To facilitate the identification of an ETHERNET Powerlink file, the ETHERNET Powerlink file header shall comply with the model shown in Figure G.3, which is directly inherited from ISO 15745-1.

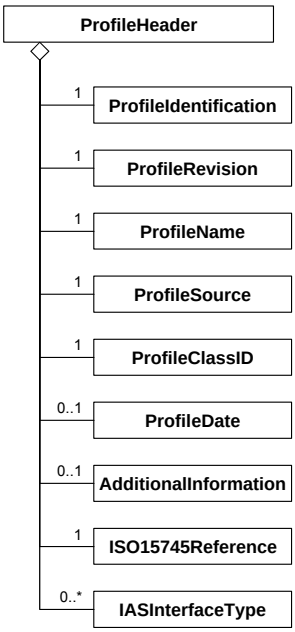


Figure G.3 — Profile header class diagram

The ProfileHeader element is composed of the following elements:

— the ProfileIdentification element identifies the current profile,

— the ProfileRevision element identifies the current profile revision,

— the ProfileName element contains a descriptive English name of the current profile. In case that more than one ProfileBody element are present within a device profile, it is suggested that the value of the

ProfileName element should be the concatenation of the values of the productName elements inside the respective DeviceIdentity elements,

— the ProfileSource element identifies the validator of the current profile,

— the ProfileClassID element identifies the class of the current profile according to ISO 15745-1,

— the ISO15745Reference element states the ISO 15745 part, edition and technology, to which this description conforms.

G.4 Device profile template description

G.4.1 ProfileBody_Device_EPL

This part defines the ETHERNET Powerlink device profile.

The ProfileBody_Device_EPL contains the DeviceIdentity, the DeviceManager, the DeviceFunction and the ApplicationProcess elements shown in Figure 32.

The ProfileBody element contains the description of

— a single device (for example a proximity sensor or an electromechanical limit switch), or of a more complex one (for example a circuit breaker up to 2500 parameters, more than 100 functions), or

— a part of a device also called "module" in the PLC world (for example I/O controllers or electrical protection units).

The ProfileBody element contains the attributes given in Table G.4.

Table G.4 — Attributes of element ProfileBody

Attribute	Data type	Use	Description
formatName	xsd:string	fixed	Format identifier
formatVersion	xsd:string	fixed	Format version identifier
fileName	xsd:string	required	Name of the file with extension without path
fileCreator	xsd:string	required	Person creating the file
fileCreationDate	xsd:date	required	Date of file creation
fileCreationTime	xsd:time	optional	Time of file creation
fileModifiedBy	xsd:string	optional	Person modifying the file
fileModificationDate	xsd.date	optional	Date of last file change
fileModificationTime	xsd:time	optional	Time of last file change
fileVersion	xsd:string	required	Vendor specific version of the file
supportedLanguages	xsd:NMTOKENS	optional	List of supported languages

G.4.2 DeviceIdentity

G.4.2.1 General

The DeviceIdentity class (see Figure 33) contains elements, which are independent of the network and of the process. It describes the identity of a single device or of a group of devices.

Table G.5 specifies attribute readOnly, which is attached to the vendorName, vendorID, vendorText, deviceFamily, productFamily, productName, productID, productText, orderNumber, version, specificationRevision, and instanceName elements.

Table G.5 — Attribute of element vendorName

Attribute	Data type	Use	Description
readOnly	xsd:boolean	default	Indicates whether the value is read-only for a user: false, true (default)

G.4.2.2 Element vendorName

The vendorName element identifies the name or the brand name of the vendor of the device.

G.4.2.3 Element vendorID

The vendorID element identifies the vendor. This information has to be filled in when the product described is recognized and validated by a consortium.

NOTE Consortia specific product families and vendor identifiers are linked.

G.4.2.4 Element vendorText

The vendorText element allows the vendor to provide additional company information, like address or hotline number. The g_labels group offers the possibility to include a vendor URI inside the vendorText element.

G.4.2.5 Element deviceFamily

The deviceFamily element states the family of the device.

EXAMPLE

Examples for device families are:

— Variable speed drive

— Circuit breaker

— Pressure sensor

G.4.2.6 Element productFamily

The productFamily element states a vendor specific affiliation of the device type to a certain set of devices inside a family. The list of valid productFamily values is system, tool or consortia specific.

NOTE Consortia specific product families and vendor identifiers are linked.

G.4.2.7 Element productName

The productName element states a vendor specific designation or name of the device type.

G.4.2.8 Element productID

The productID element states a vendor specific unique identification for the device type described.

G.4.2.9 Element productText

The productText element allows the vendor to provide a short textual description of the device type.

G.4.2.10 Element orderNumber

The orderNumber element is used to store the single order number of a given product or the set of different order numbers of the products of a product family, depending upon whether the device profile describes a product or a product family.

G.4.2.11 Element version

The version element is used to store different types of version information. Multiple version elements are possible.

The version element has the attributes given in Table G.6.

Table G.6 — Attributes of element version

Attribute	Data type	Use	Description
versionType	xsd:NMTOKEN	required	Type of version: — SW – Software — FW – Firmware — HW – Hardware
readOnly	xsd:boolean	default	Indicates whether the value is read-only for a user: false, true (default)

G.4.2.12 Element buildDate

The buildDate element specifies the build date of the software unit.

G.4.2.13 Element specificationRevision

The specificationRevision element contains the revision of the specification, to which this device conforms.

G.4.2.14 Element instanceName

This element contains the instance name of the device.

G.4.3 DeviceManager

G.4.3.1 General

The DeviceManager element defines the list of indicators provided by the device type, if any.

G.4.3.2 LEDList

G.4.3.2.1 General

The LEDList elements shown in Figure G.4 specify the number and type of indicators, which are provided by a device type.

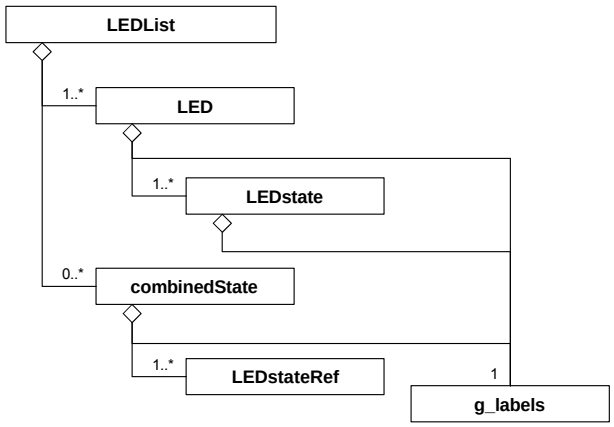


Figure G.4 — LEDList

G.4.3.2.2 LED

The LED element describes the features of a single LED of the device type. A detailed feature description may be provided through the g_labels group.

Further properties of the LED are represented as attributes of the LED element given in Table G.7.

Table G.7 — Attributes of element LED

Attribute	Data type	Use	Description
LEDcolors	xsd:string	required	Colours of the LED; valid values are monocolor and bicolor
LEDtype	xsd:string	optional	Rough classification of the supervised item or functionality; valid values are IO, device and communication

In addition to the descriptive parts introduced above, the LED element contains one to many LEDstate elements, which define the device states signalled by the LED and the visual behaviour used for signalling the states.

The visual behaviour used for signalling the state is encoded as attribute values of the LEDstate element, as given in Table G.8. Additionally a unique ID is allocated for the LED state.

Table G.8 — Attributes of element LEDstate

Attribute	Data type	Use	Description
uniqueID	xsd:ID	required	Unique ID for the LED state; may be referenced from an LEDstateRef element
state	xsd:string	required	State of the LED; possible attribute values: On, off, flashing
LEDcolor	xsd:string	required	Colour of the LED state; valid values: green, amber, red
flashingPeriod	xsd:unsignedInt	optional	If state=flashing: flashing period of the LED in milliseconds
impulsWidth	xsd:unsignedByte	default	Width of the flashing impulse given in percent (%) of the flashing period; if the attribute impulsWidth is missing, the default value is 50 (%)
numberOfImpulses	xsd:unsignedByte	default	Number of impulses in case that more than one flashing impulse is inside one flashing period; if the attribute is present, the attribute impulsWidth shall be present, too; if the attribute numberOfImpulses is missing, the default value is 1

G.4.3.2.3 Element combinedState

The combinedState element allows the indication of device states, which are signalled by more than one LED.

The description of the combined state is provided through the g_labels group.

The LED states participating in the signalling of the combined state are referenced by means of at least two LEDstateRef sub-elements of the combinedState element.

The reference to a LEDstate element is encoded as the attribute value of the single attribute of the LEDstateRef element (see Table G.9).

Table G.9 — Attributes of element LEDstateRef

Attribute	Data type	Use	Description
stateIDRef	xsd:IDREF	required	Unique ID of the referenced LEDstate element

G.4.4 DeviceFunction

G.4.4.1 General

The DeviceFunction element shown in Figure 35 defines the catalogue view of the device, presented as a set of capabilities listing both device characteristics and compliances with various standards.

G.4.4.2 Element capabilities

G.4.4.2.1 General

The mandatory capabilities element describes all functionalities, their characteristics, and the important parameters of the device, that need to be known by tools which use the device profile to select products with the same or similar properties.

The capabilities element describes device features in a purely textual form. It contains a sequence of one to many characteristicsList elements and an optional standardComplianceList element.

G.4.4.2.2 Element characteristicsList

G.4.4.2.2.1 General

The characteristicsList element is a collection of characteristics. The element shall contain at least one characteristic sub-element. The characteristics inside a list may be associated with a category, which can be expressed as textual content of the g_labels sub-element of the optional category sub-element of the characteristicsList element.

G.4.4.2.2.2 Element characteristic

The characteristic element describes a single characteristic of a device. It contains a mandatory characteristicName element and one to many characteristicContent elements.

G.4.4.2.2.3 Element characteristicName

The mandatory characteristicName element denotes a major technical characteristic of the device. The vocabulary used in the product data sheet is recommended for the names of characteristics.

EXAMPLE

"Maximum operational voltage", "Overload protection", "Electrical durability".

G.4.4.2.2.4 Element characteristicContent

This mandatory element contains a value for the characteristic. Multiple values may be expressed by using multiple characteristicContent elements.

EXAMPLE

An example of a single value for "Maximum operational voltage" is 680V.

G.4.4.2.3 Element standardComplianceList

The standardComplianceList element is a collection of compliantWith elements. The element itself is optional; if it exists, it shall contain at least one compliantWith sub-element.

The compliantWith sub-element has attributes, which state the compliance of the device with an international or company internal standard. The content of type g_labels of this element may contain remarks concerning that standard.

The name or number of the standard is provided through the required name attribute of the compliantWith element. The second, default valued range attribute of the compliantWith element defines the range of applicability of the standard as given in Table G.10.

Table G.10 — Attributes of element compliantWith

Attribute	Data type	Use	Description
name	xsd:string	required	Name or number of the standard
range	xsd:NMTOKEN	default	The two possible enumerated values of the attribute are international (default) or internal

G.4.4.3 Element picturesList

The picturesList element offers the possibility to link pictures to the device profile. It contains one to many picture sub-elements, whose caption is provided via a g_labels sub-element.

Table G.11 defines attributes of the picture sub-element: an optional number of the picture, and the mandatory link to an external resource containing the graphical information.

Table G.11 — Attributes of element picture

Attribute	Data type	Use	Description
URI	xsd:anyURI	required	Link to the external resource
number	xsd:unsignedInt	optional	Number of the picture

G.4.4.4 Element dictionaryList

The optional dictionaryList element offers the possibility to include links to external text resource files to the device profile. It contains one to many dictionary elements, where each one contains one to many file sub-elements. Several files are necessary if different file formats are needed within a dictionary.

A mandatory lang attribute of type xsd:language defines the language used in the files which are linked to the dictionary element (see Table G.12). A mandatory uniqueID attribute of type xsd:ID holds the unique identification of the dictionary entry which is referenced from the attribute dictID of a labelRef element as given in Table G.2.

Table G.12 — Attributes of element dictionary

Attribute	Data type	Use	Description
lang	xsd:language	required	Language used for files belonging to a dictionary entry
uniqueID	xsd:ID	required	Unique ID of the dictionary entry

A file sub-element contains a single mandatory attribute given in Table G.13.

Table G.13 — Attributes of element file

Attribute	Data type	Use	Description
URI	xsd:anyURI	required	Link to the respective file

G.4.5 ApplicationProcess

G.4.5.1 General

The ApplicationProcess element represents the set of services and parameters, which constitute the behaviour, and the interfaces of the device in terms of the application, independent of the device technology and the underlying communication networks and communication protocols.

