

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Engineering data exchange format for use in industrial automation systems
engineering – Automation markup language –
Part 5: Communication**

**Format d'échange de données techniques pour une utilisation dans l'ingénierie
des systèmes d'automatisation industrielle – Automation markup language –
Partie 5: Communication**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2022 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

IEC Secretariat
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee, ...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

IEC Products & Services Portal - products.iec.ch

Discover our powerful search engine and read freely all the publications previews. With a subscription you will always have access to up to date content tailored to your needs.

Electropedia - www.electropedia.org

The world's leading online dictionary on electrotechnology, containing more than 22 300 terminological entries in English and French, with equivalent terms in 19 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC

Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Recherche de publications IEC - webstore.iec.ch/advsearchform

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études, ...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et une fois par mois par email.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: sales@iec.ch.

IEC Products & Services Portal - products.iec.ch

Découvrez notre puissant moteur de recherche et consultez gratuitement tous les aperçus des publications. Avec un abonnement, vous aurez toujours accès à un contenu à jour adapté à vos besoins.

Electropedia - www.electropedia.org

Le premier dictionnaire d'électrotechnologie en ligne au monde, avec plus de 22 300 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 19 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Engineering data exchange format for use in industrial automation systems
engineering – Automation markup language –
Part 5: Communication**

**Format d'échange de données techniques pour une utilisation dans l'ingénierie
des systèmes d'automatisation industrielle – Automation markup language –
Partie 5: Communication**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 25.040; 25.040.01

ISBN 978-2-8322-1085-7

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

CONTENTS

FOREWORD.....	6
1 Scope	8
2 Normative references	8
3 Terms, definitions, abbreviated terms and acronyms	9
3.1 Terms and definitions.....	9
3.2 Abbreviated terms and acronyms	9
4 Use cases and network structures	10
4.1 General.....	10
4.2 Use cases	10
4.2.1 Engineering activities.....	10
4.2.2 Lossless transfer of communication device instance information.....	11
4.2.3 Lossless transfer of communication system information	14
4.3 Delimitation of modelling range	16
4.3.1 Scope of the modelling range	16
4.3.2 Interaction structures and life cycles	16
4.3.3 Network objects	17
4.3.4 Network topologies	18
4.3.5 Communication content	23
4.4 Derived modelling requirements.....	23
5 UML model	24
5.1 Overview.....	24
5.2 Logical topology.....	24
5.2.1 Aim of logical topology.....	24
5.2.2 Item logicalTopology	25
5.2.3 Item logicalConnection	25
5.2.4 Item logicalEndPoint	25
5.3 Physical topology.....	26
5.3.1 Aim of physical topology	26
5.3.2 Item physicalTopology	26
5.3.3 Item physicalConnection	26
5.3.4 Item physicalEndPoint	27
5.4 Device	27
5.4.1 General	27
5.4.2 Item physicalDevice	27
5.4.3 Item Information	28
5.4.4 Item physicalDeviceInformation	29
5.4.5 Item logicalDeviceInformation	29
5.4.6 Item logicalDevice	29
5.4.7 Item networkDataList	29
5.4.8 Item networkDataItem	29
5.4.9 Item logicalEndPointList	29
5.4.10 Item physicalEndPointList.....	29
5.4.11 Item physicalChannelList.....	29
5.4.12 Item physicalChannel	30
5.4.13 Item deviceResource	30
5.4.14 Item variableList	30
5.4.15 Item variable.....	30

5.4.16	Item pduList.....	31
5.4.17	Item pdu	31
5.4.18	Item protocolData	32
5.4.19	Item payload.....	32
5.4.20	Item processDataItemList	32
5.4.21	Item parameterItemList	32
5.4.22	Item dataItem	32
5.4.23	Item processDataItem	32
5.4.24	Item processDataInput.....	33
5.4.25	Item processDataOutput	33
5.4.26	Item parameterItem	33
6	Representation within AutomationML.....	33
6.1	Overview of mapping	33
6.1.1	Introduction of mapping	33
6.1.2	General mapping rules.....	33
6.1.3	Basics	34
6.1.4	Modelling of relations.....	35
6.1.5	Application process	36
6.2	Basic communication role class library	38
6.2.1	General	38
6.2.2	RoleClass PhysicalDevice	39
6.2.3	RoleClass PhysicalEndpointlist.....	40
6.2.4	RoleClass PhysicalConnection	40
6.2.5	RoleClass PhysicalNetwork	40
6.2.6	RoleClass LogicalDevice	41
6.2.7	RoleClass LogicalEndpointlist.....	41
6.2.8	RoleClass LogicalConnection	41
6.2.9	RoleClass LogicalNetwork	41
6.3	Basic communication interface class library	42
6.3.1	General	42
6.3.2	InterfaceClass PhysicalEndPoint	42
6.3.3	InterfaceClass LogicalEndPoint	42
6.4	Steps to model technology specific libraries.....	43
6.4.1	General	43
6.4.2	Step 1: Development of technology specific role classes	43
6.4.3	Step 2: Development of technology specific interface classes.....	44
6.4.4	Step 3: Development of system unit class libraries	44
6.4.5	Step 4: Modelling the network.....	45
6.4.6	Step 5: Modelling the connections	46
6.5	PDU modelling.....	46
6.5.1	General	46
6.5.2	RoleClass CommunicationPackage.....	47
6.5.3	InterfaceClass DatagramObject	48
6.5.4	Steps to model technology specific libraries.....	49
6.6	References to attributes	51
6.7	Usage of metadata.....	53
	Bibliography.....	55

Figure 1 – General engineering activities communication system engineering is embedded within.....	10
Figure 2 – Information flow of the use case.....	12
Figure 3 – Alternative information flow of the use case	13
Figure 4 – Information flow of the use case.....	15
Figure 5 – Example of a logical level view on communication systems.....	17
Figure 6 – Example of a physical level view on communication systems	18
Figure 7 – Combined views on communication systems	18
Figure 8 – Star topology example.....	19
Figure 9 – Ring topology example.....	19
Figure 10 – Line topology example	20
Figure 11 – Simple network with direct wiring	20
Figure 12 – Network with active infrastructure.....	21
Figure 13 – Networks connected by gateways.....	21
Figure 14 – Hierarchical structured networks	22
Figure 15 – Network covering multiple applications.....	22
Figure 16 – General modelling strategy for PDUs.....	23
Figure 17 – Structure of communication network.....	24
Figure 18 – View on logical topology.....	25
Figure 19 – View on physical topology	26
Figure 20 – Part 1 of the device model.....	28
Figure 21 – Part 2 of the device model.....	31
Figure 22 – Communication role class library and communication interface class library	35
Figure 23 – Derived role class libraries and interface class libraries for a special example.....	35
Figure 24 – SystemUnitClassLib examples for communication system modelling	37
Figure 25 – Final network model example	38
Figure 26 – Basic communication role class library	39
Figure 27 – CommunicationRoleClassLib.....	39
Figure 28 – XML text of the communication role class library	39
Figure 29 – Basic communication interface class library	42
Figure 30 – CommunicationInterfaceClassLib	42
Figure 31 – XML text of the communication interface class library	42
Figure 32 – Derivation of a technology specific role class library out of the base role class library	43
Figure 33 – Derivation of a technology specific role class library out of the base role class library	44
Figure 34 – Technology specific <SystemUnitClassLib>s.....	45
Figure 35 – Technology specific communication network	46
Figure 36 – Extended communication role class library	47
Figure 37 – Extended CommunicationRoleClassLib	47
Figure 38 – XML text of the extended communication role class library.....	47
Figure 39 – Extended communication interface class library	48
Figure 40 – Extended CommunicationInterfaceClassLib.....	48

Figure 41 – XML text of the extended communication role class library.....	48
Figure 42 – Derivation of a technology specific role class library out of the extended role class library	49
Figure 43 – Derivation of a technology specific interface class library out of the extended interface class library.....	50
Figure 44 – Technology specific extended <SystemUnitClassLib>s.....	50
Figure 45 – Technology specific communication network with communication package models.....	51
Figure 46 – Field SourceDocumentInformation according to communication related libraries	54
Table 1 – Mapping rules	34
Table 2 – Modelling of relations in AutomationML	36
Table 3 – RoleClass PhysicalDevice	40
Table 4 – RoleClass PhysicalEndpointlist	40
Table 5 – RoleClass PhysicalConnection	40
Table 6 – RoleClass PhysicalNetwork	40
Table 7 – RoleClass LogicalDevice	41
Table 8 – RoleClass LogicalEndpointlist	41
Table 9 – RoleClass LogicalConnection	41
Table 10 – RoleClass LogicalNetwork	41
Table 11 – InterfaceClass PhysicalEndPoint	42
Table 12 – InterfaceClass LogicalEndPoint	43
Table 13 – RoleClass CommunicationPackage	48
Table 14 – InterfaceClass DatagrammObject	49
Table 15 – Communication related attributes	52

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**ENGINEERING DATA EXCHANGE FORMAT FOR USE
IN INDUSTRIAL AUTOMATION SYSTEMS ENGINEERING –
AUTOMATION MARKUP LANGUAGE –****Part 5: Communication****FOREWORD**

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

IEC 62714-5 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation. It is an International Standard.

The text of this International Standard is based on the following documents:

Draft	Report on voting
65E/844/FDIS	65E/886/RVD

Full information on the voting for its approval can be found in the report on voting indicated in the above table.

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/standardsdev/publications.

A list of all parts of the IEC 62714 series, under the general title *Engineering data exchange format for use in industrial automation systems engineering – Automation markup language*, can be found on the IEC website.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IMPORTANT – The "colour inside" logo on the cover page of this document indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

ENGINEERING DATA EXCHANGE FORMAT FOR USE IN INDUSTRIAL AUTOMATION SYSTEMS ENGINEERING – AUTOMATION MARKUP LANGUAGE –

Part 5: Communication

1 Scope

Engineering processes of technical systems and their embedded automation systems are executed with increasing efficiency and quality. Especially since the project duration tends to increase as the complexity of the engineered system increases. To solve this problem, the engineering process is more often being executed by exploiting software based engineering tools exchanging engineering information and artefacts along the engineering process related tool chain.

Communication systems establish an important part of modern technical systems and, especially, of automation systems embedded within them. Following the increasing decentralisation of automation systems and the application of fieldbus and Ethernet technology connecting automation devices and further interacting entities need to fulfil special requirements on communication quality, safety and security. Thus, within the engineering process of modern technical systems, engineering information and artefacts relating to communication systems also need to be exchanged along the engineering process tool chain.

