



BSI Standards Publication

Field device integration (FDI®)

Part 5: FDI Information Model

National foreword

This British Standard is the UK implementation of EN IEC 62769-5:2023. It is identical to IEC 62769-5:2023. It supersedes BS EN IEC 62769-5:2021, which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee GEL/65/5, Devices and integration in enterprise systems.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2023
Published by BSI Standards Limited 2023

ISBN 978 0 539 19844 7

ICS 25.040.40; 35.100.01; 35.100.05

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 June 2023.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

English Version

Field Device Integration (FDI®) - Part 5: FDI Information Model
(IEC 62769-5:2023)

Intégration des appareils de terrain (FDI®) - Partie 5:
Modèle d'Information FDI
(IEC 62769-5:2023)

Feldgeräteintegration (FDI®) - Teil 5: FDI-
Informationsmodell
(IEC 62769-5:2023)

This European Standard was approved by CENELEC on 2023-05-10. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and the United Kingdom.



European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

European foreword

The text of document 65E/858/CDV, future edition 3 of IEC 62769-5, prepared by SC 65E "Devices and integration in enterprise systems" of IEC/TC 65 "Industrial-process measurement, control and automation" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN IEC 62769-5:2023.

The following dates are fixed:

- latest date by which the document has to be implemented at national (dop) 2024-02-10 level by publication of an identical national standard or by endorsement
- latest date by which the national standards conflicting with the (dow) 2026-05-10 document have to be withdrawn

This document supersedes EN IEC 62769-5:2021 and all of its amendments and corrigenda (if any).

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

Any feedback and questions on this document should be directed to the users' national committee. A complete listing of these bodies can be found on the CENELEC website.

Endorsement notice

The text of the International Standard IEC 62769-5:2023 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following notes have to be added for the standard indicated:

IEC 61987 (series)	NOTE	Approved as EN IEC 61987 (series)
IEC/TR 62541-1	NOTE	Approved as CLC IEC/TR 62541-1
IEC 62541-7	NOTE	Approved as EN IEC 62541-7

Annex ZA (normative)

Normative references to international publications with their corresponding European publications

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.

NOTE 1 Where an International Publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

NOTE 2 Up-to-date information on the latest versions of the European Standards listed in this annex is available here: www.cencenelec.eu.

<u>Publication</u>	<u>Year</u>	<u>Title</u>	<u>EN/HD</u>	<u>Year</u>
IEC 61784-1-3	2023	Industrial networks - Profiles - Part 1-3: Fieldbus profiles - Communication Profile Family 3	EN IEC 61784-1-3	2023
IEC 61804-3	-	Devices and integration in enterprise systems - Function blocks (FB) for process control and electronic device description language (EDDL) - Part 3: EDDL syntax and semantics	EN IEC 61804-3	-
IEC 61804-4	-	Devices and integration in enterprise systems - Function blocks (FB) for process control and electronic device description language (EDDL) - Part 4: EDD interpretation	EN IEC 61804-4	-
IEC 62541-3	-	OPC Unified Architecture - Part 3: Address Space Model	EN IEC 62541-3	-
IEC 62541-4	-	OPC Unified Architecture - Part 4: Services	EN IEC 62541-4	-
IEC 62541-5	-	OPC Unified Architecture - Part 5: Information Model	EN IEC 62541-5	-
IEC 62541-6	-	OPC Unified Architecture - Part 6: Mappings	EN IEC 62541-6	-
IEC 62541-8	-	OPC Unified Architecture - Part 8: Data Access	EN IEC 62541-8	-
IEC 62541-100	-	OPC Unified Architecture - Part 100: Device Interface	EN 62541-100	-
IEC 62769-1	-	Field Device Integration (FDI®) - Part 1: Overview	EN IEC 62769-1	-
IEC 62769-2	-	Field Device Integration (FDI®) - Part 2: Client	EN IEC 62769-2	-
IEC 62769-3	-	Field Device Integration (FDI®) - Part 3: Server	EN IEC 62769-3	-
IEC 62769-4	-	Field Device Integration (FDI®) - Part 4: FDI Packages	EN IEC 62769-4	-
IEC 62769-6	-	Field Device Integration (FDI®) - Part 6: FDI Technology Mappings	EN IEC 62769-6	-
IEC 62769-7	-	Field Device Integration (FDI®) - Part 7: Communication Devices	EN IEC 62769-7	-

EN IEC 62769-5:2023 (E)

<u>Publication</u>	<u>Year</u>	<u>Title</u>	<u>EN/HD</u>	<u>Year</u>
IEC 62769-1xx	series	Field device integration (FDI®) - Part 1xx-y: Profiles	-	-
OPC 10000-19	-	OPC Unified Architecture - Part 19: Dictionary Reference	-	-

CONTENTS

FOREWORD.....	7
1 Scope.....	9
2 Normative references	10
3 Terms, definitions, abbreviated terms, acronyms and conventions.....	11
3.1 Terms and definitions.....	11
3.2 Abbreviated terms and acronyms	11
3.3 Conventions.....	11
3.3.1 Capitalization.....	11
3.3.2 Conventions for graphical notation.....	11
4 Overview of OPC Unified Architecture	13
4.1 General.....	13
4.2 Overview of OPC UA Devices	14
5 Concepts	16
5.1 General.....	16
5.2 Device topology	16
5.3 Online/offline	17
5.4 Catalogue (Type Definitions).....	18
5.5 Communication	18
5.6 Semantic Information	18
6 AddressSpace organization	20
7 Device Model for FDI®	21
7.1 General.....	21
7.2 Online/offline	21
7.3 Device health.....	22
7.3.1 DeviceHealth Mapping.....	22
7.3.2 DeviceHealth Diagnostics	23
7.4 User interface elements	24
7.4.1 General	24
7.4.2 UI Description Type	24
7.4.3 UI Plug-in Type.....	25
7.5 Type-specific support information	26
7.6 Actions	27
7.6.1 Overview	27
7.6.2 Action Type	27
7.6.3 ActionService Type.....	28
7.6.4 ActionService Object	28
7.6.5 InvokeAction Method	29
7.6.6 RespondAction Method.....	30
7.6.7 AbortAction Method	31
7.6.8 Interactive Transfer to device	32
8 Network and connectivity.....	32
9 Utility functions	32
9.1 Overview.....	32
9.2 Locking.....	32
9.3 EditContext.....	33
9.3.1 Overview	33

