

# INTERNATIONAL STANDARD

## NORME INTERNATIONALE



**OPC unified architecture –  
Part 3: Address Space Model**

**Architecture unifiée OPC –  
Partie 3: Modèle de l'Espace d'Adressage**



## THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2015 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 Central Office  
3, rue de Varembe  
CH-1211 Geneva 20  
Switzerland

Tel.: +41 22 919 02 11  
Fax: +41 22 919 03 00  
[info@iec.ch](mailto:info@iec.ch)  
[www.iec.ch](http://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](http://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](http://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](http://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](http://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](http://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](http://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](mailto: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](http://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](http://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](http://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](http://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](http://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](http://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](mailto:csc@iec.ch).

# INTERNATIONAL STANDARD

## NORME INTERNATIONALE



**OPC unified architecture –  
Part 3: Address Space Model**

**Architecture unifiée OPC –  
Partie 3: Modèle de l'Espace d'Adressage**

INTERNATIONAL  
ELECTROTECHNICAL  
COMMISSION

COMMISSION  
ELECTROTECHNIQUE  
INTERNATIONALE

ICS 25.040.40; 35.100

ISBN 978-2-8322-2385-7

**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 .....                                                         | 10 |
| 1 Scope .....                                                          | 12 |
| 2 Normative references .....                                           | 12 |
| 3 Terms, definitions, abbreviations and conventions .....              | 13 |
| 3.1 Terms and definitions .....                                        | 13 |
| 3.2 Abbreviations .....                                                | 14 |
| 3.3 Conventions .....                                                  | 14 |
| 3.3.1 Conventions for AddressSpace figures .....                       | 14 |
| 3.3.2 Conventions for defining NodeClasses .....                       | 14 |
| 4 AddressSpace concepts .....                                          | 16 |
| 4.1 Overview .....                                                     | 16 |
| 4.2 Object Model .....                                                 | 16 |
| 4.3 Node Model .....                                                   | 16 |
| 4.3.1 General .....                                                    | 16 |
| 4.3.2 NodeClasses .....                                                | 17 |
| 4.3.3 Attributes .....                                                 | 17 |
| 4.3.4 References .....                                                 | 17 |
| 4.4 Variables .....                                                    | 18 |
| 4.4.1 General .....                                                    | 18 |
| 4.4.2 Properties .....                                                 | 18 |
| 4.4.3 DataVariables .....                                              | 18 |
| 4.5 TypeDefinitionNodes .....                                          | 19 |
| 4.5.1 General .....                                                    | 19 |
| 4.5.2 Complex TypeDefinitionNodes and their InstanceDeclarations ..... | 20 |
| 4.5.3 Subtyping .....                                                  | 20 |
| 4.5.4 Instantiation of complex TypeDefinitionNodes .....               | 21 |
| 4.6 Event Model .....                                                  | 22 |
| 4.6.1 General .....                                                    | 22 |
| 4.6.2 EventTypes .....                                                 | 22 |
| 4.6.3 Event Categorization .....                                       | 23 |
| 4.7 Methods .....                                                      | 23 |
| 5 Standard NodeClasses .....                                           | 23 |
| 5.1 Overview .....                                                     | 23 |
| 5.2 Base NodeClass .....                                               | 24 |
| 5.2.1 General .....                                                    | 24 |
| 5.2.2 NodeId .....                                                     | 24 |
| 5.2.3 NodeClass .....                                                  | 24 |
| 5.2.4 BrowseName .....                                                 | 24 |
| 5.2.5 DisplayName .....                                                | 25 |
| 5.2.6 Description .....                                                | 25 |
| 5.2.7 WriteMask .....                                                  | 25 |
| 5.2.8 UserWriteMask .....                                              | 26 |
| 5.3 ReferenceType NodeClass .....                                      | 26 |
| 5.3.1 General .....                                                    | 26 |
| 5.3.2 Attributes .....                                                 | 27 |
| 5.3.3 References .....                                                 | 28 |

|       |                                                                                       |    |
|-------|---------------------------------------------------------------------------------------|----|
| 5.4   | View NodeClass.....                                                                   | 29 |
| 5.5   | Objects .....                                                                         | 31 |
| 5.5.1 | Object NodeClass .....                                                                | 31 |
| 5.5.2 | ObjectType NodeClass .....                                                            | 33 |
| 5.5.3 | Standard ObjectType FolderType.....                                                   | 35 |
| 5.5.4 | Client-side creation of Objects of an ObjectType.....                                 | 35 |
| 5.6   | Variables .....                                                                       | 35 |
| 5.6.1 | General.....                                                                          | 35 |
| 5.6.2 | Variable NodeClass .....                                                              | 35 |
| 5.6.3 | Properties .....                                                                      | 39 |
| 5.6.4 | DataVariable.....                                                                     | 39 |
| 5.6.5 | VariableType NodeClass.....                                                           | 40 |
| 5.6.6 | Client-side creation of Variables of an VariableType.....                             | 42 |
| 5.7   | Method NodeClass.....                                                                 | 42 |
| 5.8   | DataTypes .....                                                                       | 44 |
| 5.8.1 | DataType Model.....                                                                   | 44 |
| 5.8.2 | Encoding Rules for different kinds of DataTypes .....                                 | 46 |
| 5.8.3 | DataType NodeClass .....                                                              | 47 |
| 5.8.4 | DataTypeDictionary, DataTypeDescription, DataTypeEncoding and<br>DataTypeSystem ..... | 48 |
| 5.9   | Summary of Attributes of the NodeClasses .....                                        | 50 |
| 6     | Type Model for ObjectTypes and VariableTypes .....                                    | 51 |
| 6.1   | Overview.....                                                                         | 51 |
| 6.2   | Definitions.....                                                                      | 51 |
| 6.2.1 | InstanceDeclaration .....                                                             | 51 |
| 6.2.2 | Instances without ModellingRules .....                                                | 51 |
| 6.2.3 | InstanceDeclarationHierarchy.....                                                     | 52 |
| 6.2.4 | Similar Node of InstanceDeclaration .....                                             | 52 |
| 6.2.5 | BrowsePath.....                                                                       | 52 |
| 6.2.6 | Attribute Handling of InstanceDeclarations.....                                       | 52 |
| 6.2.7 | Attribute Handling of Variable and VariableTypes .....                                | 52 |
| 6.2.8 | NodeIds of InstanceDeclarations.....                                                  | 52 |
| 6.3   | Subtyping of ObjectTypes and VariableTypes .....                                      | 53 |
| 6.3.1 | Overview .....                                                                        | 53 |
| 6.3.2 | Attributes .....                                                                      | 53 |
| 6.3.3 | InstanceDeclarations .....                                                            | 53 |
| 6.4   | Instances of ObjectTypes and VariableTypes.....                                       | 56 |
| 6.4.1 | Overview .....                                                                        | 56 |
| 6.4.2 | Creating an Instance.....                                                             | 57 |
| 6.4.3 | Constraints on an Instance .....                                                      | 57 |
| 6.4.4 | ModellingRules .....                                                                  | 58 |
| 6.5   | Changing Type Definitions that are already used .....                                 | 66 |
| 7     | Standard ReferenceTypes .....                                                         | 66 |
| 7.1   | General.....                                                                          | 66 |
| 7.2   | References ReferenceType.....                                                         | 67 |
| 7.3   | HierarchicalReferences ReferenceType .....                                            | 67 |
| 7.4   | NonHierarchicalReferences ReferenceType .....                                         | 68 |
| 7.5   | HasChild ReferenceType .....                                                          | 68 |
| 7.6   | Aggregates ReferenceType.....                                                         | 68 |

|       |                                         |    |
|-------|-----------------------------------------|----|
| 7.7   | HasComponent ReferenceType.....         | 68 |
| 7.8   | HasProperty ReferenceType .....         | 69 |
| 7.9   | HasOrderedComponent ReferenceType ..... | 69 |
| 7.10  | HasSubtype ReferenceType.....           | 69 |
| 7.11  | Organizes ReferenceType.....            | 69 |
| 7.12  | HasModellingRule ReferenceType .....    | 70 |
| 7.13  | HasTypeDefinition ReferenceType .....   | 70 |
| 7.14  | HasEncoding ReferenceType .....         | 70 |
| 7.15  | HasDescription ReferenceType .....      | 70 |
| 7.16  | GeneratesEvent .....                    | 71 |
| 7.17  | AlwaysGeneratesEvent .....              | 71 |
| 7.18  | HasEventSource .....                    | 71 |
| 7.19  | HasNotifier.....                        | 71 |
| 8     | Standard DataTypes .....                | 73 |
| 8.1   | General.....                            | 73 |
| 8.2   | NodeId.....                             | 73 |
| 8.2.1 | General.....                            | 73 |
| 8.2.2 | NamespaceIndex .....                    | 73 |
| 8.2.3 | IdentifierType.....                     | 74 |
| 8.2.4 | Identifier value .....                  | 74 |
| 8.3   | QualifiedName .....                     | 75 |
| 8.4   | LocaleId.....                           | 75 |
| 8.5   | LocalizedText .....                     | 76 |
| 8.6   | Argument.....                           | 76 |
| 8.7   | BaseDataType .....                      | 76 |
| 8.8   | Boolean .....                           | 76 |
| 8.9   | Byte.....                               | 76 |
| 8.10  | ByteString.....                         | 77 |
| 8.11  | DateTime.....                           | 77 |
| 8.12  | Double.....                             | 77 |
| 8.13  | Duration.....                           | 77 |
| 8.14  | Enumeration .....                       | 77 |
| 8.15  | Float.....                              | 77 |
| 8.16  | Guid.....                               | 77 |
| 8.17  | SByte.....                              | 77 |
| 8.18  | IdType .....                            | 77 |
| 8.19  | Image .....                             | 77 |
| 8.20  | ImageBMP .....                          | 78 |
| 8.21  | ImageGIF.....                           | 78 |
| 8.22  | ImageJPG.....                           | 78 |
| 8.23  | ImagePNG .....                          | 78 |
| 8.24  | Integer .....                           | 78 |
| 8.25  | Int16 .....                             | 78 |
| 8.26  | Int32 .....                             | 78 |
| 8.27  | Int64 .....                             | 78 |
| 8.28  | TimeZoneDataType.....                   | 78 |
| 8.29  | NamingRuleType .....                    | 78 |
| 8.30  | NodeClass .....                         | 79 |
| 8.31  | Number.....                             | 79 |

|        |                                            |    |
|--------|--------------------------------------------|----|
| 8.32   | String.....                                | 79 |
| 8.33   | Structure.....                             | 79 |
| 8.34   | UInteger.....                              | 79 |
| 8.35   | UInt16.....                                | 79 |
| 8.36   | UInt32.....                                | 79 |
| 8.37   | UInt64.....                                | 79 |
| 8.38   | UtcTime .....                              | 80 |
| 8.39   | XmlElement .....                           | 80 |
| 8.40   | EnumValueType.....                         | 80 |
| 9      | Standard EventTypes .....                  | 80 |
| 9.1    | General.....                               | 80 |
| 9.2    | BaseEventType.....                         | 81 |
| 9.3    | SystemEventType .....                      | 81 |
| 9.4    | ProgressEventType.....                     | 81 |
| 9.5    | AuditEventType .....                       | 82 |
| 9.6    | AuditSecurityEventType .....               | 83 |
| 9.7    | AuditChannelEventType.....                 | 83 |
| 9.8    | AuditOpenSecureChannelEventType .....      | 83 |
| 9.9    | AuditSessionEventType .....                | 83 |
| 9.10   | AuditCreateSessionEventType.....           | 84 |
| 9.11   | AuditUrlMismatchEventType.....             | 84 |
| 9.12   | AuditActivateSessionEventType.....         | 84 |
| 9.13   | AuditCancelEventType.....                  | 84 |
| 9.14   | AuditCertificateEventType.....             | 84 |
| 9.15   | AuditCertificateDataMismatchEventType..... | 84 |
| 9.16   | AuditCertificateExpiredEventType.....      | 84 |
| 9.17   | AuditCertificateInvalidEventType.....      | 84 |
| 9.18   | AuditCertificateUntrustedEventType.....    | 84 |
| 9.19   | AuditCertificateRevokedEventType.....      | 84 |
| 9.20   | AuditCertificateMismatchEventType.....     | 85 |
| 9.21   | AuditNodeManagementEventType .....         | 85 |
| 9.22   | AuditAddNodesEventType .....               | 85 |
| 9.23   | AuditDeleteNodesEventType.....             | 85 |
| 9.24   | AuditAddReferencesEventType.....           | 85 |
| 9.25   | AuditDeleteReferencesEventType.....        | 85 |
| 9.26   | AuditUpdateEventType .....                 | 85 |
| 9.27   | AuditWriteUpdateEventType .....            | 85 |
| 9.28   | AuditHistoryUpdateEventType .....          | 85 |
| 9.29   | AuditUpdateMethodEventType.....            | 85 |
| 9.30   | DeviceFailureEventType .....               | 85 |
| 9.31   | SystemStatusChangeEventType .....          | 86 |
| 9.32   | ModelChangeEvents .....                    | 86 |
| 9.32.1 | General.....                               | 86 |
| 9.32.2 | NodeVersion Property.....                  | 86 |
| 9.32.3 | Views.....                                 | 86 |
| 9.32.4 | Event Compression.....                     | 86 |
| 9.32.5 | BaseModelChangeEventType .....             | 86 |
| 9.32.6 | GeneralModelChangeEventType.....           | 87 |
| 9.32.7 | Guidelines for ModelChangeEvents .....     | 87 |

|                       |                                                    |     |
|-----------------------|----------------------------------------------------|-----|
| 9.33                  | SemanticChangeEventType .....                      | 87  |
| 9.33.1                | General.....                                       | 87  |
| 9.33.2                | ViewVersion and NodeVersion Properties .....       | 87  |
| 9.33.3                | Views.....                                         | 88  |
| 9.33.4                | Event Compression.....                             | 88  |
| Annex A (informative) | How to use the Address Space Model .....           | 89  |
| A.1                   | Overview.....                                      | 89  |
| A.2                   | Type definitions .....                             | 89  |
| A.3                   | ObjectTypes.....                                   | 89  |
| A.4                   | VariableTypes.....                                 | 90  |
| A.4.1                 | General.....                                       | 90  |
| A.4.2                 | Properties or DataVariables .....                  | 90  |
| A.4.3                 | Many Variables and / or structured DataTypes ..... | 90  |
| A.5                   | Views.....                                         | 91  |
| A.6                   | Methods.....                                       | 91  |
| A.7                   | Defining ReferenceTypes .....                      | 91  |
| A.8                   | Defining ModellingRules .....                      | 91  |
| Annex B (informative) | OPC UA Meta Model in UML .....                     | 92  |
| B.1                   | Background.....                                    | 92  |
| B.2                   | Notation .....                                     | 92  |
| B.3                   | Meta Model.....                                    | 94  |
| B.3.1                 | Base .....                                         | 94  |
| B.3.2                 | ReferenceType .....                                | 94  |
| B.3.3                 | Predefined ReferenceTypes .....                    | 96  |
| B.3.4                 | Attributes .....                                   | 96  |
| B.3.5                 | Object and ObjectType.....                         | 97  |
| B.3.6                 | EventNotifier.....                                 | 98  |
| B.3.7                 | Variable and VariableType .....                    | 98  |
| B.3.8                 | Method .....                                       | 99  |
| B.3.9                 | DataType .....                                     | 100 |
| B.3.10                | View .....                                         | 101 |
| Annex C (normative)   | OPC Binary Type Description System .....           | 102 |
| C.1                   | Concepts.....                                      | 102 |
| C.2                   | Schema Description.....                            | 103 |
| C.2.1                 | TypeDictionary.....                                | 103 |
| C.2.2                 | TypeDescription.....                               | 103 |
| C.2.3                 | OpaqueType .....                                   | 104 |
| C.2.4                 | EnumeratedType .....                               | 104 |
| C.2.5                 | StructuredType .....                               | 105 |
| C.2.6                 | FieldType.....                                     | 105 |
| C.2.7                 | EnumeratedValue .....                              | 107 |
| C.2.8                 | ByteOrder .....                                    | 107 |
| C.2.9                 | ImportDirective .....                              | 107 |
| C.3                   | Standard Type Descriptions .....                   | 107 |
| C.4                   | Type Description Examples.....                     | 108 |
| C.5                   | OPC Binary XML Schema .....                        | 110 |
| C.6                   | OPC Binary Standard TypeDictionary .....           | 111 |
| Annex D (normative)   | Graphical Notation .....                           | 114 |

|            |                                                                                  |     |
|------------|----------------------------------------------------------------------------------|-----|
| D.1        | General.....                                                                     | 114 |
| D.2        | Notation .....                                                                   | 114 |
| D.2.1      | Overview .....                                                                   | 114 |
| D.2.2      | Simple Notation .....                                                            | 114 |
| D.2.3      | Extended Notation .....                                                          | 116 |
| Figure 1   | – AddressSpace Node diagrams .....                                               | 14  |
| Figure 2   | – OPC UA Object Model.....                                                       | 16  |
| Figure 3   | – AddressSpace Node Model .....                                                  | 17  |
| Figure 4   | – Reference Model.....                                                           | 18  |
| Figure 5   | – Example of a Variable defined by a VariableType.....                           | 19  |
| Figure 6   | – Example of a Complex TypeDefinition .....                                      | 20  |
| Figure 7   | – Object and its Components defined by an ObjectType.....                        | 21  |
| Figure 8   | – Symmetric and Non-Symmetric References.....                                    | 28  |
| Figure 9   | – Variables, VariableTypes and their DataTypes .....                             | 44  |
| Figure 10  | – DataType Model .....                                                           | 45  |
| Figure 11  | – Example of DataType Modelling .....                                            | 50  |
| Figure 12  | – Subtyping TypeDefinitionNodes.....                                             | 54  |
| Figure 13  | – The Fully-Inherited InstanceDeclarationHierarchy for BetaType .....            | 55  |
| Figure 14  | – An Instance and its TypeDefinitionNode .....                                   | 57  |
| Figure 15  | – Example for several References between InstanceDeclarations .....              | 58  |
| Figure 16  | – Example on changing instances based on InstanceDeclarations .....              | 60  |
| Figure 17  | – Example on changing InstanceDeclarations based on an InstanceDeclaration ..... | 61  |
| Figure 18  | – Use of the Standard ModellingRule New .....                                    | 62  |
| Figure 19  | – Example using the Standard ModellingRules Optional and Mandatory.....          | 63  |
| Figure 20  | – Example on using ExposesItsArray .....                                         | 64  |
| Figure 21  | – Complex example on using ExposesItsArray .....                                 | 64  |
| Figure 22  | – Example on using OptionalPlaceholder .....                                     | 65  |
| Figure 23  | – Example on using MandatoryPlaceholder .....                                    | 66  |
| Figure 24  | – Standard ReferenceType Hierarchy.....                                          | 67  |
| Figure 25  | – Event Reference Example .....                                                  | 72  |
| Figure 26  | – Complex Event Reference Example .....                                          | 73  |
| Figure 27  | – Standard EventType Hierarchy.....                                              | 81  |
| Figure 28  | – Audit Behaviour of a Server.....                                               | 82  |
| Figure 29  | – Audit Behaviour of an Aggregating Server .....                                 | 83  |
| Figure B.1 | – Background of OPC UA Meta Model .....                                          | 92  |
| Figure B.2 | – Notation (I) .....                                                             | 93  |
| Figure B.3 | – Notation (II) .....                                                            | 93  |
| Figure B.4 | – Base .....                                                                     | 94  |
| Figure B.5 | – Reference and ReferenceType.....                                               | 95  |
| Figure B.6 | – Predefined ReferenceTypes.....                                                 | 96  |
| Figure B.7 | – Attributes .....                                                               | 97  |
| Figure B.8 | – Object and ObjectType .....                                                    | 98  |

|                                                                               |     |
|-------------------------------------------------------------------------------|-----|
| Figure B.9 – EventNotifier .....                                              | 98  |
| Figure B.10 – Variable and VariableType .....                                 | 99  |
| Figure B.11 – Method .....                                                    | 100 |
| Figure B.12 – DataType .....                                                  | 100 |
| Figure B.13 – View .....                                                      | 101 |
| Figure C.1 – OPC Binary Dictionary Structure.....                             | 102 |
| Figure D.1 – Example of a Reference connecting two Nodes .....                | 115 |
| Figure D.2 – Example of using a TypeDefinition inside a Node .....            | 117 |
| Figure D.3 – Example of exposing Attributes.....                              | 117 |
| Figure D.4 – Example of exposing Properties inline .....                      | 118 |
| Table 1 – NodeClass Table Conventions.....                                    | 15  |
| Table 2 – Base NodeClass.....                                                 | 24  |
| Table 3 – Bit mask for WriteMask and UserWriteMask .....                      | 26  |
| Table 4 – ReferenceType NodeClass .....                                       | 27  |
| Table 5 – View NodeClass .....                                                | 30  |
| Table 6 – Object NodeClass .....                                              | 32  |
| Table 7 – ObjectType NodeClass .....                                          | 34  |
| Table 8 – Variable NodeClass.....                                             | 36  |
| Table 9 – VariableType NodeClass .....                                        | 41  |
| Table 10 – Method NodeClass .....                                             | 43  |
| Table 11 – DataType NodeClass.....                                            | 47  |
| Table 12 – Overview of Attributes .....                                       | 51  |
| Table 13 – The InstanceDeclarationHierarchy for BetaType .....                | 54  |
| Table 14 – The Fully-Inherited InstanceDeclarationHierarchy for BetaType..... | 55  |
| Table 15 – Rule for ModellingRules Properties when Subtyping .....            | 59  |
| Table 16 – Properties of ModellingRules .....                                 | 61  |
| Table 17 – NodeId Definition .....                                            | 73  |
| Table 18 – IdentifierType Values.....                                         | 74  |
| Table 19 – NodeId Null Values.....                                            | 75  |
| Table 20 – QualifiedName Definition .....                                     | 75  |
| Table 21 – LocaleId Examples .....                                            | 75  |
| Table 22 – LocalizedText Definition .....                                     | 76  |
| Table 23 – Argument Definition .....                                          | 76  |
| Table 24 – TimeZoneDataType Definition .....                                  | 78  |
| Table 25 – NamingRuleType Values .....                                        | 79  |
| Table 26 – NodeClass Values .....                                             | 79  |
| Table 27 – EnumValueType Definition .....                                     | 80  |
| Table C.1 – TypeDictionary Components .....                                   | 103 |
| Table C.2 – TypeDescription Components .....                                  | 104 |
| Table C.3 – OpaqueType Components.....                                        | 104 |
| Table C.4 – EnumeratedType Components .....                                   | 105 |
| Table C.5 – StructuredType Components.....                                    | 105 |

|                                                                       |     |
|-----------------------------------------------------------------------|-----|
| Table C.6 – FieldType Components .....                                | 106 |
| Table C.7 – EnumeratedValue Components .....                          | 107 |
| Table C.8 – ImportDirective Components .....                          | 107 |
| Table C.9 – Standard Type Descriptions .....                          | 108 |
| Table D.1 – Notation of Nodes depending on the NodeClass .....        | 115 |
| Table D.2 – Simple Notation of Nodes depending on the NodeClass ..... | 116 |

IECNORM.COM: Click to view the full PDF of IEC 62541-3:2015

Withd2Wm

# INTERNATIONAL ELECTROTECHNICAL COMMISSION

## OPC UNIFIED ARCHITECTURE –

### Part 3: Address Space 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.

International Standard IEC 62541-3 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

This second edition cancels and replaces the first edition published in 2010. This edition constitutes a technical revision.

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

- a) Added rules for subtyping enumerations in 8.14 (issue number 0606);
- b) Added *Property EnumValues* in 5.8.3 to support integer representation of enumerations that are not zero-based or have gaps (issue number 0876);
- c) Added *Property ValueAsText* in 5.6.2 providing a localized text representation for enumeration values (issue number 0951);

- d) Added *EventType SystemStatusChangeEvent* in 9.31 that can be used to indicate connection to sub system is lost (issue number 1255);
- e) Added *Properties MaxArrayLength* and *MaxStringLength* in 5.6.2 to identify the maximum string length and array length for clients writing values (issue number 1547);
- f) Removed the concept of *ModelParent* from document as it is not that useful. The *NodeId* of the *ReferenceType* will be kept not breaking existing applications (issue number 1554).
- g) Added *EventType ProgressEventType* in 9.4 identifying the progress of an operation such as a service call (issue number 1557);
- h) Stated in 8.38 that it is allowed to use TAI in all places where UTC time is used to avoid problems with leap seconds (issue number 1563);
- i) Added *Property EngineeringUnits* in 5.6.2 as used in IEC 62541-8 (issue number 1749);
- j) Added *ModellingRules OptionalPlaceholder* and *MandatoryPlaceholder* in 6.4.4.5.5 and 6.4.4.5.6 (issue number 1804).

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

| CDV         | Report on voting |
|-------------|------------------|
| 65E/374/CDV | 65E/402/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 *OPC Unified Architecture*, 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 3: Address Space Model

#### 1 Scope

This part of IEC 62541 describes the OPC Unified Architecture (OPC UA) *AddressSpace* and its *Objects*. This part of IEC 62541 is the OPC UA meta model on which OPC UA information models are based.

#### 2 Normative references

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 62541-4, *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 62541-11, *OPC Unified Architecture – Part 11: Historical Access*

ISO/IEC 10918-1, *Information technology – Digital compression and coding of continuous-tone still images: Requirements and guidelines*

ISO/IEC 15948, *Information technology – Computer graphics and image processing – Portable Network Graphics (PNG): Functional specification*

ISO 639 (all parts), *Codes for the representation of names of languages*

ISO 3166 (all parts), *Codes for the representation of names of countries and their subdivisions*

IEEE 754-1985, *IEEE Standard for Binary Floating-Point Arithmetic*, <http://ieeexplore.ieee.org/servlet/opac?punumber=2355>

IETF RFC 3066, *Tags for the Identification of Languages*, <http://tools.ietf.org/html/rfc3066>

XML Schema Part 1: <http://www.w3.org/TR/xmlschema-1/>

XML Schema Part 2: <http://www.w3.org/TR/xmlschema-2/>

XPATH: <http://www.w3.org/TR/xpath/>

### 3 Terms, definitions, abbreviations and conventions

#### 3.1 Terms and definitions

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

##### 3.1.1

##### **DataType**

instance of a *DataType Node* that is used together with the *ValueRank Attribute* to define the data type of a *Variable*

##### 3.1.2

##### **DataTypeId**

*NodeId* of a *DataType Node*

##### 3.1.3

##### **DataVariable**

*Variables* that represent *values* of *Objects*, either directly or indirectly for complex *Variables*, where the *Variables* are always the *TargetNode* of a *HasComponent Reference*

##### 3.1.4

##### **EventType**

*ObjectType Node* that represents the type definition of an *Event*

##### 3.1.5

##### **hierarchical Reference**

*Reference* that is used to construct hierarchies in the *AddressSpace*

Note 1 to entry: All hierarchical *ReferenceTypes* are derived from *HierarchicalReferences*.

##### 3.1.6

##### **InstanceDeclaration**

*Node* that is used by a complex *TypeDefinitionNode* to expose its complex structure

Note 1 to entry: It is an instance used by a type definition.

##### 3.1.7

##### **ModellingRule**

metadata of an *InstanceDeclaration* that defines how the *InstanceDeclaration* will be used for instantiation and also defines subtyping rules for an *InstanceDeclaration*

##### 3.1.8

##### **Property**

*Variables* that are the *TargetNode* for a *HasProperty Reference*

Note 1 to entry: *Properties* describe the characteristics of a *Node*.

##### 3.1.9

##### **SourceNode**

*Node* having a *Reference* to another *Node*

EXAMPLE: In the *Reference* "A contains B", "A" is the *SourceNode*.

##### 3.1.10

##### **TargetNode**

*Node* that is referenced by another *Node*

EXAMPLE: In the *Reference* "A Contains B", "B" is the *TargetNode*.

### 3.1.11

#### TypeDefinitionNode

Node that is used to define the type of another Node

Note 1 to entry: *ObjectType* and *VariableType* Nodes are *TypeDefinitionNodes*.

### 3.1.12

#### VariableType

Node that represents the type definition for a *Variable*

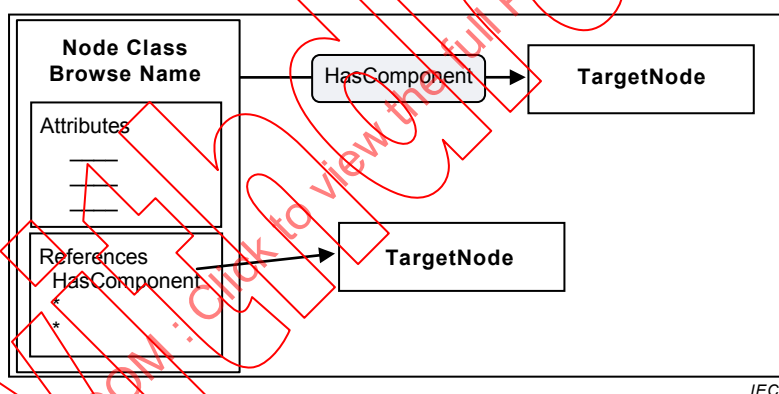
## 3.2 Abbreviations

|     |                             |
|-----|-----------------------------|
| UA  | Unified Architecture        |
| UML | Unified Modeling Language   |
| URI | Uniform Resource Identifier |
| W3C | World Wide Web Consortium   |
| XML | Extensible Markup Language  |

## 3.3 Conventions

### 3.3.1 Conventions for AddressSpace figures

Nodes and their *References* to each other are illustrated using figures. Figure 1 illustrates the conventions used in these figures.



**Figure 1 – AddressSpace Node diagrams**

In these figures, rectangles represent *Nodes*. *Node* rectangles may be titled with one or two lines of text. When two lines are used, the first text line in the rectangle identifies the *NodeClass* and the second line contains the *BrowseName*. When one line is used, it contains the *BrowseName*.

*Node* rectangles may contain boxes used to define their *Attributes* and *References*. Specific names in these boxes identify specific *Attributes* and *References*.

Shaded rectangles with rounded corners and with arrows passing through them represent *References*. The arrow that passes through them begins at the *SourceNode* and points to the *TargetNode*. *References* may also be shown by drawing an arrow that starts at the *Reference* name in the “References” box and ends at the *TargetNode*.

### 3.3.2 Conventions for defining NodeClasses

Clause 5 defines *AddressSpace NodeClasses*. Table 1 describes the format of the tables used to define *NodeClasses*.

**Table 1 – NodeClass Table Conventions**

| Name                       | Use                   | Data Type                         | Description                                                       |
|----------------------------|-----------------------|-----------------------------------|-------------------------------------------------------------------|
| <b>Attributes</b>          |                       |                                   |                                                                   |
| "Attribute name"           | "M" or "O"            | Data type of the <i>Attribute</i> | Defines the <i>Attribute</i>                                      |
| <b>References</b>          |                       |                                   |                                                                   |
| "Reference name"           | "1", "0..1" or "0..*" | Not used                          | Describes the use of the <i>Reference</i> by the <i>NodeClass</i> |
| <b>Standard Properties</b> |                       |                                   |                                                                   |
| "Property name"            | "M" or "O"            | Data type of the <i>Property</i>  | Defines the <i>Property</i>                                       |

The Name column contains the name of the *Attribute*, the name of the *ReferenceType* used to create a *Reference* or the name of a *Property* referenced using the *HasPropertyReference*.

The Use column defines whether the *Attribute* or *Property* is mandatory (M) or optional (O). When mandatory the *Attribute* or *Property* shall exist for every *Node* of the *NodeClass*. For *References* it specifies the cardinality. The following values may apply:

- "0..\*" identifies that there are no restrictions, that is, the *Reference* does not have to be provided but there is no limitation how often it can be provided;
- "0..1" identifies that the *Reference* is provided at most once;
- "1" identifies that the *Reference* shall be provided exactly once.

The Data Type column contains the name of the *DataType* of the *Attribute* or *Property*. It is not used for *References*.

The Description column contains the description of the *Attribute*, the *Reference* or the *Property*.

Only this standard may define *Attributes*. Thus, all *Attributes* of the *NodeClass* are specified in the table and can only be extended by other parts of this series of standards.

This standard also defines *ReferenceTypes*, but *ReferenceTypes* can also be specified by a *Server* or by a client using the *NodeManagement Services* specified in IEC 62541-4. Thus, the *NodeClass* tables contained in this standard can contain the base *ReferenceType* called *References* identifying that any *ReferenceType* may be used for the *NodeClass*, including system specific *ReferenceTypes*. The *NodeClass* tables only specify how the *NodeClasses* can be used as *SourceNodes* of *References*, not as *TargetNodes*. If a *NodeClass* table allows a *ReferenceType* for its *NodeClass* to be used as *SourceNode*, this is also true for subtypes of the *ReferenceType*. However, subtypes of the *ReferenceType* may restrict its *SourceNodes*.

This standard defines *Properties*, but *Properties* can be defined by other standard organizations or vendors and *Nodes* can have *Properties* that are not standardised. *Properties* defined in this standard are defined by their name, which is mapped to the *BrowseName* having the *NamespaceIndex* 0, which represents the *Namespace* for OPC UA.

The Use column (optional or mandatory) does not imply a specific *ModellingRule* for *Properties*. Different *Server* implementations will choose to use *ModellingRules* appropriate for them.

## 4 AddressSpace concepts

### 4.1 Overview

The remainder of 4 defines the concepts of the *AddressSpace*. Clause 5 defines the *NodeClasses* of the *AddressSpace* representing the *AddressSpace* concepts. Clause 6 defines details on the type model for *ObjectTypes* and *VariableTypes*. Standard *ReferenceTypes*, *DataTypes* and *EventTypes* are defined in Clauses 7 to 9.

The informative Annex A describes general considerations on how to use the Address Space Model and the informative Annex B provides a UML Model of the Address Space Model. The normative Annex C defines the OPC Binary Types Description System as a format to specify data type structures and the normative Annex D defines a graphical notation for OPC UA data.

### 4.2 Object Model

The primary objective of the OPC UA *AddressSpace* is to provide a standard way for *Servers* to represent *Objects* to *Clients*. The OPC UA Object Model has been designed to meet this objective. It defines *Objects* in terms of *Variables* and *Methods*. It also allows relationships to other *Objects* to be expressed. Figure 2 illustrates the model.

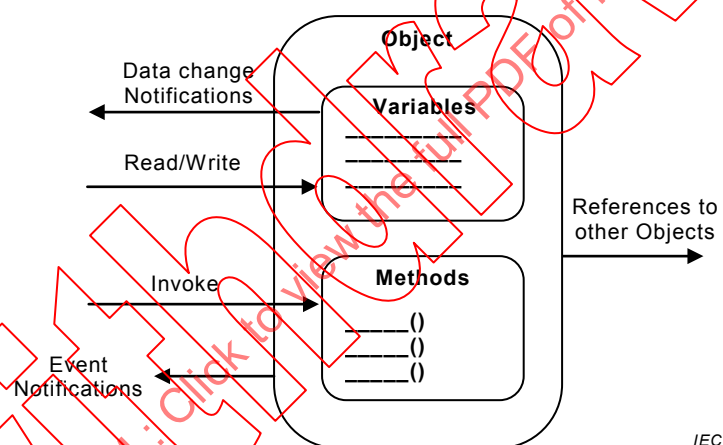


Figure 2 – OPC UA Object Model

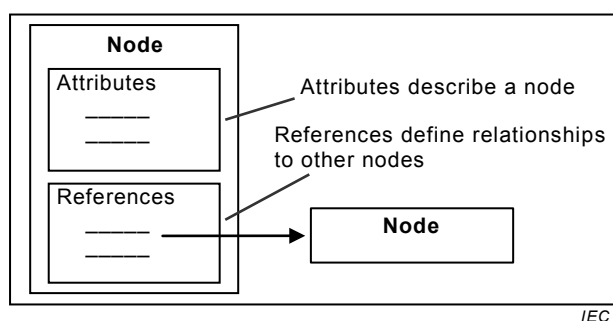
The elements of this model are represented in the *AddressSpace* as *Nodes*. Each *Node* is assigned to a *NodeClass* and each *NodeClass* represents a different element of the Object Model. Clause 5 defines the *NodeClasses* used to represent this model.

### 4.3 Node Model

#### 4.3.1 General

The set of *Objects* and related information that the OPC UA *Server* makes available to *Clients* is referred to as its *AddressSpace*. The model for *Objects* is defined by the OPC UA Object Model (see 4.2).

Objects and their components are represented in the *AddressSpace* as a set of *Nodes* described by *Attributes* and interconnected by *References*. Figure 3 illustrates the model of a *Node* and the remainder of 4.3 discusses the details of the Node Model.



**Figure 3 – AddressSpace Node Model**

### 4.3.2 NodeClasses

*NodeClasses* are defined in terms of the *Attributes* and *References* that shall be instantiated (given values) when a *Node* is defined in the *AddressSpace*. *Attributes* are discussed in 4.3.3 and *References* in 4.3.4.

Clause 5 defines the *NodeClasses* for the OPC UA *AddressSpace*. These *NodeClasses* are referred to collectively as the metadata for the *AddressSpace*. Each *Node* in the *AddressSpace* is an instance of one of these *NodeClasses*. No other *NodeClasses* shall be used to define *Nodes*, and as a result, *Clients* and *Servers* are not allowed to define *NodeClasses* or extend the definitions of these *NodeClasses*.

### 4.3.3 Attributes

*Attributes* are data elements that describe *Nodes*. *Clients* can access *Attribute* values using Read, Write, Query, and Subscription/MonitoredItem *Services*. These *Services* are defined in IEC 62541-4.

*Attributes* are elementary components of *NodeClasses*. *Attribute* definitions are included as part of the *NodeClass* definitions in Clause 5 and, therefore, are not included in the *AddressSpace*.

Each *Attribute* definition consists of an attribute id (for attribute ids of *Attributes*, see IEC 62541-6), a name, a description, a data type and a mandatory/optional indicator. The set of *Attributes* defined for each *NodeClass* shall not be extended by *Clients* or *Servers*.

When a *Node* is instantiated in the *AddressSpace*, the values of the *NodeClass Attributes* are provided. The mandatory/optional indicator for the *Attribute* indicates whether the *Attribute* has to be instantiated.

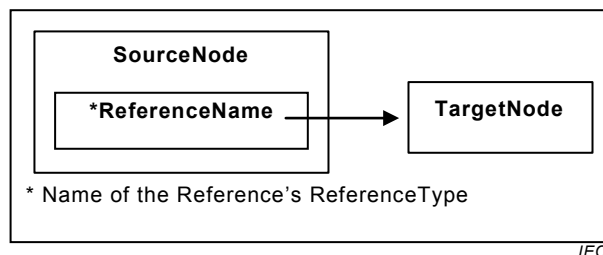
### 4.3.4 References

*References* are used to relate *Nodes* to each other. They can be accessed using the browsing and querying *Services* defined in IEC 62541-4.

Like *Attributes*, they are defined as fundamental components of *Nodes*. Unlike *Attributes*, *References* are defined as instances of *ReferenceType Nodes*. *ReferenceType Nodes* are visible in the *AddressSpace* and are defined using the *ReferenceType NodeClass* (see 5.3).

The *Node* that contains the *Reference* is referred to as the *SourceNode* and the *Node* that is referenced is referred to as the *TargetNode*. The combination of the *SourceNode*, the *ReferenceType* and the *TargetNode* are used in OPC UA *Services* to uniquely identify *References*. Thus, each *Node* can reference another *Node* with the same *ReferenceType* only once. Any subtypes of concrete *ReferenceTypes* are considered to be equal to the base

concrete *ReferenceTypes* when identifying *References* (see 5.3 for subtypes of *ReferenceTypes*). Figure 4 illustrates this model of a *Reference*.



**Figure 4 – Reference Model**

The *TargetNode* of a *Reference* may be in the same *AddressSpace* or in the *AddressSpace* of another OPC UA *Server*. *TargetNodes* located in other *Servers* are identified in OPC UA *Services* using a combination of the remote *Server* name and the identifier assigned to the *Node* by the remote *Server*.

OPC UA does not require that the *TargetNode* exists, thus *References* may point to a *Node* that does not exist.

## 4.4 Variables

### 4.4.1 General

*Variables* are used to represent *values*. Two types of *Variables* are defined, *Properties* and *DataVariables*. They differ in the kind of data that they represent and whether they can contain other *Variables*.

### 4.4.2 Properties

*Properties* are *Server*-defined characteristics of *Objects*, *DataVariables* and other *Nodes*. *Properties* differ from *Attributes* in that they characterise *what* the *Node* represents, such as a device or a purchase order. *Attributes* define additional metadata that is instantiated for all *Nodes* from a *NodeClass*. *Attributes* are common to all *Nodes* of a *NodeClass* and only defined by this specification whereas *Properties* can be *Server*-defined.

For example, an *Attribute* defines the *DataType* of *Variables* whereas a *Property* can be used to specify the engineering unit of some *Variables*.

To prevent recursion, *Properties* are not allowed to have *Properties* defined for them. To easily identify *Properties*, the *BrowseName* of a *Property* shall be unique in the context of the *Node* containing the *Properties* (see 5.6.3 for details).

A *Node* and its *Properties* shall always reside in the same *Server*.

### 4.4.3 DataVariables

*DataVariables* represent the content of an *Object*. For example, a file *Object* may be defined that contains a stream of bytes. The stream of bytes may be defined as a *DataVariable* that is an array of bytes. *Properties* may be used to expose the creation time and owner of the file *Object*.

For example, if a *DataVariable* is defined by a data structure that contains two fields, "startTime" and "endTime" then it might have a *Property* specific to that data structure, such as "earliestStartTime".

As another example, function blocks in control systems might be represented as *Objects*. The parameters of the function block, such as its setpoints, may be represented as *DataVariables*. The function block *Object* might also have *Properties* that describe its execution time and its type.

*DataVariables* may have additional *DataVariables*, but only if they are complex. In this case, their *DataVariables* shall always be elements of their complex definitions. Following the example introduced by the description of *Properties* in 4.4.2, the *Server* could expose “startTime” and “endTime” as separate components of the data structure.

As another example, a complex *DataVariable* may define an aggregate of temperature values generated by three separate temperature transmitters that are also visible in the *AddressSpace*. In this case, this complex *DataVariable* could define *HasComponentReferences* from it to the individual temperature values that it is composed of.

## 4.5 TypeDefinitionNodes

### 4.5.1 General

OPC UA *Servers* shall provide type definitions for *Objects* and *Variables*. The *HasTypeDefinition Reference* shall be used to link an instance with its type definition represented by a *TypeDefinitionNode*. Type definitions are required; however, IEC 62541-5 defines a *BaseObjectType*, a *PropertyType*, and a *BaseDataVariableType* so a *Server* can use such a base type if no more specialised type information is available. *Objects* and *Variables* inherit the *Attributes* specified by their *TypeDefinitionNode* (see 6.4 for details).

In some cases, the *NodeId* used by the *HasTypeDefinition Reference* will be well-known to *Clients* and *Servers*. Organizations may define *TypeDefinitionNodes* that are well-known in the industry. Well-known *NodeIds* of *TypeDefinitionNodes* provide for commonality across OPC UA *Servers* and allow *Clients* to interpret the *TypeDefinitionNode* without having to read it from the *Server*. Therefore, *Servers* may use well-known *NodeIds* without representing the corresponding *TypeDefinitionNodes* in their *AddressSpace*. However, the *TypeDefinitionNodes* shall be provided for generic *Clients*. These *TypeDefinitionNodes* may exist in another *Server*.

The following example, illustrated in Figure 5, describes the use of the *HasTypeDefinition Reference*. In this example, a setpoint parameter “SP” is represented as a *DataVariable* in the *AddressSpace*. This *DataVariable* is part of an *Object* not shown in the figure.

To provide for a common setpoint definition that can be used by other *Objects*, a specialised *VariableType* is used. Each setpoint *DataVariable* that uses this common definition will have a *HasTypeDefinition Reference* that identifies the common “SetPoint” *VariableType*.

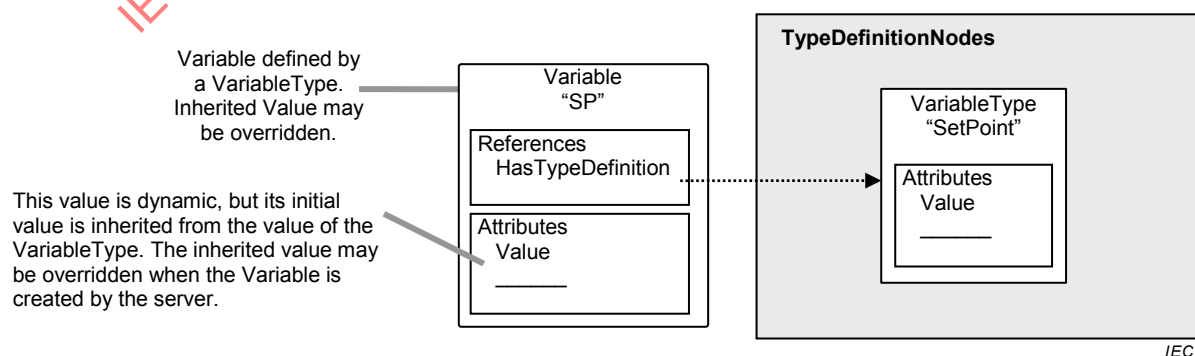
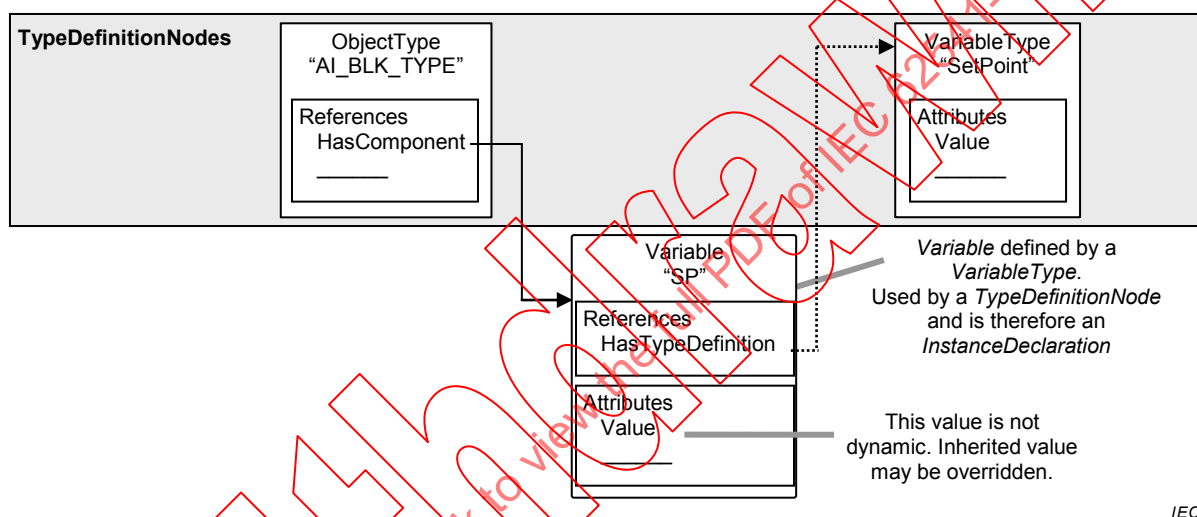


Figure 5 – Example of a Variable defined by a VariableType

#### 4.5.2 Complex TypeDefinitionNodes and their InstanceDeclarations

*TypeDefinitionNodes* can be complex. A complex *TypeDefinitionNode* also defines *References* to other *Nodes* as part of the type definition. The *ModellingRules* defined in 6.4.4 specify how those *Nodes* are handled when creating an instance of the type definition.

A *TypeDefinitionNode* references instances instead of other *TypeDefinitionNodes* to allow unique names for several instances of the same type, to define default values and to add *References* for those instances that are specific to this complex *TypeDefinitionNode* and not to the *TypeDefinitionNode* of the instance. For example, in Figure 6 the *ObjectType* “AI\_BLK\_TYPE”, representing a function block, has a *HasComponent Reference* to a *Variable* “SP” of the *VariableType* “SetPoint”. “AI\_BLK\_TYPE” could have an additional setpoint *Variable* of the same type using a different name. It could add a *Property* to the *Variable* that was not defined by its *TypeDefinitionNode* “SetPoint”. And it could define a default value for “SP”, that is, each instance of “AI\_BLK\_TYPE” would have a *Variable* “SP” initially set to this value.



IEC

Figure 6 – Example of a Complex TypeDefinition

This approach is commonly used in object-oriented programming languages in which the variables of a class are defined as instances of other classes. When the class is instantiated, each variable is also instantiated, but with the default values (constructor values) defined for the containing class. That is, typically, the constructor for the component class runs first, followed by the constructor for the containing class. The constructor for the containing class may override component values set by the component class.

To distinguish instances used for the type definitions from instances that represent real data, those instances are called *InstanceDeclarations*. However, this term is used to simplify this specification, if an instance is an *InstanceDeclaration* or not is only visible in the *AddressSpace* by following its *References*. Some instances may be shared and therefore referenced by *TypeDefinitionNodes*, *InstanceDeclarations* and instances. This is similar to class variables in object-oriented programming languages.

#### 4.5.3 Subtyping

This standard allows subtyping of type definitions. The subtyping rules are defined in Clause 6. Subtyping of *ObjectTypes* and *VariableTypes* allows:

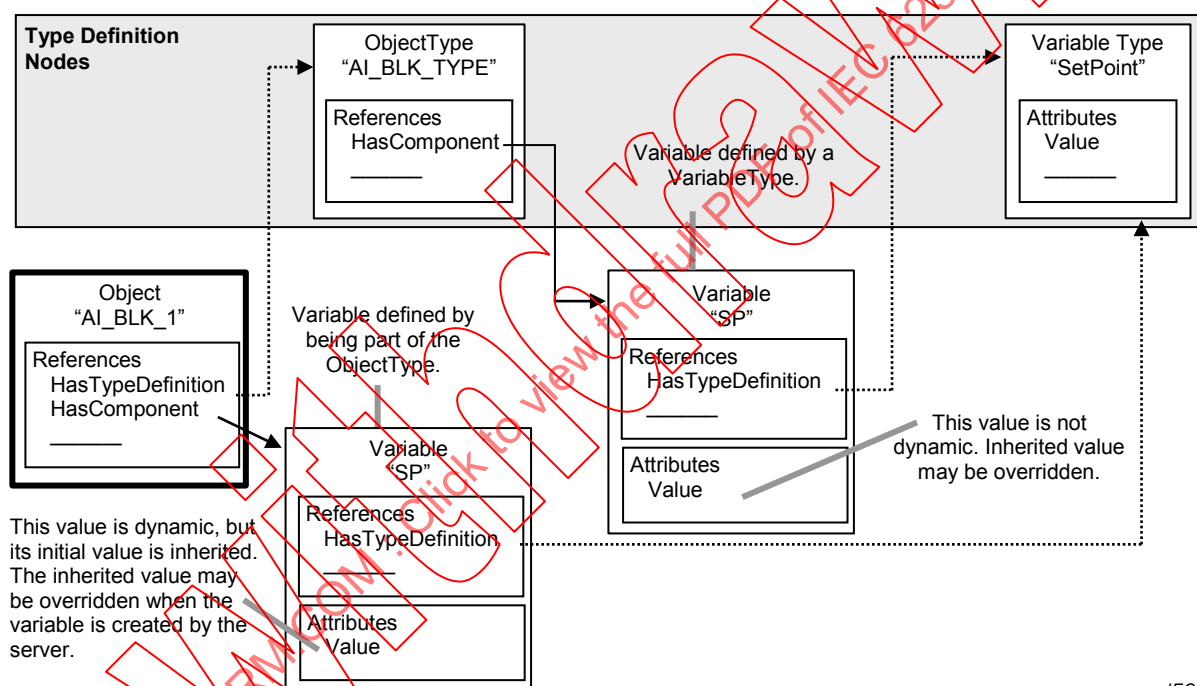
- *Clients* that only know the supertype to handle an instance of the subtype as if it were an instance of the supertype;
- instances of the supertype to be replaced by instances of the subtype;

- specialised types that inherit common characteristics of the base type.

In other words, subtypes reflect the structure defined by their supertype but may add additional characteristics. For example, a vendor may wish to extend a general “TemperatureSensor” *VariableType* by adding a *Property* providing the next maintenance interval. The vendor would do this by creating a new *VariableType* which is a *TargetNode* for a *HasSubtype* reference from the original *VariableType* and adding the new *Property* to it.

#### 4.5.4 Instantiation of complex TypeDefinitionNodes

The instantiation of complex *TypeDefinitionNodes* depends on the *ModellingRules* defined in 6.4.4. However, the intention is that instances of a type definition will reflect the structure defined by the *TypeDefinitionNode*. Figure 7 shows an instance of the *TypeDefinitionNode* “AI\_BLK\_TYPE”, where the *ModellingRule Mandatory*, defined in 6.4.4.5.2, was applied for its containing *Variable*. Thus, an instance of “AI\_BLK\_TYPE”, called “AI\_BLK\_1”, has a *HasTypeDefinition* Reference to “AI\_BLK\_TYPE”. It also contains a *Variable* “SP” having the same *BrowseName* as the *Variable* “SP” used by the *TypeDefinitionNode* and thereby reflects the structure defined by the *TypeDefinitionNode*.



IEC

Figure 7 – Object and its Components defined by an ObjectType

A client knowing the *ObjectType* “AI\_BLK\_TYPE” can use this knowledge to directly browse to the containing *Nodes* for each instance of this type. This allows programming against the *TypeDefinitionNode*. For example, a graphical element may be programmed in the client that handles all instances of “AI\_BLK\_TYPE” in the same way by showing the value of “SP”.

There are several constraints related to programming against the *TypeDefinitionNode*. A *TypeDefinitionNode* or an *InstanceDeclaration* shall never reference two *Nodes* having the same *BrowseName* using *hierarchical References* in the forward direction. Instances based on *InstanceDeclarations* shall always keep the same *BrowseName* as the *InstanceDeclaration* they are derived from. A special *Service* defined in IEC 62541-4 called *TranslateBrowsePathsToNodeIds* may be used to identify the instances based on the *InstanceDeclarations*. Using the simple *Browse Service* might not be sufficient since the uniqueness of the *BrowseName* is only required for *TypeDefinitionNodes* and *InstanceDeclarations*, not for other instances. Thus, “AI\_BLK\_1” may have another *Variable* with the *BrowseName* “SP”, although this one would not be derived from an *InstanceDeclaration* of the *TypeDefinitionNode*.

Instances derived from an *InstanceDeclaration* shall be of the same *TypeDefinitionNode* or a subtype of this *TypeDefinitionNode*.

A *TypeDefinitionNode* and its *InstanceDeclarations* shall always reside in the same *Server*. However, instances may point with their *HasTypeDefinition Reference* to a *TypeDefinitionNode* in a different *Server*.

## 4.6 Event Model

### 4.6.1 General

The Event Model defines a general purpose eventing system that can be used in many diverse vertical markets.

*Events* represent specific transient occurrences. System configuration changes and system errors are examples of *Events*. *Event Notifications* report the occurrence of an *Event*. *Events* defined in this document are not directly visible in the OPC UA *AddressSpace*. *Objects* and *Views* can be used to subscribe to *Events*. The *EventNotifier Attribute* of those *Nodes* identifies if the *Node* allows subscribing to *Events*. *Clients* subscribe to such *Nodes* to receive *Notifications* of *Event* occurrences.

*Event Subscriptions* use the Monitoring and Subscription *Services* defined in IEC 62541-4 to subscribe to the *Event Notifications* of a *Node*.

Any OPC UA *Server* that supports eventing shall expose at least one *Node* as *EventNotifier*. The *Server Object* defined in IEC 62541-5 is used for this purpose. *Events* generated by the *Server* are available via this *Server Object*. A *Server* is not expected to produce *Events* if the connection to the event source is down for some reason (i.e. the system is offline).

*Events* may also be exposed through other *Nodes* anywhere in the *AddressSpace*. These *Nodes* (identified via the *EventNotifier Attribute*) provide some subset of the *Events* generated by the *Server*. The position in the *AddressSpace* dictates what this subset will be. For example, a process area *Object* representing a functional area of the process would provide *Events* originating from that area of the process only. It should be noted that this is only an example and it is fully up to the *Server* to determine what *Events* should be provided by which *Node*.

### 4.6.2 EventTypes

Each *Event* is of a specific *EventType*. A *Server* may support many types. This part of IEC 62541 defines the *BaseEventType* that all other *EventTypes* derive from. It is expected that other companion specifications will define additional *EventTypes* deriving from the base types defined in this part of IEC 62541.

The *EventTypes* supported by a *Server* are exposed in the *AddressSpace* of a *Server*. *EventTypes* are represented as *ObjectTypes* in the *AddressSpace* and do not have a special *NodeClass* associated to them. IEC 62541-5 defines how a *Server* exposes the *EventTypes* in detail.

*EventTypes* defined in this document are specified as abstract and therefore never instantiated in the *AddressSpace*. Event occurrences of those *EventTypes* are only exposed via a *Subscription*. *EventTypes* exist in the *AddressSpace* to allow *Clients* to discover the *EventType*. This information is used by a client when establishing and working with *Event Subscriptions*. *EventTypes* defined by other parts of the IEC 62541 series or companion specifications as well as *Server* specific *EventTypes* may be defined as not abstract and therefore instances of those *EventTypes* may be visible in the *AddressSpace* although *Events* of those *EventTypes* are also accessible via the *Event Notification* mechanisms.

Standard *EventTypes* are described in Clause 9. Their representation in the *AddressSpace* is specified in IEC 62541-5.

#### 4.6.3 Event Categorization

*Events* can be categorised by creating new *EventTypes* which are subtypes of existing *EventTypes* but do not extend an existing type. They are used only to identify an event as being of the new *EventType*. For example, the *EventType* *DeviceFailureEventType* could be subtyped into *TransmitterFailureEventType* and *ComputerFailureEventType*. These new subtypes would not add new *Properties* or change the semantic inherited from the *DeviceFailureEventType* other than purely for categorization of the *Events*.

*Event* sources can also be organised into groups by using the *Event ReferenceTypes* described in 7.17 and 7.19. For example, a *Server* may define *Objects* in the *AddressSpace* representing *Events* related to physical devices, or *Event* areas of a plant or functionality contained in the *Server*. *Event References* would be used to indicate which *Event* sources represent physical devices and which ones represent some *Server*-based functionality. In addition, *References* can be used to group the physical devices or *Server*-based functionality into hierarchical *Event* areas. In some cases, an *Event* source may be categorised as being both a device and a *Server* function. In this case, two relationships would be established. Refer to the description of the *Event ReferenceTypes* for additional examples.

*Clients* can select a category or categories of *Events* by defining content filters that include terms specifying the *EventType* of the *Event* or a grouping of *Event* sources. The two mechanisms allow for a single *Event* to be categorised in multiple manners. A client could obtain all *Events* related to a physical device or all failures of a particular device.

#### 4.7 Methods

*Methods* are “lightweight” functions, whose scope is bounded by an owning (see Note) *Object*, similar to the methods of a class in object-oriented programming or an owning *ObjectType*, similar to static methods of a class. *Methods* are invoked by a client, proceed to completion on the *Server* and return the result to the client. The lifetime of the *Method*’s invocation instance begins when the client calls the *Method* and ends when the result is returned.

NOTE The owning *Object* or *ObjectType* is specified in the service call when invoking the *Method*.

While *Methods* may affect the state of the owning *Object*, they have no explicit state of their own. In this sense, they are stateless. *Methods* can have a varying number of input arguments and return resultant arguments. Each *Method* is described by a *Node* of the *Method NodeClass*. This *Node* contains the metadata that identifies the *Method*’s arguments and describes its behaviour.

*Methods* are invoked by using the *Call Service* defined in IEC 62541-4. Each *Method* is invoked within the context of an existing session. If the session is terminated during *Method* execution, the results of the *Method*’s execution cannot be returned to the client and are discarded. In that case, the *Method* execution is undefined, that is, the *Method* may be executed until it is finished or it may be aborted.

*Clients* discover the *Methods* supported by a *Server* by browsing for the owning *Objects References* that identify their supported *Methods*.

### 5 Standard NodeClasses

#### 5.1 Overview

Clause 5 defines the *NodeClasses* used to define *Nodes* in the OPC UA *AddressSpace*. *NodeClasses* are derived from a common *Base NodeClass*. This *NodeClass* is defined first,

followed by those used to organise the *AddressSpace* and then by the *NodeClasses* used to represent *Objects*.

The *NodeClasses* defined to represent *Objects* fall into three categories: those used to define instances, those used to define types for those instances and those used to define data types. Subclause 6.3 describes the rules for subtyping and 6.4 the rules for instantiation of the type definitions.

## 5.2 Base NodeClass

### 5.2.1 General

The OPC UA Address Space Model defines a *Base NodeClass* from which all other *NodeClasses* are derived. The derived *NodeClasses* represent the various components of the OPC UA Object Model (see 4.2). The *Attributes* of the *Base NodeClass* are specified in Table 2. There are no *References* specified for the *Base NodeClass*.

**Table 2 – Base NodeClass**

| Name                                       | Use | Data Type     | Description |
|--------------------------------------------|-----|---------------|-------------|
| <b>Attributes</b>                          |     |               |             |
| NodeId                                     | M   | NodeId        | See 5.2.2   |
| NodeClass                                  | M   | NodeClass     | See 5.2.3   |
| BrowseName                                 | M   | QualifiedName | See 5.2.4   |
| DisplayName                                | M   | LocalizedText | See 5.2.5   |
| Description                                | O   | LocalizedText | See 5.2.6   |
| WriteMask                                  | O   | UInt32        | See 5.2.7   |
| UserWriteMask                              | O   | UInt32        | See 5.2.8   |
| <b>References</b>                          |     |               |             |
| No References specified for this NodeClass |     |               |             |

### 5.2.2 NodeId

*Nodes* are unambiguously identified using a constructed identifier called the *NodeId*. Some *Servers* may accept alternative *NodeIds* in addition to the canonical *NodeId* represented in this *Attribute*. A *Server* shall persist the *NodeId* of a *Node*, that is, it shall not generate new *NodeIds* when rebooting. The structure of the *NodeId* is defined in 8.2.

### 5.2.3 NodeClass

The *NodeClass Attribute* identifies the *NodeClass* of a *Node*. Its data type is defined in 8.30.

### 5.2.4 BrowseName

*Nodes* have a *BrowseName Attribute* that is used as a non-localised human-readable name when browsing the *AddressSpace* to create paths out of *BrowseNames*. The *TranslateBrowsePathsToNodeIds Service* defined in IEC 62541-4 can be used to follow a path constructed of *BrowseNames*.

A *BrowseName* should never be used to display the name of a *Node*. The *DisplayName* should be used instead for this purpose.

Unlike *NodeIds*, the *BrowseName* cannot be used to unambiguously identify a *Node*. Different *Nodes* may have the same *BrowseName*.

Subclause 8.3 defines the structure of the *BrowseName*. It contains a namespace and a string. The namespace is provided to make the *BrowseName* unique in some cases in the context of a *Node* (e.g. *Properties* of a *Node*) although not unique in the context of the *Server*. If different organizations define *BrowseNames* for *Properties*, the namespace of the *BrowseName* provided by the organization makes the *BrowseName* unique, although different organizations may use the same string having a slightly different meaning.

*Servers* may often choose to use the same namespace for the *NodeId* and the *BrowseName*. However, if they want to provide a standard *Property*, its *BrowseName* shall have the namespace of the standards body although the namespace of the *NodeId* reflects something else, for example the local *Server*.

It is recommended that standard bodies defining standard type definitions use their namespace for the *NodeId* of the *TypeDefinitionNode* as well as for the *BrowseName* of the *TypeDefinitionNode*.

The string-part of the *BrowseName* is case sensitive.

### 5.2.5 DisplayName

The *DisplayName Attribute* contains the localised name of the *Node*. *Clients* should use this *Attribute* if they want to display the name of the *Node* to the user. They should not use the *BrowseName* for this purpose. The *Server* may maintain one or more localised representations for each *DisplayName*. *Clients* negotiate the locale to be returned when they open a session with the *Server*. Refer to IEC 62541-4 for a description of session establishment and locales. Subclause 8.5 defines the structure of the *DisplayName*. The string part of the *DisplayName* is restricted to 512 characters.

### 5.2.6 Description

The optional *Description Attribute* shall explain the meaning of the *Node* in a localised text using the same mechanisms for localisation as described for the *DisplayName* in 5.2.5.

### 5.2.7 WriteMask

The optional *WriteMask Attribute* exposes the possibilities of a client to write the *Attributes* of the *Node*. The *WriteMask Attribute* does not take any user access rights into account, that is, although an *Attribute* is writeable this may be restricted to a certain user/user group.

If the OPC UA *Server* does not have the ability to get the *WriteMask* information for a specific *Attribute* from the underlying system, it should state that it is writable. If a write operation is called on the *Attribute*, the *Server* should transfer this request and return the corresponding *StatusCode* if such a request is rejected. *StatusCodes* are defined in IEC 62541-4.

The *WriteMask Attribute* is a 32-bit unsigned integer with the structure defined in Table 3. If the bit is set to 0, it means the *Attribute* is not writeable, if it is set to 1, it means it is writable. If a *Node* does not support a specific *Attribute*, the corresponding bit has to be set to 0.

**Table 3 – Bit mask for WriteMask and UserWriteMask**

| Field                   | Bit   | Description                                                                                                                                                                                                                          |
|-------------------------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AccessLevel             | 0     | Indicates if the AccessLevel Attribute is writable.                                                                                                                                                                                  |
| ArrayDimensions         | 1     | Indicates if the ArrayDimensions Attribute is writable.                                                                                                                                                                              |
| BrowseName              | 2     | Indicates if the BrowseName Attribute is writable.                                                                                                                                                                                   |
| ContainsNoLoops         | 3     | Indicates if the ContainsNoLoops Attribute is writable.                                                                                                                                                                              |
| DataType                | 4     | Indicates if the DataType Attribute is writable.                                                                                                                                                                                     |
| Description             | 5     | Indicates if the Description Attribute is writable.                                                                                                                                                                                  |
| DisplayName             | 6     | Indicates if the DisplayName Attribute is writable.                                                                                                                                                                                  |
| EventNotifier           | 7     | Indicates if the EventNotifier Attribute is writable.                                                                                                                                                                                |
| Executable              | 8     | Indicates if the Executable Attribute is writable.                                                                                                                                                                                   |
| Historizing             | 9     | Indicates if the Historizing Attribute is writable.                                                                                                                                                                                  |
| InverseName             | 10    | Indicates if the InverseName Attribute is writable.                                                                                                                                                                                  |
| IsAbstract              | 11    | Indicates if the IsAbstract Attribute is writable.                                                                                                                                                                                   |
| MinimumSamplingInterval | 12    | Indicates if the MinimumSamplingInterval Attribute is writable.                                                                                                                                                                      |
| NodeClass               | 13    | Indicates if the NodeClass Attribute is writable.                                                                                                                                                                                    |
| NodeId                  | 14    | Indicates if the NodeId Attribute is writable.                                                                                                                                                                                       |
| Symmetric               | 15    | Indicates if the Symmetric Attribute is writable.                                                                                                                                                                                    |
| UserAccessLevel         | 16    | Indicates if the UserAccessLevel Attribute is writable.                                                                                                                                                                              |
| UserExecutable          | 17    | Indicates if the UserExecutable Attribute is writable.                                                                                                                                                                               |
| UserWriteMask           | 18    | Indicates if the UserWriteMask Attribute is writable.                                                                                                                                                                                |
| ValueRank               | 19    | Indicates if the ValueRank Attribute is writable.                                                                                                                                                                                    |
| WriteMask               | 20    | Indicates if the WriteMask Attribute is writable.                                                                                                                                                                                    |
| ValueForVariableType    | 21    | Indicates if the Value Attribute is writable for a VariableType. It does not apply for Variables since this is handled by the AccessLevel and UserAccessLevel Attributes for the Variable. For Variables this bit shall be set to 0. |
| Reserved                | 22:31 | Reserved for future use. Shall always be zero.                                                                                                                                                                                       |

### 5.2.8 UserWriteMask

The optional *UserWriteMask Attribute* exposes the possibilities of a client to write the *Attributes* of the *Node* taking user access rights into account. It uses the same bit mask as used in the *WriteMask Attribute*, defined in Table 3.