The sub-elements of the ApplicationProcess element in Figure 36 provide a generic approach for the description of arbitrary, flat or hierarchically structured functions of a device.

Functions are modeled as function types, which are instantiated within the device or - if hierarchical structures are needed - inside function types. The interface variables of these function instances, which may be of simple

or complex data type, are associated with the parameters of the device by building a reference from the parameter to the respective interface variable of the function instance, in flat as well as in hierarchical structures.

The ApplicationProcess element contains up to five lists of items (see Figure 36):

— two lists which define data types (optional) and function types (required),

— one required list which defines the function instances on device level (possibly including connections between instances),

— one required list which defines the device parameters, and

— one optional list which defines parameter groups (combinations of parameters for specific purposes).

G.4.5.2 Element dataTypeList

G.4.5.2.1 General

The optional dataTypeList element is present, if complex data types like arrays or data structures are needed inside variable declarations of the device profile.

If present the dataTypeList element shown in Figure G.5 contains a sequence of one to many elements out of the choice of

— an array element,

— a struct element,

— an enum element, or

— a derived element.

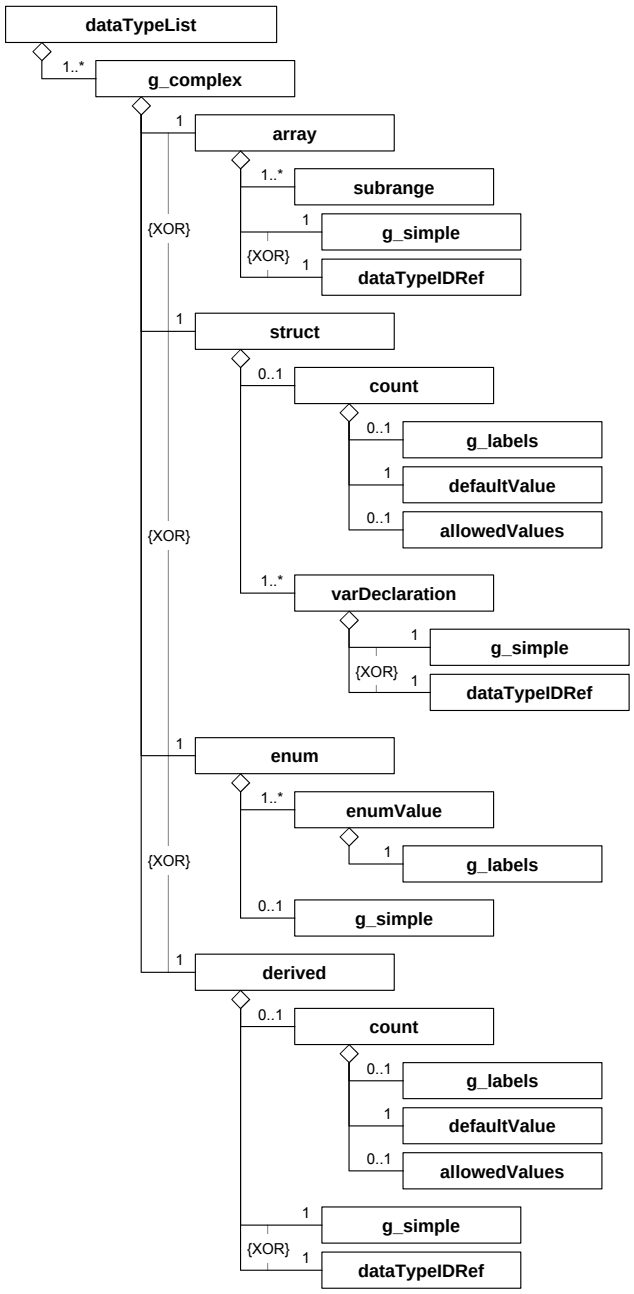


Figure G.5 — dataTypeList

G.4.5.2.2 Common elements

G.4.5.2.2.1 Group g_simple

The group g_simple contains a choice of elements, whose names represent the names of all simple data types allowed in the definition of variables inside a device profile. The simple data types conform to the elementary data types defined in IEC 61131-3; the data types BITSTRING and CHAR (=STRING[1]) are added.

These elements are introduced inside a group to allow their placement directly as a sub-element of the array element (or of the varDeclaration element, see G.4.5.4.3.2).

G.4.5.2.2.2 Element count

The count element defines the number of used units of the base type of the derived type. Multilingual names and/or descriptions for the count element are provided through the group g_labels. See G.2.2.2 for the description of the group g_labels.

The count is described by

— its attributes,

— the mandatory sub-element defaultValue and a possibly empty set of sub-elements g_labels and allowedValues.

The number of units is expressed as the value of the defaultValue attribute of the count element. The allowedValue attribute defines the range of values for the default value.

The sub-elements defaultValue and allowedValues are described in G.4.5.6.2.5 and in G.4.5.6.2.7.

The count element shall contain the attributes given in Table G.14.

Table G.14 — Attributes of element count

Attribute	Data type	Use	Description
uniqueID	xsd:ID	required	Unique ID of the count
access	xsd:NMTOKEN	required	Defines which operations are valid for the count: — read - read access only (default value) — write - write access only — readWrite - both read and write access — noAccess - access denied

G.4.5.2.3 Element array

G.4.5.2.3.1 General

The array element serves to describe an array data type, which may be referenced from an interface variable of a function type, from another array type definition, or from a component variable inside the definition of a structured data type.

The array element contains at least one subrange element and either an element describing a simple data type out of the group g_simple, or an element dataTypeIDRef, which references one of the defined complex data types within the dataTypeList element.

For multi-dimensional arrays, several subrange elements will be present. In this case, the first subrange element in the sequence defines the subrange for the leftmost array index, and the last subrange element in the sequence defines the subrange for the rightmost array index.

The array element contains the attributes given in Table G.15.

Table G.15 — Attributes of element array

Attribute	Data type	Use	Description
name	xsd:string	required	Data type name of the array type
uniqueID	xsd:ID	required	Unique ID of the array type
description	xsd:string	optional	Optional textual description of the array type

G.4.5.2.3.2 Element subrange

The subrange element defines the lower and the upper limit of an array index for one dimension of the array. This element has no sub-elements.

The limit values of type xsd:long are contained in the two attributes of the subrange element given Table G.16.

Table G.16 — Attributes of element subRange

Attribute	Data type	Use	Description
lowerLimit	xsd:long	required	Lower limit of the subrange
upperLimit	xsd:long	required	Upper limit of the subrange

G.4.5.2.4 Element struct

G.4.5.2.4.1 General

The struct element serves to describe a structured data type, which may be referenced from an interface variable of a function type, from an array type definition, or from a component variable inside the definition of another structured data type.

The struct element contains a sequence of one to many varDeclaration elements, which define the components of the structured data type.

The struct element shall contain the attributes given in Table G.17.

Table G.17 — Attributes of element struct

Attribute	Data type	Use	Description
name	xsd:string	required	Data type name of the structured data type
uniqueID	xsd:ID	required	Unique ID of the structured data type
description	xsd:string	optional	Optional textual description of the structured data type

G.4.5.2.4.2 Element varDeclaration

In the context of the definition of a structured data type, the varDeclaration element describes a single component variable (member) of the structure.

In the context of the definition of the interface of a function, the varDeclaration element describes a single interface variable of the function type.

The data type of the component variable or interface variable is either defined by an element describing a simple data type out of the group g_simple, or by an element dataTypeIDRef, which references one of the defined complex data types within the dataTypeList element.

All further properties of the variable are contained in the attributes of the varDeclaration element, as given in Table G.18.

Table G.18 — Attributes of element varDeclaration

Attribute	Data type	Use	Description
name	xsd:string	required	Name of the interface variable or structure component
uniqueID	xsd:ID	required	Unique ID of the interface variable or structure component (see NOTE 1)
size	xsd:string	optional	Number of elements, if the interface variable or structure component is of type anonymous ARRAY, BITSTRING, STRING or WSTRING (see NOTE 2)
initialValue	xsd:string	optional	Initial value of the interface variable or structure component (see NOTE 3)
description	xsd:string	optional	Optional textual description of the interface variable or structure component

NOTE 1 When creating the unique ID for a variable, it is essential that the ID is unique over all created IDs inside the XML source file. To allow equal names for component variables of different data structures and equal names for interface variables of function types, the ID of a variable should generally concatenate the type name of the structured data type or the function type with the variable name, to guarantee uniqueness.

NOTE 2 Anonymous types define the size of an array, bitstring or string directly in the variable declaration, and not through the reference to a named complex data type. In the case of an array, the type of a single array element is given by the data type of the variable. In the case of a bitstring, the single array element is a single bit. In the case of a string, the single array element is a single-byte resp. double-byte character.

NOTE 3 If present, this attribute defines the initial (default) value of the interface variable of the function type. It is overwritten by a given default value of a parameter associated with the interface variable of the function instance.

G.4.5.2.5 Element enum

G.4.5.2.5.1 General

The enum element serves to describe an enumerated data type, which may be referenced from an interface variable of a function type, from an array type definition, or from a component variable inside the definition of a structured data type.

According to Figure G.5, it contains a sequence of one to many enumValue elements, which define the enumeration constants of the enumerated data type. The data type of the enumeration constants is optionally defined by an element describing a simple data type out of the group g_simple.

The enum element contains the attributes given in Table G.19.

Table G.19 — Attributes of element enum

Attribute	Data type	Use	Description
name	xsd:string	required	Type name of the enumerated data type
uniqueID	xsd:ID	required	Unique ID of the enumerated data type
size	xsd:string	optional	Optional number of enumerated values of the enumerated data type
description	xsd:string	optional	Optional textual description of the enumerated data type

G.4.5.2.5.2 Element enumValue

The enumValue element defines the name(s) and optionally a numerical value of a single enumeration constant. The name(s) are specified through the g_labels group, whereas the value is contained in the single value attribute of the enumValue element, as given in Table G.20.

Table G.20 — Attributes of element enumValue

Attribute	Data type	Use	Description
value	xsd:string	optional	Optional attribute: fixed numerical value for the enumeration constant, represented as a string of characters

G.4.5.2.6 Element derived

The derived element serves to derive a new data type from a given base type.

The derived element contains an optional count element and either an element describing a simple data type out of the group g_simple, or an element dataTypeIDRef, which references one of the defined complex data types within the dataTypeList element.

If the count element is missing, the derived type definition just introduces a new type name for the respective base type. If the count element is present, it defines the number of units of the respective base type used to build the derived type (e.g base type BITSTRING, count = 4 defines a derived type of size 4 bit).

The derived element contains the attributes given in Table G.21.

Table G.21 — Attributes of element derived

Attribute	Data type	Use	Description
name	xsd:string	required	Data type name of the derived type
uniqueID	xsd:ID	required	Unique ID of the derived type
description	xsd:string	optional	Optional textual description of the derivedy type

G.4.5.3 Element functionTypeList

If the optional ApplicationProcess element is present in the device profile, it contains a mandatory functionTypeList element shown in Figure G.6.

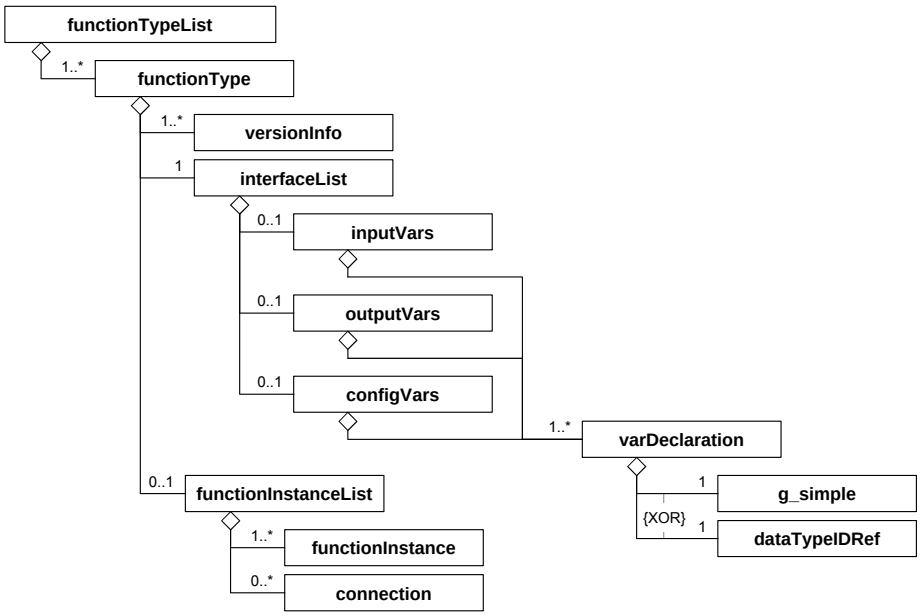


Figure G.6 — functionTypeList class diagramm

The functionTypeList represents a sequence of one to many functionType elements.