9.3.2	EditContext Type	33
9.3.3	EditContext Object.....	33
9.3.4	GetEditContext Method.....	34
9.3.5	RegisterNodes Method	35
9.3.6	Apply Method	36
9.3.7	Reset Method	37
9.3.8	Discard Method	38
9.4	DirectDeviceAccess	39
9.4.1	General	39
9.4.2	DirectDeviceAccess Type	39
9.4.3	DirectDeviceAccess Object.....	40
9.4.4	InitDirectAccess Method	41
9.4.5	EndDirectAccess Method.....	41
9.4.6	Transfer Method	42
10	Parameter Types	43
10.1	General.....	43
10.2	ScalingFactor Property	44
10.3	Min_Max_Values Property	44
11	FDI® StatusCodes.....	45
11.1	General.....	45
11.2	Structure of the StatusCode	45
11.3	FDI® specific operation level result codes	46
12	Specialized topology elements.....	49
13	Auditing.....	50
13.1	General.....	50
13.2	FDI® Client-provided context information.....	50
13.3	LogAuditTrailMessage Method	50
14	FDI® Server Version	51
15	Mapping FDI® Package information to the FDI® Information Model.....	51
15.1	General.....	51
15.2	Localization	52
15.2.1	Localized text	52
15.2.2	Engineering units.....	52
15.3	Device	52
15.3.1	General	52
15.3.2	Mapping to Attributes to a specific DeviceType Node.....	52
15.3.3	Mapping to Properties.....	52
15.3.4	Mapping to ParameterSet	53
15.3.5	Mapping to Functional Groups	53
15.3.6	Mapping to DeviceTypeImage.....	53
15.3.7	Mapping to Documentation	53
15.3.8	Mapping to ProtocolSupport.....	53
15.3.9	Mapping to ImageSet.....	54
15.3.10	Mapping to ActionSet.....	54
15.3.11	Mapping to MethodSet.....	54
15.4	Modular Device	54
15.5	Block	54
15.5.1	General	54

15.5.2	Mapping to Attributes.....	54
15.5.3	Mapping to ParameterSet	55
15.5.4	Mapping to Functional Groups	55
15.5.5	Mapping to ActionSet.....	55
15.5.6	Mapping to MethodSet.....	55
15.5.7	Instantiation rules	55
15.6	Parameter	55
15.6.1	General	55
15.6.2	Private Parameters	60
15.6.3	MIN_Value and MAX_Value.....	60
15.6.4	Engineering units.....	60
15.6.5	Enumerated Parameters	60
15.6.6	Bit-enumerated Parameters	60
15.6.7	Representation of records.....	61
15.6.8	Representation of arrays, and lists of Parameters with simple data types	62
15.6.9	Representation of values arrays, and lists of RECORD Parameters	62
15.6.10	Representation of COLLECTION and REFERENCE ARRAY	63
15.6.11	SCALING_FACTOR.....	63
15.6.12	EDDL CLASS Attributes on Parameters	63
15.7	Functional Groups.....	65
15.8	AXIS elements in UIDs.....	65
15.9	Actions	65
15.10	UIPs	66
15.11	Protocols, Networks and Connection Points	66
15.12	Semantic Identifies	66
15.13	DictionaryIds Property.....	67
15.14	MultiStateDictionaryEntryDiscreteType	67
15.15	GetNodeIdsByDictionaryEntryId	68
16	Profiles	69
Annex A (normative) Namespace and Mappings		70
Bibliography.....		71
Figure 1 – FDI® architecture diagram		10
Figure 2 – OPC UA graphical notation for NodeClasses		12
Figure 3 – OPC UA graphical notation for References.....		12
Figure 4 – OPC UA graphical notation example		13
Figure 5 – Optimized Type Reference		13
Figure 6 – OPC UA Devices example: Functional Groups		15
Figure 7 – OPC UA Devices example: Configurable components		15
Figure 8 – Example of an automation system.....		16
Figure 9 – Example of a Device topology		17
Figure 10 – Example Device Types representing a catalogue		18
Figure 11 – Example of concrete DictionaryEntryType and Object		19
Figure 12 – Example of DictionaryEntries		20
Figure 13 – Online component for access to device data		21
Figure 14 – Hierarchy of user interface Types.....		24

Figure 15 – Integration of Actions within a TopologyElement	27
Figure 16 – Action Service	29
Figure 17 – EditContext type and instance	34
Figure 18 – DirectDeviceAccessType	39
Figure 19 – DirectDeviceAccess instance	40
Figure 20 – OPC UA VariableTypes including OPC UA DataAccess	44
Figure 21 – Example: Complex variable representing a RECORD	61
Figure 22 – Complex variable representing a VALUE_ARRAY of RECORDs	62
Figure 23 – Example of EDDL CLASS Attributes in the FDI® OPC UA Information Model	64
Table 1 – DeviceHealth Mapping	22
Table 2 – DeviceType definition (excerpt applicable for Subclause 7.3.1)	22
Table 3 – DeviceType definition with DeviceHealth and DeviceHealthDiagnostics	23
Table 4 – UIDescriptionType Definition	24
Table 5 – UIPlugInType Definition	25
Table 6 – ActionType Definition	28
Table 7 – ActionServiceType Definition	28
Table 8 – InvokeAction Method Arguments	30
Table 9 – InvokeAction Method AddressSpace Definition	30
Table 10 – RespondAction Method Arguments	31
Table 11 – RespondAction Method AddressSpace Definition	31
Table 12 – AbortAction Method Arguments	31
Table 13 – AbortAction Method AddressSpace Definition	32
Table 14 – EditContextType Definition	33
Table 15 – GetEditContext Method Arguments	34
Table 16 – GetEditContext Method AddressSpace Definition	35
Table 17 – RegisterNodes Method Arguments	35
Table 18 – RegisterNodes Method AddressSpace Definition	35
Table 19 – RegistrationParameters DataType Structure	36
Table 20 – RegisterNodesResult DataType Structure	36
Table 21 – Apply Method Arguments	37
Table 22 – Apply Method AddressSpace Definition	37
Table 23 – ApplyResult DataType Structure	37
Table 24 – Reset Method Arguments	38
Table 25 – Reset Method AddressSpace Definition	38
Table 26 – Discard Method Arguments	38
Table 27 – Discard Method AddressSpace Definition	38
Table 28 – DirectDeviceAccessType Definition	40
Table 29 – DirectDeviceAccess Instance Definition	41
Table 30 – InitDirectAccess Method Arguments	41
Table 31 – InitDirectAccess Method AddressSpace Definition	41
Table 32 – EndDirectAccess Method Arguments	42

