IEC 62541-100

Edition 1.0 2015-03

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

OPC unified architecture –
Part 100: Device Interface

Architecture unifiée OPC –
Partie 100: Interface d'appareils

IEC 62541-100:2015-03(en-fr)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
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 corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue
The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub
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 It also gives information on projects, replaced and withdrawn publications.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 also once a month by email.

Electropedia - www.electropedia.org
The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in 15 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary
More than 60 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

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: csc@iec.ch.

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.

Catalogue IEC - webstore.iec.ch/catalogue
Application autonome pour consulter tous les renseignements bibliographiques sur les Normes internationales, Spécifications techniques, Rapports techniques et autres documents de l'IEC. Disponible pour PC, Mac OS, tablettes Android et iPad.

Recherche de publications IEC - www.iec.ch/searchpub
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 aussi une fois par mois par email.

Electropedia - www.electropedia.org
Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 15 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary
Plus de 60 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

Service Clients - webstore.iec.ch/csc
Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: csc@iec.ch.

IEC 62541-100

Edition 1.0 2015-03

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

OPC unified architecture – Part 100: Device Interface

Architecture unifiée OPC – Partie 100: Interface d'appareils

INTERNATIONAL ELECTROTECHNICAL COMMISSION

COMMISSION ELECTROTECHNIQUE INTERNATIONALE

ICS 25.040.40; 35.100

ISBN 978-2-8322-2299-7

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 Reference documents ... 8
3 Terms, definitions, abbreviations and used data types ... 8
3.1 Terms and definitions ... 8
3.2 Abbreviations ... 10
3.3 Used data types ... 10
4 Fundamentals ... 10
4.1 OPC UA ... 10
4.2 Conventions used in this document ... 11
4.2.1 Conventions for Node descriptions ... 11
4.2.2 NodeIds and BrowseNames ... 12
5 Device model ... 13
5.1 General ... 13
5.2 TopologyElementType ... 14
5.3 FunctionalGroupType ... 16
5.4 Identification FunctionalGroup ... 18
5.5 UIElement Type ... 19
5.6 DeviceType ... 19
5.7 Device support information ... 22
5.7.1 General ... 22
5.7.2 DeviceTypeImage ... 23
5.7.3 Documentation ... 23
5.7.4 Protocol support files ... 23
5.7.5 Images ... 24
5.8 DeviceSet entry point ... 24
5.9 ProtocolType ... 25
5.10 BlockType ... 26
5.11 Configurable components ... 28
5.11.1 General pattern ... 28
5.11.2 ConfigurableObjectType ... 28
6 Device vommunication model ... 29
6.1 General ... 29
6.2 Network ... 30
6.3 ConnectionPoint ... 31
6.4 ConnectsTo and ConnectsToParent ReferenceTypes ... 33
6.5 NetworkSet Object (mandatory) ... 34
7 Device integration host model ... 35
7.1 General ... 35
7.2 DeviceTopology Object ... 36
7.3 Online/Offline ... 37
7.3.1 General ... 37
7.3.2 IsOnline ReferenceType ... 38
8 AddIn Capabilities ... 39
8.1 Overview ... 39

8.2 Offline-Online data transfer 40
8.2.1 Definition 40
8.2.2 TransferServices Type 40
8.2.3 TransferServices Object 41
8.2.4 TransferToDevice Method 41
8.2.5 TransferFromDevice Method 42
8.2.6 FetchTransferResultData Method 43
8.3 Locking 45
8.3.1 Overview 45
8.3.2 LockingServices Type 45
8.3.3 LockingServices Object 47
8.3.4 MaxInactiveLockTime Property 47
8.3.5 InitLock Method 48
8.3.6 ExitLock Method 48
8.3.7 RenewLock Method 49
8.3.8 BreakLock Method 49
9 Specialized topology elements 50
9.1 General 50
9.2 Block Devices (BlockOriented DeviceType) 50
9.3 Modular Devices 51
10 Profiles 52
10.1 General 52
10.2 Device Server Facets 52
10.3 Device Client Facets 53
Annex A (normative) Namespace and mappings 55
Bibliography 56