In each phase of the engineering process of technical systems, communication system related information can be created which can be consumed in later engineering phases. A typical application case is the creation of configuration information for communication components of automation devices including communication addresses and communication package structuring within controller programming devices during the control programming phase and its use in a device configuration tool. Another typical application case is the transmission of communication device configurations to virtual commissioning tools, to documentation tools, or to diagnosis tools.

At present, the consistent and lossless transfer of communication system engineering information along the complete engineering chain of technical systems is unsolved. While user organisations and companies have provided data exchange formats for parts of the relevant information like FDCML, EDDL, and GSD, the above named application cases cannot be covered by a data exchange format. Notably the networking related information describing communication relations and their properties and qualities cannot be modelled by a data exchange format.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61131-3, *Programmable controllers – Part 3: Programming languages*

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

IEC 62424:2016, *Representation of process control engineering – Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools*

IEC 62714-1, *Engineering data exchange format for use in industrial systems engineering – Automation Markup Language – Part 1: Architecture and general requirements*

IEC 62714-4, *Engineering data exchange format for use in industrial systems engineering – Automation markup language – Part 4: Logic*

IEC 81346 (all parts), *Industrial systems, installations and equipment and industrial products – Structuring principles and reference designations*

3 Terms, definitions, abbreviated terms and acronyms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62714-1 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at <http://www.electropedia.org/>
- ISO Online browsing platform: available at <http://www.iso.org/obp>

3.1.1

AutomationML/AML

XML based data exchange format for plant engineering data

3.1.2

Automation object

entity in an automated system

Note 1 to entry: An example of an automation object is an automation component, a valve, or a signal.

3.2 Abbreviated terms and acronyms

For the purposes of this document, the following abbreviations apply.

AML	Automation Markup Language
CAEX	Computer Aided Engineering Exchange as defined in IEC 62424:2016
ECAD	Computer aided engineering for electrical engineering
EDDL	Electronic Device Description Language
FDCML	Field Device Configuration Markup Language
GUID	Global Unique Identifier
GSD	General Station Description
HMI	Human Machine Interface
ID	Identifier
MCAD	Computer aided engineering for mechanical engineering
OPC	Open Platform Communications
PDU	Protocol Data Unit
SCADA	Supervisory Control And Data Acquisition
UML	Unified Modelling Language
UUID	Universal Unique Identifier
XML	Extensible Markup Language

4 Use cases and network structures

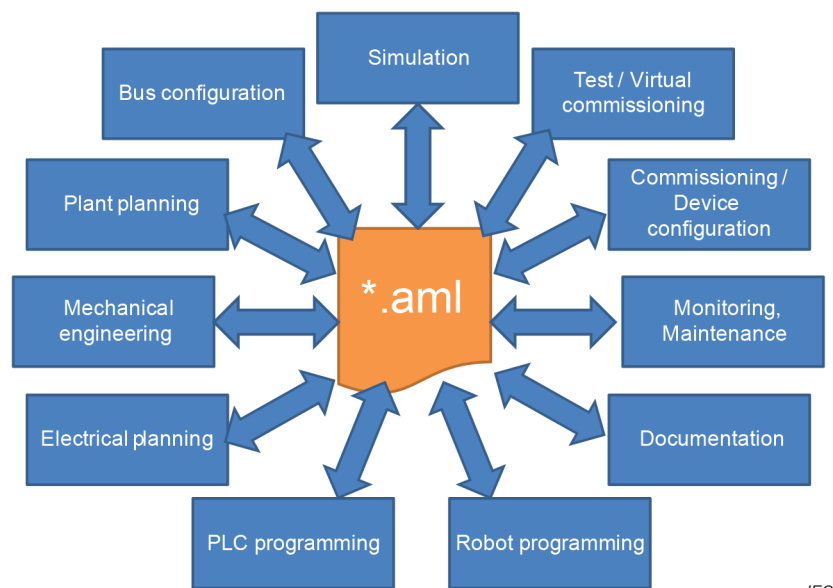
4.1 General

The modelling of communication systems based on AutomationML targets the modelling of a large amount of information created, exchanged, and applied within the engineering process of manufacturing systems. Nevertheless, not all possible communication system related information will be modelled. Within the following subclause, the use cases for the application of the communication system modelling as well as the relevant information sets within them are named.

4.2 Use cases

4.2.1 Engineering activities

Network topology and communication system related information is relevant to various engineering activities along the engineering chain of production systems. Within the engineering process of production systems, the communication system can be designed in the detailed engineering phase exploiting various tools. Thereby, communication system related information is created which subsequently can be applied within the detailed design of devices and the device commissioning. Figure 1 represents an example set of engineering activities relevant within the general engineering process of production systems and the communication system engineering embedded within.



IEC

Figure 1 – General engineering activities communication system engineering is embedded within

Within the named engineering activities, engineering tools like (but not limited to) the following will have a relevant impact:

- plant planning tools,
- mechanical engineering tools (MCAD),
- electrical engineering tools (ECAD),
- controller programming tools,
- robot programming tools,
- HMI programming tools,
- OPC system configuration tools,

- device configuration tools,
- bus configuration tools,
- simulation tools,
- SCADA systems,
- virtual commissioning tools,
- documentation tools,
- communication system security tools,
- communication system management tools,
- communication system diagnosis tools.

These tools will create and/or consume communication system related engineering information depending on the use case of the engineering chain.

Nevertheless, (among others) there are two main application cases within this engineering chain, where communication system related information can be exchanged between engineering tools. These two use cases are the main target of the modelling of communication systems based on AutomationML.

4.2.2 Lossless transfer of communication device instance information

Within the general engineering process, this use case covers the transition of communication relevant information for configuration of communication components of sensors and actuators from controller programming tool and similar tools to sensor and actuator configuration/programming tools. It contains the transition of information relevant for correct communication (like addresses and channels) as well as for correct structuring of communication data packages transmitted within communication (like transmitted data points of sensors). An overview is shown in Figure 2.

Within the related engineering activities, engineers with the engineering roles of controller programmer, HMI programmer, electrical design engineer, commissioner communication, commissioner controller, and robot programmer can be involved. They will execute the following sequence of engineering and data exchange activities which should be seen as an example sequence.

- Step 1. Design of system instrumentation, definition of used/interconnected devices
- Step 2. Export of device information from system instrumentation tool
- Step 3. Import of device information to controller programming and device configuration tools
- Step 4. Integration of device descriptions (like GSD) in controller programming tool
- Step 5. Design of controller programs and configurations within controller programming tool
- Step 6. Export of communication device relevant information from controller programming tool
- Step 7. Import of communication device relevant information to device configuration tool
- Step 8. Use of information for parameterisation of communication component of device (addresses, etc.) and for structuring of communication packages (send data points)

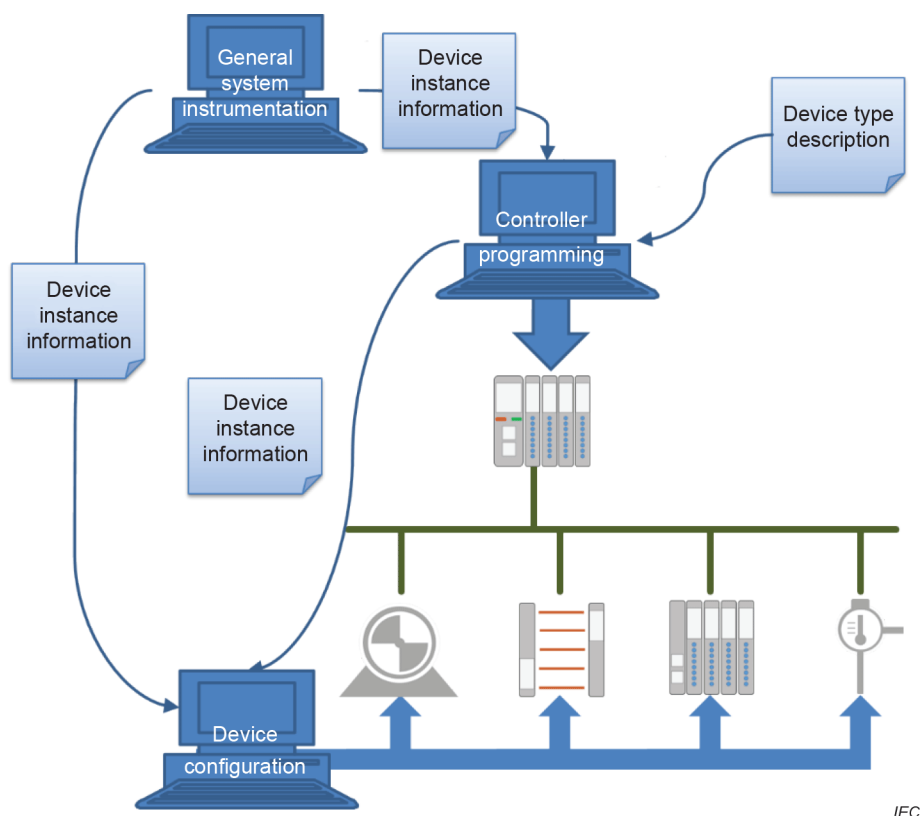


Figure 2 – Information flow of the use case

There are different alternatives possible but not usually applied. The following two sequences are imaginable which will be in the focus of the use case.

Sequence 1:

- Step 1. Design of communication system configurations within third party tool
- Step 2. Integration of device descriptions
- Step 3. Export of communication device relevant information
- Step 4. Import of communication device relevant information to device configuration/programming tool
- Step 5. Use of information for parameterization of communication component of device (addresses, etc.) and for structuring of communication packages (send data points)

Sequence 2:

- Step 1. Third party tool, e.g. the device configuration tool provides information like signals, data volume, describing the instance information
- Step 2. Device vendor provides device descriptions, describing the type information
- Step 3. Bus configuration tool consumes these snippets and device descriptions
- Step 4. Bus configurator generates bus configuration

Step 5. Import of the bus configuration into the controller programming tool

Step 6. Use of information from peripheral devices inside the controller program

Both sequences are commonly depicted in Figure 3.