The *UserWriteMask Attribute* can only further restrict the *WriteMask Attribute*, when it is set to not writable in the general case that applies for every user.

## 5.3 ReferenceType NodeClass

### 5.3.1 General

*References* are defined as instances of *ReferenceType Nodes*. *ReferenceType Nodes* are visible in the *AddressSpace* and are defined using the *ReferenceType NodeClass* as specified in Table 4. In contrast, a *Reference* is an inherent part of a *Node* and no *NodeClass* is used to represent *References*.

This standard defines a set of *ReferenceTypes* provided as an inherent part of the OPC UA Address Space Model. These *ReferenceTypes* are defined in Clause 7 and their representation in the *AddressSpace* is defined in IEC 62541-5. *Servers* may also define *ReferenceTypes*. In addition, IEC 62541-4 defines *NodeManagement Services* that allow *Clients* to add *ReferenceTypes* to the *AddressSpace*.

**Table 4 – ReferenceType NodeClass**

| Name                       | Use  | Data Type     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|----------------------------|------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Attributes</b>          |      |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Base NodeClass Attributes  | M    | --            | Inherited from the <i>Base NodeClass</i> . See 5.2.                                                                                                                                                                                                                                                                                                                                                                                                            |
| IsAbstract                 | M    | Boolean       | A boolean <i>Attribute</i> with the following values:<br>TRUE it is an abstract <i>ReferenceType</i> , i.e. no <i>Reference</i> of this type shall exist, only of its subtypes.<br>FALSE it is not an abstract <i>ReferenceType</i> , i.e. <i>References</i> of this type can exist.                                                                                                                                                                           |
| Symmetric                  | M    | Boolean       | A boolean <i>Attribute</i> with the following values:<br>TRUE the meaning of the <i>ReferenceType</i> is the same as seen from both the <i>SourceNode</i> and the <i>TargetNode</i> .<br>FALSE the meaning of the <i>ReferenceType</i> as seen from the <i>TargetNode</i> is the inverse of that as seen from the <i>SourceNode</i> .                                                                                                                          |
| InverseName                | O    | LocalizedText | The inverse name of the <i>Reference</i> , that is the meaning of the <i>ReferenceType</i> as seen from the <i>TargetNode</i> .                                                                                                                                                                                                                                                                                                                                |
| <b>References</b>          |      |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| HasProperty                | 0..* |               | Used to identify the <i>Properties</i> (see 5.3.3.2).                                                                                                                                                                                                                                                                                                                                                                                                          |
| HasSubtype                 | 0..* |               | Used to identify subtypes (see 5.3.3.3).                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Standard Properties</b> |      |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| NodeVersion                | O    | String        | The <i>NodeVersion Property</i> is used to indicate the version of a <i>Node</i> .<br>The <i>NodeVersion Property</i> is updated each time a <i>Reference</i> is added or deleted to the <i>Node</i> the <i>Property</i> belongs to. <i>Attribute</i> value changes do not cause the <i>NodeVersion</i> to change.<br><i>Clients</i> may read the <i>NodeVersion Property</i> or subscribe to it to determine when the structure of a <i>Node</i> has changed. |

### 5.3.2 Attributes

The *ReferenceType NodeClass* inherits the base *Attributes* from the *Base NodeClass* defined in 5.2. The inherited *BrowseName Attribute* is used to specify the meaning of the *ReferenceType* as seen from the *SourceNode*. For example, the *ReferenceType* with the *BrowseName* “Contains” is used in *References* that specify that the *SourceNode* contains the *TargetNode*. The inherited *DisplayName Attribute* contains a translation of the *BrowseName*.

The *BrowseName* of a *ReferenceType* shall be unique in a *Server*. It is not allowed that two different *ReferenceTypes* have the same *BrowseName*.

The *IsAbstract Attribute* indicates if the *ReferenceType* is abstract. Abstract *ReferenceTypes* cannot be instantiated and are used only for organizational reasons, for example to specify some general semantics or constraints that its subtypes inherit.

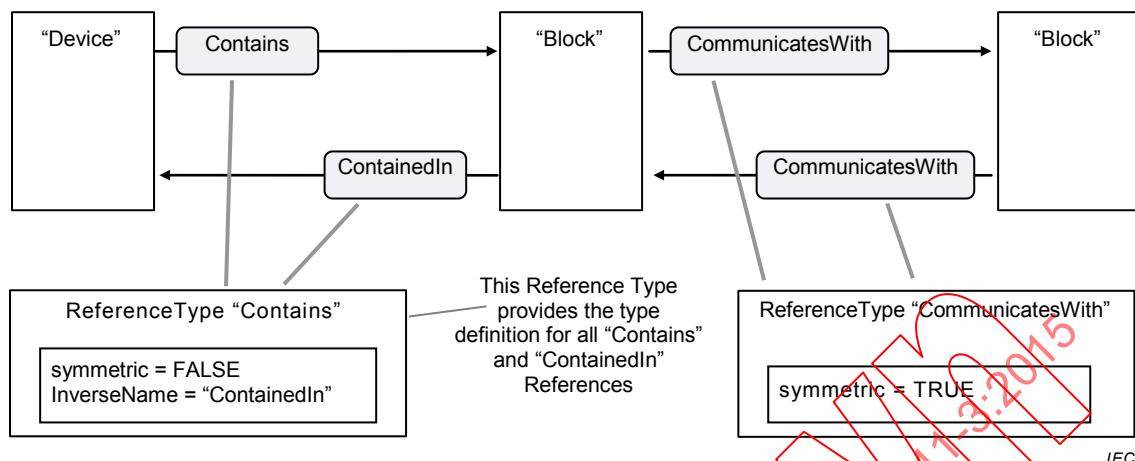
The *Symmetric Attribute* is used to indicate whether or not the meaning of the *ReferenceType* is the same for both the *SourceNode* and *TargetNode*.

If a *ReferenceType* is symmetric, the *InverseName Attribute* shall be omitted. Examples of symmetric *ReferenceTypes* are “Connects To” and “Communicates With”. Both imply the same semantic coming from the *SourceNode* or the *TargetNode*. Therefore both directions are considered to be forward *References*.

If the *ReferenceType* is non-symmetric and not abstract, the *InverseName Attribute* shall be set. The *InverseName Attribute* specifies the meaning of the *ReferenceType* as seen from the *TargetNode*. Examples of non-symmetric *ReferenceTypes* include “Contains” and “Contained In”, and “Receives From” and “Sends To”.

*References* that use the *InverseName*, such as “Contained In” *References*, are referred to as inverse *References*.

Figure 8 provides examples of symmetric and non-symmetric *References* and the use of the *BrowseName* and the *InverseName*.



**Figure 8 – Symmetric and Non-Symmetric References**

It might not always be possible for *Servers* to instantiate both forward and inverse *References* for non-symmetric *ReferenceTypes* as shown in Figure 8. When they do, the *References* are referred to as *bidirectional*. Although not required, it is recommended that all *hierarchical References* be instantiated as bidirectional to ensure browse connectivity. A bidirectional *Reference* is modelled as two separate *References*.

As an example of a *unidirectional Reference*, it is often the case that a subscriber knows its publisher, but its publisher does not know its subscribers. The subscriber would have a "Subscribes To" *Reference* to the publisher, without the publisher having the corresponding "Publishes To" inverse *References* to its subscribers.

The *DisplayName* and the *InverseName* are the only standardised places to indicate the semantic of a *ReferenceType*. There may be more complex semantics associated with a *ReferenceType* than can be expressed in those *Attributes* (e.g. the semantic of *HasSubtype*). This standard does not specify how this semantic should be exposed. However, the *Description Attribute* can be used for this purpose. This standard provides a semantic for the *ReferenceTypes* specified in Clause 7.

A *ReferenceType* can have constraints restricting its use. For example, it can specify that starting from *Node A* and only following *References* of this *ReferenceType* or one of its subtypes, it shall never be able to return to *A*, that is, a "No Loop" constraint.

This standard does not specify how those constraints could or should be made available in the *AddressSpace*. Nevertheless, for the standard *ReferenceTypes*, some constraints are specified in Clause 7. This standard does not restrict the kind of constraints valid for a *ReferenceType*. It can, for example, also affect an *ObjectType*. The restriction that a *ReferenceType* can only be used by relating *Nodes* of some *NodeClasses* with a defined cardinality is a special constraint of a *ReferenceType*.

### 5.3.3 References

#### 5.3.3.1 General

*HasSubtype References* and *HasProperty References* are the only *ReferenceTypes* that may be used with *ReferenceType Nodes* as *SourceNode*. *ReferenceType Nodes* shall not be the *SourceNode* of other types of *References*.

### 5.3.3.2 HasProperty References

*HasProperty References* are used to identify the *Properties* of a *ReferenceType* and shall only refer to *Nodes* of the *Variable NodeClass*.

The *Property NodeVersion* is used to indicate the version of the *ReferenceType*.

There are no additional *Properties* defined for *ReferenceTypes* in this standard. Additional parts the IEC 62541 series may define additional *Properties* for *ReferenceTypes*.

### 5.3.3.3 HasSubtype References

*HasSubtype References* are used to define subtypes of *ReferenceTypes*. It is not required to provide the *HasSubtype Reference* for the supertype, but it is required that the subtype provides the inverse *Reference* to its supertype. The following rules for subtyping apply.

- The semantic of a *ReferenceType* (e.g. “spans a hierarchy”) is inherited to its subtypes and can be refined there (e.g. “spans a special hierarchy”). The *DisplayName*, and also the *InverseName* for non-symmetric *ReferenceTypes*, reflect the specialization.
- If a *ReferenceType* specifies some constraints (e.g. “allow no loops”) this is inherited and can only be refined (e.g. inheriting “no loops” could be refined as “shall be a tree – only one parent”) but not lowered (e.g. “allow loops”).
- The constraints concerning which *NodeClasses* can be referenced are also inherited and can only be further restricted. That is, if a *ReferenceType* “A” is not allowed to relate an *Object* with an *ObjectType*, this is also true for its subtypes.
- A *ReferenceType* shall have exactly one supertype, except for the *References ReferenceType* defined in 7.2 as the root type of the *ReferenceType* hierarchy. The *ReferenceType* hierarchy does not support multiple inheritances.

## 5.4 View NodeClass

Underlying systems are often large and *Clients* often have an interest in only a specific subset of the data. They do not need, or want, to be burdened with viewing *Nodes* in the *AddressSpace* for which they have no interest.

To address this problem, this standard defines the concept of a *View*. Each *View* defines a subset of the *Nodes* in the *AddressSpace*. The entire *AddressSpace* is the default *View*. Each *Node* in a *View* may contain only a subset of its *References*, as defined by the creator of the *View*. The *View Node* acts as the root for the *Nodes* in the *View*. *Views* are defined using the *View NodeClass*, which is specified in Table 5.

All *Nodes* contained in a *View* shall be accessible starting from the *View Node* when browsing in the context of the *View*. It is not expected that all containing *Nodes* can be browsed directly from the *View Node* but rather browsed from other *Nodes* contained in the *View*.

A *View Node* may not only be used as additional entry point into the *AddressSpace* but as a construct to organize the *AddressSpace* and thus as the only entry point into a subset of the *AddressSpace*. Therefore *Clients* shall not ignore *View Nodes* when exposing the *AddressSpace*. Simple *Clients* that do not deal with *Views* for filtering purposes can, for example, handle a *View Node* like an *Object* of type *FolderType* (see 5.5.3).

**Table 5 – View NodeClass**

| Name                                                                                                      | Use  | Data Type | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |                                                                                                                                                                        |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
|-----------------------------------------------------------------------------------------------------------|------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|--------------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---|------------------------------------------------|-------------|---|----------------------------------------------------------------------------------------------------|--------------|---|----------------------------------------------------------------------------------------------------|----------|-----|-----------------------------------------------|
| <b>Attributes</b>                                                                                         |      |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                                                                        |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
| Base NodeClass Attributes                                                                                 | M    | --        | Inherited from the <i>Base NodeClass</i> . See 5.2.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |       |                                                                                                                                                                        |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
| ContainsNoLoops                                                                                           | M    | Boolean   | If set to “true” this <i>Attribute</i> indicates that by following the <i>References</i> in the context of the <i>View</i> there are no loops, i.e. starting from a <i>Node</i> “A” contained in the <i>View</i> and following the forward <i>References</i> in the context of the <i>View Node</i> “A” will not be reached again. It does not specify that there is only one path starting from the <i>View Node</i> to reach a <i>Node</i> contained in the <i>View</i> .<br>If set to “false” this <i>Attribute</i> indicates that following <i>References</i> in the context of the <i>View</i> may lead to loops.                                                                                                                                                                        |       |                                                                                                                                                                        |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
| EventNotifier                                                                                             | M    | Byte      | The <i>EventNotifier Attribute</i> is used to indicate if the <i>Node</i> can be used to subscribe to <i>Events</i> or to read / write historic <i>Events</i> .<br>The <i>EventNotifier</i> is an 8-bit unsigned integer with the structure defined in the following table.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |                                                                                                                                                                        |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
|                                                                                                           |      |           | <table><tr><th>Field</th><th>Bit</th><th>Description</th></tr><tr><td>SubscribeTo Events</td><td>0</td><td>Indicates if it can be used to subscribe to <i>Events</i> (0 means cannot be used to subscribe to <i>Events</i>, 1 means can be used to subscribe to <i>Events</i>).</td></tr><tr><td>Reserved</td><td>1</td><td>Reserved for future use. Shall always be zero.</td></tr><tr><td>HistoryRead</td><td>2</td><td>Indicates if the history of the <i>Events</i> is readable (0 means not readable, 1 means readable)</td></tr><tr><td>HistoryWrite</td><td>3</td><td>Indicates if the history of the <i>Events</i> is writable (0 means not writable, 1 means writable)</td></tr><tr><td>Reserved</td><td>4:7</td><td>Reserved for future use. Shall always be zero</td></tr></table> | Field | Bit                                                                                                                                                                    | Description | SubscribeTo Events | 0 | Indicates if it can be used to subscribe to <i>Events</i> (0 means cannot be used to subscribe to <i>Events</i> , 1 means can be used to subscribe to <i>Events</i> ). | Reserved | 1 | Reserved for future use. Shall always be zero. | HistoryRead | 2 | Indicates if the history of the <i>Events</i> is readable (0 means not readable, 1 means readable) | HistoryWrite | 3 | Indicates if the history of the <i>Events</i> is writable (0 means not writable, 1 means writable) | Reserved | 4:7 | Reserved for future use. Shall always be zero |
|                                                                                                           |      |           | Field                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Bit   | Description                                                                                                                                                            |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
|                                                                                                           |      |           | SubscribeTo Events                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0     | Indicates if it can be used to subscribe to <i>Events</i> (0 means cannot be used to subscribe to <i>Events</i> , 1 means can be used to subscribe to <i>Events</i> ). |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
|                                                                                                           |      |           | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 1     | Reserved for future use. Shall always be zero.                                                                                                                         |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
|                                                                                                           |      |           | HistoryRead                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 2     | Indicates if the history of the <i>Events</i> is readable (0 means not readable, 1 means readable)                                                                     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
|                                                                                                           |      |           | HistoryWrite                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 3     | Indicates if the history of the <i>Events</i> is writable (0 means not writable, 1 means writable)                                                                     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
|                                                                                                           |      |           | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 4:7   | Reserved for future use. Shall always be zero                                                                                                                          |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
| The second two bits also indicate if the history of the <i>Events</i> is available via the OPC UA Server. |      |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                                                                        |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
|                                                                                                           |      |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                                                                        |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
|                                                                                                           |      |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                                                                        |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
|                                                                                                           |      |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                                                                        |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
|                                                                                                           |      |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                                                                        |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
|                                                                                                           |      |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                                                                        |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
| <b>References</b>                                                                                         |      |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                                                                        |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
| HierarchicalReferences                                                                                    | 0..* |           | Top level <i>Nodes</i> in a <i>View</i> are referenced by <i>hierarchical References</i> (see 7.3).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |       |                                                                                                                                                                        |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
| HasProperty                                                                                               | 0..* |           | <i>HasProperty References</i> identify the <i>Properties</i> of the <i>View</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |       |                                                                                                                                                                        |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
|                                                                                                           |      |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                                                                        |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
| <b>Standard Properties</b>                                                                                |      |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                                                                                                                                                        |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
| NodeVersion                                                                                               | O    | String    | The <i>NodeVersion Property</i> is used to indicate the version of a <i>Node</i> .<br>The <i>NodeVersion Property</i> is updated each time a <i>Reference</i> is added or deleted to the <i>Node</i> the <i>Property</i> belongs to. <i>Attribute</i> value changes do not cause the <i>NodeVersion</i> to change. <i>Clients</i> may read the <i>NodeVersion Property</i> or subscribe to it to determine when the structure of a <i>Node</i> has changed.                                                                                                                                                                                                                                                                                                                                   |       |                                                                                                                                                                        |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |
| ViewVersion                                                                                               | O    | UInt32    | The version number for the <i>View</i> . When <i>Nodes</i> are added to or removed from a <i>View</i> , the value of the <i>ViewVersion Property</i> is updated. <i>Clients</i> may detect changes to the composition of a <i>View</i> using this <i>Property</i> . The value of the <i>ViewVersion</i> shall always be greater than 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                       |       |                                                                                                                                                                        |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                    |              |   |                                                                                                    |          |     |                                               |

The *View NodeClass* inherits the base *Attributes* from the *Base NodeClass* defined in 5.2. It also defines two additional *Attributes*.

The mandatory *ContainsNoLoops Attribute* is set to false if the *Server* is not able to identify if the *View* contains loops or not.

The mandatory *EventNotifier Attribute* identifies if the *View* can be used to subscribe to *Events* that either occur in the content of the *View* or as *ModelChangeEvents* (see 9.32) of the content of the *View* or to read / write the history of the *Events*. A *View* that supports *Events* shall provide all *Events* that occur in any *Object* used as *EventNotifier* that is part of the content of the *View*. In addition, it shall provide all *ModelChangeEvents* that occur in the context of the *View*.

To avoid recursion, i.e. getting all *Events* of the *Server*, the *Server Object* defined in IEC 62541-5 shall never be part of any *View* since it provides all *Events* of the *Server*.

*Views* are defined by the *Server*. The browsing and querying *Services* defined in IEC 62541-4 expect the *NodeId* of a *View Node* to provide these *Services* in the context of the *View*.

*HasProperty References* are used to identify the *Properties* of a *View*. The *Property NodeVersion* is used to indicate the version of the *View Node*. The *ViewVersion Property* indicates the version of the content of the *View*. In contrast to the *NodeVersion*, the *ViewVersion Property* is updated even if *Nodes* not directly referenced by the *View Node* are added to or deleted from the *View*. This *Property* is optional because it might not be possible for *Servers* to detect changes in the *View* contents. *Servers* may also generate a *ModelChangeEvent*, described in 9.32, if *Nodes* are added to or deleted from the *View*. There are no additional *Properties* defined for *Views* in this document. Additional parts of the IEC 62541 series may define additional *Properties* for *Views*.

*Views* can be the *SourceNode* of any *hierarchical Reference*. They shall not be the *SourceNode* of any *non-hierarchical Reference*.

## 5.5 Objects

### 5.5.1 Object NodeClass

*Objects* are used to represent systems, system components, real-world objects and software objects. *Objects* are defined using the *Object NodeClass*, specified in Table 6.

**Table 6 – Object NodeClass**

| Name                      | Use  | Data Type                                                                                                                                                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |
|---------------------------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-----|-------------|--------------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---|------------------------------------------------|-------------|---|-----------------------------------------------------------------------------------------------------|--------------|---|-----------------------------------------------------------------------------------------------------|----------|-----|------------------------------------------------|
| Attributes                |      |                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |
| Base NodeClass Attributes | M    | --                                                                                                                                                                     | Inherited from the <i>Base NodeClass</i> . See 5.2.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |
| EventNotifier             | M    | Byte                                                                                                                                                                   | <p>The <i>EventNotifier Attribute</i> is used to indicate if the <i>Node</i> can be used to subscribe to <i>Events</i> or the read / write historic <i>Events</i>. The <i>EventNotifier</i> is an 8-bit unsigned integer with the structure defined in the following table:</p> <table><tr><th>Field</th><th>Bit</th><th>Description</th></tr><tr><td>SubscribeTo Events</td><td>0</td><td>Indicates if it can be used to subscribe to <i>Events</i> (0 means cannot be used to subscribe to <i>Events</i>, 1 means can be used to subscribe to <i>Events</i>).</td></tr><tr><td>Reserved</td><td>1</td><td>Reserved for future use. Shall always be zero.</td></tr><tr><td>HistoryRead</td><td>2</td><td>Indicates if the history of the <i>Events</i> is readable (0 means not readable, 1 means readable).</td></tr><tr><td>HistoryWrite</td><td>3</td><td>Indicates if the history of the <i>Events</i> is writable (0 means not writable, 1 means writable).</td></tr><tr><td>Reserved</td><td>4:7</td><td>Reserved for future use. Shall always be zero.</td></tr></table> <p>The second two bits also indicate if the history of the <i>Events</i> is available via the OPC UA Server.</p> | Field | Bit | Description | SubscribeTo Events | 0 | Indicates if it can be used to subscribe to <i>Events</i> (0 means cannot be used to subscribe to <i>Events</i> , 1 means can be used to subscribe to <i>Events</i> ). | Reserved | 1 | Reserved for future use. Shall always be zero. | HistoryRead | 2 | Indicates if the history of the <i>Events</i> is readable (0 means not readable, 1 means readable). | HistoryWrite | 3 | Indicates if the history of the <i>Events</i> is writable (0 means not writable, 1 means writable). | Reserved | 4:7 | Reserved for future use. Shall always be zero. |
| Field                     | Bit  | Description                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |
| SubscribeTo Events        | 0    | Indicates if it can be used to subscribe to <i>Events</i> (0 means cannot be used to subscribe to <i>Events</i> , 1 means can be used to subscribe to <i>Events</i> ). |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |
| Reserved                  | 1    | Reserved for future use. Shall always be zero.                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |
| HistoryRead               | 2    | Indicates if the history of the <i>Events</i> is readable (0 means not readable, 1 means readable).                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |
| HistoryWrite              | 3    | Indicates if the history of the <i>Events</i> is writable (0 means not writable, 1 means writable).                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |
| Reserved                  | 4:7  | Reserved for future use. Shall always be zero.                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |
| References                |      |                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |
| HasComponent              | 0..* |                                                                                                                                                                        | <i>HasComponent References</i> identify the <i>DataVariables</i> , the <i>Methods</i> and <i>Objects</i> contained in the <i>Object</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |
| HasProperty               | 0..* |                                                                                                                                                                        | <i>HasProperty References</i> identify the <i>Properties</i> of the <i>Object</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |
| HasModellingRule          | 0..1 |                                                                                                                                                                        | <i>Objects</i> can point to at most one <i>ModellingRule Object</i> using a <i>HasModellingRule Reference</i> (see 6.4.4 for details on <i>ModellingRules</i> ).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |
| HasTypeDefinition         | 1    |                                                                                                                                                                        | The <i>HasTypeDefinition Reference</i> points to the type definition of the <i>Object</i> . Each <i>Object</i> shall have exactly one type definition and therefore be the <i>SourceNode</i> of exactly one <i>HasTypeDefinition Reference</i> pointing to an <i>ObjectType</i> . See 4.5 for a description of type definitions.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |
| HasEventSource            | 0..* |                                                                                                                                                                        | The <i>HasEventSource Reference</i> points to event sources of the <i>Object</i> . <i>References</i> of this type can only be used for <i>Objects</i> having their "SubscribeToEvents" bit set in the <i>EventNotifier Attribute</i> . See 7.17 for details.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |
| HasNotifier               | 0..* |                                                                                                                                                                        | The <i>HasNotifier Reference</i> points to notifiers of the <i>Object</i> . <i>References</i> of this type can only be used for <i>Objects</i> having their "SubscribeToEvents" bit set in the <i>EventNotifier Attribute</i> . See 7.19 for details.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |
| Organizes                 | 0..* |                                                                                                                                                                        | This <i>Reference</i> should be used only for <i>Objects</i> of the <i>ObjectType FolderType</i> (see 5.5.3).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |
| HasDescription            | 0..1 |                                                                                                                                                                        | This <i>Reference</i> shall be used only for <i>Objects</i> of the <i>ObjectType DataTypeEncodingType</i> (see 5.8.4).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |
| <other References>        | 0..* |                                                                                                                                                                        | <i>Objects</i> may contain other <i>References</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |
| Standard Properties       |      |                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |
| NodeVersion               | O    | String                                                                                                                                                                 | The <i>NodeVersion Property</i> is used to indicate the version of a <i>Node</i> . The <i>NodeVersion Property</i> is updated each time a <i>Reference</i> is added or deleted to the <i>Node</i> the <i>Property</i> belongs to. <i>Attribute</i> value changes do not cause the <i>NodeVersion</i> to change. <i>Clients</i> may read the <i>NodeVersion Property</i> or subscribe to it to determine when the structure of a <i>Node</i> has changed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |
| Icon                      | O    | Image                                                                                                                                                                  | The <i>Icon Property</i> provides an image that can be used by <i>Clients</i> when displaying the <i>Node</i> . It is expected that the <i>Icon Property</i> contains a relatively small image.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |
| NamingRule                | O    | NamingRuleType                                                                                                                                                         | The <i>NamingRule Property</i> defines the <i>NamingRule</i> of a <i>ModellingRule</i> (see 6.4.4.2 for details). This <i>Property</i> shall only be used for <i>Objects</i> of the type <i>ModellingRuleType</i> defined in 6.4.4.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |                    |   |                                                                                                                                                                        |          |   |                                                |             |   |                                                                                                     |              |   |                                                                                                     |          |     |                                                |

The *Object NodeClass* inherits the base *Attributes* from the *Base NodeClass* defined in 5.2.

The mandatory *EventNotifier Attribute* identifies whether the *Object* can be used to subscribe to *Events* or to read and write the history of the *Events*.

The *Object NodeClass* uses the *HasComponent Reference* to define the *DataVariables*, *Objects* and *Methods* of an *Object*.

It uses the *HasProperty Reference* to define the *Properties* of an *Object*. The *Property NodeVersion* is used to indicate the version of the *Object*. The *Property Icon* provides an icon of the *Object*. The *Property NamingRule* defines the *NamingRule* of a *ModellingRule* and shall only be applied to *Objects* of type *ModellingRuleType*. There are no additional *Properties* defined for *Objects* in this document. Additional parts of the IEC 62541 series may define additional *Properties* for *Objects*.

To specify its *ModellingRule*, an *Object* can use at most one *HasModellingRule Reference* pointing to a *ModellingRule Object*. *ModellingRules* are defined in 6.4.4.

*HasNotifier* and *HasEventSource References* are used to provide information about eventing and can only be applied to *Objects* used as event notifiers. Details are defined in 7.17 and 7.19.

The *HasTypeDefinition Reference* points to the *ObjectType* used as type definition of the *Object*.

*Objects* may use any additional *References* to define relationships to other *Nodes*. No restrictions are placed on the types of *References* used or on the *NodeClasses* of the *Nodes* that may be referenced. However, restrictions may be defined by the *ReferenceType* excluding its use for *Objects*. Standard *ReferenceTypes* are described in Clause 7.

If the *Object* is used as an *InstanceDeclaration* (see 4.5) then all *Nodes* referenced with *hierarchical References* in a forward direction shall have unique *BrowseNames* in the context of this *Object*.

If the *Object* is created based on an *InstanceDeclaration* then it shall have the same *BrowseName* as its *InstanceDeclaration*.

### 5.5.2 **ObjectType NodeClass**

*ObjectTypes* provide definitions for *Objects*. *ObjectTypes* are defined using the *ObjectType NodeClass*, which is specified in Table 7.

Table 7 – ObjectType NodeClass

| Name                       | Use  | Data Type | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------------------|------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Attributes</b>          |      |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Base NodeClass Attributes  | M    | --        | Inherited from the <i>Base NodeClass</i> . See 5.2.                                                                                                                                                                                                                                                                                                                                                                                                         |
| IsAbstract                 | M    | Boolean   | A boolean <i>Attribute</i> with the following values:<br>TRUE it is an abstract <i>ObjectType</i> , i.e. no <i>Objects</i> of this type shall exist, only <i>Objects</i> of its subtypes.<br>FALSE it is not an abstract <i>ObjectType</i> , i.e. <i>Objects</i> of this type can exist.                                                                                                                                                                    |
| <b>References</b>          |      |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| HasComponent               | 0..* |           | <i>HasComponent References</i> identify the <i>DataVariables</i> , the <i>Methods</i> , and <i>Objects</i> contained in the <i>ObjectType</i> .<br>If and how the referenced <i>Nodes</i> are instantiated when an <i>Object</i> of this type is instantiated, is specified in 6.4.                                                                                                                                                                         |
| HasProperty                | 0..* |           | <i>HasProperty References</i> identify the <i>Properties</i> of the <i>ObjectType</i> . If and how the <i>Properties</i> are instantiated when an <i>Object</i> of this type is instantiated, is specified in 6.4.                                                                                                                                                                                                                                          |
| HasSubtype                 | 0..* |           | <i>HasSubtype References</i> identify <i>ObjectTypes</i> that are subtypes of this type. The inverse <i>SubtypeOf Reference</i> identifies the parent type of this type.                                                                                                                                                                                                                                                                                    |
| GeneratesEvent             | 0..* |           | <i>GeneratesEvent References</i> identify the type of <i>Events</i> instances of this type may generate.                                                                                                                                                                                                                                                                                                                                                    |
| <other References>         | 0..* |           | <i>ObjectTypes</i> may contain other <i>References</i> that can be instantiated by <i>Objects</i> defined by this <i>ObjectType</i> .                                                                                                                                                                                                                                                                                                                       |
| <b>Standard Properties</b> |      |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| NodeVersion                | O    | String    | The <i>NodeVersion Property</i> is used to indicate the version of a <i>Node</i> .<br>The <i>NodeVersion Property</i> is updated each time a <i>Reference</i> is added or deleted to the <i>Node</i> the <i>Property</i> belongs to. <i>Attribute</i> value changes do not cause the <i>NodeVersion</i> to change. <i>Clients</i> may read the <i>NodeVersion Property</i> or subscribe to it to determine when the structure of a <i>Node</i> has changed. |
| Icon                       | O    | Image     | The <i>Icon Property</i> provides an image that can be used by <i>Clients</i> when displaying the <i>Node</i> . It is expected that the <i>Icon Property</i> contains a relatively small image.                                                                                                                                                                                                                                                             |

The *ObjectType NodeClass* inherits the base *Attributes* from the *Base NodeClass* defined in 5.2. The additional *IsAbstract Attribute* indicates if the *ObjectType* is abstract or not.

The *ObjectType NodeClass* uses the *HasComponent References* to define the *DataVariables*, *Objects*, and *Methods* for it.

The *HasProperty Reference* is used to identify the *Properties*. The *Property NodeVersion* is used to indicate the version of the *ObjectType*. The *Property Icon* provides an icon of the *ObjectType*. There are no additional *Properties* defined for *ObjectTypes* in this document. Additional parts of the IEC 62541 series may define additional *Properties* for *ObjectTypes*.

*HasSubtype References* are used to subtype *ObjectTypes*. *ObjectType* subtypes inherit the general semantics from the parent type. The general rules for subtyping apply as defined in Clause 6. It is not required to provide the *HasSubtype Reference* for the supertype, but it is required that the subtype provides the inverse *Reference* to its supertype.

*GeneratesEvent References* identify the type of *Events* that instances of the *ObjectType* may generate. These *Objects* may be the source of an *Event* of the specified type or one of its subtypes. *Servers* should make *GeneratesEvent References* bidirectional *References*. However, it is allowed to be unidirectional when the *Server* is not able to expose the inverse direction pointing from the *EventType* to each *ObjectType* supporting the *EventType*. Note that the *EventNotifier Attribute* of an *Object* and the *GeneratesEvent References* of its *ObjectType* are completely unrelated. *Objects* that can generate *Events* might not be used as *Objects* to which *Clients* subscribe to get the corresponding *Event* notifications.

*GeneratesEvent References* are optional, i.e. *Objects* may generate *Events* of an *EventType* that is not exposed by its *ObjectType*.

*ObjectTypes* may use any additional *References* to define relationships to other *Nodes*. No restrictions are placed on the types of *References* used or on the *NodeClasses* of the *Nodes* that may be referenced. However, restrictions may be defined by the *ReferenceType* excluding its use for *ObjectTypes*. Standard *ReferenceTypes* are described in Clause 7.

All *Nodes* referenced with *hierarchical References* shall have unique *BrowseNames* in the context of an *ObjectType* (see 4.5).

### 5.5.3 Standard ObjectType FolderType

The *ObjectType FolderType* is formally defined in IEC 62541-5. Its purpose is to provide *Objects* that have no other semantic than organizing of the *AddressSpace*. A special *ReferenceType* is introduced for those *Folder Objects*, the *Organizes ReferenceType*. The *SourceNode* of such a *Reference* should always be a *View* or an *Object* of the *ObjectType FolderType*; the *TargetNode* can be of any *NodeClass*. *Organizes References* can be used in any combination with *HasChild References* (*HasComponent*, *HasProperty*, etc.; see 7.5) and do not prevent loops. Thus, they can be used to span multiple hierarchies.

### 5.5.4 Client-side creation of Objects of an ObjectType

*Objects* are always based on an *ObjectType*, i.e. they have a *HasTypeDefinition Reference* pointing to its *ObjectType*.

*Clients* can create *Objects* using the *AddNodes Service* defined in IEC 62541-4. The *Service* requires specifying the *TypeDefinitionNode* of the *Object*. An *Object* created by the *AddNodes Service* contains all components defined by its *ObjectType* dependent on the *ModellingRules* specified for the components. However, the *Server* may add additional components and *References* to the *Object* and its components that are not defined by the *ObjectType*. This behaviour is *Server* dependent. The *ObjectType* only specifies the minimum set of components that shall exist for each *Object* of an *ObjectType*.

In addition to the *AddNodes Service*, *ObjectTypes* may have a special *Method* with the *BrowseName* "Create". This *Method* is used to create an *Object* of this *ObjectType*. This *Method* may be useful for the creation of *Objects* where the semantic of the creation should differ from the default behaviour expected in the context of the *AddNodes Service*. For example, the values should directly differ from the default values or additional *Objects* should be added, etc. The input and output arguments of this *Method* depend on the *ObjectType*; the only commonality is the *BrowseName* identifying that this *Method* will create an *Object* based on the *ObjectType*. *Servers* should not provide a *Method* on an *ObjectType* with the *BrowseName* "Create" for any other purpose than creating *Objects* of the *ObjectType*.

## 5.6 Variables

### 5.6.1 General

Two types of *Variables* are defined, *Properties* and *DataVariables*. Although they differ in the way they are used as described in 4.4 and have different constraints described in the remainder of 5.6 they use the same *NodeClass* described in 5.6.2. The constraints of *Properties* based on this *NodeClass* are defined in 5.6.3, the constraints of *DataVariables* in 5.6.4.

### 5.6.2 Variable NodeClass

*Variables* are used to represent values which may be simple or complex. *Variables* are defined by *VariableTypes*, as specified in 5.6.5.

*Variables* are always defined as *Properties* or *DataVariables* of other *Nodes* in the *AddressSpace*. They are never defined by themselves. A *Variable* is always part of at least one other *Node*, but may be related to any number of other *Nodes*. *Variables* are defined using the *Variable NodeClass*, specified in Table 8.

**Table 8 – Variable NodeClass**

| Name                      | Use | Data Type                                                                                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |                |   |                                                                                                            |          |     |                                                |
|---------------------------|-----|------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-----|-------------|-------------|---|--------------------------------------------------------------------------------------|--------------|---|--------------------------------------------------------------------------------------|-------------|---|---------------------------------------------------------------------------------------------|--------------|---|---------------------------------------------------------------------------------------------|----------------|---|------------------------------------------------------------------------------------------------------------|----------|-----|------------------------------------------------|
| Attributes                |     |                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |                |   |                                                                                                            |          |     |                                                |
| Base NodeClass Attributes | M   | --                                                                                                         | Inherited from the <i>Base NodeClass</i> . See 5.2.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |                |   |                                                                                                            |          |     |                                                |
| Value                     | M   | Defined by the <i>DataType Attribute</i>                                                                   | The most recent value of the <i>Variable</i> that the <i>Server</i> has. Its data type is defined by the <i>DataType Attribute</i> . It is the only <i>Attribute</i> that does not have a data type associated with it. This allows all <i>Variables</i> to have a value defined by the same <i>Value Attribute</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |                |   |                                                                                                            |          |     |                                                |
| DataType                  | M   | NodeId                                                                                                     | <i>NodeId</i> of the <i>DataType</i> definition for the <i>Value Attribute</i> . Standard <i>DataTypes</i> are defined in Clause 8.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |                |   |                                                                                                            |          |     |                                                |
| ValueRank                 | M   | Int32                                                                                                      | This <i>Attribute</i> indicates whether the <i>Value Attribute</i> of the <i>Variable</i> is an array and how many dimensions the array has.<br>It may have the following values:<br><i>n</i> > 1: the Value is an array with the specified number of dimensions.<br>OneDimension (1): The value is an array with one dimension.<br>OneOrMoreDimensions (0): The value is an array with one or more dimensions.<br>Scalar (-1): The value is not an array.<br>Any (-2): The value can be a scalar or an array with any number of dimensions.<br>ScalarOrOneDimension (-3): The value can be a scalar or a one dimensional array.<br>NOTE All <i>DataTypes</i> are considered to be scalar, even if they have array-like semantics like <i>ByteString</i> and <i>String</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |                |   |                                                                                                            |          |     |                                                |
| ArrayDimensions           | O   | UInt32[]                                                                                                   | This <i>Attribute</i> specifies the length of each dimension for an array value. The <i>Attribute</i> is intended to describe the capability of the <i>Variable</i> , not the current size.<br>The number of elements shall be equal to the value of the <i>ValueRank Attribute</i> . Shall be null if <i>ValueRank</i> ≤ 0.<br>A value of 0 for an individual dimension indicates that the dimension has a variable length.<br>For example, if a <i>Variable</i> is defined by the following C array:<br>Int32 myArray[346];<br>then this <i>Variable's DataType</i> would point to an Int32, the <i>Variable's ValueRank</i> has the value 1 and the <i>ArrayDimensions</i> is an array with one entry having the value 346.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |                |   |                                                                                                            |          |     |                                                |
| AccessLevel               | M   | Byte                                                                                                       | <div>The <i>AccessLevel Attribute</i> is used to indicate how the <i>Value</i> of a <i>Variable</i> can be accessed (read/write) and if it contains current and/or historic data. The <i>AccessLevel</i> does not take any user access rights into account, i.e. although the <i>Variable</i> is writable this may be restricted to a certain user / user group.<br/>The <i>AccessLevel</i> is an 8-bit unsigned integer with the structure defined in the following table:</div> <table><tr><th>Field</th><th>Bit</th><th>Description</th></tr><tr><td>CurrentRead</td><td>0</td><td>Indicates if the current value is readable (0 means not readable, 1 means readable).</td></tr><tr><td>CurrentWrite</td><td>1</td><td>Indicates if the current value is writable (0 means not writable, 1 means writable).</td></tr><tr><td>HistoryRead</td><td>2</td><td>Indicates if the history of the value is readable (0 means not readable, 1 means readable).</td></tr><tr><td>HistoryWrite</td><td>3</td><td>Indicates if the history of the value is writable (0 means not writable, 1 means writable).</td></tr><tr><td>SemanticChange</td><td>4</td><td>Indicates if the <i>Variable</i> used as <i>Property</i> generates <i>SemanticChangeEvents</i> (see 9.33).</td></tr><tr><td>Reserved</td><td>5:7</td><td>Reserved for future use. Shall always be zero.</td></tr></table> <div>The first two bits also indicate if a current value of this <i>Variable</i> is available and the second two bits indicates if the history of the <i>Variable</i> is available via the OPC UA <i>Server</i>.</div> | Field | Bit | Description | CurrentRead | 0 | Indicates if the current value is readable (0 means not readable, 1 means readable). | CurrentWrite | 1 | Indicates if the current value is writable (0 means not writable, 1 means writable). | HistoryRead | 2 | Indicates if the history of the value is readable (0 means not readable, 1 means readable). | HistoryWrite | 3 | Indicates if the history of the value is writable (0 means not writable, 1 means writable). | SemanticChange | 4 | Indicates if the <i>Variable</i> used as <i>Property</i> generates <i>SemanticChangeEvents</i> (see 9.33). | Reserved | 5:7 | Reserved for future use. Shall always be zero. |
| Field                     | Bit | Description                                                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |                |   |                                                                                                            |          |     |                                                |
| CurrentRead               | 0   | Indicates if the current value is readable (0 means not readable, 1 means readable).                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |                |   |                                                                                                            |          |     |                                                |
| CurrentWrite              | 1   | Indicates if the current value is writable (0 means not writable, 1 means writable).                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |                |   |                                                                                                            |          |     |                                                |
| HistoryRead               | 2   | Indicates if the history of the value is readable (0 means not readable, 1 means readable).                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |                |   |                                                                                                            |          |     |                                                |
| HistoryWrite              | 3   | Indicates if the history of the value is writable (0 means not writable, 1 means writable).                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |                |   |                                                                                                            |          |     |                                                |
| SemanticChange            | 4   | Indicates if the <i>Variable</i> used as <i>Property</i> generates <i>SemanticChangeEvents</i> (see 9.33). |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |                |   |                                                                                                            |          |     |                                                |
| Reserved                  | 5:7 | Reserved for future use. Shall always be zero.                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |                |   |                                                                                                            |          |     |                                                |

| Name                       | Use  | Data Type                                                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |          |     |                                                |
|----------------------------|------|---------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-----|-------------|-------------|---|--------------------------------------------------------------------------------------|--------------|---|--------------------------------------------------------------------------------------|-------------|---|---------------------------------------------------------------------------------------------|--------------|---|---------------------------------------------------------------------------------------------|----------|-----|------------------------------------------------|
| UserAccessLevel            | M    | Byte                                                                                        | <p>The <i>UserAccessLevel Attribute</i> is used to indicate how the <i>Value</i> of a <i>Variable</i> can be accessed (read/write) and if it contains current or historic data taking user access rights into account.</p> <p>The <i>UserAccessLevel</i> is an 8-bit unsigned integer with the structure defined in the following table:</p> <table><tr><th>Field</th><th>Bit</th><th>Description</th></tr><tr><td>CurrentRead</td><td>0</td><td>Indicates if the current value is readable (0 means not readable, 1 means readable).</td></tr><tr><td>CurrentWrite</td><td>1</td><td>Indicates if the current value is writable (0 means not writable, 1 means writable).</td></tr><tr><td>HistoryRead</td><td>2</td><td>Indicates if the history of the value is readable (0 means not readable, 1 means readable).</td></tr><tr><td>HistoryWrite</td><td>3</td><td>Indicates if the history of the value is writable (0 means not writable, 1 means writable).</td></tr><tr><td>Reserved</td><td>4:7</td><td>Reserved for future use. Shall always be zero.</td></tr></table> <p>The first two bits also indicate if a current value of this <i>Variable</i> is available and the second two bits indicate if the history of the <i>Variable</i> is available via the OPC UA Server.</p> | Field | Bit | Description | CurrentRead | 0 | Indicates if the current value is readable (0 means not readable, 1 means readable). | CurrentWrite | 1 | Indicates if the current value is writable (0 means not writable, 1 means writable). | HistoryRead | 2 | Indicates if the history of the value is readable (0 means not readable, 1 means readable). | HistoryWrite | 3 | Indicates if the history of the value is writable (0 means not writable, 1 means writable). | Reserved | 4:7 | Reserved for future use. Shall always be zero. |
| Field                      | Bit  | Description                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |          |     |                                                |
| CurrentRead                | 0    | Indicates if the current value is readable (0 means not readable, 1 means readable).        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |          |     |                                                |
| CurrentWrite               | 1    | Indicates if the current value is writable (0 means not writable, 1 means writable).        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |          |     |                                                |
| HistoryRead                | 2    | Indicates if the history of the value is readable (0 means not readable, 1 means readable). |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |          |     |                                                |
| HistoryWrite               | 3    | Indicates if the history of the value is writable (0 means not writable, 1 means writable). |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |          |     |                                                |
| Reserved                   | 4:7  | Reserved for future use. Shall always be zero.                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |          |     |                                                |
| MinimumSamplingInterval    | O    | Duration                                                                                    | <p>The <i>MinimumSamplingInterval Attribute</i> indicates how “current” the <i>Value</i> of the <i>Variable</i> will be kept. It specifies (in milliseconds) how fast the <i>Server</i> can reasonably sample the value for changes (see IEC 62541-4 for a detailed description of sampling interval).</p> <p>A <i>MinimumSamplingInterval</i> of 0 indicates that the <i>Server</i> is to monitor the item continuously. A <i>MinimumSamplingInterval</i> of -1 means indeterminate.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |          |     |                                                |
| Historizing                | M    | Boolean                                                                                     | <p>The <i>Historizing Attribute</i> indicates whether the <i>Server</i> is actively collecting data for the history of the <i>Variable</i>. This differs from the <i>AccessLevel Attribute</i> which identifies if the <i>Variable</i> has any historical data. A value of TRUE indicates that the <i>Server</i> is actively collecting data. A value of FALSE indicates the <i>Server</i> is not actively collecting data. Default value is FALSE.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |          |     |                                                |
| <b>References</b>          |      |                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |          |     |                                                |
| HasModellingRule           | 0..1 |                                                                                             | <p><i>Variables</i> can point to at most one <i>ModellingRule Object</i> using a <i>HasModellingRule Reference</i> (see 6.4.4 for details on <i>ModellingRules</i>).</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |          |     |                                                |
| HasProperty                | 0..* |                                                                                             | <p><i>HasProperty References</i> are used to identify the <i>Properties</i> of a <i>DataVariable</i>.</p> <p><i>Properties</i> are not allowed to be the <i>SourceNode</i> of <i>HasProperty References</i>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |          |     |                                                |
| HasComponent               | 0..* |                                                                                             | <p><i>HasComponent References</i> are used by complex <i>DataVariables</i> to identify their composed <i>DataVariables</i>.</p> <p><i>Properties</i> are not allowed to use this <i>Reference</i>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |          |     |                                                |
| HasTypeDefinition          | 1    |                                                                                             | <p>The <i>HasTypeDefinition Reference</i> points to the type definition of the <i>Variable</i>. Each <i>Variable</i> shall have exactly one type definition and therefore be the <i>SourceNode</i> of exactly one <i>HasTypeDefinition Reference</i> pointing to a <i>VariableType</i>. See 4.5 for a description of type definitions.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |          |     |                                                |
| <other References>         | 0..* |                                                                                             | <p><i>Data Variables</i> may be the <i>SourceNode</i> of any other <i>References</i>.</p> <p><i>Properties</i> may only be the <i>SourceNode</i> of any non-hierarchical <i>Reference</i>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |          |     |                                                |
| <b>Standard Properties</b> |      |                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |          |     |                                                |
| NodeVersion                | O    | String                                                                                      | <p>The <i>NodeVersion Property</i> is used to indicate the version of a <i>DataVariable</i>. It does not apply to <i>Properties</i>.</p> <p>The <i>NodeVersion Property</i> is updated each time a <i>Reference</i> is added or deleted to the <i>Node</i> the <i>Property</i> belongs to. <i>Attribute</i> value changes except for the <i>DataType Attribute</i> do not cause the <i>NodeVersion</i> to change. <i>Clients</i> may read the <i>NodeVersion Property</i> or subscribe to it to determine when the structure of a <i>Node</i> has changed.</p> <p>Although the relationship of a <i>Variable</i> to its <i>DataType</i> is not modelled using <i>References</i>, changes to the <i>DataType Attribute</i> of a <i>Variable</i> lead to an update of the <i>NodeVersion Property</i>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |          |     |                                                |
| LocalTime                  | O    | TimeZone<br>DataType                                                                        | <p>The <i>LocalTime Property</i> is only used for <i>DataVariables</i>. It does not apply to <i>Properties</i>.</p> <p>This <i>Property</i> is a structure containing the Offset and the DaylightSavingInOffset flag. The Offset specifies the time difference (in minutes) between the SourceTimestamp (UTC) associated with the value and the time at the location in which the value was obtained. The SourceTimestamp is defined in IEC 62541-4.</p> <p>If DaylightSavingInOffset is TRUE, then Standard/Daylight savings time (DST) at the originating location is in effect and Offset includes the DST correction. If FALSE then the Offset does not include DST correction and</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |       |     |             |             |   |                                                                                      |              |   |                                                                                      |             |   |                                                                                             |              |   |                                                                                             |          |     |                                                |

| Name               | Use | Data Type      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------|-----|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    |     |                | DST may or may not have been in effect.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| DataTypeVersion    | O   | String         | Only used for <i>Variables</i> of the <i>VariableType</i> <i>DataTypeDictionaryType</i> and <i>DataTypeDescriptionType</i> as described in 5.8.                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| DictionaryFragment | O   | ByteString     | Only used for <i>Variables</i> of the <i>VariableType</i> <i>DataTypeDescriptionType</i> as described in 5.8.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| AllowNulls         | O   | Boolean        | The <i>AllowNulls</i> <i>Property</i> is only used for <i>DataVariables</i> . It does not apply to <i>Properties</i> .<br>This <i>Property</i> specifies if a null value is allowed for the <i>Value</i> <i>Attribute</i> of the <i>DataVariable</i> . If it is set to true, the <i>Server</i> may return null values and accept writing of null values. If it is set to false, the <i>Server</i> shall never return a null value and shall reject any request writing a null value.<br>If this <i>Property</i> is not provided, it is <i>Server</i> -specific if null values are allowed or not. |
| ValueAsText        | O   | Localized Text | Only used for <i>DataVariables</i> having an <i>Enumeration</i> <i>DataType</i> .<br>This optional <i>Property</i> provides the localized text representation of the enumeration value. It can be used by <i>Clients</i> only interested in displaying the text to subscribe to the <i>Property</i> instead of the value attribute.                                                                                                                                                                                                                                                               |
| MaxStringLength    | O   | UInt32         | Only used for <i>DataVariables</i> having a <i>String</i> <i>DataType</i> .<br>This optional <i>Property</i> indicates the maximum number of characters supported by the <i>DataVariable</i> .                                                                                                                                                                                                                                                                                                                                                                                                    |
| MaxArrayLength     | O   | UInt32         | Only used for <i>DataVariables</i> having its <i>ValueRank</i> <i>Attribute</i> not set to scalar.<br>This optional <i>Property</i> indicates the maximum length of an array supported by the <i>DataVariable</i> . In a multidimensional array it indicates the overall length. For example, a three-dimensional array of 2 x 3 x 10 has the array length of 60.                                                                                                                                                                                                                                 |
| EngineeringUnits   | O   | EU Information | Only used for <i>DataVariables</i> having a <i>Number</i> <i>DataType</i> .<br>This optional <i>Property</i> indicates the engineering units for the value of the <i>DataVariable</i> (e.g. hertz or seconds). Details about the <i>Property</i> and what engineering units should be used are defined in IEC 62541-8. The <i>DataType</i> <i>EUInformation</i> is also defined in IEC 62541-8.                                                                                                                                                                                                   |

The *Variable NodeClass* inherits the base *Attributes* from the *Base NodeClass* defined in 5.2.

The *Variable NodeClass* also defines a set of *Attributes* that describe the *Variable*'s Runtime value. The *Value* *Attribute* represents the *Variable* value. The *DataType*, *ValueRank* and *ArrayDimensions* *Attributes* provide the capability to describe simple and complex values.

The *AccessLevel* *Attribute* indicates the accessibility of the *Value* of a *Variable* not taking user access rights into account. If the OPC UA *Server* does not have the ability to get the *AccessLevel* information from the underlying system then it should state that it is readable and writable. If a read or write operation is called on the *Variable* then the *Server* should transfer this request and return the corresponding *StatusCode* even if such a request is rejected. *StatusCodes* are defined in IEC 62541-4.

The *SemanticChange* bit of the *AccessLevel* *Attribute* shall be set when the *Property* describes the semantic of the *Node* that owns the *Property* and changes of the *Property* value will generate *SemanticChangeEvents*. For example, a *Property* describing the engineering unit of a *DataVariable* will have the bit set, whereas a *Property* containing an Icon of the *DataVariable* will not. This behaviour is exactly the same as described by the *SemanticsChanged* bit of the *StatusCode* defined in IEC 62541-4. However, if subscribing to a *Variable* then one should look at the *StatusCode* to identify if the semantic has changed in order to receive this information before processing the value of the *Variable*.

The *UserAccessLevel* *Attribute* indicates the accessibility of the *Value* of a *Variable* taking user access rights into account. If the OPC UA *Server* does not have the ability to get any user access rights related information from the underlying system then it should use the same bit mask as used in the *AccessLevel* *Attribute*. The *UserAccessLevel* *Attribute* can restrict the accessibility indicated by the *AccessLevel* *Attribute*, but not exceed it.

The *MinimumSamplingInterval* *Attribute* specifies how fast the *Server* can reasonably sample the *value* for changes. The accuracy of this value (the ability of the *Server* to attain "best case" performance) can be greatly affected by system load and other factors.

The *Historizing Attribute* indicates whether the *Server* is actively collecting data for the history of the *Variable*. See IEC 62541-11 for details on historizing *Variables*.

*Clients* may read or write *Variable* values, or monitor them for value changes, as specified in IEC 62541-4. IEC 62541-8 defines additional rules when using the *Services* for automation data.

To specify its *ModellingRule*, a *Variable* can use at most one *HasModellingRule Reference* pointing to a *ModellingRule Object*. *ModellingRules* are defined in 6.4.4.

If the *Variable* is created based on an *InstanceDeclaration* (see 4.5) it shall have the same *BrowseName* as its *InstanceDeclaration*.

The other *References* are described separately for *Properties* and *DataVariables* in the remainder of 5.6

### 5.6.3 Properties

*Properties* are used to define the characteristics of *Nodes*. *Properties* are defined using the *Variable NodeClass*, specified in Table 8. However, they restrict their use.

*Properties* are the leaf of any hierarchy; therefore they shall not be the *SourceNode* of any *hierarchical References*. This includes the *HasComponent* or *HasProperty Reference*, that is, *Properties* do not contain *Properties* and cannot expose their complex structure. However, they may be the *SourceNode* of any *non-hierarchical References*.

The *HasTypeDefinition Reference* points to the *VariableType* of the *Property*. Since *Properties* are uniquely identified by their *BrowseName*, all *Properties* shall point to the *PropertyType* defined in IEC 62541-5.

*Properties* shall always be defined in the context of another *Node* and shall be the *TargetNode* of at least one *HasProperty Reference*. To distinguish them from *DataVariables*, they shall not be the *TargetNode* of any *HasComponent Reference*. Thus, a *HasProperty Reference* pointing to a *Variable Node* defines this *Node* as a *Property*.

The *BrowseName* of a *Property* is always unique in the context of a *Node*. It is not permitted for a *Node* to refer to two *Variables* using *HasProperty References* having the same *BrowseName*.

### 5.6.4 DataVariable

*DataVariables* represent the content of an *Object*. *DataVariables* are defined using the *Variable NodeClass*, specified in Table 8.

*DataVariables* identify their *Properties* using *HasProperty References*. Complex *DataVariables* use *HasComponent References* to expose their component *DataVariables*.

The *Property NodeVersion* indicates the version of the *DataVariable*. The *Property LocalTime* indicates the difference between the *SourceTimestamp* of the value and the standard time at the location in which the value was obtained. The *Property DataTypeVersion* is used only for *DataTypeDictionaries* and *DataTypeDescriptions* as defined in 5.8. The *Standard Property DictionaryFragment* is used only for *DataTypeDescriptions* as defined in 5.8. The *Property AllowNulls* indicates if null values are allowed for the *Value Attribute*. The *Property ValueAsText* provides a localized text representation for enumeration values. The *Property MaxStringLength* indicates the maximum allowed length of a string value and the *Property MaxArrayLength* the maximum allowed array length of the value. The *Property EngineeringUnits* indicates the engineering units of the value. There are no additional *Properties* defined for *DataVariables* in this part of this document. Additional parts of the

IEC 62541 series may define additional *Properties* for *DataVariables*. IEC 62541-8 defines a set of *Properties* that can be used for *DataVariables*.

*DataVariables* may use additional *References* to define relationships to other *Nodes*. No restrictions are placed on the types of *References* used or on the *NodeClasses* of the *Nodes* that may be referenced. However, restrictions may be defined by the *ReferenceType* excluding its use for *DataVariables*. Standard *ReferenceTypes* are described in Clause 7.

A *DataVariable* is intended to be defined in the context of an *Object*. However, complex *DataVariables* may expose other *DataVariables*, and *ObjectTypes* and complex *VariableTypes* may also contain *DataVariables*. Therefore each *DataVariable* shall be the *TargetNode* of at least one *HasComponent Reference* coming from an *Object*, an *ObjectType*, a *DataVariable* or a *VariableType*. *DataVariables* shall not be the *TargetNode* of any *HasProperty References*. Therefore, a *HasComponent Reference* pointing to a *Variable Node* identifies it as a *DataVariable*.

The *HasTypeDefinition Reference* points to the *VariableType* used as type definition of the *DataVariable*.

If the *DataVariable* is used as *InstanceDeclaration* (see 4.5) all *Nodes* referenced with *hierarchical References* in the forward direction shall have unique *BrowseNames* in the context of this *DataVariable*.

#### 5.6.5 VariableType NodeClass

*VariableTypes* are used to provide type definitions for *Variables*. *VariableTypes* are defined using the *VariableType NodeClass*, as specified in Table 9.

Table 9 – VariableType NodeClass

| Name                       | Use  | Data Type                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------------------|------|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Attributes</b>          |      |                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Base NodeClass Attributes  | M    | --                                        | Inherited from the <i>Base NodeClass</i> . See 5.2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Value                      | O    | Defined by the <i>Data Type attribute</i> | The default <i>Value</i> for instances of this type.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| DataType                   | M    | NodeId                                    | <i>NodeId</i> of the data type definition for instances of this type.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| ValueRank                  | M    | Int32                                     | This <i>Attribute</i> indicates whether the <i>Value Attribute</i> of the <i>VariableType</i> is an array and how many dimensions the array has.<br>It may have the following values:<br>$n > 1$ : the <i>Value</i> is an array with the specified number of dimensions.<br>OneDimension (1): The value is an array with one dimension.<br>OneOrMoreDimensions (0): The value is an array with one or more dimensions.<br>Scalar (-1): The value is not an array.<br>Any (-2): The value can be a scalar or an array with any number of dimensions.<br>ScalarOrOneDimension (-3): The value can be a scalar or a one dimensional array.<br>NOTE All <i>DataTypes</i> are considered to be scalar, even if they have array-like semantics like <i>ByteString</i> and <i>String</i> . |
| ArrayDimensions            | O    | UInt32[]                                  | This <i>Attribute</i> specifies the length of each dimension for an array value. The <i>Attribute</i> is intended to describe the capability of the <i>VariableType</i> , not the current size.<br>The number of elements shall be equal to the value of the <i>ValueRank Attribute</i> . Shall be null if <i>ValueRank</i> $\leq 0$ .<br>A value of 0 for an individual dimension indicates that the dimension has a variable length.<br>For example, if a <i>VariableType</i> is defined by the following C array:<br>Int32 myArray[346];<br>then this <i>VariableType's DataType</i> would point to an Int32, the <i>VariableType's ValueRank</i> has the value 1 and the <i>ArrayDimensions</i> is an array with one entry having the value 346.                                |
| IsAbstract                 | M    | Boolean                                   | A boolean <i>Attribute</i> with the following values:<br>TRUE it is an abstract <i>VariableType</i> , i.e. no <i>Variable</i> of this type shall exist, only of its subtypes.<br>FALSE it is not an abstract <i>VariableType</i> , i.e. <i>Variables</i> of this type can exist.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>References</b>          |      |                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| HasProperty                | 0..* |                                           | <i>HasProperty References</i> are used to identify the <i>Properties</i> of the <i>VariableType</i> . The referenced <i>Nodes</i> may be instantiated by the instances of this type, depending on the <i>ModellingRules</i> defined in 6.4.4.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| HasComponent               | 0..* |                                           | <i>HasComponent References</i> are used for complex <i>VariableTypes</i> to identify their containing <i>DataVariables</i> . Complex <i>VariableTypes</i> can only be used for <i>DataVariables</i> . The referenced <i>Nodes</i> may be instantiated by the instances of this type, depending on the <i>ModellingRules</i> defined in 6.4.4.                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| HasSubtype                 | 0..* |                                           | <i>HasSubtype References</i> identify <i>VariableTypes</i> that are subtypes of this type. The inverse <i>subtype of Reference</i> identifies the parent type of this type.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| GeneratesEvent             | 0..* |                                           | <i>GeneratesEvent References</i> identify the type of <i>Events</i> instances of this type may generate.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <other References>         | 0..* |                                           | <i>VariableTypes</i> may contain other <i>References</i> that can be instantiated by <i>Variables</i> defined by this <i>VariableType</i> . <i>ModellingRules</i> are defined in 6.4.4.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Standard Properties</b> |      |                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| NodeVersion                | O    | String                                    | The <i>NodeVersion Property</i> is used to indicate the version of a <i>Node</i> . The <i>NodeVersion Property</i> is updated each time a <i>Reference</i> is added or deleted to the <i>Node</i> the <i>Property</i> belongs to. <i>Attribute</i> value changes except for the <i>Data Type Attribute</i> do not cause the <i>NodeVersion</i> to change. <i>Clients</i> may read the <i>NodeVersion Property</i> or subscribe to it to determine when the structure of a <i>Node</i> has changed.<br>Although the relationship of a <i>VariableType</i> to its <i>DataType</i> is not modelled using <i>References</i> , changes to the <i>Data Type Attribute</i> of a <i>VariableType</i> lead to an update of the <i>NodeVersion Property</i> .                                 |

The *VariableType NodeClass* inherits the base *Attributes* from the *Base NodeClass* defined in 5.2. The *VariableType NodeClass* also defines a set of *Attributes* that describe the default or initial value of its instance *Variables*. The *Value Attribute* represents the default value. The *DataType*, *ValueRank* and *ArrayDimensions Attributes* provide the capability to describe simple and complex values. The *IsAbstract Attribute* defines if the type can be directly instantiated.

The *VariableType NodeClass* uses *HasProperty References* to define the *Properties* and *HasComponent References* to define *DataVariables*. Whether they are instantiated depends on the *ModellingRules* defined in 6.4.4.

The *Property NodeVersion* indicates the version of the *VariableType*. There are no additional *Properties* defined for *VariableTypes* in this document. Additional parts of the IEC 62541 series may define additional *Properties* for *VariableTypes*. IEC 62541-8 defines a set of *Properties* that can be used for *VariableTypes*.

*HasSubtype References* are used to subtype *VariableTypes*. *VariableType* subtypes inherit the general semantics from the parent type. The general rules for subtyping are defined in Clause 6. It is not required to provide the *HasSubtype Reference* for the supertype, but it is required that the subtype provides the inverse *Reference* to its supertype.

*GeneratesEvent References* identify that *Variables* of the *VariableType* may be the source of an *Event* of the specified *EventType* or one of its subtypes. *Servers* should make *GeneratesEvent References* bidirectional *References*. However, it is allowed to be unidirectional when the *Server* is not able to expose the inverse direction pointing from the *EventType* to each *VariableType* supporting the *EventType*.

*GeneratesEvent References* are optional, i.e. *Variables* may generate *Events* of an *EventType* that is not exposed by its *VariableType*.

*VariableTypes* may use any additional *References* to define relationships to other *Nodes*. No restrictions are placed on the types of *References* used or on the *NodeClasses* of the *Nodes* that may be referenced. However, restrictions may be defined by the *ReferenceType* excluding its use for *VariableTypes*. Standard *ReferenceTypes* are described in Clause 7.

All *Nodes* referenced with *hierarchical References* shall have unique *BrowseNames* in the context of the *VariableType* (see 4.5).

### 5.6.6 Client-side creation of Variables of an VariableType

*Variables* are always based on a *VariableType*, i.e. they have a *HasTypeDefinition Reference* pointing to its *VariableType*.

*Clients* can create *Variables* using the *AddNodes Service* defined in IEC 62541-4. The *Service* requires specifying the *TypeDefinitionNode* of the *Variable*. A *Variable* created by the *AddNodes Service* contains all components defined by its *VariableType* dependent on the *ModellingRules* specified for the components. However, the *Server* may add additional components and *References* to the *Variable* and its components that are not defined by the *VariableType*. This behaviour is *Server* dependent. The *VariableType* only specifies the minimum set of components that shall exist for each *Variable* of a *VariableType*.

## 5.7 Method NodeClass

*Methods* define callable functions. *Methods* are invoked using the *Call Service* defined in IEC 62541-4. Method invocations are not represented in the *AddressSpace*. Method invocations always run to completion and always return responses when complete. *Methods* are defined using the *Method NodeClass*, specified in Table 10.