Figure 1 – Device model overview 13
Figure 2 – Components of the TopologyElementType 14
Figure 3 – FunctionalGroupType 16
Figure 4 – Analyser Device use for FunctionalGroups (UA Companion ADI) 17
Figure 5 – PLCopen use for FunctionalGroups (UA Companion PLCopen) 18
Figure 6 – Example of an Identification FunctionalGroup 19
Figure 7 – DeviceType 20
Figure 8 – Integration of support information within a DeviceType 22
Figure 9 – Standard entry point for Devices 25
Figure 10 – Example of a ProtocolType hierarchy with instances that represent specific communication profiles 26
Figure 11 – BlockType hierarchy 27
Figure 12 – Configurable component pattern 28
Figure 13 – ConfigurableObjectType 29
Figure 14 – Initial example of a communication topology 30
Figure 15 – NetworkType 30
Figure 16 – Example of ConnectionPointType hierarchy 31
Figure 17 – ConnectionPointType 32
Figure 18 – ConnectionPoint usage 33

Figure 19 – Type hierarchy for ConnectsTo and ConnectsToParent References 33
Figure 20 – Example with ConnectsTo and ConnectsToParent References 34
Figure 21 – Example of an automation system 35
Figure 22 – Example of a Device topology 36
Figure 23 – Online component for access to device data 37
Figure 24 – Type hierarchy for *IsOnline Reference* 39
Figure 25 – TransferServicesType 40
Figure 26 – TransferServices 41
Figure 27 – LockingServicesType 46
Figure 28 – LockingServices 47
Figure 29 – Block-oriented Device structure example 50
Figure 30 – Modular Device structure example 51

Table 1 – DataTypes defined in IEC 62541-3 10
Table 2 – Type definition table 11
Table 3 – Examples of DataTypes 12
Table 4 – TopologyElementType definition 15
Table 5 – ParameterSet definition 15
Table 6 – MethodSet definition 15
Table 7 – FunctionalGroupType definition 16
Table 8 – UIElementType definition 19
Table 9 – DeviceType definition 20
Table 10 – DeviceHealth values 22
Table 11 – DeviceTypeImage definition 23
Table 12 – Documentation definition 23
Table 13 – ProtocolSupport definition 23
Table 14 – ImageSet definition 24
Table 15 – DeviceSet definition 25
Table 16 – ProtocolType definition 26
Table 17 – BlockType definition 27
Table 18 – ConfigurableObjectType definition 29
Table 19 – NetworkType definition 31
Table 20 – ConnectionPointType definition 32
Table 21 – ConnectsTo ReferenceType 34
Table 22 – ConnectsToParent ReferenceType 34
Table 23 – NetworkSet definition 34
Table 24 – DeviceTopology definition 37
Table 25 – IsOnline *ReferenceType* 39
Table 26 – TransferServicesType definition 40
Table 27 – TransferToDevice Method arguments 42
Table 28 – TransferToDevice Method AddressSpace definition 42
Table 29 – TransferFromDevice Method arguments 42
Table 30 – TransferFromDevice Method AddressSpace definition 43

Table 31 – FetchTransferResultData Method Arguments 44
Table 32 – FetchTransferResultData Method AddressSpace definition 44
Table 33 – FetchResultDataType structure 44
Table 34 – TransferResultError DataType structure 44
Table 35 – TransferResultData DataType structure 45
Table 36 – LockingServicesType definition 46
Table 37 – MaxInactiveLockTime Property definition 47
Table 38 – InitLock Method Arguments 48
Table 39 – InitLock Method AddressSpace definition 48
Table 40 – ExitLock Method Arguments 49
Table 41 – ExitLock Method AddressSpace definition 49
Table 42 – RenewLock Method Arguments 49
Table 43 – RenewLock Method AddressSpace definition 49
Table 44 – BreakLock Method Arguments 50
Table 45 – BreakLock Method AddressSpace definition 50
Table 46 – BaseDevice_Server_Facet definition 52
Table 47 – DeviceIdentification_Server_Facet definition 52
Table 48 – BlockDevice_Server_Facet definition 52
Table 49 – Locking_Server_Facet definition 52
Table 50 – DeviceCommunication_Server_Facet definition 53
Table 51 – DeviceIntegrationHost_Server_Facet definition 53
Table 52 – BaseDevice_Client_Facet definition 53
Table 53 – DeviceIdentification_Client_Facet definition 53
Table 54 – BlockDevice_Client_Facet definition 54
Table 55 – Locking_Client_Facet definition 54
Table 56 – DeviceCommunication_Client_Facet definition 54
Table 57 – DeviceIntegrationHost_Client_Facet definition 54

INTERNATIONAL ELECTROTECHNICAL COMMISSION

OPC UNIFIED ARCHITECTURE –

Part 100: Device Interface

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.

International Standard IEC 62541-100 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

The text of this standard is based on the following documents:

CDV	Report on voting
65E/372/CDV	65E/412/RVC

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

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts of the IEC 62541 series, published under the general title , can be found on the IEC website.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be

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

IMPORTANT – The 'colour inside' logo on the cover page of this publication 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.