Each of the functionType elements represents the type description of a device function, which is referenced from at least one instance of that function type inside a functionInstanceList element. References from more than one instance of the same function type are also possible.

The description of a function type contains all those objects and data which are common for all instances of a given function type.

EXAMPLE 1

Examples are the variable - or function parameter - objects that constitute the interface of the function (type respectively instance).

EXAMPLE 2

Other examples are instances contained within the body of a function in a hierarchically structured functional description. These instances, which are located within a functionInstanceList element inside the function type, reference other function types in the list of function types.

G.4.5.4 Element functionType

G.4.5.4.1 General

The functionType element contains one to many versionInfo elements, a mandatory interfaceList element and an optional functionInstanceList element. The functionInstanceList element is only present within a functionType element, if the function is hierarchically structured.

Additionally, the functionType element shall contain the attributes given in Table G.22.

Table G.22 — Attributes of element functionType

Attribute	Data type	Use	Description
name	xsd:string	required	Type name of the function type
uniqueID	xsd:ID	required	Unique ID of the function type
description	xsd:string	optional	Optional textual description of the function type
package	xsd:string	optional	Optional textual association of the function type with a "package" or similar classification scheme - the usage of this attribute is left to the profile validator

G.4.5.4.2 Element versionInfo

The mandatory versionInfo element within the functionType element provides information on the versioning history of a function type (concerning the definition of the interface).

To keep track of the versioning history, the versionInfo element may be entered multiple times. The multiple entries shall be arranged within the functionType element in the following sequence:

a) the first entry represents the most recent version,

b) the second entry represents the immediately preceding version,

c) the last entry represents the first released version.

This element will be provided once at the creation of the description of the function type. New elements will only be added, if modifications of a function type are introduced, which leads to a modified version of the device profile.

The versionInfo element shall contain the attributes given in Table G.23.

Table G.23 — Attributes of element versionInfo

Attribute	Data type	Use	Description
organization	xsd:string	required	Name of the organization maintaining the function type
version	xsd:string	required	Version identification in the versioning history; suggested format: "xx.yy" (xx,yy = 0..255)
author	xsd:string	required	Name of the person maintaining the function type
date	xsd:date	required	Date of this version
remarks	xsd:string	optional	Descriptive information concerning the specific step in the versioning history

G.4.5.4.3 Element interfaceList

G.4.5.4.3.1 General

The mandatory interfaceList element within the functionType element provides the definition of the interface of the function type. Elements of the interface are:

— the input variables, and/or

— the output variables, and/or

— the configuration variables

of the function type.

Consequently the interfaceList element contains a sequence of three elements, where each element represents lists of one to many variable declarations encoded as varDeclaration elements.

— one optional element inputVars,

— one optional element outputVars, and

— one optional element configVars.

Neither the interfaceList, nor the inputVars, outputVars or configVars elements have any attributes.

G.4.5.4.3.2 Element varDeclaration

In the context of the definition of a structured data type, the varDeclaration element describes a single component variable (member) of the structure.

In the context of the definition of the interface of a function type, the varDeclaration element describes a single interface variable of the function type.

The data type of the component variable or interface variable is either defined by an element describing a simple data type out of the group g_simple, or by an element dataTypeIDRef, which references one of the defined complex data types within the dataTypeList element.

G.4.5.2.2.1 describes the group g_simple and G.4.5.4.3.3 describes the dataTypeIDRef element.

All further properties of the variable are contained in the attributes of the varDeclaration element, as given in Table G.24.

Table G.24 — Attributes of element varDeclaration

Attribute	Data type	Use	Description
name	xsd:string	required	Name of the interface variable or structure component
uniqueID	xsd:ID	required	Unique ID of the interface variable or structure component
size	xsd:string	optional	Number of elements, if the interface variable or structure component is of type anonymous ARRAY, BITSTRING, STRING or WSTRING.
initialValue	xsd:string	optional	Initial value of the interface variable or structure component.
description	xsd:string	optional	Optional textual description of the interface variable or structure component

G.4.5.4.3.3 Element dataTypeIDRef

The dataTypeIDRef element serves to reference a complex data type inside the dataTypeList element (see G.4.5.2), either from an interface variable of a function type, or from an array type definition, or from a component variable inside the definition of a structured data type.

The reference of type xsd:IDREF is provided as an attribute of the dataTypeIDRef element, as given in Table G.25.

Table G.25 — Attributes of element dataTypeIDRef

Attribute	Data type	Use	Description
uniqueIDRef	xsd:IDREF	required	Unique ID of the referenced data type

G.4.5.5 Element functionInstanceList

G.4.5.5.1 General

If the optional ApplicationProcess element is present in the device profile, it contains a mandatory functionInstanceList element, which contains a sequence of one to many functionInstance elements and zero to many connection elements.

At the application process level, the functionInstance elements represent the accessible application functions of the device type, independent of the network type or protocol. The connection elements represent connections - if any - between specific output and input variables of those function instances.

The functionInstanceList element also appears as an optional sub-element of the functionType element (see G.4.5.4). Like on application process level, the functionInstanceList element in that case contains a sequence of one to many functionInstance elements and zero to many connection elements.

The functionInstanceList element is only present within a functionType element, if a function is hierarchically structured. In this case the functionInstance elements represent the internal functions contained in the function type, and the connection elements the optional internal connections. These functions and their optional connections would be instantiated together with the instantiation of the containing function type.

The functionInstanceList element does not have any attributes.

G.4.5.5.2 Element functionInstance

The mandatory functionInstance element does not contain any sub-elements.

The functionInstance element shall contain the attributes given in Table G.26.

Table G.26 — Attributes of element functionInstance

Attribute	Data type	Use	Description
name	xsd:string	required	Name of the function instance
uniqueID	xsd:ID	required	Unique ID of the function instance (see NOTE)
typeIDRef	xsd:IDREF	required	Unique ID of the referenced function type
description	xsd:string	optional	Optional textual description of the function instance
NOTE When creating the unique ID for a function instance, it is essential that the ID is unique over all created IDs inside the XML source file. To allow equal names for function instances inside different function types, the ID of a function instance should generally concatenate the name of the containing function type with the instance name, to guarantee uniqueness.			

G.4.5.5.3 Element connection

The optional connection element defines a connection between an output variable of a function instance and an input variable of another function instance. Inside function types, the connection may also be drawn between an input variable of the function type and an input variable of a contained function instance, or between an output variable of a contained function instance and an output variable of the function type. The connection element may appear zero to many times.

The connection element contains the attributes given in Table G.27.

Table G.27 — Attributes of element connection

Attribute	Data type	Use	Description
source	xsd:string	required	Starting point of connection
destination	xsd:string	required	Endpoint of connection
description	xsd:string	optional	Optional textual description of the connection

EXAMPLE

The values of the source and the destination attributes may be used to encode the starting point and the endpoint of a connection using the syntax <function_instance_name>'.'<variable_name>; example for the value of a source attribute: 'PowerMeasures.Frequency'. Connections to interface variables of a function type use the name of the interface variable only.

G.4.5.6 Element parameterList

G.4.5.6.1 General

If the optional ApplicationProcess element is present in the device profile, it contains a mandatory parameterList element shown in Figure G.7, which represents a sequence of one to many parameter elements.

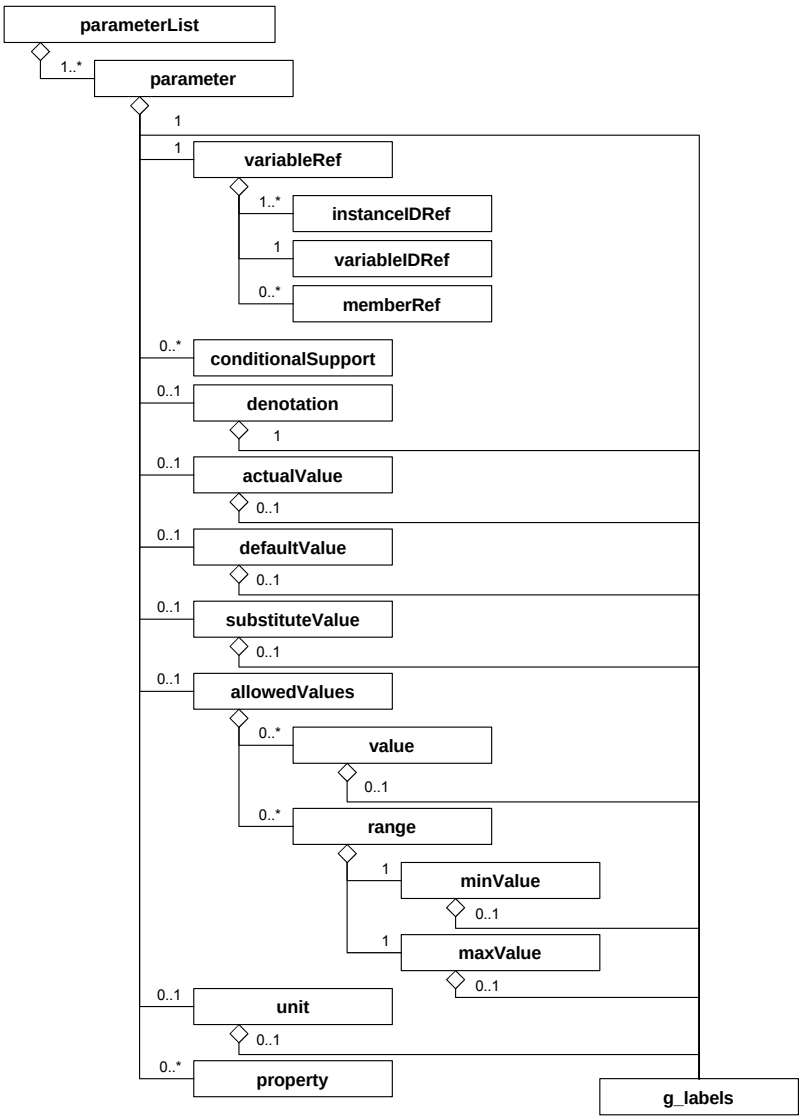


Figure G.7 — parameterList

Each of the parameter elements represents a parameter of the device profile. Multilingual names and/or descriptions for the parameters are provided through the group g_labels. G.2.2.2 describes the group g_labels.

The parameter is described by:

— its attributes,

— a reference to one (or more) interface variable(s) of one (or more) function instance(s) (mandatory element variableRef),

— a possibly empty set of sub-elements (conditionalSupport, denotation, actualValue, defaultValue, substituteValue, allowedValues, unit, property and g_labels).

NOTE References to multiple variables are a special case: specific parameters may reference an output variable of one function instance and an input variable of another function instance at the same time. In this case the data types of the two variables shall be the same. The XML parser cannot check the equality of data types. This can only be checked by a supporting tool.

G.4.5.6.2 Element parameter

G.4.5.6.2.1 General

The parameter element shall contain the attributes given in Table G.28.

Table G.28 — Attributes of element parameter

Attribute	Data type	Use	Description
uniqueID	xsd:ID	required	Unique ID of the parameter
access	xsd:NMTOKEN	required	Defines which operations are valid for the parameter: — read - read access only (default value) — write - write access only — readWrite - both read and write access — noAccess - access denied
support	xsd:NMTOKEN	optional	Defines whether or not the parameter has to be implemented in the device; valid values: — mandatory - parameter implementation is required — optional - parameter implementation is possible but not required — conditional - parameter implementation is required if one or more other optional parameter(s) is (are) implemented; these parameters are specified using the sub-element conditionalSupport
persistence	xsd:boolean	default	Defines the behaviour after power failure; valid are values false (default) and true.
offset	xsd:string	optional	Offset which is added to an actual value to form a scaled value: EngineeringValue = (ParameterValue + offset) * multiplier; if not present, offset = 0 is assumed
multiplier	xsd:string	optional	Scaling factor by which an actual value is multiplied to form a scaled value: EngineeringValue = (ParameterValue + offset) * multiplier; if not present, multiplier = 1 is assumed

G.4.5.6.2.2 Element conditionalSupport

One or more elements conditionalSupport are only present if the value of the attribute support of the parameter element is conditional. Each element refers to a single optional parameter. If at least one of those optional parameters is implemented, the conditional parameter has also to be implemented.

The element conditionalSupport shall contain the single attribute given in Table G.29.

Table G.29 — Attributes of element conditionalSupport

Attribute	Data type	Use	Description
paramIDRef	xsd:IDREF	required	Unique ID of the referenced optional parameter

G.4.5.6.2.3 Element denotation

The denotation element serves to provide application specific alias names in different languages for a parameter. The multilingual alias names (and optional, additional descriptions) for the parameter are provided through the group g_labels. The denotation element does not have any attributes.