Table 10 – Method NodeClass

| Name                       | Use  | Data Type  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------------------|------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Attributes</b>          |      |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Base NodeClass Attributes  | M    | --         | Inherited from the <i>Base NodeClass</i> . See 5.2.                                                                                                                                                                                                                                                                                                                                                                                                      |
| Executable                 | M    | Boolean    | The <i>Executable Attribute</i> indicates if the <i>Method</i> is currently executable ("False" means not executable, "True" means executable). The <i>Executable Attribute</i> does not take any user access rights into account, i.e. although the <i>Method</i> is executable this may be restricted to a certain user / user group.                                                                                                                  |
| UserExecutable             | M    | Boolean    | The <i>UserExecutable Attribute</i> indicates if the <i>Method</i> is currently executable taking user access rights into account ("False" means not executable, "True" means executable).                                                                                                                                                                                                                                                               |
| <b>References</b>          |      |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| HasProperty                | 0..* |            | <i>HasProperty References</i> identify the <i>Properties</i> for the <i>Method</i> .                                                                                                                                                                                                                                                                                                                                                                     |
| HasModellingRule           | 0..1 |            | <i>Methods</i> can point to at most one <i>ModellingRule Object</i> using a <i>HasModellingRule Reference</i> (see 6.4.4 for details on <i>ModellingRules</i> ).                                                                                                                                                                                                                                                                                         |
| GeneratesEvent             | 0..* |            | <i>GeneratesEvent References</i> identify the type of <i>Events</i> that will be generated whenever the <i>Method</i> is called.                                                                                                                                                                                                                                                                                                                         |
| AlwaysGeneratesEvent       | 0..* |            | <i>AlwaysGeneratesEvent References</i> identify the type of <i>Events</i> that shall be generated whenever the <i>Method</i> is called.                                                                                                                                                                                                                                                                                                                  |
| <other References>         | 0..* |            | <i>Methods</i> may contain other <i>References</i> .                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Standard Properties</b> |      |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| NodeVersion                | O    | String     | The <i>NodeVersion Property</i> is used to indicate the version of a <i>Node</i> . The <i>NodeVersion Property</i> is updated each time a <i>Reference</i> is added or deleted to the <i>Node</i> the <i>Property</i> belongs to. <i>Attribute</i> value changes do not cause the <i>NodeVersion</i> to change. <i>Clients</i> may read the <i>NodeVersion Property</i> or subscribe to it to determine when the structure of a <i>Node</i> has changed. |
| InputArguments             | O    | Argument[] | The <i>InputArguments Property</i> is used to specify the arguments that shall be used by a client when calling the <i>Method</i> .                                                                                                                                                                                                                                                                                                                      |
| OutputArguments            | O    | Argument[] | The <i>OutputArguments Property</i> specifies the result returned from the <i>Method</i> call.                                                                                                                                                                                                                                                                                                                                                           |

The *Method NodeClass* inherits the base *Attributes* from the *Base NodeClass* defined in 5.2. The *Method NodeClass* defines no additional *Attributes*.

The *Executable Attribute* indicates whether the *Method* is executable, not taking user access rights into account. If the OPC UA Server cannot get the *Executable* information from the underlying system, it should state that it is executable. If a *Method* is called then the *Server* should transfer this request and return the corresponding *StatusCode* even if such a request is rejected. *StatusCodes* are defined in IEC 62541-4.

The *UserExecutable Attribute* indicates whether the *Method* is executable, taking user access rights into account. If the OPC UA Server cannot get any user rights related information from the underlying system, it should use the same value as used in the *Executable Attribute*. The *UserExecutable Attribute* can be set to "False", even if the *Executable Attribute* is set to "True", but it shall be set to "False" if the *Executable Attribute* is set to "False".

*Properties* may be defined for *Methods* using *HasProperty References*. The *Properties InputArguments* and *OutputArguments* specify the input arguments and output arguments of the *Method*. Both contain an array of the *DataType Argument* as specified in 8.6. An empty array or a *Property* that is not provided indicates that there are no input arguments or output arguments for the *Method*. The *Property NodeVersion* indicates the version of the *Method*. There are no additional *Properties* defined for *Methods* in this document. Additional parts of the IEC 62541 series may define additional *Properties* for *Methods*.

To specify its *ModellingRule*, a *Method* can use at most one *HasModellingRule Reference* pointing to a *ModellingRule Object*. *ModellingRules* are defined in 6.4.4.

*GeneratesEvent References* identify that *Methods* may generate an *Event* of the specified *EventType* or one of its subtypes for every call of the *Method*. A *Server* may generate one *Event* for each referenced *EventType* when a *Method* is successfully called.

*AlwaysGeneratesEvent References* identify that *Methods* will generate an *Event* of the specified *EventType* or one of its subtypes for every call of the *Method*. A *Server* shall always generate one *Event* for each referenced *EventType* when a *Method* is successfully called.

*Servers* should make *GeneratesEvent References* bidirectional *References*. However, it is allowed to be unidirectional when the *Server* is not able to expose the inverse direction pointing from the *EventType* to each *Method* generating the *EventType*.

*GeneratesEvent References* are optional, i.e. the call of a *Method* may produce *Events* of an *EventType* that is not referenced with a *GeneratesEvent Reference* by the *Method*.

*Methods* may use additional *References* to define relationships to other *Nodes*. No restrictions are placed on the types of *References* used or on the *NodeClasses* of the *Nodes* that may be referenced. However, restrictions may be defined by the *ReferenceType* excluding its use for *Methods*. Standard *ReferenceTypes* are described in Clause 7.

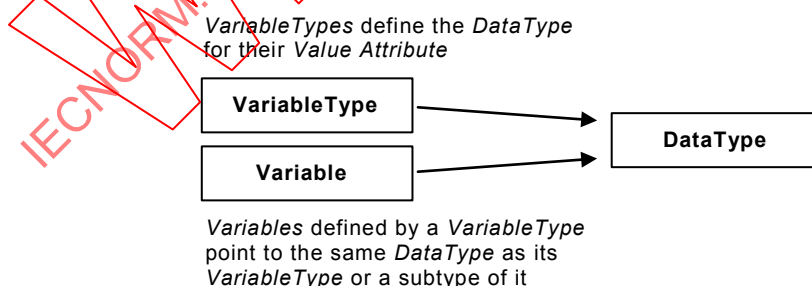
A *Method* shall always be the *TargetNode* of at least one *HasComponent Reference*. The *SourceNode* of these *HasComponent References* shall be an *Object* or an *ObjectType*. If a *Method* is called then the *NodeId* of one of those *Nodes* shall be put into the Call Service defined in IEC 62541-4 as parameter to detect the context of the *Method* operation.

If the *Method* is used as *InstanceDeclaration* (see 4.5) all *Nodes* referenced with *hierarchical References* in forward direction shall have unique *BrowseNames* in the context of this *Method*.

## 5.8 DataTypes

### 5.8.1 DataType Model

The *DataType Model* is used to define simple and structured data types. Data types are used to describe the structure of the *Value Attribute* of *Variables* and their *VariableTypes*. Therefore each *Variable* and *VariableType* is pointing with its *DataType Attribute* to a *Node* of the *DataType NodeClass* as shown in Figure 9.



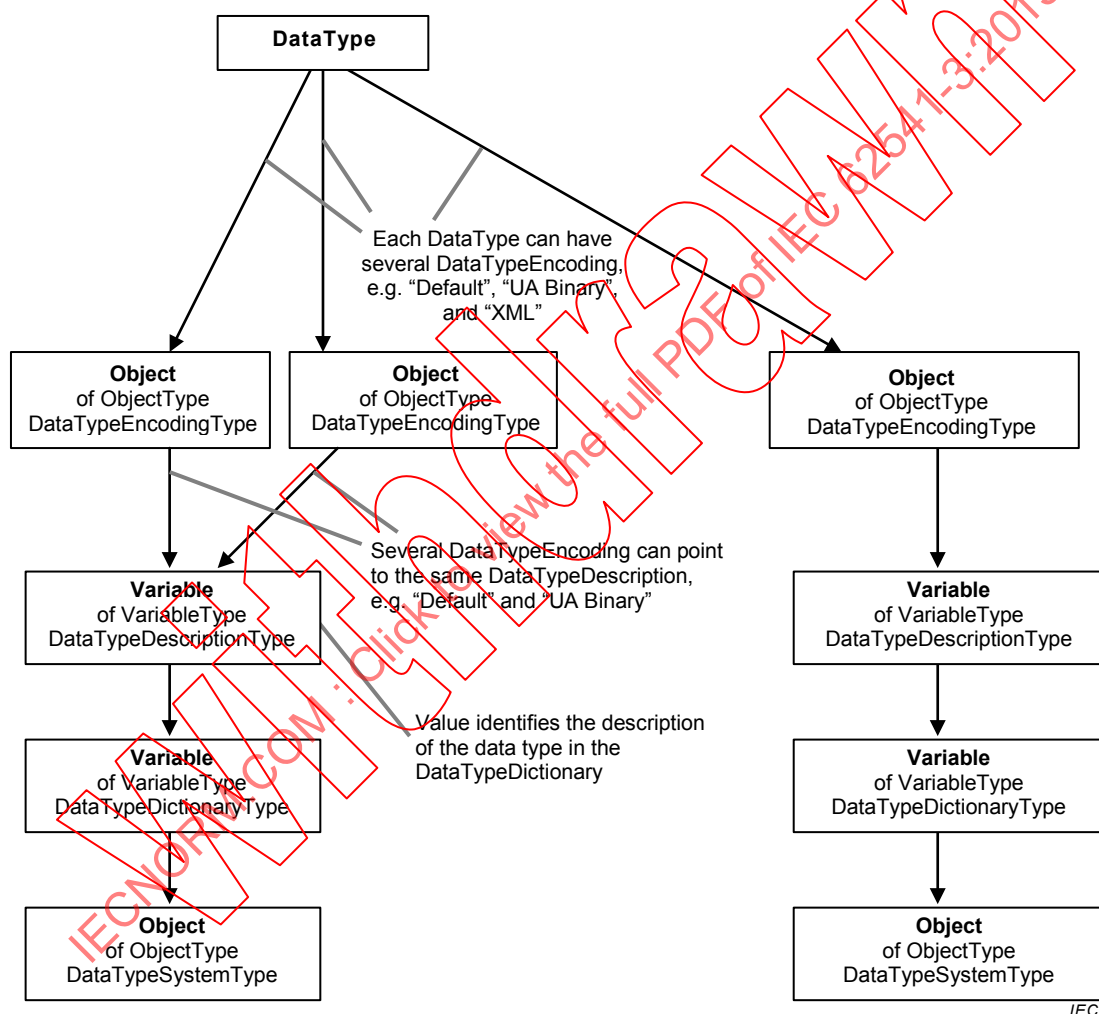
IEC

**Figure 9 – Variables, VariableTypes and their DataTypes**

In many cases, the *NodeId* of the *DataType Node* – the *DataTypeId* – will be well-known to *Clients* and *Servers*. Clause 8 defines *DataTypes* and IEC 62541-6 defines their *DataTypeIds*. In addition, other organizations may define *DataTypes* that are well-known in the industry. Well-known *DataTypeIds* provide for commonality across OPC UA *Servers* and allow *Clients* to interpret values without having to read the type description from the *Server*. Therefore, *Servers* may use well-known *DataTypeIds* without representing the corresponding *DataType Nodes* in their *AddressSpaces*.

In other cases, *DataTypes* and their corresponding *DataTypeIds* may be vendor-defined. *Servers* should attempt to expose the *DataType Nodes* and the information about the structure of those *DataTypes* for *Clients* to read, although this information might not always be available to the *Server*.

Figure 10 illustrates the *Nodes* used in the *AddressSpace* to describe the structure of a *DataType*. The *DataType* points to an *Object* of type *DataTypeEncodingType*. Each *DataType* can have several *DataTypeEncoding*, for example “Default”, “UA Binary” and “XML” encoding. Services in IEC 62541-4 allow *Clients* to request an encoding or choosing the “Default” encoding. Each *DataTypeEncoding* is used by exactly one *DataType*, that is, it is not permitted for two *DataTypes* to point to the same *DataTypeEncoding*. The *DataTypeEncoding Object* points to exactly one *Variable* of type *DataTypeDescriptionType*. The *DataTypeDescription Variable* belongs to a *DataTypeDictionary Variable*.



**Figure 10 – DataType Model**

Since the *NodeId* of the *DataTypeEncoding* will be used in some Mappings to identify the *DataType* and its encoding as defined in IEC 62541-6, those *NodeIds* may also be well-known for well-known *DataTypeIds*.

The *DataTypeDictionary* describes a set of *DataTypes* in sufficient detail to allow *Clients* to parse/interpret *Variable Values* that they receive and to construct *Values* that they send. The *DataTypeDictionary* is represented as a *Variable* of type *DataTypeDictionaryType* in the *AddressSpace*, the description about the *DataTypes* is contained in its *Value Attribute*. All containing *DataTypes* exposed in the *AddressSpace* are represented as *Variables* of type

*DataTypeDescriptionType*. The *Value* of one of these Variables identifies the description of a *DataType* in the *Value Attribute* of the *DataTypeDictionary*.

The *DataType* of a *DataTypeDictionary Variable* is always a *ByteString*. The format and conventions for defining *DataTypes* in this *ByteString* are defined by *DataTypeSystems*. *DataTypeSystems* are identified by *NodeIds*. They are represented in the *AddressSpace* as *Objects* of the *ObjectType DataTypeSystemType*. Each *Variable* representing a *DataTypeDictionary* references a *DataTypeSystem Object* to identify their *DataTypeSystem*.

A client shall recognise the *DataTypeSystem* to parse any of the type description information. OPC UA *Clients* that do not recognise a *DataTypeSystem* will not be able to interpret its type descriptions, and consequently, the values described by them. In these cases, *Clients* interpret these values as opaque *ByteStrings*.

OPC Binary and W3C XML Schema are examples of *DataTypeSystems*. The OPC Binary *DataTypeSystem* is defined in Annex C. OPC Binary uses XML to describe binary data values. W3C XML Schema is specified in XML Schema Part 1 and XML Schema Part 2

### 5.8.2 Encoding Rules for different kinds of DataTypes

Different kinds of *DataTypes* are handled differently regarding their encoding and according to whether this encoding is represented in the *AddressSpace*.

*Built-in DataTypes* are a fixed set of *DataTypes* (see IEC 62541-6 for a complete list of *Built-in DataTypes*). They have no encodings visible in the *AddressSpace* since the encoding should be known to all OPC UA products. Examples of *Built-in DataTypes* are *Int32* (see 8.26) and *Double* (see 8.12).

*Simple DataTypes* are subtypes of the *Built-in DataTypes*. They are handled on the wire like the *Built-in DataType*, i.e. they cannot be distinguished on the wire from their *Built-in* supertypes. Since they are handled like *Built-in DataTypes* regarding the encoding they cannot have encodings defined in the *AddressSpace*. *Clients* can read the *DataType Attribute* of a *Variable* or *VariableType* to identify the *Simple DataType* of the *Value Attribute*. An example of a *Simple DataType* is *Duration*. It is handled on the wire as a *Double* but the Client can read the *DataType Attribute* and thus interpret the value as defined by *Duration* (see 8.13).

*Structured DataTypes* are *DataTypes* that represent structured data and are not defined as *Built-in DataTypes*. *Structured DataTypes* inherit directly or indirectly from the *DataType Structure* defined in 8.33. *Structured DataTypes* may have several encodings and the encodings are exposed in the *AddressSpace*. How the encoding of *Structured DataTypes* is handled on the wire is defined in IEC 62541-6. The encoding of the *Structured DataType* is transmitted with each value, thus *Clients* are aware of the *DataType* without reading the *DataType Attribute*. The encoding has to be transmitted so the Client is able to interpret the data. An example of a *Structured DataType* is *Argument* (see 8.6).

*Enumeration DataTypes* are *DataTypes* that represent discrete sets of named values. Enumerations are always encoded as *Int32* on the wire as defined in IEC 62541-6. *Enumeration DataTypes* inherit directly or indirectly from the *DataType Enumeration* defined in 8.14. Enumerations have no encodings exposed in the *AddressSpace*. To expose the human-readable representation of an enumerated value the *DataType Node* may have the *EnumStrings Property* that contains an array of *LocalizedText*. The Integer representation of the enumeration value points to a position within that array. *EnumValues Property* can be used instead of the *EnumStrings* to support integer representation of enumerations that are not zero-based or have gaps. It contains an array of a *Structured DataType* containing the integer representation as well as the human-readable representation. An example of an enumeration *DataType* containing a sparse list of Integers is *NodeClass* which is defined in 8.30.

In addition to the *DataTypes* described above, abstract *DataTypes* are also supported, which do not have any encodings and cannot be exchanged on the wire. *Variables* and *VariableTypes* use abstract *DataTypes* to indicate that their *Value* may be any one of the subtypes of the abstract *DataType*. An example of an abstract *DataType* is Integer which is defined in 8.24.

### 5.8.3 DataType NodeClass

The *DataType NodeClass* describes the syntax of a *Variable Value*. The *DataTypes* may be simple or complex, depending on the *DataTypeSystem*. *DataTypes* are defined using the *DataType NodeClass*, as specified in Table 11.

**Table 11 – DataType NodeClass**

| Name                       | Use  | Data Type       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------------------|------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Attributes</b>          |      |                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Base NodeClass Attributes  | M    | --              | Inherited from the <i>Base NodeClass</i> . See 5.2.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| IsAbstract                 | M    | Boolean         | A boolean <i>Attribute</i> with the following values:<br>TRUE it is an abstract <i>DataType</i> .<br>FALSE it is not an abstract <i>DataType</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>References</b>          |      |                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| HasProperty                | 0..* |                 | <i>HasProperty References</i> identify the <i>Properties</i> for the <i>DataType</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| HasSubtype                 | 0..* |                 | <i>HasSubtype References</i> may be used to span a data type hierarchy.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| HasEncoding                | 0..* |                 | <i>HasEncoding References</i> identify the encodings of the <i>DataType</i> represented as <i>Objects</i> of type <i>DataTypeEncodingType</i> . Only concrete <i>Structured DataTypes</i> may use <i>HasEncoding References</i> . Abstract, Built-in, Enumeration, and Simple <i>DataTypes</i> are not allowed to be the <i>SourceNode</i> of a <i>HasEncoding Reference</i> . Each concrete <i>Structured DataType</i> shall point to at least one <i>DataTypeEncoding Object</i> with the <i>BrowseName</i> "Default Binary" or "Default XML" having the <i>NamespaceIndex</i> 0. The <i>BrowseName</i> of the <i>DataTypeEncoding Objects</i> shall be unique in the context of a <i>DataType</i> , i.e. a <i>DataType</i> shall not point to two <i>DataTypeEncodings</i> having the same <i>BrowseName</i> . |
| <b>Standard Properties</b> |      |                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| NodeVersion                | O    | String          | The <i>NodeVersion Property</i> is used to indicate the version of a <i>Node</i> . The <i>NodeVersion Property</i> is updated each time a <i>Reference</i> is added or deleted to the <i>Node</i> the <i>Property</i> belongs to. <i>Attribute</i> value changes do not cause the <i>NodeVersion</i> to change. Clients may read the <i>NodeVersion Property</i> or subscribe to it to determine when the structure of a <i>Node</i> has changed.                                                                                                                                                                                                                                                                                                                                                                 |
| EnumStrings                | O    | LocalizedText[] | The <i>EnumStrings Property</i> only applies for <i>Enumeration DataTypes</i> . It shall not be applied for other <i>DataTypes</i> . If the <i>EnumValues Property</i> is provided, the <i>EnumStrings Property</i> shall not be provided. Each entry of the array of <i>LocalizedText</i> in this <i>Property</i> represents the human-readable representation of an enumerated value. The Integer representation of the enumeration value points to a position of the array.                                                                                                                                                                                                                                                                                                                                    |
| EnumValues                 | O    | EnumValueType[] | The <i>EnumValues Property</i> only applies for <i>Enumeration DataTypes</i> . It shall not be applied for other <i>DataTypes</i> . If the <i>EnumStrings Property</i> is provided, the <i>EnumValues Property</i> shall not be provided. Using the <i>EnumValues Property</i> it is possible to represent Enumerations with integers that are not zero-based or have gaps (e.g. 1, 2, 4, 8, 16). Each entry of the array of <i>EnumValueType</i> in this <i>Property</i> represents one enumeration value with its integer notation, human-readable representation and help information.                                                                                                                                                                                                                         |

The *DataType NodeClass* inherits the base *Attributes* from the *Base NodeClass* defined in 5.2. The *IsAbstract Attribute* specifies if the *DataType* is abstract or not. Abstract *DataTypes* can be used in the *AddressSpace*, i.e. *Variables* and *VariableTypes* can point with their *DataType Attribute* to an abstract *DataType*. However, concrete values can never be of an abstract *DataType* and shall always be of a concrete subtype of the abstract *DataType*.

*HasProperty References* are used to identify the *Properties* of a *DataType*. The *Property NodeVersion* is used to indicate the version of the *DataType*. This Version is not affected by the *DataTypeVersion Property* of *DataTypeDictionaries* and *DataTypeDescriptions*. The *Property EnumStrings* contains human-readable representations of enumeration values and is only applied to *Enumeration DataTypes*. Instead of the *EnumStrings Property* an *Enumeration DataType* can also use the *EnumValues Property* to represent *Enumerations* with integer values that are not zero-based or containing gaps. There are no additional *Properties* defined for *DataTypes* in this standard. Additional parts of the IEC 62541 series may define additional *Properties* for *DataTypes*.

*HasSubtype References* may be used to expose a data type hierarchy in the *AddressSpace*. This hierarchy shall reflect the hierarchy specified in the *DataTypeDictionary*. The semantic of subtyping depends on the *DataTypeSystem*. Servers need not provide *HasSubtype References*, even if their *DataTypes* span a type hierarchy. Clients should not make any assumptions about any other semantic with that information than provided by the *DataTypeDictionary*. For example, it might not be possible to cast a value of one data type to its base data type. Some restrictions apply for subtyping enumeration *DataTypes* as defined in 8.14.

*HasEncoding References* point from the *DataType* to its *DataTypeEncodings*. Following such a *Reference*, the client can browse to the *DataTypeDictionary* describing the structure of the *DataType* for the used encoding. Each concrete *Structured DataType* can point to many *DataTypeEncodings*, but each *DataTypeEncoding* shall belong to one *DataType*, that is, it is not permitted for two *DataType Nodes* to point to the same *DataTypeEncoding Object* using *HasEncoding References*.

An abstract *DataType* is not the *SourceNode* of a *HasEncoding Reference*. The *DataTypeEncoding* of an abstract *DataType* is provided by its concrete subtypes.

*DataType Nodes* shall not be the *SourceNode* of other types of *References*. However, they may be the *TargetNode* of other *References*.

#### 5.8.4 **DataTypeDictionary, DataTypeDescription, DataTypeEncoding and DataTypeSystem**

A *DataTypeDictionary* is an entity that contains a set of type descriptions, such as an XML schema. *DataTypeDictionaries* are defined as *Variables* of the *VariableType DataTypeDictionaryType*.

A *DataTypeSystem* specifies the format and conventions for defining *DataTypes* in *DataTypeDictionaries*. *DataTypeSystems* are defined as *Objects* of the *ObjectType DataTypeSystemType*.

The *ReferenceType* used to relate *Objects* of the *ObjectType DataTypeSystemType* to *Variables* of the *VariableType DataTypeDictionaryType* is the *HasComponent ReferenceType*. Thus, the *Variable* is always the *TargetNode* of a *HasComponent Reference*; this is a requirement for *Variables*. However, for *DataTypeDictionaries* the *Server* shall always provide the inverse *Reference*, since it is necessary to know the *DataTypeSystem* when processing the *DataTypeDictionary*.

An example of a *DataTypeDictionary* is an XML document containing an XML schema. In this case, the *DataTypeSystem* is the W3C XML Schema and the top level element declarations in the schema document are the data type descriptions. Each of these descriptions is defined in different versions of an XML schema using the same XML target namespace. This target namespace is used as the namespace component of the *DataTypeId* in the *Server's AddressSpace*. Since the same target namespace can be used in other XML schemas, *Clients* shall be aware that two *DataTypeIds* with the same namespace are not necessarily defined in the same *DataTypeDictionary*.

Changes may be a result of a change to a type description, but it is more likely that dictionary changes are a result of the addition or deletion of type descriptions. This includes changes made while the *Server* is offline so that the new version is available when the *Server* restarts. *Clients* may subscribe to the *DataTypeVersion Property* to determine if the *DataTypeDictionary* has changed since it was last read.

The *Server* may, but is not required to, make the *DataTypeDictionary* contents available to *Clients* through the *Value Attribute*. *Clients* should assume that *DataTypeDictionary* contents are relatively large and that they will encounter performance problems if they automatically read the *DataTypeDictionary* contents each time they encounter an instance of a specific *DataType*. The client should use the *DataTypeVersion Property* to determine whether the locally cached copy is still valid. If the client detects a change to the *DataTypeVersion*, then it shall re-read the *DataTypeDictionary*. This implies that the *DataTypeVersion* shall be updated by a *Server* even after restart since *Clients* may persistently store the locally cached copy.

The *Value Attribute* of the *DataTypeDictionary* containing the type descriptions is a *ByteString* whose formatting is defined by the *DataTypeSystem*. For the “XML Schema” *DataTypeSystem*, the *ByteString* contains a valid XML Schema document. For the “OPC Binary” *DataTypeSystem*, the *ByteString* contains a string that is a valid XML document. The *Server* shall ensure that any change to the contents of the *ByteString* is matched with a corresponding change to the *DataTypeVersion Property*. In other words, the client may safely use a cached copy of the *DataTypeDictionary*, as long as the *DataTypeVersion* remains the same.

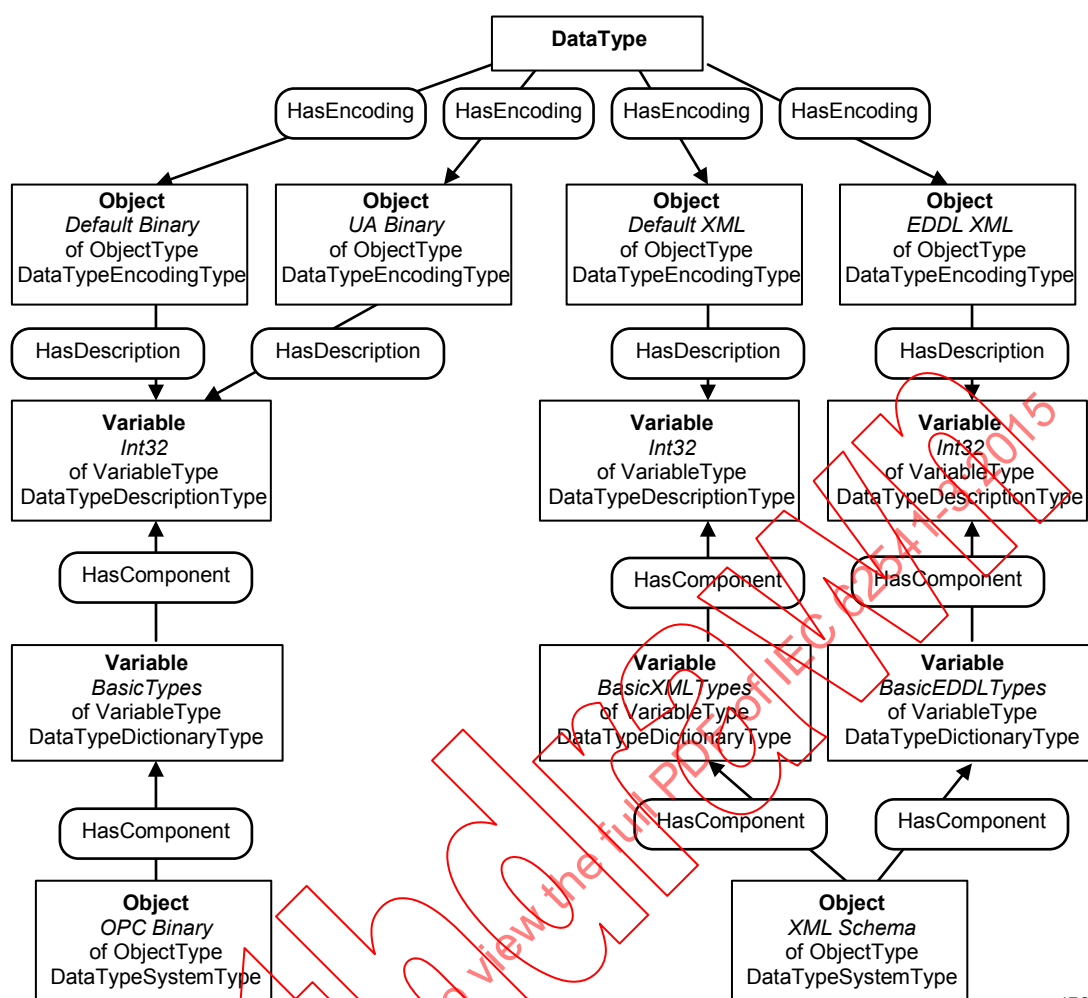
*DataTypeDictionaries* are complex *Variables* which expose their *DataTypeDescriptions* as *Variables* using *HasComponent References*. A *DataTypeDescription* provides the information necessary to find the formal description of a *DataType* within the *DataTypeDictionary*. The *Value* of a *DataTypeDescription* depends on the *DataTypeSystem* of the *DataTypeDictionary*. When using “OPC Binary” dictionaries the *Value* shall be the name of the *TypeDescription*. When using “XML Schema” dictionaries the *Value* shall be an Xpath expression (see XPATH) which points to an XML element in the schema document.

Like *DataTypeDictionaries* each *DataTypeDescription* provides the *Property DataTypeVersion* indicating whether the type description of the *DataType* has changed. Changes to the *DataTypeVersion* may impact the operation of *Subscriptions*. If the *DataTypeVersion* changes for a *Variable* that is being monitored for a *Subscription* and that uses this *DataTypeDescription*, then the next data change *Notification* sent for the *Variable* will contain a status that indicates the change in the *DataTypeDescription*.

*DataTypeEncoding Objects* of the *DataTypes* reference their *DataTypeDescriptions* of the *DataTypeDictionaries* using *HasDescription References*. However, *Servers* are not required to provide the inverse *References* that relate the *DataTypeDescriptions* back to the *DataTypeEncoding Objects*. If a *DataType Node* is exposed in the *AddressSpace*, it shall provide its *DataTypeEncodings* and if a *DataTypeDictionary* is exposed then it should expose all of its *DataTypeDescriptions*. Both of these *References* shall be bi-directional.

The *VariableTypes* *DataTypeDictionaryType* and *DataTypeDescriptionType* and the *ObjectTypes* *DataTypeSystemType* and *DataTypeEncodingType* are formally defined in IEC 62541-5.

Figure 11 provides an example how *DataTypes* are modelled in the *AddressSpace*.



IEC

**Figure 11 – Example of DataType Modelling**

In some scenarios an OPC UA Server may have resource limitations which make it impractical to expose large *DataTypeDictionaries*. In these scenarios the Server may be able to provide access to descriptions for individual DataTypes even if the entire dictionary cannot be read. For this reason, this standard defines a *Property* for the *DataTypeDescription* called *DictionaryFragment* (see 5.6.2). This *Property* is a *ByteString* that contains a subset of the *DataTypeDictionary* which describes the format of the *DataType* associated with the *DataTypeDescription*. Thus, the Server splits the large *DataTypeDictionary* into several small parts and *Clients* can access without affecting the overall system performance.

However, Servers should provide the whole *DataTypeDictionary* at once if this is possible. It is typically more efficient to read the whole *DataTypeDictionary* at once instead of reading individual parts.

## 5.9 Summary of Attributes of the NodeClasses

Table 12 summarises all *Attributes* defined in this document and points out which *NodeClasses* use them either in an optional (O) or mandatory (M) way.

Table 12 – Overview of Attributes

| Attribute               | Variable | Variable Type | Object | Object Type | Reference Type | Data Type | Method | View |
|-------------------------|----------|---------------|--------|-------------|----------------|-----------|--------|------|
| AccessLevel             | M        |               |        |             |                |           |        |      |
| ArrayDimensions         | O        | O             |        |             |                |           |        |      |
| BrowseName              | M        | M             | M      | M           | M              | M         | M      | M    |
| ContainsNoLoops         |          |               |        |             |                |           |        | M    |
| Data Type               | M        | M             |        |             |                |           |        |      |
| Description             | O        | O             | O      | O           | O              | O         | O      | O    |
| DisplayName             | M        | M             | M      | M           | M              | M         | M      | M    |
| EventNotifier           |          |               | M      |             |                |           |        | M    |
| Executable              |          |               |        |             |                |           | M      |      |
| Historizing             | M        |               |        |             |                |           |        |      |
| InverseName             |          |               |        |             | O              |           |        |      |
| IsAbstract              |          | M             |        | M           | M              | M         |        |      |
| MinimumSamplingInterval | O        |               |        |             |                |           |        |      |
| NodeClass               | M        | M             | M      | M           | M              | M         | M      | M    |
| NodeId                  | M        | M             | M      | M           | M              | M         | M      | M    |
| Symmetric               |          |               |        |             | M              |           |        |      |
| UserAccessLevel         | M        |               |        |             |                |           |        |      |
| UserExecutable          |          |               |        |             |                |           | M      |      |
| UserWriteMask           | O        | O             | O      | O           | O              | O         | O      | O    |
| Value                   | M        | O             |        |             |                |           |        |      |
| ValueRank               | M        | M             |        |             |                |           |        |      |
| WriteMask               | O        | O             | O      | O           | O              | O         | O      | O    |

## 6 Type Model for ObjectTypes and VariableTypes

### 6.1 Overview

In the remainder of 6 the type model of *ObjectTypes* and *VariableTypes* is defined regarding subtyping and instantiation.

### 6.2 Definitions

#### 6.2.1 InstanceDeclaration

An *InstanceDeclaration* is an *Object*, *Variable* or *Method* that references a *ModellingRule* with a *HasModellingRule* *Reference* and is the *TargetNode* of a *hierarchical Reference* from a *TypeDefinitionNode* or another *InstanceDeclaration*.

#### 6.2.2 Instances without ModellingRules

If no *ModellingRule* exists then the *Node* is neither considered for instantiation of a type nor for subtyping.

If a *Node* referenced by a *TypeDefinitionNode* does not reference a *ModellingRule* it indicates that this *Node* only belongs to the *TypeDefinitionNode* and not to the instances. For example, an *ObjectType Node* may contain a *Property* that describes scenarios where the type could be used. This *Property* would not be considered when creating instances of the type. This is also true for subtyping, that is, subtypes of the type definition would not inherit the referenced *Node*.

### 6.2.3 InstanceDeclarationHierarchy

The *InstanceDeclarationHierarchy* of a *TypeDefinitionNode* contains the *TypeDefinitionNode* and all *InstanceDeclarations* that are directly or indirectly referenced from the *TypeDefinitionNode* using *hierarchical References* in the forward direction.

### 6.2.4 Similar Node of InstanceDeclaration

A similar *Node* of an *InstanceDeclaration* is a *Node* that has the same *BrowseName* and *NodeClass* as the *InstanceDeclaration* and in cases of *Variables* and *Objects* the same *TypeDefinitionNode* or a subtype of it.

### 6.2.5 BrowsePath

All targets of forward *hierarchical References* from a *TypeDefinitionNode* shall have a *BrowseName* that is unique within the *TypeDefinitionNode*. The same restriction applies to the targets of *hierarchical References* in forward direction from any *InstanceDeclaration*. This means that any *InstanceDeclaration* within the *InstanceDeclarationHierarchy* can be uniquely identified by a sequence of *BrowseNames*. This sequence of *BrowseNames* is called a *BrowsePath*.

### 6.2.6 Attribute Handling of InstanceDeclarations

Some restrictions exist regarding the *Attributes* of *InstanceDeclarations* when the *InstanceDeclaration* is overridden or instantiated. The *BrowseName* and the *NodeClass* shall never change and always be the same as the original *InstanceDeclaration*.

In addition, the rules defined in 6.2.7 apply for *InstanceDeclarations* of the *NodeClass Variable*.

### 6.2.7 Attribute Handling of Variable and VariableTypes

Some restrictions exist regarding the *Attributes* of a *VariableType* or a *Variable* used as an *InstanceDeclaration* with regard to the data type of the *Value Attribute*.

When a *Variable* used as *InstanceDeclaration* or a *VariableType* is overridden or instantiated the following rules apply:

- a) The *DataType Attribute* can only be changed to a new *DataType* if the new *DataType* is a subtype of the *DataType* originally used.
- b) The *ValueRank Attribute* may only be further restricted
  - 1) 'Any' may be set to any other value;
  - 2) 'ScalarOrOneDimension' may be set to 'Scalar' or 'OneDimension';
  - 3) 'OneOrMoreDimensions' may be set to a concrete number of dimensions (value > 0).
  - 4) All other values of this *Attribute* shall not be changed.
- c) The *ArrayDimensions Attribute* may be added if it was not provided or when modifying the value of an entry in the array from 0 to a different value. All other values in the array shall remain the same.

### 6.2.8 NodeIds of InstanceDeclarations

*InstanceDeclarations* are identified by their *BrowsePath*. Different *Servers* might use different *NodeIds* for the *InstanceDeclarations* of common *TypeDefinitionNodes*, unless the definition of the *TypeDefinitionNode* already defines a *NodeId* for the *InstanceDeclaration*. All *TypeDefinitionNodes* defined in IEC 62541-5 already define the *NodeIds* for their *InstanceDeclarations* and therefore shall be used in all *Servers*.

## 6.3 Subtyping of ObjectTypes and VariableTypes

### 6.3.1 Overview

The *HasSubtype ReferenceType* defines subtypes of types. Subtyping can only occur between *Nodes* of the same *NodeClass*. Rules for subtyping *ReferenceTypes* are described in 5.3.3.3. There is no common definition for subtyping *DataTypes*, as described in 5.8.3. The remainder of 6.3 specify subtyping rules for single inheritance on *ObjectTypes* and *VariableTypes*.

### 6.3.2 Attributes

Subtypes inherit the parent type's *Attribute* values, except for the *NodeId*. Inherited *Attribute* values may be overridden by the subtype, the *BrowseName* and *DisplayName* values should be overridden. Special rules apply for some *Attributes* of *VariableTypes* as defined in 6.2.7. Optional *Attributes*, not provided by the parent type, may be added to the subtype.

### 6.3.3 InstanceDeclarations

#### 6.3.3.1 Overview

Subtypes inherit the fully-inherited parent type's *InstanceDeclarations*.

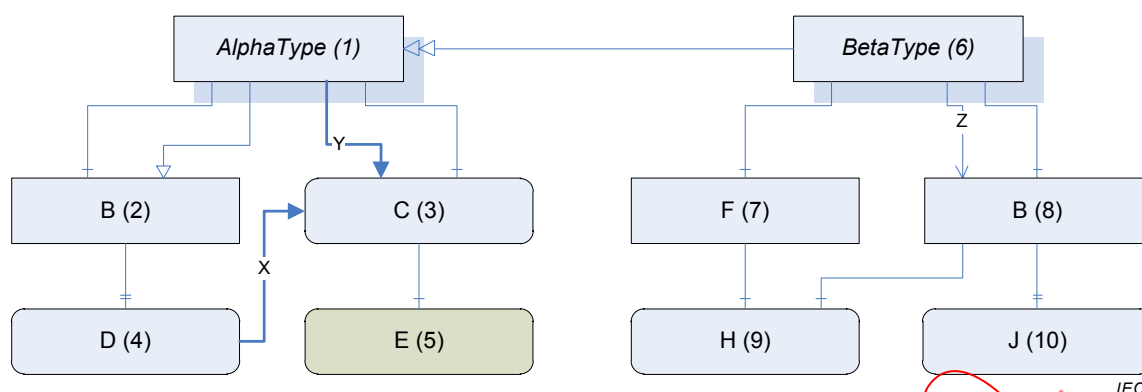
As long as those *InstanceDeclarations* are not overridden they are not referenced by the subtype. *InstanceDeclarations* can be overridden by adding *References*, changing *References* to reference different *Nodes*, changing *References* to be subtypes of the original *ReferenceType*, changing values of the *Attributes* or adding optional *Attributes*. In order to get the full information about a subtype, the inherited *InstanceDeclarations* have to be collected from all types that can be found by recursively following the inverse *HasSubtype References* from the subtype. This collection of *InstanceDeclarations* is called the fully-inherited *InstanceDeclarationHierarchy* of a subtype.

The remainder of 6.3.3 define how to construct the fully-inherited *InstanceDeclarationHierarchy* and how *InstanceDeclarations* can be overridden.

#### 6.3.3.2 Fully-inherited InstanceDeclarationHierarchy

An instance of a *TypeDefinitionNode* is described by the fully-inherited *InstanceDeclarationHierarchy* of the *TypeDefinitionNode*. The fully-inherited *InstanceDeclarationHierarchy* can be created by starting with the *InstanceDeclarationHierarchy* of the *TypeDefinitionNode* and merging the fully-inherited *InstanceDeclarationHierarchy* of its parent type.

The process of merging *InstanceDeclarationHierarchies* is straightforward and can be illustrated with the example shown in Figure 12 which specifies a *TypeDefinitionNode* "BetaType" which is a subtype of "AlphaType". The name in each box is the *BrowseName* and the number is the *NodeId*.



**Figure 12 – Subtyping TypeDefinitionNodes**

An *InstanceDeclarationHierarchy* can be fully described as a table of *Nodes* identified by their *BrowsePaths* with a corresponding table of *References*. The *InstanceDeclarationHierarchy* for “BetaType” is described in Table 13 where the top half of the table is the table of *Nodes* and the bottom half is the table of *References* (the *HasModellingRule* references have been omitted from the table for the sake of clarity; all *Nodes* except for 1, 6, and 5 have *ModellingRules*). All *InstanceDeclarations* of the *InstanceDeclarationHierarchy* and all *Nodes* referenced with a non-hierarchical *Reference* from such an *InstanceDeclaration* are added to the table. *Hierarchical References* to *Nodes* without a *ModellingRule* are not considered.

**Table 13 – The InstanceDeclarationHierarchy for BetaType**

| BrowsePath | NodeId |
|------------|--------|
| /          | 6      |
| /F         | 7      |
| /B         | 8      |
| /F/H       | 9      |
| /B/J       | 10     |
| /B/H       | 9      |

| Source Path | ReferenceType     | Target Path | TargetNodeId         |
|-------------|-------------------|-------------|----------------------|
| /           | HasComponent      | /F          | -                    |
| /           | HasComponent      | /B          | -                    |
| /           | Z                 | /B          | -                    |
| /           | HasTypeDefinition | -           | BetaType             |
| /F          | HasComponent      | /F/H        | -                    |
| /F          | HasTypeDefinition | -           | BaseObjectType       |
| /B          | HasProperty       | /B/J        | -                    |
| /B          | HasTypeDefinition | -           | BaseObjectType       |
| /F/H        | HasTypeDefinition | -           | PropertyType         |
| /B/J        | HasTypeDefinition | -           | PropertyType         |
| /B          | HasComponent      | /B/H        | -                    |
| /B/H        | HasTypeDefinition | -           | BaseDataVariableType |

Multiple *BrowsePaths* to the same *Node* shall be treated as separate *Nodes*. An *Instance* may provide different *Nodes* for each *BrowsePath*.

The fully-inherited *InstanceDeclarationHierarchy* for “BetaType” can now be constructed by merging the *InstanceDeclarationHierarchy* for “AlphaType”. The result is shown in Table 14 where the entries added from “AlphaType” are shaded with grey.

**Table 14 – The Fully-Inherited InstanceDeclarationHierarchy for BetaType**

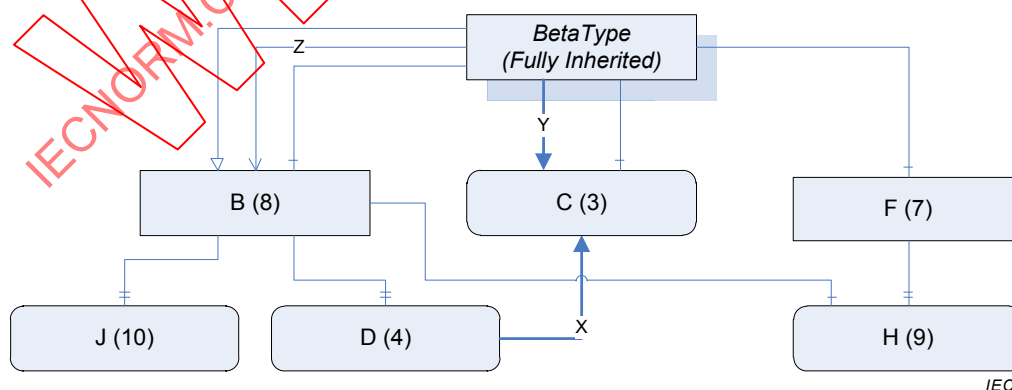
| BrowsePath | NodeId |
|------------|--------|
| /          | 6      |
| /F         | 7      |
| /B         | 8      |
| /F/H       | 9      |
| /B/J       | 10     |
| /B/H       | 9      |
| /B/D       | 4      |
| /C         | 3      |

| Source Path | ReferenceType     | Target Path | TargetNodeId         |
|-------------|-------------------|-------------|----------------------|
| /           | HasComponent      | /F          | -                    |
| /           | HasComponent      | /B          | -                    |
| /           | Z                 | /B          | -                    |
| /           | HasTypeDefinition | -           | BetaType             |
| /F          | HasComponent      | /F/H        | -                    |
| /F          | HasTypeDefinition | -           | BaseObjectType       |
| /B          | HasProperty       | /B/J        | -                    |
| /B          | HasTypeDefinition | -           | BaseObjectType       |
| /F/H        | HasTypeDefinition | -           | PropertyType         |
| /B/J        | HasTypeDefinition | -           | PropertyType         |
| /B          | HasComponent      | /B/H        | -                    |
| /B/H        | HasTypeDefinition | -           | BaseDataVariableType |
| /           | HasNotifier       | /B          | -                    |
| /B          | HasProperty       | /B/D        | -                    |
| /           | HasComponent      | /C          | -                    |
| /           | Y                 | /C          | -                    |
| /C          | HasTypeDefinition | -           | BaseDataVariableType |
| /B/D        | HasTypeDefinition | -           | PropertyType         |
| /B/D        | X                 | /C          | -                    |

The *BrowsePath* “/B” already exists in the table so it does not need to be added. However, the *HasNotifier* reference from “/” to “/B” does not exist and was added.

The *Nodes* and *References* defined in Table 14 can be used to create the fully-inherited *InstanceDeclarationHierarchy* shown in Figure 13. The fully-inherited *InstanceDeclarationHierarchy* contains all necessary information about a *TypeDefinitionNode* regarding its complex structure without needing any additional information from its supertypes.

**Figure 13 – The Fully-Inherited InstanceDeclarationHierarchy for BetaType**

### 6.3.3.3 Overriding InstanceDeclarations

A subtype overrides an *InstanceDeclaration* by specifying an *InstanceDeclaration* with the same *BrowsePath*. An overridden *InstanceDeclaration* shall have the same *NodeClass* and

**BrowseName.** The *TypeDefinitionNode* of the overridden *InstanceDeclaration* shall be the same or a subtype of the *TypeDefinitionNode* specified in the supertype.

When overriding an *InstanceDeclaration* it is necessary to provide *hierarchical References* that link the new *Node* back to the subtype (the *References* are used to determine the *BrowsePath* of the *Node*).

It is only possible to override *InstanceDeclarations* that are directly referenced from the *TypeDefinitionNode*. If an indirect referenced *InstanceDeclaration*, such as “J” in Figure 13, has to be overridden, then the directly referenced *InstanceDeclarations* that includes “J”, in that case “B”, have to be overridden first and then “J” can be overridden in a second step.

A *Reference* is replaced if it goes between two overridden *Nodes* and has the same *ReferenceType* as a *Reference* defined in the supertype. The *Reference* specified in the subtype may be a subtype of the *ReferenceType* used in the parent type.

Any *non-hierarchical References* specified for the overridden *InstanceDeclaration* are treated as new *References* unless the *ReferenceType* only allows a single *Reference* per *SourceNode*. If this situation exists the subtype can change the target of the *Reference* but the new target shall have the same *NodeClass* and for *Objects* and *Variables* also the same type or a subtype of the type specified in the parent.

The overriding *Node* may specify new values for the *Node Attributes* other than the *NodeClass* or *BrowseName*, however, the restrictions on *Attributes* specified in 6.2.6 apply. Any *Attribute* provided by the overridden *InstanceDeclaration* has to be provided by the overriding *InstanceDeclaration*, additional optional *Attributes* may be added.

The *ModellingRule* of the overriding *InstanceDeclaration* may be changed as defined in 6.4.4.3.

Each overriding *InstanceDeclaration* needs its own *HasModellingRule* and *HasTypeDefinition References*, even if they have not been changed.

A subtype should not override a *Node* unless it needs to change it.

The semantics of certain *TypeDefinitionNodes* and *ReferenceTypes* may impose additional restrictions with regard to overriding *Nodes*.

## 6.4 Instances of ObjectTypes and VariableTypes

### 6.4.1 Overview

Any *Instance* of a *TypeDefinitionNode* will be the root of a hierarchy which mirrors the *InstanceDeclarationHierarchy* for the *TypeDefinitionNode*. Each *Node* in the hierarchy of the *Instance* will have a *BrowsePath* which may be the same as the *BrowsePath* for one of the *InstanceDeclarations* in the hierarchy of the *TypeDefinitionNode*. The *InstanceDeclaration* with the same *BrowsePath* is called the *InstanceDeclaration* for the *Node*. If a *Node* has an *InstanceDeclaration* then it shall have the same *BrowseName* and *NodeClass* as the *InstanceDeclaration* and, in cases of *Variables* and *Objects*, the same *TypeDefinitionNode* or a subtype of it.

Instances may reference several *Nodes* with the same *BrowsePath*. *Clients* that need to distinguish between the *Nodes* based on the *InstanceDeclarationHierarchy* and the *Nodes* that are not based on the *InstanceDeclarationHierarchy* can accomplish this using the *TranslateBrowsePathsToNodeIds* service defined in IEC 62541-4.

### 6.4.2 Creating an Instance

Instances inherit the initial values for the *Attributes* that they have in common with the *TypeDefinitionNode* from which they are instantiated, with the exceptions of the *NodeClass* and *NodeId*.

When a *Server* creates an instance of a *TypeDefinitionNode* it shall create the same hierarchy of *Nodes* beneath the new *Object* or *Variable* depending on the *ModellingRule* of each *InstanceDeclaration*. Standard *ModellingRules* are defined in 6.4.4.5. The *Nodes* within the newly created hierarchy may be copies of the *InstanceDeclarations*, the *InstanceDeclaration* itself or another *Node* in the *AddressSpace* that has the same *TypeDefinitionNode* and *BrowseName*. If new copies are created, then the *Attribute* values of the *InstanceDeclarations* are used as the initial values.

Figure 14 provides a simple example of a *TypeDefinitionNode* and an *Instance*. *Nodes* referenced by the *TypeDefinitionNode* without a *ModellingRule* do not appear in the instance. *Instances* may have children with duplicate *BrowseNames*; however, only one of those children will correspond to the *InstanceDeclaration*.

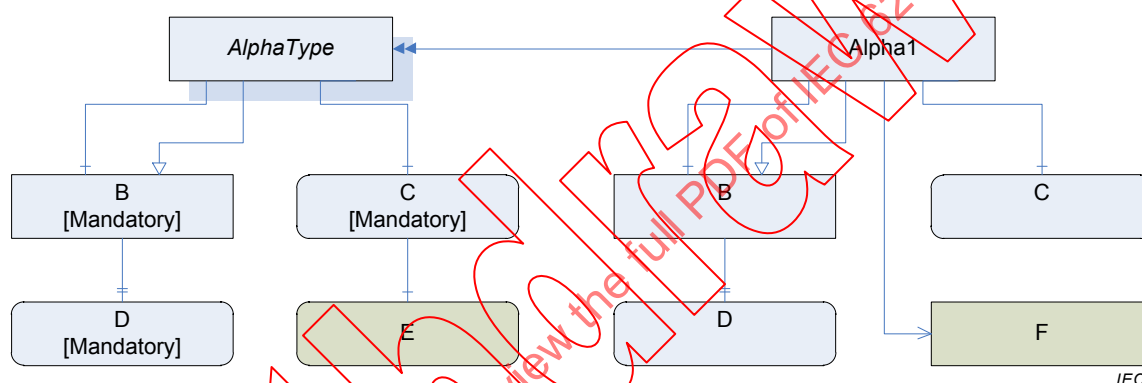


Figure 14 – An Instance and its TypeDefinitionNode

It is up to the *Server* to decide which *InstanceDeclarations* appear in any single instance. In some cases, the *Server* will not define the entire instance and will provide remote references to *Nodes* in another *Server*. The *ModellingRules* described in 6.4.4.5 allow *Servers* to indicate that some *Nodes* are always present; however, the *Client* shall be prepared for the case where the *Node* exists in a different *Server*.

A *Client* can use the information of *TypeDefinitionNodes* to access *Nodes* which are in the hierarchy of the instance. It shall pass the *NodeId* of the instance and the *BrowsePath* of the child *Nodes* based on the *TypeDefinitionNode* to the *TranslateBrowsePathsToNodeIds* service (see IEC 62541-4). This *Service* returns the *NodeId* for each of the child *Nodes*. If a child *Node* exists then the *BrowseName* and *NodeClass* shall match the *InstanceDeclaration*. In the case of *Objects* or *Variables*, also the *TypeDefinitionNode* shall either match or be a subtype of the original *TypeDefinitionNode*.

### 6.4.3 Constraints on an Instance

*Objects* and *Variables* may change their *Attribute* values after being created. Special rules apply for some *Attributes* as defined in 6.2.6.

Additional *References* may be added to the *Nodes*, and *References* may be deleted as long as the *ModellingRules* defined on the *InstanceDeclarations* of the *TypeDefinitionNode* are still fulfilled.

For *Variables* and *Objects* the *HasTypeDefinition Reference* shall always point to the same *TypeDefinitionNode* as the *InstanceDeclaration* or a subtype of it.

If two *InstanceDeclarations* of the fully-inherited *InstanceDeclarationHierarchy* have been connected directly with several *References*, all those *References* shall connect the same *Nodes*. An example is given in Figure 15. The instances A1 and A2 are allowed since B1 references the same *Node* with both *References*, whereas A3 is not allowed since two different *Nodes* are referenced. Note that this restriction only applies for directly connected *Nodes*. For example, A2 references a C1 directly and a different C1 via B1.

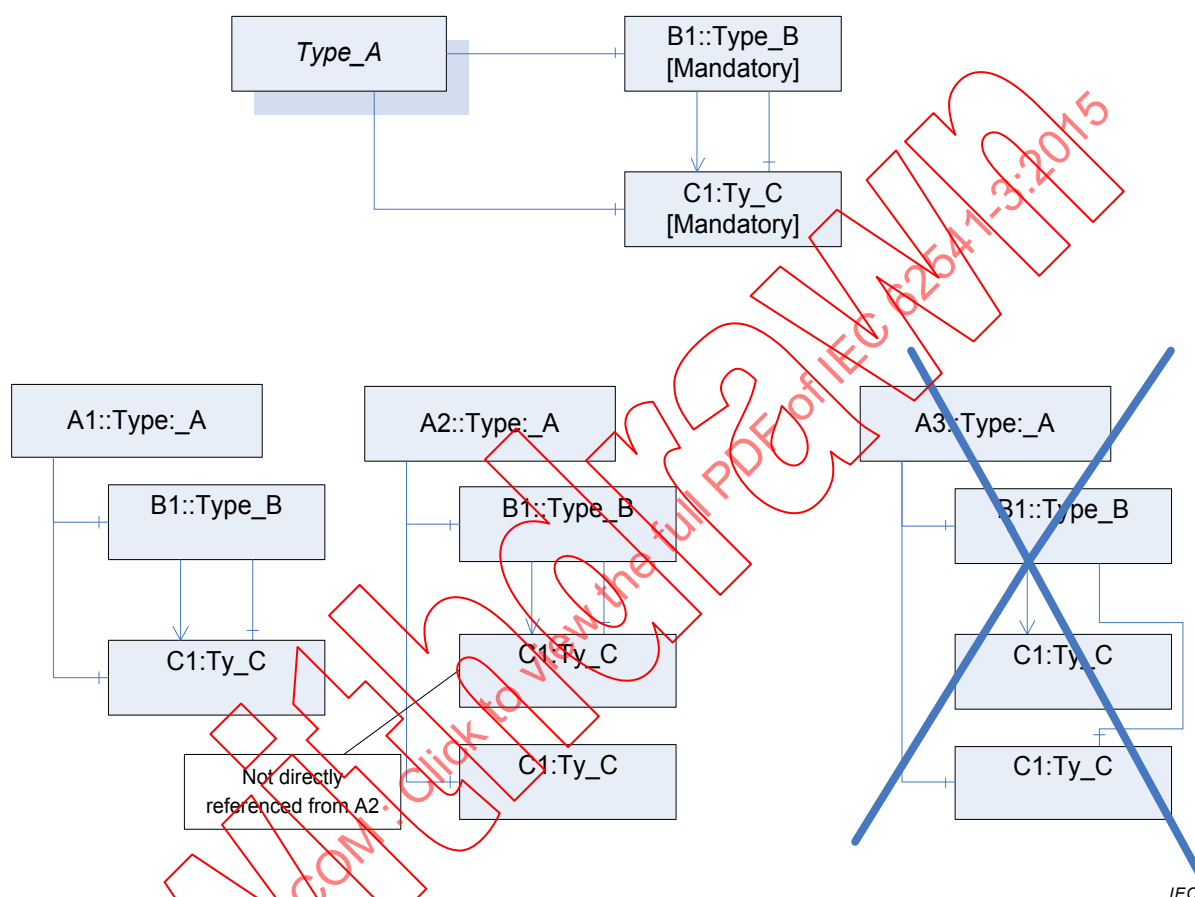


Figure 15 – Example for several References between InstanceDeclarations

#### 6.4.4 ModellingRules

##### 6.4.4.1 General

For a definition of *ModellingRules*, see 6.4.4.5. Other parts of the IEC 62541 series may define additional *ModellingRules*. *ModellingRules* are an extendable concept in OPC UA; therefore vendors may define their own *ModellingRules*.

Note that the *ModellingRules* defined in this standard do not define how to deal with non-hierarchical *References* between *InstanceDeclarations*, i.e. it is *Server-specific* if those *References* exist in an instance hierarchy or not. Other *ModellingRules* may define behaviour for non-hierarchical *References* between *InstanceDeclaration* as well.

*ModellingRules* are represented in the *AddressSpace* as *Objects* of the *ObjectType ModellingRuleType*. There are some *Properties* defining common semantic of *ModellingRules*. This edition of this standard only specifies one *Property* for *ModellingRules*. Future editions may define additional *Properties* for *ModellingRules*. IEC 62541-5 specifies the representation

of the *ModellingRule Objects*, their *Properties* and their type in the *AddressSpace*. The semantic of the *Properties* for *ModellingRules* is defined in 6.4.4.2.

Subclause 6.4.4.4 defines how the *ModellingRule* may be changed when instantiating *InstanceDeclarations* with respect to the *Properties*. Subclause 6.4.4.3 defines how the *ModellingRule* may be changed when overriding *InstanceDeclarations* in subtypes with respect to the *Properties*.

#### 6.4.4.2 Properties describing ModellingRules – NamingRule

*NamingRule* is a mandatory *Property* of a *ModellingRule*. It specifies the purpose of an *InstanceDeclaration*. Each *InstanceDeclaration* references a *ModellingRule* and thus the *NamingRule* is defined per *InstanceDeclaration*.

Three values are allowed for the *NamingRule* of a *ModellingRule*: *Optional*, *Mandatory*, and *Constraint*.

The following semantic is valid for the entire life-time of an instance that is based on a *TypeDefinitionNode* having an *InstanceDeclaration*.

For an instance A1 of a *TypeDefinitionNode* AlphaType with an *InstanceDeclaration* B1 having a *ModellingRule* using the *NamingRule Optional* the following rule applies: For each *BrowsePath* from AlphaType to B1 the instance A1 may or may not have a *similar Node* (see 6.2.4) for B1 with the same *BrowsePath*. If such a *Node* exists then the *TranslateBrowsePathsToNodeIds Service* (see IEC 62541-4) returns this *Node* as the first entry in the list.

For an instance A1 of a *TypeDefinitionNode* AlphaType with an *InstanceDeclaration* B1 having a *ModellingRule* using the *NamingRule Mandatory* the following rule applies: For each *BrowsePath* from AlphaType to B1 the instance A1 shall have a *similar Node* (see 6.2.4) for B1 using the same *BrowsePath* if all *Nodes* of the *BrowsePath* exist. For example, if a *Node* in the *BrowsePath* has a *NamingRule Optional* and is omitted in the instance, then all children of this *Node* would also be omitted, independent of their *ModellingRules*.

If an *InstanceDeclaration* has a *ModellingRule* using the *NamingRule Constraint* it identifies that the *BrowseName* of the *InstanceDeclaration* is of no significance but other semantic is defined with the *ModellingRule*. The *TranslateBrowsePathsToNodeIds Service* (see IEC 62541-4) can typically not be used to access instances based on those *InstanceDeclarations*.

#### 6.4.4.3 Subtyping Rules for Properties of ModellingRules

It is allowed that subtypes override *ModellingRules* on their *InstanceDeclarations*. As a general rule for subtyping, constraints shall only be tightened, not loosened. Therefore, it is not allowed to specify on the supertype that an instance shall exist with the name (*NamingRule Mandatory*) and on the subtype make this optional (*NamingRule Optional*). Table 15 specifies the allowed changes on the *Properties* when exchanging the *ModellingRules* in the subtype.

**Table 15 – Rule for ModellingRules Properties when Subtyping**

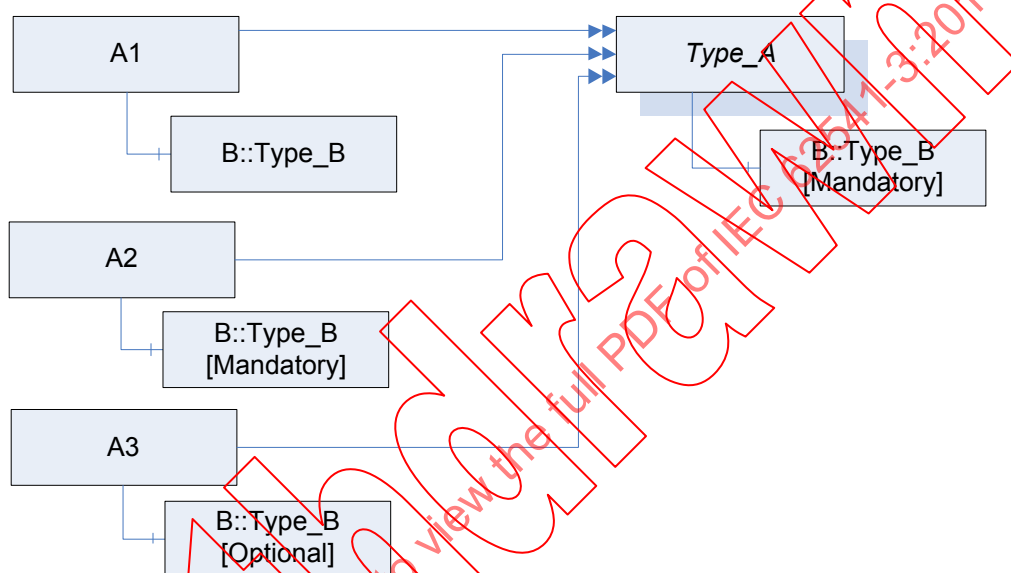
|            | Value on supertype | Value on subtype      |
|------------|--------------------|-----------------------|
| NamingRule | Mandatory          | Mandatory             |
| NamingRule | Optional           | Mandatory or Optional |
| NamingRule | Constraint         | Constraint            |

#### 6.4.4.4 Instantiation Rules for Properties of ModellingRules

There are two different use cases when creating an instance 'A' based on a *TypeDefinitionNode* 'A\_Type'. Either 'A' is used as normal instance or it is used as *InstanceDeclaration* of another *TypeDefinitionNode*.

In the first case, it is not required that newly created or referenced instances based on *InstanceDeclarations* have a *ModellingRule*, however, it is allowed that they have any *ModellingRule* independent of the *ModellingRule* of their *InstanceDeclaration*.

In Figure 16 an example is given. The instances A1, A2, and A3 are all valid instances of Type\_A, although B of A1 has no *ModellingRule* and B of A3 has a different *ModellingRule* than B of Type\_A.

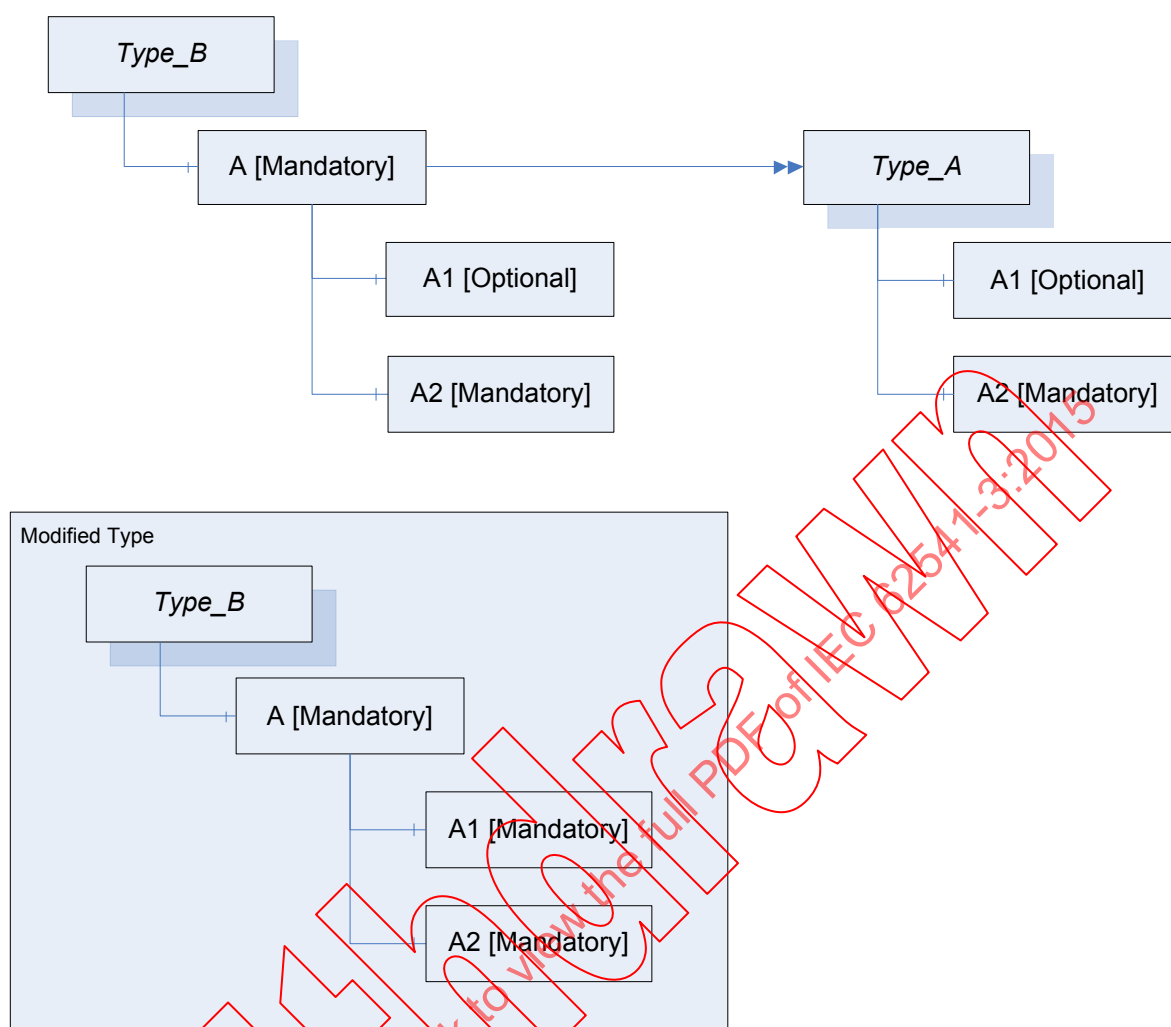


IEC

**Figure 16 – Example on changing instances based on InstanceDeclarations**

In the second case, all instances that are referenced directly or indirectly from 'A' based on *InstanceDeclarations* of 'A\_Type' initially maintain the same *ModellingRule* as their *InstanceDeclarations*. The *ModellingRules* may be updated; the allowed changes to the *ModellingRules* of these *Nodes* are the same as those defined for subtyping in 6.4.4.3.