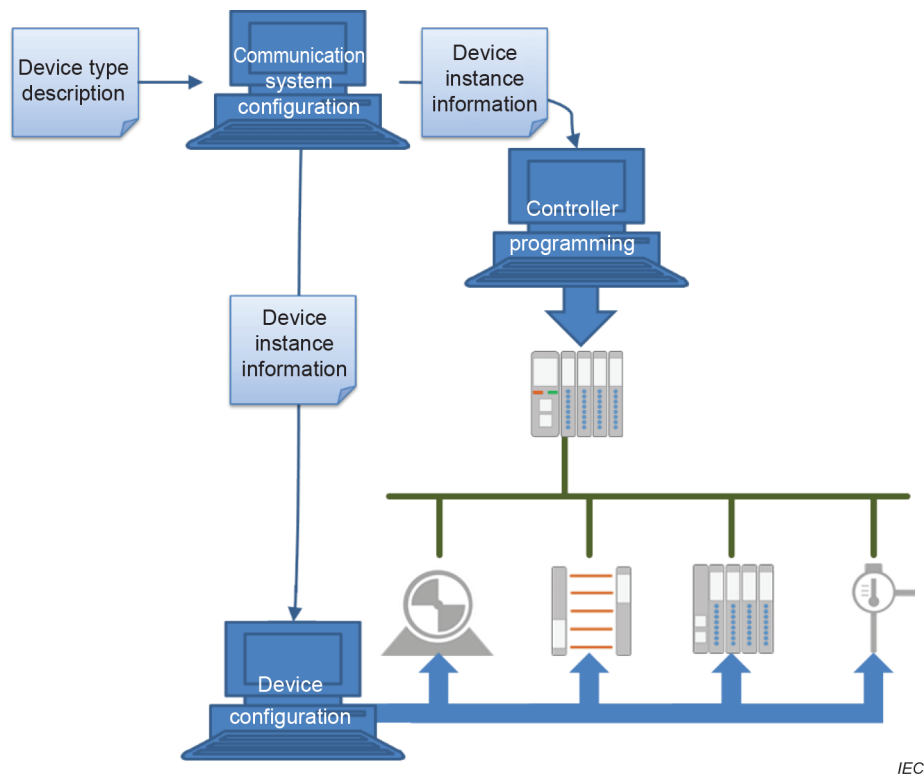


Figure 3 – Alternative information flow of the use case

Possible tools exporting communication system information can be e.g. controller programming tools. They cover controller programming projects with data points (variables), device configurations, and indirect communication network descriptions. In addition, communication system engineering tools can be data sources as well as instrumentation tools like ECAD tools.

Data sinks of the data exchange can be tools for sensor communication configuration, HMI programming, robot programming, OPC system configuration, or actor communication configuration. They mostly cover programming projects with data points (variables), device configurations, and indirect communication network descriptions. Relevant tools can be FDT tools, HMI programming tools, and robot programming tools.

Based on this use case, the modelling of communication systems based on AutomationML should cover information about IO lists, association of variables (IOs) to communication data packages and device parameters for parameterization of communication devices such as addresses (e.g. IP address), media access (e.g. MAC address), subnet masks, gateway addresses, and communication objects used, such as profile information.

In addition, the modelling should fulfil the following non-functional requirements. Device parameter list shall be extendable by users to cover upcoming technologies. Appropriate RoleClassLibs and/or SystemUnitClassLibs enabling the identification of object semantics shall be defined.

4.2.3 Lossless transfer of communication system information

Beyond the configuration of communication devices, communication system information is exploited in different engineering, monitoring, maintenance, etc. tools. Therefore, this use case covers the transmission of communication network configuration and structure information including infrastructure device configuration, end device configuration with respect to communication system parameters, network structure and wiring, quality of service, etc. This information can be provided to device configuration tools, documentation and maintenance tools, and network management tools. They can enable the combination of physical wiring with logical communication connections for error detection.

Within this use case, engineers with the engineering roles controller programmer, HMI programmer, electrical design engineer, commissioner communication, commissioner controller, robot programmer, and operator are involved. They will execute the following engineering process which should be seen as an example sequence.

- Step 1. Engineering of sets of variables to be transmitted within communication system within controller programming tools and sensor/actuator programming/configuration tools
- Step 2. Export of these variable sets from controller programming tools and sensor/actuator programming/configuration tools
- Step 3. Import of variable sets to communication network engineering tools
- Step 4. Engineering of mapping lists of variables and engineering of physical wiring
- Step 5. Export of communication system design from communication network engineering tools
- Step 6. Import of communication system design to continuing engineering or execution tools

The sequence is commonly depicted in Figure 4.

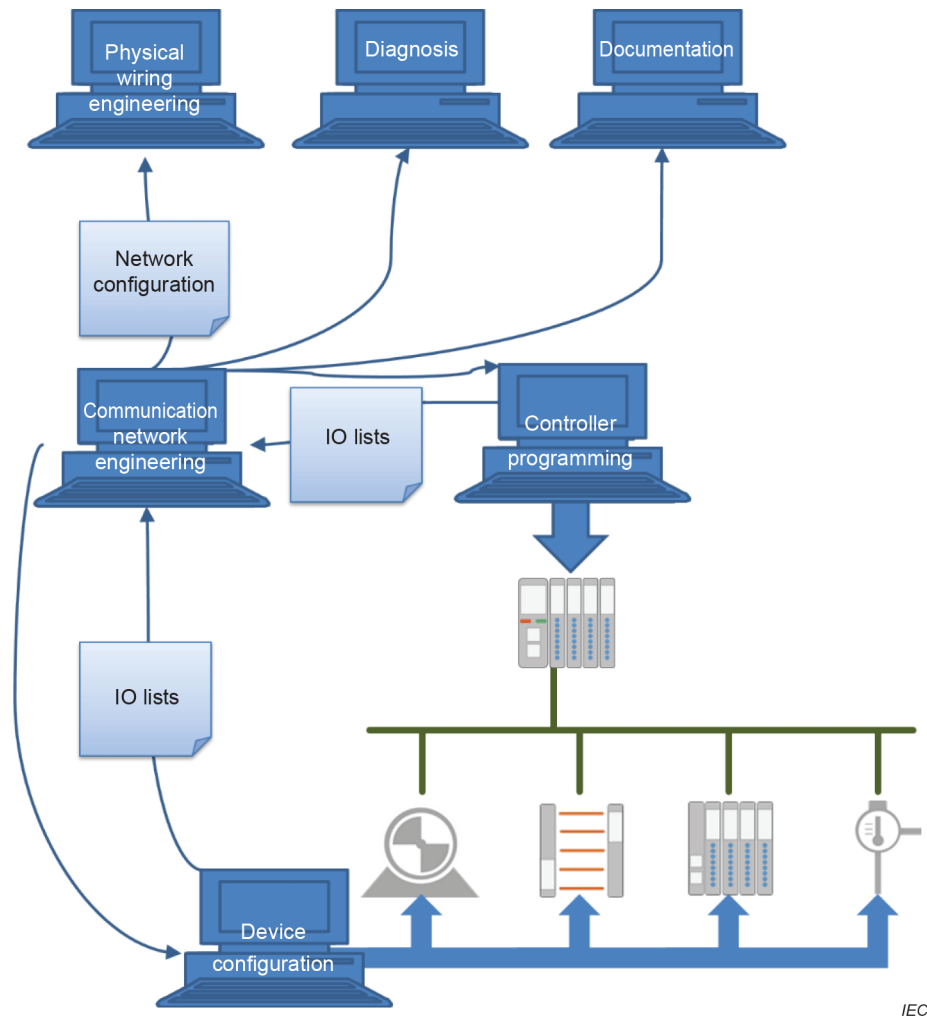


Figure 4 – Information flow of the use case

Source of the engineering data exchanged can be controller programming tools covering controller programming projects with data points (variables), device configurations, and indirect communication network descriptions, communication system engineering tools covering communication system projects including device configurations and communication connection properties, sensor communication configuration tools, HMI programming tools, robot programming tools, OPC system configuration tools, or actor communication configuration tools all together covering programming projects with data points (variables), device configurations, and indirect communication network descriptions.

Sinks of the data exchange can be tools for communication system security monitoring, communication system configuration, communication system management, and communication system diagnosis.

Based on this use case, the modelling of communication systems based on AutomationML shall cover information about network topology, connector lists of devices and their properties, physical connections between connectors of devices and their properties, logical links between device based applications and their properties, mapping of logical links to physical connections, association of link and/or connection attributes to links or connections like sender and receiver addresses, quality of service, protocols used, etc.

In addition, some non-functional requirements are derived. Link and connection attribute list shall be extendable by users to cover upcoming technologies. Appropriate RoleClassLibs and/or SystemUnitClassLibs enabling the identification of object semantics shall be defined.

4.3 Delimitation of modelling range

4.3.1 Scope of the modelling range

Within the scope of the use cases named above, different communication technologies, communication system use strategies and life cycles as well as communication network architectures can be considered. Within the following, the broad theoretical range will be limited to most applied cases.

4.3.2 Interaction structures and life cycles

Following the above described use cases, the communication system modelling based on AutomationML should concentrate on the description of cyclic communication based on a fixed assignment between communication partners. These communication partners will run (more or less complex) parts of the control logic of control applications on different communicating control devices.

Therefore, there are at least two different views relevant to the communication system; a logical view and a physical view. On a high level logical view, there are data processing applications exchanging application data (variables) logically. For the physical realisation of the data processing applications control devices are used. These devices will physically communicate using communication systems. Within both views, the modelled objects (application parts, devices, connections, etc.) will have special properties which are relevant for the correct use/behaviour of the communication system. Such systems and their descriptive information are in the scope of the following presentations.

Out of scope of the communication system modelling based on AutomationML are for example

- acyclic service based and only temporarily used communication structures as they are usually present in the use of the World Wide Web,
- safety or security related issues, and
- complex device behaviour and structure information as they are usually covered by device description languages.

Nevertheless, the use of AutomationML to model communication systems with these properties is not excluded although not intended.

Communication system descriptions can cover different levels of detail. They can range from very abstract descriptions only naming communicating partners and their dependencies down to a detailed description of communication connections and their properties as well as device and application parts with their properties. These different levels of detail are in the scope of the intended communication system modelling.

Communication systems are complex systems covering different technologies within one communication system. As an example, controller based control systems can be considered. They usually embed different fieldbus systems, Ethernet based communication, copper and fibre optic wiring, wireless communication, different active infrastructure, etc. Different communication protocols can operate on the same physical infrastructure (wire, wireless channel etc.). A single device may execute several control application parts. Thus, the different communication technologies are cross-linked and depend on each other. This complexity and interlinking should be expressible by AutomationML and is in scope of the following descriptions.

Communication systems are considered in different phases of the engineering of a production system such as detailed engineering and commissioning and are used within different phases of the life cycle of a production system like device configuration, production ramp up, normal production run, diagnosis and maintenance. These different phases of engineering and use of production systems and their impact on model content and model use should be covered by the AutomationML based modelling of communication systems.

4.3.3 Network objects

Each communication network will be seen as two layered: a logical level and a physical level. Mapping this view on the often referenced ISO OSI 7 layer structure of communication systems [2]¹, the physical level will cover the layers 1 and 2 of the ISO OSI 7 layer structure while the logical level subsumes layers 3 to 7 of the ISO OSI 7 layer structure and the control applications above layer 7.