G.4.5.6.2.4 Element actualValue

The actualValue element serves to hold the actual value of the parameter. An optional g_labels sub-element may provide multilingual descriptive information for this value. The value itself is provided in the value attribute of the element actualValue. An offset and multiplier may also be provided.

The attributes of the element actualValue shall be as given in Table G.30.

Table G.30 — Attributes of element actualValue

Attribute	Data type	Use	Description
value	xsd:string	required	Actual value
offset	xsd:string	optional	Offset which is added to an actual value to form a scaled value: EngineeringValue = (value + offset) * multiplier; if not present, the respective value of the parameter element shall be used
multiplier	xsd:string	optional	Scaling factor by which an actual value is multiplied to form a scaled value: EngineeringValue = (value + offset) * multiplier; if not present, the respective value of the parameter element shall be used

G.4.5.6.2.5 Element defaultValue

The defaultValue element serves to hold the default value of the parameter. This value overwrites the initial value of the interface variable of the function type associated with the parameter.

An optional g_labels sub-element may provide multilingual names and/or descriptive information for this value. The value itself is provided by the value attribute of the element defaultValue. An offset and multiplier may also be provided.

The attributes of the element defaultValue shall be as given in Table G.30.

G.4.5.6.2.6 Element substituteValue

The substituteValue element defines a specific value of the parameter that is provided to the device application in certain device operating states (for example device fault).

An optional g_labels sub-element may provide multilingual names and/or descriptive information for this value. The value itself is provided by the value attribute of the element substituteValue. An offset and multiplier may also be provided.

The attributes of the element substituteValue shall be as given in Table G.30.

G.4.5.6.2.7 Element allowedValues

The allowedValues element defines a list of supported values and/or a single range or several ranges of supported values for the parameter.

The list of supported values is represented by zero to many value sub-elements of the allowedValues element, whereas the ranges are represented by zero to many range sub-elements of the allowedValues element.

The value sub-element holds a single allowed value of the parameter. An optional g_labels sub-element may provide multilingual names and/or descriptive information for this value. The value itself is provided by the value attribute of the element value. An offset and multiplier may also be provided.

The attributes of the element value shall be as given in Table G.30.

The range sub-element contains two required sub-elements, namely the element minValue and the element maxValue, which define the limits of that range of allowed values. The minValue and the maxValue elements have the same structure and attributes as the value sub-element of the allowedValues element, such that the description of the value sub-element and Table G.30 are also valid for these sub-elements.

G.4.5.6.2.8 Element unit

The unit element defines the engineering unit of a parameter (for example time, temperature, pressure, flow, acceleration, current, energy…), as specified by ISO 1000. An optional g_labels sub-element may provide multilingual names and/or descriptive information for the engineering unit.

The attributes of the element unit shall be as given in Table G.31.

Table G.31 — Attributes of element unit

Attribute	Data type	Use	Description
multiplier	xsd:string	required	Multiplier for engineering units of analog parameters
unitURI	xsd:anyURI	optional	Link to the respective unit definition in a file containing all engineering units (for example time, temperature, pressure, flow, acceleration, current, energy…), as standardized by ISO 1000

G.4.5.6.2.9 Element variableRef

The variableRef element builds a reference to an interface variable of a function instance, or, if the variable is an array or a structure, possibly a reference to a member of the variable (array element or structure component).

In a hierarchically structured ApplicationProcess element, function instances can be located inside function instances of other function types. Therefore a specific instance in the functional tree can only be accessed by stepping through the tree, i.e. the specific instance shall be addressed through a concatenation of instance names. To map this concatenation and allow the referencing of a member, the variableRef element contains:

— a sequence of one to many instanceIDRef elements, followed by

— a single mandatory variableIDRef element, and

— an optional memberRef element.

The variableRef element has the attribute given in Table G.32.

Table G.32 — Attribute of element variableRef

Attribute	Data type	Use	Description
position	xsd:unsignedByte	default	Defines the sequence of multiple mapped data items into a single parameter object; position=1 means starting the mapping at the lowest bit position; the number of bits is defined by the data type of the data item; subsequent data items are packed without gaps; default value: 1 (see Note)
NOTE Attribute can be omitted for a single mapped data item.			

G.4.5.6.2.10 Element instanceIDRef

The instanceIDRef element serves to reference a function instance inside a functionInstanceList element, which may reside either on the level of the ApplicationProcess element or on the level of a functionType element.

The reference of type xsd:IDREF is provided as an attribute of the instanceIDRef element, as given in Table G.33.

Table G.33 — Attributes of element instanceIDRef

Attribute	Data type	Use	Description
uniqueIDRef	xsd:IDREF	required	Unique ID of the referenced function instance

G.4.5.6.2.11 Element variableIDRef

The variableIDRef element serves to reference an interface variable of a function type inside the functionTypeList element.

In a given variableRef element the instance of that function type is defined by the functionInstance element referenced by the instanceIDRef element, which is just preceding the variableIDRef element.

The reference of type xsd:IDREF is provided as an attribute of the variableIDRef element, as given in Table G.34.

Table G.34 — Attributes of element variableIDRef

Attribute	Data type	Use	Description
uniqueIDRef	xsd:IDREF	required	Unique ID of the referenced interface variable of a function type

G.4.5.6.2.12 Element memberRef

The optional memberRef element either references the respective component of an interface variable of structured data type (attribute uniqueIDRef is used), or the respective array element of an interface variable of array data type (attribute index is used). One of these two attributes shall be present, if the memberRef element is present.

The memberRef element shall contain the attributes given in Table G.35.

Table G.35 — Attributes of element memberRef

Attribute	Data type	Use	Description
uniqueIDRef	xsd:IDREF	optional	Unique ID of the referenced component of a structured data type
index	xsd:long	optional	Index of the referenced array element

G.4.5.6.3 Element property

The property element is introduced as a generic element to allow the inclusion of values for additional, specific properties into the description of a parameter.

The property element shall contain the attributes given in Table G.36.

Table G.36 — Attributes of element property

Attribute	Data type	Use	Description
name	xsd:string	required	Name of the property
value	xsd:string	required	Value of the property

G.4.5.7 Element parameterGroupList

G.4.5.7.1 General

The optional parameterGroupList element, if present, contains a sequence of one to many parameterGroup elements, as shown in Figure G.8. Multilingual names and/or descriptions for the parameter groups are provided through the group g_labels. See G.2.2.2 for the description of the group g_labels.

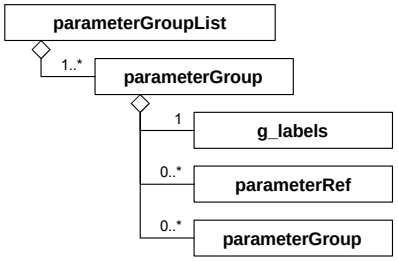


Figure G.8 — parameterGroupList

G.4.5.7.2 Element parameterGroup

Each of the parameterGroup elements combines a set of parameters out of the parameterList element to build a group of parameters which serve a specific purpose, for example to prepare HMI views. This purpose is indicated by the value of the kindOfAccess attribute of the parameterGroup element. It is possible to define a hierarchy of parameter groups.

The respective parameters in the set are referenced through the corresponding number of parameterRef elements.

The parameterGroup element contains the attributes given in Table G.37.

Table G.37 — Attributes of element parameterGroup

Attribute	Data type	Use	Description
uniqueID	xsd:ID	required	Unique ID of the parameter group
kindOfAccess	xsd:string	optional	Classifies parameters of the parameter group.

G.4.5.7.3 Element parameterRef

The parameterRef element serves to reference a parameter element inside the parameterList element of the ApplicationProcess element.

The reference of type xsd:IDREF is provided as an attribute of the parameterRef element, as given in Table G.38.

Table G.38 — Attributes of element parameterRef

Attribute	Data type	Use	Description
uniqueIDRef	xsd:IDREF	required	Unique ID of the referenced parameter

G.4.6 EPL device profile template schemas

G.4.6.1 XML schema: ISO15745ProfileContainer.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <xsd:element name="ISO15745ProfileContainer">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="ISO15745Profile" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="ISO15745Profile">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="ProfileHeader" type="ProfileHeader_DataType"/>
            <xsd:element name="ProfileBody" type="ProfileBody_DataType"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:annotation>
      <xsd:documentation>* HEADER SECTION *</xsd:documentation>
   </xsd:annotation>
   <xsd:complexType name="ProfileHeader_DataType">
      <xsd:sequence>
         <xsd:element name="ProfileIdentification" type="xsd:string"/>
         <xsd:element name="ProfileRevision" type="xsd:string"/>
         <xsd:element name="ProfileName" type="xsd:string"/>
         <xsd:element name="ProfileSource" type="xsd:string"/>
         <xsd:element name="ProfileClassID" type="ProfileClassID_DataType"/>
         <xsd:element name="ProfileDate" type="xsd:date" minOccurs="0"/>
         <xsd:element name="AdditionalInformation" type="xsd:anyURI" minOccurs="0"/>
         <xsd:element name="ISO15745Reference" type="ISO15745Reference_DataType"/>
         <xsd:element name="IASInterfaceType" type="IASInterface_DataType" minOccurs="0"
maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:annotation>
      <xsd:documentation>* BODY SECTION *</xsd:documentation>
   </xsd:annotation>
   <xsd:complexType name="ProfileBody_DataType" abstract="true"/>
   <xsd:annotation>
      <xsd:documentation>* HEADER DATA TYPES *</xsd:documentation>
   </xsd:annotation>
   <xsd:simpleType name="ProfileClassID_DataType">
      <xsd:restriction base="xsd:string">
         <xsd:enumeration value="AIP"/>
         <xsd:enumeration value="Process"/>
         <xsd:enumeration value="InformationExchange"/>
         <xsd:enumeration value="Resource"/>
         <xsd:enumeration value="Device"/>
         <xsd:enumeration value="CommunicationNetwork"/>
         <xsd:enumeration value="Equipment"/>
         <xsd:enumeration value="Human"/>
         <xsd:enumeration value="Material"/>
      </xsd:restriction>
   </xsd:simpleType>
   <xsd:complexType name="ISO15745Reference_DataType">
      <xsd:sequence>
         <xsd:element name="ISO15745Part" type="xsd:positiveInteger"/>
         <xsd:element name="ISO15745Edition" type="xsd:positiveInteger"/>
         <xsd:element name="ProfileTechnology" type="xsd:string"/>
```