In Figure 17 an example of such a scenario is given. Type\_B uses an *InstanceDeclaration* based on Type\_A (upper part of the Figure). Later on the *ModellingRule* of the *InstanceDeclaration* A1 is changed (lower part of the Figure). A1 has become the *NamingRule* of *Mandatory* (changed from *Optional*).



IEC

**Figure 17 – Example on changing InstanceDeclarations based on an InstanceDeclaration**

#### 6.4.4.5 Standard ModellingRules

##### 6.4.4.5.1 Titles of Standard ModellingRules

The remainder of 6.4.4.5 defines *ModellingRules*. In Table 16 the *Properties* of those *ModellingRules* are summarized.

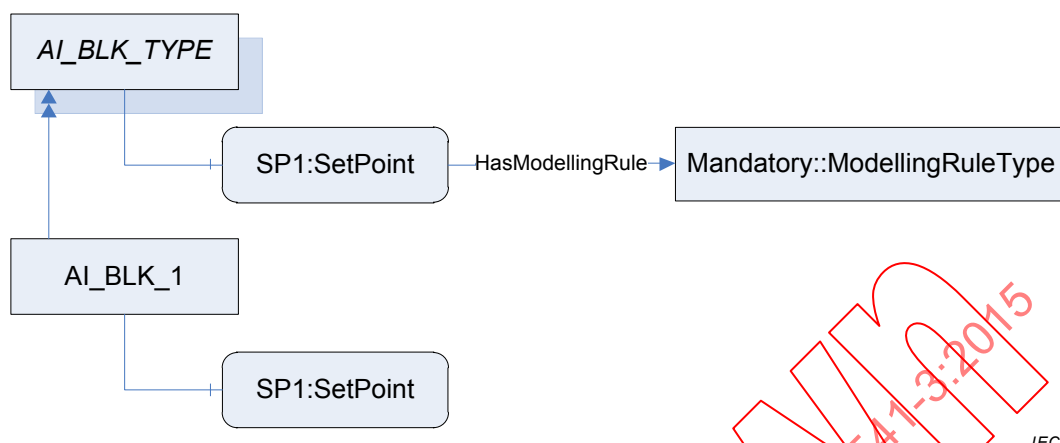
**Table 16 – Properties of ModellingRules**

| Title                | NamingRule |
|----------------------|------------|
| Mandatory            | Mandatory  |
| Optional             | Optional   |
| ExposesItsArray      | Constraint |
| OptionalPlaceholder  | Constraint |
| MandatoryPlaceholder | Constraint |

##### 6.4.4.5.2 Mandatory

An *InstanceDeclaration* marked with the *ModellingRule Mandatory* fulfils exactly the semantic defined for the *NamingRule Mandatory*. That means that for each existing *BrowsePath* on the instance a similar *Node* shall exist, but it is not defined whether a new *Node* is created or an existing *Node* is referenced.

For example, the *TypeDefinitionNode* of a functional block “AI\_BLK\_TYPE” will have a setpoint “SP1”. An instance of this type “AI\_BLK\_1” will have a newly-created setpoint “SP1” as a similar Node to the *InstanceDeclaration* SP1. Figure 18 illustrates the example.



**Figure 18 – Use of the Standard ModellingRule New**

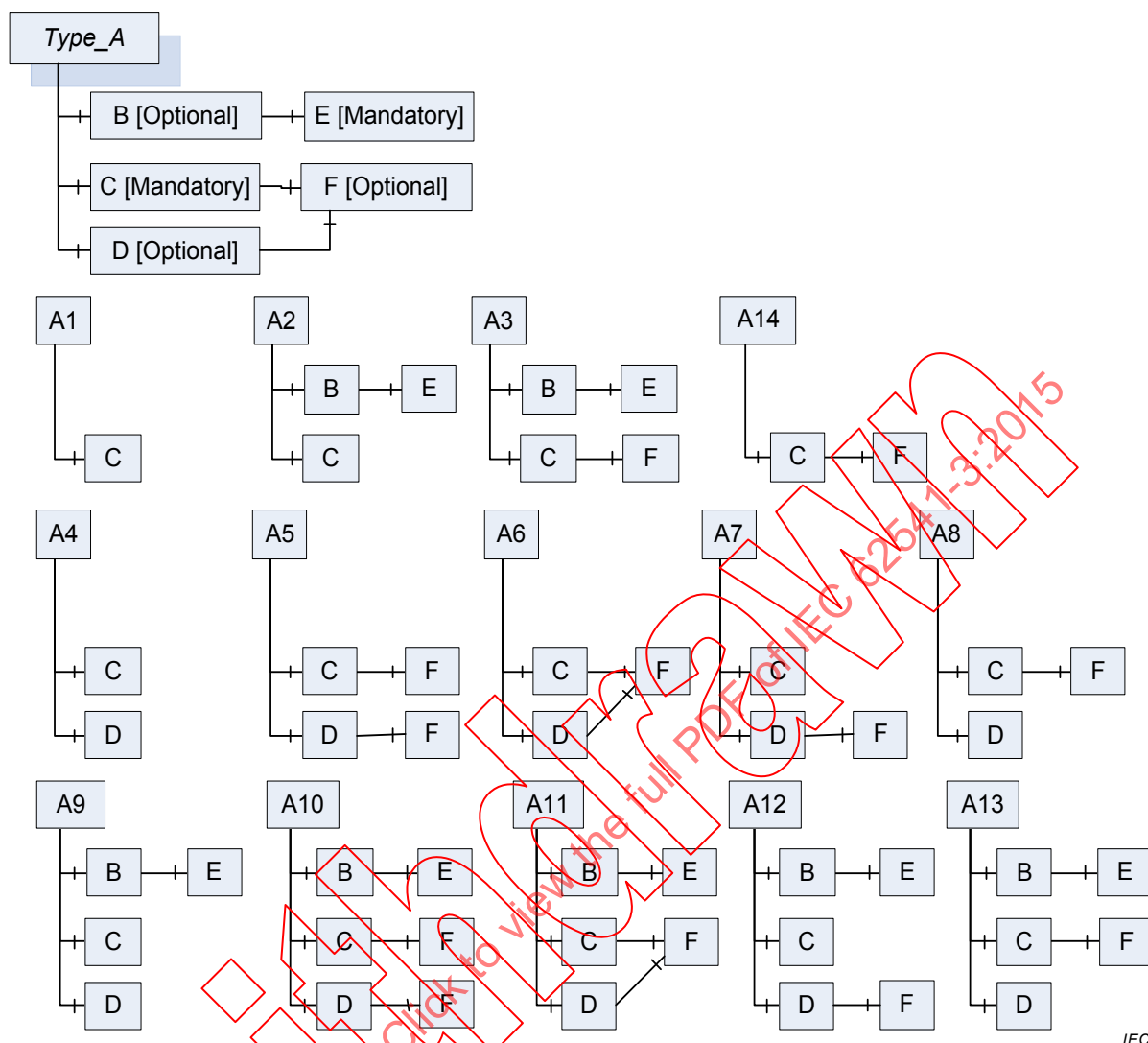
In 6.4.4.5.3 a complex example combining the *Mandatory* and *Optional ModellingRules* is given.

#### 6.4.4.5.3 Optional

An *InstanceDeclaration* marked with the *ModellingRule Optional* fulfils exactly the semantic defined for the *NamingRule Optional*. That means that for each existing *BrowsePath* on the instance a similar *Node* may exist, but it is not defined whether a new *Node* is created or an existing *Node* is referenced.

In Figure 19 an example using the *ModellingRules Optional* and *Mandatory* is shown. The example contains an *ObjectType* *Type\_A* and all valid combinations of instances named A1 to A13. Note that if the optional B is provided, the mandatory E has to be provided as well, otherwise not. F is referenced by C and D. On the instance, this can be the same *Node* or two different *Nodes* with the same *BrowseName* (similar *Node* to *InstanceDeclaration* F). Not considered in the example is if the instances have *ModellingRules* or not. It is assumed that each F is similar to the *InstanceDeclaration* F, etc.

If there would be a non-hierarchical *Reference* between E and F in the *InstanceDeclaration-Hierarchy*, it is not specified if it occurs in the instance hierarchy or not. In the case of A10, there could be a reference from E to one F but not to the other F, or to both or none of them.



IEC

**Figure 19 – Example using the Standard ModellingRules Optional and Mandatory**

#### 6.4.4.5.4 ExposesItsArray

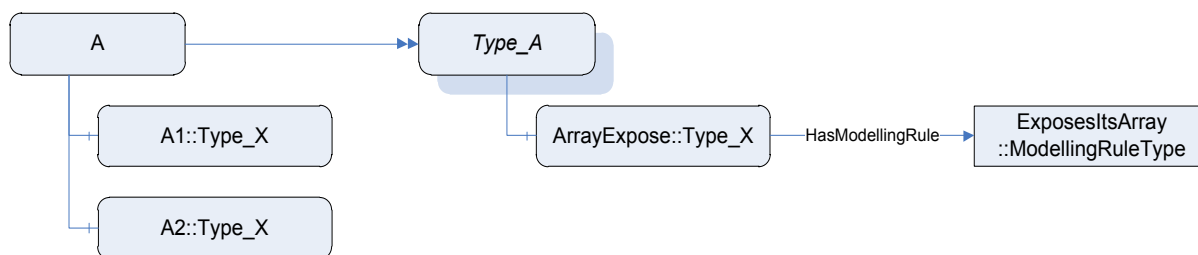
The *ExposesItsArray* *ModellingRule* exposes a special semantic on *VariableTypes* having a single- or multidimensional array as the data type. It indicates that each value of the array will also be exposed as a *Variable* in the *AddressSpace*.

The *ExposesItsArray* *ModellingRule* can only be applied on *InstanceDeclarations* of *NodeClass Variable* that are part of a *VariableType* having a single- or multidimensional array as its data type.

The *Variable* A having this *ModellingRule* shall be referenced by a *hierarchical Reference* in a forward direction from a *VariableType* B. B shall have a *ValueRank* value that is equal to or larger than zero. A should have a data type that reflects at least parts of the data that is managed in the array of B. Each instance of B shall reference one instance of A for each of its array elements. The used *Reference* shall be of the same type as the *hierarchical Reference* that connects B with A or a subtype of it. If there are more than one *hierarchical References* in the forward direction between A and B, then all instances based on B shall be referenced with all those *References*.

Figure 20 gives an example. A is an instance of *Type\_A* having two entries in its value array. Therefore it references two instances of the same type as the *InstanceDeclaration* *ArrayExpose*. The *BrowseNames* of those instances are not defined by the *ModellingRule*. In

general, it is not possible to get a *Variable* representing a specific entry in the array (e.g. the second). *Clients* will typically either get the array or access the *Variables* directly, so there is no need to provide that information.

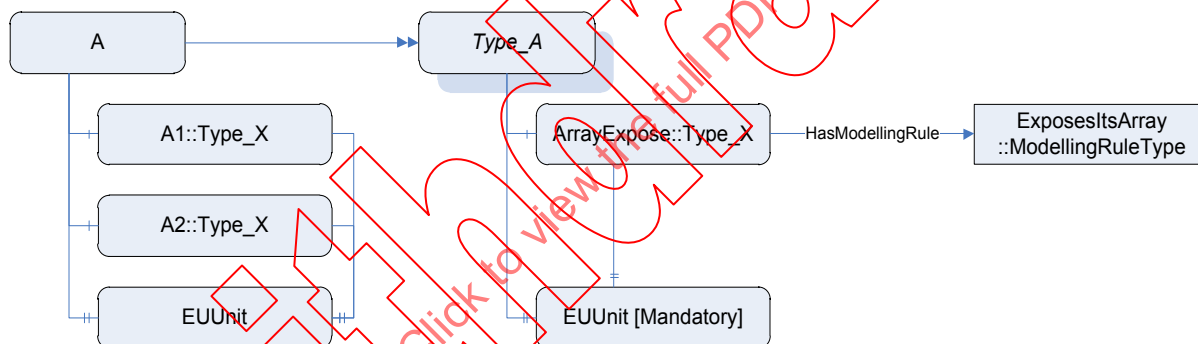


IEC

**Figure 20 – Example on using ExposesItsArray**

It is allowed to reference *A* by other *InstanceDeclarations* as well. Those *References* have to be reflected on each instance based on *A*.

Figure 21 gives an example. The *Property* EUUnit is referenced by *ArrayExpose* and therefore each instance based on *ArrayExpose* references the instance based on the *InstanceDeclaration* EUUnit.

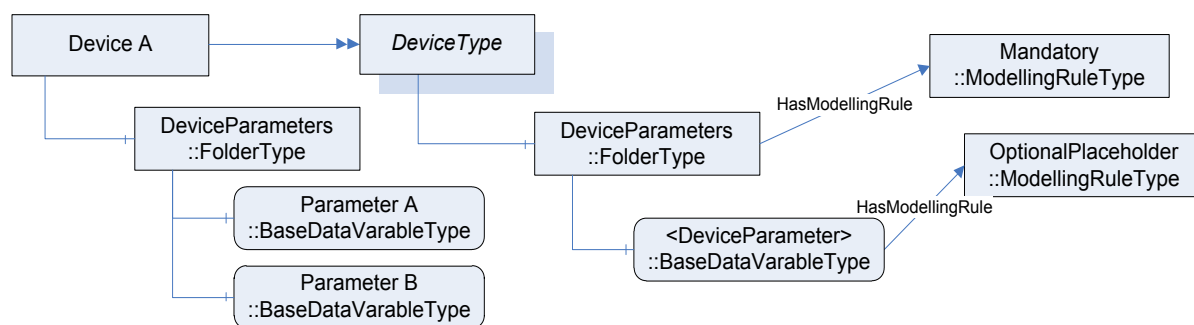


IEC

**Figure 21 – Complex example on using ExposesItsArray**

#### 6.4.4.5.5 OptionalPlaceholder

The intention of the *ModellingRule OptionalPlaceholder* is to expose the information that a complex *TypeDefinition* expects from instances of the *TypeDefinition* to add instances with specific *References* without defining *BrowseNames* for the instances. For example, a *Device* might have a *Folder* for *DeviceParameters*, and the *DeviceParameters* should be connected with a *HasComponent Reference*. However, the names of the *DeviceParameters* are specific to the instances. The example is shown in Figure 22, where an instance *Device A* adds two *DeviceParameters* in the *Folder*.



IEC

**Figure 22 – Example on using OptionalPlaceholder**

The *ModellingRule OptionalPlaceholder* adds no additional constraints on instances of the *TypeDefinition*. It just provides useful information when exposing a *TypeDefinition*. When the *InstanceDeclaration* is complex, i.e. it references other *InstanceDeclarations* with hierarchical *References*, these *InstanceDeclarations* are not further considered for instantiating the *TypeDefinition*.

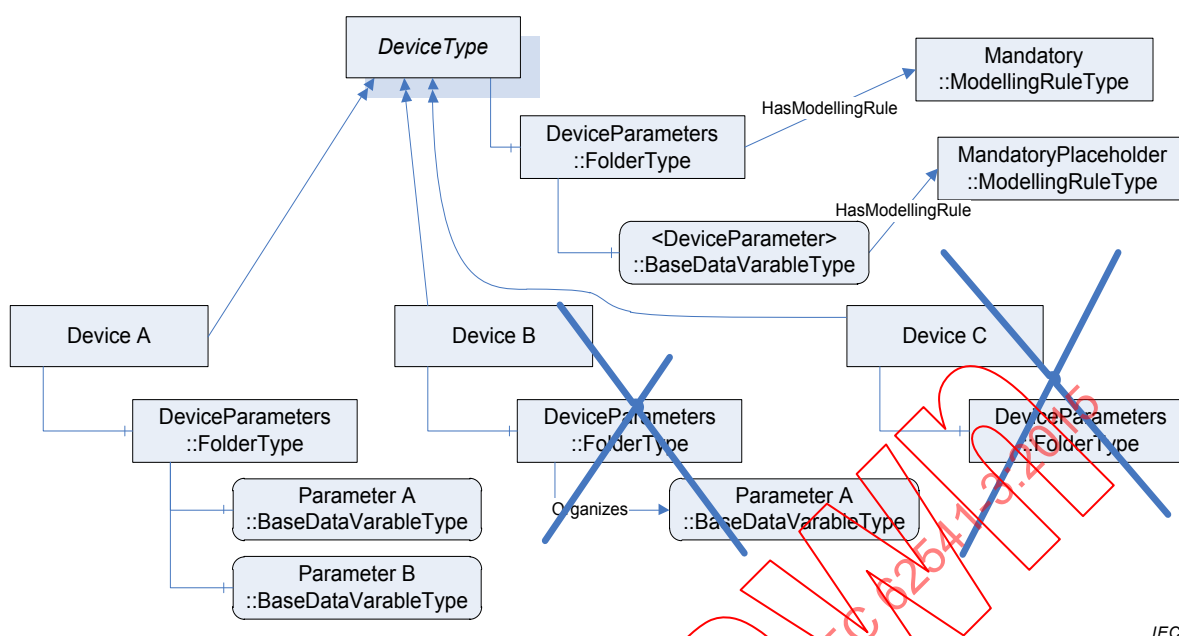
It is recommended that the *BrowseName* and the *DisplayName* of *InstanceDeclarations* having the *OptionalPlaceholder ModellingRule* should be enclosed within angle brackets.

When overriding the *InstanceDeclaration*, the *ModellingRule* shall remain *OptionalPlaceholder*.

#### 6.4.4.5.6 MandatoryPlaceholder

The *ModellingRule MandatoryPlaceholder* has a similar intention as the *ModellingRule OptionalPlaceholder*. It exposes the information that a *TypeDefinition* expects of instances of the *TypeDefinition* to add instances defined by the *InstanceDeclaration*. However, *MandatoryPlaceholder* requires that at least one of those instances shall exist.

For example, when the *DeviceType* requires that at least one *DeviceParameter* shall exist without specifying the *BrowseName* for it, it uses *MandatoryPlaceholder* as shown in Figure 23. Device A is a valid instance as it has the required *DeviceParameter*. Device B is not valid as it uses the wrong *ReferenceType* to reference a *DeviceParameter* (*Organizes* instead of *HasComponent*) and Device C is not valid because it does not provide a *DeviceParameter* at all.



**Figure 23 – Example on using MandatoryPlaceholder**

The *ModellingRule MandatoryPlaceholder* requires that each instance provides at least one instance with the *TypeDefinition* of the *InstanceDeclaration* or a subtype, and is referenced with the same *ReferenceType* or a subtype as the *InstanceDeclaration*. It does not require a specific *BrowseName* and thus cannot be used for the *TranslateBrowsePathsToNodeIds Service* (see IEC 62541-4).

When the *InstanceDeclaration* is complex, i.e. it references other *InstanceDeclarations* with hierarchical *References*, these *InstanceDeclarations* are not further considered for instantiating the *TypeDefinition*.

It is recommended that the *BrowseName* and the *DisplayName* of *InstanceDeclarations* having the *MandatoryPlaceholder ModellingRule* should be enclosed within angle brackets.

When overriding the *InstanceDeclaration*, the *ModellingRule* shall remain *MandatoryPlaceholder*.

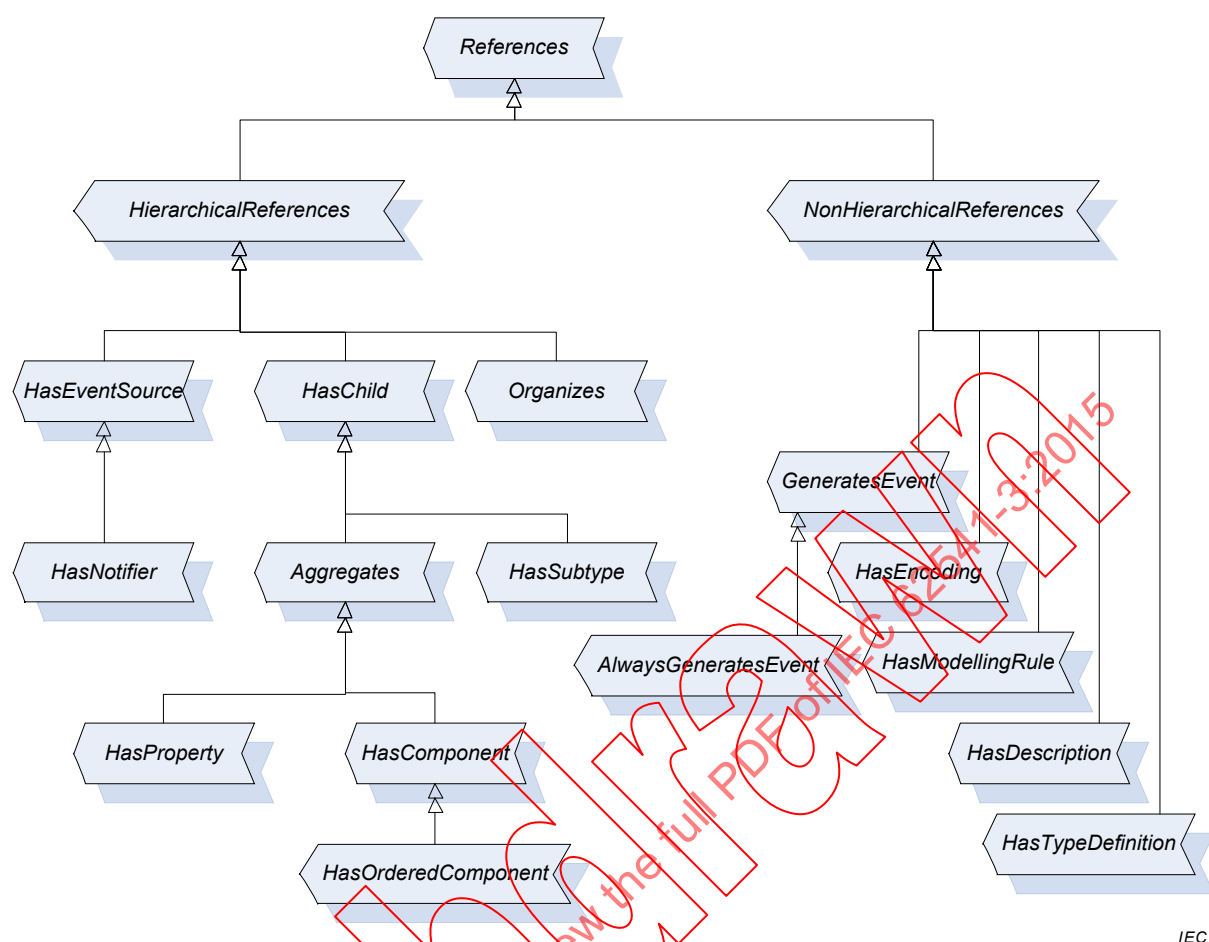
## 6.5 Changing Type Definitions that are already used

There is no behaviour specified regarding subtypes and instances when changing *ObjectTypes* and *VariableTypes*. It is *Server-dependent*, if those changes are reflected on the subtypes and instances of the types. However, all constraints defined for subtypes and instances have to be fulfilled. For example, it is not allowed to add a *Property* using the *ModellingRule Mandatory* on a type if instances of this type exist without the *Property*. In that case, the *Server* either has to add the *Property* to all instances of the type or adding the *Property* on the type has to be rejected.

## 7 Standard ReferenceTypes

### 7.1 General

This standard defines *ReferenceTypes* as an inherent part of the OPC UA Address Space Model. Figure 24 informally describes the hierarchy of these *ReferenceTypes*. Other parts of the IEC 62541 series may specify additional *ReferenceTypes*. The remainder of 7 defines the *ReferenceTypes*. IEC 62541-5 defines their representation in the *AddressSpace*.



IEC

Figure 24 – Standard ReferenceType Hierarchy

## 7.2 References ReferenceType

The *References ReferenceType* is an abstract *ReferenceType*; only subtypes of it can be used.

There is no semantic associated with this *ReferenceType*. This is the base type of all *ReferenceTypes*. All *ReferenceTypes* shall be a subtype of this base *ReferenceType* – either direct or indirect. The main purpose of this *ReferenceType* is allowing simple filter and queries in the corresponding *Services* of IEC 62541-5.

There are no constraints defined for this abstract *ReferenceType*.

## 7.3 HierarchicalReferences ReferenceType

The *HierarchicalReferences ReferenceType* is an abstract *ReferenceType*; only subtypes of it can be used.

The semantic of *HierarchicalReferences* is to denote that *References* of *HierarchicalReferences* span a hierarchy. It means that it may be useful to present *Nodes* related with *References* of this type in a hierarchical-like way. *HierarchicalReferences* does not forbid loops. For example, starting from *Node* “A” and following *HierarchicalReferences* it may be possible to browse to *Node* “A”, again.

It is not permitted to have a *Property* as *SourceNode* of a *Reference* of any subtype of this abstract *ReferenceType*.

It is not allowed that the *SourceNode* and the *TargetNode* of a *Reference* of the *ReferenceType HierarchicalReferences* are the same, that is, it is not allowed to have self-references using *HierarchicalReferences*.

#### 7.4 NonHierarchicalReferences ReferenceType

The *NonHierarchicalReferences ReferenceType* is an abstract *ReferenceType*; only subtypes of it can be used.

The semantic of *NonHierarchicalReferences* is to denote that its subtypes do not span a hierarchy and should not be followed when trying to present a hierarchy. To distinguish hierarchical and non-hierarchical *References*, all concrete *ReferenceTypes* shall inherit from either *hierarchical References* or *non-hierarchical References*, either direct or indirect.

There are no constraints defined for this abstract *ReferenceType*.

#### 7.5 HasChild ReferenceType

The *HasChild ReferenceType* is an abstract *ReferenceType*; only subtypes of it can be used. It is a subtype of *HierarchicalReferences*.

The semantic is to indicate that *References* of this type span a non-looping hierarchy.

Starting from *Node* “A” and only following *References* of the subtypes of the *HasChild ReferenceType* it shall never be possible to return to “A”. But it is allowed that following the *References* there may be more than one path leading to another *Node* “B”.

#### 7.6 Aggregates ReferenceType

The *Aggregates ReferenceType* is an abstract *ReferenceType*; only subtypes of it can be used. It is a subtype of *HasChild*.

The semantic is to indicate a part (the *TargetNode*) belongs to the *SourceNode*. It does not specify the ownership of the *TargetNode*.

There are no constraints defined for this abstract *ReferenceType*.

#### 7.7 HasComponent ReferenceType

The *HasComponent ReferenceType* is a concrete *ReferenceType* that can be used directly. It is a subtype of the *Aggregates ReferenceType*.

The semantic is a part-of relationship. The *TargetNode* of a *Reference* of the *HasComponent ReferenceType* is a part of the *SourceNode*. This *ReferenceType* is used to relate *Objects* or *ObjectTypes* with their containing *Objects*, *DataVariables*, and *Methods* as well as complex *Variables* or *VariableTypes* with their *DataVariables*.

Like all other *ReferenceTypes*, this *ReferenceType* does not specify anything about the ownership of the parts, although it represents a part-of relationship semantic. That is, it is not specified if the *TargetNode* of a *Reference* of the *HasComponent ReferenceType* is deleted when the *SourceNode* is deleted.

The *TargetNode* of this *ReferenceType* shall be a *Variable*, an *Object* or a *Method*.

If the *TargetNode* is a *Variable*, the *SourceNode* shall be an *Object*, an *ObjectType*, a *DataVariable* or a *VariableType*. By using the *HasComponent Reference*, the *Variable* is defined as *DataVariable*.

If the *TargetNode* is an *Object* or a *Method*, the *SourceNode* shall be an *Object* or *ObjectType*.

### 7.8 HasProperty ReferenceType

The *HasProperty ReferenceType* is a concrete *ReferenceType* that can be used directly. It is a subtype of the *Aggregates ReferenceType*.

The semantic is to identify the *Properties* of a *Node*. *Properties* are described in 4.4.2.

The *SourceNode* of this *ReferenceType* can be of any *NodeClass*. The *TargetNode* shall be a *Variable*. By using the *HasProperty Reference*, the *Variable* is defined as *Property*. Since *Properties* shall not have *Properties*, a *Property* shall never be the *SourceNode* of a *HasProperty Reference*.

### 7.9 HasOrderedComponent ReferenceType

The *HasOrderedComponent ReferenceType* is a concrete *ReferenceType* that can be used directly. It is a subtype of the *HasComponent ReferenceType*.

The semantic of the *HasOrderedComponent ReferenceType* – besides the semantic of the *HasComponent ReferenceType* – is that when browsing from a *Node* and following *References* of this type or its subtype all *References* are returned in the Browse Service defined in IEC 62541-4 in a well-defined order. The order is Server-specific, but the *Client* can assume that the *Server* always returns them in the same order.

There are no additional constraints defined for this *ReferenceType*.

### 7.10 HasSubtype ReferenceType

The *HasSubtype ReferenceType* is a concrete *ReferenceType* that can be used directly. It is a subtype of the *HasChild ReferenceType*.

The semantic of this *ReferenceType* is to express a subtype relationship of types. It is used to span the *ReferenceType* hierarchy, whose semantic is specified in 5.3.3.3; a *DataType* hierarchy is specified in 5.8.3, and other subtype hierarchies are specified in Clause 6.

The *SourceNode* of *References* of this type shall be an *ObjectType*, a *VariableType*, a *DataType* or a *ReferenceType* and the *TargetNode* shall be of the same *NodeClass* as the *SourceNode*. Each *ReferenceType* shall be the *TargetNode* of at most one *Reference* of type *HasSubtype*.

### 7.11 Organizes ReferenceType

The *Organizes ReferenceType* is a concrete *ReferenceType* and can be used directly. It is a subtype of *HierarchicalReferences*.

The semantic of this *ReferenceType* is to organise *Nodes* in the *AddressSpace*. It can be used to span multiple hierarchies independent of any hierarchy created with the non-looping *Aggregates References*.

The *SourceNode* of *References* of this type shall be an *Object* or a *View*. If it is an *Object* then it should be an *Object* of the *ObjectType FolderType* or one of its subtypes (see 5.5.3).

The *TargetNode* of this *ReferenceType* can be of any *NodeClass*.

### 7.12 HasModellingRule ReferenceType

The *HasModellingRule ReferenceType* is a concrete *ReferenceType* and can be used directly. It is a subtype of *NonHierarchicalReferences*.

The semantic of this *ReferenceType* is to bind the *ModellingRule* to an *Object*, *Variable* or *Method*. The *ModellingRule* mechanisms are described in 6.4.4.

The *SourceNode* of this *ReferenceType* shall be an *Object*, *Variable* or *Method*. The *TargetNode* shall be an *Object* of the *ObjectType* “ModellingRule” or one of its subtypes.

Each *Node* shall be the *SourceNode* of at most one *HasModellingRule Reference*.

### 7.13 HasTypeDefinition ReferenceType

The *HasTypeDefinition ReferenceType* is a concrete *ReferenceType* and can be used directly. It is a subtype of *NonHierarchicalReferences*.

The semantic of this *ReferenceType* is to bind an *Object* or *Variable* to its *ObjectType* or *VariableType*, respectively. The relationships between types and instances are described in 4.5.

The *SourceNode* of this *ReferenceType* shall be an *Object* or *Variable*. If the *SourceNode* is an *Object*, then the *TargetNode* shall be an *ObjectType*; if the *SourceNode* is a *Variable*, then the *TargetNode* shall be a *VariableType*.

Each *Variable* and each *Object* shall be the *SourceNode* of exactly one *HasTypeDefinition Reference*.

### 7.14 HasEncoding ReferenceType

The *HasEncoding ReferenceType* is a concrete *ReferenceType* and can be used directly. It is a subtype of *NonHierarchicalReferences*.

The semantic of this *ReferenceType* is to reference *DataTypeEncodings* of a *DataType*.

The *SourceNode* of *References* of this type shall be a *DataType*.

The *TargetNode* of this *ReferenceType* shall be an *Object* of the *ObjectType* *DataTypeEncodingType* or one of its subtypes (see 5.8.4).

### 7.15 HasDescription ReferenceType

The *HasDescription ReferenceType* is a concrete *ReferenceType* and can be used directly. It is a subtype of *NonHierarchicalReferences*.

The semantic of this *ReferenceType* is to reference the *DataTypeDescription* of a *DataTypeEncoding*.

The *SourceNode* of *References* of this type shall be an *Object* of the *ObjectType* *DataTypeEncodingType* or one of its subtypes.

The *TargetNode* of this *ReferenceType* shall be a *Variable* of the *VariableType* *DataTypeDescriptionType* or one of its subtypes (see 5.8.4).

### 7.16 GeneratesEvent

The *GeneratesEvent ReferenceType* is a concrete *ReferenceType* and can be used directly. It is a subtype of *NonHierarchicalReferences*.

The semantic of this *ReferenceType* is to identify the types of *Events* instances of *ObjectTypes* or *VariableTypes* may generate and *Methods* may generate on each *Method* call.

The *SourceNode* of *References* of this type shall be an *ObjectType*, a *VariableType* or a *Method*.

The *TargetNode* of this *ReferenceType* shall be an *ObjectType* representing *EventTypes*, that is, the *BaseEventType* or one of its subtypes.

### 7.17 AlwaysGeneratesEvent

The *AlwaysGeneratesEvent ReferenceType* is a concrete *ReferenceType* and can be used directly. It is a subtype of *GeneratesEvent*.

The semantic of this *ReferenceType* is to identify the types of *Events* *Methods* have to generate on each *Method* call.

The *SourceNode* of *References* of this type shall be a *Method*.

The *TargetNode* of this *ReferenceType* shall be an *ObjectType* representing *EventTypes*, that is, the *BaseEventType* or one of its subtypes.

### 7.18 HasEventSource

The *HasEventSource ReferenceType* is a concrete *ReferenceType* and can be used directly. It is a subtype of *HierarchicalReferences*.

The semantic of this *ReferenceType* is to relate event sources in a hierarchical, non-looping organization. This *ReferenceType* and any subtypes are intended to be used for discovery of *Event* generation in a *Server*. They are not required to be present for a *Server* to generate an *Event* from its source (causing the *Event*) to its notifying *Nodes*. In particular, the root notifier of a *Server*, the *Server Object* defined in IEC 62541-5, is always capable of supplying all *Events* from a *Server* and as such has implied *HasEventSource References* to every event source in a *Server*.

The *SourceNode* of this *ReferenceType* shall be an *Object* that is a source of event subscriptions. A source of event subscriptions is an *Object* that has its "SubscribeToEvents" bit set within the *EventNotifier Attribute*.

The *TargetNode* of this *ReferenceType* can be a *Node* of any *NodeClass* that can generate event notifications via a subscription to the reference source.

Starting from *Node* "A" and only following *References* of the *HasEventSource ReferenceType* or of its subtypes it shall never be possible to return to "A". But it is permitted that, following the *References*, there may be more than one path leading to another *Node* "B".

### 7.19 HasNotifier

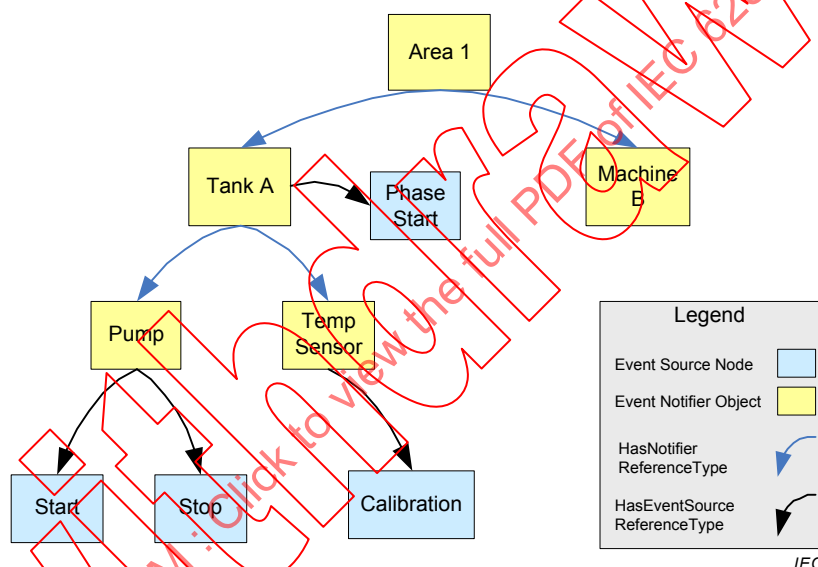
The *HasNotifier ReferenceType* is a concrete *ReferenceType* and can be used directly. It is a subtype of *HasEventSource*.

The semantic of this *ReferenceType* is to relate *Object Nodes* that are notifiers with other notifier *Object Nodes*. The *ReferenceType* is used to establish a hierarchical organization of event notifying *Objects*. It is a subtype of the *HasEventSource ReferenceType* defined in 7.17.

The *SourceNode* of this *ReferenceType* shall be *Objects* or *Views* that are a source of event subscriptions. The *TargetNode* of this *ReferenceType* shall be *Objects* that are a source of event subscriptions. A source of event subscriptions is an *Object* that has its “SubscribeToEvents” bit set within the *EventNotifier Attribute*.

If the *TargetNode* of a *Reference* of this type generates an *Event*, then this *Event* shall also be provided in the *SourceNode* of the *Reference*.

An example of a possible organization of *Event References* is represented in Figure 25. In this example an unfiltered *Event* subscription directed to the “Pump” *Object* will provide the *Event* sources “Start” and “Stop” to the subscriber. An unfiltered *Event* subscription directed to the “Area 1” *Object* will provide *Event* sources from “Machine B”, “Tank A” and all notifier sources below “Tank A”.



**Figure 25 – Event Reference Example**

A second example of a more complex organization of *Event References* is represented in Figure 26. In this example, explicit *References* are included from the *Server's Server Object*, which is a source of all *Server Events*. A second *Event* organization has been introduced to collect the *Events* related to “Tank Farm 1”. An unfiltered *Event* subscription directed to the “Tank Farm 1” *Object* will provide *Event* sources from “Tank B”, “Tank A” and all notifier sources below “Tank B” and “Tank A”.

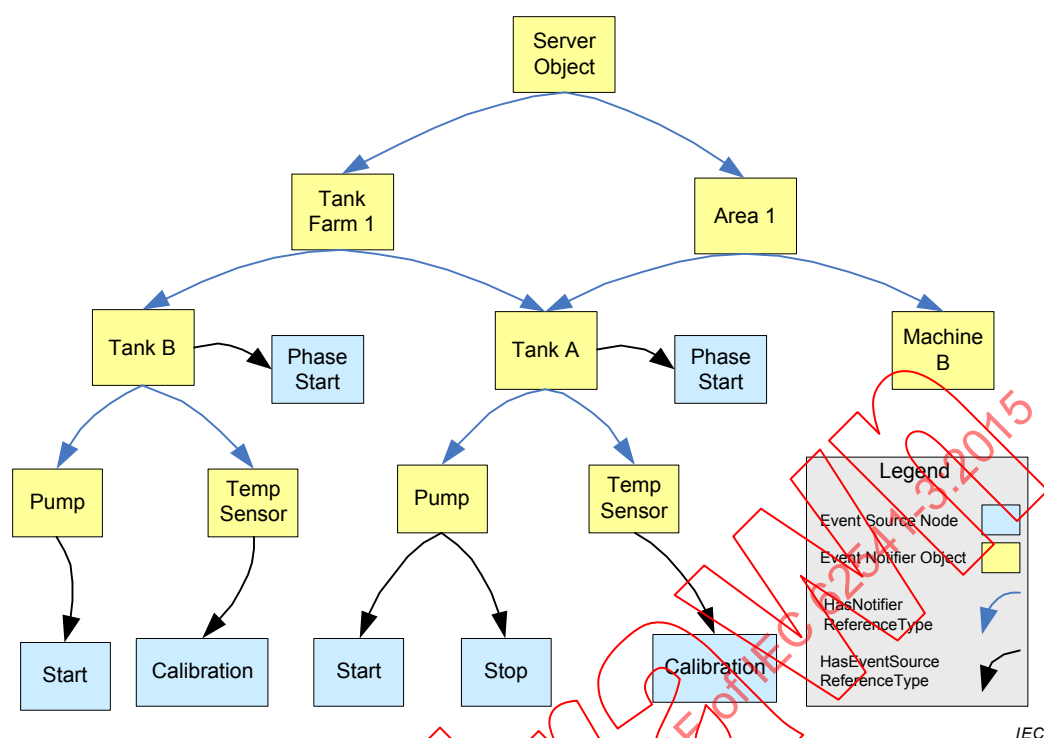


Figure 26 – Complex Event Reference Example

## 8 Standard DataTypes

### 8.1 General

The remainder of 8 defines *DataTypes*. Their representation in the *AddressSpace* and the *DataType* hierarchy is specified in IEC 62541-5. Other parts of the IEC 62541 series may specify additional *DataTypes*.

### 8.2 NodeId

#### 8.2.1 General

This *Built-in DataType* is composed of three elements that identify a *Node* within a *Server*. They are defined in Table 17.

Table 17 – NodeId Definition

| Name           | Type      | Description                                                                                         |
|----------------|-----------|-----------------------------------------------------------------------------------------------------|
| NodeId         | structure |                                                                                                     |
| namespaceIndex | UInt16    | The index for a namespace URI (see 8.2.2).                                                          |
| identifierType | Enum      | The format and data type of the identifier (see 8.2.3).                                             |
| identifier     | *         | The identifier for a <i>Node</i> in the <i>AddressSpace</i> of an OPC UA <i>Server</i> (see 8.2.4). |

See IEC 62541-6 for a description of the encoding of the identifier into OPC UA Messages.

#### 8.2.2 NamespaceIndex

The namespace is a URI that identifies the naming authority responsible for assigning the identifier element of the *NodeId*. Naming authorities include the local *Server*, the underlying system, standards bodies and consortia. It is expected that most *Nodes* will use the URI of the *Server* or of the underlying system.

Using a namespace URI allows multiple OPC UA *Servers* attached to the same underlying system to use the same identifier to identify the same *Object*. This enables *Clients* that connect to those *Servers* to recognise *Objects* that they have in common.

Namespace URIs, like *Server* names, are identified by numeric values in OPC UA *Services* to permit more efficient transfer and processing (e.g. table lookups). The numeric values used to identify namespaces correspond to the index into the *NamespaceArray*. The *NamespaceArray* is a *Variable* that is part of the *Server Object* in the *AddressSpace* (see IEC 62541-5 for its definition).

The URI for the OPC UA namespace is:

"http://opcfoundation.org/UA/"

Its corresponding index in the namespace table is 0.

The namespace URI is case sensitive.

### 8.2.3 IdentifierType

The *IdentifierType* element identifies the type of the *NodeId*, its format and its scope. Its values are defined in Table 18.

**Table 18 – IdentifierType Values**

| Value     | Description                |
|-----------|----------------------------|
| NUMERIC_0 | Numeric value              |
| STRING_1  | String value               |
| GUID_2    | Globally Unique Identifier |
| OPAQUE_3  | Namespace specific format  |

Normally the scope of *NodeIds* is the *Server* in which they are defined. For certain types of *NodeIds*, *NodeIds* can uniquely identify a *Node* within a system, or across systems (e.g. GUIDs). System-wide and globally-unique identifiers allow *Clients* to track *Nodes*, such as work orders, as they move between OPC UA *Servers* as they progress through the system.

Opaque identifiers are identifiers that are free-format byte strings that might or might not be human interpretable.

String identifiers are case sensitive. That is, *Clients* shall consider them case sensitive. *Servers* are allowed to provide alternative *NodeIds* (see 5.2.2) and using this mechanism servers can handle *NodeIds* as case insensitive.

### 8.2.4 Identifier value

The identifier value element is used within the context of the first three elements to identify the *Node*. Its data type and format is defined by the *IdType*.

Identifier values of *IdType* STRING\_1 are restricted to 4 096 characters. Identifier values of *IdType* OPAQUE\_3 are restricted to 4 096 bytes.

A null *NodeId* has special meaning. For example, many services defined in IEC 62541-4 define special behaviour if a null *NodeId* is passed as a parameter. Each *IdType* has a set of identifier values that represent a null *NodeId*. These values are summarised in Table 19.

**Table 19 – NodeId Null Values**

| IdType    | Identifier                                                            |
|-----------|-----------------------------------------------------------------------|
| NUMERIC_0 | 0                                                                     |
| STRING_1  | A null or Empty String ("")                                           |
| GUID_2    | A Guid initialised with zeros (e.g. 00000000-0000-0000-0000-00000000) |
| OPAQUE_3  | A ByteString with Length=0                                            |

A null *NodeId* always has a NamespaceIndex equal to 0.

A *Node* in the *AddressSpace* shall not have a null as its *NodeId*.

### 8.3 QualifiedName

This *Built-in DataType* contains a qualified name. It is, for example, used as *BrowseName*. Its elements are defined in Table 20. The name part of the *QualifiedName* is restricted to 512 characters.

**Table 20 – QualifiedName Definition**

| Name           | Type      | Description                                                                                                                                                                                                                                                         |
|----------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| QualifiedName  | structure |                                                                                                                                                                                                                                                                     |
| namespaceIndex | UInt16    | Index that identifies the namespace that defines the name.<br>This index is the index of that namespace in the local <i>Server's NamespaceArray</i> .<br>The <i>Client</i> may read the <i>NamespaceArray</i> Variable to access the string value of the namespace. |
| name           | String    | The text portion of the <i>QualifiedName</i> .                                                                                                                                                                                                                      |

### 8.4 LocaleId

This *Simple DataType* is specified as a string that is composed of a language component and a country/region component as specified by IETF RFC 3066. The <country/region> component is always preceded by a hyphen. The format of the *LocaleId* string is shown below:

<language>[-<country/region>], where  
 <language> is the two letter ISO 639 code for a language,  
 <country/region> is the two letter ISO 3166 code for the country/region.

The rules for constructing *LocaleIds* defined by IETF RFC 3066 are restricted as follows:

- this specification permits only zero or one <country/region> component to follow the <language> component;
- this specification also permits the "-CHS" and "-CHT" three-letter <country/region> codes for "Simplified" and "Traditional" Chinese locales;
- this specification also allows the use of other <country/region> codes as deemed necessary by the *Client* or the *Server*.

Table 21 shows examples of OPC UA *LocaleIds*. *Clients* and *Servers* always provide *LocaleIds* that explicitly identify the language and the country/region.

**Table 21 – LocaleId Examples**

| Locale            | OPC UA LocaleId |
|-------------------|-----------------|
| English           | en              |
| English (US)      | en-US           |
| German            | de              |
| German (Germany)  | de-DE           |
| German (Austrian) | de-AT           |

An empty or null string indicates that the *LocaleId* is unknown.

## 8.5 LocalizedText

This *Built-in DataType* defines a structure containing a String in a locale-specific translation specified in the identifier for the locale. Its elements are defined in Table 22.

**Table 22 – LocalizedText Definition**

| Name          | Type      | Description                                   |
|---------------|-----------|-----------------------------------------------|
| LocalizedText | structure |                                               |
| text          | String    | The localized text.                           |
| locale        | LocaleId  | The identifier for the locale (e.g. "en-US"). |

## 8.6 Argument

This *Structured DataType* defines a *Method* input or output argument specification. It is for example used in the input and output argument *Properties* for *Methods*. Its elements are described in Table 23.

**Table 23 – Argument Definition**

| Name            | Type          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argument        | structure     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| name            | String        | The name of the argument.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| dataType        | NodeId        | The <i>NodeId</i> of the <i>DataType</i> of this argument.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| valueRank       | Int32         | Indicates whether the <i>dataType</i> is an array and how many dimensions the array has.<br>It may have the following values:<br>$n > 1$ : the <i>dataType</i> is an array with the specified number of dimensions.<br>OneDimension (1): The <i>dataType</i> is an array with one dimension.<br>OneOrMoreDimensions (0): The <i>dataType</i> is an array with one or more dimensions.<br>Scalar (-1): The <i>dataType</i> is not an array.<br>Any (-2): The <i>dataType</i> can be a scalar or an array with any number of dimensions.<br>ScalarOrOneDimension (-3): The <i>dataType</i> can be a scalar or a one dimensional array.<br>NOTE All <i>DataTypes</i> are considered to be scalar, even if they have array-like semantics like <i>ByteString</i> and <i>String</i> . |
| arrayDimensions | UInt32[]      | Specifies the length of each dimension for an array <i>dataType</i> . It is intended to describe the capability of the <i>dataType</i> , not the current size.<br>The number of elements shall be equal to the value of the <i>valueRank</i> . Shall be null if <i>valueRank</i> $\leq 0$ .<br>A value of 0 for an individual dimension indicates that the dimension has a variable length.                                                                                                                                                                                                                                                                                                                                                                                      |
| description     | LocalizedText | A localised description of the argument.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

## 8.7 BaseDataType

This abstract *DataType* defines a value that can have any valid *DataType*.

It defines a special value null indicating that a value is not present.

## 8.8 Boolean

This *Built-in DataType* defines a value that is either TRUE or FALSE.

## 8.9 Byte

This *Built-in DataType* defines a value in the range of 0 to 255.

### 8.10 ByteString

This *Built-in DataType* defines a value that is a sequence of Byte values.

### 8.11 DateTime

This *Built-in DataType* defines a Gregorian calendar date. Details about this *DataType* are defined in IEC 62541-6.

### 8.12 Double

This *Built-in DataType* defines a value that adheres to the IEEE 754-1985 double precision data type definition.

### 8.13 Duration

This *Simple DataType* is a *Double* that defines an interval of time in milliseconds (fractions can be used to define sub-millisecond values). Negative values are generally invalid but may have special meanings where the *Duration* is used.

### 8.14 Enumeration

This abstract *DataType* is the base *DataType* for all enumeration *DataTypes* like *NodeClass* defined in 8.30. All *DataTypes* inheriting from this *DataType* have a special handling for the encoding as defined in IEC 62541-6. All enumeration *DataTypes* shall inherit from this *DataType*.

Some special rules apply when subtyping enumerations. Any enumeration *DataType* not directly inheriting from the *Enumeration DataType* can only restrict the enumeration values of its supertype. That is, it shall neither add enumeration values nor change the text associated to the enumeration value. As an example, the enumeration Days having {'Mo', 'Tu', 'We', 'Th', 'Fr', 'Sa', 'Su'} as values can be subtyped to the enumeration Workdays having {'Mo', 'Tu', 'We', 'Th', 'Fr'}. The other direction, subtyping Workdays to Days would not be allowed as Days has values not allowed by Workdays ('Sa' and 'Su').

### 8.15 Float

This *Built-in DataType* defines a value that adheres to the IEEE 754-1985 single precision data type definition.

### 8.16 Guid

This *Built-in DataType* defines a value that is a 128-bit Globally Unique Identifier. Details about this *DataType* are defined in IEC 62541-6.

### 8.17 SByte

This *Built-in DataType* defines a value that is a signed integer between -128 and 127 inclusive.

### 8.18 IdType

This *DataType* is an enumeration that identifies the *IdType* of a *NodeId*. Its values are defined in Table 18. See 8.2.3 for a description of the use of this *DataType* in *NodeIds*.

### 8.19 Image

This abstract *DataType* defines a *ByteString* representing an image.

## 8.20 ImageBMP

This *Simple DataType* defines a *ByteString* representing an image in BMP format.

## 8.21 ImageGIF

This *Simple DataType* defines a *ByteString* representing an image in GIF format.

## 8.22 ImageJPG

This *Simple DataType* defines a *ByteString* representing an image in JPG format. JPG is defined in ISO/IEC 10918-1.

## 8.23 ImagePNG

This *Simple DataType* defines a *ByteString* representing an image in PNG format. PNG is defined in ISO/IEC 15948.

## 8.24 Integer

This abstract *DataType* defines an integer whose length is defined by its subtypes.

## 8.25 Int16

This *Built-in DataType* defines a value that is a signed integer between –32 768 and 32 767 inclusive.

## 8.26 Int32

This *Built-in DataType* defines a value that is a signed integer between –2 147 483 648 and 2 147 483 647 inclusive.

## 8.27 Int64

This *Built-in DataType* defines a value that is a signed integer between –9 223 372 036 854 775 808 and 9 223 372 036 854 775 807 inclusive.

## 8.28 TimeZoneDataType

This *Structured DataType* defines the local time that may or may not take daylight saving time into account. Its elements are described in Table 24.

**Table 24 – TimeZoneDataType Definition**

| Name                   | Type      | Description                                                                                                                                                                                                          |
|------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TimeZoneDataType       | structure |                                                                                                                                                                                                                      |
| offset                 | Int16     | The offset in minutes from UtcTime                                                                                                                                                                                   |
| daylightSavingInOffset | Boolean   | If TRUE, then daylight saving time (DST) is in effect and <i>offset</i> includes the DST correction. If FALSE then the <i>offset</i> does not include the DST correction and DST may or may not have been in effect. |

## 8.29 NamingRuleType

This *DataType* is an enumeration that identifies the *NamingRule* (see 6.4.4.2). Its values are defined in Table 25.

**Table 25 – NamingRuleType Values**

| Name         |
|--------------|
| MANDATORY_1  |
| OPTIONAL_2   |
| CONSTRAINT_3 |

### 8.30 NodeClass

This *DataType* is an enumeration that identifies a *NodeClass*. Its values are defined in Table 26.

**Table 26 – NodeClass Values**

| Name              |
|-------------------|
| OBJECT_1          |
| VARIABLE_2        |
| METHOD_4          |
| OBJECT_TYPE_8     |
| VARIABLE_TYPE_16  |
| REFERENCE_TYPE_32 |
| DATA_TYPE_64      |
| VIEW_128          |

### 8.31 Number

This abstract *DataType* defines a number. Details are defined by its subtypes.

### 8.32 String

This *Built-in DataType* defines a Unicode character string that should exclude control characters that are not whitespaces.

### 8.33 Structure

This abstract *DataType* is the base *DataType* for all *Structured DataTypes* like *Argument* defined in 8.6. All *DataTypes* inheriting from this *DataType* have a special handling for the encoding as defined in IEC 62541-6. All *Structured DataTypes* shall inherit from this *DataType* if they are not defined as primitives in this standard (like *NodeId* defined in 8.2; a *NodeId* is structured but treated in a special way as defined in IEC 62541-6).

### 8.34 UInteger

This abstract *DataType* defines an unsigned integer whose length is defined by its subtypes.

### 8.35 UInt16

This *Built-in DataType* defines a value that is an unsigned integer between 0 and 65 535 inclusive.

### 8.36 UInt32

This *Built-in DataType* defines a value that is an unsigned integer between 0 and 4 294 967 295 inclusive.

### 8.37 UInt64

This *Built-in DataType* defines a value that is an unsigned integer between 0 and 18 446 744 073 709 551 615 inclusive.

### 8.38 UtcTime

This *simple DataType* is a *DateTime* used to define Coordinated Universal Time (UTC) values. All time values conveyed between OPC UA Servers and Clients are UTC values. Clients shall provide any conversions between UTC and local time.

UTC has the concept of leap seconds. Leap seconds can lead to repeating seconds. Therefore applications are allowed to use TAI<sup>1</sup> (International Atomic Time) instead of UTC in any place where UtcTime is used. Details on time synchronization are discussed in IEC 62541-6.

### 8.39 XmlElement

This *Built-in DataType* is used to define XML elements. IEC 62541-6 defines details about this *DataType*.

XML data can always be modelled as a subtype of the *Structure DataType* with a single *DataTypeEncoding* that represents the XML complexType that defines the XML element (it is not necessary to have access to the XML Schema to define a *DataTypeEncoding*). For this reason a Server should never define *Variables* that use the *XmlElement DataType* unless the Server has no information about the XML elements that might be in the *Variable Value*.

### 8.40 EnumValueType

This *Structured DataType* is used to represent a human-readable representation of an Enumeration. Its elements are described in Table 23. When this type is used in an array representing human-readable representations of an enumeration, each Value shall be unique in that array.

**Table 27 – EnumValueType Definition**

| Name          | Type          | Description                                                                                                              |
|---------------|---------------|--------------------------------------------------------------------------------------------------------------------------|
| EnumValueType | structure     |                                                                                                                          |
| Value         | Int64         | The Integer representation of an Enumeration.                                                                            |
| DisplayName   | LocalizedText | A human-readable representation of the Value of the Enumeration.                                                         |
| Description   | LocalizedText | A localized description of the enumeration value. This field can contain an empty string if no description is available. |

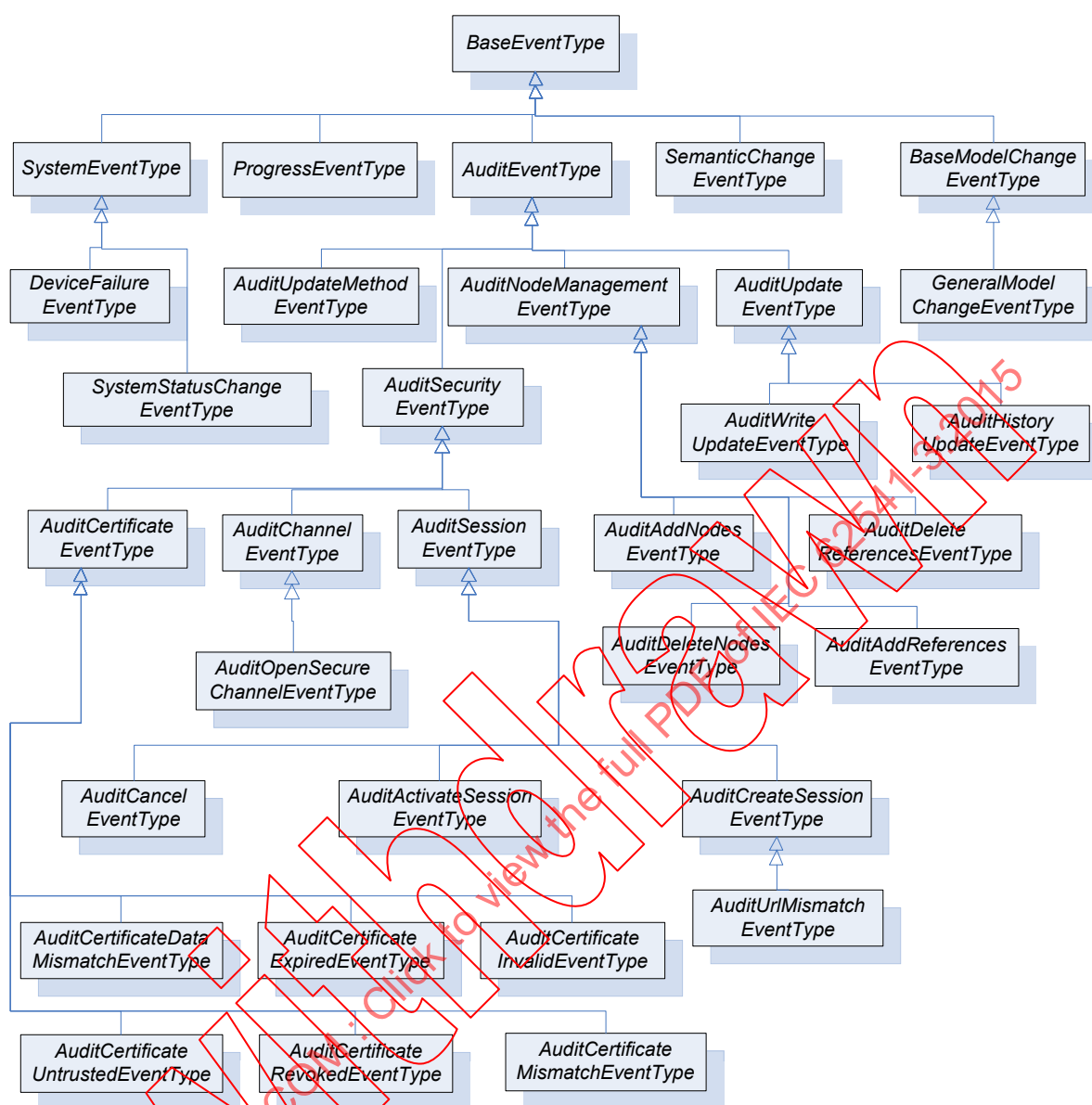
Note that the Value is an Int64 although the DataType to transfer an Enumeration DataType is an Int32. This is because this DataType is also used in places, e.g. in IEC 62541-8, where an Int64 is appropriate.

## 9 Standard EventTypes

### 9.1 General

The remainder of 9 defines *EventTypes*. Their representation in the *AddressSpace* is specified in IEC 62541-5. Other parts of the IEC 62541 series may specify additional *EventTypes*. Figure 27 informally describes the hierarchy of these *EventTypes*.

<sup>1</sup> TAI = temps atomique international.



IEC

Figure 27 – Standard EventType Hierarchy

## 9.2 BaseEventType

The *BaseEventType* defines all general characteristics of an *Event*. All other *EventTypes* derive from it. There is no other semantic associated with this type.

## 9.3 SystemEventType

*SystemEvents* are *Events* of *SystemEventType* that are generated as a result of some *Event* that occurs within the *Server* or by a system that the *Server* is representing.

## 9.4 ProgressEventType

*ProgressEvents* are *Events* of *ProgressEventType* that are generated to identify the progress of an operation. An operation can be a service call or something application specific like a program execution.

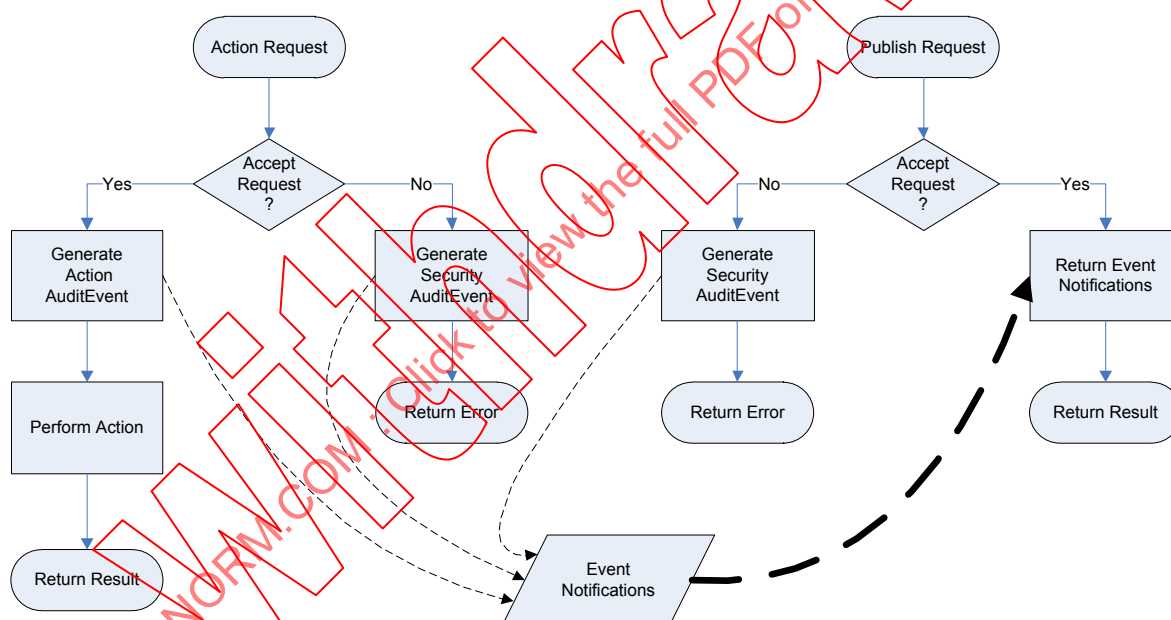
## 9.5 AuditEventType

*AuditEvents* are *Events* of *AuditEventType* that are generated as a result of an action taken on the *Server* by a *Client* of the *Server*. For example, in response to a *Client* issuing a write to a *Variable*, the *Server* would generate an *AuditEvent* describing the *Variable* as the source and the user and *Client* session as the initiators of the *Event*.

Figure 28 illustrates the defined behaviour of an OPC UA *Server* in response to an auditable action request. If the action is accepted, then an action *AuditEvent* is generated and processed by the *Server*. If the action is not accepted due to security reasons, a security *AuditEvent* is generated and processed by the *Server*. The *Server* may involve the underlying device or system in the process but it is the *Server's* responsibility to provide the *Event* to any interested *Clients*. *Clients* are free to subscribe to *Events* from the *Server* and will receive the *AuditEvents* in response to normal Publish requests.

All action requests include a human readable *AuditEntryId*. The *AuditEntryId* is included in the *AuditEvent* to allow human readers to correlate an *Event* with the initiating action. The *AuditEntryId* typically contains who initiated the action and from where it was initiated.

The *Server* may elect to optionally persist the *AuditEvents* in addition to the mandatory *Event Subscription* delivery to *Clients*.

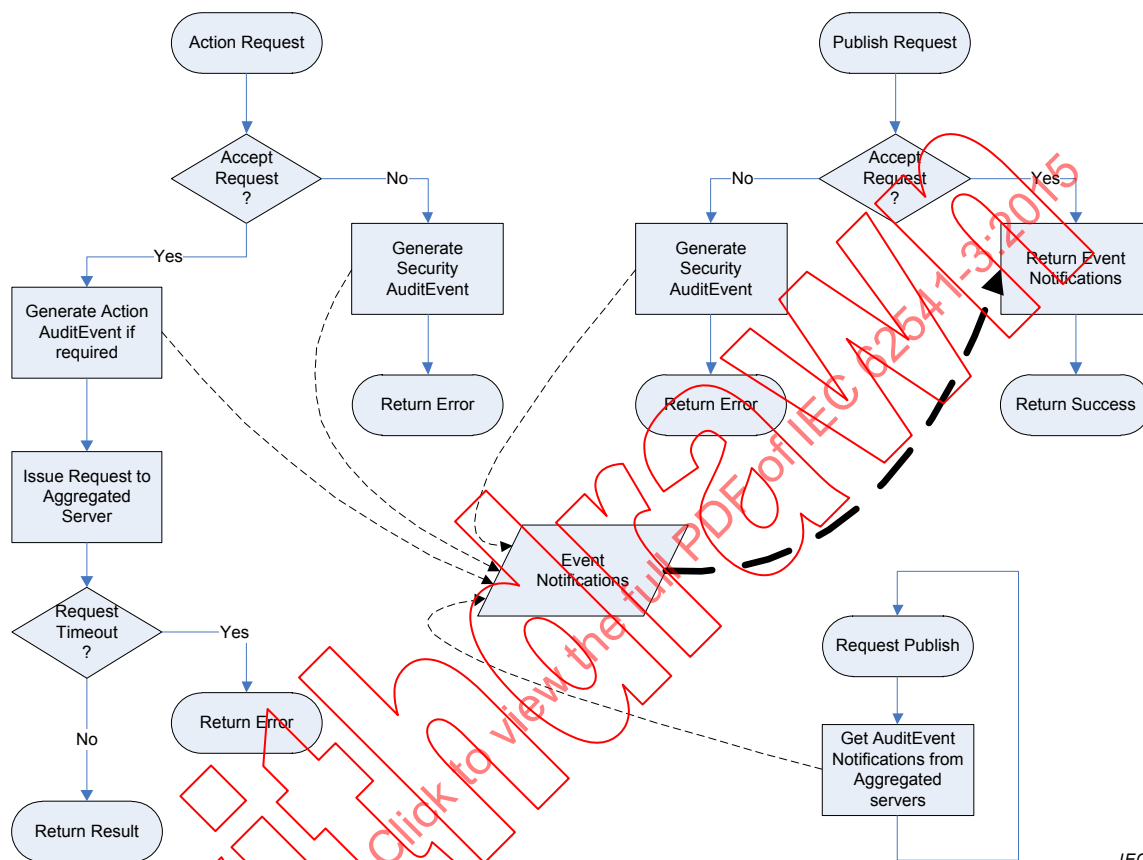


IEC

**Figure 28 – Audit Behaviour of a Server**

Figure 29 illustrates the expected behaviour of an aggregating *Server* in response to an auditable action request. This use case involves the aggregating *Server* passing on the action to one of its aggregated *Servers*. The general behaviour described above is extended by this behaviour and not replaced. That is, the request could fail and generate a security *AuditEvent* within the aggregating *Server*. The normal process is to pass the action down to an aggregated *Server* for processing. The aggregated *Server* will, in turn, follow this behaviour or the general behaviour and generate the appropriate *AuditEvents*. The aggregating *Server* periodically issues publish requests to the aggregated *Servers*. These collected *Events* are merged with self-generated *Events* and made available to subscribing *Clients*. If the aggregating *Server* supports the optional persisting of *AuditEvent*, then the collected *Events* are persisted along with locally-generated *Events*.