Looking on the logical layer we can find the following entities to be modelled. Control applications usually consist of multiple distributed application parts providing different functionalities of the control process and forming logical devices. An example (given in Figure 5) can be a main control application exchanging sensor and actuator information with two IO functions. In general, these logical devices (control application parts) exchange information of different types, in our example different sensor signals and actuator states, which can be seen as connection points to the logical devices and end points of the information exchange between logical devices. The information exchange itself is executed by different logical connections.

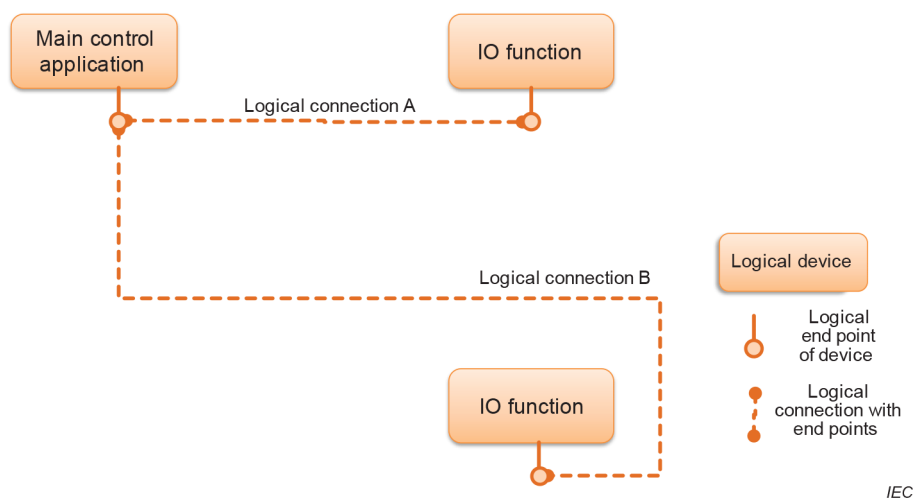


Figure 5 – Example of a logical level view on communication systems

The logical network may contain different objects with different descriptive properties. While logical control devices may have unique identifiers, cycle times, and foot print (to name only a few examples), logical end points may have a data type, and logical connections may have a required transmission rate. In any case, these descriptive properties can be considered as attributes of the objects of interest.

On the physical level, physical devices with physical interfaces can be found. In the example given in Figure 6, a controller is connected via an active infrastructure with two IO devices. Physical devices have physical endpoints representing network interfaces like plugs and sockets and are connected via physical connections to a communication system. Compared to the logical level view, there are additional physical entities representing infrastructure components of the network (like switches, etc.).

As in the case of the logical network, the physical network objects may have different descriptive properties. While physical devices may have identifiers and processing capabilities, physical end points may have an address, a maximal data rate and a connector type, whereas physical connections may have a possible transmission rate, a wire type, antenna type, plug type or a documentation number. These descriptive properties can be considered as attributes of the objects of interest.

¹ Figures in square brackets refer to the Bibliography.

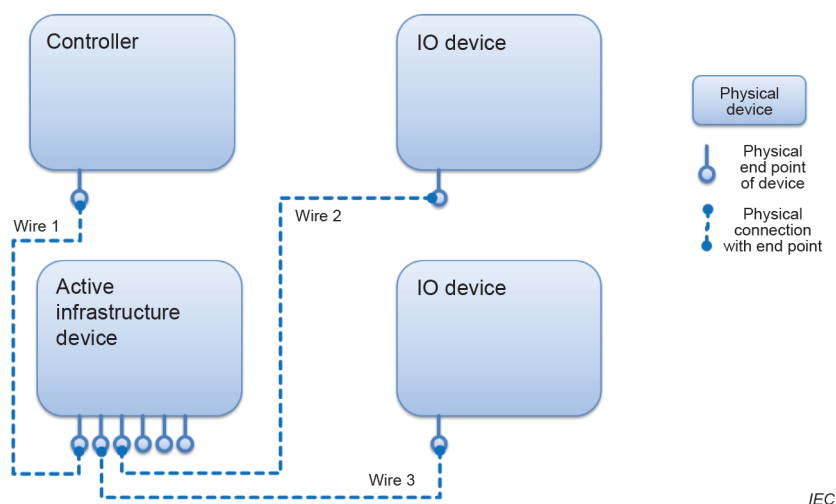


Figure 6 – Example of a physical level view on communication systems

Both views need to be combined to get the complete communication system description. Therefore, logical devices are hosted by physical devices interconnected in a physical network topology. In the example given in Figure 7, a controller hosts a main control application while the IO devices host IO functions. In addition, physical interfaces and logical interfaces are mapped. Thus, each logical connection is mapped virtually to a set of physical connections implementing it. It is not strictly necessary that there is a unique chain of physical connections representing this implementation since not all communication systems support this.

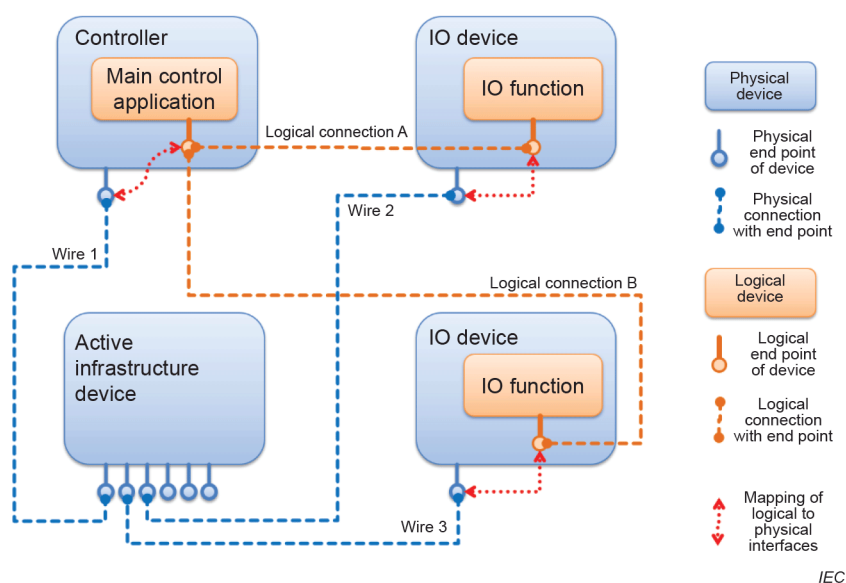


Figure 7 – Combined views on communication systems

4.3.4 Network topologies

Within industrial communication systems, different network architectures are to be considered. On the one hand different wiring topologies can be distinguished. On the other hand different physical network topologies are possible, often dependent on cable routing or wireless channel settings. The communication system modelling presented here should be applicable to all of them.

With respect to different wiring topologies, the modelling should be able to cover star, line, and ring topologies including their combinations. Some examples are given in Figure 8, Figure 9, and Figure 10.

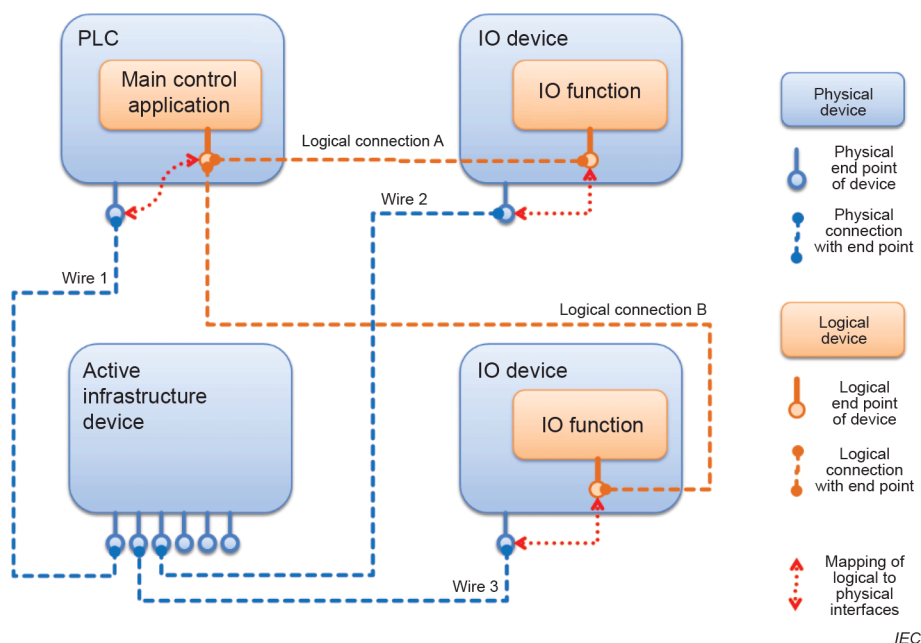


Figure 8 – Star topology example

Figure 8 presents a typical star topology where the active infrastructure device establishes connections between the different physical devices while the logical connections remain bilateral connections between logical devices.

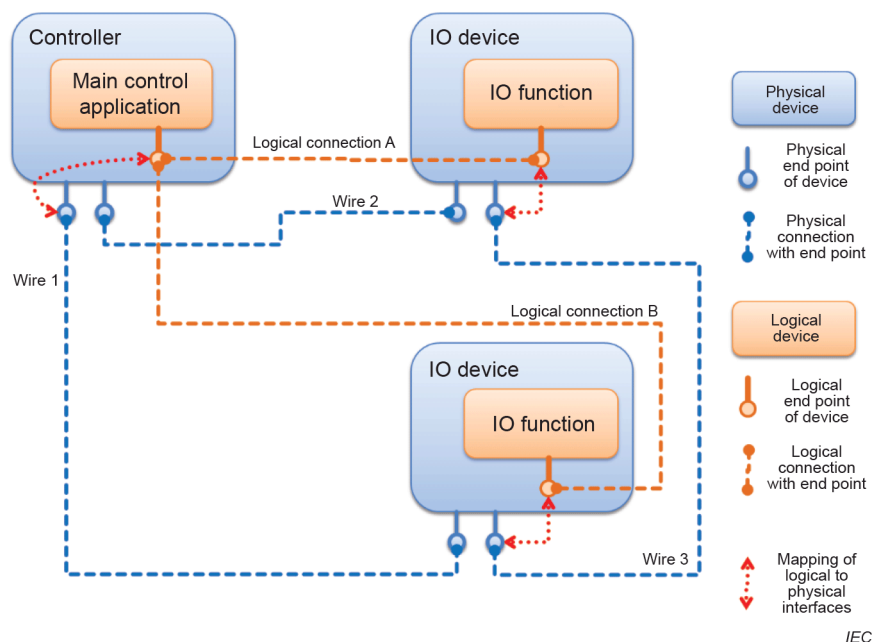


Figure 9 – Ring topology example

Figure 9 presents a typical ring topology where each physical device has two physical interfaces connecting the devices in a physical ring. Again, the logical connections remain bilateral connections between logical devices.