```
      </xsd:sequence>
   </xsd:complexType>
   <xsd:simpleType name="IASInterface_DataType">
      <xsd:union>
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="CSI"/>
               <xsd:enumeration value="HCI"/>
               <xsd:enumeration value="ISI"/>
               <xsd:enumeration value="API"/>
               <xsd:enumeration value="CMI"/>
               <xsd:enumeration value="ESI"/>
               <xsd:enumeration value="FSI"/>
               <xsd:enumeration value="MTI"/>
               <xsd:enumeration value="SEI"/>
               <xsd:enumeration value="USI"/>
            </xsd:restriction>
         </xsd:simpleType>
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:length value="4"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:union>
   </xsd:simpleType>
   <xsd:annotation>
      <xsd:documentation>* ISO 15745 DEFINED DATA TYPES *</xsd:documentation>
   </xsd:annotation>
   <xsd:complexType name="ProfileHandle_DataType">
      <xsd:sequence>
         <xsd:element name="ProfileIdentification" type="xsd:string"/>
         <xsd:element name="ProfileRevision" type="xsd:string"/>
         <xsd:element name="ProfileLocation" type="xsd:anyURI" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:schema>
```

G.4.6.2 XML schema: CommonElements.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <!--################ common attribute group-->
   <xsd:attributeGroup name="ag_formatAndFile">
      <xsd:attribute name="formatName" type="xsd:string" fixed="DDXML" form="unqualified"/>
      <xsd:attribute name="formatVersion" type="xsd:string" fixed="2.0" form="unqualified"/>
      <xsd:attribute name="fileName" type="xsd:string" use="required" form="unqualified"/>
      <xsd:attribute name="fileCreator" type="xsd:string" use="required" form="unqualified"/>
      <xsd:attribute name="fileCreationDate" type="xsd:date" use="required" form="unqualified"/>
      <xsd:attribute name="fileCreationTime" type="xsd:time" use="optional"/>
      <xsd:attribute name="fileModificationDate" type="xsd:date" use="optional" form="unqualified"/>
      <xsd:attribute name="fileModificationTime" type="xsd:time" use="optional"/>
      <xsd:attribute name="fileModifiedBy" type="xsd:string" use="optional"/>
      <xsd:attribute name="fileVersion" type="xsd:string" use="required" form="unqualified"/>
   </xsd:attributeGroup>
   <!--################ common groups-->
   <xsd:group name="g_labels">
      <xsd:sequence>
         <xsd:choice maxOccurs="unbounded">
            <xsd:element name="label">
               <xsd:complexType>
                  <xsd:simpleContent>
                     <xsd:extension base="xsd:string">
                        <xsd:attribute name="lang" type="xsd:language" use="required"/>
                        <xsd:attribute name="URI" type="xsd:anyURI" use="optional"/>
                     </xsd:extension>
                  </xsd:simpleContent>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="description">
               <xsd:complexType>
                  <xsd:simpleContent>
                     <xsd:extension base="xsd:string">
                        <xsd:attribute name="lang" type="xsd:language" use="required"/>
                        <xsd:attribute name="URI" type="xsd:anyURI" use="optional"/>
                     </xsd:extension>
```

```
               </xsd:simpleContent>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="labelRef">
            <xsd:complexType>
               <xsd:simpleContent>
                  <xsd:extension base="xsd:anyURI">
                     <xsd:attribute name="dictID" type="xsd:IDREF" use="required"/>
                     <xsd:attribute name="textID" type="xsd:string" use="optional"/>
                  </xsd:extension>
               </xsd:simpleContent>
            </xsd:complexType>
         </xsd:element>
      </xsd:choice>
   </xsd:sequence>
</xsd:group>
<xsd:group name="g_simple">
   <xsd:choice>
      <xsd:element name="BOOL"/>
      <xsd:element name="BITSTRING"/>
      <xsd:element name="BYTE"/>
      <xsd:element name="CHAR"/>
      <xsd:element name="WORD"/>
      <xsd:element name="DWORD"/>
      <xsd:element name="LWORD"/>
      <xsd:element name="SINT"/>
      <xsd:element name="INT"/>
      <xsd:element name="DINT"/>
      <xsd:element name="LINT"/>
      <xsd:element name="USINT"/>
      <xsd:element name="UINT"/>
      <xsd:element name="UDINT"/>
      <xsd:element name="ULINT"/>
      <xsd:element name="REAL"/>
      <xsd:element name="LREAL"/>
      <xsd:element name="TIME"/>
      <xsd:element name="DATE"/>
      <xsd:element name="DT"/>
      <xsd:element name="TOD"/>
      <xsd:element name="STRING"/>
      <xsd:element name="WSTRING"/>
   </xsd:choice>
</xsd:group>
<!--################ common elements-->
<xsd:element name="vendorID">
   <xsd:complexType>
      <xsd:simpleContent>
         <xsd:extension base="xsd:string">
            <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
         </xsd:extension>
      </xsd:simpleContent>
   </xsd:complexType>
</xsd:element>
<xsd:element name="deviceFamily">
   <xsd:complexType>
      <xsd:group ref="g_labels"/>
      <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="productID">
   <xsd:complexType>
      <xsd:simpleContent>
         <xsd:extension base="xsd:string">
            <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
         </xsd:extension>
      </xsd:simpleContent>
   </xsd:complexType>
</xsd:element>
<xsd:element name="version">
   <xsd:complexType>
      <xsd:simpleContent>
         <xsd:extension base="xsd:string">
            <xsd:attribute name="versionType" use="required">
               <xsd:simpleType>
                  <xsd:restriction base="xsd:NMTOKEN">
                     <xsd:enumeration value="SW"/>
```

```
                        <xsd:enumeration value="FW"/>
                        <xsd:enumeration value="HW"/>
                     </xsd:restriction>
                  </xsd:simpleType>
               </xsd:attribute>
               <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
            </xsd:extension>
         </xsd:simpleContent>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="buildDate" type="xsd:date"/>
   <xsd:element name="specificationRevision">
      <xsd:complexType>
         <xsd:simpleContent>
            <xsd:extension base="xsd:string">
               <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
            </xsd:extension>
         </xsd:simpleContent>
      </xsd:complexType>
   </xsd:element>
</xsd:schema>
```

G.4.6.3 XML schema: ProfileBody_Device_EPL.xsd

The XML schema ProfileBody_Device_EPL.xsd includes the schema ISO15745ProfileContainer.xsd in G.4.6.1 and the schema CommonElements.xsd in G.4.6.2.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <xsd:include schemaLocation="ISO15745ProfileContainer.xsd"/>
   <xsd:include schemaLocation="CommonElements.xsd"/>
   <!--################ profile body device -->
   <xsd:complexType name="ProfileBody_Device_EPL">
      <xsd:complexContent>
         <xsd:extension base="ProfileBody_DataType">
            <xsd:sequence>
               <xsd:element ref="DeviceIdentity" minOccurs="0"/>
               <xsd:element ref="DeviceManager" minOccurs="0"/>
               <xsd:element ref="DeviceFunction" maxOccurs="unbounded"/>
               <xsd:element ref="ApplicationProcess" minOccurs="0" maxOccurs="unbounded"/>
               <xsd:element name="ExternalProfileHandle" type="ProfileHandle_DataType" minOccurs="0"
maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attributeGroup ref="ag_formatAndFile"/>
            <xsd:attribute name="supportedLanguages" type="xsd:NMTOKENS" use="optional"/>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
   <!--################ device identity elements -->
   <xsd:element name="DeviceIdentity">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="vendorName"/>
            <xsd:element ref="vendorID" minOccurs="0"/>
            <xsd:element ref="vendorText" minOccurs="0"/>
            <xsd:element ref="deviceFamily" minOccurs="0"/>
            <xsd:element ref="productFamily" minOccurs="0"/>
            <xsd:element ref="productName"/>
            <xsd:element ref="productID" minOccurs="0"/>
            <xsd:element ref="productText" minOccurs="0"/>
            <xsd:element ref="orderNumber" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element ref="version" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element ref="buildDate" minOccurs="0"/>
            <xsd:element ref="specificationRevision" minOccurs="0"/>
            <xsd:element ref="instanceName" minOccurs="0"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="productFamily">
      <xsd:complexType>
         <xsd:simpleContent>
            <xsd:extension base="xsd:string">
               <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
            </xsd:extension>
```

```
      </xsd:simpleContent>
   </xsd:complexType>
</xsd:element>
<xsd:element name="instanceName">
   <xsd:complexType>
      <xsd:simpleContent>
         <xsd:extension base="xsd:string">
            <xsd:attribute name="readOnly" type="xsd:boolean" default="false"/>
         </xsd:extension>
      </xsd:simpleContent>
   </xsd:complexType>
</xsd:element>
<xsd:element name="orderNumber">
   <xsd:complexType>
      <xsd:simpleContent>
         <xsd:extension base="xsd:string">
            <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
         </xsd:extension>
      </xsd:simpleContent>
   </xsd:complexType>
</xsd:element>
<xsd:element name="productName">
   <xsd:complexType>
      <xsd:simpleContent>
         <xsd:extension base="xsd:string">
            <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
         </xsd:extension>
      </xsd:simpleContent>
   </xsd:complexType>
</xsd:element>
<xsd:element name="productText">
   <xsd:complexType>
      <xsd:group ref="g_labels"/>
      <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="vendorName">
   <xsd:complexType>
      <xsd:simpleContent>
         <xsd:extension base="xsd:string">
            <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
         </xsd:extension>
      </xsd:simpleContent>
   </xsd:complexType>
</xsd:element>
<xsd:element name="vendorText">
   <xsd:complexType>
      <xsd:group ref="g_labels"/>
      <xsd:attribute name="readOnly" type="xsd:boolean" default="true"/>
   </xsd:complexType>
</xsd:element>
<!--################ device manager elements -->
<xsd:element name="DeviceManager">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="indicatorList" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="indicatorList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="LEDList" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="LEDList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="LED" maxOccurs="unbounded"/>
         <xsd:element ref="combinedState" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="LED">
   <xsd:complexType>
```

```
      <xsd:sequence>
         <xsd:group ref="g_labels"/>
         <xsd:element ref="LEDstate" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="LEDcolors" use="required">
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="monocolor"/>
               <xsd:enumeration value="bicolor"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="LEDtype" use="optional">
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="IO"/>
               <xsd:enumeration value="device"/>
               <xsd:enumeration value="communication"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:attribute>
   </xsd:complexType>
</xsd:element>
<xsd:element name="LEDstate">
   <xsd:complexType>
      <xsd:group ref="g_labels"/>
      <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
      <xsd:attribute name="state" use="required">
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="on"/>
               <xsd:enumeration value="off"/>
               <xsd:enumeration value="flashing"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="LEDcolor" use="required">
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="green"/>
               <xsd:enumeration value="amber"/>
               <xsd:enumeration value="red"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="flashingPeriod" type="xsd:unsignedInt" use="optional"/>
      <xsd:attribute name="impulsWidth" type="xsd:unsignedByte" default="50"/>
      <xsd:attribute name="numberOfImpulses" type="xsd:unsignedByte" default="1"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="combinedState">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:group ref="g_labels"/>
         <xsd:element name="LEDstateRef" minOccurs="2" maxOccurs="unbounded">
            <xsd:complexType>
               <xsd:attribute name="stateIDRef" type="xsd:IDREF" use="required"/>
            </xsd:complexType>
         </xsd:element>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<!--################ device function elements -->
<xsd:element name="DeviceFunction">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="capabilities"/>
         <xsd:element ref="picturesList" minOccurs="0"/>
         <xsd:element ref="dictionaryList" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="capabilities">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="characteristicsList" maxOccurs="unbounded"/>
```