The aggregating *Server* may map the authenticated user account making the request to one of its own accounts when passing on the request to an aggregated *Server*. It shall, however, preserve the *AuditEntryId* by passing it on as received. The aggregating *Server* may also generate its own *AuditEvent* for the request prior to passing it on to the aggregated *Server*, in particular, if the aggregating *Server* needs to break a request into multiple requests that are each directed to separate aggregated *Servers* or if part of a request is denied due to security on the aggregating *Server*.



IEC

Figure 29 – Audit Behaviour of an Aggregating Server

## 9.6 AuditSecurityEventType

This is a subtype of *AuditEventType* and is used only for categorization of security-related *Events*. This type follows all behaviour of its parent type.

## 9.7 AuditChannelEventType

This is a subtype of *AuditSecurityEventType* and is used for categorization of security-related *Events* from the *SecureChannel Service Set* defined in IEC 62541-4.

## 9.8 AuditOpenSecureChannelEventType

This is a subtype of *AuditChannelEventType* and is used for *Events* generated from calling the *OpenSecureChannel Service* defined in IEC 62541-4.

## 9.9 AuditSessionEventType

This is a subtype of *AuditSecurityEventType* and is used for categorization of security-related *Events* from the *Session Service Set* defined in IEC 62541-4.

### 9.10 AuditCreateSessionEventType

This is a subtype of *AuditSessionEventType* and is used for *Events* generated from calling the *CreateSession Service* defined in IEC 62541-4.

### 9.11 AuditUrlMismatchEventType

This is a subtype of *AuditCreateSessionEventType* and is used for *Events* generated from calling the *CreateSession Service* defined in IEC 62541-4 if the *EndpointUrl* used in the service call does not match the *Server's HostNames* (see IEC 62541-4 for details).

### 9.12 AuditActivateSessionEventType

This is a subtype of *AuditSessionEventType* and is used for *Events* generated from calling the *ActivateSession Service* defined in IEC 62541-4.

### 9.13 AuditCancelEventType

This is a subtype of *AuditSessionEventType* and is used for *Events* generated from calling the *Cancel Service* defined in IEC 62541-4.

### 9.14 AuditCertificateEventType

This is a subtype of *AuditSecurityEventType* and is used only for categorization of Certificate related *Events*. This type follows all behaviours of its parent type. These *AuditEvents* will be generated for Certificate errors in addition to other *AuditEvents* related to service calls.

### 9.15 AuditCertificateDataMismatchEventType

This is a subtype of *AuditCertificateEventType* and is used only for categorization of Certificate related *Events*. This type follows all behaviours of its parent type. This *AuditEvent* is generated if the *HostName* in the URL used to connect to the *Server* is not the same as one of the *HostNames* specified in the Certificate or if the *Application* and *Software Certificates* contain an application or product URI that does not match the URI specified in the *ApplicationDescription* provided with the Certificate. For more details on Certificates see IEC 62541-4.

### 9.16 AuditCertificateExpiredEventType

This is a subtype of *AuditCertificateEventType* and is used only for categorization of Certificate related *Events*. This type follows all behaviours of its parent type. This *AuditEvent* is generated if the current time is outside the validity period's start date and end date.

### 9.17 AuditCertificateInvalidEventType

This is a subtype of *AuditCertificateEventType* and is used only for categorization of Certificate related *Events*. This type follows all behaviours of its parent type. This *AuditEvent* is generated if the certificate structure is invalid or if the Certificate has an invalid signature.

### 9.18 AuditCertificateUntrustedEventType

This is a subtype of *AuditCertificateEventType* and is used only for categorization of Certificate related *Events*. This type follows all behaviours of its parent type. This *AuditEvent* is generated if the Certificate is not trusted, that is, if the *Issuer Certificate* is unknown.

### 9.19 AuditCertificateRevokedEventType

This is a subtype of *AuditCertificateEventType* and is used only for categorization of Certificate related *Events*. This type follows all behaviours of its parent type. This *AuditEvent*

is generated if a Certificate has been revoked or if the revocation list is not available (i.e. a network interruption prevents the Application from accessing the list).

### 9.20 AuditCertificateMismatchEventType

This is a subtype of *AuditCertificateEventType* and is used only for categorization of Certificate related *Events*. This type follows all behaviours of its parent type. This *AuditEvent* is generated if a Certificate set of uses does not match the requested use for the Certificate (i.e. Application, Software or Certificate Authority).

### 9.21 AuditNodeManagementEventType

This is a subtype of *AuditEventType* and is used for categorization of node management related *Events*. This type follows all behaviours of its parent type.

### 9.22 AuditAddNodesEventType

This is a subtype of *AuditNodeManagementEventType* and is used for *Events* generated from calling the AddNodes Service defined in IEC 62541-4.

### 9.23 AuditDeleteNodesEventType

This is a subtype of *AuditNodeManagementEventType* and is used for *Events* generated from calling the DeleteNodes Service defined in IEC 62541-4.

### 9.24 AuditAddReferencesEventType

This is a subtype of *AuditNodeManagementEventType* and is used for *Events* generated from calling the AddReferences Service defined in IEC 62541-4.

### 9.25 AuditDeleteReferencesEventType

This is a subtype of *AuditNodeManagementEventType* and is used for *Events* generated from calling the DeleteReferences Service defined in IEC 62541-4.

### 9.26 AuditUpdateEventType

This is a subtype of *AuditEventType* and is used for categorization of update related *Events*. This type follows all behaviours of its parent type.

### 9.27 AuditWriteUpdateEventType

This is a subtype of *AuditUpdateEventType* and is used for categorization of write update related *Events*. This type follows all behaviours of its parent type.

### 9.28 AuditHistoryUpdateEventType

This is a subtype of *AuditUpdateEventType* and is used for categorization of history update related *Events*. This type follows all behaviours of its parent type.

### 9.29 AuditUpdateMethodEventType

This is a subtype of *AuditEventType* and is used for categorization of *Method* related *Events*. This type follows all behaviours of its parent type.

### 9.30 DeviceFailureEventType

A *DeviceFailureEvent* is an *Event* of *DeviceFailureEventType* that indicates a failure in a device of the underlying system.

### 9.31 SystemStatusChangeEvent

A *SystemStatusChangeEvent* is an *Event* of *SystemStatusChangeEvent* type that indicates a status change in a system. For example, if the status indicates an underlying system is not running, then a *Client* cannot expect any *Events* from the underlying system. A *Server* can identify its own status changes using this *EventType*.

### 9.32 ModelChangeEvents

#### 9.32.1 General

*ModelChangeEvents* are generated to indicate a change of the *AddressSpace* structure. The change may consist of adding or deleting a *Node* or *Reference*. Although the relationship of a *Variable* or *VariableType* to its *DataType* is not modelled using *References*, changes to the *DataType Attribute* of a *Variable* or *VariableType* are also considered as model changes and therefore a *ModelChangeEvent* is generated if the *DataType Attribute* changes.

#### 9.32.2 NodeVersion Property

There is a correlation between *ModelChangeEvents* and the *NodeVersion Property* of *Nodes*. Every time a *ModelChangeEvent* is issued for a *Node*, its *NodeVersion* shall be changed, and every time the *NodeVersion* is changed, a *ModelChangeEvent* shall be generated. A *Server* shall support both the *ModelChangeEvent* and the *NodeVersion Property* or neither, but never only one of the two mechanisms.

This relation also implies that only those *Nodes* of the *AddressSpace* having a *NodeVersion* shall trigger a *ModelChangeEvent*. Other *Nodes* shall not trigger a *ModelChangeEvent*.

#### 9.32.3 Views

A *ModelChangeEvent* is always generated in the context of a *View*, including the default *View* where the whole *AddressSpace* is considered. Therefore the only *Notifiers* which report the *ModelChangeEvents* are *View Nodes* and the *Server Object* representing the default *View*. Each action generating a *ModelChangeEvent* may lead to several *Events* since it may affect different *Views*. If, for example, a *Node* was deleted from the *AddressSpace*, and this *Node* was also contained in a *View* "A", there would be one *Event* having the *AddressSpace* as context and another having the *View* "A" as context. If a *Node* would only be removed from *View* "A", but still exists in the *AddressSpace*, it would generate only a *ModelChangeEvent* for *View* "A".

If a *Client* does not want to receive duplicates of changes then it shall use the filter mechanisms of the *Event* subscription to filter only for the default *View* and suppress the *ModelChangeEvents* having other *Views* as the context.

When a *ModelChangeEvent* is issued on a *View* and the *View* supports the *ViewVersion Property*, then the *ViewVersion* shall be updated.

#### 9.32.4 Event Compression

An implementation is not required to issue an *Event* for every update as it occurs. An OPC UA *Server* may be capable of grouping a series of transactions or simple updates into a larger unit. This series may constitute a logical grouping or a temporal grouping of changes. A single *ModelChangeEvent* may be issued after the last change of the series, to cover all of the changes. This is referred to as *Event compression*. A change in the *NodeVersion* and the *ViewVersion* may thus reflect a group of changes and not a single change.

#### 9.32.5 BaseModelChangeEvent

*BaseModelChangeEvents* are *Events* of the *BaseModelChangeEvent* type. The *BaseModelChangeEvent* is the base type for *ModelChangeEvents* and does not contain

information about the changes but only indicates that changes occurred. Therefore the *Client* shall assume that any or all of the *Nodes* may have changed.

### 9.32.6 GeneralModelChangeEvent

*GeneralModelChangeEvents* are *Events* of the *GeneralModelChangeEvent* type. The *GeneralModelChangeEvent* is a subtype of the *BaseModelChangeEvent*. It contains information about the *Node* that was changed and the action that occurred to cause the *ModelChangeEvent* (e.g. add a *Node*, delete a *Node*, etc.). If the affected *Node* is a *Variable* or *Object*, then the *TypeDefinitionNode* is also present.

To allow *Event* compression, a *GeneralModelChangeEvent* contains an array of changes.

### 9.32.7 Guidelines for ModelChangeEvents

Two types of *ModelChangeEvents* are defined: the *BaseModelChangeEvent* that does not contain any information about the changes and the *GeneralModelChangeEvent* that identifies the changed *Nodes* via an array. The precision used depends on both the capability of the OPC UA Server and the nature of the update. An OPC UA Server may use either *ModelChangeEvent* type depending on circumstances. It may also define subtypes of these *EventTypes* adding additional information.

To ensure interoperability, the following guidelines for *Events* should be observed.

- If the array of the *GeneralModelChangeEvent* is present, then it should identify every *Node* that has changed since the preceding *ModelChangeEvent*.
- The OPC UA Server should emit exactly one *ModelChangeEvent* for an update or series of updates. It should not issue multiple types of *ModelChangeEvent* for the same update.
- Any *Client* that responds to *ModelChangeEvents* should respond to any *Event* of the *BaseModelChangeEvent* type including its subtypes like the *GeneralModelChangeEvent*.

If a *Client* is not capable of interpreting additional information of the subtypes of the *BaseModelChangeEvent*, it should treat *Events* of these types the same way as *Events* of the *BaseModelChangeEvent*.

## 9.33 SemanticChangeEventType

### 9.33.1 General

*SemanticChangeEvents* are *Events* of *SemanticChangeEventType* that are generated to indicate a change of the *AddressSpace* semantics. The change consists of a change to the *Value Attribute* of a *Property*.

The *SemanticChangeEvent* contains information about the *Node* owning the *Property* that was changed. If this is a *Variable* or *Object*, the *TypeDefinitionNode* is also present.

The *SemanticChange* bit of the *AccessLevel Attribute* of a *Property* indicates whether changes of the *Property* value are considered for *SemanticChangeEvents* (see 5.6.2).

### 9.33.2 ViewVersion and NodeVersion Properties

The *ViewVersion* and *NodeVersion Properties* do not change due to the publication of a *SemanticChangeEvent*.

There is no standard way to identify which *Nodes* trigger a *SemanticChange Event* and which *Nodes* do not.

### 9.33.3 Views

*SemanticChangeEvents* are handled in the context of a *View* the same way as *ModelChangeEvents*. This is defined in 9.32.3.

### 9.33.4 Event Compression

*SemanticChangeEvents* can be compressed the same way as *ModelChangeEvents*. This is defined in 9.32.4.

IECNORM.COM: Click to view the full PDF of IEC 62541-3:2015

## Annex A (informative)

### How to use the Address Space Model

#### A.1 Overview

Annex A points out some general considerations on how the Address Space Model can be used. Annex A is for information only, that is, each *Server* vendor can model its data in the appropriate way that fits its needs. However, it gives some hints the *Server* vendor may consider.

Typically OPC UA *Servers* will offer data provided by an underlying system like a device, a configuration database, an OPC COM *Server*, etc. Therefore the modelling of the data depends on the model of the underlying system as well as the requirements of the *Clients* accessing the OPC UA *Server*. It is also expected that companion specifications will be developed on top of OPC UA with additional rules on how to model the data. However, the remainder of Annex A will give some general considerations about the different concepts of OPC UA to model data and when they should be used, and when not.

IEC 62541-5:–, Annex A, provides an overview of the design decisions made when modelling the information about the *Server* defined in IEC 62541-5.

#### A.2 Type definitions

Type definitions should be used whenever it is expected that the type information may be used more than once in the same system or for interoperability between different systems supporting the same type definitions.

#### A.3 ObjectTypes

Subclause 5.5.1 states: “*Objects* are used to represent systems, system components, real-world objects, and software objects.” Therefore *ObjectTypes* should be used if a type definition of those *ObjectTypes* is useful (see A.2).

From a more abstract point of view *Objects* are used to group *Variables* and other *Objects* in the *AddressSpace*. Therefore *ObjectTypes* should be used when some common structures/groups of *Objects* and/or *Variables* should be described. *Clients* can use this knowledge to program against the *ObjectType* structure and use the *TranslateBrowsePathsToNodeIds Service* defined in IEC 62541-4 on the instances.

Simple objects only having one value (e.g. a simple heat sensor) can also be modelled as *VariableTypes*. However, extensibility mechanisms should be considered (e.g. a complex heat sensor subtype could have several values) and whether that object should be exposed as an object in the *Client's* GUI or just as a value. Whenever a modeller is in doubt as to which solution to use the *ObjectType* having one *Variable* should be preferred.

## A.4 VariableTypes

### A.4.1 General

*VariableTypes* are only used for *DataVariables*<sup>2</sup> and should be used when there are several *Variables* having the same semantic (e.g. set point). It is not necessary to define a *VariableType* that only reflects the *DataType* of a *Variable*, e.g. an “Int32VariableType”.

### A.4.2 Properties or DataVariables

Besides the semantic differences of *Properties* and *DataVariables* described in Clause 4 there are also syntactical differences. A *Property* is identified by its *BrowseName*, that is, if *Properties* having the same semantic are used several times, they should always have the same *BrowseName*. The same semantic of *DataVariables* is captured in the *VariableType*.

If it is not clear which concept to use based on the semantic described in Clause 4, then the different syntax can help. The following points identify when it shall be a *DataVariable*.

- If it is a complex *Variable* or it should contain additional information in the form of *Properties*.
- If the type definition may be refined (subtyping).
- If the type definition should be made available so the *Client* can use the *AddNodes Service* defined in IEC 62541-4 to create new instances of the type definition.
- If it is a component of a complex *Variable* exposing a part of the value of the complex *Variable*.

### A.4.3 Many Variables and / or structured DataTypes

When structured data structures should be made available to the *Client* there are basically three different approaches:

- a) Create several simple *Variables* using simple *DataTypes* always reflecting parts of the simple structure. *Objects* are used to group the *Variables* according to the structure of the data.
- b) Create a structured *DataType* and a simple *Variable* using this *DataType*.
- c) Create a structured *DataType* and a complex *Variable* using this *DataType* and also exposing the structured data structure as *Variables* of the complex *Variable* using simple *DataTypes*.

The advantages of the first approach are that the complex structure of the data is visible in the *AddressSpace*. A generic *Client* can easily access the data without knowledge of user-defined *DataTypes* and the *Client* can access individual parts of the structured data. The disadvantages of the first approach are that accessing the individual data does not provide any transactional context and for a specific *Client* the *Server* first has to convert the data and the *Client* has to convert the data, again, to get the data structure the underlying system provides.

The advantages of the second approach are, that the data is accessed in a transactional context and the structured *DataType* can be constructed in a way that the *Server* does not have to convert the data and can pass directly to the specific *Client* that can directly use them. The disadvantages are that the generic *Client* might not be able to access and interpret the data or has at least the burden to read the *DataTypeDescription* to interpret the data. The structure of the data is not visible in the *AddressSpace*; additional *Properties* describing the data structure cannot be added to the adequate places since they do not exist in the *AddressSpace*. Individual parts of the data cannot be read without accessing the whole data structure.

<sup>2</sup> *VariableTypes* other than the *PropertyType* which is used for all *Properties*.

The third approach combines the other two approaches. Therefore a specific *Client* can access data in its native format in a transactional context, whereas a generic *Client* can access simple *DataTypes* of the components of the complex *Variable*. The disadvantage is that the *Server* must be able to provide the native format and also interpret it to be able to provide the information in simple *DataTypes*.

It is recommended to use the first approach. When a transactional context is needed or the *Client* should be able to get a large amount of data instead of subscribing to several individual values, then the third approach is suitable. However, the *Server* might not always have the knowledge to interpret the structured data of the underlying system and therefore has to use the second approach just passing the data to the specific *Client* who is able to interpret the data.

## A.5 Views

*Server*-defined *Views* can be used to present an excerpt of the *AddressSpace* suitable for a special class of *Clients*, for example maintenance *Clients*, engineering *Clients*, etc. The *View* only provides the information needed for the purpose of the *Client* and hides unnecessary information.

## A.6 Methods

*Methods* should be used whenever some input is expected and the *Server* delivers a result. One should avoid using *Variables* to write the input values and other *Variables* to get the output results as it was necessary to do in OPC COM since there was no concept of a *Method* available. However, a simple OPC COM wrapper might not be able to do this.

*Methods* can also be used to trigger some execution in the *Server* that does not require input and / or output parameters.

Global *Methods*, that is, *Methods* that cannot directly be assigned to a special *Object*, should be assigned to the *Server Object* defined in IEC 62541-5.

## A.7 Defining ReferenceTypes

Defining new *ReferenceTypes* should only be done if the predefined *ReferenceTypes* are not suitable. Whenever a new *ReferenceType* is defined, the most appropriate *ReferenceType* should be used as its supertype.

It is expected that *Servers* will have new defined hierarchical *ReferenceTypes* to expose different hierarchies, and new non-hierarchical *References* to expose relationships between *Nodes* in the *AddressSpace*.

## A.8 Defining ModellingRules

New *ModellingRules* have to be defined if the predefined *ModellingRules* are not appropriate for the model exposed by the *Server*.

Depending on the model used by the underlying system the *Server* may need to define new *ModellingRules*, since the OPC UA *Server* may only pass the data to the underlying system and this system may use its own internal rules for instantiation, subtyping, etc.

Beside this, the predefined *ModellingRules* might not be sufficient to specify the required behaviour for instantiation and subtyping.

## Annex B (informative)

### OPC UA Meta Model in UML

#### B.1 Background

The OPC UA Meta Model (the OPC UA Address Space Model) is represented by UML classes and UML objects marked with the stereotype <<TypeExtension>>. Those stereotyped UML objects represent *DataTypes* or *ReferenceTypes*. The domain model can contain user-defined *ReferenceTypes* and *DataTypes*, also marked as <<TypeExtension>>. In addition, the domain model contains *ObjectTypes*, *VariableTypes* etc. represented as UML objects (see Figure B.1).

The OPC Foundation specifies not only the OPC UA Meta Model, but also defines some *Nodes* to organise the *AddressSpace* and to provide information about the *Server* as specified in IEC 62541-5.

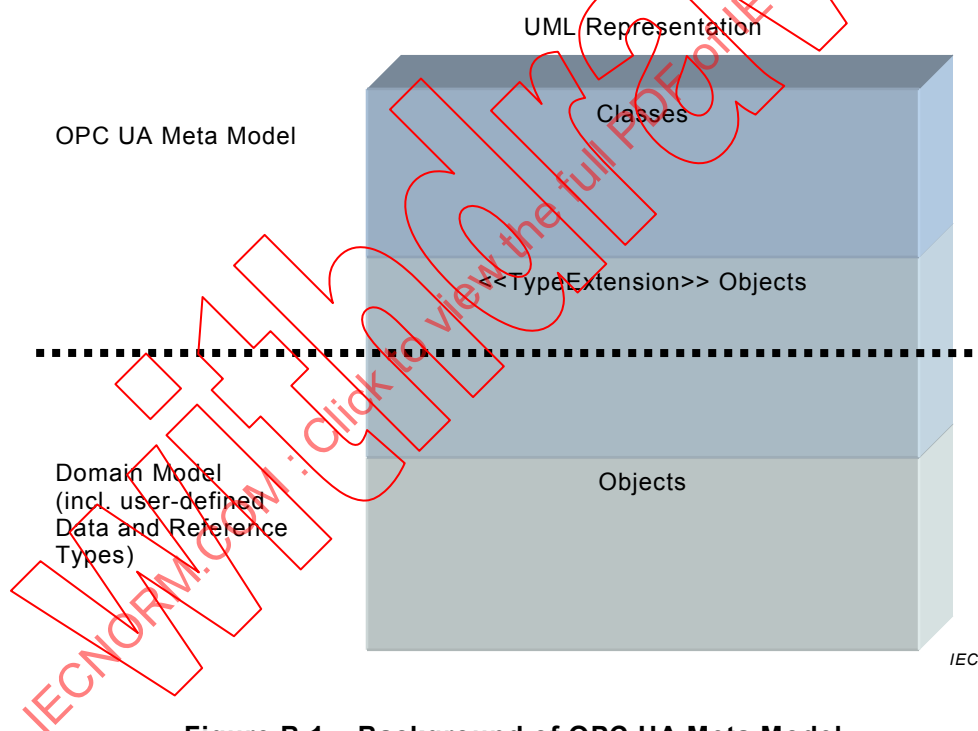


Figure B.1 – Background of OPC UA Meta Model

#### B.2 Notation

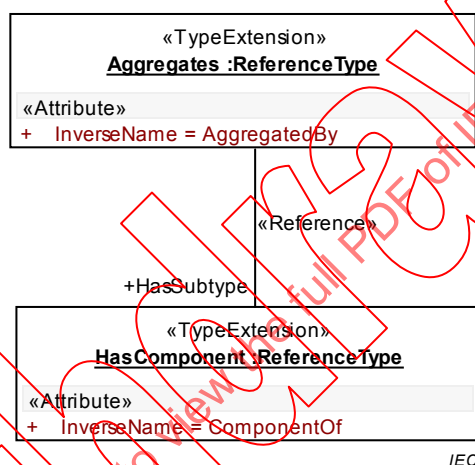
An example of a UML class representing the OPC UA concept *Base* is given in the UML class diagram in Figure B.2. OPC Attributes inherit from the abstract class *Attribute* and have a value identifying their data type. They are composed of a *Node* which is either optional (0..1) or required (1), such as *BrowseName* to *Base* in Figure B.2.



IEC

Figure B.2 – Notation (I)

UML object diagrams are used to display <<TypeExtension>> objects (e.g. *HasComponent* in Figure B.3). In object diagrams, OPC *Attributes* are represented as UML attributes without data types and marked with the stereotype <<Attribute>>, like *InverseName* in the UML object *HasComponent*. They have values, like *InverseName = ComponentOf* for *HasComponent*. To keep the object diagrams simple, not all *Attributes* are shown (e.g. the *NodeId* of *HasComponent*).



IEC

Figure B.3 – Notation (II)

OPC *References* are represented as UML associations marked with the stereotype <<Reference>>. If a particular *ReferenceType* is used, its name is used as the role name, identifying the direction of the *Reference* (e.g. *Aggregates* has the subtype *HasComponent*). For simplicity, the inverse role name is not shown (in the example *SubtypeOf*). When no role name is provided, it means that any *ReferenceType* can be used (only valid for class diagrams).

There are some special *Attributes* in OPC UA containing a *NodeId* and thereby referencing another *Node*. Those *Attributes* are represented as associations marked with the stereotype <<Attribute>>. The name of the *Attribute* is displayed as the role name of the *TargetNode*.

The value of the OPC *Attribute BrowseName* is represented by the UML object name, for example the *BrowseName* of the UML object *HasComponent* in Figure B.3 is "HasComponent".

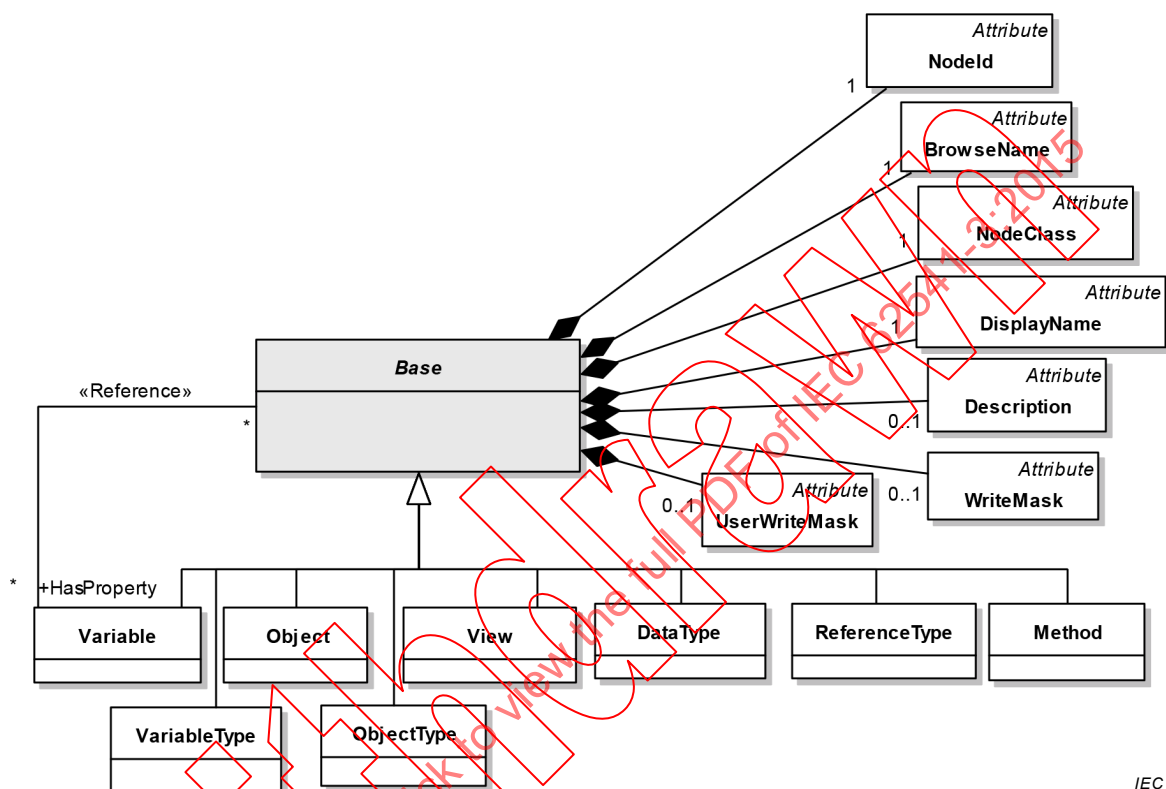
To highlight the classes explained in a class diagram, they are marked in grey (e.g. *Base* in Figure B.2). Only those classes have all of their relationships to other classes and attributes shown in the diagram. For the other classes, we provide only those attributes and relationships needed to understand the main classes of the diagram.

## B.3 Meta Model

NOTE Other parts of the IEC 62541 series can extend the OPC UA Meta Model by adding *Attributes* and defining new *ReferenceTypes*.

### B.3.1 Base

Base is shown in Figure B.4.

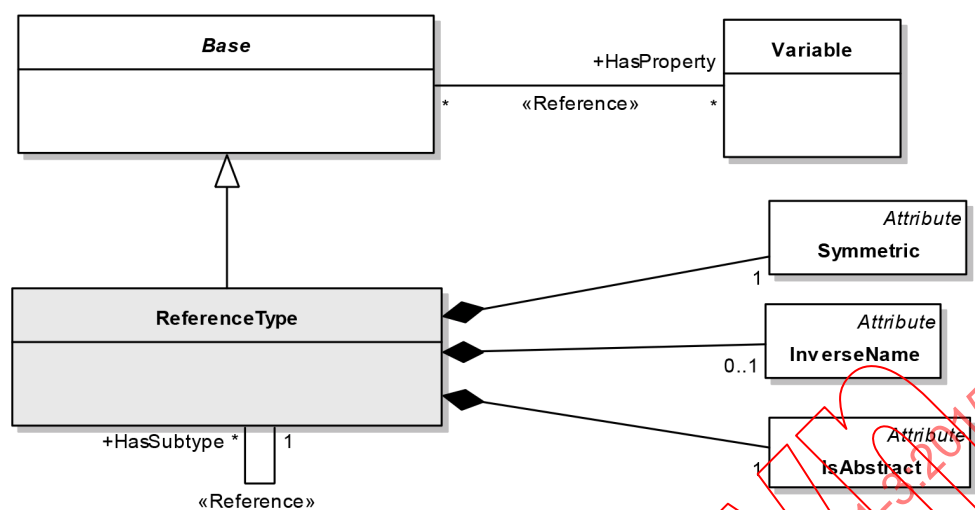


IEC

Figure B.4 – Base

### B.3.2 ReferenceType

*ReferenceType* is shown in Figure B.5 and predefined *ReferenceTypes* in Figure B.6.



**Figure B.5 – Reference and ReferenceType**

If *Symmetric* is “false” and *IsAbstract* is “false” an *InverseName* shall be provided.

### B.3.3 Predefined ReferenceTypes

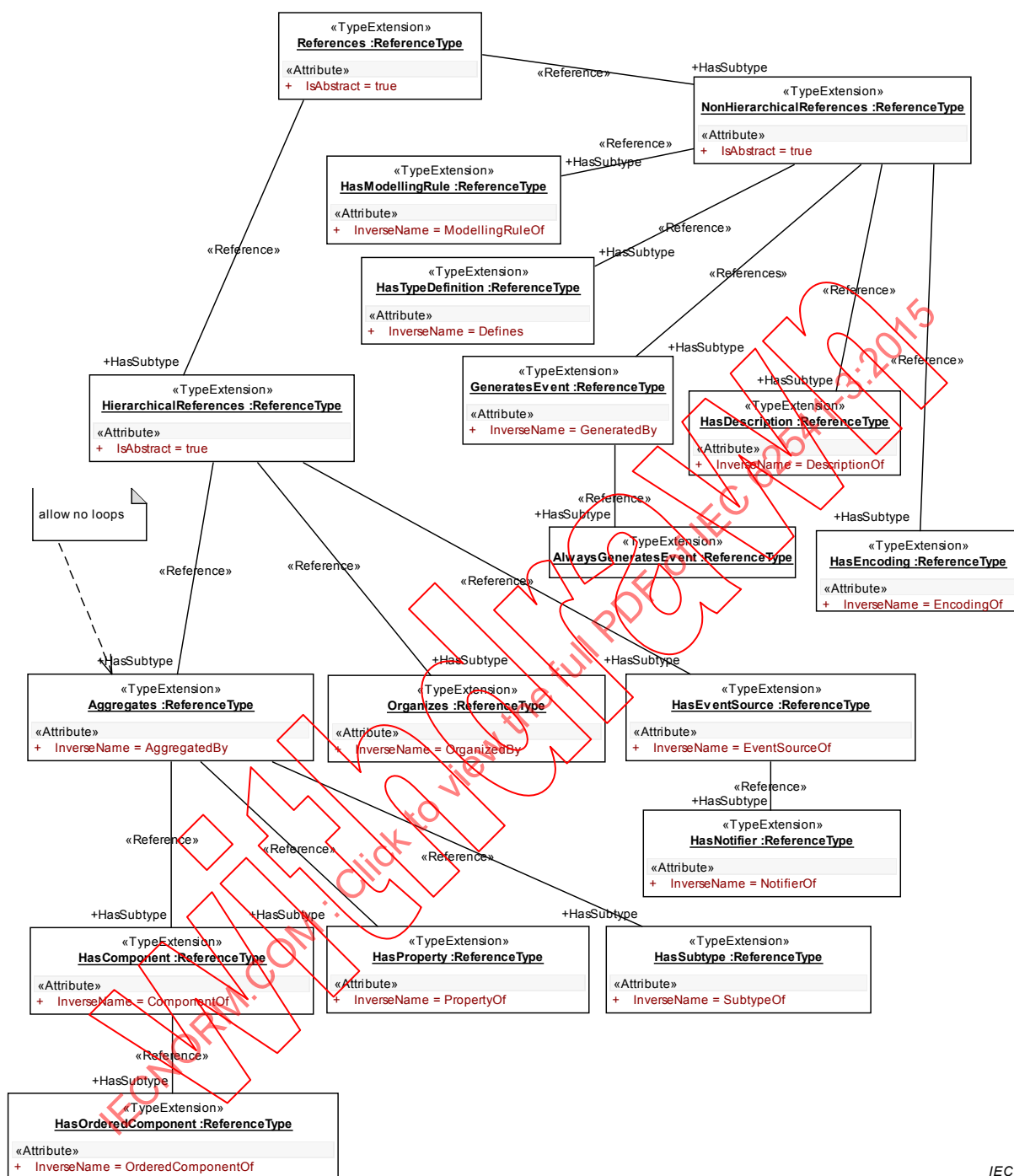


Figure B.6 – Predefined ReferenceTypes

### B.3.4 Attributes

Attributes are shown in Figure B.7.

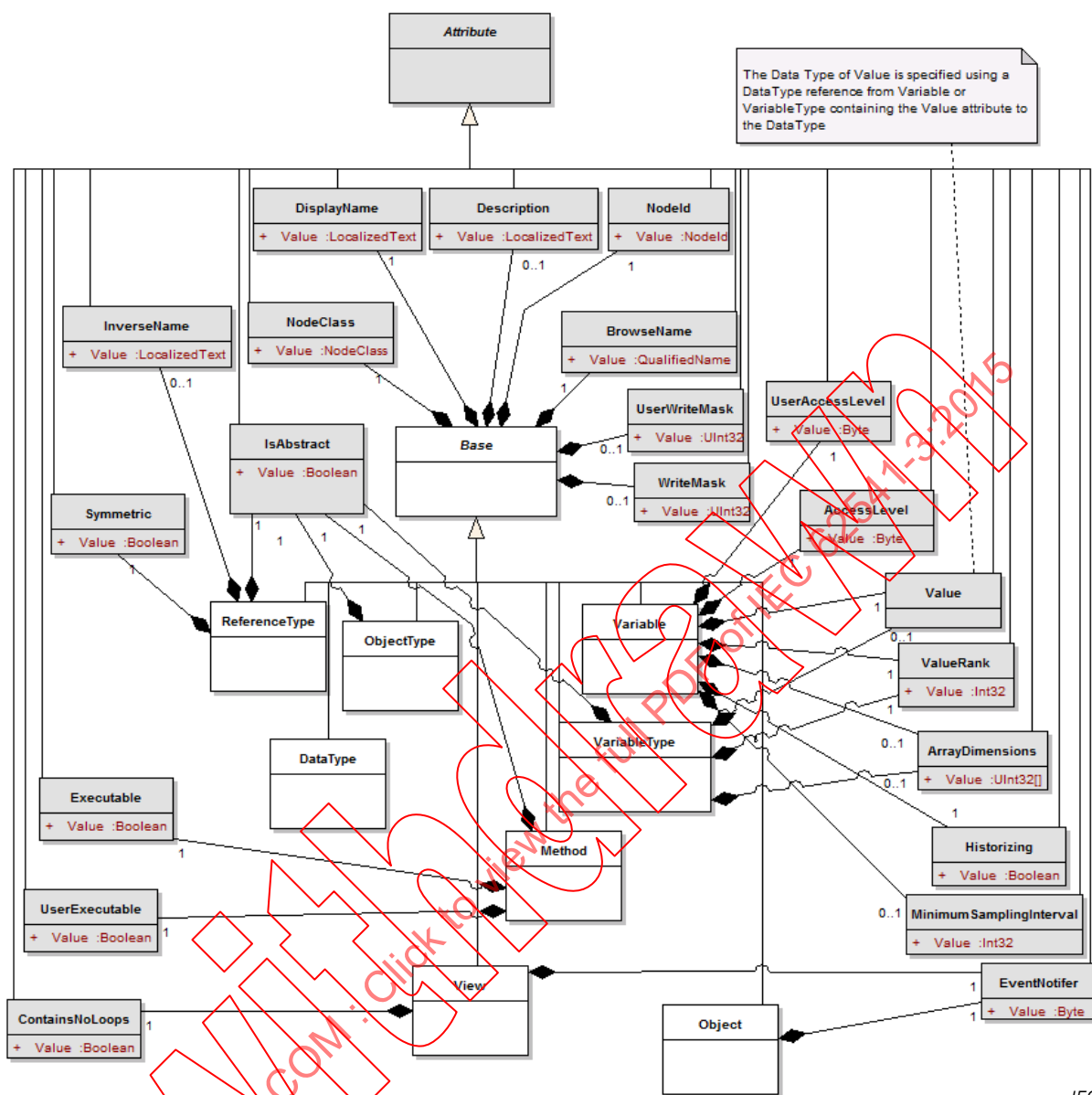


Figure B.7 – Attributes

There may be more *Attributes* defined in other parts of the IEC 62541 series.

*Attributes* used for references, which have a *NodeId* as *DataType*, are not shown in this diagram but are shown as stereotyped associations in the other diagrams.

### B.3.5 Object and ObjectType

*Objects* and *ObjectTypes* are shown in Figure B.8.



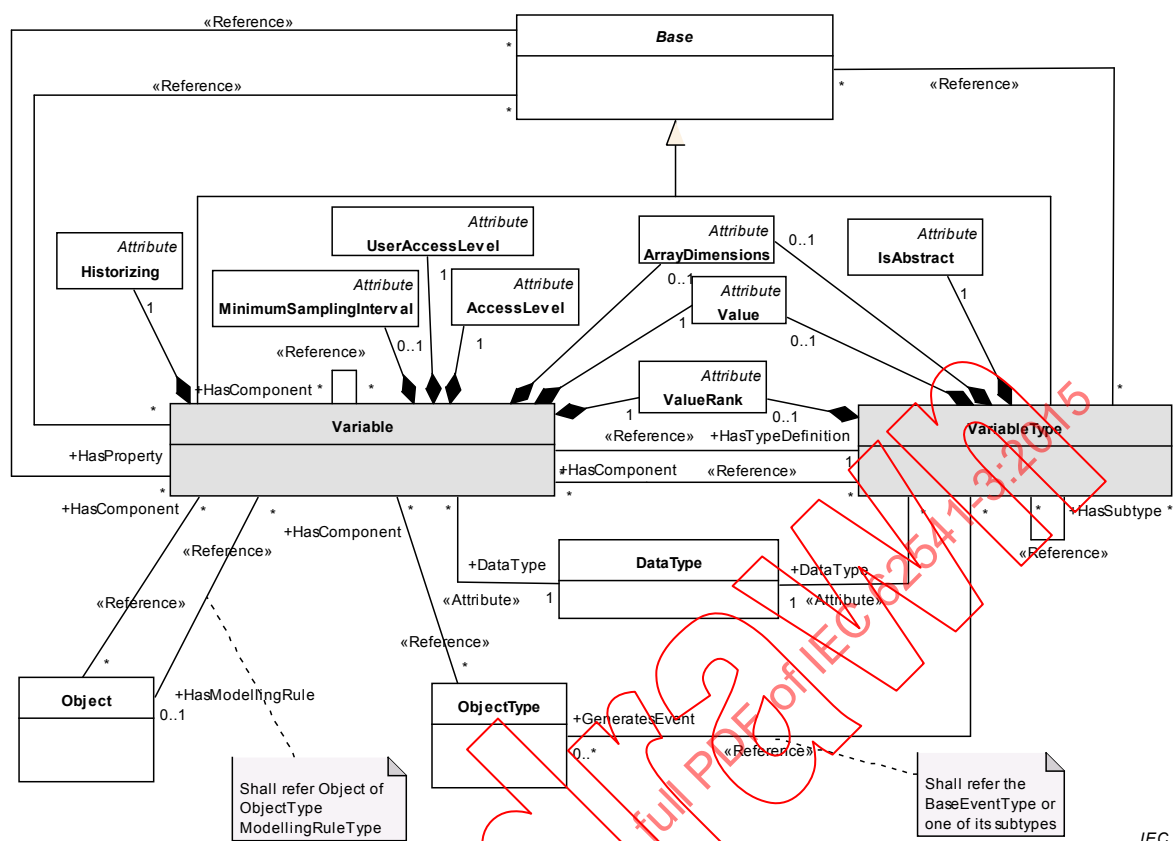
### B.3.6 EventNotifier

```
classDiagram
    class Base {
    }
    class Object {
    }
    Base "1" --> "1" Object : «Reference»
    class Ref1 {
    }
    Ref1 --> Object : «Reference»
    Ref1 --> Base : +HasEventSource
```

The diagram illustrates a reference relationship between two classes, **Base** and **Object**. The **Base** class is represented by a white rectangle with the name **Base** in the top compartment. The **Object** class is represented by a gray rectangle with the name **Object** in the top compartment. A directed association arrow points from **Object** to **Base**, labeled with the stereotype «Reference» in the middle. To the left of the **Object** class, there is a small white rectangle representing a reference object. This object has two outgoing directed association arrows: one pointing to the **Object** class (labeled «Reference») and another pointing to the **Base** class (labeled +HasEventSource).

### B.3.7 Variable and VariableType

Variable and VariableType are shown in Figure B.10.



**Figure B.10 – Variable and VariableType**

The *DataType* of a *Variable* shall be the same as or a subtype of the *DataType* of its *VariableType* (referred with *HasTypeDefinition*).

If a *HasProperty* points to a *Variable* from a Base “A” then the following constraints apply:

- The *Variable* shall not be the *SourceNode* of a *HasProperty* or any other *HierarchicalReferences* Reference.
- All *Variables* having “A” as the *SourceNode* of a *HasProperty* Reference shall have a unique *BrowseName* in the context of “A”.

### B.3.8 Method

Method is shown in Figure B.11

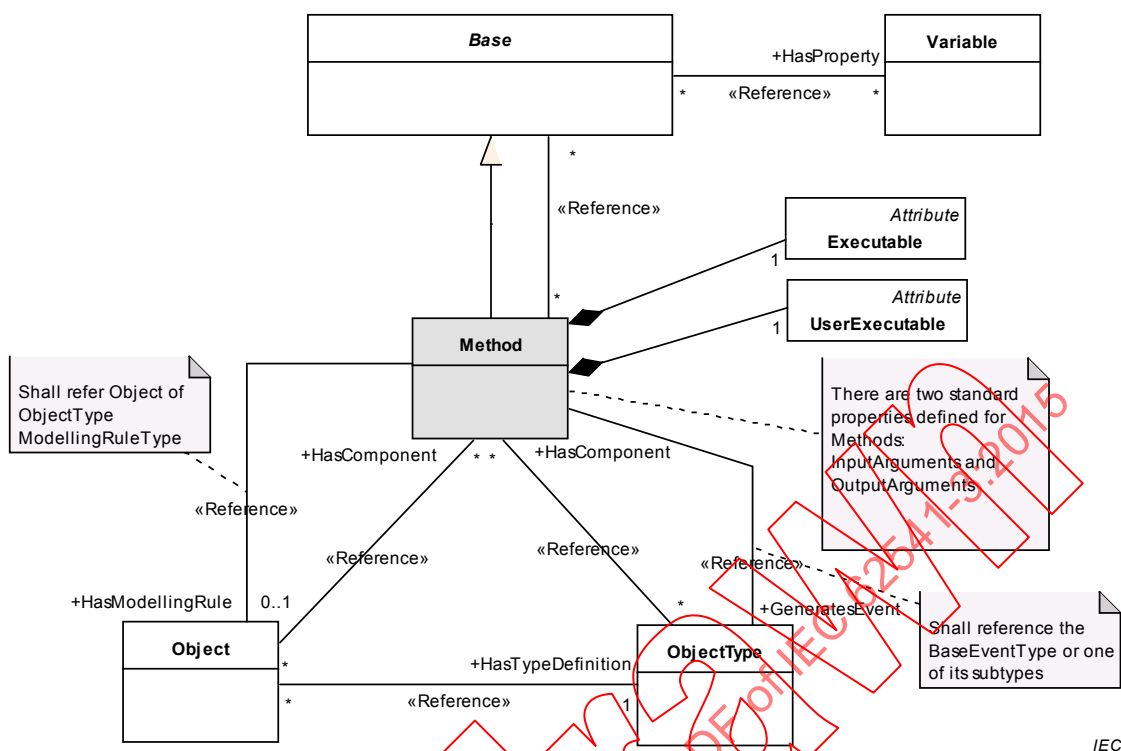


Figure B.11 – Method

### B.3.9 DataType

DataType is shown in Figure B.12.

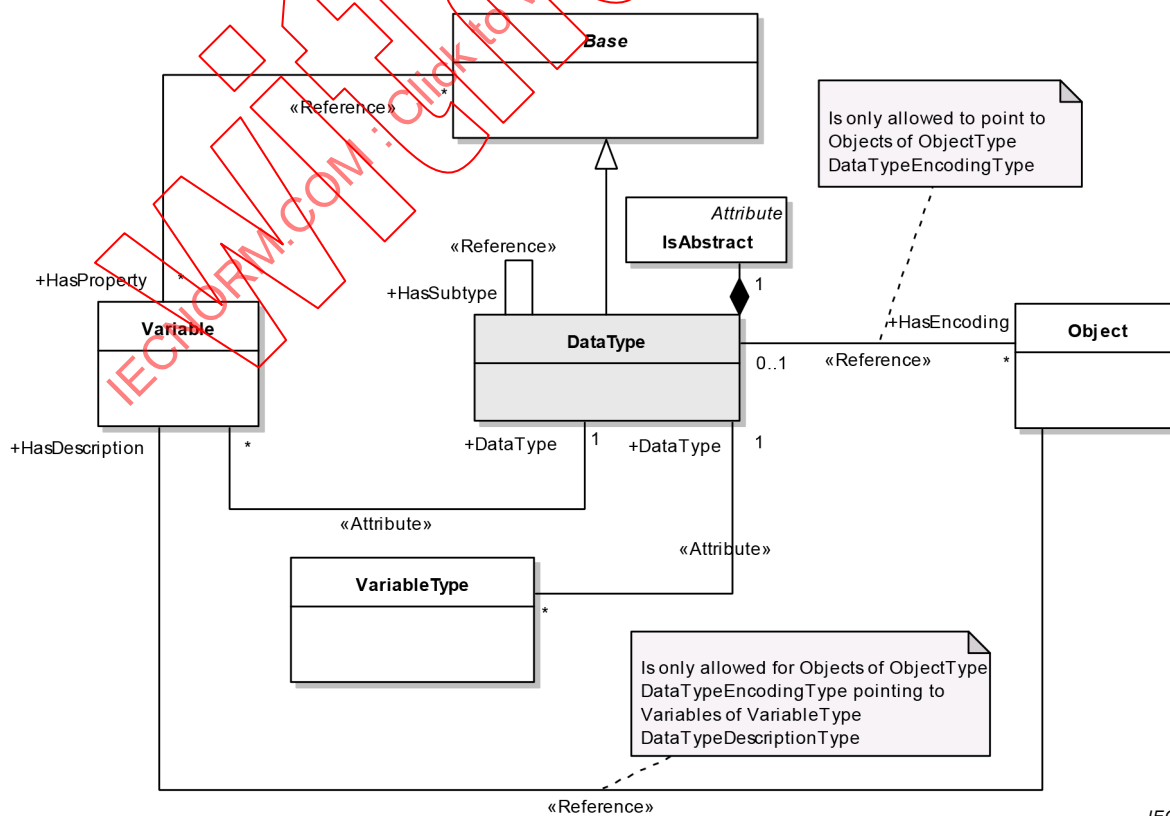


Figure B.12 – DataType

**B.3.10 View**

View is shown in Figure B.13.

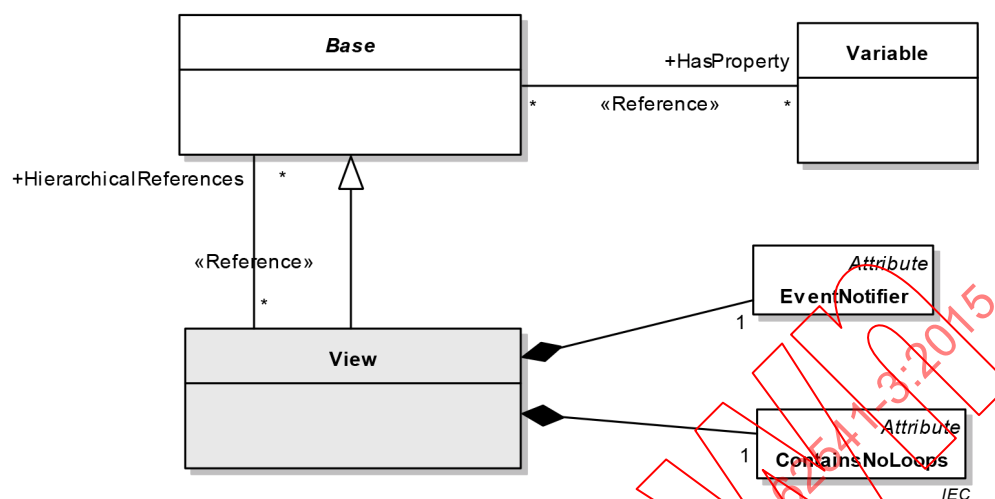


Figure B.13 – View

## Annex C (normative)

### OPC Binary Type Description System

#### C.1 Concepts

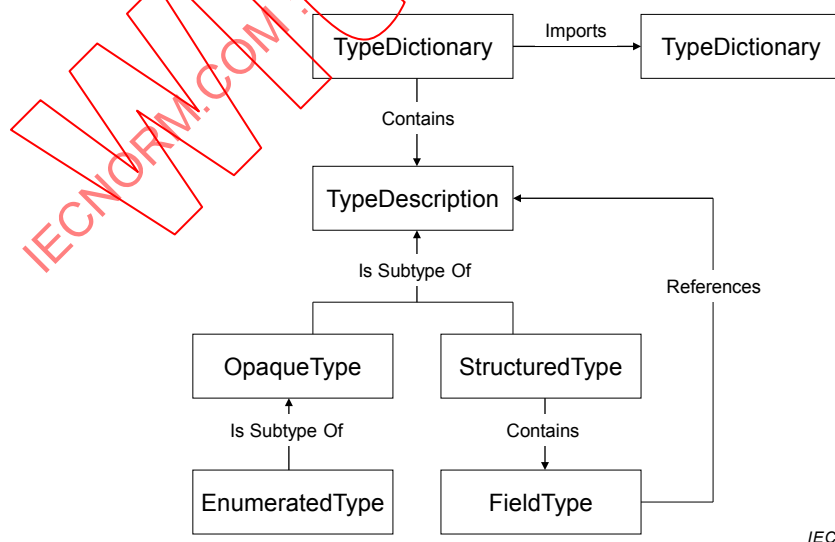
The OPC Binary XML Schema defines the format of OPC Binary *TypeDictionaries*. Each OPC Binary *TypeDictionary* is an XML document that contains one or more *TypeDescriptions* that describe the format of a binary-encoded value. Applications that have no advanced knowledge of a particular binary encoding can use the OPC Binary *TypeDescription* to interpret or construct a value.

The OPC Binary Type Description System does not define a standard mechanism to *encode* data in binary. It only provides a standard way to describe an existing binary encoding. Many binary encodings will have a mechanism to describe types that could be encoded; however, these descriptions are useful only to applications that have knowledge of the type description system used with each binary encoding. The OPC Binary Type Description System is a generic syntax that can be used by any application to interpret any binary encoding.

The OPC Binary Type Description System was originally defined in the OPC Complex Data Specification. The OPC Binary Type Description System described in Annex C is quite different and is correctly described as the OPC Binary Type Description System Version 2.0.

Each *TypeDescription* is identified by a *TypeName* which shall be unique within the *TypeDictionary* that defines it. Each *TypeDictionary* also has a *TargetNamespace* which should be unique among all OPC Binary *TypeDictionaries*. This means that the *TypeName* qualified with the *TargetNamespace* for the dictionary should be a globally-unique identifier for a *TypeDescription*.

Figure C.1 below illustrates the structure of an OPC Binary *TypeDictionary*.



IEC

Figure C.1 – OPC Binary Dictionary Structure

Each binary encoding is built from a set of opaque building blocks that are either primitive types with a fixed length or variable-length types with a structure that is too complex to

describe properly in an XML document. These building blocks are described with an *OpaqueType*. An instance of one of these building blocks is a binary-encoded value.

The OPC Binary Type Description System defines a set of standard *OpaqueTypes* that all OPC Binary *TypeDictionaries* should use to build their *TypeDescriptions*. These standard type descriptions are described in Clause C.3.

In some cases, the binary encoding described by an *OpaqueType* may have a fixed size which would allow an application to skip an encoded value that it does not understand. If that is the case, then the *LengthInBits* attribute should be specified for the *OpaqueType*. If authors of *TypeDictionaries* need to define new *OpaqueTypes* that do not have a fixed size then they should use the documentation elements to describe how to encode binary values for the type. This description should provide enough detail to allow a human to write a program that can interpret instances of the type.

A *StructuredType* breaks a complex value into a sequence of values that are described by a *FieldType*. Each *FieldType* has a name, type and a number of qualifiers that specify when the field is used and how many instances of the type exist. A *FieldType* is described completely in C.2.6.

An *EnumeratedType* describes a numeric value that has a limited set of possible values, each of which has a descriptive name. *EnumeratedTypes* provide a convenient way to capture semantic information associated with what would otherwise be an opaque numeric value.

## C.2 Schema Description

### C.2.1 TypeDictionary

The *TypeDictionary* element is the root element of an OPC Binary Dictionary. The components of this element are described in Table C.1.

**Table C.1 – TypeDictionary Components**

| Name             | Type              | Description                                                                                                                                                                                                                                                                                                                                       |
|------------------|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Documentation    | Documentation     | An element that contains human-readable text and XML that provides an overview of what is contained in the dictionary.                                                                                                                                                                                                                            |
| Import           | ImportDirective[] | Zero or more elements that specify other <i>TypeDictionaries</i> that are referenced by <i>StructuredTypes</i> defined in the dictionary. Each import element specifies the <i>NamespaceUri</i> of the <i>TypeDictionary</i> being imported. The <i>TypeDictionary</i> element shall declare an XML namespace prefix for each imported namespace. |
| TargetNamespace  | xs:string         | Specifies the URI that qualifies all <i>TypeDescriptions</i> defined in the dictionary.                                                                                                                                                                                                                                                           |
| DefaultByteOrder | ByteOrder         | Specifies the default <i>ByteOrder</i> for all <i>TypeDescriptions</i> that have the <i>ByteOrderSignificant</i> attribute set to "true".<br>This value overrides the setting in any imported <i>TypeDictionary</i> .<br>This value is overridden by the <i>DefaultByteOrder</i> specified on a <i>TypeDescription</i> .                          |
| TypeDescription  | TypeDescription[] | One or more elements that describe the structure of a binary encoded value.<br>A <i>TypeDescription</i> is an abstract type. A dictionary may only contain the <i>OpaqueType</i> , <i>EnumeratedType</i> and <i>StructuredType</i> elements.                                                                                                      |

### C.2.2 TypeDescription

A *TypeDescription* describes the structure of a binary encoded value. A *TypeDescription* is an abstract base type and only instances of subtypes may appear in a *TypeDictionary*. The components of a *TypeDescription* are described in Table C.2.

**Table C.2 – TypeDescription Components**

| Name             | Type          | Description                                                                                                                                                                                                                                                                                   |
|------------------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Documentation    | Documentation | An element that contains human readable text and XML that describes the type. This element should capture any semantic information that would help a human to understand what is contained in the value.                                                                                      |
| Name             | xs:NCName     | An attribute that specifies a name for the <i>TypeDescription</i> that is unique within the dictionary. The fields of structured types reference <i>TypeDescriptions</i> by using this name qualified with the dictionary namespace URI.                                                      |
| DefaultByteOrder | ByteOrder     | An attribute that specifies the default <i>ByteOrder</i> for the type description. This value overrides the setting in any <i>TypeDictionary</i> or in any <i>StructuredType</i> that references the type description.                                                                        |
| anyAttribute     | *             | Authors of a <i>TypeDictionary</i> may add their own attributes to any <i>TypeDescription</i> that shall be qualified with a namespace defined by the author. Applications should not be required to understand these attributes in order to interpret a binary encoded instance of the type. |

### C.2.3 OpaqueType

An *OpaqueType* describes a binary encoded value that is either a primitive fixed length type or that has a structure too complex to capture in an OPC Binary type dictionary. Authors of type dictionaries should avoid defining *OpaqueTypes* that do not have a fixed length because it would prevent applications from interpreting values that use these types without having built-in knowledge of the *OpaqueType*. The OPC Binary Type Description System defines many standard *OpaqueTypes* that should allow authors to describe most binary encoded values as *StructuredTypes*.

The components of an *OpaqueType* are described in Table C.3.

**Table C.3 – OpaqueType Components**

| Name                 | Type            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TypeDescription      | TypeDescription | An <i>OpaqueType</i> inherits all elements and attributes defined for a <i>TypeDescription</i> in Table C.2.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| LengthInBits         | xs:string       | An attribute which specifies the length of the <i>OpaqueType</i> in bits. This value should always be specified. If this value is not specified the <i>Documentation</i> element should describe the encoding in a way that a human understands.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| ByteOrderSignificant | xs:boolean      | An attribute that indicates whether byte order is significant for the type.<br>If byte order is significant then the application shall determine the byte order to use for the current context before interpreting the encoded value. The application determines the byte order by looking for the <i>DefaultByteOrder</i> attribute specified for containing <i>StructuredTypes</i> or the <i>TypeDictionary</i> . If <i>StructuredTypes</i> are nested the inner <i>StructuredTypes</i> override the byte order of the outer descriptions.<br>If the <i>DefaultByteOrder</i> attribute is specified for the <i>OpaqueType</i> , then the <i>ByteOrder</i> is fixed and does not change according to context.<br>If this attribute is “true”, then the <i>LengthInBits</i> attribute shall be specified and it shall be an integer multiple of 8 bits. |

### C.2.4 EnumeratedType

An *EnumeratedType* describes a binary-encoded numeric value that has a fixed set of valid values. The encoded binary value described by an *EnumeratedType* is always an unsigned integer with a length specified by the *LengthInBits* attribute.

The names for each of the enumerated values are not required to interpret the binary encoding, however, they form part of the documentation for the type.

The components of an *EnumeratedType* are described in Table C.4.

**Table C.4 – EnumeratedType Components**

| Name            | Type                  | Description                                                                                                                                                                                                                        |
|-----------------|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OpaqueType      | OpaqueTypeDescription | An <i>EnumeratedType</i> inherits all elements and attributes defined for a <i>TypeDescription</i> in Table C.2 and for an <i>OpaqueType</i> defined in Table C.3.<br>The <i>LengthInBits</i> attribute shall always be specified. |
| EnumeratedValue | EnumeratedValue       | One or more elements that describe the possible values for the instances of the type.                                                                                                                                              |

### C.2.5 StructuredType

A *StructuredType* describes a type as a sequence of binary-encoded values. Each value in the sequence is called a *Field*. Each *Field* references a *TypeDescription* that describes the binary-encoded value that appears in the field. A *Field* may specify that zero, one or multiple instances of the type appear within the sequence described by the *StructuredType*.

Authors of type dictionaries should use *StructuredTypes* to describe a variety of common data constructs including arrays, unions and structures.

Some fields have lengths that are not multiples of 8 bits. Several of these fields may appear in a sequence in a structure, however, the total number of bits used in the sequence shall be fixed and it shall be a multiple of 8 bits. Any field which does not have a fixed length shall be aligned on a byte boundary.

A sequence of fields which do not line up on byte boundaries are specified from the least significant bit to the most significant bit. Sequences which are longer than one byte overflow from the most significant bit of the first byte into the least significant bit of the next byte.

The components of a *StructuredType* are described in Table C.5.

**Table C.5 – StructuredType Components**

| Name            | Type            | Description                                                                                                                                                                                                                            |
|-----------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TypeDescription | TypeDescription | A <i>StructuredType</i> inherits all elements and attributes defined for a <i>TypeDescription</i> in Table C.2.                                                                                                                        |
| Field           | FieldType       | One or more elements that describe the fields of the structure. Each field shall have a name that is unique within the <i>StructuredType</i> . Some fields may reference other fields in the <i>StructuredType</i> by using this name. |

### C.2.6 FieldType

A *FieldType* describes a binary encoded value that appears in sequence within a *StructuredType*. Every *FieldType* shall reference a *TypeDescription* that describes the encoded value for the field.

A *FieldType* may specify an array of encoded values.

*Fields* may be optional and they reference other *FieldTypes*, which indicate if they are present in any specific instance of the type.

The components of a *FieldType* are described in Table C.6.

Table C.6 – FieldType Components

| Name            | Type           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Documentation   | Documentation  | An element that contains human readable text and XML that describes the field. This element should capture any semantic information that would help a human to understand what is contained in the field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Name            | xs:string      | An attribute that specifies a name for the <i>Field</i> that is unique within the <i>StructuredType</i> .<br>Other fields in the structured type reference a <i>Field</i> by using this name.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| TypeName        | xs:QName       | An attribute that specifies the <i>TypeDescription</i> that describes the contents of the field. A field may contain zero or more instances of this type depending on the settings for the other attributes and the values in other fields.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Length          | xs:unsignedInt | An attribute that indicates the length of the field. This value may be the total number of encoded bytes or it may be the number of instances of the type referenced by the field. The <i>IsLengthInBytes</i> attributes specifies which of these definitions applies.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| LengthField     | xs:string      | An attribute that indicates which other field in the <i>StructuredType</i> specifies the length of the field. The length of the field may be in bytes or it may be the number of instances of the type referenced by the field. The <i>IsLengthInBytes</i> attributes specify which of these definitions applies.<br>If this attribute refers to a field that is not present in an encoded value, then the default value for the length is 1. This situation could occur if the field referenced is an optional field (see the <i>SwitchField</i> attribute).<br>The length field shall be a fixed length Base-2 representation of an integer. If the length field is one of the standard signed integer types and the value is a negative integer, then the field is not present in the encoded stream.<br>The <i>FieldType</i> referenced by this attribute shall precede the field with the <i>StructuredType</i> .                                                                                                                                                                                                                                                                                                                                  |
| IsLengthInBytes | xs:boolean     | An attribute that indicates whether the <i>Length</i> or <i>LengthField</i> attributes specify the length of the field in bytes or in the number of instances of the type referenced by the field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| SwitchField     | xs:string      | If this attribute is specified, then the field is optional and may not appear in every instance of the encoded value.<br>This attribute specifies the name of another <i>Field</i> that controls whether this field is present in the encoded value. The field referenced by this attribute shall be an integer value (see the <i>LengthField</i> attribute).<br>The current value of the switch field is compared to the <i>SwitchValue</i> attribute using the <i>SwitchOperand</i> . If the condition evaluates to true then the field appears in the stream.<br>If the <i>SwitchValue</i> attribute is not specified, then this field is present if the value of the switch field is non-zero. The <i>SwitchOperand</i> field is ignored if it is present.<br>If the <i>SwitchOperand</i> attribute is missing, then the field is present if the value of the switch field is equal to the value of the <i>SwitchValue</i> attribute.<br>The <i>Field</i> referenced by this attribute shall precede the field with the <i>StructuredType</i> .                                                                                                                                                                                                     |
| SwitchValue     | xs:unsignedInt | This attribute specifies when the field appears in the encoded value. The value of the field referenced by the <i>SwitchField</i> attribute is compared using the <i>SwitchOperand</i> attribute to this value. The field is present if the expression evaluates to true. The field is not present otherwise.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| SwitchOperand   | xs:string      | This attribute specifies how the value of the switch field should be compared to the switch value attribute. This field is an enumeration with the following values:<br><div style="display: flex; justify-content: space-between;"> <div>Equal</div> <div><i>SwitchField</i> is equal to the <i>SwitchValue</i>.</div> </div> <div style="display: flex; justify-content: space-between;"> <div>GreaterThan</div> <div><i>SwitchField</i> is greater than the <i>SwitchValue</i>.</div> </div> <div style="display: flex; justify-content: space-between;"> <div>LessThan</div> <div><i>SwitchField</i> is less than the <i>SwitchValue</i>.</div> </div> <div style="display: flex; justify-content: space-between;"> <div>GreaterThanOrEqual</div> <div><i>SwitchField</i> is greater than or equal to the <i>SwitchValue</i>.</div> </div> <div style="display: flex; justify-content: space-between;"> <div>LessThanOrEqual</div> <div><i>SwitchField</i> is less than or equal to the <i>SwitchValue</i>.</div> </div> <div style="display: flex; justify-content: space-between;"> <div>NotEqual</div> <div><i>SwitchField</i> is not equal to the <i>SwitchValue</i>.</div> </div> In each case the field is present if the expression is true. |
| Terminator      | xs:hexBinary   | This attribute indicates that the field contains one or more instances of <i>TypeDescription</i> referenced by this field and that the last value has the binary encoding specified by the value of this attribute.<br>If this attribute is specified then the <i>TypeDescription</i> referenced by this field shall either have a fixed byte order (i.e. byte order is not significant or explicitly specified) or the containing <i>StructuredType</i> shall explicitly specify the byte order.<br>Examples:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

| Name            | Type          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                    |            |            |                    |      |               |                |    |           |               |           |      |           |               |              |      |       |   |           |      |       |   |              |      |
|-----------------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|------------|------------|--------------------|------|---------------|----------------|----|-----------|---------------|-----------|------|-----------|---------------|--------------|------|-------|---|-----------|------|-------|---|--------------|------|
|                 |               | <table><tr><th>Field Data Type</th><th>Terminator</th><th>Byte Order</th><th>Hexadecimal String</th></tr><tr><td>Char</td><td>tab character</td><td>not applicable</td><td>09</td></tr><tr><td>WideChar:</td><td>tab character</td><td>BigEndian</td><td>0009</td></tr><tr><td>WideChar:</td><td>tab character</td><td>LittleEndian</td><td>0900</td></tr><tr><td>Int16</td><td>1</td><td>BigEndian</td><td>0001</td></tr><tr><td>Int16</td><td>1</td><td>LittleEndian</td><td>0100</td></tr></table> | Field Data Type    | Terminator | Byte Order | Hexadecimal String | Char | tab character | not applicable | 09 | WideChar: | tab character | BigEndian | 0009 | WideChar: | tab character | LittleEndian | 0900 | Int16 | 1 | BigEndian | 0001 | Int16 | 1 | LittleEndian | 0100 |
| Field Data Type | Terminator    | Byte Order                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Hexadecimal String |            |            |                    |      |               |                |    |           |               |           |      |           |               |              |      |       |   |           |      |       |   |              |      |
| Char            | tab character | not applicable                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 09                 |            |            |                    |      |               |                |    |           |               |           |      |           |               |              |      |       |   |           |      |       |   |              |      |
| WideChar:       | tab character | BigEndian                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 0009               |            |            |                    |      |               |                |    |           |               |           |      |           |               |              |      |       |   |           |      |       |   |              |      |
| WideChar:       | tab character | LittleEndian                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0900               |            |            |                    |      |               |                |    |           |               |           |      |           |               |              |      |       |   |           |      |       |   |              |      |
| Int16           | 1             | BigEndian                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 0001               |            |            |                    |      |               |                |    |           |               |           |      |           |               |              |      |       |   |           |      |       |   |              |      |
| Int16           | 1             | LittleEndian                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0100               |            |            |                    |      |               |                |    |           |               |           |      |           |               |              |      |       |   |           |      |       |   |              |      |
| anyAttribute    | *             | Authors of a <i>TypeDictionary</i> may add their own attributes to any <i>FieldType</i> which shall be qualified with a namespace defined by the authors. Applications should not be required to understand these attributes in order to interpret a binary encoded field value.                                                                                                                                                                                                                      |                    |            |            |                    |      |               |                |    |           |               |           |      |           |               |              |      |       |   |           |      |       |   |              |      |

### C.2.7 EnumeratedValue

An *EnumeratedValue* describes a possible value for an *EnumeratedType*.