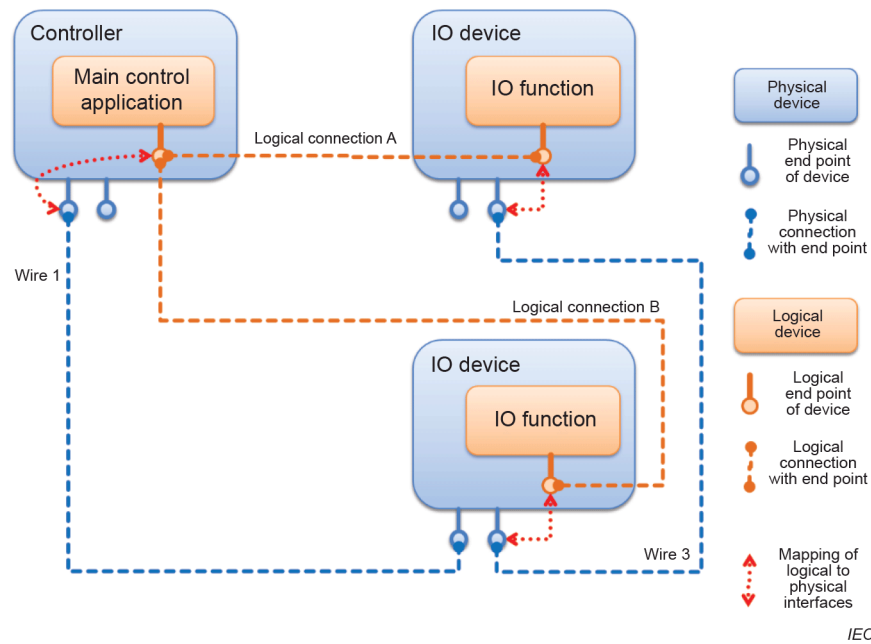


Figure 10 – Line topology example

Figure 10 presents a typical line topology. Here again each physical device has two physical interfaces connecting the devices in a physical line. Thus, it is an open ring topology. The logical connections remain bilateral connections between logical devices.

With respect to network topologies, the modelling should enable the representation of simple networks of direct wiring of components, networks with active infrastructures, networks connected by gateways and hierarchical networks (i.e. networks, where one net component contains a complete further network like a backplane bus in a modular controller). Examples of such network structures are given in Figure 11, Figure 12, Figure 13, and Figure 14.

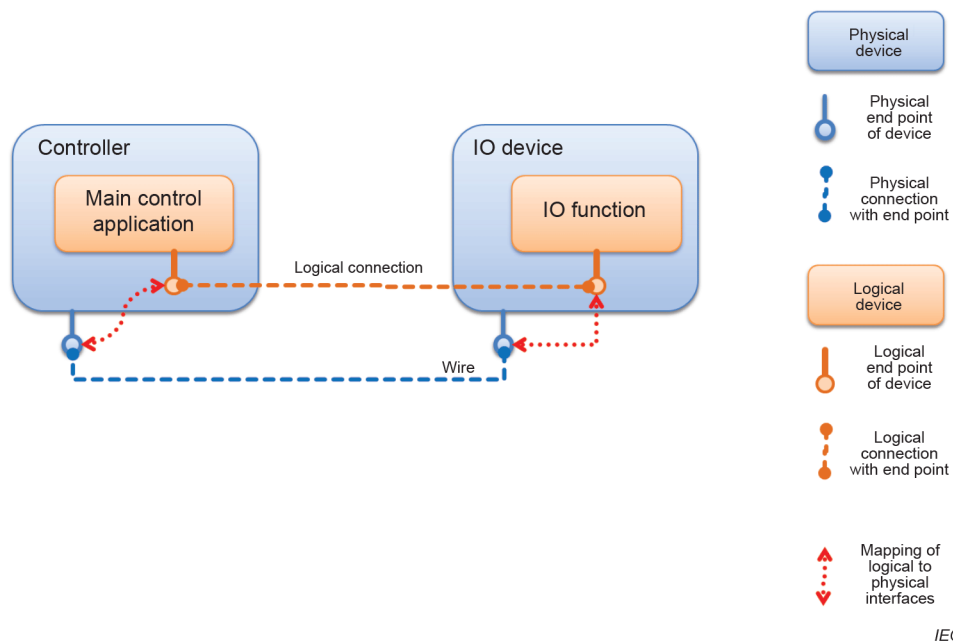


Figure 11 – Simple network with direct wiring

Figure 11 represents a very basic network where two physical devices are directly wired establishing a logical connection between the relevant logical devices. It can be seen as the minimal network which shall be representable.

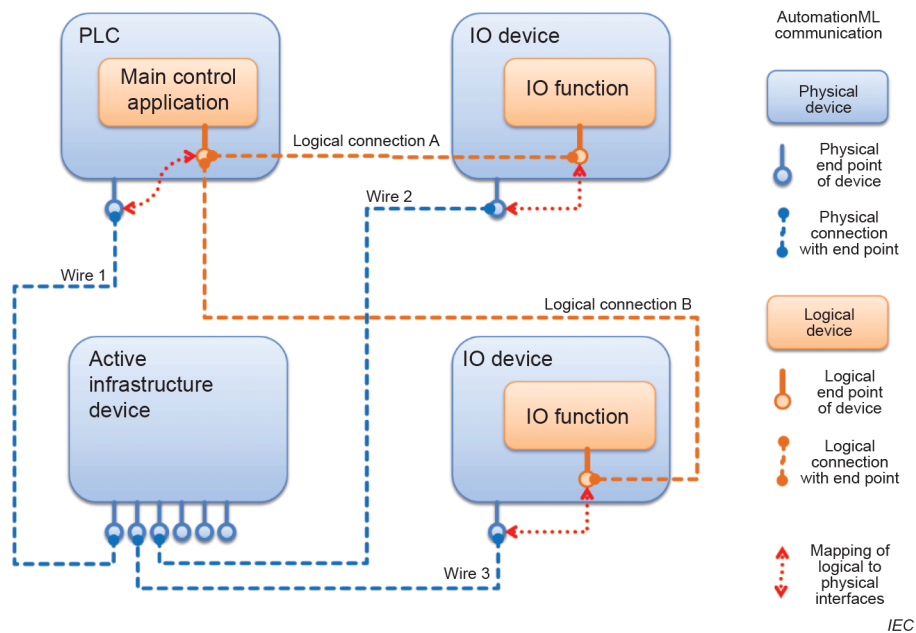


Figure 12 – Network with active infrastructure

Figure 12 represents a network containing active infrastructure devices as is usually the case in Ethernet based networks where switches, hubs, and routers are integrated to forward communication.

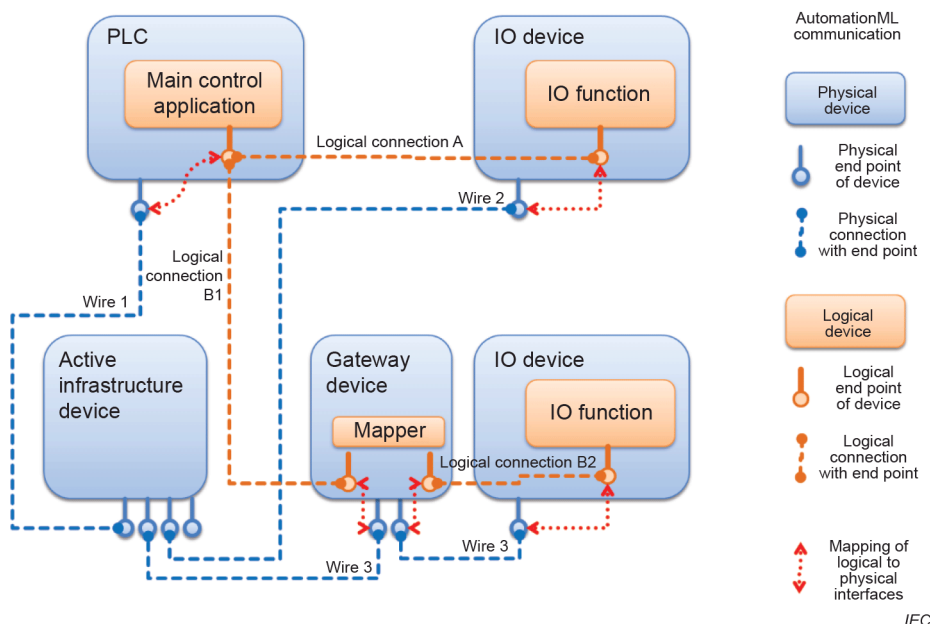


Figure 13 – Networks connected by gateways

Figure 13 represents a network where a logical connection spans two different networks exploiting an application gateway for mapping of logical connections. It depicts that within this gateway, the data items of the two relevant connections are mapped to each other.

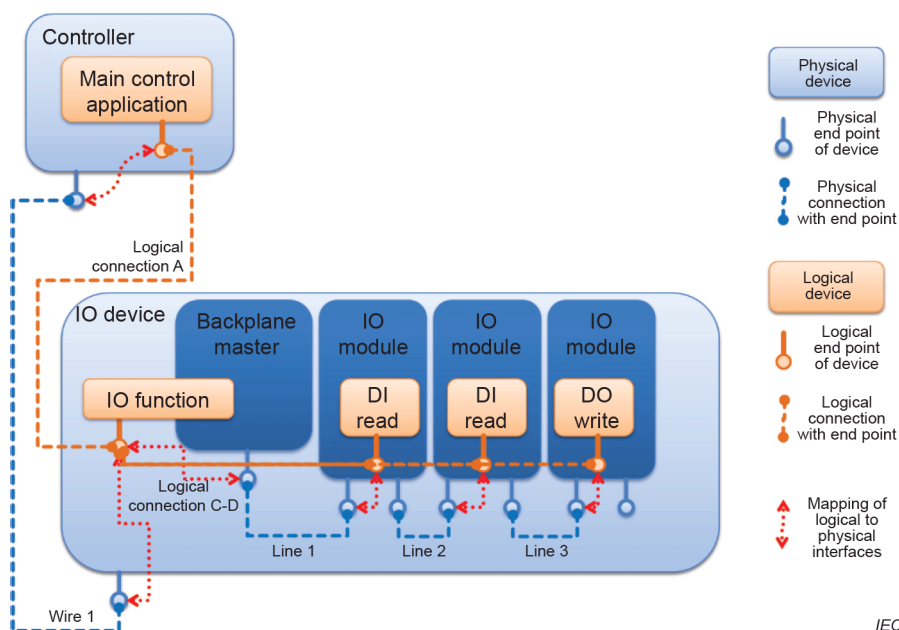


Figure 14 – Hierarchical structured networks

Figure 14 represents a hierarchical network structure as usually given in the case of a communication system containing a physical device with a backplane network connecting the different internal components of the physical device. This case is given if modular fieldbus couplers are exploited in a network.