OPC UNIFIED ARCHITECTURE –

Part 100: Device Interface

1 Scope

This part of IEC 62541 is an extension of the overall OPC Unified Architecture standard series and defines the information model associated with *Devices*. This part of IEC 62541 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.

2 Reference documents

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC TR 62541-1, *OPC Unified Architecture – Part 1: Overview and Concepts*

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 Architecure – Part 5: Information Model*

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

IEC 62541-7, *OPC Unified Architecure – Part 7: Profiles*

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

NAMUR Recommendation NE107: *Self-monitoring and diagnosis of field devices*

3 Terms, definitions, abbreviations and used data types

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC TR 62541-1, IEC 62541-3, and IEC 62541-8 as well as the following apply.

3.1.1
block
functional *Parameter* grouping entity

Note 1 to entry: It could map to a function block (see IEC 62769 (all parts), *Field Device Integration (FDI)*

) or to the resource parameters of the device itself.

3.1.2
blockMode
mode of operation (target mode, permitted modes, actual mode, and normal mode) for a *Block*

Note 1 to entry: Further details about *Block* modes are defined by standard organisations.

3.1.3
Communication Profile
fixed set of mapping rules to allow unambiguous interoperability between *Devices* or Applications, respectively

Note 1 to entry: Examples of such profiles are the "Wireless communication network and communication profiles for WirelessHART" in IEC 62591 and the Protocol Mappings for OPC UA in IEC 62541-6.

3.1.4
Connection Point
logical representation of the interface between a *Device* and a *Network*

3.1.5
device
independent physical entity capable of performing one or more specified functions in a particular context and delimited by its interfaces

Note 1 to entry: See IEC 61499-1.

Note 2 to entry: *Devices* provide sensing, actuating, communication, and/or control functionality. Examples include transmitters, valve controllers, drives, motor controllers, PLCs, and communication gateways.

3.1.6
Device Integration Host
Server that manages integration of multiple *Devices* in an automation system

3.1.7
Device Topology
arrangement of *Networks* and *Devices* that constitute a communication topology

3.1.8
fieldbus
communication system based on serial data transfer and used in industrial automation or process control applications

Note 1 to entry: See IEC 61784.

Note 2 to entry: Designates the communication bus used by a *Device*.

3.1.9
parameter
variable of the *Device* that can be used for configuration, monitoring or control purposes

Note 1 to entry: In the information model it is synonymous to an OPC UA *DataVariable*.

3.1.10
network
means used to communicate with one specific protocol

3.2 Abbreviations

ADI Analyser Device Integration

CP Communication Processor (hardware module)

CPU Central Processing Unit (of a *Device*)

DA Data Access

DI Device Integration (the short name for this standard)

UA Unified Architecture

UML Unified Modelling Language

XML Extensible Mark-up Language

3.3 Used data types

Table 1 describes the *DataTypes* that are used throughout this document.

Table 1 – DataTypes defined in IEC 62541-3

Parameter Type
Argument
Boolean
Duration
LocalizedText
String
Int32

4 Fundamentals

4.1 OPC UA

The main use case for OPC standards is the online data exchange between devices and HMI or SCADA systems using Data Access functionality. 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 DA provides functionality to browse through hierarchical namespaces containing data items and to read, write and to monitor these items for data changes. The classic OPC standards are based on Microsoft COM/DCOM technology for the communication between software components from different vendors. Therefore classic OPC *Server* and *Clients* are restricted to Windows PC based automation systems.

OPC UA incorporates all features of classic OPC standards like OPC DA, A&E and HDA but defines platform independent communication mechanisms and generic, extensible and object-oriented modelling capabilities for the information a system wants to expose.

The OPC UA network communication part defines different mechanisms optimized for different use cases. The current version of OPC UA is defining an optimized binary protocol for high performance intranet communication as well as Web Services. It 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 eight classes of *Nodes* to represent *AddressSpace* components. The classes are *Object*, *Variable*, *Method*, *ObjectType*, *VariableType*, *DataType*, *ReferenceType* and *View*. Each *NodeClass* has a defined set of *Attributes*.

This standard makes use of almost all OPC UA *NodeClasses*.

Objects are used to represent real-world entities such as *Devices* and (communication) Networks as well as software entities such as *Blocks*. 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. Examples for *Properties* of *Objects* are the device serial number and the block tag.