Table 33 – EndDirectAccess Method AddressSpace Definition.....	42
Table 34 – Transfer Method Arguments	42
Table 35 – Transfer Method AddressSpace Definition	43
Table 36 – ScalingFactor Property Definition	44
Table 37 – Min_Max_Values Property Definition	45
Table 38 – Variant_Range DataType Structure	45
Table 39 – Variant_Range Definition.....	45
Table 40 – StatusCode Bit Assignments	46
Table 41 – DataValue InfoBits.....	46
Table 42 – Good operation level result codes	47
Table 43 – Uncertain operation level result codes	48
Table 44 – Bad operation level result codes.....	48
Table 45 – LogAuditTrailMessage Method Arguments.....	51
Table 46 – LogAuditTrailMessage Method AddressSpace Definition	51
Table 47 – FDIServerVersion Property Definition	51
Table 48 – DeviceType Property Mapping.....	53
Table 49 – Setting OPC UA Variable Attributes from EDDL variable attributes	56
Table 50 – Correspondence between EDDL and OPC UA standard data types	57
Table 51 – Definition of EddIDictionaryType.....	63
Table 52 – Definition of EddIDictionary Object	63
Table 53 – Definition of Parameter Class Attributes	64
Table 54 – DictionaryIds Definition.....	67
Table 55 – MultiStateDictionaryEntryDiscreteType definition.....	67
Table 56 – GetNodeIdsByDictionaryEntryId Method arguments.....	68
Table 57 – GetNodeIdsByDictionaryEntryId Method result codes	68
Table 58 – GetNodeIdsByDictionaryEntryId	68
Table 59 – FDI® Server Facet Definition.....	69
Table 60 – FDI® Client Facet Definition	69

INTERNATIONAL ELECTROTECHNICAL COMMISSION

FIELD DEVICE INTEGRATION (FDI®) –

Part 5: FDI® Information Model

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 62769-5 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.

This third edition cancels and replaces the second edition published in 2021. This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

- a) added INTERACTIVE_TRANSFER_TO_DEVICE ACTION.

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

Draft	Report on voting
65E/858/CDV	65E/915/RVC

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/publications.

A list of all parts in the IEC 62769 series, published under the general title *Field device integration (FDI®)*, 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.

FIELD DEVICE INTEGRATION (FDI®) –

Part 5: FDI® Information Model

1 Scope

This part of IEC 62769 defines the FDI®¹ Information Model. One of the main tasks of the Information Model is to reflect the topology of the automation system. Therefore, it represents the devices of the automation system as well as the connecting communication networks including their properties, relationships, and the operations that can be performed on them. The types in the AddressSpace of the FDI® Server constitute some kind of catalogue, which is built from FDI® Packages.

The fundamental types for the FDI® Information Model are well defined in OPC UA for Devices (IEC 62541-100). The FDI® Information Model specifies extensions for a few special cases and otherwise explains how these types are used and how the contents are built from elements of DevicePackages.

The overall FDI® architecture is illustrated in Figure 1. The architectural components that are within the scope of this document have been highlighted in this illustration.

¹ FDI® is a registered trademark of the non-profit organization Fieldbus Foundation, Inc. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance does not require use of the trade name. Use of the trade name requires permission of the trade name holder.

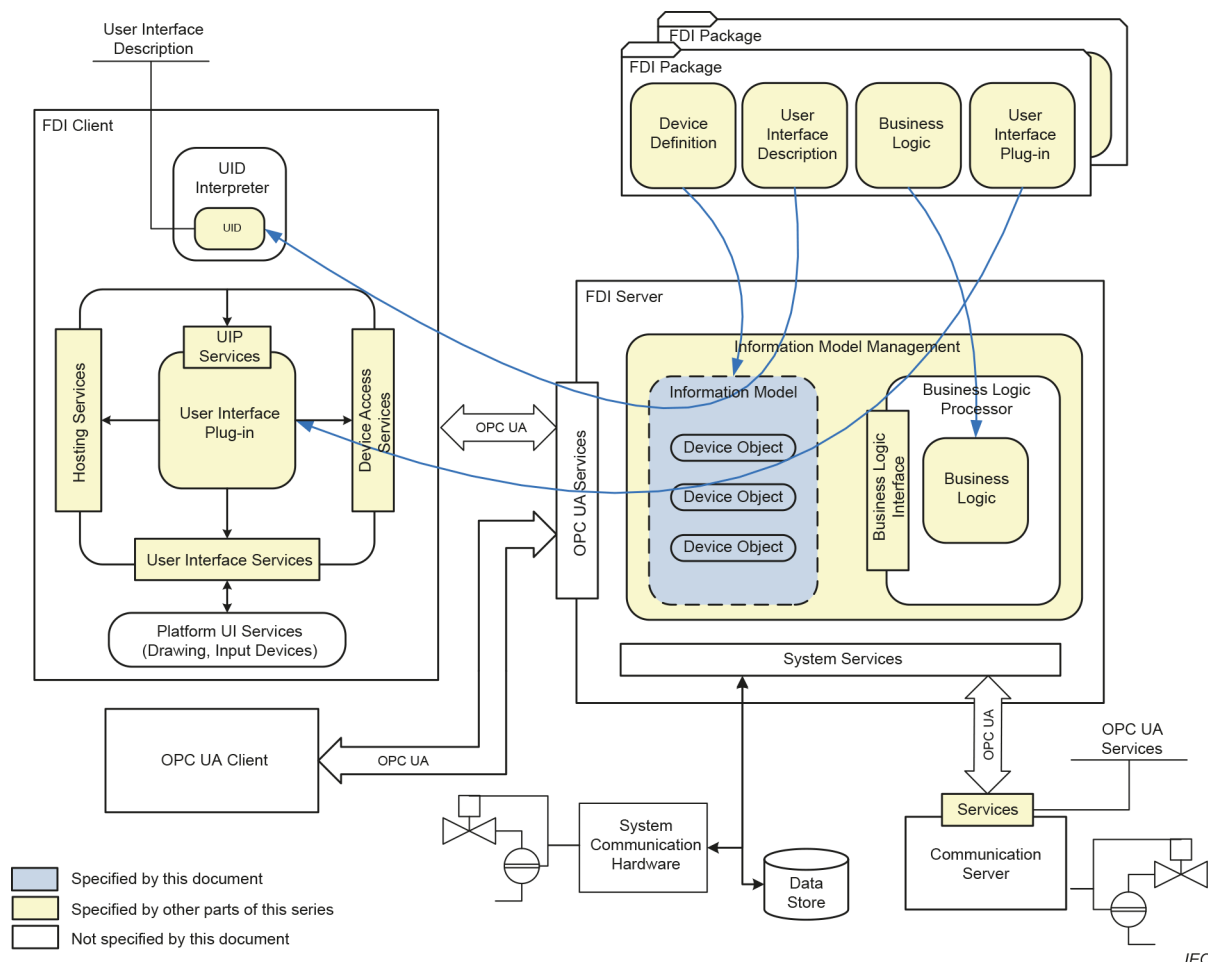


Figure 1 – FDI® architecture diagram

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 61784-1-3:2023, *Industrial networks – Profiles – Part 1-3: Fieldbus profiles – Communication Profile Family 3*

IEC 61804-3, *Devices and integration in enterprise systems – Function blocks (FB) for process control and electronic device description language (EDDL) – Part 3: EDDL syntax and semantics*