The components of an *EnumeratedValue* are described in Table C.7.

**Table C.7 – EnumeratedValue Components**

| Name  | Type           | Description                                                                          |
|-------|----------------|--------------------------------------------------------------------------------------|
| Name  | xs:string      | This attribute specifies a descriptive name for the enumerated value.                |
| Value | xs:unsignedInt | This attribute specifies the numeric value that could appear in the binary encoding. |

### C.2.8 ByteOrder

A *ByteOrder* is an enumeration that describes a possible value byte orders for *TypeDescriptions* that allow different byte orders to be used. There are two possible values: *BigEndian* and *LittleEndian*. *BigEndian* indicates the most significant byte appears first in the binary encoding. *LittleEndian* indicates that the least significant byte appears first.

### C.2.9 ImportDirective

An *ImportDirective* specifies a *TypeDictionary* that is referenced by types defined in the current dictionary.

The components of an *ImportDirective* are described in Table C.8.

**Table C.8 – ImportDirective Components**

| Name      | Type      | Description                                                                                                                                                                                                                               |
|-----------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Namespace | xs:string | This attribute specifies the <i>TargetNamespace</i> for the <i>TypeDictionary</i> being imported. This may be a well-known URI which means applications need not have access to the physical file to recognise types that are referenced. |
| Location  | xs:string | This attribute specifies the physical location of the XML file containing the <i>TypeDictionary</i> to import. This value could be a URL for a network resource, a NodeId in an OPC UA <i>Server</i> address space or a local file path.  |

## C.3 Standard Type Descriptions

The OPC Binary Type Description System defines a number of standard type descriptions that can be used to describe many common binary encodings using a *StructuredType*. The standard type descriptions are described in Table C.9.

**Table C.9 – Standard Type Descriptions**

| Type name     | Description                                                                                                                                          |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bit           | A single bit value.                                                                                                                                  |
| Boolean       | A two-state logical value represented as an 8-bit value.                                                                                             |
| SByte         | An 8-bit signed integer.                                                                                                                             |
| Byte          | An 8-bit unsigned integer.                                                                                                                           |
| Int16         | A 16-bit signed integer.                                                                                                                             |
| UInt16        | A 16-bit unsigned integer.                                                                                                                           |
| Int32         | A 32-bit signed integer.                                                                                                                             |
| UInt32        | A 32-bit unsigned integer.                                                                                                                           |
| Int64         | A 64-bit signed integer.                                                                                                                             |
| UInt64        | A 64-bit unsigned integer.                                                                                                                           |
| Float         | An IEEE 754-1985 single precision floating point value.                                                                                              |
| Double        | An IEEE 754-1985 double precision floating point value.                                                                                              |
| Char          | An 8-bit UTF-8 character value.                                                                                                                      |
| WideChar      | A 16-bit UTF-16 character value.                                                                                                                     |
| String        | A null terminated sequence of UTF-8 characters.                                                                                                      |
| CharArray     | A sequence of UTF-8 characters preceded by the number of characters.                                                                                 |
| WideString    | A null terminated sequence of UTF-16 characters.                                                                                                     |
| WideCharArray | A sequence of UTF-16 characters preceded by the number of characters.                                                                                |
| DateTime      | A 64-bit signed integer representing the number of 100 nanoseconds intervals since 1601-01-01 00:00:00. This is the same as the WIN32 FILETIME type. |
| ByteString    | A sequence of bytes preceded by its length in bytes.                                                                                                 |
| Guid          | A 128-bit structured type that represents a WIN32 GUID value.                                                                                        |

## C.4 Type Description Examples

### 1. A 128-bit signed integer.

```
<opc:OpaqueType Name="Int128" LengthInBits="128">
  <opc:Documentation>A 128-bit signed integer.</opc:Documentation>
</opc:OpaqueType>
```

### 2. A 16-bit value divided into several fields.

```
<opc:StructuredType Name="Quality">
  <opc:Documentation>An OPC COM-DA quality value.</opc:Documentation>
  <opc:Field Name="LimitBits" TypeName="opc:Bit" Length="2" />
  <opc:Field Name="QualityBits" TypeName="opc:Bit" Length="6"/>
  <opc:Field Name="VendorBits" TypeName="opc:Byte" />
</opc:StructuredType>
```

When using bit fields, the least significant bits within a byte shall appear first.

### 3. A structured type with optional fields.

```
<opc:StructuredType Name="DataValue">
  <opc:Documentation>A value with an associated timestamp, and
  quality.</opc:Documentation>
  <opc:Field Name="ValueSpecified" TypeName="Bit" />
  <opc:Field Name="StatusCodeSpecified" TypeName="Bit" />
  <opc:Field Name="TimestampSpecified" TypeName="Bit" />
  <opc:Field Name="Reserved1" TypeName="Bit" Length="5" />
  <opc:Field Name="Value" TypeName="Variant" SwitchField="ValueSpecified" />
  <opc:Field Name="Quality" TypeName="Quality" SwitchField="StatusCodeSpecified" />
  <opc:Field Name="Timestamp" TypeName="opc:DateTime" />
  <opc:Field Name="SourceTimestampSpecified" />
</opc:StructuredType>
```

It is necessary to explicitly specify any padding bits required to ensure subsequent fields line up on byte boundaries.

### 4. An array of integers.

```
<opc:StructuredType Name="IntegerArray">
```

```

<opc:Documentation>An array of integers prefixed by its length.</opc:Documentation>
<opc:Field Name="Size" TypeName="opc:Int32" />
<opc:Field Name="Array" TypeName="opc:Int32" LengthField="Size" />
</opc:StructuredType>

```

Nothing is encoded for the Array field if the Size field has a value  $\leq 0$ .

#### 5. An array of integers with a terminator instead of a length prefix.

```

<opc:StructuredType Name="IntegerArray" DefaultByteOrder="LittleEndian">
  <opc:Documentation>An array of integers terminated with a known
  value.</opc:Documentation>
  <opc:Field Name="Value" TypeName="opc:Int16" Terminator="FF7F" />
</opc:StructuredType>

```

The terminator is 32,767 converted to hexadecimal with LittleEndian byte order.

#### 6. A simple union.

```

<opc:StructuredType Name="Variant">
  <opc:Documentation>A union of several types.</opc:Documentation>
  <opc:Field Name="ArrayLengthSpecified" TypeName="opc:Bit" Length="1"/>
  <opc:Field Name="VariantType" TypeName="opc:Bit" Length="7" />
  <opc:Field Name="ArrayLength" TypeName="opc:Int32"
    SwitchField="ArrayLengthSpecified" />
  <opc:Field Name="Int32" TypeName="opc:Int32" LengthField="ArrayLength"
    SwitchField="VariantType" SwitchValue="1" />
  <opc:Field Name="String" TypeName="opc:String" LengthField="ArrayLength"
    SwitchField="VariantType" SwitchValue="2" />
  <opc:Field Name="DateTime" TypeName="opc:DateTime" LengthField="ArrayLength"
    SwitchField="VariantType" SwitchValue="3" />
</opc:StructuredType>

```

The *ArrayLength* field is optional. If it is not present in an encoded value, then the length of all fields with *LengthField* set to "ArrayLength" have a length of 1.

It is valid for the *VariantType* field to have a value that has no matching field defined. This simply means all optional fields are not present in the encoded value.

#### 7. An enumerated type.

```

<opc:EnumeratedType Name="TrafficLight" LengthInBits="32">
  <opc:Documentation>The possible colours for a traffic signal.</opc:Documentation>
  <opc:EnumeratedValue Name="Red" Value="4">
    <opc:Documentation>Red says stop immediately.</opc:Documentation>
  </opc:EnumeratedValue>
  <opc:EnumeratedValue Name="Yellow" Value="3">
    <opc:Documentation>Yellow says prepare to stop.</opc:Documentation>
  </opc:EnumeratedValue>
  <opc:EnumeratedValue Name="Green" Value="2">
    <opc:Documentation>Green says you may proceed.</opc:Documentation>
  </opc:EnumeratedValue>
</opc:EnumeratedType>

```

The documentation element is used to provide human readable description of the type and values.

#### 8. A nillable array.

```

<opc:StructuredType Name="NillableArray">
  <opc:Documentation>An array where a length of -1 means null.</opc:Documentation>
  <opc:Field Name="Length" TypeName="opc:Int32" />
  <opc:Field
    Name="Int32"
    TypeName="opc:Int32"
    LengthField="Length"
    SwitchField="Length"
    SwitchValue="0"
    SwitchOperand="GreaterThanOrEqual" />
</opc:StructuredType>

```

If the length of the array is -1 then the array does not appear in the stream.

## C.5 OPC Binary XML Schema

```

<?xml version="1.0" encoding="utf-8" ?>
<xs:schema
  targetNamespace="http://opcfoundation.org/BinarySchema/"
  elementFormDefault="qualified"
  xmlns="http://opcfoundation.org/BinarySchema/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
>
  <xs:element name="Documentation">
    <xs:complexType mixed="true">
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:any minOccurs="0" maxOccurs="unbounded" />
      </xs:choice>
      <xs:anyAttribute />
    </xs:complexType>
  </xs:element>

  <xs:complexType name="ImportDirective">
    <xs:attribute name="Namespace" type="xs:string" use="optional" />
    <xs:attribute name="Location" type="xs:string" use="optional" />
  </xs:complexType>

  <xs:simpleType name="ByteOrder">
    <xs:restriction base="xs:string">
      <xs:enumeration value="BigEndian" />
      <xs:enumeration value="LittleEndian" />
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="TypeDescription">
    <xs:sequence>
      <xs:element ref="Documentation" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
    <xs:attribute name="Name" type="xs:NCName" use="required" />
    <xs:attribute name="DefaultByteOrder" type="ByteOrder" use="optional" />
    <xs:anyAttribute processContents="lax" />
  </xs:complexType>

  <xs:complexType name="OpaqueType">
    <xs:complexContent>
      <xs:extension base="TypeDescription">
        <xs:attribute name="LengthInBits" type="xs:int" use="optional" />
        <xs:attribute name="ByteOrderSignificant" type="xs:boolean" default="false" />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="EnumeratedValue">
    <xs:sequence>
      <xs:element ref="Documentation" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
    <xs:attribute name="Name" type="xs:string" use="optional" />
    <xs:attribute name="Value" type="xs:unsignedInt" use="optional" />
  </xs:complexType>

  <xs:complexType name="EnumeratedType">
    <xs:complexContent>
      <xs:extension base="OpaqueTypeDescription">
        <xs:sequence>
          <xs:element name="EnumeratedValue" type="EnumeratedValueDescription"
maxOccurs="unbounded" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:simpleType name="SwitchOperand">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Equals" />
      <xs:enumeration value="GreaterThan" />
      <xs:enumeration value="LessThan" />
      <xs:enumeration value="GreaterThanOrEqual" />
      <xs:enumeration value="LessThanOrEqual" />
      <xs:enumeration value="NotEqual" />
    </xs:restriction>
  </xs:simpleType>

```

```

<xs:complexType name="FieldType">
  <xs:sequence>
    <xs:element ref="Documentation" minOccurs="0" maxOccurs="1" />
  </xs:sequence>
  <xs:attribute name="Name" type="xs:string" use="required" />
  <xs:attribute name="TypeName" type="xs:QName" use="optional" />
  <xs:attribute name="Length" type="xs:unsignedInt" use="optional" />
  <xs:attribute name="LengthField" type="xs:string" use="optional" />
  <xs:attribute name="IsLengthInBytes" type="xs:boolean" default="false" />
  <xs:attribute name="SwitchField" type="xs:string" use="optional" />
  <xs:attribute name="SwitchValue" type="xs:unsignedInt" use="optional" />
  <xs:attribute name="SwitchOperand" type="SwitchOperand" use="optional" />
  <xs:attribute name="Terminator" type="xs:hexBinary" use="optional" />
  <xs:anyAttribute processContents="lax" />
</xs:complexType>

<xs:complexType name="StructuredType">
  <xs:complexContent>
    <xs:extension base="TypeDescription">
      <xs:sequence>
        <xs:element name="Field" type="FieldType" minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="TypeDictionary">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Documentation" minOccurs="0" maxOccurs="1" />
      <xs:element name="Import" type="ImportDirective" minOccurs="0" maxOccurs="unbounded" />
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="OpaqueType" type="OpaqueType" />
        <xs:element name="EnumeratedType" type="EnumeratedType" />
        <xs:element name="StructuredType" type="StructuredType" />
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="TargetNamespace" type="xs:string" use="required" />
    <xs:attribute name="DefaultByteOrder" type="ByteOrder" use="optional" />
  </xs:complexType>
</xs:element>
</xs:schema>

```

## C.6 OPC Binary Standard TypeDictionary

```

<?xml version="1.0" encoding="utf-8"?>
<opc:TypeDictionary
  xmlns="http://opcfoundation.org/BinarySchema/"
  xmlns:opc="http://opcfoundation.org/BinarySchema/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  TargetNamespace="http://opcfoundation.org/BinarySchema/"
>
  <opc:Documentation>This dictionary defines the standard types used by the OPC Binary
  type description system.</opc:Documentation>

  <opc:OpaqueType Name="Bit" LengthInBits="1">
    <opc:Documentation>A single bit.</opc:Documentation>
  </opc:OpaqueType>

  <opc:OpaqueType Name="Boolean" LengthInBits="8">
    <opc:Documentation>A two state logical value represented as a 8-bit
    value.</opc:Documentation>
  </opc:OpaqueType>

  <opc:OpaqueType Name="SByte" LengthInBits="8">
    <opc:Documentation>An 8-bit signed integer.</opc:Documentation>
  </opc:OpaqueType>

  <opc:OpaqueType Name="Byte" LengthInBits="8">
    <opc:Documentation>A 8-bit unsigned integer.</opc:Documentation>
  </opc:OpaqueType>

  <opc:OpaqueType Name="Int16" LengthInBits="16" ByteOrderSignificant="true">
    <opc:Documentation>A 16-bit signed integer.</opc:Documentation>
  </opc:OpaqueType>

```

```

<opc:OpaqueType Name="UInt16" LengthInBits="16" ByteOrderSignificant="true">
  <opc:Documentation>A 16-bit unsigned integer.</opc:Documentation>
</opc:OpaqueType>

<opc:OpaqueType Name="Int32" LengthInBits="32" ByteOrderSignificant="true">
  <opc:Documentation>A 32-bit signed integer.</opc:Documentation>
</opc:OpaqueType>

<opc:OpaqueType Name="UInt32" LengthInBits="32" ByteOrderSignificant="true">
  <opc:Documentation>A 32-bit unsigned integer.</opc:Documentation>
</opc:OpaqueType>

<opc:OpaqueType Name="Int64" LengthInBits="64" ByteOrderSignificant="true">
  <opc:Documentation>A 64-bit signed integer.</opc:Documentation>
</opc:OpaqueType>

<opc:OpaqueType Name="UInt64" LengthInBits="64" ByteOrderSignificant="true">
  <opc:Documentation>A 64-bit unsigned integer.</opc:Documentation>
</opc:OpaqueType>

<opc:OpaqueType Name="Float" LengthInBits="32" ByteOrderSignificant="true">
  <opc:Documentation>An IEEE-754 single precision floating point
value.</opc:Documentation>
</opc:OpaqueType>

<opc:OpaqueType Name="Double" LengthInBits="64" ByteOrderSignificant="true">
  <opc:Documentation>An IEEE-754 double precision floating point
value.</opc:Documentation>
</opc:OpaqueType>

<opc:OpaqueType Name="Char" LengthInBits="8">
  <opc:Documentation>A 8-bit character value.</opc:Documentation>
</opc:OpaqueType>

<opc:StructuredType Name="String">
  <opc:Documentation>A UTF-8 null terminated string value.</opc:Documentation>
  <opc:Field Name="Value" TypeName="Char" Terminator="00" />
</opc:StructuredType>

<opc:StructuredType Name="CharArray">
  <opc:Documentation>A UTF-8 string prefixed by its length in
characters.</opc:Documentation>
  <opc:Field Name="Length" TypeName="Int32" />
  <opc:Field Name="Value" TypeName="Char" LengthField="Length" />
</opc:StructuredType>

<opc:OpaqueType Name="WideChar" LengthInBits="16" ByteOrderSignificant="true">
  <opc:Documentation>A 16-bit character value.</opc:Documentation>
</opc:OpaqueType>

<opc:StructuredType Name="WideString">
  <opc:Documentation>A UTF-16 null terminated string value.</opc:Documentation>
  <opc:Field Name="Value" TypeName="WideChar" Terminator="0000" />
</opc:StructuredType>

<opc:StructuredType Name="WideCharArray">
  <opc:Documentation>A UTF-16 string prefixed by its length in
characters.</opc:Documentation>
  <opc:Field Name="Length" TypeName="Int32" />
  <opc:Field Name="Value" TypeName="WideChar" LengthField="Length" />
</opc:StructuredType>

<opc:StructuredType Name="ByteString">
  <opc:Documentation>An array of bytes prefixed by its length.</opc:Documentation>
  <opc:Field Name="Length" TypeName="Int32" />
  <opc:Field Name="Value" TypeName="Byte" LengthField="Length" />
</opc:StructuredType>

<opc:OpaqueType Name="DateTime" LengthInBits="64" ByteOrderSignificant="true">
  <opc:Documentation>The number of 100 nanosecond intervals since January 01,
1601.</opc:Documentation>
</opc:OpaqueType>

<opc:StructuredType Name="Guid">
  <opc:Documentation>A 128-bit globally unique identifier.</opc:Documentation>
  <opc:Field Name="Data1" TypeName="UInt32" />
  <opc:Field Name="Data2" TypeName="UInt16" />

```

```
<opc:Field Name="Data3" TypeName="UInt16" />  
<opc:Field Name="Data4" TypeName="Byte" Length="8" />  
</opc:StructuredType>  
</opc:TypeDictionary>
```

IECNORM.COM: Click to view the full PDF of IEC 62541-3:2015

Withd2W

## Annex D (normative)

### Graphical Notation

#### D.1 General

Annex D defines a graphical notation for OPC UA data. Annex D is normative, that is, the notation is used in this standard to expose examples of OPC UA data. However, it is not required to use this notation to expose OPC UA data.

The graphical notation is able to expose all structural data of OPC UA. *Nodes*, their *Attributes* including their current value and *References* between the *Nodes* including the *ReferenceType* can be exposed. The graphical notation provides no mechanism to expose events or historical data.

#### D.2 Notation

##### D.2.1 Overview

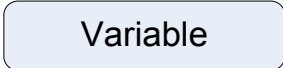
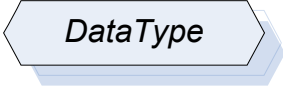
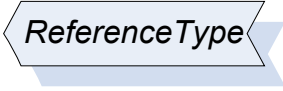
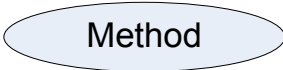
The notation is divided into two parts. The simple notation only provides a simplified view on the data hiding some details like *Attributes*. The extended notation allows exposing all structure information of OPC UA, including *Attribute* values. The simple and the extended notation can be combined to expose OPC UA data in one figure.

Common to both notations is that neither any colour nor the thickness or style of lines is relevant for the notation. Those effects can be used to highlight certain aspects of a figure.

##### D.2.2 Simple Notation

Depending on their *NodeClass* *Nodes* are represented by different graphical forms as defined in Table D.1.

**Table D.1 – Notation of Nodes depending on the NodeClass**

| NodeClass     | Graphical Representation                                                           | Comment                                                                                                                                                                     |
|---------------|------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Object        |   | Rectangle including text representing the string-part of the <i>DisplayName</i> of the <i>Object</i> . The font shall not be set to italic.                                 |
| ObjectType    |   | Shadowed rectangle including text representing the string-part of the <i>DisplayName</i> of the <i>ObjectType</i> . The font shall be set in italic.                        |
| Variable      |   | Rectangle with rounded corners including text representing the string-part of the <i>DisplayName</i> of the <i>Variable</i> . The font shall not be set in italic.          |
| VariableType  |   | Shadowed rectangle with rounded corners including text representing the string-part of the <i>DisplayName</i> of the <i>VariableType</i> . The font shall be set in italic. |
| DataType      |   | Shadowed hexagon including text representing the string-part of the <i>DisplayName</i> of the <i>DataType</i> .                                                             |
| ReferenceType |   | Shadowed six-sided polygon including text representing the string-part of the <i>DisplayName</i> of the <i>ReferenceType</i> .                                              |
| Method        |   | Oval including text representing the string-part of the <i>DisplayName</i> of the <i>Method</i> .                                                                           |
| View          |  | Trapezium including text representing the string-part of the <i>DisplayName</i> of the <i>View</i> .                                                                        |

*References* are represented as lines between *Nodes* as exemplified in Figure D.1. Those lines can vary in their form. They do not have to connect the *Nodes* with a straight line; they can have angles, arches, etc.



IEC

**Figure D.1 – Example of a Reference connecting two Nodes**

Table D.2 defines how symmetric and asymmetric *References* are represented in general, and also defines shortcuts for some *ReferenceTypes*. Although it is recommended to use those shortcuts, it is not required. Thus, instead of using the shortcut, the generic solution can also be used.

**Table D.2 – Simple Notation of Nodes depending on the NodeClass**

| ReferenceType                  | Graphical Representation                                                           | Comment                                                                                                                                                                                                                                                                             |
|--------------------------------|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Any symmetric ReferenceType    |   | Symmetric <i>ReferenceTypes</i> are represented as lines between <i>Nodes</i> with closed and filled arrows on both sides pointing to the connected <i>Nodes</i> . Near the line has to be a text containing the string-part of the <i>BrowseName</i> of the <i>ReferenceType</i> . |
| Any asymmetric ReferenceType   |   | Asymmetric <i>ReferenceTypes</i> are represented as lines between <i>Nodes</i> with a closed and filled arrow on the side pointing to the <i>TargetNode</i> . Near the line has to be a text containing the string-part of the <i>BrowseName</i> of the <i>ReferenceType</i> .      |
| Any hierarchical ReferenceType |   | Asymmetric <i>ReferenceTypes</i> that are subtypes of <i>HierarchicalReferences</i> should be exposed the same way as asymmetric <i>ReferenceTypes</i> except that an open arrow is used.                                                                                           |
| HasComponent                   |   | The notation provides a shortcut for <i>HasComponent References</i> shown on the left. The single hashed line has to be near the <i>TargetNode</i> .                                                                                                                                |
| HasProperty                    |   | The notation provides a shortcut for <i>HasProperty References</i> shown on the left. The double hashed lines have to be near the <i>TargetNode</i> .                                                                                                                               |
| HasTypeDefinition              |   | The notation provides a shortcut for <i>HasTypeDefinition References</i> shown on the left. The double closed and filled arrows have to point to the <i>TargetNode</i> .                                                                                                            |
| HasSubtype                     |   | The notation provides a shortcut for <i>HasSubtype References</i> shown on the left. The double closed arrows have to point to the <i>SourceNode</i> .                                                                                                                              |
| HasEventSource                 |  | The notation provides a shortcut for <i>HasEventSource References</i> shown on the left. The closed arrow has to point to the <i>TargetNode</i> .                                                                                                                                   |

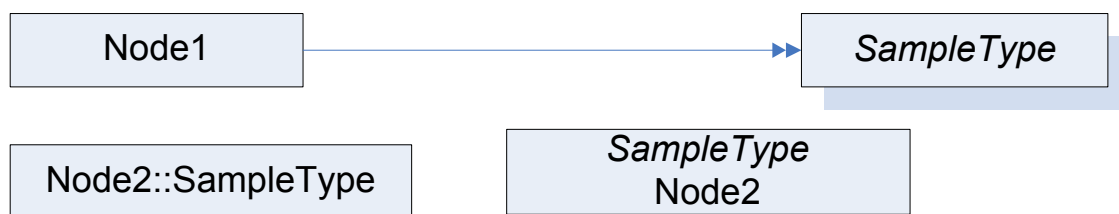
### D.2.3 Extended Notation

In the extended notation some additional concepts are introduced. It is allowed only to use some of those concepts on elements of a figure.

The following rules define some special handling of structures.

- In general, values of all *DataTypes* should be represented by an appropriate string representation. Whenever a *NamespaceIndex* or *LocaleId* is used in those structures they can be omitted.
- The *DisplayName* contains a *LocaleId* and a *String*. Such a structure can be exposed as [*<LocaleId>*]:*<String>* where the *LocaleId* is optional. For example, a *DisplayName* can be "en:MyName". Instead of that, "MyName" can also be used. This rule applies whenever a *DisplayName* is shown, including the text used in the graphical representation of a *Node*.
- The *BrowseName* contains the *NamespaceIndex* and a *String*. Such a structure can be exposed as [*<NamespaceIndex>*]:*<String>* where the *NamespaceIndex* is optional. For example, a *BrowseName* can be "1:MyName". Instead of that, "MyName" can also be used. This rule applies whenever a *BrowseName* is shown, including the text used in the graphical representation of a *Node*.

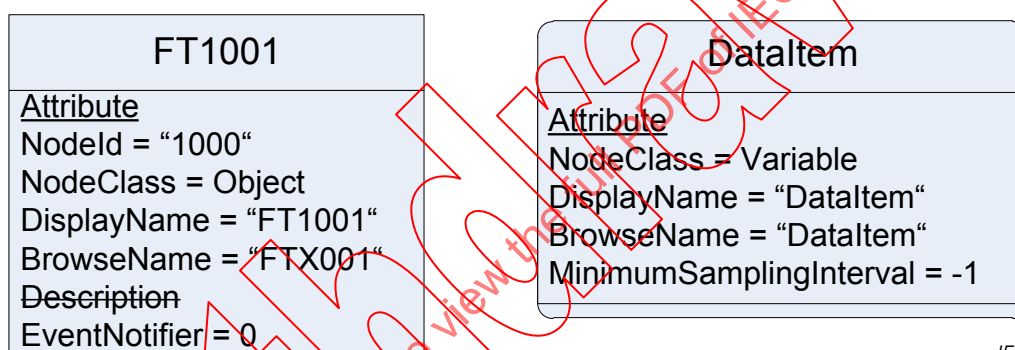
Instead of using the *HasTypeDefinition* reference to point from an *Object* or *Variable* to its *ObjectType* or *VariableType* the name of the *TypeDefinition* can be added to the text used in the *Node*. The *TypeDefinition* shall either be prefixed with "::" or it is put in italic as the top line. Figure D.2 gives an example, where "Node1" uses a *Reference* and "Node2" the shortcut in both notation variants. A figure can contain *HasTypeDefinition References* for some *Nodes* and the shortcut for other *Nodes*. It is not allowed that a *Node* uses the shortcut and additionally is the *SourceNode* of a *HasTypeDefinition*.



IEC

**Figure D.2 – Example of using a TypeDefinition inside a Node**

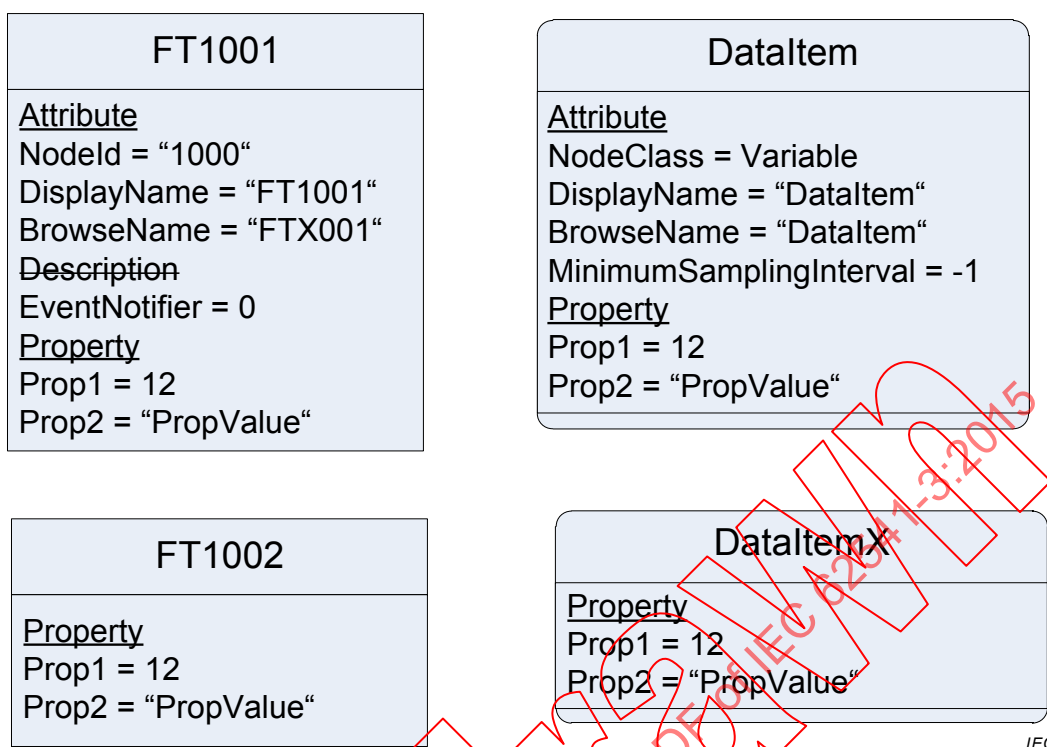
To display *Attributes* of a *Node* additional text can be put inside the form representing the *Node* under the text representing the *DisplayName*. The *DisplayName* and the text describing the *Attributes* have to be separated using a horizontal line. Each *Attribute* has to be set into a new text line. Each text line shall contain the *Attribute* name followed by an “=” and the value of the *Attribute*. On top of the first text line containing an *Attribute* shall be a text line containing the underlined text “*Attribute*”. It is not required to expose all *Attributes* of a *Node*. It is allowed to show only a subset of *Attributes*. If an optional *Attribute* is not provided, the *Attribute* can be marked by a strike-through line, for example “*Description*”. Examples of exposing *Attributes* are shown in Figure D.3.



IEC

**Figure D.3 – Example of exposing Attributes**

To avoid too many *Nodes* in a figure it is allowed to expose *Properties* inside a *Node*, similar to *Attributes*. Therefore, the text field used for exposing *Attributes* is extended. Under the last text line containing an *Attribute* a new text line containing the underlined text “*Property*” has to be added. If no *Attribute* is provided, the text has to start with this text line. After this text line, each new text line shall contain a *Property*, starting with the *BrowseName* of the *Property* followed by “=” and the value of the *Value Attribute* of the *Property*. Figure D.4 shows some examples exposing *Properties* inline. It is allowed to expose some *Properties* of a *Node* inline, and other *Properties* as *Nodes*. It is not allowed to show a *Property* inline as well as an additional *Node*.



**Figure D.4 – Example of exposing Properties inline**

It is allowed to add additional information to a figure using the graphical representation, for example callouts.

IECNORM.COM :: Click to view the full PDF of IEC 62541-3:2015

## SOMMAIRE

|                                                                            |     |
|----------------------------------------------------------------------------|-----|
| AVANT-PROPOS .....                                                         | 128 |
| 1 Domaine d'application .....                                              | 130 |
| 2 Références normatives .....                                              | 130 |
| 3 Termes, définitions, abréviations et conventions .....                   | 131 |
| 3.1 Termes et définitions .....                                            | 131 |
| 3.2 Abréviations .....                                                     | 132 |
| 3.3 Conventions .....                                                      | 132 |
| 3.3.1 Conventions pour les figures Espace d'Adressage .....                | 132 |
| 3.3.2 Conventions pour la définition des Classes de Nœuds .....            | 133 |
| 4 Concepts de l'Espace d'Adressage .....                                   | 134 |
| 4.1 Vue d'ensemble .....                                                   | 134 |
| 4.2 Modèle d'objet .....                                                   | 134 |
| 4.3 Modèle de Nœud .....                                                   | 135 |
| 4.3.1 Généralités .....                                                    | 135 |
| 4.3.2 Classes de Nœuds .....                                               | 136 |
| 4.3.3 Attributs .....                                                      | 136 |
| 4.3.4 Références .....                                                     | 136 |
| 4.4 Variables .....                                                        | 137 |
| 4.4.1 Généralités .....                                                    | 137 |
| 4.4.2 Propriétés .....                                                     | 137 |
| 4.4.3 Variables de Données .....                                           | 138 |
| 4.5 TypeDefinitionNodes (Nœuds de Définition de Type) .....                | 138 |
| 4.5.1 Généralités .....                                                    | 138 |
| 4.5.2 TypeDefinitionNodes complexes et leurs Déclarations d'Instance ..... | 139 |
| 4.5.3 Sous-typage .....                                                    | 140 |
| 4.5.4 Instanciation de TypeDefinitionNodes complexes .....                 | 141 |
| 4.6 Modèle d'événement .....                                               | 143 |
| 4.6.1 Généralités .....                                                    | 143 |
| 4.6.2 Types d'Événements .....                                             | 143 |
| 4.6.3 Catégorisation des Événements .....                                  | 144 |
| 4.7 Méthodes .....                                                         | 144 |
| 5 Classes de Nœuds normalisées .....                                       | 145 |
| 5.1 Vue d'ensemble .....                                                   | 145 |
| 5.2 Classe de Nœud de base .....                                           | 145 |
| 5.2.1 Généralités .....                                                    | 145 |
| 5.2.2 NodeId (Identificateur de Nœud) .....                                | 146 |
| 5.2.3 NodeClass (Classe de Nœud) .....                                     | 146 |
| 5.2.4 BrowseName (Nom de Navigation) .....                                 | 146 |
| 5.2.5 DisplayName (Nom d'Affichage) .....                                  | 147 |
| 5.2.6 Description .....                                                    | 147 |
| 5.2.7 WriteMask (Masque d'Écriture) .....                                  | 147 |
| 5.2.8 UserWriteMask (Masque d'Écriture Utilisateur) .....                  | 148 |
| 5.3 ReferenceTypeNodeClass (Classe de Nœud "Types de Références") .....    | 148 |
| 5.3.1 Généralités .....                                                    | 148 |
| 5.3.2 Attributs .....                                                      | 149 |
| 5.3.3 Références .....                                                     | 151 |

|       |                                                                                                                                   |     |
|-------|-----------------------------------------------------------------------------------------------------------------------------------|-----|
| 5.4   | Classe de Nœud "Vues" (View) .....                                                                                                | 152 |
| 5.5   | Objets .....                                                                                                                      | 154 |
| 5.5.1 | Classe de Nœud "Objets" .....                                                                                                     | 154 |
| 5.5.2 | Classe de Nœud "ObjectType" (Types d'Objets) .....                                                                                | 157 |
| 5.5.3 | Type d'Objet normalisé "FolderType" (Type de Dossier) .....                                                                       | 158 |
| 5.5.4 | Création du côté client d'Objets d'un Type d'Objet donné.....                                                                     | 158 |
| 5.6   | Variables .....                                                                                                                   | 159 |
| 5.6.1 | Généralités .....                                                                                                                 | 159 |
| 5.6.2 | Classe de Nœud "Variables" .....                                                                                                  | 159 |
| 5.6.3 | Propriétés .....                                                                                                                  | 163 |
| 5.6.4 | Variable de Données .....                                                                                                         | 163 |
| 5.6.5 | Classe de nœud "VariableType" (Types de Variables) .....                                                                          | 164 |
| 5.6.6 | Création du côté client de Variables d'un Type de Variable donné.....                                                             | 167 |
| 5.7   | Classe de Nœud "Méthodes" .....                                                                                                   | 167 |
| 5.8   | Types de Données .....                                                                                                            | 169 |
| 5.8.1 | Modèle de Type de Données .....                                                                                                   | 169 |
| 5.8.2 | Règles de codage pour différentes sortes de Types de Données .....                                                                | 171 |
| 5.8.3 | Classe de nœud "DataType" ("Types de Données") .....                                                                              | 172 |
| 5.8.4 | Dictionnaire de Type de Données, Description de Type de Données,<br>Codage de Type de Données et Système de Type de Données ..... | 174 |
| 5.9   | Synthèse des Attributs de NodeClasses (Classes de Nœuds) .....                                                                    | 177 |
| 6     | Modèle Type de Types d'Objets et de Types de Variables .....                                                                      | 178 |
| 6.1   | Vue d'ensemble .....                                                                                                              | 178 |
| 6.2   | Définitions.....                                                                                                                  | 178 |
| 6.2.1 | Déclaration d'Instance .....                                                                                                      | 178 |
| 6.2.2 | Instances sans Règles de Modélisation .....                                                                                       | 178 |
| 6.2.3 | Hierarchie de Déclaration d'Instance.....                                                                                         | 179 |
| 6.2.4 | Nœud de Déclaration d'Instance similaire .....                                                                                    | 179 |
| 6.2.5 | Chemin de Navigation (BrowsePath).....                                                                                            | 179 |
| 6.2.6 | Traitement des Attributs de Déclarations d'Instance.....                                                                          | 179 |
| 6.2.7 | Traitement des Attributs de Variables et Types de Variables.....                                                                  | 179 |
| 6.2.8 | Identificateurs de Nœuds de Déclarations d'Instance.....                                                                          | 179 |
| 6.3   | Sous-typage de Types d'Objets et de Types de Variables .....                                                                      | 180 |
| 6.3.1 | Vue d'ensemble .....                                                                                                              | 180 |
| 6.3.2 | Attributs .....                                                                                                                   | 180 |
| 6.3.3 | Déclarations d'Instance.....                                                                                                      | 180 |
| 6.4   | Instances de Types d'Objets et de Types de Variables.....                                                                         | 184 |
| 6.4.1 | Vue d'ensemble .....                                                                                                              | 184 |
| 6.4.2 | Création d'une Instance .....                                                                                                     | 184 |
| 6.4.3 | Contraintes applicables à une Instance.....                                                                                       | 185 |
| 6.4.4 | Règles de Modélisation.....                                                                                                       | 186 |
| 6.5   | Modifications de définitions de type déjà utilisées .....                                                                         | 195 |
| 7     | Types de Références normalisés .....                                                                                              | 195 |
| 7.1   | Généralités .....                                                                                                                 | 195 |
| 7.2   | Type de Référence "Références" .....                                                                                              | 197 |
| 7.3   | Type de Référence "HierarchicalReferences" (Références Hiérarchiques) .....                                                       | 197 |
| 7.4   | Type de Référence "NonHierarchicalReferences" (Références Non<br>Hiérarchiques) .....                                             | 197 |
| 7.5   | Type de Référence "HasChild" (Dispose d'une Référence Enfant) .....                                                               | 197 |

|       |                                                                                    |     |
|-------|------------------------------------------------------------------------------------|-----|
| 7.6   | Type de Référence "Aggregates" (Regroupe).....                                     | 198 |
| 7.7   | Type de Référence "HasComponent" (Dispose d'un composant).....                     | 198 |
| 7.8   | Type de Référence "HasProperty" (Dispose d'une Propriété).....                     | 198 |
| 7.9   | Type de Référence "HasOrderedComponent " (Dispose de Composants<br>Ordonnés).....  | 198 |
| 7.10  | Type de Référence "HasSubtype" (Dispose d'un Sous-type).....                       | 199 |
| 7.11  | Type de référence "Organizes" (Organise).....                                      | 199 |
| 7.12  | Type de Référence "HasModellingRule" (Dispose d'une Règle de<br>Modélisation)..... | 199 |
| 7.13  | Type de Référence "HasTypeDefinition" (Dispose d'une Définition de Type).....      | 200 |
| 7.14  | Type de Référence "HasEncoding" (Dispose d'un Codage).....                         | 200 |
| 7.15  | Type de Référence "HasDescription" (Dispose d'une Description).....                | 200 |
| 7.16  | Type de Référence "GeneratesEvent" (Génère un Événement).....                      | 200 |
| 7.17  | Type de Référence "AlwaysGeneratesEvent" (Génère Toujours un<br>Événement).....    | 201 |
| 7.18  | Type de Référence "HasEventSource" (Dispose d'une Source d'Événement).....         | 201 |
| 7.19  | Type de Référence "HasNotifier" (Dispose d'un Notificateur).....                   | 201 |
| 8     | Types de Données Normalisés.....                                                   | 204 |
| 8.1   | Généralités.....                                                                   | 204 |
| 8.2   | Identificateur de Nœud.....                                                        | 204 |
| 8.2.1 | Généralités.....                                                                   | 204 |
| 8.2.2 | Indice d'Espace de Nom.....                                                        | 204 |
| 8.2.3 | Type d'Identificateur.....                                                         | 205 |
| 8.2.4 | Valeur d'Identificateur.....                                                       | 205 |
| 8.3   | Nom Qualifié.....                                                                  | 206 |
| 8.4   | Identificateur de Paramètre de lieu.....                                           | 206 |
| 8.5   | Texte Localisé.....                                                                | 206 |
| 8.6   | Argument.....                                                                      | 207 |
| 8.7   | Type de Données de Base (BaseDataType).....                                        | 207 |
| 8.8   | Booléen (Boolean).....                                                             | 207 |
| 8.9   | Octet (Byte).....                                                                  | 207 |
| 8.10  | Chaîne d'octets (ByteString).....                                                  | 207 |
| 8.11  | Date et Heure (DateTime).....                                                      | 208 |
| 8.12  | Double.....                                                                        | 208 |
| 8.13  | Durée (Duration).....                                                              | 208 |
| 8.14  | Énumération (Enumeration).....                                                     | 208 |
| 8.15  | Virgule flottante (Float).....                                                     | 208 |
| 8.16  | Guid (Globally Unique Identifier – Identificateur Globalement Unique).....         | 208 |
| 8.17  | SByte.....                                                                         | 208 |
| 8.18  | IdType.....                                                                        | 208 |
| 8.19  | Image.....                                                                         | 208 |
| 8.20  | ImageBMP.....                                                                      | 209 |
| 8.21  | ImageGIF.....                                                                      | 209 |
| 8.22  | ImageJPG.....                                                                      | 209 |
| 8.23  | ImagePNG.....                                                                      | 209 |
| 8.24  | Entier (Integer).....                                                              | 209 |
| 8.25  | Int16.....                                                                         | 209 |
| 8.26  | Int32.....                                                                         | 209 |
| 8.27  | Int64.....                                                                         | 209 |

|      |                                                                                                                         |     |
|------|-------------------------------------------------------------------------------------------------------------------------|-----|
| 8.28 | Type de Données de Fuseau Horaire (TimeZoneDataType) .....                                                              | 209 |
| 8.29 | Type de Règle de Nommage (NamingRuleType) .....                                                                         | 209 |
| 8.30 | Classe de Nœud (NodeClass) .....                                                                                        | 210 |
| 8.31 | Nombre (Number) .....                                                                                                   | 210 |
| 8.32 | Chaîne (String) .....                                                                                                   | 210 |
| 8.33 | Structure .....                                                                                                         | 210 |
| 8.34 | UInteger .....                                                                                                          | 210 |
| 8.35 | UInt16 .....                                                                                                            | 210 |
| 8.36 | UInt32 .....                                                                                                            | 210 |
| 8.37 | UInt64 .....                                                                                                            | 210 |
| 8.38 | UtcTime .....                                                                                                           | 211 |
| 8.39 | XmlElement .....                                                                                                        | 211 |
| 8.40 | Type de Valeurs d'Énumération (EnumValueType) .....                                                                     | 211 |
| 9    | Types d'Événements normalisés .....                                                                                     | 211 |
| 9.1  | Généralités .....                                                                                                       | 211 |
| 9.2  | Type d'Événement de Base (BaseEventType) .....                                                                          | 213 |
| 9.3  | Type d'Événement Système (SystemEventType) .....                                                                        | 213 |
| 9.4  | Type d'Événement de Progrès (ProgressEventType) .....                                                                   | 213 |
| 9.5  | Type d'Événement d'Audit (AuditEventType) .....                                                                         | 213 |
| 9.6  | Type d'Événement Audit de Sécurité (AuditSecurityEventType) .....                                                       | 216 |
| 9.7  | Type d'Événement Canal d'Audit (AuditChannelEventType) .....                                                            | 216 |
| 9.8  | Type d'Événement Audit d'Ouverture de Canal Sécurisé<br>(AuditOpenSecureChannelEventType) .....                         | 216 |
| 9.9  | Type d'Événement Session d'Audit (AuditSessionEventType) .....                                                          | 216 |
| 9.10 | Type d'Événement Création de Session d'Audit<br>(AuditCreateSessionEventType) .....                                     | 216 |
| 9.11 | Type d'Événement Audit de Non Correspondance d'Url<br>(AuditUriMismatchEventType) .....                                 | 216 |
| 9.12 | Type d'Événement Audit d'Activation de Session<br>(AuditActivateSessionEventType) .....                                 | 216 |
| 9.13 | Type d'Événement Audit d'Annulation (AuditCancelEventType) .....                                                        | 216 |
| 9.14 | Type d'Événement Certificat d'Audit (AuditCertificateEventType) .....                                                   | 217 |
| 9.15 | Type d'Événement Audit de Non Correspondance des Données de Certificat<br>(AuditCertificateDataMismatchEventType) ..... | 217 |
| 9.16 | Type d'Événement Certificat d'Audit Expiré<br>(AuditCertificateExpiredEventType) .....                                  | 217 |
| 9.17 | Type d'Événement Certificat d'Audit Invalide<br>(AuditCertificateInvalidEventType) .....                                | 217 |
| 9.18 | Type d'Événement Certificat d'Audit Douteux<br>(AuditCertificateUntrustedEventType) .....                               | 217 |
| 9.19 | Type d'Événement Certificat d'Audit Révoqué<br>(AuditCertificateRevokedEventType) .....                                 | 217 |
| 9.20 | Type d'Événement Non Correspondance de Certificat d'Audit<br>(AuditCertificateMismatchEventType) .....                  | 217 |
| 9.21 | Type d'Événement Audit de Gestion de Nœuds<br>(AuditNodeManagementEventType) .....                                      | 218 |
| 9.22 | Type d'Événement Audit d'Ajout de Nœuds (AuditAddNodesEventType) .....                                                  | 218 |
| 9.23 | Type d'Événement Audit de Suppression de Nœuds<br>(AuditDeleteNodesEventType) .....                                     | 218 |
| 9.24 | Type d'Événement Audit d'Ajout de Références<br>(AuditAddReferencesEventType) .....                                     | 218 |

|                                                                                 |                                                                                               |     |
|---------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|-----|
| 9.25                                                                            | Type d'Événement Audit de Suppression de Références<br>(AuditDeleteReferencesEventType) ..... | 218 |
| 9.26                                                                            | Type d'Événement Audit de Mise à Jour (AuditUpdateEventType) .....                            | 218 |
| 9.27                                                                            | Type d'Événement Audit de Mise à Jour d'Ecriture<br>(AuditWriteUpdateEventType) .....         | 218 |
| 9.28                                                                            | Type d'Événement Audit de Mise à Jour de l'Historique<br>(AuditHistoryUpdateEventType) .....  | 218 |
| 9.29                                                                            | Type d'Événement Audit de Mise à Jour de Méthode<br>(AuditUpdateMethodEventType) .....        | 218 |
| 9.30                                                                            | Type d'Événement de Défaillance d'Appareil (DeviceFailureEventType) .....                     | 219 |
| 9.31                                                                            | Type d'Événement Changement de Statut du Système<br>(SystemStatusChangeEvent) .....           | 219 |
| 9.32                                                                            | Événements de Modification de Modèle (ModelChangeEvents) .....                                | 219 |
| 9.32.1                                                                          | Généralités .....                                                                             | 219 |
| 9.32.2                                                                          | Propriété "NodeVersion" (Version de Nœud) .....                                               | 219 |
| 9.32.3                                                                          | Vues .....                                                                                    | 219 |
| 9.32.4                                                                          | Compression d'Événements .....                                                                | 220 |
| 9.32.5                                                                          | Type d'Événement de Modification du Modèle de Base<br>(BaseModelChangeEvent) .....            | 220 |
| 9.32.6                                                                          | Type d'Événement de Modification du Modèle Général<br>(GeneralModelChangeEvent) .....         | 220 |
| 9.32.7                                                                          | Principes directeurs des Événements de Modification de Modèle .....                           | 220 |
| 9.33                                                                            | Type d'Événement de Modification Sémantique<br>(SemanticChangeEvent) .....                    | 221 |
| 9.33.1                                                                          | Généralités .....                                                                             | 221 |
| 9.33.2                                                                          | Propriétés "ViewVersion" et "NodeVersion" .....                                               | 221 |
| 9.33.3                                                                          | Vues .....                                                                                    | 221 |
| 9.33.4                                                                          | Compression d'Événements .....                                                                | 221 |
| Annexe A (informative) Comment utiliser le Modèle de l'Espace d'Adressage ..... |                                                                                               | 222 |
| A.1                                                                             | Vue d'ensemble .....                                                                          | 222 |
| A.2                                                                             | Définitions de type .....                                                                     | 222 |
| A.3                                                                             | Types d'Objets .....                                                                          | 222 |
| A.4                                                                             | Types de Variables .....                                                                      | 223 |
| A.4.1                                                                           | Généralités .....                                                                             | 223 |
| A.4.2                                                                           | Propriétés ou Variables de Données .....                                                      | 223 |
| A.4.3                                                                           | Variables nombreuses et / ou Types de Données structurés .....                                | 223 |
| A.5                                                                             | Vues .....                                                                                    | 224 |
| A.6                                                                             | Méthodes .....                                                                                | 224 |
| A.7                                                                             | Définition des Types de Références .....                                                      | 224 |
| A.8                                                                             | Définition des Règles de Modélisation .....                                                   | 225 |
| Annexe B (informative) Métamodèle d'OPC UA en langage UML .....                 |                                                                                               | 226 |
| B.1                                                                             | Contexte .....                                                                                | 226 |
| B.2                                                                             | Notation .....                                                                                | 227 |
| B.3                                                                             | Métamodèle .....                                                                              | 228 |
| B.3.1                                                                           | Base .....                                                                                    | 228 |
| B.3.2                                                                           | Type de Référence .....                                                                       | 229 |
| B.3.3                                                                           | Types de Références prédéfinis .....                                                          | 231 |
| B.3.4                                                                           | Attributs .....                                                                               | 232 |
| B.3.5                                                                           | Objet et Type d'Objet .....                                                                   | 234 |
| B.3.6                                                                           | Notificateur d'Événements .....                                                               | 235 |

|                                                                                                                |                                                  |     |
|----------------------------------------------------------------------------------------------------------------|--------------------------------------------------|-----|
| B.3.7                                                                                                          | Variable et Type de Variable .....               | 236 |
| B.3.8                                                                                                          | Méthode .....                                    | 238 |
| B.3.9                                                                                                          | Type de Données .....                            | 239 |
| B.3.10                                                                                                         | Vue .....                                        | 239 |
| Annexe C (normative) Système de Description du Type OPC Binaire .....                                          |                                                  | 241 |
| C.1                                                                                                            | Concepts .....                                   | 241 |
| C.2                                                                                                            | Description de Schéma .....                      | 243 |
| C.2.1                                                                                                          | Dictionnaire de Type .....                       | 243 |
| C.2.2                                                                                                          | Description de Type .....                        | 243 |
| C.2.3                                                                                                          | Type Opaque .....                                | 244 |
| C.2.4                                                                                                          | Type Enuméré .....                               | 244 |
| C.2.5                                                                                                          | Type Structuré .....                             | 244 |
| C.2.6                                                                                                          | Type de Champ .....                              | 245 |
| C.2.7                                                                                                          | Valeur Enumérée .....                            | 247 |
| C.2.8                                                                                                          | Poids d'Octet .....                              | 247 |
| C.2.9                                                                                                          | Directive d'Importation .....                    | 248 |
| C.3                                                                                                            | Descriptions de Types normalisés .....           | 248 |
| C.4                                                                                                            | Exemples de Descriptions de Types .....          | 248 |
| C.5                                                                                                            | Schéma XML OPC Binaire .....                     | 250 |
| C.6                                                                                                            | Dictionnaire de Type normalisé OPC Binaire ..... | 252 |
| Annexe D (normative) Notation graphique .....                                                                  |                                                  | 254 |
| D.1                                                                                                            | Généralités .....                                | 254 |
| D.2                                                                                                            | Notation .....                                   | 254 |
| D.2.1                                                                                                          | Vue d'ensemble .....                             | 254 |
| D.2.2                                                                                                          | Notation Simple .....                            | 254 |
| D.2.3                                                                                                          | Notation Etendue .....                           | 256 |
| Figure 1 – Diagrammes des Nœuds de l'Espace d'Adressage .....                                                  |                                                  | 132 |
| Figure 2 – Modèle d'Objet d'OPC UA .....                                                                       |                                                  | 135 |
| Figure 3 – Modèle de Nœud de l'Espace d'Adressage .....                                                        |                                                  | 136 |
| Figure 4 – Modèle de référence .....                                                                           |                                                  | 137 |
| Figure 5 – Exemple d'une Variable définie par un Type de Variable .....                                        |                                                  | 139 |
| Figure 6 – Exemple d'une Définition de Type Complexe .....                                                     |                                                  | 140 |
| Figure 7 – Objet et ses composants définis par un Type d'Objet .....                                           |                                                  | 142 |
| Figure 8 – Références Symétriques et Non Symétriques .....                                                     |                                                  | 150 |
| Figure 9 – Variables, Types de Variables et leurs Types de Données .....                                       |                                                  | 169 |
| Figure 10 – Modèle de Type de Données .....                                                                    |                                                  | 171 |
| Figure 11 – Exemple de modélisation de Type de Données .....                                                   |                                                  | 177 |
| Figure 12 – Sous-typage des Nœuds de Définition de Type .....                                                  |                                                  | 181 |
| Figure 13 – Hiérarchie de Déclaration d'Instance intégralement héritée pour le Type<br>Bêta183 .....           |                                                  | 183 |
| Figure 14 – Instance et son Nœud de Définition de Type .....                                                   |                                                  | 185 |
| Figure 15 – Exemple de plusieurs Références entre Déclarations d'Instance .....                                |                                                  | 186 |
| Figure 16 – Exemple de modification d'instances sur la base des Déclarations<br>d'Instance .....               |                                                  | 188 |
| Figure 17 – Exemple de modification de Déclarations d'Instance fondées sur une<br>Déclaration d'Instance ..... |                                                  | 189 |

|                                                                                                               |     |
|---------------------------------------------------------------------------------------------------------------|-----|
| Figure 18 – Utilisation de la Règle de Modélisation normalisée "Nouveau".....                                 | 190 |
| Figure 19 – Exemple d'utilisation des Règles de Modélisation normalisées "Facultative" et "Obligatoire" ..... | 191 |
| Figure 20 – Exemple d'utilisation de la Règle de Modélisation "ExposesItsArray".....                          | 192 |
| Figure 21 – Exemple complexe d'utilisation de la Règle de Modélisation "ExposesItsArray" .....                | 193 |
| Figure 22 – Exemple d'utilisation de la Règle de Modélisation "OptionalPlaceHolder" .....                     | 193 |
| Figure 23 – Exemple d'utilisation de la Règle de Modélisation "MandatoryPlaceHolder".....                     | 194 |
| Figure 24 – Hiérarchie d'un Type de Référence normalisé .....                                                 | 196 |
| Figure 25 – Exemple de Référence d'Événement .....                                                            | 202 |
| Figure 26 – Exemple de Référence à des Événements complexes .....                                             | 203 |
| Figure 27 – Hiérarchie d'un Type d'Événement normalisé .....                                                  | 213 |
| Figure 28 – Comportement d'un Serveur en Audit .....                                                          | 214 |
| Figure 29 – Comportement d'un Serveur de regroupement en Audit.....                                           | 216 |
| Figure B.1 – Contexte du Métamodèle d'OPC UA .....                                                            | 226 |
| Figure B.2 – Notation (I) .....                                                                               | 227 |
| Figure B.3 – Notation (II) .....                                                                              | 227 |
| Figure B.4 – Base .....                                                                                       | 229 |
| Figure B.5 – Référence et Type de Référence .....                                                             | 230 |
| Figure B.6 – Types de Références prédéfinis.....                                                              | 232 |
| Figure B.7 – Attributs .....                                                                                  | 234 |
| Figure B.8 – Objet et Type d'Objet.....                                                                       | 235 |
| Figure B.9 – Notificateur d'Événements .....                                                                  | 236 |
| Figure B.10 – Variable et Type de Variable.....                                                               | 237 |
| Figure B.11 – Méthode.....                                                                                    | 239 |
| Figure B.12 – Type de Données.....                                                                            | 239 |
| Figure B.13 – Vue.....                                                                                        | 240 |
| Figure C.1 – Structure du Dictionnaire d'OPC Binaire .....                                                    | 242 |
| Figure D.1 – Exemple d'une Référence reliant deux Nœuds.....                                                  | 255 |
| Figure D.2 – Exemple d'utilisation d'une Définition de Type dans un Nœud.....                                 | 257 |
| Figure D.3 – Exemple de présentation des Attributs.....                                                       | 257 |
| Figure D.4 – Exemple de présentation de Propriétés en ligne .....                                             | 258 |
| Tableau 1 – Conventions utilisées dans les Tableaux de Classes de Nœuds .....                                 | 133 |
| Tableau 2 – Classe de Nœud de base .....                                                                      | 146 |
| Tableau 3 – Masque de bit pour Masque d'Écriture et Masque d'Écriture Utilisateur .....                       | 148 |
| Tableau 4 – Classe de Nœud "Types de Références" .....                                                        | 149 |
| Tableau 5 – Classe de Nœud "Vues" .....                                                                       | 153 |
| Tableau 6 – Classe de Nœud "Objets" .....                                                                     | 155 |
| Tableau 7 – Classe de Nœud "ObjectType" .....                                                                 | 157 |
| Tableau 8 – Classe de Nœud "Variables".....                                                                   | 159 |
| Tableau 9 – Classe de nœud "VariableType" .....                                                               | 165 |
| Tableau 10 – Classe de Nœud "Méthodes" .....                                                                  | 167 |
| Tableau 11 – Classe de nœud "DataType".....                                                                   | 173 |

|                                                                                                       |     |
|-------------------------------------------------------------------------------------------------------|-----|
| Tableau 12 – Vue d'ensemble des Attributs .....                                                       | 178 |
| Tableau 13 – Hiérarchie de Déclaration d'Instance pour un Type Bêta .....                             | 181 |
| Tableau 14 – Hiérarchie de Déclaration d'Instance intégralement héritée pour un Type Bêta .....       | 182 |
| Tableau 15 – Règles applicables aux Propriétés des Règles de Modélisation en cas de Sous-typage ..... | 188 |
| Tableau 16 – Propriétés des Règles de Modélisation .....                                              | 190 |
| Tableau 17 – Définition d'un Identificateur de Nœud .....                                             | 204 |
| Tableau 18 – Valeurs du Type d'Identificateur .....                                                   | 205 |
| Tableau 19 – Valeurs Nulles d'Identificateur de Nœud .....                                            | 205 |
| Tableau 20 – Définition de Nom Qualifié .....                                                         | 206 |
| Tableau 21 – Exemples d'Identificateurs de Paramètre de lieu .....                                    | 206 |
| Tableau 22 – Définition de Texte Localisé .....                                                       | 207 |
| Tableau 23 – Définition des Arguments .....                                                           | 207 |
| Tableau 24 – Définition du Type de Données de Fuseau Horaire .....                                    | 209 |
| Tableau 25 – Valeurs du Type de Règle de Nommage .....                                                | 210 |
| Tableau 26 – Valeurs de Classe de Nœud .....                                                          | 210 |
| Tableau 27 – Définition du Type de Valeurs d'Énumération .....                                        | 211 |
| Tableau C.1 – Composants d'un Dictionnaire de Type .....                                              | 243 |
| Tableau C.2 – Composants d'une Description de Type .....                                              | 243 |
| Tableau C.3 – Composants de Type Opaque .....                                                         | 244 |
| Tableau C.4 – Composants de Type Enuméré .....                                                        | 244 |
| Tableau C.5 – Composants d'un Type Structuré .....                                                    | 245 |
| Tableau C.6 – Composants d'un Type de Champ .....                                                     | 246 |
| Tableau C.7 – Composants de Valeur Enumérée .....                                                     | 247 |
| Tableau C.8 – Composants d'une Directive d'Importation .....                                          | 248 |
| Tableau C.9 – Descriptions de Types normalisés .....                                                  | 248 |
| Tableau D.1 – Notation des Nœuds en fonction de la Classe de Nœud .....                               | 255 |
| Tableau D.2 – Notation Simple de Nœuds en fonction de la Classe de Nœud .....                         | 256 |

## COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

## ARCHITECTURE UNIFIEE OPC –

## Partie 3: Modèle de l'Espace d'Adressage

## AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. À cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

La Norme internationale IEC 62541-3 a été établie par le sous-comité 65E: Les dispositifs et leur intégration dans les systèmes de l'entreprise, du comité d'études 65 de l'IEC: Mesure, commande et automation dans les processus industriels.

Cette deuxième édition annule et remplace la première édition parue en 2010. Cette édition constitue une révision technique.

Cette édition inclut les modifications techniques majeures suivantes par rapport à l'édition précédente:

- a) Ajout de règles de sous-typage pour les énumérations en 8.14 (numéro d'édition 0606);
- b) Ajout de la *Propriété EnumValues (Valeurs d'Énumération)* en 5.8.3 pour prendre en charge la représentation Entier des énumérations qui ne sont pas basées sur zéro ou qui présentent des espaces (numéro d'édition 0876);

- c) Ajout de la *Propriété ValueAsText (Valeur comme Texte)* en 5.6.2 fournissant une représentation du texte localisé des valeurs d'énumération (numéro d'édition 0951);
- d) Ajout du *Type d'Événement SystemStatusChange (Changement de Statut du Système)* en 9.31 qui peut être utilisé pour indiquer la perte de la connexion au sous-système (numéro d'édition 1255);
- e) Ajout des *Propriétés MaxArrayLength (Longueur de Matrice Maximale)* et *MaxStringLength (Longueur de Chaîne Maximale)* en 5.6.2 pour identifier la longueur maximale admise d'une valeur de chaîne et de matrice pour les clients écrivant des valeurs (numéro d'édition 1547);
- f) Suppression du concept de *Modèle Parent (ModelParent)* du document du fait qu'il n'est plus d'usage. L'*Identificateur de Nœud du Type de Référence* est conservé pour ne pas interrompre les applications existantes (numéro d'édition 1554).
- g) Ajout du *Type d'Événement ProgressEventType (Type d'Événement de Progrès)* en 9.4 pour identifier la progression d'une opération, comme une invocation de service (numéro d'édition 1557);
- h) Indication en 8.38 qu'il est admis d'utiliser le TAI (Temps Atomique International – International Atomic Time) au lieu du temps UTC afin d'éviter les problèmes dus aux secondes intercalaires (numéro d'édition 1563);
- i) Ajout de la *Propriété EngineeringUnits (Unités Techniques)* en 5.6.2 telle qu'utilisée dans l'IEC 62541-8 (numéro d'édition 1749);
- j) Ajout des *Règles de Modélisation OptionalPlaceholder (Espace Réservé Facultatif)* et *MandatoryPlaceholder (Espace Réservé Obligatoire)* en 6.4.4.5.5 et 6.4.4.5.6 (numéro d'édition 1804).

Le texte de cette norme est issu des documents suivants:

| CDV         | Rapport de vote |
|-------------|-----------------|
| 65E/374/CDV | 65E/402/RVC     |

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette norme.

Cette publication a été rédigée selon les Directives ISO/IEC, Partie 2.

Une liste de toutes les parties de la série IEC 62541, publiées sous le titre général *Architecture unifiée OPC*, peut être consultée sur le site web de l'IEC.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous "<http://webstore.iec.ch>" dans les données relatives à la publication recherchée. À cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

**IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.**

## ARCHITECTURE UNIFIEE OPC –

### Partie 3: Modèle de l'Espace d'Adressage

#### 1 Domaine d'application

La présente partie de l'IEC 62541 décrit l'*Espace d'Adressage (AddressSpace)* de l'Architecture Unifiée OPC (OPC UA) ainsi que les *Objets* correspondants. La présente partie de l'IEC 62541 est le métamodèle OPC UA sur lequel se fondent les modèles d'informations OPC UA.