```
         <xsd:element ref="standardComplianceList" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="characteristicsList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="category" minOccurs="0">
            <xsd:complexType>
               <xsd:group ref="g_labels"/>
            </xsd:complexType>
         </xsd:element>
         <xsd:element ref="characteristic" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="characteristic">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="characteristicName"/>
         <xsd:element ref="characteristicContent" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="characteristicContent">
   <xsd:complexType>
      <xsd:group ref="g_labels"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="characteristicName">
   <xsd:complexType>
      <xsd:group ref="g_labels"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="standardComplianceList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="compliantWith" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="compliantWith">
   <xsd:complexType>
      <xsd:group ref="g_labels"/>
      <xsd:attribute name="name" type="xsd:string" use="required"/>
      <xsd:attribute name="range" default="international">
         <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
               <xsd:enumeration value="international"/>
               <xsd:enumeration value="internal"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:attribute>
   </xsd:complexType>
</xsd:element>
<xsd:element name="picturesList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="picture" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="picture">
   <xsd:complexType>
      <xsd:group ref="g_labels"/>
      <xsd:attribute name="URI" type="xsd:anyURI" use="required"/>
      <xsd:attribute name="number" type="xsd:unsignedInt" use="optional"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="dictionaryList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="dictionary" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

```
<xsd:element name="dictionary">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="file" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="lang" type="xsd:language" use="required"/>
      <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="file">
   <xsd:complexType>
      <xsd:attribute name="URI" type="xsd:anyURI" use="required"/>
   </xsd:complexType>
</xsd:element>
<!--################ application process elements -->
<xsd:element name="ApplicationProcess">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="dataTypeList" minOccurs="0"/>
         <xsd:element ref="functionTypeList"/>
         <xsd:element ref="functionInstanceList"/>
         <xsd:element ref="parameterList"/>
         <xsd:element ref="parameterGroupList" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="dataTypeList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:group ref="g_complex" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="functionTypeList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="functionType" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="functionType">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="versionInfo" maxOccurs="unbounded"/>
         <xsd:element ref="interfaceList"/>
         <xsd:element ref="functionInstanceList" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="name" type="xsd:string" use="required"/>
      <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
      <xsd:attribute name="description" type="xsd:string" use="optional"/>
      <xsd:attribute name="package" type="xsd:string" use="optional"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="versionInfo">
   <xsd:complexType>
      <xsd:attribute name="organization" type="xsd:string" use="required"/>
      <xsd:attribute name="version" type="xsd:string" use="required"/>
      <xsd:attribute name="author" type="xsd:string" use="required"/>
      <xsd:attribute name="date" type="xsd:date" use="required"/>
      <xsd:attribute name="remarks" type="xsd:string" use="optional"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="interfaceList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="inputVars" minOccurs="0"/>
         <xsd:element ref="outputVars" minOccurs="0"/>
         <xsd:element ref="configVars" minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
<xsd:element name="inputVars">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="varDeclaration" maxOccurs="unbounded"/>
      </xsd:sequence>
```

```
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="outputVars">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="varDeclaration" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="configVars">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="varDeclaration" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="varDeclaration">
      <xsd:complexType>
         <xsd:choice>
            <xsd:group ref="g_simple"/>
            <xsd:element ref="dataTypeIDRef"/>
         </xsd:choice>
         <xsd:attribute name="name" type="xsd:string" use="required"/>
         <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
         <xsd:attribute name="size" type="xsd:string" use="optional"/>
         <xsd:attribute name="initialValue" type="xsd:string" use="optional"/>
         <xsd:attribute name="description" type="xsd:string" use="optional"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="dataTypeIDRef">
      <xsd:complexType>
         <xsd:attribute name="uniqueIDRef" type="xsd:IDREF" use="required"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="functionInstanceList">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="functionInstance" maxOccurs="unbounded"/>
            <xsd:element ref="connection" minOccurs="0" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="functionInstance">
      <xsd:complexType>
         <xsd:attribute name="name" type="xsd:string" use="required"/>
         <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
         <xsd:attribute name="typeIDRef" type="xsd:IDREF" use="required"/>
         <xsd:attribute name="description" type="xsd:string" use="optional"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="connection">
      <xsd:complexType>
         <xsd:attribute name="source" type="xsd:string" use="required"/>
         <xsd:attribute name="destination" type="xsd:string" use="required"/>
         <xsd:attribute name="description" type="xsd:string" use="optional"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="parameterList">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="parameter" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="parameter">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:group ref="g_labels"/>
            <xsd:element ref="variableRef" maxOccurs="unbounded"/>
            <xsd:element ref="conditionalSupport" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element ref="denotation" minOccurs="0"/>
            <xsd:element ref="actualValue" minOccurs="0"/>
            <xsd:element ref="defaultValue" minOccurs="0"/>
            <xsd:element ref="substituteValue" minOccurs="0"/>
            <xsd:element ref="allowedValues" minOccurs="0"/>
            <xsd:element ref="unit" minOccurs="0"/>
```

```
            <xsd:element ref="property" minOccurs="0" maxOccurs="unbounded"/>
         </xsd:sequence>
         <xsd:attributeGroup ref="ag_parameter"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="variableRef">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="instanceIDRef" maxOccurs="unbounded"/>
            <xsd:element ref="variableIDRef"/>
            <xsd:element ref="memberRef" minOccurs="0" maxOccurs="unbounded"/>
         </xsd:sequence>
         <xsd:attribute name="position" type="xsd:unsignedByte" default="1"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="instanceIDRef">
      <xsd:complexType>
         <xsd:attribute name="uniqueIDRef" type="xsd:IDREF"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="variableIDRef">
      <xsd:complexType>
         <xsd:attribute name="uniqueIDRef" type="xsd:IDREF"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="memberRef">
      <xsd:complexType>
         <xsd:attribute name="uniqueIDRef" type="xsd:IDREF" use="optional"/>
         <xsd:attribute name="index" type="xsd:long" use="optional"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="actualValue">
      <xsd:complexType>
         <xsd:group ref="g_labels" minOccurs="0"/>
         <xsd:attributeGroup ref="ag_value"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="conditionalSupport">
      <xsd:complexType>
         <xsd:attribute name="paramIDRef" type="xsd:IDREF" use="required"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="denotation">
      <xsd:complexType>
         <xsd:group ref="g_labels"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="defaultValue">
      <xsd:complexType>
         <xsd:group ref="g_labels" minOccurs="0"/>
         <xsd:attributeGroup ref="ag_value"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="substituteValue">
      <xsd:complexType>
         <xsd:group ref="g_labels" minOccurs="0"/>
         <xsd:attributeGroup ref="ag_value"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="allowedValues">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="value" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element ref="range" minOccurs="0" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="value">
      <xsd:complexType>
         <xsd:group ref="g_labels" minOccurs="0"/>
         <xsd:attributeGroup ref="ag_value"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="range">
      <xsd:complexType>
         <xsd:sequence>
```

```
            <xsd:element name="minValue">
               <xsd:complexType>
                  <xsd:group ref="g_labels" minOccurs="0"/>
                  <xsd:attributeGroup ref="ag_value"/>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="maxValue">
               <xsd:complexType>
                  <xsd:group ref="g_labels" minOccurs="0"/>
                  <xsd:attributeGroup ref="ag_value"/>
               </xsd:complexType>
            </xsd:element>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="unit">
      <xsd:complexType>
         <xsd:group ref="g_labels"/>
         <xsd:attribute name="multiplier" type="xsd:string" use="required"/>
         <xsd:attribute name="unitURI" type="xsd:anyURI" use="optional"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="property">
      <xsd:complexType>
         <xsd:attribute name="name" type="xsd:string" use="required"/>
         <xsd:attribute name="value" type="xsd:string" use="required"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="parameterGroupList">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="parameterGroup" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="parameterGroup">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:group ref="g_labels"/>
            <xsd:element ref="parameterGroup" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element ref="parameterRef" minOccurs="0" maxOccurs="unbounded"/>
         </xsd:sequence>
         <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
         <xsd:attribute name="kindOfAccess" type="xsd:string" use="optional"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="parameterRef">
      <xsd:complexType>
         <xsd:attribute name="uniqueIDRef" type="xsd:IDREF" use="required"/>
      </xsd:complexType>
   </xsd:element>
   <!--################ complex types -->
   <xsd:element name="array">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="subrange" maxOccurs="unbounded"/>
            <xsd:choice>
               <xsd:group ref="g_simple"/>
               <xsd:element ref="dataTypeIDRef"/>
            </xsd:choice>
         </xsd:sequence>
         <xsd:attribute name="name" type="xsd:string" use="required"/>
         <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
         <xsd:attribute name="description" type="xsd:string" use="optional"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="subrange">
      <xsd:complexType>
         <xsd:attribute name="lowerLimit" type="xsd:long" use="required"/>
         <xsd:attribute name="upperLimit" type="xsd:long" use="required"/>
      </xsd:complexType>
   </xsd:element>
   <xsd:element name="struct">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="count" minOccurs="0"/>
```

```
         <xsd:element ref="varDeclaration" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="name" type="xsd:string" use="required"/>
      <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
      <xsd:attribute name="description" type="xsd:string" use="optional"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="enum">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="enumValue" maxOccurs="unbounded"/>
         <xsd:group ref="g_simple" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="name" type="xsd:string" use="required"/>
      <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
      <xsd:attribute name="size" type="xsd:string" use="optional"/>
      <xsd:attribute name="description" type="xsd:string" use="optional"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="enumValue">
   <xsd:complexType>
      <xsd:group ref="g_labels"/>
      <xsd:attribute name="value" type="xsd:string" use="optional"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="derived">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="count" minOccurs="0"/>
         <xsd:choice>
            <xsd:group ref="g_simple"/>
            <xsd:element ref="dataTypeIDRef"/>
         </xsd:choice>
      </xsd:sequence>
      <xsd:attribute name="name" type="xsd:string" use="required"/>
      <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
      <xsd:attribute name="description" type="xsd:string" use="optional"/>
   </xsd:complexType>
</xsd:element>
<xsd:element name="count">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:group ref="g_labels" minOccurs="0"/>
         <xsd:element ref="defaultValue"/>
         <xsd:element ref="allowedValues" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
      <xsd:attribute name="access" default="read">
         <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
               <xsd:enumeration value="read"/>
               <xsd:enumeration value="write"/>
               <xsd:enumeration value="readWrite"/>
               <xsd:enumeration value="noAccess"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:attribute>
   </xsd:complexType>
</xsd:element>
<!--################ group-->
<xsd:group name="g_complex">
   <xsd:choice>
      <xsd:element ref="array"/>
      <xsd:element ref="struct"/>
      <xsd:element ref="enum"/>
      <xsd:element ref="derived"/>
   </xsd:choice>
</xsd:group>
<!--################ attribute groups-->
<xsd:attributeGroup name="ag_parameter">
   <xsd:attribute name="uniqueID" type="xsd:ID" use="required"/>
   <xsd:attribute name="access" default="read">
      <xsd:simpleType>
         <xsd:restriction base="xsd:NMTOKEN">
            <xsd:enumeration value="read"/>
            <xsd:enumeration value="write"/>
```

```
               <xsd:enumeration value="readWrite"/>
               <xsd:enumeration value="noAccess"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="support" use="optional">
         <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
               <xsd:enumeration value="mandatory"/>
               <xsd:enumeration value="optional"/>
               <xsd:enumeration value="conditional"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="persistence" type="xsd:boolean" default="false"/>
      <xsd:attribute name="offset" type="xsd:string" use="optional"/>
      <xsd:attribute name="multiplier" type="xsd:string" use="optional"/>
   </xsd:attributeGroup>
   <xsd:attributeGroup name="ag_value">
      <xsd:attribute name="value" type="xsd:string" use="required"/>
      <xsd:attribute name="offset" type="xsd:string" use="optional"/>
      <xsd:attribute name="multiplier" type="xsd:string" use="optional"/>
   </xsd:attributeGroup>
</xsd:schema>
```

G.5 Communication network profile template description

G.5.1 ProfileBody

For the communication network profile the ProfileBody contains the ApplicationLayers, the TransportLayers and the NetworkManagement elements shown in Figure 37.

The ProfileBody element contains the attributes given in Table G.39.

Table G.39 — Attributes of element ProfileBody

Attribute	Data type	Use	Description
formatName	xsd:string	fixed	Format identifier
formatVersion	xsd:string	fixed	Format version identifier
fileName	xsd:string	required	Name of the file with extension without path
fileCreator	xsd:string	required	Person creating the file
fileCreationDate	xsd:date	required	Date of file creation
fileCreationTime	xsd:time	optional	Time of file creation
fileModifiedBy	xsd:string	optional	Person modifying the file
fileModificationDate	xsd.date	optional	Date of last file change
fileModificationTime	xsd:time	optional	Time of last file change
fileVersion	xsd:string	required	Vendor specific version of the file
supportedLanguages	xsd:NMTOKENS	optional	List of supported languages

G.5.2 ApplicationLayers

G.5.2.1 General

Figure 37 shows the structure of the ETHERNET Powerlink ApplicationLayers class.

G.5.2.2 CANopenObjectList

G.5.2.2.1 General

Figure 37 shows the structure of the CANopenObjectList element. The element contains one to many CANopenObject elements.

NOTE ETHERNET Powerlink has adopted CANopen (EN 50325-4) terminology and object structure (CANopen over ETHERNET Powerlink). Therefore CANopen terms are used here.

G.5.2.2.2 CANopenObject

G.5.2.2.2.1 General

Figure 37 shows the structure of the CANopenObject element. The element contains zero to many CANopenSubObject elements. The CANopenObject element and the CANopenSubObject element map the functional part of the ETHERNET Powerlink device profile to the CANopen over ETHERNET Powerlink communication network profile.

The CANopenObject element contains the attributes given in Table G.40.

Table G.40 — Attributes of element CANopenObject

Attribute	Data type	Use	Description
index	xsd:hexBinary	required	Index of the object (four hex digits)
name	xsd:string	required	Name of the object
objectType	xsd:unsignedByte	required	CANopen object type
dataType	xsd:hexBinary	optional	CANopen data type (two hex digits)
lowLimit	xsd:string	optional	Low limit of the parameter value
highLimit	xsd:string	optional	High limit of the parameter value
accessType	xsd:string	optional	Access type of the object; valid values: — ro – read only access — wo – write only access — rw – both read and write access — rwr – both read and write access; prefered read access — rww – both read and write access; preferred write access — const – read only access; the value is not changing
defaultValue	xsd:string	optional	Default value of the object
pdoMapping	xsd:boolean	optional	PDO mapping of the object; valid values: — true – mapped — false – not mapped
objFlags	xsd:hexBinary	optional	Controls the behaviour of tools (four hex digits)
uniqueIDRef	xsd:IDREF	optional	Unique ID of an appropriate element in the application process part referenced from this object. If the attribute is given the attributes dataType, lowLimit, highLimit, accessType, and defaultValue are given by the referenced element from the application process part.
subNumber	xsd:unsignedByte	optional	Number of sub-objects of the object

G.5.2.2.2.2 CANopenSubObject

The CANopenSubObject element has an empty content.

The CANopenSubObject element contains the attributes given in Table G.41.

Table G.41 — Attributes of element CANopenSubObject

Attribute	Data type	Use	Description
subIndex	xsd:hexBinary	required	Subindex of the object (two hex digits)
name	xsd:string	required	Name of the object
objectType	xsd:unsignedByte	required	CANopen object type
dataType	xsd:hexBinary	optional	CANopen data type (two hex digits)
lowLimit	xsd:string	optional	Low limit of the parameter value
highLimit	xsd:string	optional	High limit of the parameter value
accessType	xsd:string	optional	Access type of the object; valid values: — ro – read only access — wo – write only access — rw – both read and write access — rwr – both read and write access; preferred read access — rww – both read and write access; preferred write access — const – read only access; the value is not changing
defaultValue	xsd:string	optional	Default value of the object
pdoMapping	xsd:boolean	optional	PDO mapping of the object; valid values: — true – mapped — false – not mapped
objFlags	xsd:hexBinary	optional	Controls the behaviour of tools (four hex digits)
uniqueIDRef	xsd:IDREF	optional	Unique ID of an appropriate element in the application process part referenced from this object. If the attribute is given the attributes dataType, lowLimit, highLimit, accessType, and defaultValue are given by the referenced element from the application process part.