IEC 61804-4, *Devices and integration in enterprise systems – Function blocks (FB) for process control and electronic device description language (EDDL) – Part 4: EDD interpretation*

IEC 62541-3, *OPC Unified Architecture – Part 3: Address Space Model*

IEC 62541-4, *OPC Unified Architecture – Part 4: Services*

IEC 62541-5, *OPC Unified Architecture – Part 5: Information Model*

IEC 62541-6, *OPC Unified Architecture – Part 6: Mappings*

IEC 62541-8, *OPC Unified Architecture – Part 8: Data Access*

IEC 62541-100, *OPC Unified Architecture – Part 100: Device Interface*

IEC 62769-1, *Field Device Integration (FDI®) – Part 1: Overview*

IEC 62769-2, *Field Device Integration (FDI®) – Part 2: Client*

IEC 62769-3, *Field Device Integration (FDI®) – Part 3: Server*

IEC 62769-4, *Field Device Integration (FDI®) – Part 4: FDI® Packages*

IEC 62769-6, *Field Device Integration (FDI®) – Part 6: FDI® Technology Mappings*

IEC 62769-7, *Field Device Integration (FDI®) – Part 7: Communication Devices*

IEC 62769-1xx (all parts), *Field Device Integration (FDI®) – Part 1xx-y: Profiles*

OPC 10000-19, *OPC Unified Architecture – Part 19: Dictionary Reference*

3 Terms, definitions, abbreviated terms, acronyms and conventions

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62769-1 and IEC 62769-3 apply.

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

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

3.2 Abbreviated terms and acronyms

For the purposes of this document, the abbreviated terms and acronyms given in IEC 62769-1 and the following apply.

HMI	Human Machine Interface
SCADA	Supervisory Control and Data Acquisition
TCP	Transmission Control Protocol

3.3 Conventions

3.3.1 Capitalization

Capitalization of the first letter of words is used in the IEC 62769 series to emphasize an FDI® defined term.

3.3.2 Conventions for graphical notation

OPC UA defines a graphical notation for an OPC UA AddressSpace. It defines graphical symbols for all NodeClasses and how different types of References between Nodes can be visualized. Figure 2 shows the symbols for the NodeClasses used in this document. NodeClasses representing types always have a shadow.

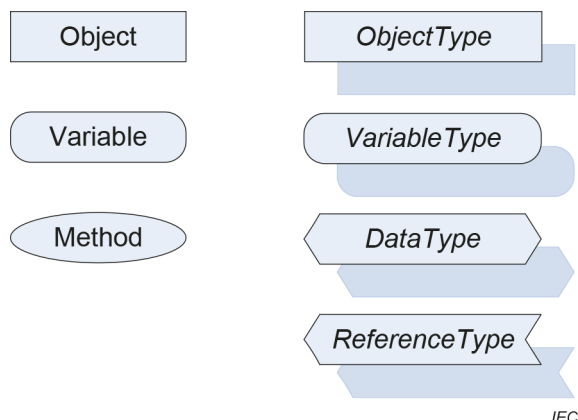


Figure 2 – OPC UA graphical notation for NodeClasses

Figure 3 shows the symbols for the ReferenceTypes used in this document. The Reference symbol is normally pointing from the source Node to the target Node. The only exception is the HasSubType Reference. The most important References such as HasComponent, HasProperty, HasTypeDefinition and HasSubType have special symbols avoiding the name of the Reference. For other ReferenceTypes or derived ReferenceTypes, the name of the ReferenceType is used together with the symbol.

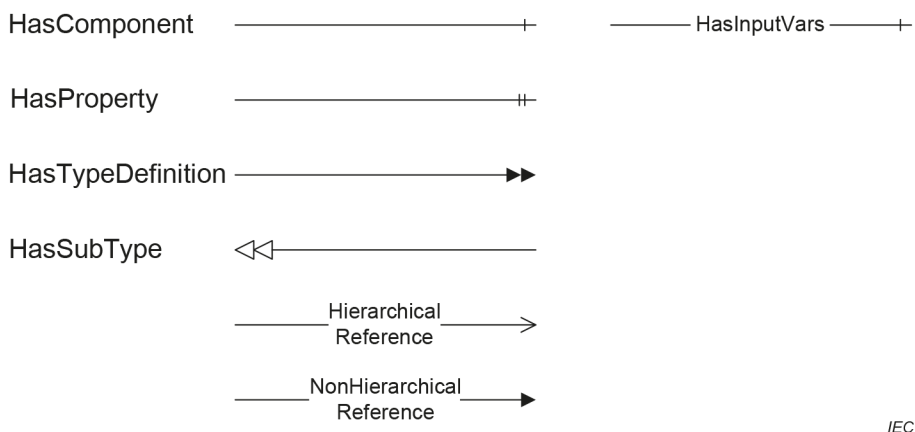


Figure 3 – OPC UA graphical notation for References

Figure 4 shows a typical example for the use of the graphical notation. Object_A and Object_B are instances of the ObjectType_Y indicated by the HasTypeDefinition References. The ObjectType_Y is derived from ObjectType_X indicated by the HasSubType Reference. The Object_A has the components Variable_1, Variable_2 and Method_1.

To describe the components of an Object on the ObjectType, the same NodeClasses and References are used on the Object and on the ObjectType such as for ObjectType_Y in the example. The Nodes used to describe an ObjectType are instance declaration Nodes.

To provide more detailed information for a Node, a subset or all Attributes and their values can be added to a graphical symbol (see for example Variable_1, the component of Object_A in Figure 4).

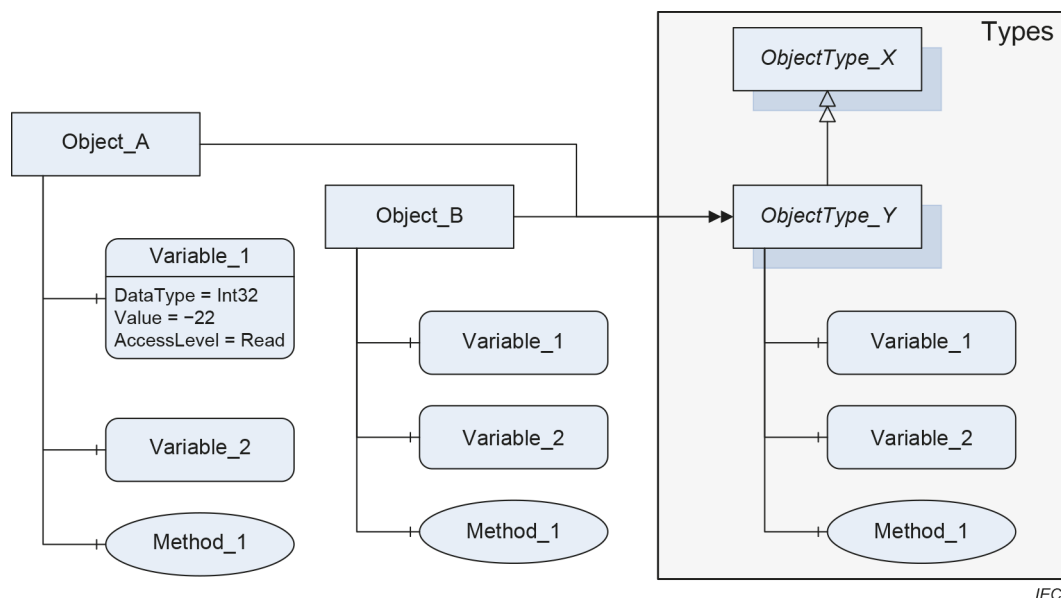


Figure 4 – OPC UA graphical notation example

To improve readability, this document frequently includes the type name inside the instance box rather than displaying both boxes and a reference between them. This optimization is shown in Figure 5.