DataVariables represent the contents of an *Object*. *DataVariables* may have component *DataVariables*. This is typically used by *Servers* to expose individual elements of arrays and structures. This standard uses *DataVariables* to represent the *Parameters* of both *Blocks* and *Devices*.

4.2 Conventions used in this document

4.2.1 Conventions for Node descriptions

Node definitions are specified using tables (see Table 2).

Table 2 – Type definition table

Attribute	**Value**				
Attribute name	Attribute value. If it is an optional Attribute that is not set "--" will be used.				
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
ReferenceType name	*NodeClass* of the Target*Node*.	*BrowseName* of the target *Node*. If the *Reference* is to be instantiated by the server, then the value of the target Node's BrowseName is "--".	*Attributes* of the referenced *Node*, only applicable for *Variables* and *Objects*.		Referenced *ModellingRule* of the referenced *Object*.
NOTE Notes referencing footnotes of the table content.					

Attributes are defined by providing the *Attribute* name and a value, or a description of the value.

References are defined by providing the *ReferenceType* name, the *BrowseName* of the *TargetNode* and its *NodeClass*.

- If the *TargetNode* is a component of the *Node* being defined in the table the *Attributes* of the composed *Node* are defined in the same row of the table.
- The *DataType* is only specified for *Variables*; "[<number>]" indicates a single-dimensional array, for multi-dimensional arrays the expression is repeated for each dimension (e.g. [2][3] for a two-dimensional array). For all arrays the *ArrayDimensions* is set as identified by <number> values. If no <number> is set, the corresponding dimension is set to 0, indicating an unknown size. If no number is provided at all the *ArrayDimensions* can be omitted. If no brackets are provided, it identifies a scalar *DataType* and the *ValueRank* is set to the corresponding value (see IEC 62541-3). In

addition, *ArrayDimensions* is set to null or is omitted. If it can be *Any* or *ScalarOrOneDimension*, the value is put into “{<value>}”, so either “{*Any*}” or “{*ScalarOrOneDimension*}”. The *ValueRank* is set to the corresponding value (see IEC 62541-3) and the *ArrayDimensions* is set to null or is omitted. In Table 3 examples are given.

Table 3 – Examples of DataTypes

Notation	Data-Type	Value-Rank	ArrayDimensions	Description
Int32	Int32	-1	omitted or NULL	A scalar Int32
Int32[]	Int32	1	omitted or {0}	Single-dimensional array of Int32 with an unknown size
Int32[][]	Int32	2	omitted or {0,0}	Two-dimensional array of Int32 with unknown sizes for both dimensions
Int32[3][]	Int32	2	{3,0}	Two-dimensional array of Int32 with a size of 3 for the first dimension and an unknown size for the second dimension
Int32[5][3]	Int32	2	{5,3}	Two-dimensional array of Int32 with a size of 5 for the first dimension and a size of 3 for the second dimension
Int32{Any}	Int32	-2	omitted or NULL	An Int32 where it is unknown if it is scalar or array with any number of dimensions
Int32{ScalarOrOneDimension}	Int32	-3	omitted or NULL	An Int32 where it is either a single-dimensional array or a scalar

- The TypeDefinition is specified for *Objects* and *Variables*.
- The TypeDefinition column specifies a *NodeId* of a *TypeDefinitionNode*, i.e. the specified *Node* points with a *HasTypeDefinition Reference* to the corresponding *TypeDefinitionNode*. The symbolic name of the *NodeId* is used in the table.
- The *ModellingRule* of the referenced component is provided by specifying the symbolic name of the rule in the *ModellingRule* column. In the *AddressSpace*, the *Node* shall use a *HasModellingRule Reference* to point to the corresponding *ModellingRule Object*.

If the *NodeId* of a *DataType* is provided, the symbolic name of the *Node* representing the *DataType* shall be used.

If no components are provided, the *DataType*, *TypeDefinition* and *ModellingRule* columns may be omitted and only a *Comment* column is introduced to point to the *Node* definition.

Components of *Nodes* can be complex, i.e. contain components by themselves. The *TypeDefinition*, *NodeClass*, *DataType* and *ModellingRule* can be derived from the type definitions, and the symbolic name can be created as defined in 4.2.2.1. Therefore those containing components are not explicitly specified; they are implicitly specified by the type definitions.

4.2.2 NodeIds and BrowseNames

4.2.2.1 NodeIds

The *NodeIds* of all *Nodes* described in this document are only symbolic names. Annex A defines the actual *NodeIds*.