G.5.2.3 Identity

Since different communication profiles may require different identity information, an optional local identity element may be used within an ApplicationLayers element. This identity element may contain a subset of the sub-elements of the DeviceIdentity element described in G.4.2. All sub-element descriptions given there also apply for the sub-elements of this identity element.

G.5.2.4 Element dummyUsage

G.5.2.4.1 General

Figure 37 shows the structure of the dummyUsage element. The element contains one to many dummy elements.

G.5.2.4.2 Element dummy

The dummy element has no content. The element is used to enable and disable certain dummy entries for the dummy mapping.

The dummy element contains the attributes given in Table G.42.

Table G.42 — Attributes of element dummy

Attribute	Data type	Use	Description
entry	xsd:string	required	The string is constructed using the name of the dummy object, followed by the equal sign and then the value of either 0 for disabled mapping or 1 for enabled mapping. The allowed values are as follows: — Dummy0001=0 — Dummy0002=0 — Dummy0003=0 — Dummy0004=0 — Dummy0005=0 — Dummy0006=0 — Dummy0007=0 — Dummy0001=1 — Dummy0002=1 — Dummy0003=1 — Dummy0004=1 — Dummy0005=1 — Dummy0006=1 — Dummy0007=1

G.5.2.5 Element dynamicChannels

G.5.2.5.1 General

Figure 37 shows the structure of the dynamicChannels element. The element contains one to many dynamicChannel elements.

G.5.2.5.2 Element dynamicChannel

The dynamicChannel element contains an element describing a simple data type out of the group g_simple. The element is used to mark available channels that can be used to create a link between the data to be communicated in the ETHERNET Powerlink network and the application programme running on the device.

The dynamicChannel element contains the attributes given in Table G.43.

Table G.43 — Attributes of element dynamicChannel

Attribute	Data type	Use	Description
accessType	xsd:NMTOKEN	required	Access type of the object; valid values: — readOnly – read only access — writeOnly – write only access — readWriteWrite – both read and write access; preferred write access
startIndex	xsd:hexBinary	required	Start index of the object
endIndex	xsd:hexBinary	required	End index of the object
maxNumber	xsd:unsignedInt	required	Maximum number of links to application programme
addressOffset	xsd:hexBinary	required	Address offset within application programme memory
bitAlignment	xsd:unsignedByte	optional	Bit alingment of data within the object counted from the least significant bit.
manufacturerSpecific	xsd:string	optional	Available for manufacturer specific use

G.5.3 TransportLayers

The TransportLayers element has no content.

G.5.4 NetworkManagement

G.5.4.1 General

Figure 37 shows the structure of the ETHERNET Powerlink NetworkManagement class.

G.5.4.2 PowerlinkGeneralFeatures

The PowerlinkGeneralFeatures element has an empty content.

The PowerlinkGeneralFeatures element contains the attributes given in Table G.44.

Table G.44 — Attributes of element PowerlinkGeneralFeatures

Attribute	Data type	Use	Description
sdoCommunication	xsd:NMTOKENS	required	SDO communication role. Allowed values are: — sdo_client — sdo_server
sdoCommands	xsd:NMTOKENS	required	SDO commands. The allowed values are: — writeByIndex — readByIndex — writeAllByIndex — readAllByIndex — writeByName — readByName — fileWrite — fileRead — writeMultipleParam — readMultipleParam — linkName
sdoClientTimeout	xsd:double	optional	SDO client timeout. value provided in [ns]. In case the attribute is not given the default value is 1 000 000.
sdoServerTimeout	xsd:double	optional	SDO server timeout. value provided in [ns]. In case the attribute is not given the default value is 1 000 000.
sdoMaxParallelConnections	xsd:unsignedInt	required	Maximum number of parallel SDO connections. The minimal value is 1.
pdoMapGranularity	xsd:unsignedByte	optional	PDO mapping Granularity. value provided in [Byte]. In case the attribute is not given the default value is 8.
emergencyStatusEntries	xsd:unsignedByte	required	Number of Errors/Event entries signalled by a Node via network. The minimal value is 2 and the maximal value is 14.
emergencyQueueSize	xsd:unsignedByte	optional	Number of Errors/Event entries in the Node's emergency queue. In case the attribute is not given the default is 0
DCFsupport	xsd:NMTOKENS	optional	Managing node's capability to support Device Configuration File (DCF) handling. The allowed values are — supportConciseDCF — supportDCF — noDCFsupport
routingCapability	xsd:boolean	optional	Capability to perform routing functions. In case the attribute is not given the default is false
multiplexedCycles	xsd:boolean	optional	Capability to perform multiplexed access mode. In case the attribute is not given the default value is true.

G.5.4.3 PowerlinkMNFeatures

The PowerlinkMNFeatures element has an empty content.

The PowerlinkMNFeatures element contains the attributes given in Table G.45.

Table G.45 — Attributes of element PowerlinkMNFeatures

Attribute	Data type	Use	Description
bootupMode	xsd:NMTOKEN	optional	System bootup methodology. The allowed values are — bootup_simple System boot without CN verification and delayed boot of optional CNs — bootup_individual system boot with CN verification and delayed boot of optional CNs In case the attribute is not given the default value is "bootup_simple".
configurationManager	xsd:boolean	optional	Ability to perform Configuration Manager functions. In case the attribute is not given the default value is false.
sendPRes	xsd:boolean	optional	Managing node's ability to transmit multicast PRes frames. In case the attribute is not given the default value is false.
pdoMaxTxChannels	xsd:unsignedByte	required	Maximum number of TX PDO channels. The default value is 1.
pdoMaxRxChannels	xsd:unsignedByte	required	Maximum number of RX PDO channels. The default value is 1.
sdoTransferTypePdo	xsd:boolean	optional	Ability to transmit SDO data integrated in PDO frames. In case the attribute is not given the default value is false

G.5.4.4 PowerlinkCNFeatures

The PowerlinkCNFeatures element has an empty content.

The PowerlinkCNFeatures element contains the attributes given in Table G.46.

Table G.46 — Attributes of element PowerlinkCNFeatures

Attribute	Data type	Use	Description
basicEthernetMode	xsd:boolean	optional	Support of Basic Ethernet Mode. In case the attribute is not given the default value is true
isoCommFeature	xsd:boolean	required	Support of isochronous network traffic. The default value is true.
sdoTransferType	xsd:NMTOKENS	required	Supported SDO transfer methodology. The allowed values are — sdo_udp SDO hosted by UDP/IP frames — sdo_asend SDO hosted by EPL ASnd frames — sdo_pdo SDO hosted by data container integrated in PDO
pdoMaxRxChannels	xsd:unsignedByte	optional	Maximum number of RX PDO channels In case the attribute is not given the default value is 0

G.5.4.5 Element deviceCommissioning

The deviceCommissioning element has an empty content. The deviceCommissioning element contains the attributes given in Table G.47.

Table G.47 — Attributes of element deviceCommissioning

Attribute	Data type	Use	Description
nodeID	xsd:unsignedByte	required	Node ID (address). Valid node ids for a CN are 1 to 239 and 253 to 254. The valid node id for an MN is 240.
nodeType	xsd:NMTOKENS	required	Ability to perform MN resp. CN functions. The allowed values are: — CN — MN — redundantMN Any combination of these values is allowed.
pdoMaxMapBytes	xsd:unsignedInt	optional	Maximum PDO mapping size. The value is provided in [Byte]. The minimal value is 0 and the maximal value is 1489. In case the attribute is not given the default value is 1489.
pdoMaxMapObjects	xsd:byte	optional	Maximum number of map-able objects per PDO channel. In case the attribute is not given the default value is 255.
maxSupportedNodes	xsd:unsignedByte	optional	Maximum number of nodes in the Powerlink network segment. In case the attribute is not given the default value is 240.
usedNetworkInterface	xsd:unsignedByte	optional	Number of active network interfaces. In case the attribute is not given the default value is 0.
maxHeartbeatNodes	xsd:unsignedByte	optional	Number of nodes guardable via heartbeat. In case the attribute is not given, the default value is 240. This parameter is only valid if the device supports the heartbeating mechanism at all.
cycleTimingMinSupport	xsd:double	required	Minimum cycle time. The value is provided in [µs]. Default is 0
cycleTimingMaxSupport	xsd:double	required	Maximum cycle time The value is provided in [µs]. All values greater or equal than cycleTimingMinSupport are allowed.
cycleTimingGranularity	xsd:double	optional	Cycle time granularity. The value is provided in [µs]. In case the attribute is not given the default value is 1 (µs).
maxDomainSize	xsd:double	required	No default value.

G.5.5 EPL communication network profile template schema

The XML schema ProfileBody_CommunicationNetwork_EPL.xsd includes the schema ISO15745ProfileContainer.xsd in G.4.6.1 and the schema CommonElements.xsd in G.4.6.2.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <xsd:include schemaLocation="ISO15745ProfileContainer.xsd"/>
   <xsd:include schemaLocation="CommonElements.xsd"/>
   <!-- profile body -->
   <xsd:complexType name="ProfileBody_CommunicationNetwork_EPL">
      <xsd:complexContent>
         <xsd:extension base="ProfileBody_DataType">
            <xsd:choice>
               <xsd:sequence>
                  <xsd:element name="ApplicationLayers">
                     <xsd:complexType>
                        <xsd:sequence>
                           <xsd:element ref="CANopenObjectList"/>
                           <xsd:element name="identity" minOccurs="0">
                              <xsd:complexType>
                                 <xsd:sequence>
                                    <xsd:element ref="vendorID" minOccurs="0"/>
                                    <xsd:element ref="deviceFamily" minOccurs="0"/>
                                    <xsd:element ref="productID" minOccurs="0"/>
                                    <xsd:element ref="version" minOccurs="0" maxOccurs="unbounded"/>
                                    <xsd:element ref="buildDate" minOccurs="0"/>
                                    <xsd:element ref="specificationRevision" minOccurs="0"/>
                                 </xsd:sequence>
                              </xsd:complexType>
                           </xsd:element>
                           <xsd:element name="dummyUsage" minOccurs="0">
                              <xsd:complexType>
                                 <xsd:sequence>
                                    <xsd:element name="dummy" maxOccurs="unbounded">
                                       <xsd:complexType>
                                          <xsd:attribute name="entry" use="required">
                                             <xsd:simpleType>
                                                <xsd:restriction base="xsd:string">
                                                   <xsd:enumeration value="Dummy0001=0"/>
                                                   <xsd:enumeration value="Dummy0002=0"/>
                                                   <xsd:enumeration value="Dummy0003=0"/>
                                                   <xsd:enumeration value="Dummy0004=0"/>
                                                   <xsd:enumeration value="Dummy0005=0"/>
                                                   <xsd:enumeration value="Dummy0006=0"/>
                                                   <xsd:enumeration value="Dummy0007=0"/>
                                                   <xsd:enumeration value="Dummy0001=1"/>
                                                   <xsd:enumeration value="Dummy0002=1"/>
                                                   <xsd:enumeration value="Dummy0003=1"/>
                                                   <xsd:enumeration value="Dummy0004=1"/>
                                                   <xsd:enumeration value="Dummy0005=1"/>
                                                   <xsd:enumeration value="Dummy0006=1"/>
                                                   <xsd:enumeration value="Dummy0007=1"/>
                                                </xsd:restriction>
                                             </xsd:simpleType>
                                          </xsd:attribute>
                                       </xsd:complexType>
                                    </xsd:element>
                                 </xsd:sequence>
                              </xsd:complexType>
                           </xsd:element>
                           <xsd:element name="dynamicChannels" minOccurs="0">
                              <xsd:complexType>
                                 <xsd:sequence>
                                    <xsd:element name="dynamicChannel" maxOccurs="unbounded">
                                       <xsd:complexType>
                                          <xsd:sequence>
                                             <xsd:group ref="g_simple"/>
                                          </xsd:sequence>
                                          <xsd:attribute name="accessType" use="required">
                                             <xsd:simpleType>
                                                <xsd:restriction base="xsd:NMTOKEN">
                                                   <xsd:enumeration value="readOnly"/>
                                                   <xsd:enumeration value="writeOnly"/>
                                                   <xsd:enumeration value="readWriteWrite"/>
                                                </xsd:restriction>
                                             </xsd:simpleType>
                                          </xsd:attribute>
                                          <xsd:attribute name="startIndex" type="xsd:hexBinary"
use="required"/>
```