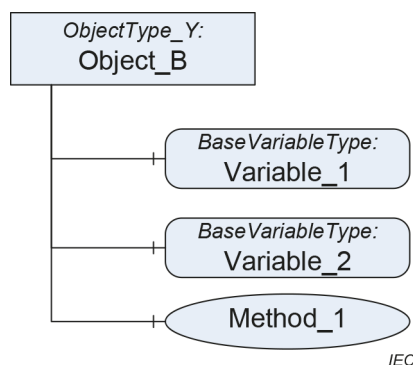


Figure 5 – Optimized Type Reference

4 Overview of OPC Unified Architecture

4.1 General

The main use case for OPC standards is the online data exchange between devices and HMI or SCADA systems. In this use case, the device data is provided by an OPC server and is consumed by an OPC client integrated into the HMI or SCADA system. OPC provides functionality to browse through a hierarchical namespace containing data items and to read, write and monitor these items for data changes.

OPC UA incorporates features like Data Access, Alarms and Historical Data via platform independent communication mechanisms and generic, extensible and object-oriented modelling capabilities for the information a system wants to expose.

The current version of OPC UA defines an optimized binary TCP protocol for high performance intranet communication as well as a mapping to Web Services. The abstract service model does not depend on a specific protocol mapping and allows adding new protocols in the future. Features like security, access control and reliability are directly built into the transport

mechanisms. Based on the platform independence of the protocols, OPC UA servers and clients can be directly integrated into devices and controllers.

The OPC UA information model provides a standard way for Servers to expose Objects to Clients. Objects in OPC UA terms are composed of other Objects, Variables and Methods. OPC UA also allows relationships to other Objects to be expressed.

The set of Objects and related information that an OPC UA Server makes available to Clients is referred to as its AddressSpace. The elements of the OPC UA Object Model are represented in the AddressSpace as a set of Nodes described by Attributes and interconnected by References. OPC UA defines various classes of Nodes to represent AddressSpace components, most importantly Objects, Variables, Methods, ObjectTypes, DataTypes and ReferenceTypes. Each NodeClass has a defined set of Attributes.

Objects are used to represent components like folders, Devices or Networks. An Object is associated to a corresponding ObjectType that provides definitions for that Object.

Variables are used to represent values. Two categories of Variables are defined, Properties and DataVariables.

Properties are Server-defined characteristics of Objects, DataVariables and other Nodes. Properties are not allowed to have Properties defined for them. An example for Properties of Objects is the Manufacturer Property of a Device.

DataVariables represent the contents of an Object. DataVariables can have component DataVariables. This is typically used by Servers to expose individual elements of arrays and structures. This document uses DataVariables mainly to represent the Parameters of Devices.

4.2 Overview of OPC UA Devices

The OPC Unified Architecture for Devices (DI) (IEC 62541-100) standard is an extension of the overall OPC Unified Architecture standard series and defines information models associated with Devices. IEC 62541-100 describes three models which build upon each other as follows:

- The (base) Device Model is intended to provide a unified view of devices irrespective of the underlying device protocols.
- The Device Communication Model adds Network and Connection information elements so that communication topologies can be created.
- The Device Integration Host Model finally adds additional elements and rules required for host systems to manage integration for a complete system. It allows reflecting the topology of the automation system with the devices as well as the connecting communication networks.

The Devices information model specifies different ObjectTypes and other AddressSpace elements used to represent Devices and related components such as the communication infrastructure in an OPC UA AddressSpace. The main use cases are Device configuration and diagnostic but it allows a general and standardized way for any kind of application to access Device related information.

Figure 6 shows an example for a temperature controller represented as Device Object. It is a DeviceType Object that is a subtype of TopologyElementType and inherits all components of this type. The component ParameterSet contains all Variables describing the Device. The component MethodSet contains all Methods provided by the Device. Components of the FunctionalGroupType are used to collect the Parameters and Methods of the Device into logical groups. The FunctionalGroupType and the grouping concept are defined in IEC 62541-100 but the groups are DeviceType specific, i.e., the groups ProcessData and Configuration are defined by the TemperatureControllerType in this example.

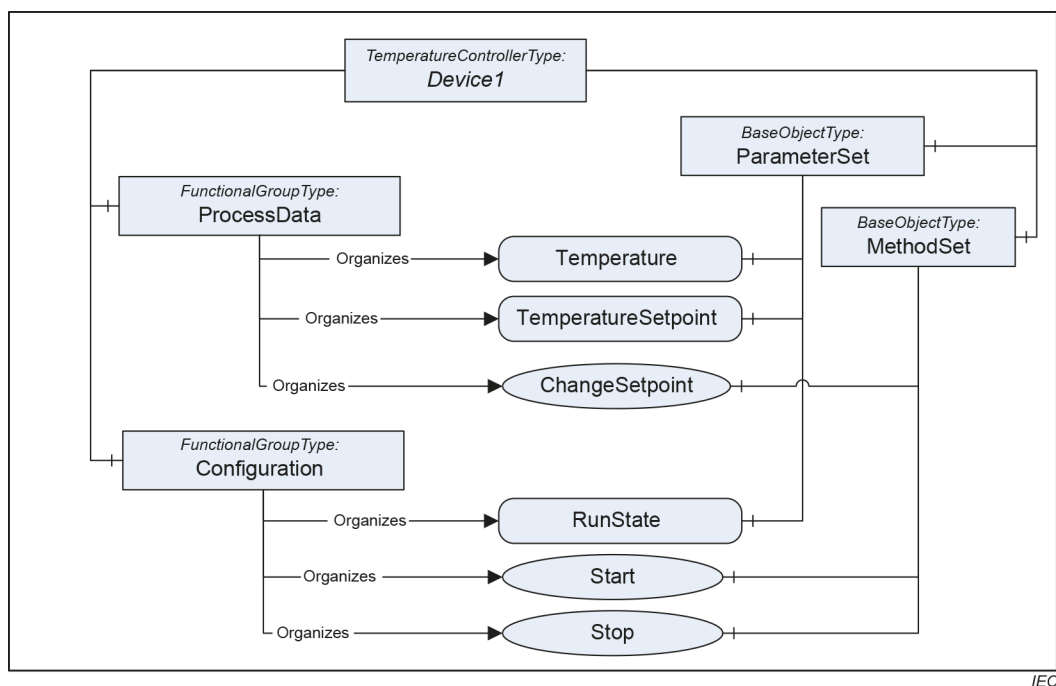


Figure 6 – OPC UA Devices example: Functional Groups

Another IEC 62541-100 concept is illustrated in Figure 7. The ConfigurableObjectType is used to provide a way to group sub components of a Device and to indicate which types of sub components can be instantiated. The allowed types are referenced from the SupportedTypes folder. This information can be used by configuration clients to allow a user to select the type to instantiate as sub component of the Device.