#### 2 Références normatives

Les documents suivants sont cités en référence de manière normative, en intégralité ou en partie, dans le présent document et sont indispensables pour son application. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC TR 62541-1, *OPC Unified Architecture – Part 1: Overview and Concepts* (disponible en anglais seulement)

IEC 62541-4<sup>1</sup>, *Architecture unifiée OPC – Partie 4: Services*

IEC 62541-5:–<sup>1</sup>, *Architecture unifiée OPC – Partie 5: Modèle d'Information*

IEC 62541-6<sup>1</sup>, *Architecture unifiée OPC – Partie 6: Correspondances*

IEC 62541-8<sup>1</sup>, *Architecture unifiée OPC – Partie 8: Accès aux données*

IEC 62541-11<sup>1</sup>, *OPC Unified Architecture – Part 11: Historical Access* (disponible en anglais seulement)

ISO/IEC 10918-1, *Technologies de l'information – Compression numérique et codage des images fixes de nature photographique: Prescriptions et lignes directrices*

ISO/IEC 15948, *Technologies de l'information – Infographie et traitement d'images – Graphiques de réseau portables (PNG): Spécification fonctionnelle*

ISO 639 (toutes les parties), *Codes pour la représentation des noms de langue*

ISO 3166 (toutes les parties), *Codes pour la représentation des noms de pays et de leurs subdivisions*

IEEE 754-1985, *IEEE Standard for Binary Floating-Point Arithmetic*,  
<http://ieeexplore.ieee.org/servlet/opac?punumber=2355> (disponible en anglais seulement)

IETF RFC 3066, *Tags for the Identification of Languages*, <http://tools.ietf.org/html/rfc3066> (disponible en anglais seulement)

---

<sup>1</sup> À publier.

Partie 1 du Schéma XML, <http://www.w3.org/TR/xmlschema-1/> (disponible en anglais seulement)

Partie 2 du Schéma XML, <http://www.w3.org/TR/xmlschema-2/> (disponible en anglais seulement)

XPATH, <http://www.w3.org/TR/xpath/> (disponible en anglais seulement)

### 3 Termes, définitions, abréviations et conventions

#### 3.1 Termes et définitions

Pour les besoins du présent document, les termes et définitions donnés dans l'IEC TR 62541-1 ainsi que les suivants s'appliquent.

##### 3.1.1

##### **DataType (Type de Données)**

instance d'un *DataType Node* (*Nœud de Type de Données*) qui est utilisée avec l'attribut *ValueRank* (*Rang de Valeur*) pour définir le type de données d'une *Variable*

##### 3.1.2

##### **DataTypeId (Identificateur de Type de Données)**

*NodeId* (identificateur de Nœud) d'un *DataType Node* (*Nœud de Type de Données*)

##### 3.1.3

##### **DataVariable (Variable de Données)**

*variables* qui représentent des *valeurs* d'*Objets*, que ce soit directement ou indirectement pour des *Variables* complexes, les *Variables* étant toujours le *TargetNode* (*Nœud Cible*) d'une *Référence "HasComponent"* (*Dispose d'un composant*)

##### 3.1.4

##### **EventType (Type d'Événement)**

*ObjectType Node* (*Nœud de Type d'Objet*) représentant la définition du type d'un *Événement* donné

##### 3.1.5

##### **référence hiérarchique**

*référence* qui est utilisée pour construire des hiérarchies dans *AddressSpace* (*Espace d'Adressage*)

Note 1 à l'article: Tous les Types de Références hiérarchiques sont issus des Références Hiérarchiques.

##### 3.1.6

##### **InstanceDeclaration (Déclaration d'Instance)**

*nœud* utilisé par un *TypeDefinitionNode* (*Nœud de définition de type*) complexe pour présenter la complexité de sa structure

Note 1 à l'article: Cette instance est utilisée par une définition de type.

##### 3.1.7

##### **ModellingRule (Règle de Modélisation)**

métadonnée d'une *Déclaration d'Instance* qui définit la manière dont la *Déclaration d'Instance* sera utilisée pour l'instanciation et définit également les règles de sous-typage d'une *Déclaration d'Instance*

##### 3.1.8

##### **Property (Propriété)**

*variables* qui constituent le *Nœud Cible* d'une *Référence "HasProperty"*

Note 1 à l'article: Les *Propriétés* décrivent les caractéristiques d'un *Nœud*.

### 3.1.9

#### SourceNode (Nœud Source)

*nœud* qui sert de *Référence* à un autre *Nœud*

EXEMPLE Dans la *Référence* "A contient B", "A" est le *Nœud Source*.

### 3.1.10

#### TargetNode (Nœud Cible)

*nœud* qui est référencé par un autre *Nœud*

EXEMPLE Dans la *Référence* "A contient B", "B" est le *Nœud Cible*.

### 3.1.11

#### TypeDefinitionNode (Nœud de Définition de Type)

*nœud* utilisé pour définir le type d'un autre *Nœud*

Note 1 à l'article: Les *Nœuds Type d'Objet* et *Type de Variable* sont des *Nœuds de Définition de Type*.

### 3.1.12

#### VariableType (Type de Variable)

*nœud* représentant la définition du type d'une *Variable*

## 3.2 Abréviations

|     |                                                                                  |
|-----|----------------------------------------------------------------------------------|
| UA  | Unified Architecture (Architecture unifiée)                                      |
| UML | Unified Modeling Language (Langage de modélisation unifié)                       |
| URI | Uniform Resource Identifier (Identificateur de ressources uniformes)             |
| W3C | World Wide Web Consortium (consortium chargé de la normalisation du web mondial) |
| XML | eXtensible Markup Language (Langage de balisage extensible)                      |

## 3.3 Conventions

### 3.3.1 Conventions pour les figures Espace d'Adressage

Les *Nœuds* et leurs *Références* respectives sont illustrés par des figures. La Figure 1 représente les conventions utilisées dans ces figures.

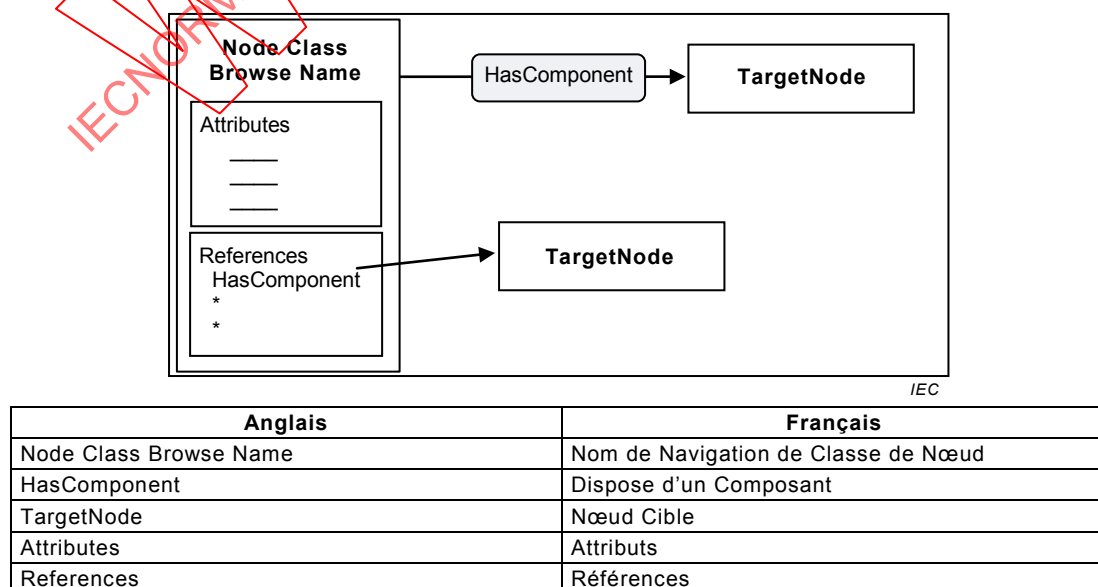


Figure 1 – Diagrammes des Nœuds de l'Espace d'Adressage

Sur ces figures, les rectangles représentent les *Nœuds*. L'intitulé des rectangles représentant les *Nœuds* peut être donné par une ou deux lignes de texte. Lorsque deux lignes sont utilisées dans le rectangle, la première identifie la *Classe de Nœud* et la seconde donne le *Nom de Navigation*. Lorsqu'une seule ligne est utilisée, elle contient le *Nom de Navigation*.

Les rectangles représentant les *Nœuds* peuvent inclure des cases qui définissent leurs *Attributs* et les *Références*. Des dénominations spécifiques dans ces cases identifient des *Attributs* et des *Références* spécifiques.

Des rectangles grisés aux bords arrondis et traversés par des flèches représentent des *Références*. La flèche part du *Nœud Source* (*SourceNode*) et pointe vers le *Nœud Cible* (*TargetNode*). Les *Références* peuvent également être illustrées en traçant une flèche partant de la dénomination de la *Référence* dans la case "Références" et aboutissant au *Nœud Cible*.

### 3.3.2 Conventions pour la définition des Classes de Nœuds

L'Article 5 définit les *Classes de Nœuds* de l'*Espace d'Adressage*. Le Tableau 1 décrit le format des tableaux utilisés pour définir les *Classes de Nœuds*.

**Tableau 1 – Conventions utilisées dans les Tableaux de Classes de Nœuds**

| Dénomination                   | Usage                          | Type de Données                 | Description                                                                |
|--------------------------------|--------------------------------|---------------------------------|----------------------------------------------------------------------------|
| <b>Attributs</b>               |                                |                                 |                                                                            |
| "Dénomination de l'attribut"   | "M" ou "O"                     | Type de Données de l'Attribut   | Définit l'Attribut.                                                        |
| <b>Références</b>              |                                |                                 |                                                                            |
| "Dénomination de la référence" | "1", "0..1" ou "0..*" ou "0.." | Non utilisé                     | Décrit l'utilisation de la <i>Référence</i> par la <i>Classe de Nœud</i> . |
| <b>Propriétés normalisées</b>  |                                |                                 |                                                                            |
| "Dénomination de la Propriété" | "M" ou "O"                     | Type de Données de la Propriété | Définit la Propriété.                                                      |

La colonne Dénomination donne le nom de l'Attribut, le nom du Type de Référence utilisé pour créer une Référence ou le nom d'une Propriété référencée par la Référence "HasProperty" (Dispose d'une Propriété).

La colonne Usage définit le caractère obligatoire (M pour Mandatory) ou facultatif (O pour Optional) de l'Attribut ou de la Propriété. Lorsqu'il est obligatoire, l'Attribut ou la Propriété doit être présent pour tout Nœud de la Classe de Nœud. Pour les Références, cette colonne précise la cardinalité. Les valeurs suivantes peuvent s'appliquer:

- "0..\*" indique une absence totale de restrictions; en d'autres termes il n'est pas nécessaire de fournir la Référence, mais il n'y a par ailleurs aucune limite quant au nombre de fois où elle est fournie;
- "0..1" signifie que la Référence est fournie une fois au plus;
- "1" signifie que la Référence doit être fournie une fois exactement.

La colonne Type de Données donne la dénomination du Type de Données de l'Attribut ou de la Propriété. Elle n'est pas utilisée pour les Références.

La colonne Description décrit l'Attribut, la Référence ou la Propriété.

Seule la présente norme peut définir des Attributs. De ce fait, tous les Attributs de la Classe de Nœud sont spécifiés dans le tableau et ne peuvent être étendus que par d'autres parties de la série IEC 62541.

La présente norme définit également des *Types de Références*; ces derniers ne peuvent cependant être spécifiés que par un *Serveur* ou par un *Client* au moyen des *Services de Gestion de Nœuds* décrits dans l'IEC 62541-4. Ainsi, les tableaux de *Classes de Nœuds* de la présente norme peuvent contenir les *Types de Références* de base, appelés *Références*, indiquant de ce fait que tout *Type de Référence* peut être utilisé pour la *Classe de Nœud*, y compris des *Types de Références* spécifiques au système. Les tableaux de *Classes de Nœuds* spécifient uniquement la manière dont les *Classes de Nœuds* peuvent être utilisées en tant que *Nœuds Sources de Références*, et non en tant que *Nœuds Cibles*. Si un tableau de *Classe de Nœud* permet l'utilisation d'un *Type de Référence* de sa *Classe de Nœud* comme *Nœud Source*, ceci est également vrai pour des sous-types du *Type de Référence*. Des sous-types du *Type de Référence* peuvent toutefois restreindre ses *Nœuds Sources*.

La présente norme définit des *Propriétés*, mais d'autres organismes de normalisation ou des fournisseurs peuvent aussi définir des *Propriétés*; il est également possible d'avoir des *Nœuds* dont les *Propriétés* ne sont pas normalisées. Les *Propriétés* établies dans la présente norme sont définies par leur dénomination, qui est mise en correspondance avec le *Nom de Navigation d'Indice de l'espace de Nom 0*; celui-ci représente l'*Espace de Nom* pour l'OPC UA.

La colonne Usage (facultatif ou obligatoire) n'implique pas une *Règle de Modélisation* spécifique pour modéliser les *Propriétés*. Chaque application de *Serveur* particulier choisira d'utiliser les *Règles de Modélisation* qui lui conviennent.

## 4 Concepts de l'Espace d'Adressage

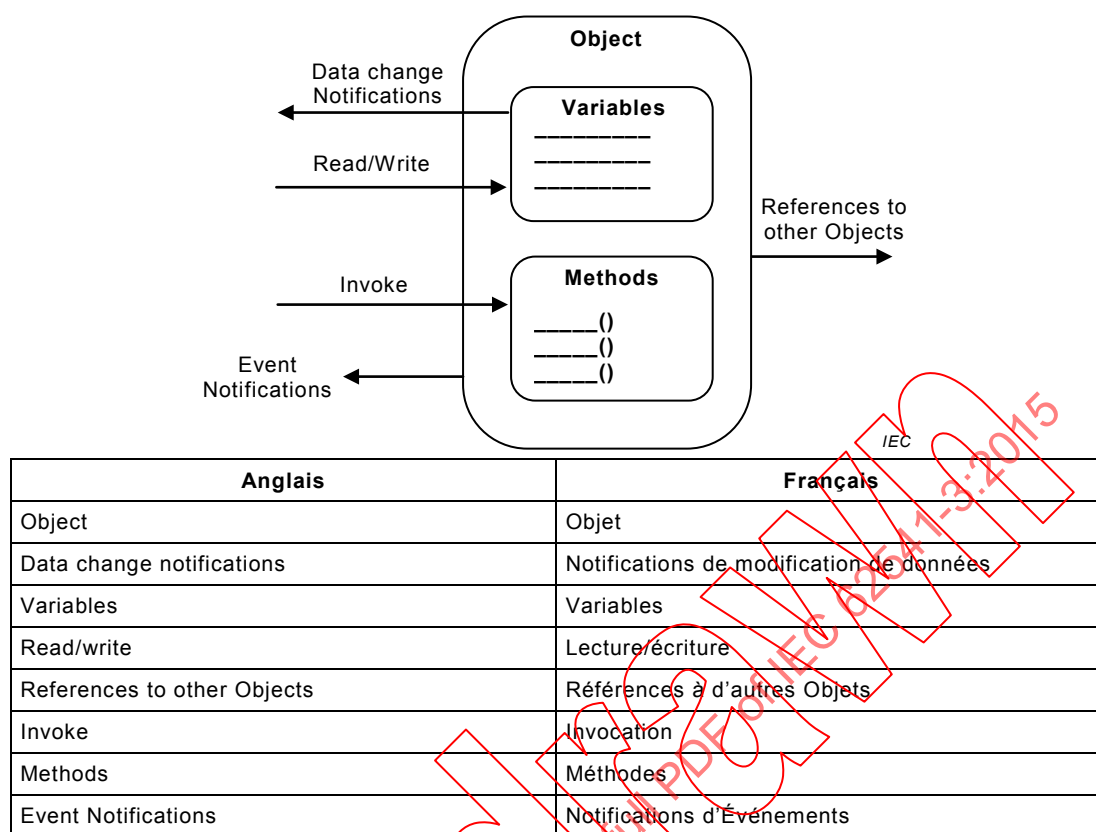
### 4.1 Vue d'ensemble

Les paragraphes suivants de l'Article 4 définissent les concepts de l'*Espace d'Adressage*. L'Article 5 définit les *Classes de Nœuds* de l'*Espace d'Adressage* représentant les concepts de l'*Espace d'Adressage*. L'Article 6 décrit en détail le modèle type des *Types d'Objets* et des *Types de Variables*. Les Articles 7 à 9 définissent des types normalisés de *Types de Références*, *Types de Données* et *Types d'Événements*.

L'Annexe A informative donne des considérations d'ordre général quant à la manière d'utiliser le Modèle de l'Espace d'Adressage et l'Annexe B informative fournit un modèle en UML du Modèle de l'Espace d'Adressage. L'Annexe C normative définit le système de description de types OPC binaire comme un format permettant de spécifier des structures de types de données et l'Annexe D normative définit une notation graphique des données d'OPC UA.

### 4.2 Modèle d'objet

L'objectif premier de l'*Espace d'Adressage* d'OPC UA est de mettre à disposition une méthode normalisée permettant aux *Serveurs* de représenter des *Objets* pour leurs *Clients*. C'est dans ce but qu'a été conçu le Modèle d'Objet d'OPC UA. Il définit des *Objets* en termes de *Variables* et de *Méthodes*. Il permet également d'exprimer les relations entre *Objets*. Ce modèle est illustré en Figure 2.



**Figure 2 – Modèle d'Objet d'OPC UA**

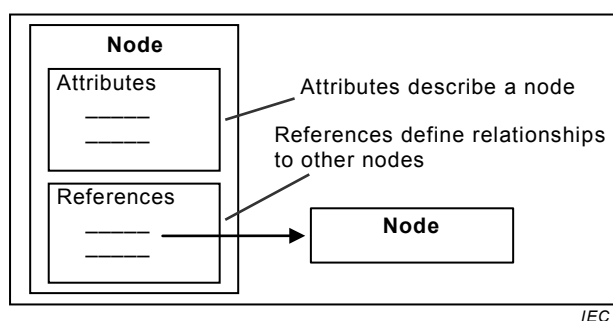
Dans l'*Espace d'Adressage* les éléments de ce modèle sont représentés comme des *Nœuds*. Chaque *Nœud* est attribué à une *Classe de Nœud* et chaque *Classe de Nœud* représente un élément différent du Modèle d'Objet. L'Article 5 définit les *Classes de Nœuds* utilisées pour représenter ce modèle.

### 4.3 Modèle de Nœud

#### 4.3.1 Généralités

Le jeu d'*Objets* et les informations correspondantes que le *Serveur* OPC UA met à la disposition des *Clients* est appelé son *Espace d'Adressage*. Le modèle utilisé à cet effet est défini par le Modèle d'Objet d'OPC UA (voir 4.2).

Les *Objets* et leurs composants sont représentés dans l'*Espace d'Adressage* comme un jeu de *Nœuds* décrits par des *Attributs* et interconnectés par des *Références*. La Figure 3 illustre le modèle d'un *Nœud* et 4.3 en décrit les détails.



| Anglais                                        | Français                                                     |
|------------------------------------------------|--------------------------------------------------------------|
| Node                                           | Nœud                                                         |
| Attributes                                     | Attributs                                                    |
| Attributes describe a node                     | Les Attributs décrivent un nœud                              |
| References define relationships to other nodes | Les Références définissent des relations avec d'autres nœuds |
| References                                     | Références                                                   |

Figure 3 – Modèle de Nœud de l'Espace d'Adressage

#### 4.3.2 Classes de Nœuds

Les *Classes de Nœuds* sont définies en termes d'*Attributs* et de *Références* qui doivent être instanciés (recevoir des valeurs) lorsqu'un *Nœud* est défini dans l'*Espace d'Adressage*. Les *Attributs* sont traités en 4.3.3 et les *Références* en 4.3.4.

L'Article 5 définit les *Classes de Nœuds* utilisées pour l'*Espace d'Adressage* d'OPC UA. Ces *Classes de Nœuds* sont collectivement appelées les métadonnées de l'*Espace d'Adressage*. Chaque *Nœud* dans l'*Espace d'Adressage* est une instance de l'une de ces *Classes de Nœuds*. Aucune autre *Classe de Nœud* ne doit être utilisée pour définir des *Nœuds* et, par conséquent, les *Clients* et les *Serveurs* ne sont pas autorisés à définir des *Classes de Nœuds* ou à étendre les définitions de ces *Classes de Nœuds*.

#### 4.3.3 Attributs

Les *Attributs* sont des éléments de données qui décrivent des *Nœuds*. Les *Clients* peuvent accéder à des valeurs d'*Attributs* en utilisant des *Services* de Lecture, d'Écriture, de Requête et d'Abonnement/Élément surveillé. Ces *Services* sont définis dans l'IEC 62541-4.

Les *Attributs* sont des composants élémentaires de *Classes de Nœuds*. Les définitions des *Attributs* font partie intégrante de celles des *Classes de Nœuds* de l'Article 5 et, par conséquent, elles ne sont pas incluses dans l'*Espace d'Adressage*.

Chaque définition d'*Attribut* comprend un identificateur d'attribut (pour les identificateurs d'*Attributs*, voir l'IEC 62541-6), une dénomination, une description, un type de données et un indicateur obligatoire/facultatif. L'ensemble d'*Attributs* défini pour chaque *Classe de Nœud* ne doit pas être étendu par les *Clients* ou les *Serveurs*.

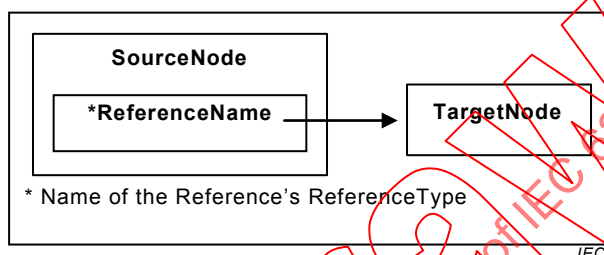
Lorsqu'un *Nœud* est instancié dans l'*Espace d'Adressage*, les valeurs des *Attributs* de la *Classe de Nœud* sont fournies. L'indicateur obligatoire/facultatif de l'*Attribut* indique s'il est nécessaire d'instancier l'*Attribut*.

#### 4.3.4 Références

Les *Références* sont utilisées pour corréler les *Nœuds* entre eux. Leur accès est possible en utilisant les *Services* navigation et interrogation définis dans l'IEC 62541-4.

De la même manière que les *Attributs*, elles sont définies comme des composants fondamentaux des *Nœuds*. Contrairement aux *Attributs*, les *Références* sont définies comme des instances des *Nœuds de Types de Références*. Les *Nœuds de Types de Références* sont visibles dans l'*Espace d'Adressage* et sont définis en utilisant la *Classe de Nœud "Types de Références"* (voir 5.3).

Le *Nœud* qui contient la *Référence* est appelé *SourceNode* (*Nœud Source*) et le *Nœud* qui est référencé est appelé *TargetNode* (*Nœud Cible*). L'association du *SourceNode*, du *Type de Référence* et du *TargetNode* est utilisée par les *Services OPC UA* pour identifier les *Références* de manière unique. Ainsi, chaque *Nœud* peut référencer une seule fois un autre *Nœud* avec le même *Type de Référence*. Tous les sous-types de *Types de Références* concrets sont considérés égaux aux *Types de Références* concrets de base lorsqu'il s'agit d'identifier des *Références* (voir 5.3 pour les sous-types de *Types de Références*). Ce modèle de *Référence* est illustré en Figure 4.



| Anglais                               | Français                                          |
|---------------------------------------|---------------------------------------------------|
| SourceNode                            | Nœud Source                                       |
| ReferenceName                         | Nom de référence                                  |
| TargetNode                            | Nœud cible                                        |
| Name of the Reference's ReferenceType | Dénomination du Type de Référence de la Référence |

**Figure 4 – Modèle de référence**

Le *TargetNode* d'une *Référence* peut être situé dans le même *Espace d'Adressage* ou dans l'*Espace d'Adressage* d'un autre *Serveur OPC UA*. Les *TargetNodes* qui se trouvent dans d'autres *Serveurs* sont identifiés dans les *Services OPC UA* en utilisant une combinaison de la dénomination du *Serveur* distant et de l'identificateur attribué au *Nœud* par le *Serveur* distant.

L'OPC UA n'exige pas que le *TargetNode* existe, ainsi les *Références* peuvent désigner un *Nœud* qui n'existe pas.

## 4.4 Variables

### 4.4.1 Généralités

Les *Variables* sont utilisées pour représenter des *valeurs*. Il est défini deux types de *Variables*: les *Propriétés* et les *Variables de Données*. Elles diffèrent par le type de données qu'elles représentent et par le fait qu'elles contiennent ou non d'autres *Variables*.

### 4.4.2 Propriétés

Les *Propriétés* sont des caractéristiques d'*Objets*, de *Variables de Données* et d'autres *Nœuds* définis par le *Serveur*. Les *Propriétés* sont différentes des *Attributs* du fait qu'elles caractérisent ce que représente le *Nœud*, par exemple un appareil ou un bon de commande. Les *Attributs* définissent des métadonnées supplémentaires qui sont instanciées pour tous les *Nœuds* à partir d'une *Classe de Nœud*. Les *Attributs* sont communs à tous les *Nœuds* d'une *Classe de Nœud* donnée et ne sont définis que par cette spécification tandis que les *Propriétés* peuvent être définies par le *Serveur*.

Par exemple, un *Attribut* définit le *Type de Données* des *Variables* tandis qu'une *Propriété* peut être utilisée pour spécifier l'unité technique de certaines *Variables*.

Pour éviter les répétitions, il n'est pas admis de définir des *Propriétés* par d'autres *Propriétés*. Pour identifier facilement des *Propriétés*, le *Nom de Navigation* d'une *Propriété* doit être unique dans le contexte d'un *Nœud* contenant les *Propriétés* (pour plus de détails, voir 5.6.3).

Un *Nœud* et ses *Propriétés* doivent toujours résider dans le même *Serveur*.

#### 4.4.3 Variables de Données

Les *Variables de Données* représentent le contenu d'un *Objet*. Il peut être par exemple défini par un *Objet* fichier qui contient un train d'octets. Le train d'octets peut être défini comme une *Variable de Données* qui est une suite d'octets. Les *Propriétés* peuvent être utilisées pour présenter la date de création et le propriétaire de l'*Objet* fichier.

Par exemple, si une *Variable de Données* est définie par une structure de données qui contient deux champs "heure de début" et "heure de fin", elle peut alors disposer d'une *Propriété* spécifique à cette structure de données comme "heure de début au plus tôt".

Comme autre exemple, des blocs fonctionnels des systèmes de commande peuvent être représentés comme des *Objets*. Les paramètres du bloc fonctionnel tels que ses points de consigne, peuvent être représentés comme des *Variables de Données*. L'*Objet* bloc fonctionnel peut également disposer de *Propriétés* décrivant son temps d'exécution et son type.

Les *Variables de Données* peuvent avoir des *Variables de Données* supplémentaires, mais uniquement si elles sont complexes. Dans ce cas, leurs *Variables de Données* doivent toujours être des éléments de définition complexe. En suivant l'exemple donné dans la description des *Propriétés* en 4.4.2, le *Serveur* peut présenter l'"heure de début" et l'"heure de fin" en tant que composants séparés de la structure de données.

Comme autre exemple, une *Variable de Données* complexe peut définir un ensemble de valeurs de température générées par trois transmetteurs de température séparés mais tous visibles dans l'*Espace d'Adressage*. Dans ce cas, cette *Variable de Données* complexe peut, à partir de là, définir des *Références "HasComponent"* (*Dispose d'un composant*) pour les différentes valeurs de température dont elle est constituée.

### 4.5 TypeDefinitionNodes (Nœuds de Définition de Type)

#### 4.5.1 Généralités

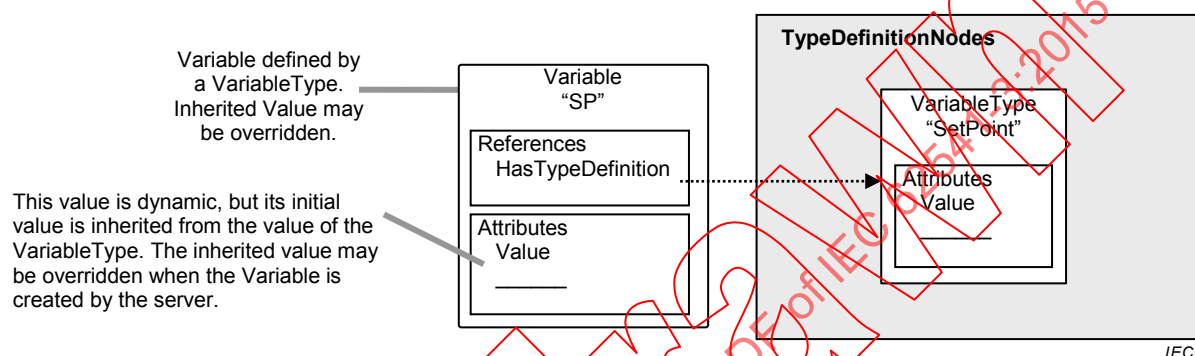
Les *Serveurs* OPC UA doivent fournir des définitions de type pour les *Objets* et les *Variables*. La *Référence "HasTypeDefinition"* (*Dispose d'une Définition de Type*) doit être utilisée pour relier une instance à sa définition de type représentée par un *TypeDefinitionNode*. Les définitions de type sont cependant exigées et l'IEC 62541-5 définit un *Type d'Objet de Base*, un *Type de Propriété* et un *Type de Variable de Données de Base* de sorte qu'un *Serveur* puisse utiliser un type de base si aucune information de type plus spécialisée n'est disponible. Les *Objets* et *Variables* héritent des *Attributs* spécifiés par leur *TypeDefinitionNodes* (pour plus de détails, voir 6.4).

Dans certains cas, l'*Identificateur de Nœud (NodeId)* utilisé par la *Référence "HasTypeDefinition"* est bien connu des *Clients* et des *Serveurs*. Les différents organismes peuvent définir des *Nœuds de Définition de Type* bien connus dans l'industrie. Les *Identificateurs de Nœuds* des *TypeDefinitionNodes* bien connus permettent de généraliser ce terme sur l'ensemble des *Serveurs* OPC UA et les *Clients* peuvent ainsi interpréter le *TypeDefinitionNode* sans avoir à le lire à partir du *Serveur*. Par conséquent, les *Serveurs* peuvent utiliser des *Identificateurs de Nœuds* bien connus sans avoir à représenter les *TypeDefinitionNodes* correspondants dans leur *Espace d'Adressage*. Cependant, pour les

*Clients* génériques, les *TypeDefinitionNodes* doivent être fournis. Ces *TypeDefinitionNodes* peuvent exister dans un autre *Serveur*.

L'exemple suivant, illustré en Figure 5, décrit l'utilisation d'une *Référence "Dispose d'une Définition de Type"*. Dans cet exemple, un paramètre de point de consigne "PC" est représenté par une *DataVariable* (*Variable de Données*) dans l'*Espace d'Adressage*. Cette *DataVariable* fait partie d'un *Objet* qui n'est pas illustré sur la figure.

Pour fournir une définition de point de consigne commune qui peut être utilisée par d'autres *Objets*, on emploie un *Type de Variable* spécialisé. Chaque *DataVariable* de point de consigne qui utilise cette définition commune aura une *Référence "Dispose d'une Définition de Type"* qui identifie le *Type de Variable* du "point de consigne" commun.



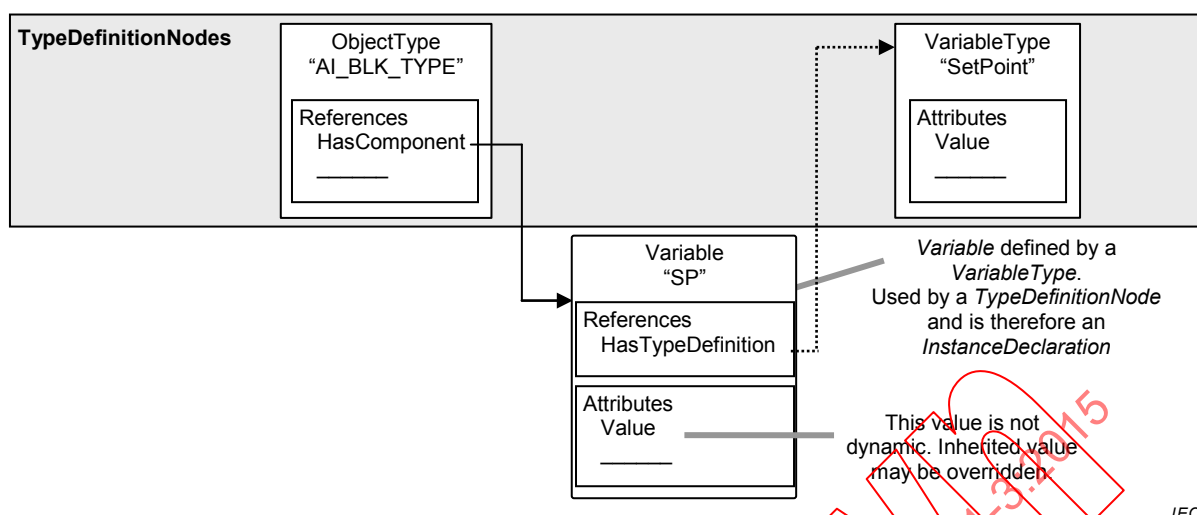
| Anglais                                                                                                                                                                         | Français                                                                                                                                                                                        |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TypeDefinitionNodes                                                                                                                                                             | Nœuds de Définition de Type                                                                                                                                                                     |
| Variable defined by a VariableType. Inherited value may be overridden.                                                                                                          | Variable définie par un Type de Variable. La valeur héritée peut être écrasée et remplacée.                                                                                                     |
| Variable "SP"                                                                                                                                                                   | Variable PC                                                                                                                                                                                     |
| References HasTypeDefinition                                                                                                                                                    | Références "Dispose d'une Définition de Type"                                                                                                                                                   |
| VariableType "SetPoint"                                                                                                                                                         | Type de Variable "Point de Consigne"                                                                                                                                                            |
| This value is dynamic, but its initial value is inherited from the value of the VariableType. The inherited value may be overridden when the Variable is created by the server. | Cette valeur est dynamique mais sa valeur initiale est héritée de la valeur du Type de Variable. La valeur héritée peut être écrasée et remplacée lorsque la Variable est créée par le serveur. |
| Attributes Value                                                                                                                                                                | Attributs Valeur                                                                                                                                                                                |

Figure 5 – Exemple d'une Variable définie par un Type de Variable

#### 4.5.2 TypeDefinitionNodes complexes et leurs Déclarations d'Instance

Les *TypeDefinitionNodes* peuvent être complexes. Un *TypeDefinitionNode* complexe définit également des *Références* à d'autres *Nœuds* dans le cadre de la définition de type. Les *Règles de Modélisation* définies en 6.4.4 précisent la manière dont ces *Nœuds* sont traités pour la création d'une instance de la définition de type.

Un *TypeDefinitionNode* référence des instances plutôt que d'autres *TypeDefinitionNodes* (*Nœuds de Définition de Type*), afin que des noms uniques soient utilisés pour plusieurs instances du même type, pour définir des valeurs par défaut et ajouter des *Références* à d'autres instances spécifiques à ce *TypeDefinitionNode* complexe et non au *TypeDefinitionNode* de l'instance. Par exemple, dans la Figure 6, le *Type d'Objet* "AI\_BLK\_TYPE", qui représente un bloc fonctionnel, dispose d'une *Référence "Dispose d'un composant"* à une *Variable* "PC" du *Type de Variable* "Point de Consigne". Le "AI\_BLK\_TYPE" peut avoir une *Variable* de point de consigne supplémentaire du même type utilisant un nom différent. Il peut être ajouté à la *Variable* une *Propriété* qui n'était pas définie par son *TypeDefinitionNode* "Point de Consigne". Et il peut également définir une valeur par défaut pour "PC", de sorte que chaque instance de "AI\_BLK\_TYPE" aurait une *Variable* "PC" initialement réglée sur cette valeur.



IEC

| Anglais                                                                                                                        | Français                                                                                                                                              |
|--------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| TypeDefinitionNodes                                                                                                            | Nœuds de Définition de Type                                                                                                                           |
| ObjectType                                                                                                                     | Type d'Objet                                                                                                                                          |
| VariableType                                                                                                                   | Type de Variable                                                                                                                                      |
| References HasComponent                                                                                                        | Références «Dispose d'un composant»                                                                                                                   |
| Attributes Value                                                                                                               | Attributs Valeur                                                                                                                                      |
| Variable "SP"                                                                                                                  | Variable "PC"                                                                                                                                         |
| References HasTypeDefinition                                                                                                   | Références "Dispose d'une Définition de Type"                                                                                                         |
| Variable defined by a <i>VariableType</i> . Used by a <i>TypeDefinitionNode</i> and is therefore an <i>InstanceDeclaration</i> | Variable définie par un <i>Type de Variable</i> . Utilisée par un <i>TypeDefinitionNode</i> et est par conséquent une <i>Déclaration d'Instance</i> . |
| This value is not dynamic. Inherited value may be overridden.                                                                  | Cette valeur n'est pas dynamique. La valeur héritée peut être écrasée et remplacée.                                                                   |

**Figure 6 – Exemple d'une Définition de Type Complexe**

Cette approche est souvent utilisée dans les langages de programmation orientés objet qui définissent les variables d'une classe comme des instances d'autres classes. Lorsque la classe est instanciée, chaque variable est également instanciée, mais en utilisant les valeurs par défaut (valeurs du constructeur) définies pour la classe contenant. En d'autres termes, de manière générale, le constructeur de la classe composante vient en premier, suivi par le constructeur de la classe contenant. Un constructeur de classe contenant peut prendre le pas sur les valeurs de composants définis par la classe composante.

Afin de faire la différence entre les instances utilisées pour les définitions de types et celles qui représentent des données réelles, ces instances sont appelées *Déclarations d'Instance*. Ce terme est cependant utilisé pour simplifier la présente spécification; on ne peut savoir si une instance est ou n'est pas une *Déclaration d'Instance* que dans l'*Espace d'Adressage*, en suivant ses *Références*. Certaines instances peuvent être partagées et par conséquent référencées par des *Nœuds de Définition de Type*, des *Déclarations d'Instance* et des instances. Cette méthodologie est similaire aux variables de classes dans les langages de programmation orientés objet.

#### 4.5.3 Sous-typage

La présente norme permet l'utilisation de sous-types de définitions de types. Les règles de sous-typage sont définies à l'Article 6. Le sous-typage de *Types d'Objets* et *Types de Variables* permet:

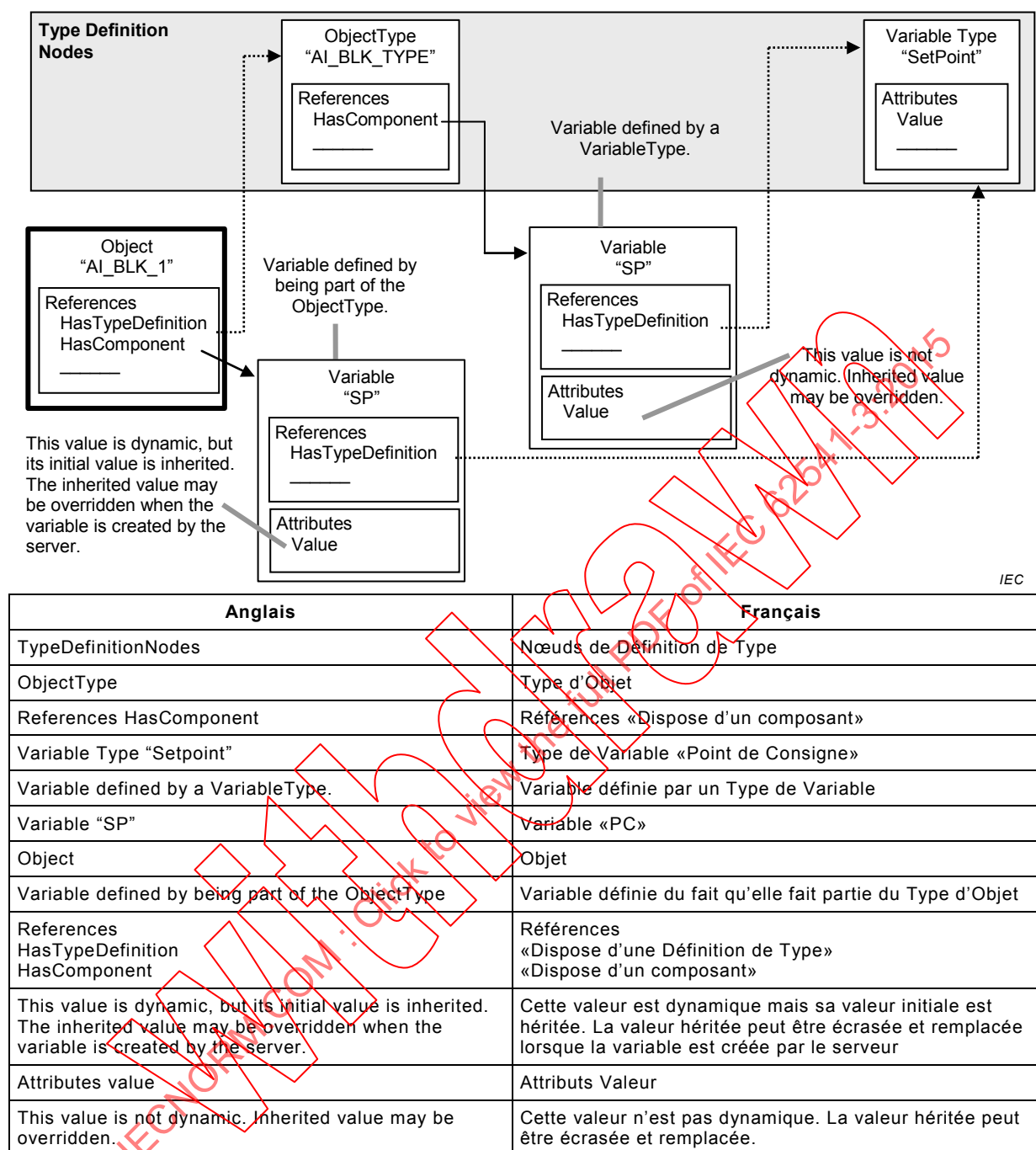
- aux *Clients* qui connaissent uniquement le supertype, de traiter une instance du sous-type comme s'il s'agissait d'une instance du supertype;
- de remplacer des instances du supertype par des instances du sous-type;

- d'utiliser des types spécialisés qui héritent des caractéristiques communes du type de base.

En d'autres termes, les sous-types reflètent la structure définie par leur supertype mais peuvent apporter des caractéristiques supplémentaires. Par exemple, un fournisseur peut souhaiter étendre un *Type de Variable* général "Capteur de Température" en ajoutant une *Propriété* indiquant la périodicité de maintenance. Pour cela, le fournisseur crée un nouveau *Type de Variable* qui est un *TargetNode* pour une référence "*HasSubtype*" à partir du *Type de Variable* initial et y ajoute la nouvelle *Propriété*.

#### 4.5.4 Instanciation de *TypeDefinitionNodes* complexes

L'instanciation des *TypeDefinitionNodes* (*Nœuds de Définition de Type*) complexes dépend des *Règles de Modélisation* définies en 6.4.4. Il est cependant prévu que des instances d'une définition de type reflètent la structure définie par le *TypeDefinitionNode*. La Figure 7 présente une instance du *TypeDefinitionNode* "AI\_BLK\_TYPE", dans lequel la *Règle de Modélisation Obligatoire*, définie en 6.4.4.5.2, a été appliquée à sa *Variable* contenant. Ainsi, une instance de "AI\_BLK\_TYPE", appelée AI\_BLK\_1", dispose d'une *Référence "Dispose d'une Définition de Type"* vers "AI\_BLK\_TYPE". Elle contient également une *Variable* "PC" ayant le même *Nom de Navigation* que la *Variable* "PC" utilisée par le *TypeDefinitionNode* et par conséquent reflète la structure définie par le *TypeDefinitionNode* (*Nœud de Définition de Type*).



**Figure 7 – Objet et ses composants définis par un Type d'Objet**

Un *Client* connaissant le *Type d'Objet* "AI\_BLK\_TYPE" peut utiliser cette information pour explorer directement les *Nœuds* contenant de chaque instance de ce type. Ceci permet une programmation tenant compte du *TypeDefinitionNode* (*Nœud de Définition de Type*). Par exemple, un élément graphique peut être programmé dans le *Client* qui traite toutes les instances de "AI\_BLK\_TYPE" de la même manière qu'en présentant la valeur de "PC".

Une programmation selon le *TypeDefinitionNode* présente plusieurs contraintes. Un *TypeDefinitionNode* ou une *Déclaration d'Instance* ne doit jamais référencer deux *Nœuds* ayant un même *Nom de Navigation* en utilisant les *Références hiérarchiques* dans le sens direct. Les instances fondées sur des *Déclarations d'Instance* doivent toujours conserver le même *Nom de Navigation* que la *Déclaration d'Instance* dont elles résultent. Un *Service* spécial défini dans l'IEC 62541-4, appelé Traduire les Chemins de Navigation en Identificateurs de Nœuds (TranslateBrowsePathsToNodeIds), peut être utilisé pour identifier

les instances sur la base des *Déclarations d'Instance*. L'utilisation du *Service de Navigation* simple peut ne pas suffire car l'unicité du *Nom de Navigation* n'est exigée que pour les *Nœuds de Définition de Type* et les *Déclarations d'Instance*, et non pour d'autres instances. Ainsi, "AI\_BLK\_1" peut disposer d'une autre *Variable* portant le *Nom de Navigation* "PC", même si celle-ci ne résulte pas d'une *Déclaration d'Instance* du *Nœud de Définition de Type*.

Les instances dérivées d'une *Déclaration d'Instance* doivent avoir le même *TypeDefinitionNode* ou un sous-type de ce *TypeDefinitionNode*.

Un *TypeDefinitionNode* et ses *Déclarations d'Instance* doivent toujours appartenir au même *Serveur*. Cependant, des instances peuvent pointer, avec leur *Référence "Dispose d'une Définition de Type"*, vers un *TypeDefinitionNode* situé dans un *Serveur* différent.

## 4.6 Modèle d'événement

### 4.6.1 Généralités

Le Modèle d'événement définit un système de traitement des événements d'usage général qui peut servir dans de nombreux marchés verticaux différents.

Les *Événements* représentent des occurrences transitoires spécifiques. Les modifications de configuration du système et les erreurs du système sont des exemples d'*Événements*. Les *Notifications d'Événements* rendent compte de l'occurrence d'un *Événement* donné. Les *Événements* définis dans le présent document ne sont pas directement visibles dans l'*Espace d'Adressage* d'OPC UA. Des *Objets* et des *Vues* peuvent être utilisés pour s'abonner à des *Événements*. L'*Attribut "Notificateur d'Événements"* indique si le *Nœud* peut s'abonner à des *Événements*. Les *Clients* s'abonnent à ces *Nœuds* afin de recevoir des *Notifications d'occurrences d'Événements*.

L'*Abonnement aux Événements* utilise les *Services Surveillance et Abonnement* définis dans l'IEC 62541-4 pour s'abonner aux *Notifications d'Événements* d'un *Nœud*.

Tout *Serveur* OPC UA qui prend en charge le traitement des événements doit présenter au moins un *Nœud* comme *Notificateur d'Événements*. L'*Objet Serveur* défini dans l'IEC 62541-5 est utilisé à cet effet. Les *Événements* générés par le *Serveur* sont disponibles par l'intermédiaire de cet *Objet Serveur*. Il n'est pas prévu qu'un *Serveur* produise des *Événements* si, pour quelque raison que ce soit (c'est-à-dire si le système est hors ligne), la connexion à la source de l'événement est interrompue.

Des *Événements* peuvent également être présentés par l'intermédiaire d'autres *Nœuds*, partout dans l'*Espace d'Adressage*. Ces *Nœuds* (identifiés par l'*Attribut "Notificateur d'Événements"*) fournissent un sous-ensemble des *Événements* générés par le *Serveur*. La position dans l'*Espace d'Adressage* indique ce que ce sous-ensemble sera. Par exemple, un *Objet* zone de processus, représentant une zone fonctionnelle du processus, fournirait des *Événements* provenant uniquement de cette zone du processus. Il convient de noter qu'il s'agit seulement d'un exemple et qu'il incombe entièrement au *Serveur* de déterminer les *Événements* qu'il convient que tel ou tel *Nœud* fournisse.

### 4.6.2 Types d'Événements

Chaque *Événement* correspond à un *Type d'Événement* spécifique. Un *Serveur* peut prendre en charge plusieurs types. La présente partie de l'IEC 62541 définit le *Type d'Événement de Base* dont dérivent tous les autres *Types d'Événements*. Il est prévu que d'autres spécifications associées définissent des *Types d'Événements* supplémentaires dérivant des types de base définis dans la présente partie de l'IEC 62541.

Les *Types d'Événements* pris en charge par un *Serveur* sont présentés dans l'*Espace d'Adressage* du *Serveur*. Les *Types d'Événements* sont représentés comme des *Types d'Objets* dans l'*Espace d'Adressage* et il ne leur est pas associé de *Classe de Nœud* spéciale.

L'IEC 62541-5 définit de manière détaillée la manière dont un *Serveur* présente les *Types d'Événements*.

Les *Types d'Événements* définis dans le présent document sont spécifiés comme étant abstraits et par conséquent ils ne sont jamais instanciés dans l'*Espace d'Adressage*. Les occurrences d'Événements de ces *Types d'Événements* ne sont présentées que par l'intermédiaire d'un *Abonnement*. Il existe dans l'*Espace d'Adressage* des *Types d'Événements* qui permettent aux *Clients* de découvrir le *Type d'Événement*. Cette information est utilisée par un *Client* lorsqu'il établit et utilise des *Abonnements* aux *Événements*. Les *Types d'Événements* définis par d'autres parties de la série IEC 62541 ou par des spécifications associées, ainsi que les *Types d'Événements* spécifiques au *Serveur*, peuvent être définis comme non abstraits et par conséquent les instances de ces *Types d'Événements* peuvent être visibles dans l'*Espace d'Adressage* même si les *Événements* de ces *Types d'Événements* sont également accessibles par l'intermédiaire des mécanismes de *Notification d'Événements*.

Les *Types d'Événements* normalisés sont décrits à l'Article 9. Leur représentation dans l'*Espace d'Adressage* est spécifiée dans l'IEC 62541-5.

#### 4.6.3 Catégorisation des Événements

Les *Événements* peuvent être classés par catégories en créant de nouveaux *Types d'Événements* qui sont des sous-types de *Types d'Événements* existants mais qui n'étendent pas un type existant. Ils sont uniquement utilisés pour identifier un événement comme appartenant au nouveau *Type d'Événement*. Par exemple, le *Type d'Événement* "Type d'Événement de Défaillance d'un Appareil" (*DeviceFailureEventType*) peut avoir comme sous-type un "Type d'Événement de Défaillance de Transmetteur" (*TransmitterFailureEventType*) et un "Type d'Événement de Défaillance d'Ordinateur" (*ComputerFailureEventType*). Ces nouveaux sous-types ne sont pas intégrés comme de nouvelles *Propriétés* ou ne modifient pas la sémantique héritée du *Type d'Événement* de Défaillance d'un Appareil, il s'agit purement et simplement d'une catégorisation des *Événements*.

Les sources d'*Événements* peuvent également être organisées en groupes, en utilisant les *Types de Références d'Événements* décrits en 7.17 et 7.19. Par exemple, un *Serveur* peut définir des *Objets* dans l'*Espace d'Adressage* représentant des *Événements* liés aux appareils proprement dits ou des zones d'*Événements* d'une installation ou une fonctionnalité contenue dans le *Serveur*. Les *Références d'Événements* sont utilisées pour indiquer quelles sont les sources d'*Événements* qui représentent des appareils concrets et celles qui représentent une certaine fonctionnalité basée dans le *Serveur*. En outre, des *Références* peuvent être utilisées pour grouper les appareils proprement dits ou les fonctionnalités basées dans le *Serveur* en zones d'*Événements* hiérarchisées. Dans certains cas, une source d'*Événement* peut être catégorisée comme étant à la fois un appareil et une fonction du *Serveur*. Dans ce cas, deux relations sont établies. Pour des exemples supplémentaires, voir la description des *Types de Références d'Événements*.

Les *Clients* peuvent sélectionner une ou plusieurs catégories d'*Événements* en définissant des filtres de contenu qui incluent des termes spécifiant le *Type d'Événement* de l'*Événement* ou un groupement de sources d'*Événements*. Ces deux mécanismes permettent de catégoriser de plusieurs manières un *Événement* particulier. Un *Client* peut obtenir tous les *Événements* correspondant à un appareil concret ou toutes les défaillances d'un appareil particulier.

#### 4.7 Méthodes

Les *Méthodes* sont des fonctions "légères" dont le champ d'application est limité par un *Objet* appartenance (voir Note), comme pour les méthodes d'une classe donnée en programmation orientée objet ou un *Type d'Objet* appartenance, comme dans les méthodes statiques d'une classe. Les *Méthodes* sont invoquées par un *Client*, elles sont menées à bien sur le *Serveur* et les résultats sont renvoyés au *Client*. La durée de vie d'une instance d'invocation de

*Méthodes* commence lorsque le *Client* fait appel à la *Méthode* et se termine lorsque le résultat lui est renvoyé.

NOTE L'*Objet* ou le *Type d'Objet* appartenance est spécifié dans l'appel au service lorsque la *Méthode* est invoquée.

Même si les *Méthodes* peuvent affecter l'état de l'*Objet* appartenance, elles n'ont pas d'état explicite propre. De ce fait, elles sont sans état. Les *Méthodes* peuvent avoir un nombre variable d'arguments d'entrée et renvoyer des arguments résultants. Chaque *Méthode* est décrite par un *Nœud* de la *Classe de Nœud "Méthodes"*. Ce *Nœud* contient les métadonnées qui identifient les arguments de la *Méthode* et décrit son comportement.

Les *Méthodes* sont invoquées par le *Service "Appel"* défini dans l'IEC 62541-4. Chaque *Méthode* est invoquée dans le contexte d'une session existante. Si la session se termine pendant l'exécution de la *Méthode*, les résultats correspondants ne peuvent pas être renvoyés au *Client* et sont rejetés. Dans ce cas, l'exécution de la *Méthode* est non définie, ce qui veut dire que la *Méthode* peut être exécutée jusqu'à ce qu'elle s'achève ou bien qu'elle soit abandonnée.

Les *Clients* trouvent les *Méthodes* prises en charge par un *Serveur* en explorant les *Références d'Objets* appartenance qui identifient les *Méthodes* prises en charge.

## 5 Classes de Nœuds normalisées

### 5.1 Vue d'ensemble

L'Article 5 définit les *Classes de Nœuds* utilisées pour définir les *Nœuds* dans l'*Espace d'Adressage* d'OPC UA. Les *Classes de Nœuds* sont dérivées d'une *Classe de Nœud de base* commune. On définit en premier lieu cette *Classe de Nœud*, puis celles qui sont utilisées pour organiser l'*Espace d'Adressage* et ensuite les *Classes de Nœuds* utilisées pour représenter les *Objets*.

Les *Classes de Nœuds* définies pour représenter des *Objets* s'inscrivent dans trois catégories: celles utilisées pour définir les instances, celles utilisées pour définir les types d'instances et celles utilisées pour définir les types de données. Le paragraphe 6.3 décrit les règles de sous-typage et le 6.4 indique les règles d'instanciation des définitions de types.

### 5.2 Classe de Nœud de base

#### 5.2.1 Généralités

Le Modèle de l'Espace d'Adressage d'OPC UA définit une *Classe de Nœud de Base* dont sont issues toutes les autres *Classes de Nœuds*. Les *Classes de Nœuds* dérivées représentent les divers composants du modèle d'Objet OPC UA (voir 4.2). Les *Attributs* de la *Classe de Nœud de Base* sont spécifiés dans le Tableau 2. Aucune *Référence* n'est spécifiée pour la *Classe de Nœud de Base*.

**Tableau 2 – Classe de Nœud de base**

| Dénomination      | Usage | Type de Données        | Description                                                              |
|-------------------|-------|------------------------|--------------------------------------------------------------------------|
| <b>Attributs</b>  |       |                        |                                                                          |
| NodeId            | M     | Identificateur de Nœud | Voir 5.2.2                                                               |
| NodeClass         | M     | Classe de Nœud         | Voir 5.2.3                                                               |
| BrowseName        | M     | Dénomination Qualifiée | Voir 5.2.4                                                               |
| DisplayName       | M     | Texte Localisé         | Voir 5.2.5                                                               |
| Description       | O     | Texte Localisé         | Voir 5.2.6                                                               |
| WriteMask         | O     | UInt32                 | Voir 5.2.7                                                               |
| UserWriteMask     | O     | UInt32                 | Voir 5.2.8                                                               |
| <b>Références</b> |       |                        | Aucune <i>Référence</i> n'est spécifiée pour cette <i>Classe de Nœud</i> |

### 5.2.2 NodeId (Identificateur de Nœud)

Les *Nœuds* sont identifiés sans aucune ambiguïté par un identificateur construit appelé *NodeId* (*Identificateur de Nœud*). Certains *Serveurs* peuvent accepter d'autres *Identificateurs de Nœuds* outre l'*Identificateur de Nœud* canonique représenté dans cet *Attribut*. Un *Serveur* doit conserver l'*Identificateur de Nœud* d'un *Nœud*, c'est-à-dire il ne doit pas générer de nouveaux *Identificateurs de Nœuds* lors de la réinitialisation. La structure de l'*Identificateur de Nœud* est définie en 8.2.

### 5.2.3 NodeClass (Classe de Nœud)

L'*Attribut NodeClass* (*Classe de Nœud*) identifie la *Classe de Nœud* d'un *Nœud*. Le type de données correspondant est défini en 8.30.

### 5.2.4 BrowseName (Nom de Navigation)

Les *Nœuds* disposent d'un *Attribut BrowseName* (*Nom de Navigation*) utilisé comme dénomination non localisée et interprétable par l'utilisateur lorsqu'il explore l'*Espace d'Adressage* pour créer des chemins à partir des *Noms de Navigation*. Le *Service "Traduire Chemins de Navigation en Identificateurs de Nœuds"* défini dans l'IEC 62541-4 peut être utilisé pour suivre un chemin constitué de *Noms de Navigation*.

Il convient de ne jamais utiliser un *Nom de Navigation* pour afficher le nom d'un *Nœud*. Il convient d'utiliser pour cela le *Nom d'Affichage*.

Contrairement aux *Identificateurs de Nœuds*, le *Nom de Navigation* ne peut pas être utilisé pour identifier de manière univoque un *Nœud*. Différents *Nœuds* peuvent utiliser le même *Nom de Navigation*.

Le paragraphe 8.3 définit la structure du *Nom de Navigation*. Il contient un espace de nom et une chaîne de caractères. L'espace de nom permet d'assurer l'unicité du *Nom de Navigation* dans certains cas, dans le contexte d'un *Nœud* particulier (par exemple, les *Propriétés* d'un *Nœud*), même s'il n'est pas unique dans le contexte du *Serveur*. Si différents organismes définissent des *Noms de Navigation* pour les *Propriétés*, l'espace de nom du *Nom de Navigation* fourni par l'organisme rend le *Nom de Navigation* unique, même si différents organismes peuvent utiliser la même chaîne de caractères avec une signification légèrement différente.

Il arrive souvent que les *serveurs* choisissent d'utiliser le même espace de nom pour l'*Identificateur de Nœud* et le *Nom de Navigation*. Cependant, s'ils souhaitent fournir une *Propriété* normalisée, son *Nom de Navigation* doit avoir l'espace de nom de l'organisme de normalisation, même si l'espace de nom de l'*Identificateur de Nœud* reflète quelque chose de différent, par exemple le *Serveur* local.

Il est recommandé que les organismes de normalisation établissant des définitions de type normalisées utilisent leur espace de nom pour l'*Identificateur de Nœud* d'un *Nœud de Définition de Type* ainsi que le *Nom de Navigation* du *Nœud de Définition de Type*.

La partie chaîne de caractères du *Nom de navigation* est sensible à la casse.

### 5.2.5 DisplayName (Nom d’Affichage)

L'*Attribut DisplayName (Nom d’Affichage)* contient la dénomination localisée du *Nœud*. Il convient que les *Clients* utilisent cet *Attribut* s'ils souhaitent afficher le nom du *Nœud* pour l'utilisateur. Il convient qu'ils n'utilisent pas le *Nom de Navigation* à cet effet. Le *Serveur* peut maintenir une ou plusieurs représentations localisées pour chaque *Nom d’Affichage*. Les *Clients* négocient les paramètres de lieu à renvoyer lorsqu'ils ouvrent une session sur le *Serveur*. Pour une description de l'établissement de session et des paramètres de lieu, voir l'IEC 62541-4. Le paragraphe 8.5 définit la structure du *Nom d’Affichage*. La partie chaîne de caractères du *Nom d’Affichage* est limitée à 512 caractères.

### 5.2.6 Description

L'*Attribut Description* qui est facultatif doit expliquer la signification du *Nœud* dans un texte localisé en utilisant les mêmes mécanismes de localisation que ceux décrits pour le *Nom d’Affichage* en 5.2.5.

### 5.2.7 WriteMask (Masque d’Écriture)

L'*Attribut WriteMask (Masque d’Écriture)* qui est facultatif présente au *Client* des possibilités d'écriture des *Attributs* du *Nœud*. L'*Attribut "Masque d’Écriture"* ne tient compte d'aucun droit d'accès utilisateur, ce qui signifie que même si un *Attribut* est modifiable, cette possibilité peut être limitée à un certain utilisateur / groupe d'utilisateurs.

Si le *Serveur* OPC UA n'a pas la possibilité d'obtenir les informations *Masque d’Écriture* d'un *Attribut* spécifique à partir du système sous-jacent, il convient de déclarer qu'il est modifiable. Si une opération d'écriture est invoquée pour l'*Attribut*, il convient que le *Serveur* transfère cette requête et renvoie le *Code de Statut* correspondant si une cette requête est rejetée. Les *Codes de Statut* sont définis dans l'IEC 62541-4.

L'*Attribut "Masque d’Écriture"* est un entier de 32 bits non signé dont la structure est définie dans le Tableau 3. Si le bit est réglé à 0, ceci veut dire que l'*Attribut* n'est pas modifiable, s'il est réglé à 1, cela signifie qu'il est modifiable. Si un *Nœud* ne prend pas en charge un *Attribut* spécifique, le bit correspondant est à mettre à 0.

**Tableau 3 – Masque de bit pour Masque d'Écriture et Masque d'Écriture Utilisateur**

| Champ                   | Bit   | Description                                                                                                                                                                                                                                                                           |
|-------------------------|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AccessLevel             | 0     | Indique si l'Attribut "Niveau d'Accès" est modifiable.                                                                                                                                                                                                                                |
| ArrayDimensions         | 1     | Indique si l'Attribut "Dimensions de Matrice" est modifiable.                                                                                                                                                                                                                         |
| BrowseName              | 2     | Indique si l'Attribut "Nom de Navigation" est modifiable.                                                                                                                                                                                                                             |
| ContainsNoLoops         | 3     | Indique si l'Attribut "Ne Contient Aucune Boucle" est modifiable.                                                                                                                                                                                                                     |
| DataType                | 4     | Indique si l'Attribut "Type de Données" est modifiable.                                                                                                                                                                                                                               |
| Description             | 5     | Indique si l'Attribut "Description" est modifiable.                                                                                                                                                                                                                                   |
| DisplayName             | 6     | Indique si l'Attribut "Nom d'Affichage" est modifiable.                                                                                                                                                                                                                               |
| EventNotifier           | 7     | Indique si l'Attribut "Notificateur d'Événements" est modifiable.                                                                                                                                                                                                                     |
| Executable              | 8     | Indique si l'Attribut "Exécutable" est modifiable.                                                                                                                                                                                                                                    |
| Historizing             | 9     | Indique si l'Attribut "Historisation" est modifiable.                                                                                                                                                                                                                                 |
| InverseName             | 10    | Indique si l'Attribut "Nom Inverse" est modifiable.                                                                                                                                                                                                                                   |
| IsAbstract              | 11    | Indique si l'Attribut "Est Abstrait" est modifiable.                                                                                                                                                                                                                                  |
| MinimumSamplingInterval | 12    | Indique si l'Attribut "Intervalle d'Échantillonnage Minimal" est modifiable.                                                                                                                                                                                                          |
| NodeClass               | 13    | Indique si l'Attribut "Classe de Nœud" est modifiable.                                                                                                                                                                                                                                |
| NodeId                  | 14    | Indique si l'Attribut "Identificateur de Nœud" est modifiable.                                                                                                                                                                                                                        |
| Symmetric               | 15    | Indique si l'Attribut "Symétrique" est modifiable.                                                                                                                                                                                                                                    |
| UserAccessLevel         | 16    | Indique si l'Attribut "Niveau d'Accès Utilisateur" est modifiable.                                                                                                                                                                                                                    |
| UserExecutable          | 17    | Indique si l'Attribut "Exécutable par l'Utilisateur" est modifiable.                                                                                                                                                                                                                  |
| UserWriteMask           | 18    | Indique si l'Attribut "Masque d'Écriture Utilisateur" est modifiable.                                                                                                                                                                                                                 |
| ValueRank               | 19    | Indique si l'Attribut "Rang de Valeur" est modifiable.                                                                                                                                                                                                                                |
| WriteMask               | 20    | Indique si l'Attribut "Masque d'Écriture" est modifiable.                                                                                                                                                                                                                             |
| ValueForVariableType    | 21    | Indique si l'Attribut "Valeur" est modifiable pour un Type de Variable donné. Il ne s'applique pas aux Variables puisqu'il est traité par les Attributs "Niveau d'Accès" et "Niveau d'Accès Utilisateur" pour la Variable. Dans ce cas, pour les Variables, ce bit doit être mis à 0. |
| Reserved                | 22:31 | Réservé pour usage ultérieur. Doit toujours être à zéro.                                                                                                                                                                                                                              |

### 5.2.8 UserWriteMask (Masque d'Écriture Utilisateur)

L'Attribut *UserWriteMask* (Masque d'Écriture Utilisateur) qui est facultatif présente à un Client les possibilités d'écriture des Attributs du Nœud en tenant compte des droits d'accès de l'utilisateur. Il utilise le même masque de bit que celui de l'Attribut "Masque d'Écriture", défini dans le Tableau 3.

L'Attribut "Masque d'Écriture Utilisateur" étend la restriction de l'Attribut "Masque d'Écriture", lorsqu'il est mis sur non modifiable dans le cas général applicable à chaque utilisateur.

## 5.3 ReferenceTypeNodeClass (Classe de Nœud "Types de Références")

### 5.3.1 Généralités

Les Références sont définies comme des instances des Nœuds de Type de Référence. Les Nœuds de Type de Référence sont visibles dans l'Espace d'Adressage et sont définis en utilisant la Classe de Nœud "Types de Références" comme spécifié dans le Tableau 4. En revanche, une Référence est une partie inhérente d'un Nœud et aucune Classe de Nœud n'est utilisée pour représenter des Références.

La présente norme définit un ensemble de Types de Références fournis comme parties inhérentes du Modèle de l'Espace d'Adressage d'OPC UA. Ces Types de Références sont définis dans l'Article 7 et leur représentation dans l'Espace d'Adressage est définie dans l'IEC 62541-5. Les Serveurs peuvent également définir des Types de Références. Par ailleurs, l'IEC 62541-4 définit des Services de Gestion de Nœuds qui permettent aux Clients d'ajouter des Types de Références à l'Espace d'Adressage.