In addition to the possible application of different topologies and network interaction structures within industrial communication systems, there are usually different applications connected by a communication system like control and system configuration/programming. Networks covering multiple applications should also be modelled as shown in Figure 15.

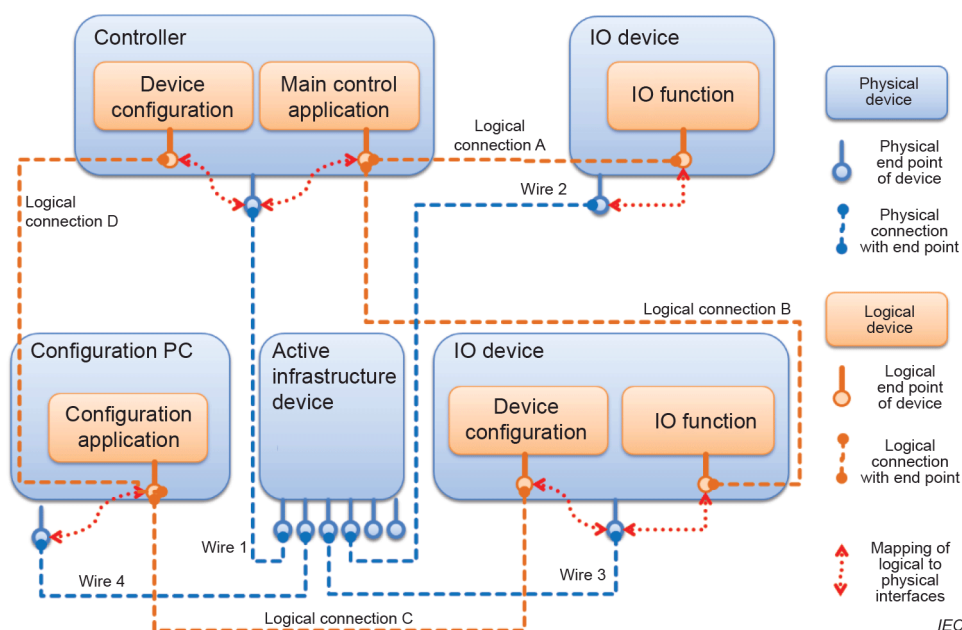


Figure 15 – Network covering multiple applications

4.3.5 Communication content

Within communication systems, communication datagrams (known as Protocol Data Units/PDUs) are exchanged between communication partners, i.e. control application parts. Therefore, each logical connection contains PDU objects exchanged via this connection. In the case of a PDU object containing control information (sensor and actor signals, status, alarms, etc.), it may be linked to a PLCopenXMLInterface (see IEC 61131-10) or a SignalInterface modelling the exchanged information.

This structure is depicted in Figure 16.

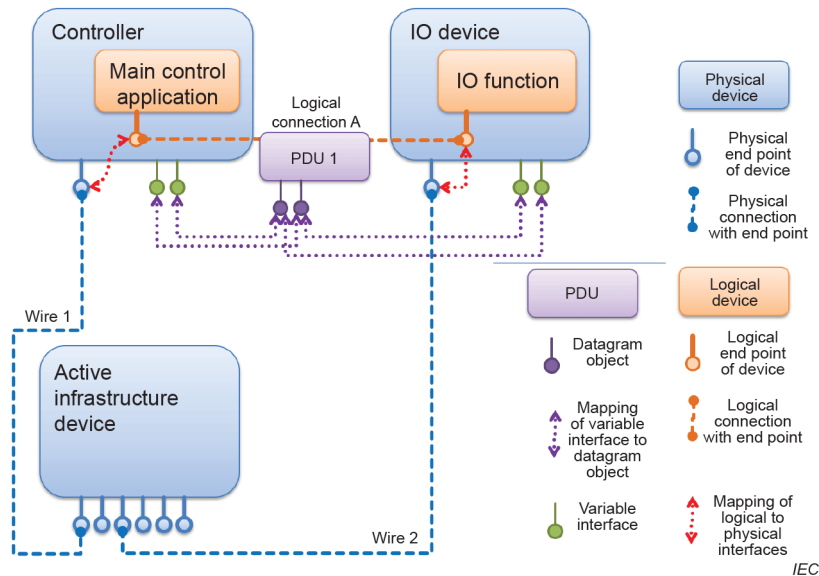


Figure 16 – General modelling strategy for PDUs

4.4 Derived modelling requirements

Based on the observations given in this subclause, the following general information is modelled:

a) Physical level information

- 1) Physical devices representing the physically communicating entities with their properties,
- 2) End points of physical connections attached to physical devices representing the physical interface of the devices (socket) with their descriptive properties,
- 3) Physical connections representing the physical wiring or used wireless channels with their descriptive properties.

b) Logical level information

- 1) Logical devices representing application parts with their descriptive properties,
- 2) End points of logical connections attached to logical devices representing the application part interface with their descriptive properties,
- 3) Logical connections representing the information exchange between application parts with their descriptive properties.
- 4) Process data units modelling the exchanged data packages.

c) Relations between logical and physical levels

- 1) Mapping of end points to represent the implementation of logical connections by physical connections,
- 2) Relations between variables exchanged and applied communication technology/protocol especially by mapping of variables to communication package structures.

d) Communication networks

- 1) Unique association of devices and connections to networks,
- 2) Unique hierarchical network structuring,
- 3) Dependencies between networks,
- 4) Dependencies between devices such as redundancy properties,
- 5) Dependencies between connections such as redundancy properties.

5 UML model

5.1 Overview

The UML model uses class diagrams showing the static relationships between network entities. The entities are always described as types. The real implementation of the model depends on tool preferences. An aggregation demonstrates here a "has a" association or relationship between the classes and the aggregation is more specific than association [1].

More important in the UML class diagrams are the defined cardinalities. If the cardinality is not defined then it is to be read as 0...n.

Figure 17 shows the focal points of the modelled communication structure of a single communication network. The physical device is the central object of the further descriptions. In particular, the modelled structure is divided into the logical and physical topological views.

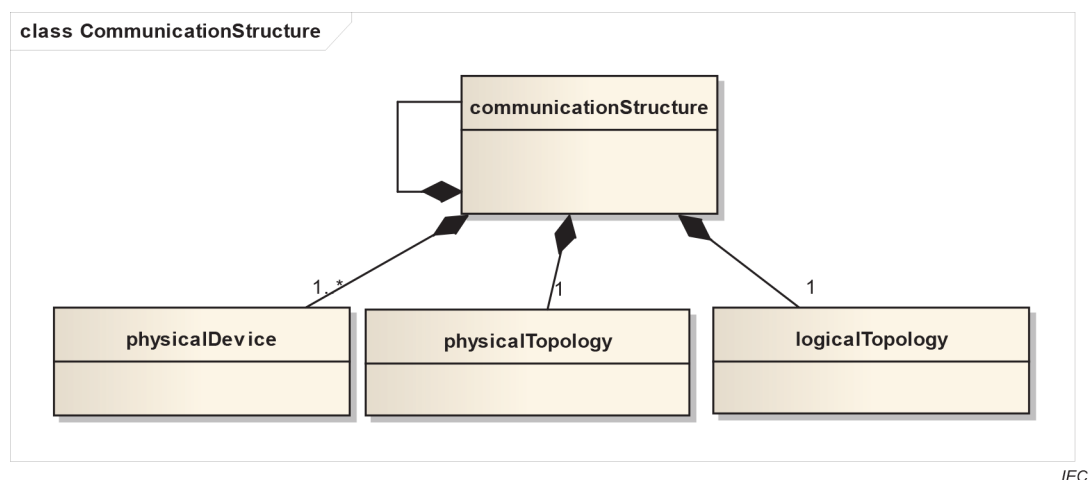


Figure 17 – Structure of communication network

The relationship of the communication structure to itself (reflexive association) allows the representation of nested communication networks or the representation of several interconnected network segments.

5.2 Logical topology

5.2.1 Aim of logical topology

The logical topology describes the logical view of the communication partners. It represents communication relations from the point of view of application programs. Figure 18 shows the three involved (abstract) items and their relationships:

- logicalTopology,
- logicalConnection,
- logicalEndPoint.

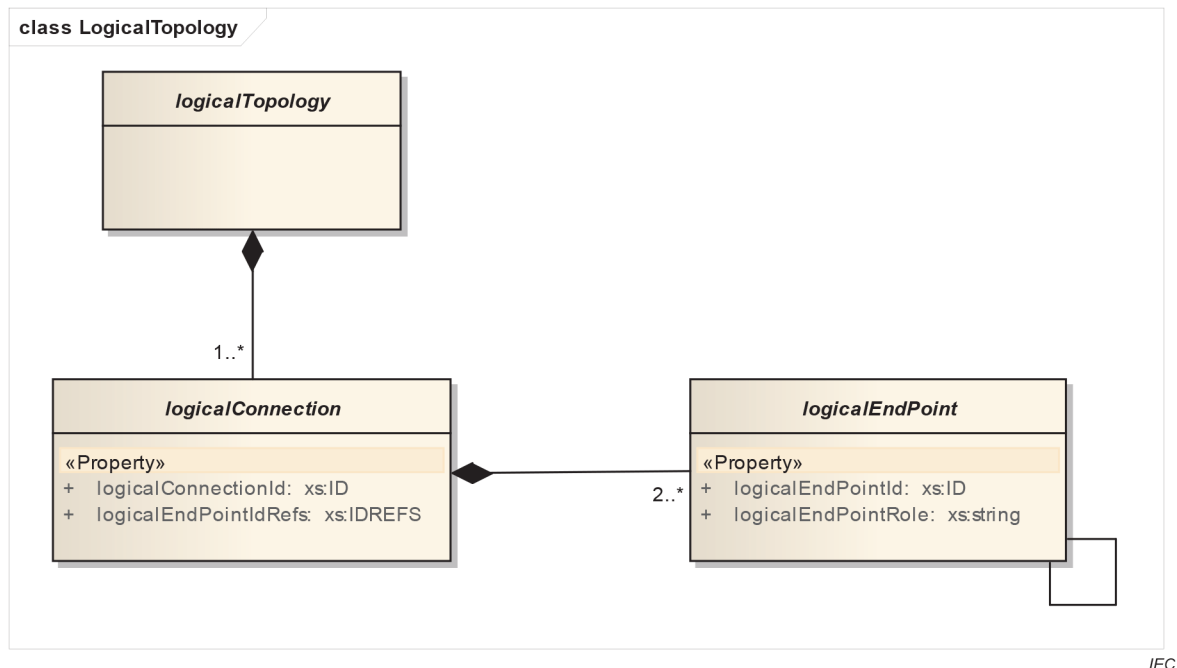


Figure 18 – View on logical topology

5.2.2 Item logicalTopology

The class logicalTopology allows the modelling of an automation or communication network on top of a physical network. It shall aggregate all communication relations between logical end points, which in turn are associated to physical devices.