The symbolic name of each *Node* defined in this document is its *BrowseName*, or, when it is part of another *Node*, the *BrowseName* of the other *Node*, a “.”, and the *BrowseName* of itself. In this case “part of” means that the whole has a *HasProperty* or *HasComponent Reference* to its part. Since all *Nodes* not being part of another *Node* have a unique name in this document, the symbolic name is unique.

The namespace for this standard is defined in Annex A. The *NamespaceIndex* for all *NodeIds* and *BrowseNames* defined in this standard is server specific and depends on the position of the namespace URI in the server namespace table.

4.2.2.2 BrowseNames

The text part of the *BrowseNames* for all *Nodes* defined in this standard is specified in the tables defining the *Nodes*. The *NamespaceIndex* for all *BrowseNames* defined in this standard is server specific and depends on the position of the namespace URI in the server namespace table.

5 Device model

5.1 General

Figure 1 depicts the main *ObjectTypes* of the base *Device* model and their relationship. The drawing is not intended to be complete. For the sake of simplicity only a few components and relations were captured so as to give a rough idea of the overall structure.

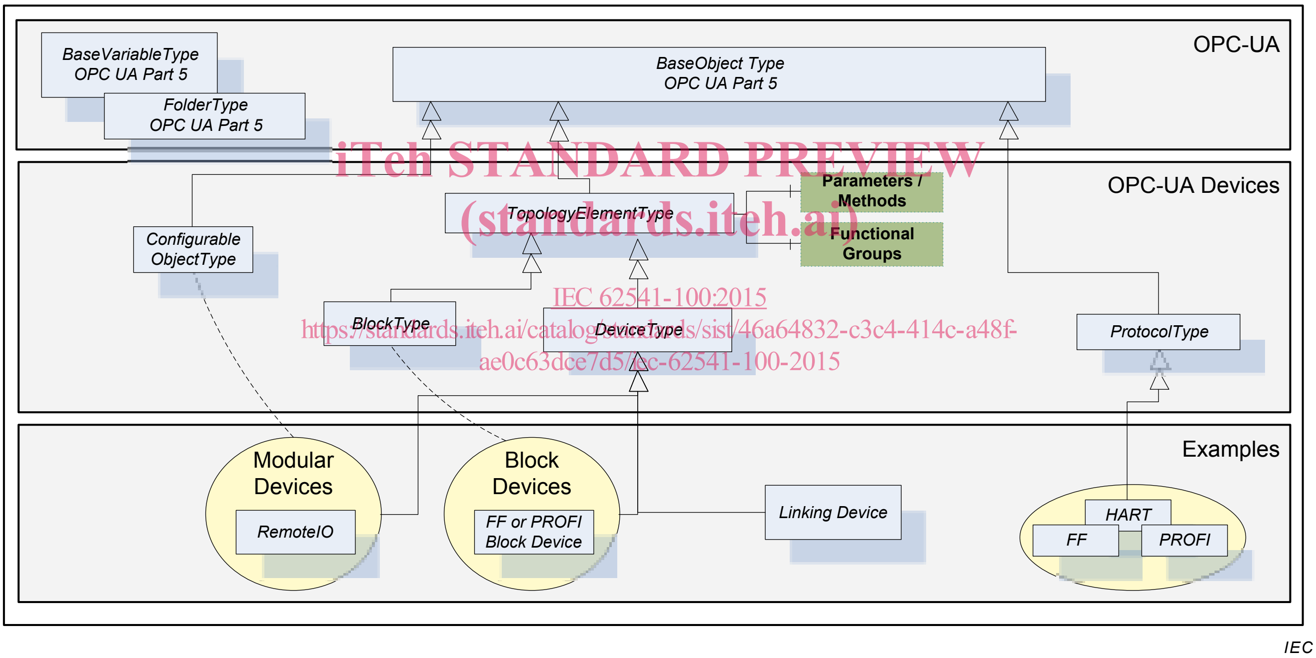

Figure 1 – Device model overview

The boxes in this drawing show the *ObjectTypes* used in this standard as well as some elements from other standards that help understand some modelling decisions. The upper grey box shows the OPC UA core *ObjectTypes* from which the *TopologyElementType* and *ProtocolType* are derived. The grey box in the second level shows the main *ObjectTypes* that this standard introduces. The components of those *ObjectTypes* are illustrated only in an abstract way in this overall picture.

The grey box in the third level shows real-world examples as they will be used in products and plants. In general, such subtypes are defined by other organizations.

The *TopologyElementType* is the base *ObjectType* for elements in a device topology. It introduces *Parameters* and *Methods*. This standard also defines a functional grouping concept to provide alternative viewpoints.