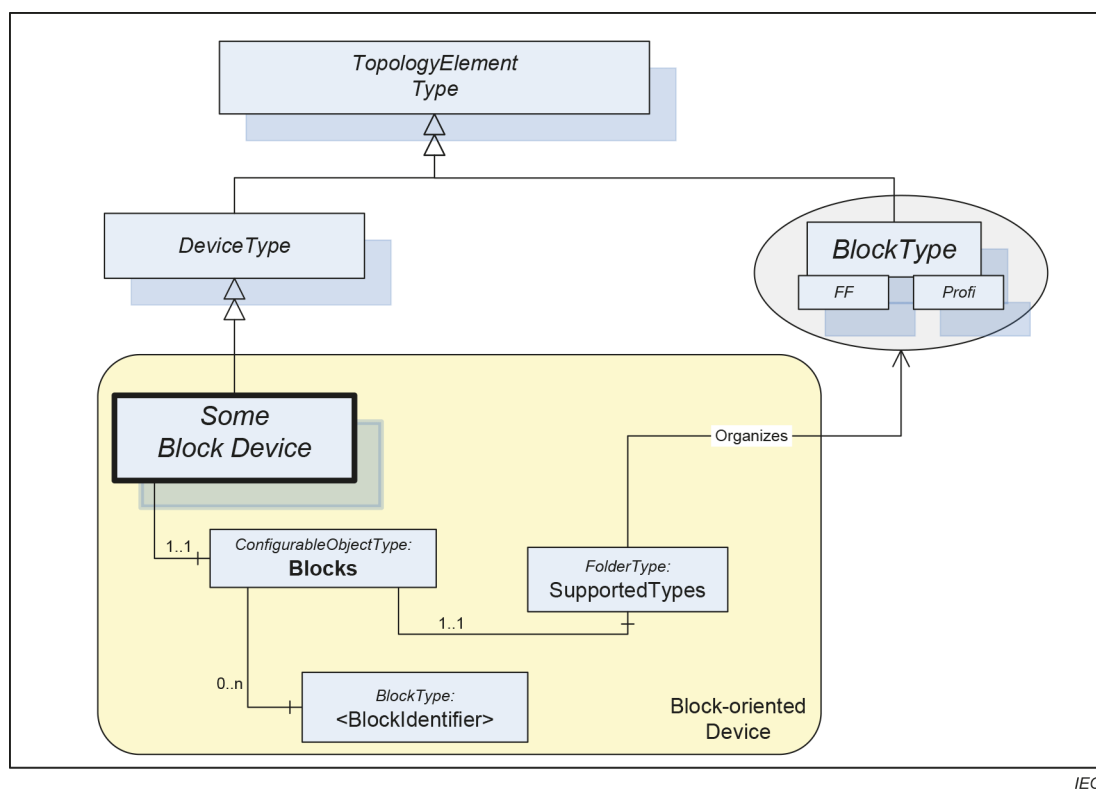


Figure 7 – OPC UA Devices example: Configurable components

The SupportedTypes folder can contain different subsets of ObjectTypes for different instances of the Block-oriented Device depending on their current configuration since the list contains only types that can be instantiated for the current configuration.

5 Concepts

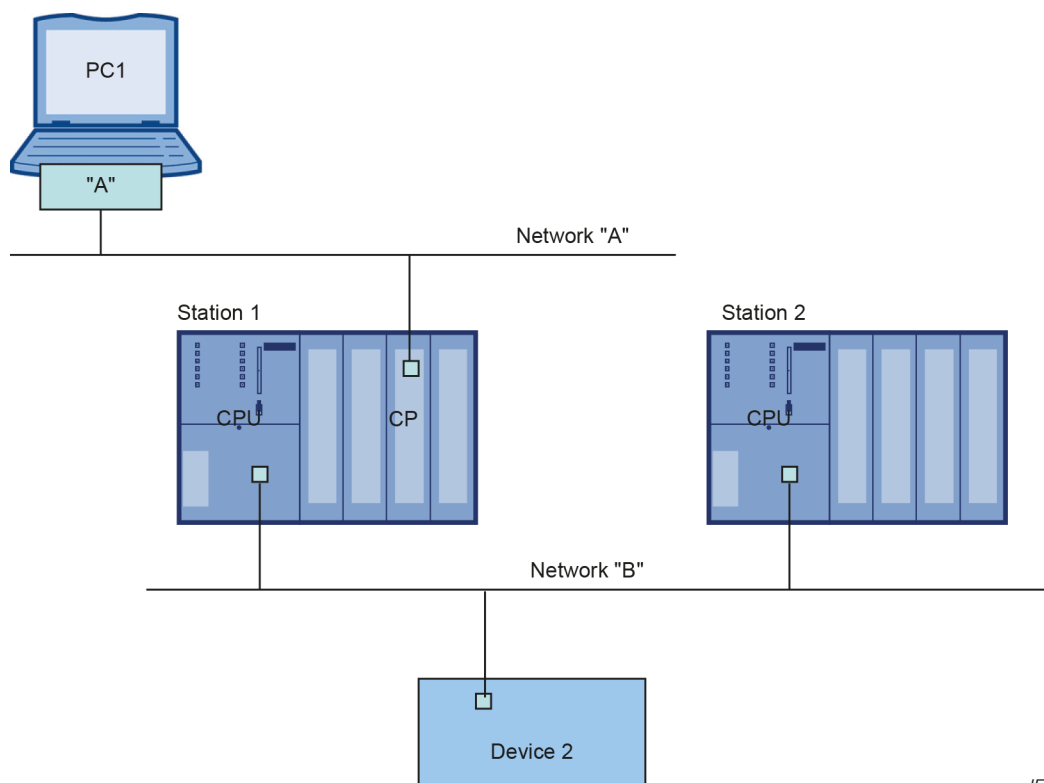
5.1 General

The FDI® Server provides FDI® Clients access to information about Device instances and Device types regardless of where the information is stored, for example, in the Device itself or in a data store. This information is provided via OPC UA Services and is called the FDI® Information Model.