A real derivation of logicalTopology may be the representation of a control network in a plant.

5.2.3 Item logicalConnection

The entity logicalConnection shall describe the communication relation between two logical end points, which are communication partners. A more generic term is 'logical link', indicating the relationship between at least two logicalEndpoints.

The property logicalConnectionId is a GUID within the communicationStructure and shall identify the logicalConnection.

The property logicalEndPointIdRefs is an array and shall contain all GUIDs of logicalEndpoints related to the logicalConnection. Thus, m:n relationships in a communication relation can be modelled.

A real derivation of logicalConnection may be a connection relation in a peer-to-peer-relationship.

5.2.4 Item logicalEndPoint

The entity logicalEndPoint shall represent a communication partner in such a relationship.

The property logicalEndPointId is a GUID within the communicationStructure and shall identify the logicalEndPoint.

The property `logicalEndPointRole` is of data type string and shall be used to specify the logical role of the `logicalEndPointId` in the communication relation. Conceivable values could be pairs such as "source" – "destination", "source" – "sink", "publisher" – "subscriber" or "master" – "slave", etc.

Derivations of such a communication end point may be an IP port or a peer.

5.3 Physical topology

5.3.1 Aim of physical topology

The physical topology describes the physical view on the communication partners. It shall represent communication relations from the point of view of wiring or channel settings. Figure 19 shows the three involved (abstract) items and their relationships:

- `physicalTopology`,
- `physicalConnection`,
- `physicalEndPoint`.

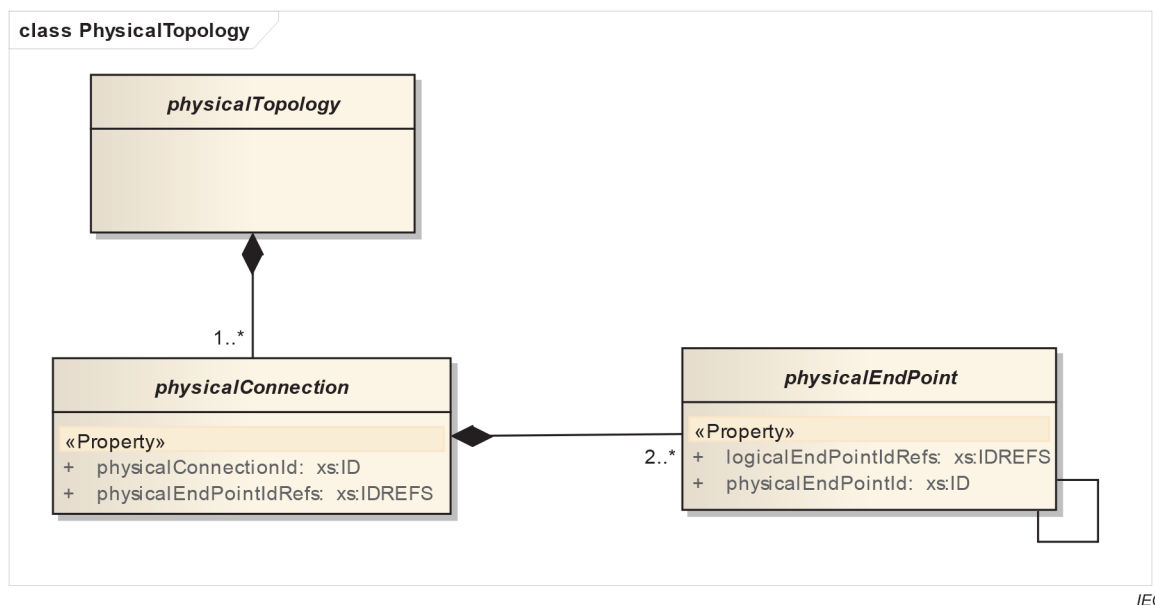


Figure 19 – View on physical topology

5.3.2 Item `physicalTopology`

The class `physicalTopology` is the entry point for modelling the physical topology of an automation network. It shall aggregate all communication relations between physical end points, associated with automation devices.

A real derivation of `physicalTopology` may be the representation of physical networks in a plant.

5.3.3 Item `physicalConnection`

The entity `physicalConnection` shall describe the communication relation between two physical end points. The `physicalEndPoints` may also be connected by means of a wireless network. In this case a link via the ether is established.

The property `physicalConnectionId` is a GUID within the `communicationStructure` and shall identify the `physicalConnection`.

The property `physicalEndPointIdRefs` is an array and shall contain all GUIDs of `physicalEndPoints` related to the `physicalConnection`. Thus, m:n connections in a communication network can be modelled.

A real derivation of `physicalConnection` may be a cable connecting sockets in a peer-to-peer-relationship.

5.3.4 Item `physicalEndPoint`

The entity `physicalEndPoint` shall represent a real communication endpoint in such a relationship.

The property `physicalEndPointId` is a GUID within the `communicationStructure` and shall identify the `physicalEndPoint`.

The property `logicalEndPointIdRefs` is an array and shall contain GUIDs of `logicalEndPoints` related to this `physicalEndPoint`.

Derivation of such an end point may be an Ethernet port.

5.4 Device

5.4.1 General

The model of the device is the most complex in the UML model. Figure 20 shows the first part and Figure 21 the second part of it.

5.4.2 Item `physicalDevice`

Central component is the `physicalDevice`. The attributes allow the unique identification of the instances and sub-instances of `physicalDevice`. Furthermore, there is an aggregation relationship to itself. By means of this relation, modular devices can be modelled. The aggregation of the information item summarises all important product data descriptions, identifiers, documentation, etc. as a central object, which may also be used by instances of a `logicalDevice`. A `physicalDevice` or its sub modules aggregate the `logicalDevices`. Furthermore, the physical end points, e.g. ports, are assigned to the `physicalDevice` by means of the `physicalEndPointList`. The same pattern is used for the physical channels. In the case that the `physicalDevice` is a controller, it may contain several `deviceResources`.

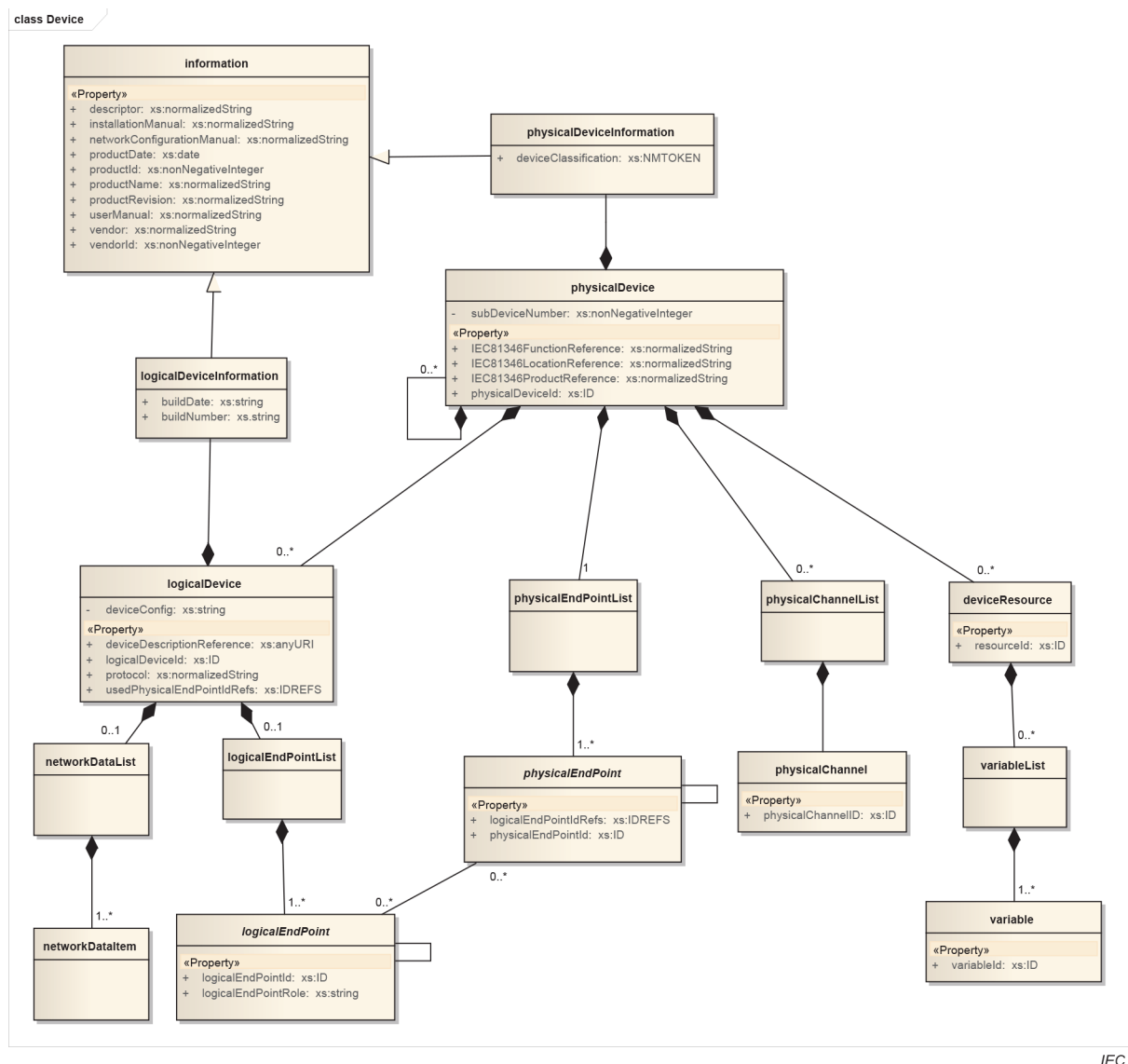


Figure 20 – Part 1 of the device model

NOTE The item logicalEndPoint is the transition point to Figure 21 and is repeated there.

The property physicalDeviceId is a GUID within the communicationStructure and shall identify the physicalDevice. If it is a submodule inside a device, the property subDeviceNumber shall define the manufacturer specific order of the submodule in the device.

The property IEC81346FunctionReference shall describe the functional aspect of the physicalDevice according to the rules of IEC 81346.

The property IEC81346LocationReference shall describe the location of the physicalDevice in the plant according to the rules of IEC 81346.