```
                                        <xsd:attribute name="endIndex" type="xsd:hexBinary"
use="required"/>
                                        <xsd:attribute name="maxNumber" type="xsd:unsignedInt"
use="required"/>
                                        <xsd:attribute name="addressOffset" type="xsd:hexBinary"
use="required"/>
                                        <xsd:attribute name="bitAlignment"
type="xsd:unsignedByte" use="optional"/>
                                        <xsd:attribute name="manufacturerSpecific"
type="xsd:string" use="optional"/>
                                     </xsd:complexType>
                                  </xsd:element>
                               </xsd:sequence>
                            </xsd:complexType>
                         </xsd:element>
                      </xsd:sequence>
                   </xsd:complexType>
                </xsd:element>
                <xsd:element name="TransportLayers">
                   <xsd:complexType/>
                </xsd:element>
                <xsd:element name="NetworkManagement" minOccurs="0">
                   <xsd:complexType>
                      <xsd:sequence>
                         <xsd:element name="PowerlinkGeneralFeatures">
                            <xsd:complexType>
                               <xsd:attribute name="sdoCommunication" use="required">
                                  <xsd:simpleType>
                                     <xsd:restriction base="xsd:NMTOKENS">
                                        <xsd:enumeration value="sdo_client"/>
                                        <xsd:enumeration value="sdo_server"/>
                                     </xsd:restriction>
                                  </xsd:simpleType>
                               </xsd:attribute>
                               <xsd:attribute name="sdoCommands" use="required">
                                  <xsd:simpleType>
                                     <xsd:restriction base="xsd:NMTOKENS">
                                        <xsd:enumeration value="writeByIndex"/>
                                        <xsd:enumeration value="readByIndex"/>
                                        <xsd:enumeration value="writeAllByIndex"/>
                                        <xsd:enumeration value="readAllByIndex"/>
                                        <xsd:enumeration value="writeByName"/>
                                        <xsd:enumeration value="readByName"/>
                                        <xsd:enumeration value="fileWrite"/>
                                        <xsd:enumeration value="fileRead"/>
                                        <xsd:enumeration value="writeMultipleParam"/>
                                        <xsd:enumeration value="readMultipleParam"/>
                                        <xsd:enumeration value="linkName"/>
                                     </xsd:restriction>
                                  </xsd:simpleType>
                               </xsd:attribute>
                               <xsd:attribute name="sdoClientTimeout" type="xsd:double"
use="optional" default="1000000"/>
                               <xsd:attribute name="sdoServerTimeout" type="xsd:double"
use="optional" default="1000000"/>
                               <xsd:attribute name="sdoMaxParallelConnections" use="required">
                                  <xsd:simpleType>
                                     <xsd:restriction base="xsd:unsignedInt">
                                        <xsd:minInclusive value="1"/>
                                     </xsd:restriction>
                                  </xsd:simpleType>
                               </xsd:attribute>
                               <xsd:attribute name="pdoMapGranularity" type="xsd:unsignedByte"
use="optional" default="8"/>
                               <xsd:attribute name="parameterStorage" type="xsd:boolean"
use="optional" default="false"/>
                               <xsd:attribute name="emergencyStatusEntries" default="2">
                                  <xsd:simpleType>
                                     <xsd:restriction base="xsd:unsignedByte">
                                        <xsd:minInclusive value="2"/>
                                        <xsd:maxInclusive value="14"/>
                                     </xsd:restriction>
                                  </xsd:simpleType>
                               </xsd:attribute>
                               <xsd:attribute name="emergencyQueueSize" type="xsd:unsignedByte"
use="optional"/>
```

```
                                    <xsd:attribute name="DCFsupport" use="optional"
default="supportConciseDCF">
                                       <xsd:simpleType>
                                          <xsd:restriction base="xsd:NMTOKENS">
                                             <xsd:enumeration value="supportConciseDCF"/>
                                             <xsd:enumeration value="supportDCF"/>
                                          </xsd:restriction>
                                       </xsd:simpleType>
                                    </xsd:attribute>
                                    <xsd:attribute name="routingCapability" type="xsd:boolean"
use="optional" default="false"/>
                                    <xsd:attribute name="multiplexedCycles" type="xsd:boolean"
use="optional" default="true"/>
                                 </xsd:complexType>
                              </xsd:element>
                              <xsd:element name="PowerlinkMNFeatures" minOccurs="0">
                                 <xsd:complexType>
                                    <xsd:attribute name="bootupMode" use="optional"
default="bootup_simple">
                                       <xsd:simpleType>
                                          <xsd:restriction base="xsd:NMTOKEN">
                                             <xsd:enumeration value="bootup_simple"/>
                                             <xsd:enumeration value="bootup_individual"/>
                                          </xsd:restriction>
                                       </xsd:simpleType>
                                    </xsd:attribute>
                                    <xsd:attribute name="configurationManager" type="xsd:boolean"
use="optional" default="false"/>
                                    <xsd:attribute name="sendPRes" type="xsd:boolean" use="optional"
default="false"/>
                                    <xsd:attribute name="pdoMaxTxChannels" type="xsd:unsignedByte"
use="required"/>
                                    <xsd:attribute name="pdoMaxRxChannels" use="required">
                                       <xsd:simpleType>
                                          <xsd:restriction base="xsd:unsignedByte">
                                             <xsd:minInclusive value="1"/>
                                             <xsd:maxInclusive value="252"/>
                                          </xsd:restriction>
                                       </xsd:simpleType>
                                    </xsd:attribute>
                                    <xsd:attribute name="sdoTransferTypePDO" type="xsd:boolean"
use="optional" default="false"/>
                                 </xsd:complexType>
                              </xsd:element>
                              <xsd:element name="PowerlinkCNFeatures" minOccurs="0">
                                 <xsd:complexType>
                                    <xsd:attribute name="basicEthernetMode" type="xsd:boolean"
use="optional" default="true"/>
                                    <xsd:attribute name="isoCommFeature" type="xsd:boolean"
use="required"/>
                                    <xsd:attribute name="sdoTransferType" use="required">
                                       <xsd:simpleType>
                                          <xsd:restriction base="xsd:NMTOKENS">
                                             <xsd:enumeration value="sdo_udp"/>
                                             <xsd:enumeration value="sdo_asend"/>
                                             <xsd:enumeration value="sdo_pdo"/>
                                          </xsd:restriction>
                                       </xsd:simpleType>
                                    </xsd:attribute>
                                    <xsd:attribute name="pdoMaxRxChannels" use="optional">
                                       <xsd:simpleType>
                                          <xsd:restriction base="xsd:unsignedByte">
                                             <xsd:minExclusive value="0"/>
                                             <xsd:maxExclusive value="253"/>
                                          </xsd:restriction>
                                       </xsd:simpleType>
                                    </xsd:attribute>
                                 </xsd:complexType>
                              </xsd:element>
                              <xsd:element name="deviceCommissioning">
                                 <xsd:complexType>
                                    <xsd:attribute name="nodeID" type="xsd:unsignedByte"
use="required"/>
                                    <xsd:attribute name="nodeType" type="xsd:NMTOKENS" use="required"/>
                                    <xsd:attribute name="pdoMaxMapBytes" use="optional" default="1489">
                                       <xsd:simpleType>
```

```
                                    <xsd:restriction base="xsd:unsignedInt">
                                       <xsd:minInclusive value="0"/>
                                       <xsd:maxInclusive value="1489"/>
                                    </xsd:restriction>
                                 </xsd:simpleType>
                              </xsd:attribute>
                              <xsd:attribute name="pdoMaxMapObjects" type="xsd:unsignedByte"
use="optional" default="255"/>
                              <xsd:attribute name="maxSupportedNodes" type="xsd:unsignedByte"
use="optional" default="240"/>
                              <xsd:attribute name="usedNetworkInterface" type="xsd:unsignedByte"
use="optional" default="0"/>
                              <xsd:attribute name="maxHeartbeatNodes" type="xsd:unsignedByte"
use="optional" default="240"/>
                              <xsd:attribute name="cycleTimingMinSupport" type="xsd:double"
use="required"/>
                              <xsd:attribute name="cycleTimingMaxSupport" type="xsd:double"
use="required"/>
                              <xsd:attribute name="cycleTimingGranularity" type="xsd:double"
use="optional" default="1"/>
                              <xsd:attribute name="maxDomainSize" type="xsd:double"
use="required"/>
                           </xsd:complexType>
                        </xsd:element>
                     </xsd:sequence>
                  </xsd:complexType>
               </xsd:element>
            </xsd:sequence>
            <xsd:element name="ExternalProfileHandle" type="ProfileHandle_DataType"/>
         </xsd:choice>
         <xsd:attributeGroup ref="ag_formatAndFile"/>
         <xsd:attribute name="supportedLanguages" use="optional">
            <xsd:simpleType>
               <xsd:restriction base="xsd:NMTOKENS"/>
            </xsd:simpleType>
         </xsd:attribute>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
<!-- CANopen object dictionary-->
<xsd:element name="CANopenObjectList">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="CANopenObject" maxOccurs="65535">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="CANopenSubObject" minOccurs="0" maxOccurs="255">
                     <xsd:complexType>
                        <xsd:attribute name="subIndex" type="xsd:unsignedByte" use="required"/>
                        <xsd:attribute name="name" type="xsd:string" use="required"/>
                        <xsd:attribute name="objectType" type="xsd:unsignedByte" use="required"/>
                        <xsd:attribute name="dataType" type="xsd:hexBinary" use="optional"/>
                        <xsd:attribute name="lowLimit" type="xsd:string" use="optional"/>
                        <xsd:attribute name="highLimit" type="xsd:string" use="optional"/>
                        <xsd:attribute name="accessType" use="optional">
                           <xsd:simpleType>
                              <xsd:restriction base="xsd:string">
                                 <xsd:enumeration value="ro"/>
                                 <xsd:enumeration value="wo"/>
                                 <xsd:enumeration value="rw"/>
                                 <xsd:enumeration value="rwr"/>
                                 <xsd:enumeration value="rww"/>
                                 <xsd:enumeration value="const"/>
                              </xsd:restriction>
                           </xsd:simpleType>
                        </xsd:attribute>
                        <xsd:attribute name="defaultValue" type="xsd:string" use="optional"/>
                        <xsd:attribute name="actualValue" type="xsd:string" use="optional"/>
                        <xsd:attribute name="denotation" type="xsd:string" use="optional"/>
                        <xsd:attribute name="PDOmapping" type="xsd:boolean" use="optional"/>
                        <xsd:attribute name="objFlags" type="xsd:unsignedInt" use="optional"/>
                        <xsd:attribute name="uniqueIDRef" type="xsd:IDREF" use="optional"/>
                     </xsd:complexType>
                  </xsd:element>
               </xsd:sequence>
               <xsd:attribute name="index" type="xsd:hexBinary" use="required"/>
```

```
                  <xsd:attribute name="name" type="xsd:string" use="required"/>
                  <xsd:attribute name="objectType" type="xsd:unsignedByte" use="required"/>
                  <xsd:attribute name="dataType" type="xsd:hexBinary" use="optional"/>
                  <xsd:attribute name="lowLimit" type="xsd:string" use="optional"/>
                  <xsd:attribute name="highLimit" type="xsd:string" use="optional"/>
                  <xsd:attribute name="accessType" use="optional">
                     <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                           <xsd:enumeration value="ro"/>
                           <xsd:enumeration value="wo"/>
                           <xsd:enumeration value="rw"/>
                           <xsd:enumeration value="rwr"/>
                           <xsd:enumeration value="rww"/>
                           <xsd:enumeration value="const"/>
                        </xsd:restriction>
                     </xsd:simpleType>
                  </xsd:attribute>
                  <xsd:attribute name="defaultValue" type="xsd:string" use="optional"/>
                  <xsd:attribute name="actualValue" type="xsd:string" use="optional"/>
                  <xsd:attribute name="denotation" type="xsd:string" use="optional"/>
                  <xsd:attribute name="PDOmapping" type="xsd:boolean" use="optional"/>
                  <xsd:attribute name="objFlags" type="xsd:hexBinary" use="optional"/>
                  <xsd:attribute name="uniqueIDRef" type="xsd:IDREF" use="optional"/>
                  <xsd:attribute name="subNumber" type="xsd:unsignedByte" use="optional"/>
               </xsd:complexType>
            </xsd:element>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
</xsd:schema>
```

Page 125, Bibliography

Add the following to the list of bibliographic references after References [6] to [9] inserted from Amendment 1 to ISO 15745-4:2003:

"[10] EN 50325-4 *Industrial communications subsystem based on ISO 11898 (CAN) for controller-device interfaces, Part 4: CANopen"*

ICS 25.040.40

Price based on 168 pages