The FDI® Information Model specifies the entities that can be accessed in the FDI® Server, including their properties, relationships, and the operations that can be performed on them. Which types of Devices or other topological elements are available in a given FDI® Server is driven largely by the information in the FDI® Packages.

5.2 Device topology

One of the main tasks of the Information Model is to reflect the topology of the automation system. Therefore, the Information Model represents the devices of the automation system as well as the connecting communication networks. The entry point Device Topology is the starting point within the Information Model for the topology of the automation system. The entry point Communication Devices contains the communication devices that are used by the FDI® Server to access the elements of the topology. Figure 8 and Figure 9 illustrate an example configuration and the configured topology as it will appear in the FDI® Server AddressSpace (details left out).



IEC

Figure 8 – Example of an automation system

The PC in Figure 8 represents the FDI® Server box. The FDI® Server communicates with devices connected to Network "A" via a Native Communication, and it communicates with devices connected to Network "B" via a Nested Communication.

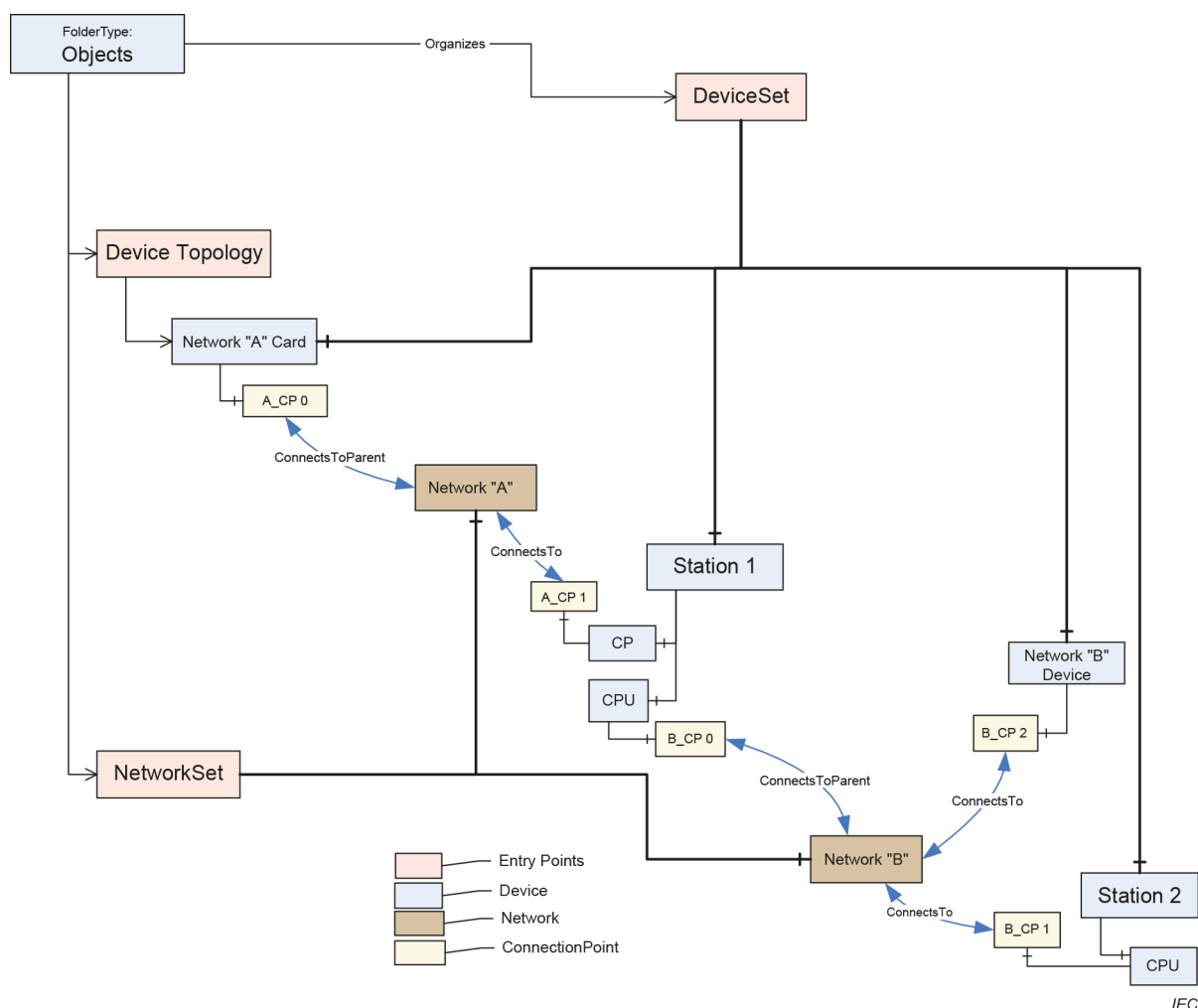


Figure 9 – Example of a Device topology

Coloured boxes are used to easily recognize the various types of information.

Brown boxes represent the networks. Light blue boxes represent the Devices and light yellow is used for Connection Points.

Light pink boxes represent the entry points that assure common behaviour across different implementations:

- DeviceTopology: Starting node for the topology configuration.
- DeviceSet: All instantiated Devices are components of this Object, i.e., they exist in the AddressSpace independently of the Device Topology.
- NetworkSet: All Networks are components of this Object.

5.3 Online/offline

Management of the Device Topology is a configuration task, i.e., the elements in the topology (Devices, Networks, and Connection Points) are usually configured "offline" and – at a later time – will be validated against their physical representative in a real network.

To support explicit access to either the online or the offline information, each element is represented by two instances that are schematically identical, i.e., there exists a ParameterSet, FunctionalGroups, and so on. A Reference connects the online and offline representation and allows navigating between them.

5.4 Catalogue (Type Definitions)

The supported (sub-types of) TopologyElements are organised as Type definitions in the OPC UA AddressSpace forming some kind of a catalogue. These definitions typically are generated based on descriptive information from FDI® Packages. The Type definitions contain the Parameters, and default values for Parameters, Methods, Actions and Functional Groups including user interface elements. The FDI® Server can include folders in the type model to organise the types according to manufacturer or other criteria.

Type definitions can then be used to create instances of Devices in the OPC UA AddressSpace. Instances can be created either offline or based on data determined by Scanning. Figure 10 illustrates an example of some Type definitions (details left out) as they can exist in the AddressSpace.