The property IEC81346ProductReference shall describe the product information of the physicalDevice according to the rules of IEC 81346.

5.4.3 Item Information

The item Information is a placeholder for any kind of information associated with the physical and logical device type. It shall be used to describe the type information of a device instance in detail. The proposed properties may be enhanced on demand.

5.4.4 Item physicalDeviceInformation

The item physicalDeviceInformation expands the generic placeholder Information and shall represent device information related to the physicalDevice.

5.4.5 Item logicalDeviceInformation

The item logicalDeviceInformation expands the generic placeholder Information and shall represent device information related to the logicalDevice.

5.4.6 Item logicalDevice

The item logicalDevice shall represent non-physical aspects of a real device. Often more complex field devices contain several logicalDevices with their own address information and specific tasks.

The logicalDevice shall aggregate the networkDataItems via the networkDataList, used for configuration tasks. Also the logicalEndPoints shall be aggregated via the logicalEndpointList. In addition to the representation of the logical topology in 4.2, the logicalEndPoint also shall aggregate the protocol data units, exchanged via the logical connection.

The property deviceConfig shall describe information used during start-up of the logicalDevice. They may be exchanged at runtime between different communication partners, depending on the communication system.

The property deviceDescriptionReference shall reference to an external device description file, if available.

The property logicalDeviceId is a GUID inside the automation system and shall identify the logicalDevice.

The property protocol shall describe the used communication protocol, e.g. Ethernet based industrial bus, fieldbus etc.

The property usedPhysicalEndPointIdRefs shall represent a list of GUID with associated physicalEndPoints. This means that these end points will be used by the logicalDevice.

5.4.7 Item networkDataList

The item networkDataList shall represent the networkDataItems associated to the logicalDevice.

5.4.8 Item networkDataItem

A networkDataItem shall represent one specific configuration item that belongs to the logicalDevice. Typical networkDataItems can be MAC address, ip address, subnet mask, station name etc.

5.4.9 Item logicalEndPointList

This item is a pattern to enumerate all available logicalEndPoints.

5.4.10 Item physicalEndPointList

This item is a pattern to enumerate all available physicalEndPoints.

5.4.11 Item physicalChannelList

The item physicalChannelList shall represent a placeholder for elements of physicalChannel

5.4.12 Item physicalChannel

The item physicalChannel shall represent a connecting point to a signal, e.g. an analogue or binary in- or output. The data type is not explicitly modelled here. The physicalChannel is listed via physicalChannelList.

The property physicalChannelId is a GUID within the communicationStructure and shall identify the channel or signal.

5.4.13 Item deviceResource

The item deviceResource shall represent a runtime environment (e.g. a CPU or virtual machine with specific software) inside a device allowing executing programs or parts of programs with its man-machine interface or sensor and actuator interface functions. The item is a representation of the term RESOURCE used in IEC 61131-3.

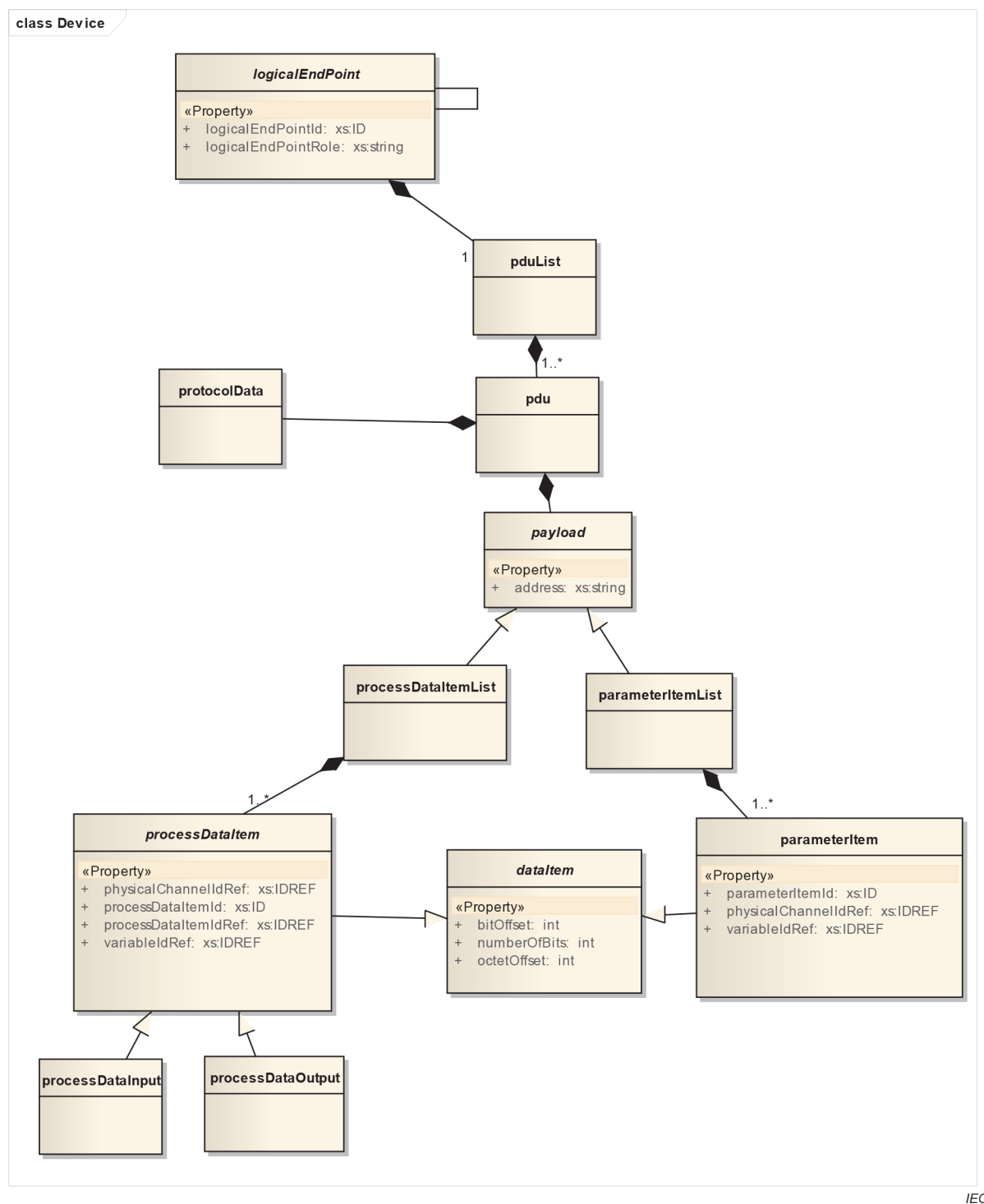
5.4.14 Item variableList

This item is a pattern to enumerate all available variables inside a deviceResource.

5.4.15 Item variable

The item variable represents a variable e.g. of a controller program or in another type of controller. The data type is not explicitly modelled here. The item variable shall represent a VARIABLE inside a programmable component, e.g. a controller according to the definition of IEC 61131-3. The item variable is already defined in IEC 62714-4 as an interface and references to a variable element of PLCopen XML (IEC 61131-10) by means of the GUID.

The property variableId is a GUID within the communicationStructure and shall identify the variable.



IEC

Figure 21 – Part 2 of the device model

NOTE The item **logicalEndPoint** is the transition point from Figure 20 and is repeated here.

5.4.16 Item pduList

The item **pduList** shall represent the Protocol Data Units exchanged via the **logicalEndPoint**.

5.4.17 Item pdu

The item **pdu** shall represent one specific Protocol Data Unit related to this **logicalConnection**. It shall aggregate the **protocolData** and the **payload**. The **payload** can be specialised into **processDataItemList** or **parameterItemList**.

5.4.18 Item protocolData

The item protocolData shall be used to describe the protocol overhead, which is normally not modelled by the processDataItem or parameterItem. For real communication protocols, specialisations of these items can be defined.

5.4.19 Item payload

The item payload shall represent the processDataItem and parameterItem as a generalized abstract item. The optional property address may be used for address or identification purposes of a dataItem.

5.4.20 Item processDataItemList

This item is a pattern to enumerate all processDataItems.

5.4.21 Item parameterItemList

This item is a pattern to enumerate all parameterItems. In addition, a pdu of this type shall have a communication protocol specific address.

5.4.22 Item dataItem

The item dataItem shall represent a data object communicated via the logicalConnection.

The property octetOffset shall define the octet offset of the data object in the PDU payload. The counting shall start with the first octet of the payload starting with index '0' for first octet to send/receive.

The property bitOffset shall define the bit offset of the data object inside an addressed octet of the PDU payload.

The properties octetOffset and bitOffset may be used alternatively. If both properties are used, they shall be added in order to calculate the bit position.

The property numberOfBits shall define the length of the data object in bits.

5.4.23 Item processDataItem

The item processDataItem shall represent a process data object communicated via the logicalConnection. Such data objects are often transmitted by means of cyclic service primitives of the communication system.

The property processDataItemId is a GUID inside the automation system and shall identify the processDataItem.

The property variableIdRef shall be the reference of the processDataItem inside the process data image associated with a variable of the controller or controller program. The reference is set, if the physicalDevice is a controller.

The property processDataItemIdRef shall be the reference of the processDataItem inside the process data image associated with a variable of the device program. The reference is set, if the physicalDevice is not a controller.

The property physicalChannelIdRef shall be the reference of the processDataItem to a physicalChannel of a device.

5.4.24 Item processDataInput

The item processDataInput is a specialisation of processDataItem and shall represent the data flow of the process data as an input from the controlled process.

5.4.25 Item processDataOutput

The item processDataOutput is a specialisation of processDataItem and shall represent the data flow of the process data as an output to the controlled process.

5.4.26 Item parameterItem

The item parameterItem shall represent a parameter object communicated via the logicalConnection. Such data objects are often transmitted by means of acyclic service primitives of the communication system.

The property parameterItemId is a GUID inside the automation system and shall identify the parameterItem.

The property variableItemIdRef shall be the reference to a variable of the controller program. The reference is set, if the physicalDevice is a controller.

The property physicalChannelIdRef shall be the reference of the processDataItem to a physicalChannel of a device.

6 Representation within AutomationML

6.1 Overview of mapping

6.1.1 Introduction of mapping

The modelling method for communication system modelling within AutomationML is based on the use of dedicated <RoleClass>es and <InterfaceClass>es as well as a special application process. This method will take up the distinction between application networks and physical networks and its interconnection as described in Clauses 4 and 5.

6.1.2 General mapping rules

For representation of the different entities named in Clauses 4 and 5, the following mapping rules as shown in Table 1 will be applied. It is distinguished in what manner the semantic of an object is represented and a related entity is modelled.