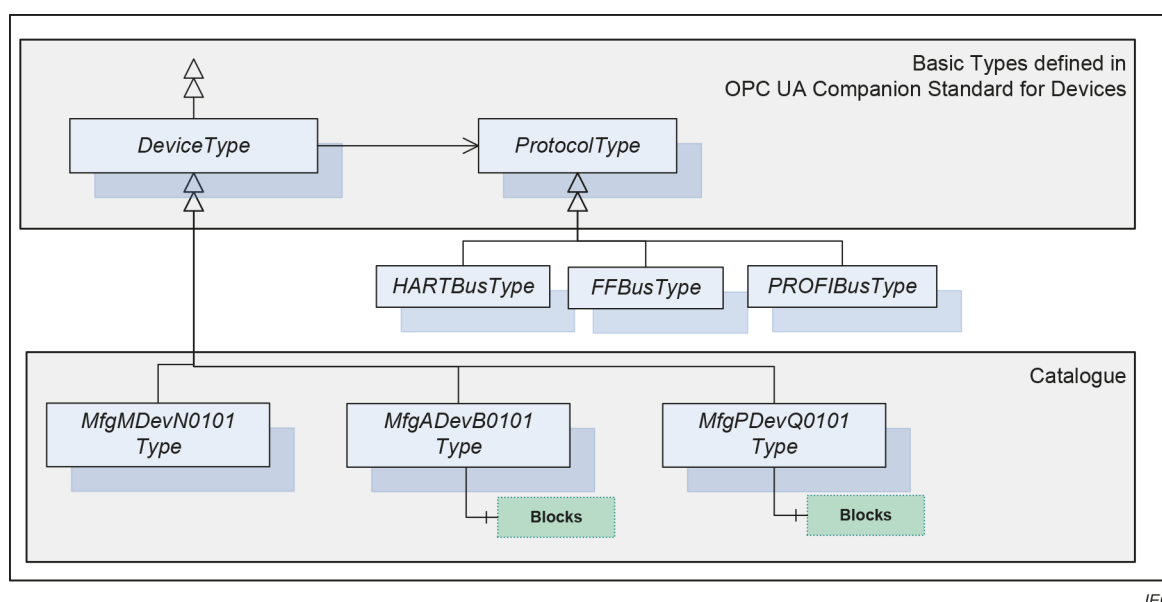


Figure 10 – Example Device Types representing a catalogue

5.5 Communication

In order to integrate Devices, the FDI® Server needs to be able to communicate to them. This can be done using Native Communication or Nested Communication.

The example in Figure 9 above for instance specifies that the FDI® Server has direct access to the PROFINET Network using its PROFINET network card. In order to access "Station 2", the FDI® Server shall go through Station 1, which provides the communication services for the PROFIBUS DP Network (see IEC 61784-1-3, CPF 3). This can be achieved through Nested Communication, which is specified in IEC 62769-3. Communication Devices and Communication Servers are specified in IEC 62769-7.

5.6 Semantic Information

The Type System of OPC UA already provides means to express the semantic of an Object. As an example, The Devices Companion Specification defines the DeviceType ObjectType expressing that instances of this ObjectType represent devices. However, the detailed semantic of the device is not specified further. Semantic information provides a means to represent that

an Object has the semantic differential pressure transmitter for instance. This allows clients to automatically retrieve and identify such devices.

Objects in OPC UA typically have Variables holding variable values. While the BrowseName or DisplayName provide a "hint" regarding the semantic of the Variable, this cannot be used to automatically identify the semantic of a Variable. As an example, the ParameterSet of a DeviceType Object contains a flat list of the parameters of the device. Semantic information provides means to represent that a Variable (e.g. in the ParameterSet) has a specific semantic associated with it. This allows clients to automatically retrieve and identify such Variables.

The DictionaryEntryType Object, defined in OPC-10000-19, is used to add semantic information to objects in the FDI® Information Model. Subtypes of DictionaryEntryType shall define their own namespace (e.g. <http://iec.ch/cdd/iec61987>). The NodeId of Objects shall consist of the corresponding namespace index and the value of the ID Property as the identifier part. This allows direct access to the Semantic Object without further indirection. Annex A defines the numeric identifiers for all of the numeric NodeIds used in this document.

The OPC UA Server shall create a concrete object type derived from the abstract DictionaryEntryType for each dictionary referenced. Objects such as DeviceType, FunctionalGroupType, or VariableType can reference one or more DictionaryEntryType objects using the HasDictionaryEntry reference to indicate the semantic meaning for the object. The DictionaryId property of the DictionaryEntryType object contains the identifier value of an external dictionary entry. A client can use these references to access device data with only needing to know semantically what information is desired (i.e. specific object BrowseNames are not required).

The following example provides an overview of this design principle (Figure 11). This example uses the IEC Common Data Dictionary (CDD); however, it should be noted that any qualifying dictionary can be used. The IEC Common Data Dictionary (CDD) describes classes and properties with the purpose to characterize equipment. The IEC 61987 series for instance describes process automation equipment. Every class and property provide a set of attributes such as Version, Revision, Preferred name, Definition, etc. Most importantly, it describes a unique identifier for every class of property. In the case of IEC CDD this is an International Registration Data Identifier (IRDI). In this example, IEC61987DictionaryEntryType is defined as a subtype of DictionaryEntryType. The Object DifferentialPressureTransmitter of this ObjectType contains the IEC CDD attribute values as its property values. The IRDI is set as the value of the ID property.

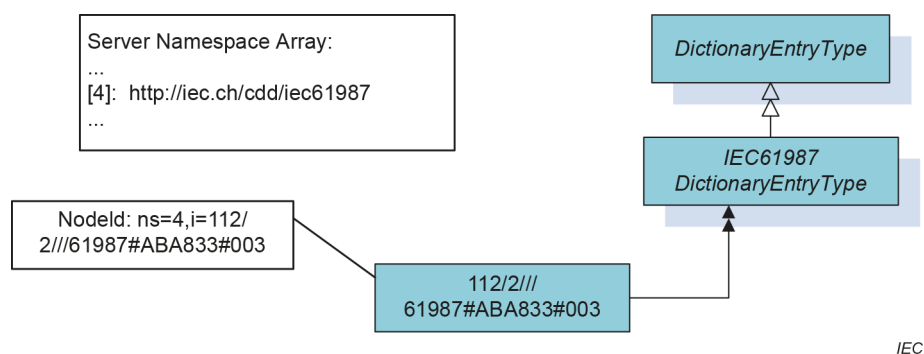


Figure 11 – Example of concrete DictionaryEntryType and Object

Variables can be enumerations. In such cases, the Variable can have a specific semantic as well as each enumeration item of the Variable. As an example, a Variable can contain the units of the measurement value that can be configured. So, the semantic of the Variable itself is "measurement unit". The enumeration items of that Variable then stand for a specific unit of measure. In case of temperature, the first enumeration item can represent degrees Celsius, the second item represents degrees Fahrenheit, the third item represents degrees Kelvin and so on. Associating enumeration items with a specific semantic allows clients to automatically