**Tableau 4 – Classe de Nœud "Types de Références"**

| Dénomination                  | Usage | Type de Données      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------------------|-------|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Attributs</b>              |       |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Base NodeClass                | M     | --                   | Hérité de la <i>Classe de Nœud de Base</i> . Voir 5.2.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| IsAbstract                    | M     | Booléen              | Un <i>Attribut</i> booléen prenant les valeurs suivantes:<br>VRAI Il s'agit d'un <i>Type de Référence</i> abstrait, c'est-à-dire qu'aucune <i>Référence</i> de ce type ne doit exister, mais uniquement ses sous-types.<br>FAUX Il s'agit d'un <i>Type de Référence</i> qui n'est pas abstrait, c'est-à-dire qu'il peut y avoir des <i>Références</i> de ce type.                                                                                                                                                                                |
| Symmetric                     | M     | Booléen              | Un <i>Attribut</i> booléen prenant les valeurs suivantes:<br>VRAI La signification du <i>Type de Référence</i> est la même que celle qui est perçue à la fois du point de vue du <i>SourceNode</i> et du <i>TargetNode</i> .<br>FAUX La signification du <i>Type de Référence</i> telle que perçue du point de vue du <i>TargetNode</i> est l'inverse de celle qui est perçue à partir du <i>SourceNode</i> .                                                                                                                                    |
| InverseName                   | O     | Texte Localisé       | La dénomination inverse de la <i>Référence</i> , c'est-à-dire la signification du <i>Type de Référence</i> telle que perçue à partir du <i>TargetNode</i> .                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Références</b>             |       |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| HasProperty                   | 0..*  |                      | Utilisé pour identifier les <i>Propriétés</i> (voir 5.3.3.2).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| HasSubtype                    | 0..*  |                      | Utilisé pour identifier des sous-types (voir 5.3.3.3).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Propriétés normalisées</b> |       |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| NodeVersion                   | O     | Chaîne de caractères | La <i>Propriété</i> «NodeVersion» est utilisée pour indiquer la version d'un <i>Nœud</i> .<br>La <i>Propriété</i> «NodeVersion» est mise à jour à chaque fois qu'une <i>Référence</i> est ajoutée ou retirée au <i>Nœud</i> auquel la <i>Propriété</i> appartient. Les modifications de la valeur de l' <i>Attribut</i> n'entraînent pas une modification de la <i>Version de Nœud</i> . Les <i>Clients</i> peuvent consulter la <i>Propriété</i> «NodeVersion» ou s'y abonner pour établir le moment où la structure d'un <i>Nœud</i> a changé. |

### 5.3.2 Attributs

La *Classe de Nœud* "Types de Références" hérite des *Attributs* de base de la *Classe de Nœud de Base* définie en 5.2. L'*Attribut* "Nom de Navigation" hérité est utilisé pour préciser la signification du *Type de Référence* telle que perçue à partir du *Nœud Source*. Par exemple, le *Type de Référence* portant le *Nom de Navigation* "Contains" (Contient) est utilisé dans les *Références* pour indiquer que le *SourceNode* contient le *Nœud Cible*. L'*Attribut* "Nom d'Affichage" hérité contient une traduction du *Nom de Navigation*.

Le *Nom de Navigation* d'un *Type de Référence* donné doit être unique dans un *Serveur*. Il n'est pas admis que deux *Types de Références* différents portent le même *Nom de Navigation*.

L'*Attribut IsAbstract* (*Est Abstrait*) indique que ce *Type de Référence* est abstrait. Les *Types de Références* abstraits ne peuvent être instanciés et sont utilisés uniquement pour des raisons d'organisation, par exemple pour spécifier certaines caractéristiques sémantiques ou contraintes d'ordre général dont ont héritées ses sous-types.

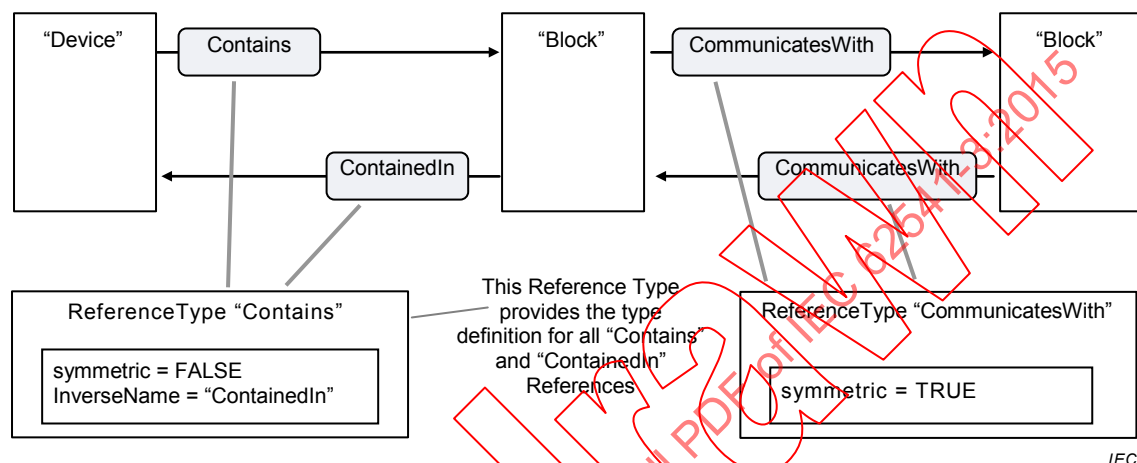
L'*Attribut Symmetric* (*Symétrique*) est utilisé pour indiquer si la signification du *Type de Référence* est (ou n'est pas) identique à la fois pour le *SourceNode* et le *TargetNode*.

Si un *Type de Référence* est symétrique, l'*Attribut InverseName* (*Nom Inverse*) doit être omis. Les exemples de *Types de Références* symétriques sont "Se Connecte A" et "Communique Avec". Les deux impliquent la même sémantique provenant du *SourceNode* ou du *TargetNode*. Par conséquent, les deux directions sont considérées comme des *Références* directes.

Si le *Type de Référence* est non symétrique et non abstrait, l'*Attribut "Nom Inverse"* doit être établi. L'*Attribut "Nom Inverse"* spécifie la signification du *Type de Référence* telle que perçue à partir du *TargetNode*. Les exemples de *Types de Références* non symétriques sont notamment "Contient" et "Contenu Dans", et "Reçoit De" et "Envoie A".

Les *Références* qui utilisent le *Nom Inverse*, par exemple les *Références* "Contenu Dans", sont appelées *Références inverses*.

La Figure 8 donne des exemples de *Références* symétriques et non symétriques ainsi que d'utilisation du *Nom de Navigation* et du *Nom Inverse*.



| Anglais                                                                                         | Français                                                                                                 |
|-------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| Device                                                                                          | Appareil                                                                                                 |
| Contains                                                                                        | Contient                                                                                                 |
| Block                                                                                           | Bloc                                                                                                     |
| CommunicatesWith                                                                                | Communique avec                                                                                          |
| ContainedIn                                                                                     | Contenu dans                                                                                             |
| ReferenceType "Contains"                                                                        | Type de référence "contient"                                                                             |
| This referencetype provides the type definition for all "Contains" and "ContainedIn" references | Ce Type de Référence fournit la définition de type de toutes les références "contient" et "contenu dans" |
| Referencetype "CommunicatesWith"                                                                | Type de Référence "communique avec"                                                                      |
| Symmetric = FALSE                                                                               | Symétrique = Faux                                                                                        |
| Inversename = "ContainedIn"                                                                     | Nom inverse = "contenu dans"                                                                             |
| Symmetric = True                                                                                | Symétrique = Vrai                                                                                        |

**Figure 8 – Références Symétriques et Non Symétriques**

Il est admis qu'il ne soit pas toujours possible pour les *Serveurs* d'instancier à la fois des *Références* directes et inverses pour des *Types de Références* non symétriques, comme illustré en Figure 8. Lorsque c'est le cas, les *Références* sont appelées *bidirectionnelles*. Bien que cela ne soit pas exigé, il est recommandé que toutes les *Références hiérarchiques* soient instanciées de manière bidirectionnelle afin d'assurer la connectivité de la navigation. Une *Référence* bidirectionnelle est modélisée comme deux *Références* séparées.

Un exemple de *Référence unidirectionnelle*: il arrive souvent qu'un abonné connaisse son éditeur, mais que ce dernier ne connaisse pas ses abonnés. L'abonné aurait une *Référence* "S'Abonne A" vers l'éditeur, sans que l'éditeur n'ait les *Références* inverses correspondantes "Edite Vers" pointant vers ses abonnés.

Le *Nom d'Affichage* et le *Nom Inverse* sont les seuls emplacements normalisés utilisés pour indiquer la sémantique d'un *Type de Référence* donné. Il peut y avoir plusieurs règles sémantiques complexes associées à un *Type de Référence* qui peuvent être exprimées dans

ces *Attributs* (par exemple, la sémantique de la *Référence "HasSubtype"*). La présente norme ne spécifie pas la manière dont il convient d'exprimer cette sémantique. Il est cependant possible d'utiliser à cet effet l'*Attribut "Description"*. La présente norme fournit une règle sémantique pour les *Types de Références* spécifiés à l'Article 7.

Un *Type de Référence* peut avoir des contraintes limitant son utilisation. Il peut spécifier par exemple qu'en partant du *Nœud A* et uniquement en suivant les *Références* de ce *Type de Référence* ou d'un de ses sous-types, il ne doit jamais être possible de revenir à *A*; ce qui signifie une contrainte "Pas de Boucle".

La présente norme ne spécifie pas la manière dont il est possible ou dont il convient de mettre à disposition ces contraintes dans l'*Espace d'Adressage*. Cependant, pour les *Types de Références* normalisés, certaines contraintes sont spécifiées à l'Article 7. La présente norme ne limite pas le type de contraintes utilisables pour un *Type de Référence* donné. La contrainte peut, par exemple, affecter également un *Type d'Objet*. La restriction selon laquelle un *Type de Référence* ne peut être utilisé que par des *Nœuds* de certaines *Classes de Nœuds* ayant une cardinalité définie est une contrainte particulière d'un *Type de Référence*.

### 5.3.3 Références

#### 5.3.3.1 Généralités

Les *Références HasSubtype* (*Dispose d'un Sous-type*) et *HasProperty* (*Dispose d'une Propriété*) sont les seuls *Types de Références* qui peuvent être utilisés avec des *Nœuds de Type de Référence* tels que le *SourceNode* (*Nœud Source*). Les *Nœuds de Type de Référence* ne doivent pas être le *SourceNode* d'autres types de *Références*.

#### 5.3.3.2 Références "HasProperty" (*Dispose d'une Propriété*)

Les *Références HasProperty* (*Dispose d'une Propriété*) sont utilisées pour identifier les *Propriétés* d'un *Type de Référence* et ne doivent se référer qu'aux *Nœuds* de la *Classe de Nœud "Variables"*.

La *Propriété NodeVersion* (*Version de Nœud*) est utilisée pour indiquer la version du *Type de Référence*.

Il n'y a pas d'autres *Propriétés* définies pour les *Types de Références* dans la présente norme. D'autres parties de la série IEC 62541 peuvent définir des *Propriétés* supplémentaires pour les *Types de Références*.

#### 5.3.3.3 Références "HasSubtype" (*Dispose d'un Sous-type*)

Les *Références HasSubtype* (*Dispose d'un Sous-type*) sont utilisées pour définir des sous-types de *Types de Références*. Il n'est pas exigé que la *Référence "HasSubtype"* soit fournie pour le supertype; il est cependant exigé que ce sous-type fournisse la *Référence inverse* à son supertype. Les règles suivantes s'appliquent au sous-typage.

- a) La sémantique d'un *Type de Référence* (par exemple "couvre une hiérarchie") est héritée par ses sous-types et peut être définie plus finement (par exemple "couvre une hiérarchie spéciale"). Le *Nom d'Affichage*, ainsi que le *Nom Inverse* pour les *Types de Références* non symétriques, reflètent cette spécialisation.
- b) Si un *Type de Référence* spécifie certaines contraintes (par exemple "ne permet aucune boucle"), cette restriction est héritée et ne peut qu'être rendue plus sévère (par exemple hérité "aucune boucle" peut être affiné en "doit être une arborescence – un parent uniquement") mais non réduit (par exemple "autoriser les boucles").
- c) Les contraintes concernant les *Classes de Nœuds* qui peuvent être référencées sont également héritées et ne peuvent être que plus strictes. En d'autres termes, s'il n'est pas admis qu'un *Type de Référence "A"* relie un *Objet* à un *Type d'Objet*, ceci est également vrai pour tous ses sous-types.

- d) Un *Type de Référence* doit avoir exactement un supertype, sauf en ce qui concerne le *Type de Référence "Références"* défini en 7.2 comme étant le type racine de la hiérarchie du *Type de Référence*. La hiérarchie du *Type de Référence* ne prend pas en charge les héritages multiples.

#### 5.4 Classe de Nœud "Vues" (View)

Les systèmes sous-jacents sont souvent de grande taille et les *Clients* n'ont souvent besoin que d'un sous-ensemble spécifique de données. Ils ne souhaitent pas ou ne veulent pas être surchargés par des *Nœuds* de Vues dans l'*Espace d'Adressage* qui ne les intéressent aucunement.

Pour résoudre ce problème, la présente norme définit le concept de *Vue* (View). Chaque *Vue* définit un sous-ensemble de *Nœuds* dans l'*Espace d'Adressage*. L'ensemble *Espace d'Adressage* est la *Vue* par défaut. Chaque *Nœud* dans une *Vue* peut contenir uniquement un sous-ensemble de ses *Références*, comme défini par le créateur de la *Vue*. Le *Nœud de Vues* agit comme la racine des *Nœuds* dans la *Vue*. Les *Vues* sont définies au moyen de la *Classe de Nœud "Vues"* spécifiée dans le Tableau 5.

Tous les *Nœuds* contenus dans une *Vue* doivent être accessibles à partir du *Nœud de Vues* lorsqu'une navigation est effectuée dans le contexte de la *Vue*. Il n'est pas prévu que tous les *Nœuds* contenant puissent être explorés directement à partir du *Nœud de Vues* mais plutôt d'autres *Nœuds* contenus dans la *Vue*.

Un *Nœud de Vues* peut ne pas être uniquement utilisé comme entrée supplémentaire dans l'*Espace d'Adressage* mais comme un concept permettant d'organiser l'*Espace d'Adressage* et ainsi comme le seul point d'entrée dans un sous-ensemble de l'*Espace d'Adressage*. Par conséquent, les *Clients* ne doivent pas ignorer les *Nœuds de Vues* lorsqu'ils présentent l'*Espace d'Adressage*. Les *Clients* simples qui n'utilisent pas les *Vues* à des fins de filtrage peuvent par exemple traiter un *Nœud de Vues* comme un type d'*Objet* de type "*Type de Dossier*" (voir 5.5.3).

Tableau 5 – Classe de Nœud "Vues"

| Dénomination                  | Usage | Type de Données                                                                                                                                                                                                                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
|-------------------------------|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-----|-------------|----------------------------|---|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|---|----------------------------------------------------------|------------------------------|---|---------------------------------------------------------------------------------------------------------------------|------------------------------|---|------------------------------------------------------------------------------------------------------------------|---------|-----|----------------------------------------------------------|
| <b>Attributs</b>              |       |                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| Base NodeClass Attributes     | M     | --                                                                                                                                                                                                                                     | Hérité de la <i>Classe de Nœud de Base</i> . Voir 5.2.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| ContainsNoLoops               | M     | Booléen                                                                                                                                                                                                                                | S'il est mis sur "vrai", cet <i>Attribut</i> indique que les <i>Références</i> suivantes dans le contexte de la <i>Vue</i> ne contiennent aucune boucle; en d'autres termes, à partir d'un <i>Nœud</i> "A" contenu dans la <i>Vue</i> et suivant les <i>Références</i> directes dans le contexte du <i>Nœud de Vues</i> , "A" ne sera jamais atteint de nouveau. Ceci ne signifie pas qu'il y a qu'un chemin à partir du <i>Nœud de Vues</i> pour atteindre un <i>Nœud</i> contenu dans la <i>Vue</i> .<br>S'il est réglé sur "faux", cet <i>Attribut</i> indique que les <i>Références</i> suivantes dans le contexte de la <i>Vue</i> peuvent donner lieu à des boucles.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| EventNotifier                 | M     | Octet                                                                                                                                                                                                                                  | L' <i>Attribut</i> "Notificateur d'Événements" est utilisé pour indiquer si le <i>Nœud</i> peut être utilisé pour s'abonner à des <i>Événements</i> ou pour consulter / modifier des <i>Événements</i> historiques.<br>Le <i>Notificateur d'Événements</i> est un entier de 8 bits non signé dont la structure est définie dans le tableau suivant. <table><tr><th>Champ</th><th>Bit</th><th>Description</th></tr><tr><td>S'Abonner à des Événements</td><td>0</td><td>Indique s'il peut être utilisé pour s'abonner à des <i>Événements</i> (0 signifie qu'il ne peut pas être utilisé pour s'abonner à des <i>Événements</i>, 1 signifie qu'il peut être utilisé pour s'abonner à des <i>Événements</i>).</td></tr><tr><td>Réservé</td><td>1</td><td>Réservé pour usage ultérieur. Doit toujours être à zéro.</td></tr><tr><td>Consultation de l'Historique</td><td>2</td><td>Indique si l'historique des <i>Événements</i> est consultable (0 signifie non consultable, 1 signifie consultable).</td></tr><tr><td>Modification de l'Historique</td><td>3</td><td>Indique si l'historique des <i>Événements</i> est modifiable (0 signifie non modifiable, 1 signifie modifiable).</td></tr><tr><td>Réservé</td><td>4:7</td><td>Réservé pour usage ultérieur. Doit toujours être à zéro.</td></tr></table><br>Les deux bits suivants indiquent également si l'historique des <i>Événements</i> est disponible via le <i>Serveur</i> OPC UA. | Champ | Bit | Description | S'Abonner à des Événements | 0 | Indique s'il peut être utilisé pour s'abonner à des <i>Événements</i> (0 signifie qu'il ne peut pas être utilisé pour s'abonner à des <i>Événements</i> , 1 signifie qu'il peut être utilisé pour s'abonner à des <i>Événements</i> ). | Réservé | 1 | Réservé pour usage ultérieur. Doit toujours être à zéro. | Consultation de l'Historique | 2 | Indique si l'historique des <i>Événements</i> est consultable (0 signifie non consultable, 1 signifie consultable). | Modification de l'Historique | 3 | Indique si l'historique des <i>Événements</i> est modifiable (0 signifie non modifiable, 1 signifie modifiable). | Réservé | 4:7 | Réservé pour usage ultérieur. Doit toujours être à zéro. |
| Champ                         | Bit   | Description                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| S'Abonner à des Événements    | 0     | Indique s'il peut être utilisé pour s'abonner à des <i>Événements</i> (0 signifie qu'il ne peut pas être utilisé pour s'abonner à des <i>Événements</i> , 1 signifie qu'il peut être utilisé pour s'abonner à des <i>Événements</i> ). |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| Réservé                       | 1     | Réservé pour usage ultérieur. Doit toujours être à zéro.                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| Consultation de l'Historique  | 2     | Indique si l'historique des <i>Événements</i> est consultable (0 signifie non consultable, 1 signifie consultable).                                                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| Modification de l'Historique  | 3     | Indique si l'historique des <i>Événements</i> est modifiable (0 signifie non modifiable, 1 signifie modifiable).                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| Réservé                       | 4:7   | Réservé pour usage ultérieur. Doit toujours être à zéro.                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| <b>Références</b>             |       |                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| HierarchicalReferences        | 0..*  |                                                                                                                                                                                                                                        | Les <i>Nœuds</i> de niveau supérieur dans une <i>Vue</i> sont référencés par des <i>Références hiérarchiques</i> (voir 7.3).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| HasProperty                   | 0..*  |                                                                                                                                                                                                                                        | Les <i>Références</i> "Dispose d'une <i>Propriété</i> " identifient les <i>Propriétés</i> de la <i>Vue</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| <b>Propriétés normalisées</b> |       |                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| NodeVersion                   | O     | Chaîne                                                                                                                                                                                                                                 | La <i>Propriété</i> «NodeVersion» est utilisée pour indiquer la version d'un <i>Nœud</i> .<br>La <i>Propriété</i> «NodeVersion» est mise à jour à chaque fois qu'une <i>Référence</i> est ajoutée ou retirée du <i>Nœud</i> auquel la <i>Propriété</i> appartient. Les modifications de la valeur de l' <i>Attribut</i> n'entraînent pas une modification de la <i>Version de Nœud</i> . Les <i>Clients</i> peuvent consulter la <i>Propriété</i> «NodeVersion» ou s'y abonner pour établir le moment où la structure d'un <i>Nœud</i> a changé.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| ViewVersion                   | O     | UInt32                                                                                                                                                                                                                                 | Le numéro de Version de la <i>Vue</i> . Lorsque des <i>Nœuds</i> sont ajoutés ou retirés d'une <i>Vue</i> , la valeur de la <i>Propriété</i> "Version de Vue" est mise à jour. Les <i>Clients</i> peuvent détecter d'éventuelles modifications dans la composition d'une <i>Vue</i> en utilisant cette <i>Propriété</i> . La valeur de la Version de la <i>Vue</i> doit toujours être supérieure à 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |

La *Classe de Nœud "Vues"* hérite des *Attributs* de base de la *Classe de Nœud de Base* définie en 5.2. Deux *Attributs* supplémentaires sont également spécifiés.

L'*Attribut* obligatoire *ContainsNoLoops* (*Ne Contient Aucune Boucle*) est réglé sur faux si le *Serveur* est incapable de savoir si la *Vue* contient ou non des boucles.

L'*Attribut* obligatoire *EventNotifier* (*Notificateur d'Événements*) indique si la *Vue* peut être utilisée pour s'abonner à des *Événements* qui ont lieu soit dans le contenu de la *Vue* soit en tant qu'*Événements de Modification du Modèle* (voir 9.32) du contenu de la *Vue*; il est également utilisé pour consulter/modifier l'historique des *Événements*. Une *Vue* prenant en charge des *Événements* doit prévoir tous les *Événements* qui ont lieu dans tout *Objet* utilisé comme *Notificateur d'Événements* faisant partie du contenu de la *Vue*. En outre, il doit tenir compte de tous les *Événements de Modification du Modèle* qui ont lieu dans le contexte de la *Vue*.

Pour éviter les répétitions, c'est-à-dire obtenir tous les *Événements* du *Serveur*, l'*Objet Serveur* défini dans l'IEC 62541-5 ne doit jamais faire partie d'une quelconque *Vue* étant donné qu'il fournit tous les *Événements* du *Serveur*.

Les *Vues* sont définies par le *Serveur*. Les *Services* de navigation et d'interrogation définis dans l'IEC 62541-4 attendent de recevoir l'*Identificateur de Nœud* d'un *Nœud de Vues* pour fournir ces *Services* dans le contexte de la *Vue*.

Les *Références "Dispose d'une Propriété"* sont utilisées pour identifier les *Propriétés* d'une *Vue*. La *Propriété «NodeVersion»* est utilisée pour indiquer la version du *Nœud de Vues*. La *Propriété "Version de Vue"* est utilisée pour indiquer la version du contenu de la *Vue*. Contrairement à la *Version de Nœud*, la *Propriété "Version de Vue"* est mise à jour même s'il est ajouté ou retiré de la *Vue* des *Nœuds* qui ne sont pas directement référencés par le *Nœud de Vues*. Cette *Propriété* est facultative car il est possible que certains *Serveurs* ne puissent pas détecter des modifications dans le contenu de la *Vue*. Les *Serveurs* peuvent également générer un *Événement de Modification du Modèle* (décrit en 9.32) si des *Nœuds* sont ajoutés ou supprimés de la *Vue*. Il n'y a pas d'autres *Propriétés* définies pour les *Vues* dans le présent document. D'autres parties de la série IEC 62541 peuvent définir des *Propriétés* supplémentaires pour les *Vues*.

Les *Vues* peuvent être le *SourceNode* de toute *Référence hiérarchique*. Elles ne doivent pas être le *SourceNode* d'une quelconque *Référence non hiérarchique*.

## 5.5 Objets

### 5.5.1 Classe de Nœud "Objets"

Les *Objets* sont utilisés pour représenter des systèmes, des composants de systèmes, des objets concrets et des objets logiciels. Les *Objets* sont définis en utilisant la *Classe de Nœud "Objets"* spécifiée dans le Tableau 6.

Tableau 6 – Classe de Nœud "Objets"

| Dénomination                 | Usage | Type de Données                                                                                                                                                                                                                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
|------------------------------|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-----|-------------|----------------------------|---|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|---|----------------------------------------------------------|------------------------------|---|---------------------------------------------------------------------------------------------------------------------|------------------------------|---|------------------------------------------------------------------------------------------------------------------|---------|-----|----------------------------------------------------------|
| <b>Attributs</b>             |       |                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| Base NodeClass Attributes    | M     | --                                                                                                                                                                                                                                     | Hérité de la <i>Classe de Nœud de Base</i> . Voir 5.2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| EventNotifier                | M     | Octet                                                                                                                                                                                                                                  | <p>L'Attribut "<i>Notificateur d'Événements</i>" est utilisé pour indiquer si le <i>Nœud</i> peut être utilisé pour s'abonner à des <i>Événements</i> ou pour consulter / modifier des <i>Événements</i> historiques. Le <i>Notificateur d'Événements</i> est un entier de 8 bits non signé dont la structure est définie dans le tableau suivant:</p> <table><tr><th>Champ</th><th>Bit</th><th>Description</th></tr><tr><td>S'Abonner à des Événements</td><td>0</td><td>Indique s'il peut être utilisé pour s'abonner à des <i>Événements</i> (0 signifie qu'il ne peut pas être utilisé pour s'abonner à des <i>Événements</i>, 1 signifie qu'il peut être utilisé pour s'abonner à des <i>Événements</i>).</td></tr><tr><td>Réservé</td><td>1</td><td>Réservé pour usage ultérieur. Doit toujours être à zéro.</td></tr><tr><td>Consultation de l'Historique</td><td>2</td><td>Indique si l'historique des <i>Événements</i> est consultable (0 signifie non consultable, 1 signifie consultable).</td></tr><tr><td>Modification de l'Historique</td><td>3</td><td>Indique si l'historique des <i>Événements</i> est modifiable (0 signifie non modifiable, 1 signifie modifiable).</td></tr><tr><td>Réservé</td><td>4-7</td><td>Réservé pour usage ultérieur. Doit toujours être à zéro.</td></tr></table> <p>Les deux bits suivants indiquent également si l'historique des <i>Événements</i> est disponible via le <i>Serveur OPC UA</i>.</p> | Champ | Bit | Description | S'Abonner à des Événements | 0 | Indique s'il peut être utilisé pour s'abonner à des <i>Événements</i> (0 signifie qu'il ne peut pas être utilisé pour s'abonner à des <i>Événements</i> , 1 signifie qu'il peut être utilisé pour s'abonner à des <i>Événements</i> ). | Réservé | 1 | Réservé pour usage ultérieur. Doit toujours être à zéro. | Consultation de l'Historique | 2 | Indique si l'historique des <i>Événements</i> est consultable (0 signifie non consultable, 1 signifie consultable). | Modification de l'Historique | 3 | Indique si l'historique des <i>Événements</i> est modifiable (0 signifie non modifiable, 1 signifie modifiable). | Réservé | 4-7 | Réservé pour usage ultérieur. Doit toujours être à zéro. |
| Champ                        | Bit   | Description                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| S'Abonner à des Événements   | 0     | Indique s'il peut être utilisé pour s'abonner à des <i>Événements</i> (0 signifie qu'il ne peut pas être utilisé pour s'abonner à des <i>Événements</i> , 1 signifie qu'il peut être utilisé pour s'abonner à des <i>Événements</i> ). |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| Réservé                      | 1     | Réservé pour usage ultérieur. Doit toujours être à zéro.                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| Consultation de l'Historique | 2     | Indique si l'historique des <i>Événements</i> est consultable (0 signifie non consultable, 1 signifie consultable).                                                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| Modification de l'Historique | 3     | Indique si l'historique des <i>Événements</i> est modifiable (0 signifie non modifiable, 1 signifie modifiable).                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| Réservé                      | 4-7   | Réservé pour usage ultérieur. Doit toujours être à zéro.                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| <b>Références</b>            |       |                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| HasComponent                 | 0..*  |                                                                                                                                                                                                                                        | Les <i>Références "Dispose d'un composant"</i> identifient les <i>Variables de Données</i> , les <i>Méthodes</i> et les <i>Objets</i> contenus dans l' <i>Objet</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| HasProperty                  | 0..*  |                                                                                                                                                                                                                                        | Les <i>Références "Dispose d'une Propriété"</i> identifient les <i>Propriétés</i> de l' <i>Objet</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| HasModellingRule             | 0..1  |                                                                                                                                                                                                                                        | Les <i>Objets</i> peuvent pointer vers, au plus, un <i>Objet Règle de Modélisation</i> en utilisant une <i>Référence "Dispose d'une Règle de Modélisation"</i> (pour plus de détails concernant les <i>Règles de Modélisation</i> voir 6.4.4).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| HasTypeDefinition            | 1     |                                                                                                                                                                                                                                        | La <i>Référence "Dispose d'une Définition de Type"</i> pointe vers la définition de type de l' <i>Objet</i> . Chaque <i>Objet</i> doit avoir précisément une définition de type et par conséquent être le <i>SourceNode</i> d'une seule et unique <i>Référence "Dispose d'une Définition de Type"</i> pointant vers un <i>Type d'Objet</i> . Voir 4.5 pour une description des définitions de types.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| HasEventSource               | 0..*  |                                                                                                                                                                                                                                        | Les <i>Références "Dispose d'une Source d'Événement"</i> pointent vers les sources d'événements de l' <i>Objet</i> . Les <i>Références</i> de ce type peuvent uniquement être utilisées pour des <i>Objets</i> dont le bit "S'Abonner à des Événements" est à Un dans l'Attribut " <i>Notificateur d'Événements</i> ". Une description plus détaillée est fournie en 7.17.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| HasNotifier                  | 0..*  |                                                                                                                                                                                                                                        | Les <i>Références "Dispose d'un Notificateur"</i> pointent vers les <i>Notificateurs</i> de l' <i>Objet</i> . Les <i>Références</i> de ce type peuvent uniquement être utilisées pour des <i>Objets</i> dont le bit "S'Abonner à des Événements" est à Un dans l'Attribut " <i>Notificateur d'Événements</i> ". Une description plus détaillée est fournie en 7.19.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| Organizes                    | 0..*  |                                                                                                                                                                                                                                        | Il convient que cette <i>Référence</i> soit uniquement utilisée pour des <i>Objets</i> du <i>Type d'Objet "Type de Dossier"</i> (voir 5.5.3).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| HasDescription               | 0..1  |                                                                                                                                                                                                                                        | Cette <i>Référence</i> doit uniquement être utilisée pour des <i>Objets</i> du <i>Type d'Objet "Type de Codage de Type de Données"</i> (voir 5.8.4).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |
| <other References>           | 0..*  |                                                                                                                                                                                                                                        | Les <i>Objets</i> peuvent contenir d'autres <i>Références</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |       |     |             |                            |   |                                                                                                                                                                                                                                        |         |   |                                                          |                              |   |                                                                                                                     |                              |   |                                                                                                                  |         |     |                                                          |

| Dénomination                  | Usage | Type de Données          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------------------|-------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Propriétés normalisées</b> |       |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| NodeVersion                   | O     | Chaîne                   | La <i>Propriété</i> «NodeVersion» est utilisée pour indiquer la version d'un <i>Nœud</i> .<br>La <i>Propriété</i> «NodeVersion» est mise à jour à chaque fois qu'une <i>Référence</i> est ajoutée ou retirée du <i>Nœud</i> auquel la <i>Propriété</i> appartient. Les modifications de la valeur de l' <i>Attribut</i> n'entraînent pas une modification de la <i>Version de Nœud</i> . Les <i>Clients</i> peuvent consulter la <i>Propriété</i> «NodeVersion» ou s'y abonner pour établir le moment où la structure d'un <i>Nœud</i> a changé. |
| Icon                          | O     | Image                    | La <i>Propriété</i> "Icône" fournit une image qui peut être utilisée par les <i>Clients</i> lorsqu'ils affichent le <i>Nœud</i> . Il est prévu que la <i>Propriété</i> "Icône" contienne une image relativement petite.                                                                                                                                                                                                                                                                                                                          |
| NamingRule                    | O     | Type de Règle de Nommage | La <i>Propriété</i> "Règle de Nommage" définit la <i>Règle de Nommage</i> d'une <i>Règle de Modélisation</i> (voir 6.4.4.2 pour plus de détails). Cette <i>Propriété</i> doit uniquement être utilisée pour des <i>Objets</i> de type "Type de Règle de Modélisation" défini en 6.4.4.                                                                                                                                                                                                                                                           |

La *Classe de Nœud* "Objets" hérite des *Attributs* de base de la *Classe de Nœud* de Base définie en 5.2.

L'*Attribut* obligatoire *EventNotifier* (*Notificateur d'Événements*) est utilisé pour indiquer si l'*Objet* peut être utilisé pour s'abonner à des *Événements* ou pour consulter et modifier l'historique des *Événements*.

La *Classe de Nœud* "Objets" utilise la *Référence* *HasComponent* (*Dispose d'un composant*) pour définir les *Variables de Données*, les *Objets* et les *Méthodes* d'un *Objet* donné.

Elle utilise la *Référence* *HasProperty* (*Dispose d'une Propriété*) pour définir les *Propriétés* d'un *Objet*. La *Propriété* "NodeVersion" est utilisée pour indiquer la version de l'*Objet*. La *Propriété* *Icon* (*Icône*) fournit une icône de l'*Objet*. La *Propriété* *NamingRule* (*Règle de Nommage*) définit la *Règle de Nommage* d'une *Règle de Modélisation* et doit uniquement être appliquée à des *Objets* de type "Type de Règle de Modélisation". Il n'y a pas d'autres *Propriétés* définies pour les *Objets* dans le présent document. D'autres parties de la série IEC 62541 peuvent définir des *Propriétés* supplémentaires pour les *Objets*.

Pour spécifier sa *Règle de Modélisation*, un *Objet* peut utiliser au plus une *Référence* "Dispose d'une Règle de Modélisation" pointant vers un *Objet* "Règle de Modélisation". Les *Règles de Modélisation* sont définies en 6.4.4.

Les *Références* *HasNotifier* (*Dispose d'un Notificateur*) et *HasEventSource* (*Dispose d'une Source d'Événement*) sont utilisées pour fournir des informations concernant les événements et ne peuvent être appliquées qu'aux *Objets* utilisés comme notificateurs d'événements. Une description plus détaillée est fournie en 7.17 et 7.19.

La *Référence* *HasTypeDefinition* (*Dispose d'une Définition de Type*) pointe vers le *Type* d'*Objet* utilisé comme définition de type de l'*Objet*.

Les *Objets* peuvent utiliser des *Références* supplémentaires pour définir des relations avec d'autres *Nœuds*. Aucune contrainte n'est appliquée aux types de *Références* utilisés ou aux *Classes de Nœuds* correspondant aux *Nœuds* pouvant être référencés. Cependant, le *Type de Référence* peut définir des restrictions qui excluent son utilisation pour des *Objets*. Des *Types de Références* normalisés sont décrits à l'Article 7.

Si l'*Objet* est utilisé comme une *Déclaration d'Instance* (voir 4.5), tous les *Nœuds* référencés avec des *Références hiérarchiques* dans un sens direct doivent alors avoir des *Noms de Navigation* uniques dans le contexte de cet *Objet*.

Si l'*Objet* est créé sur la base d'une *Déclaration d'Instance*, il doit alors avoir le même *Nom de Navigation* que sa *Déclaration d'Instance*.

### 5.5.2 Classe de Nœud "ObjectType" (Types d'Objets)

Les *Types d'Objets* fournissent des définitions pour les *Objets*. Les *Types d'Objets* sont définis en utilisant la *Classe de Nœud "ObjectType"* qui est spécifiée dans le Tableau 7.

**Tableau 7 – Classe de Nœud "ObjectType"**

| Dénomination                  | Usage | Type de Données      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------------------|-------|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Attributs</b>              |       |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Base NodeClass Attributes     | M     | --                   | Hérité de la <i>Classe de Nœud de Base</i> . Voir 5.2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| IsAbstract                    | M     | Booléen              | Un <i>Attribut</i> booléen prenant les valeurs suivantes:<br>VRAI Il s'agit d'un <i>Type d'Objet</i> abstrait, ce qui signifie qu'aucun <i>Objet</i> de ce type ne doit exister, mais uniquement des <i>Objets</i> de ses sous-types.<br>FAUX Il s'agit d'un <i>Type d'Objet</i> qui n'est pas abstrait, c'est-à-dire qu'il peut y avoir des <i>Objets</i> de ce type.                                                                                                                                                                           |
| <b>Références</b>             |       |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| HasComponent                  | 0..*  |                      | Les <i>Références "Dispose d'un composant"</i> identifient les <i>Variables de Données</i> , les <i>Méthodes</i> et les <i>Objets</i> contenus dans le <i>Type d'Objet</i> .<br>Le paragraphe 6.4. spécifie l'éventuelle instanciation des <i>Nœuds</i> référencés et la manière dont cette instanciation est réalisée lorsqu'un <i>Objet</i> de ce type est instancié.                                                                                                                                                                          |
| HasProperty                   | 0..*  |                      | Les <i>Références "Dispose d'une Propriété"</i> identifient les <i>Propriétés</i> du <i>Type d'Objet</i> . Le paragraphe 6.4. spécifie l'éventuelle instanciation des <i>Propriétés</i> et la manière dont cette instanciation est réalisée lorsqu'un <i>Objet</i> de ce type est instancié.                                                                                                                                                                                                                                                     |
| HasSubtype                    | 0..*  |                      | Les <i>Références "Dispose d'un sous-type"</i> indiquent les <i>Types d'Objets</i> qui sont des sous-types de ce type. La <i>Référence inverse "Sous-type de"</i> identifie le type parent de ce type.                                                                                                                                                                                                                                                                                                                                           |
| GeneratesEvent                | 0..*  |                      | Les <i>Références "Génère un Événement"</i> identifient le type d'instances d' <i>Événements</i> que ce type peut générer.                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <other References>            | 0..*  |                      | Les <i>Types d'Objets</i> peuvent contenir d'autres <i>Références</i> qui peuvent être instanciées par des <i>Objets</i> définis par ce <i>Type d'Objet</i> .                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Propriétés normalisées</b> |       |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| NodeVersion                   | 0     | Chaîne de caractères | La <i>Propriété "NodeVersion"</i> est utilisée pour indiquer la version d'un <i>Nœud</i> .<br>La <i>Propriété "NodeVersion"</i> est mise à jour à chaque fois qu'une <i>Référence</i> est ajoutée ou retirée du <i>Nœud</i> auquel la <i>Propriété</i> appartient. Les modifications de la valeur de l' <i>Attribut</i> n'entraînent pas une modification de la <i>Version de Nœud</i> . Les <i>Clients</i> peuvent consulter la <i>Propriété "NodeVersion"</i> ou s'y abonner pour établir le moment où la structure d'un <i>Nœud</i> a changé. |
| Icon                          | 0     | Image                | La <i>Propriété "Icône"</i> fournit une image qui peut être utilisée par les <i>Clients</i> lorsqu'ils affichent le <i>Nœud</i> . Il est prévu que la <i>Propriété "Icône"</i> contienne une image relativement petite.                                                                                                                                                                                                                                                                                                                          |

La *Classe de Nœud "ObjectType"* hérite des *Attributs* de base de la *Classe de Nœud de Base* définie en 5.2. L'*Attribut* supplémentaire *IsAbstract* (*Est Abstrait*) indique si le *Type d'Objet* est ou non abstrait.

La *Classe de Nœud "ObjectType"* utilise les *Références "Dispose d'un composant"* pour définir les *Variables de Données*, les *Objets* et les *Méthodes* d'un *Type d'Objet* donné.

La *Référence HasProperty* (*Dispose d'une Propriété*) est utilisée pour identifier les *Propriétés*. La *Propriété NodeVersion* (*Version de Nœud*) est utilisée pour indiquer la version du *Type d'Objet*. La *Propriété Icon* (*Icône*) fournit une icône du *Type d'Objet*. Il n'y a pas d'autres *Propriétés* définies pour les *Types d'Objets* dans le présent document. D'autres parties de la série IEC 62541 peuvent définir des *Propriétés* supplémentaires pour les *Types d'Objets*.

Les *Références HasSubtype* (*Dispose d'un Sous-type*) sont utilisées pour définir des sous-types de *Types d'Objets*. Les sous-types de *Type d'Objet* héritent des caractéristiques sémantiques générales du type parent. Les règles générales de sous-typage s'appliquent comme défini à l'Article 6. Il n'est pas exigé que la *Référence "HasSubtype"* soit fournie pour le supertype; il est cependant exigé que ce sous-type fournisse la *Référence* inverse à son supertype.

Les *Références GeneratesEvent* (*Génère un Événement*) identifient le type d'*Événements* que les instances de ce *Type d'Objet* peuvent générer. Ces *Objets* peuvent être la source d'un *Événement* de type spécifié ou d'un de ses sous-types. Il convient que les *Serveurs* fassent en sorte que les *Références "Génère un Événement"* soient bidirectionnelles. Il est cependant admis qu'elles soient unidirectionnelles lorsque le *Serveur* n'est pas capable de présenter le sens inverse pointant du *Type d'Événement* vers chaque *Type d'Objet* prenant en charge le *Type d'Événement*. À noter que l'*Attribut "Notificateur d'Événements"* d'un *Objet* et les *Références "Génère un Événement"* de ce *Type d'Objet* n'ont aucune relation. Les *Objets* qui peuvent générer des *Événements* peuvent ne pas être utilisés comme des *Objets* auxquels s'abonnent les *Clients* pour obtenir les notifications d'*Événements* correspondantes.

Les *Références "Génère un Événement"* sont facultatives, c'est-à-dire que les *Objets* peuvent générer des *Événements* d'un *Type d'Événement* qui n'est pas présenté par ce *Type d'Objet*.

Les *Types d'Objets* peuvent utiliser des *Références* supplémentaires pour définir des relations avec d'autres *Nœuds*. Aucune contrainte n'est appliquée aux types de *Références* utilisés ou aux *Classes de Nœuds* correspondant aux *Nœuds* pouvant être référencés. Cependant, le *Type de Référence* peut définir des restrictions qui excluent son utilisation pour des *Types d'Objets*. Des *Types de Références* normalisés sont décrits à l'Article 7.

Tous les *Nœuds* référencés avec des *Références hiérarchiques* doivent avoir des *Noms de Navigation* uniques dans le contexte d'un *Type d'Objet* (voir 4.5).

### 5.5.3 Type d'Objet normalisé "FolderType" (Type de Dossier)

Le *Type d'Objet FolderType* (*Type de Dossier*) est formellement défini dans l'IEC 62541-5. Son objectif est de fournir des *Objets* qui n'ont aucune valeur sémantique autre que l'organisation de l'*Espace d'Adressage*. Il est introduit un *Type de Référence* particulier pour ces *Objets "Dossiers"*, le *Type de Référence "Organise"*. Il convient toujours que le *SourceNode* d'une telle *Référence* soit une *Vue* ou un *Objet* du *Type d'Objet "Type de Dossier"*; le *TargetNode* peut être de n'importe quelle *Classe de Nœud*. Les *Références Organizes* (*Organise*) peuvent être utilisées en toute combinaison avec des *Références "Dispose d'une Référence Enfant"* (*HasComponent*, *HasProperty*, etc.; voir 7.5) et n'empêchent pas les boucles. Elles peuvent ainsi être utilisées pour couvrir de multiples hiérarchies.

### 5.5.4 Création du côté client d'Objets d'un Type d'Objet donné

Les *Objets* sont toujours fondés sur un *Type d'Objet*; ce qui signifie qu'ils ont une *Référence "Dispose d'une Définition de Type"* pointant vers son *Type d'Objet*.

Les *Clients* peuvent créer des *Objets* en utilisant le *Service "Ajouter des Nœuds"* défini dans l'IEC 62541-4. Le *Service* exige que le *Nœud de Définition de Type* de l'*Objet* soit spécifié. Un *Objet* créé par le *Service "Ajouter des Nœuds"* contient tous les composants définis par son *Type d'Objet* en fonction des *Règles de Modélisation* spécifiées pour les composants. Le *Serveur* peut toutefois ajouter des composants supplémentaires et des *Références* à l'*Objet* et à ses composants qui ne sont pas définis par le *Type d'Objet*. Ce comportement dépend du *Serveur*. Le *Type d'Objet* spécifie uniquement l'ensemble minimal de composants qui doit exister pour chaque *Objet* d'un *Type d'Objet* donné.

Outre le *Service "Ajouter des Nœuds"*, les *Types d'Objets* peuvent disposer d'une *Méthode* spéciale avec le *Nom de Navigation "Créer"*. Cette *Méthode* est utilisée pour créer un *Objet*

de ce *Type d'Objet*. Cette *Méthode* peut être utile pour créer des *Objets* lorsqu'il convient que les règles sémantiques de la création soient différentes du comportement par défaut attendu dans le contexte du *Service "Ajouter des Nœuds"*. Il convient par exemple que les valeurs soient totalement différentes des valeurs par défaut ou il convient que des *Objets* supplémentaires soient ajoutés, etc. Les arguments d'entrée et de sortie de cette *Méthode* dépendent du *Type d'Objet*; la seule caractéristique commune est le *Nom de Navigation* qui indique que cette *Méthode* créera un *Objet* sur la base du *Type d'Objet*. Il convient que les *Serveurs* ne fournissent pas une *Méthode* portant sur un *Type d'Objet* avec le *Nom de Navigation "Créer"* pour des raisons autres que la création d'*Objets* du *Type d'Objet*.

## 5.6 Variables

### 5.6.1 Généralités

Il est défini deux types de *Variables*: les *Propriétés* et les *Variables de Données*. Bien qu'elles soient utilisées de manière différente comme cela est décrit en 4.4 et qu'elles aient des contraintes différentes (décrites dans le reste de 5.6), elles utilisent la même *Classe de Nœud* (décrite en 5.6.2). Les contraintes de *Propriétés* fondées sur cette *Classe de Nœud* sont définies en 5.6.3, les contraintes de *Variables de Données* sont indiquées en 5.6.4.

### 5.6.2 Classe de Nœud "Variables"

Les *Variables* sont utilisées pour représenter des valeurs qui peuvent être simples ou complexes. Les *Variables* sont définies par des *Types de Variables*, spécifiés en 5.6.5.

Les *Variables* sont toujours définies comme des *Propriétés* ou des *Variables de Données* d'autres *Nœuds* de l'*Espace d'Adressage*. Elles ne sont jamais définies par elles-mêmes. Une *Variable* fait toujours partie d'au moins un autre *Nœud*, mais elle peut être liée à n'importe quel nombre d'autres *Nœuds*. Les *Variables* sont définies en utilisant la *Classe de Nœud "Variables"* spécifiée dans le Tableau 8.

**Tableau 8 – Classe de Nœud "Variables"**

| Dénomination     | Usage | Type de Données                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------|-------|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Attributs</b> |       |                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Base NodeClass   | M     | --                               | Hérité de la <i>Classe de Nœud de Base</i> . Voir 5.2.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Value            | M     | Defini par l'attribut "DataType" | La valeur la plus récente de la <i>Variable</i> dont dispose le <i>Serveur</i> . Son type de données est défini par l'Attribut "DataType". Il s'agit du seul Attribut auquel n'est associé aucun type de données. Ceci permet à toutes les <i>Variables</i> d'avoir une valeur définie par le même Attribut "Valeur".                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| DataType         | M     | Identificateur de Nœud           | Identificateur de Nœud de la définition de <i>Type de Données</i> pour l'Attribut "Valeur". Des <i>Types de Données</i> normalisés sont décrits à l'Article 8.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| ValueRank        | M     | Int32                            | Cet Attribut indique si l'Attribut "Valeur" de la <i>Variable</i> est une matrice et le nombre de dimensions de cette matrice.<br>Il peut prendre les valeurs suivantes:<br>$n > 1$ : La valeur est une matrice ayant le nombre spécifié de dimensions.<br>Unidimensionnelle (1): La valeur est une matrice unidimensionnelle.<br>Une Ou Plusieurs Dimensions (0): La valeur est une matrice à une ou plusieurs dimensions.<br>Scalaire (-1): La valeur n'est pas une matrice.<br>Indifférente (-2): La valeur peut être scalaire ou une matrice ayant n'importe quel nombre de dimensions.<br>Scalaire Ou Unidimensionnelle (-3): La valeur peut être scalaire ou une matrice unidimensionnelle.<br>NOTE Tous les Types de Données sont considérés comme scalaires, même s'ils ont une sémantique de type matrice comme une Chaîne d'Octets (ByteString) et une Chaîne (String). |

| Dénomination    | Usage | Type de Données                                                                                                                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |     |             |             |   |                                                                                                        |              |   |                                                                                                     |             |   |                                                                                                               |              |   |                                                                                                         |                |     |                                                                                                                                        |          |     |                                                          |
|-----------------|-------|----------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-----|-------------|-------------|---|--------------------------------------------------------------------------------------------------------|--------------|---|-----------------------------------------------------------------------------------------------------|-------------|---|---------------------------------------------------------------------------------------------------------------|--------------|---|---------------------------------------------------------------------------------------------------------|----------------|-----|----------------------------------------------------------------------------------------------------------------------------------------|----------|-----|----------------------------------------------------------|
| ArrayDimensions | O     | UInt32[]                                                                                                                               | <p>Cet <i>Attribut</i> spécifie la longueur de chaque dimension d'une valeur matricielle. L'<i>Attribut</i> permet de décrire la capacité de la <i>Variable</i> et non la taille courante.</p> <p>Le nombre d'éléments doit être égal à la valeur de l'<i>Attribut</i> "<i>Rang de Valeur</i>". Doit être nul si le <i>Rang de Valeur</i> ≤ 0.</p> <p>Une valeur 0 pour une dimension particulière indique que la dimension a une longueur variable.</p> <p>Par exemple, si une <i>Variable</i> est définie par la matrice C suivante:</p> <p style="padding-left: 20px;">Int32 ma Matrice[346];</p> <p>ainsi, ce <i>Type de Données de Variable</i> pointerait vers une Int32, le <i>Rang de Valeur</i> de la <i>Variable</i> prend la valeur 1 et l'<i>Attribut</i> "<i>Dimensions de Matrice</i>" indique une matrice dont l'une des entrées a la valeur 346.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |       |     |             |             |   |                                                                                                        |              |   |                                                                                                     |             |   |                                                                                                               |              |   |                                                                                                         |                |     |                                                                                                                                        |          |     |                                                          |
| AccessLevel     | M     | Octet                                                                                                                                  | <p>L'<i>Attribut</i> "<i>Niveau d'Accès</i>" est utilisé pour indiquer l'accessibilité (lecture/écriture) de la <i>Valeur</i> d'une <i>Variable</i> et si elle contient des données courantes et/ou historiques. Le <i>Niveau d'Accès</i> ne tient compte d'aucun droit d'accès utilisateur, ce qui signifie que même si la <i>Variable</i> est modifiable, cette possibilité peut être limitée à un certain utilisateur / groupe d'utilisateurs.</p> <p>Le <i>Niveau d'Accès</i> est un entier de 8 bits non signé dont la structure est définie dans le tableau suivant:</p> <table><tr><th>Champ</th><th>Bit</th><th>Description</th></tr><tr><td>CurrentRead</td><td>0</td><td>Indique si la valeur courante est consultable<br/>(0 signifie non consultable, 1 signifie consultable).</td></tr><tr><td>CurrentWrite</td><td>1</td><td>Indique si la valeur courante est modifiable<br/>(0 signifie non modifiable, 1 signifie modifiable).</td></tr><tr><td>HistoryRead</td><td>2</td><td>Indique si l'historique de la valeur est consultable<br/>(0 signifie non consultable, 1 signifie consultable).</td></tr><tr><td>HistoryWrite</td><td>3</td><td>Indique si l'historique de la valeur est modifiable (0 signifie non modifiable, 1 signifie modifiable).</td></tr><tr><td>SemanticChange</td><td>4</td><td>Indique si la <i>Variable</i> utilisée comme <i>Propriété</i> génère des "<i>Événements de Modification Sémantique</i>" (voir 9.33).</td></tr><tr><td>Reserved</td><td>5:7</td><td>Réservé pour usage ultérieur. Doit toujours être à zéro.</td></tr></table> <p>Les deux premiers bits indiquent également si une valeur courante de cette <i>Variable</i> est disponible et les deux bits suivants indiquent si l'historique de la <i>Variable</i> est disponible via le <i>Serveur</i> OPC UA.</p> | Champ | Bit | Description | CurrentRead | 0 | Indique si la valeur courante est consultable<br>(0 signifie non consultable, 1 signifie consultable). | CurrentWrite | 1 | Indique si la valeur courante est modifiable<br>(0 signifie non modifiable, 1 signifie modifiable). | HistoryRead | 2 | Indique si l'historique de la valeur est consultable<br>(0 signifie non consultable, 1 signifie consultable). | HistoryWrite | 3 | Indique si l'historique de la valeur est modifiable (0 signifie non modifiable, 1 signifie modifiable). | SemanticChange | 4   | Indique si la <i>Variable</i> utilisée comme <i>Propriété</i> génère des " <i>Événements de Modification Sémantique</i> " (voir 9.33). | Reserved | 5:7 | Réservé pour usage ultérieur. Doit toujours être à zéro. |
| Champ           | Bit   | Description                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |             |   |                                                                                                        |              |   |                                                                                                     |             |   |                                                                                                               |              |   |                                                                                                         |                |     |                                                                                                                                        |          |     |                                                          |
| CurrentRead     | 0     | Indique si la valeur courante est consultable<br>(0 signifie non consultable, 1 signifie consultable).                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |             |   |                                                                                                        |              |   |                                                                                                     |             |   |                                                                                                               |              |   |                                                                                                         |                |     |                                                                                                                                        |          |     |                                                          |
| CurrentWrite    | 1     | Indique si la valeur courante est modifiable<br>(0 signifie non modifiable, 1 signifie modifiable).                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |             |   |                                                                                                        |              |   |                                                                                                     |             |   |                                                                                                               |              |   |                                                                                                         |                |     |                                                                                                                                        |          |     |                                                          |
| HistoryRead     | 2     | Indique si l'historique de la valeur est consultable<br>(0 signifie non consultable, 1 signifie consultable).                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |             |   |                                                                                                        |              |   |                                                                                                     |             |   |                                                                                                               |              |   |                                                                                                         |                |     |                                                                                                                                        |          |     |                                                          |
| HistoryWrite    | 3     | Indique si l'historique de la valeur est modifiable (0 signifie non modifiable, 1 signifie modifiable).                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |             |   |                                                                                                        |              |   |                                                                                                     |             |   |                                                                                                               |              |   |                                                                                                         |                |     |                                                                                                                                        |          |     |                                                          |
| SemanticChange  | 4     | Indique si la <i>Variable</i> utilisée comme <i>Propriété</i> génère des " <i>Événements de Modification Sémantique</i> " (voir 9.33). |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |             |   |                                                                                                        |              |   |                                                                                                     |             |   |                                                                                                               |              |   |                                                                                                         |                |     |                                                                                                                                        |          |     |                                                          |
| Reserved        | 5:7   | Réservé pour usage ultérieur. Doit toujours être à zéro.                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |             |   |                                                                                                        |              |   |                                                                                                     |             |   |                                                                                                               |              |   |                                                                                                         |                |     |                                                                                                                                        |          |     |                                                          |
| UserAccessLevel | M     | Octet                                                                                                                                  | <p>L'<i>Attribut</i> "<i>Niveau d'Accès Utilisateur</i>" indique l'accessibilité (lecture/écriture) de la <i>Valeur</i> d'une <i>Variable</i> et si elle contient des données courantes ou historiques en tenant compte des droits d'accès de l'utilisateur.</p> <p>Le <i>Niveau d'Accès Utilisateur</i> est un entier à 8 bits non signé dont la structure est définie dans le tableau suivant:</p> <table><tr><th>Champ</th><th>Bit</th><th>Description</th></tr><tr><td>CurrentRead</td><td>0</td><td>Indique si la valeur courante est consultable<br/>(0 signifie non consultable, 1 signifie consultable).</td></tr><tr><td>CurrentWrite</td><td>1</td><td>Indique si la valeur courante est modifiable<br/>(0 signifie non modifiable, 1 signifie modifiable).</td></tr><tr><td>HistoryRead</td><td>2</td><td>Indique si l'historique de la valeur est consultable<br/>(0 signifie non consultable, 1 signifie consultable).</td></tr><tr><td>HistoryWrite</td><td>3</td><td>Indique si l'historique de la valeur est modifiable (0 signifie non modifiable, 1 signifie modifiable).</td></tr><tr><td>Reserved</td><td>4:7</td><td>Réservé pour usage ultérieur. Doit toujours être à zéro.</td></tr></table> <p>Les deux premiers bits indiquent également si une valeur courante de</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Champ | Bit | Description | CurrentRead | 0 | Indique si la valeur courante est consultable<br>(0 signifie non consultable, 1 signifie consultable). | CurrentWrite | 1 | Indique si la valeur courante est modifiable<br>(0 signifie non modifiable, 1 signifie modifiable). | HistoryRead | 2 | Indique si l'historique de la valeur est consultable<br>(0 signifie non consultable, 1 signifie consultable). | HistoryWrite | 3 | Indique si l'historique de la valeur est modifiable (0 signifie non modifiable, 1 signifie modifiable). | Reserved       | 4:7 | Réservé pour usage ultérieur. Doit toujours être à zéro.                                                                               |          |     |                                                          |
| Champ           | Bit   | Description                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |             |   |                                                                                                        |              |   |                                                                                                     |             |   |                                                                                                               |              |   |                                                                                                         |                |     |                                                                                                                                        |          |     |                                                          |
| CurrentRead     | 0     | Indique si la valeur courante est consultable<br>(0 signifie non consultable, 1 signifie consultable).                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |             |   |                                                                                                        |              |   |                                                                                                     |             |   |                                                                                                               |              |   |                                                                                                         |                |     |                                                                                                                                        |          |     |                                                          |
| CurrentWrite    | 1     | Indique si la valeur courante est modifiable<br>(0 signifie non modifiable, 1 signifie modifiable).                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |             |   |                                                                                                        |              |   |                                                                                                     |             |   |                                                                                                               |              |   |                                                                                                         |                |     |                                                                                                                                        |          |     |                                                          |
| HistoryRead     | 2     | Indique si l'historique de la valeur est consultable<br>(0 signifie non consultable, 1 signifie consultable).                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |             |   |                                                                                                        |              |   |                                                                                                     |             |   |                                                                                                               |              |   |                                                                                                         |                |     |                                                                                                                                        |          |     |                                                          |
| HistoryWrite    | 3     | Indique si l'historique de la valeur est modifiable (0 signifie non modifiable, 1 signifie modifiable).                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |             |   |                                                                                                        |              |   |                                                                                                     |             |   |                                                                                                               |              |   |                                                                                                         |                |     |                                                                                                                                        |          |     |                                                          |
| Reserved        | 4:7   | Réservé pour usage ultérieur. Doit toujours être à zéro.                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |       |     |             |             |   |                                                                                                        |              |   |                                                                                                     |             |   |                                                                                                               |              |   |                                                                                                         |                |     |                                                                                                                                        |          |     |                                                          |

| Dénomination                  | Usage | Type de Données                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------------------|-------|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                               |       |                                  | cette <i>Variable</i> est disponible et les deux bits suivants indiquent si l'historique de la <i>Variable</i> est disponible via le <i>Serveur</i> OPC UA.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| MinimumSamplingInterval       | O     | Durée                            | L'Attribut " <i>Intervalle d'Échantillonnage Minimal</i> " indique la fréquence d'actualisation de la <i>Valeur</i> de la <i>Variable</i> . Il spécifie (en millisecondes) la fréquence raisonnable d'échantillonnage des modifications de la valeur (pour une description détaillée de l'intervalle d'échantillonnage, voir l'IEC 62541-4).<br>Un <i>Intervalle d'Échantillonnage Minimal</i> de 0 indique que le <i>Serveur</i> surveille l'élément en continu. Un <i>Intervalle d'Échantillonnage Minimal</i> de -1 signifie qu'il est indéterminé.                                                                                                                                                                                                                                                                                                                                                                                                            |
| Historizing                   | M     | Booléen                          | L'Attribut " <i>Historisation</i> " indique si le <i>Serveur</i> recueille activement les données historiques de la <i>Variable</i> . Il est différent de l'Attribut " <i>Niveau d'Accès</i> " qui indique si la <i>Variable</i> dispose d'éventuelles données historiques. Une valeur "VRAI" indique que le <i>Serveur</i> recueille activement des données. Une valeur "FAUX" indique que le <i>Serveur</i> ne recueille pas activement des données. La valeur par défaut est FAUX.                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Références</b>             |       |                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| HasModellingRule              | 0..1  |                                  | Les <i>Variables</i> peuvent pointer vers, au plus, un <i>Objet Règle de Modélisation</i> en utilisant une <i>Référence "Dispose d'une Règle de Modélisation"</i> (pour plus de détails concernant les <i>Règles de Modélisation</i> voir 6.4.4).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| HasProperty                   | 0..*  |                                  | Les <i>Références "Dispose d'une Propriété"</i> sont utilisées pour identifier les <i>Propriétés</i> d'une <i>Variable de Données</i> .<br>Il n'est pas admis que les <i>Propriétés</i> soient le <i>SourceNode</i> des <i>Références "Dispose d'une Propriété"</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| HasComponent                  | 0..*  |                                  | Les <i>Références "Dispose d'un Composant"</i> sont utilisées par des <i>Variables de Données</i> complexes pour identifier les <i>Variables de Données</i> qui les composent.<br>Les <i>Propriétés</i> ne sont pas autorisées à utiliser cette <i>Référence</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| HasTypeDefinition             | 1     |                                  | La <i>Référence "Dispose d'une Définition de Type"</i> pointe vers la définition de type de la <i>Variable</i> . Chaque <i>Variable</i> doit avoir précisément une définition de type et par conséquent être le <i>SourceNode</i> d'une seule et unique <i>Référence "Dispose d'une Définition de Type"</i> pointant vers un <i>Type de Variable</i> . Voir 4.5 pour une description des définitions de types.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <other References>            | 0..*  |                                  | Les <i>Variables de Données</i> peuvent être le <i>SourceNode</i> de toute autre <i>Référence</i> .<br>Les <i>Propriétés</i> ne peuvent être le <i>SourceNode</i> que d'une quelconque <i>Référence non hiérarchique</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Propriétés normalisées</b> |       |                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| NodeVersion                   | O     | Chaîne                           | La <i>Propriété "NodeVersion"</i> est utilisée pour indiquer la version d'une <i>Variable de Données</i> . Elle ne s'applique pas aux <i>Propriétés</i> .<br>La <i>Propriété "NodeVersion"</i> est mise à jour à chaque fois qu'une <i>Référence</i> est ajoutée ou retirée du <i>Nœud</i> auquel la <i>Propriété</i> appartient. Sauf pour ce qui concerne l'Attribut " <i>DataType</i> ", les modifications de la valeur de l'Attribut n'entraînent pas une modification de la <i>Version de Nœud</i> . Les <i>Clients</i> peuvent consulter la <i>Propriété «NodeVersion»</i> ou s'y abonner pour établir le moment où la structure d'un <i>Nœud</i> a changé.<br>Bien que la relation entre une <i>Variable</i> et son <i>Type de Données</i> ne soit pas modélisée en utilisant des <i>Références</i> , les modifications apportées à l'Attribut " <i>DataType</i> " d'une <i>Variable</i> entraînent une mise à jour de la <i>Propriété "NodeVersion"</i> . |
| LocalTime                     | O     | Fuseau Horaire - Type de Données | La <i>Propriété "Heure Locale"</i> est uniquement utilisée pour des <i>Variables de Données</i> . Elle ne s'applique pas aux <i>Propriétés</i> .<br>Cette <i>Propriété</i> est une structure contenant le Décalage ainsi que le pointeur "Décalage d'Heure d'Été". Le Décalage spécifie la différence temporelle (en minutes) entre l'Horodatage Source (UTC) associé à une valeur au lieu où la valeur a été obtenue. L'Horodatage Source est défini dans l'IEC 62541-4.<br>Si le "Décalage d'Heure d'Été" est VRAI, dans ce cas l'Heure d'Été/normalisée au lieu d'origine est en vigueur et le Décalage inclut la correction de l'heure d'été. S'il est FAUX, dans ce cas le Décalage n'inclut pas la correction de l'heure d'été et celle-ci peut avoir ou ne pas avoir été en vigueur.                                                                                                                                                                       |
| DataTypeVersion               | O     | Chaîne                           | Uniquement utilisé pour des <i>Variables</i> du <i>Type de Variable "Type de Dictionnaire de Type de Données"</i> et " <i>Type de Description de Type de Données</i> ", comme décrit en 5.8.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

| Dénomination       | Usage | Type de Données | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------|-------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DictionaryFragment | O     | Chaîne d'Octets | Uniquement utilisé pour des <i>Variables</i> du Type de Variable "Type de Description de Type de Données" comme décrit en 5.8.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| AllowNulls         | O     | Booléen         | La <i>Propriété</i> "Permet des Valeurs Nulles" est uniquement utilisée pour des <i>Variables de Données</i> . Elle ne s'applique pas aux <i>Propriétés</i> . Cette <i>Propriété</i> indique si une valeur nulle est admise pour l' <i>Attribut</i> "Valeur" de la <i>Variable de Données</i> . Si elle est mise sur vrai, le <i>Serveur</i> peut renvoyer des valeurs nulles et accepter l'écriture de valeurs nulles. S'il est mis sur faux, le <i>Serveur</i> ne doit jamais renvoyer de valeur nulle et doit refuser toute requête d'écriture d'une valeur nulle. Si cette <i>Propriété</i> n'est pas fournie, l'admission ou non des valeurs nulles est spécifique au <i>Serveur</i> . |
| ValueAsText        | O     | Texte Localisé  | Uniquement utilisé pour des <i>Variables de Données</i> ayant un Type de Données "Énumération". Cette <i>Propriété</i> facultative fournit la représentation du texte localisé de la valeur d'énumération. Elle peut être utilisée par des <i>Clients</i> qui s'intéressent seulement à l'affichage du texte pour s'abonner à la <i>Propriété</i> au lieu de l'attribut "Valeur".                                                                                                                                                                                                                                                                                                           |
| MaxStringLength    | O     | UInt32          | Uniquement utilisé pour des <i>Variables de Données</i> ayant un Type de Données "Chaîne". Cette <i>Propriété</i> facultative indique le nombre maximal de caractères pris en charge par la <i>Variable de Données</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| MaxArrayLength     | O     | UInt32          | Uniquement utilisé pour des <i>Variables de Données</i> ayant leur <i>Attribut</i> "ValueRank" (Rang de Valeur) non défini comme scalaire. Cette <i>Propriété</i> facultative indique la longueur maximale d'une matrice prise en charge par la <i>Variable de Données</i> . Dans une matrice multidimensionnelle, elle indique la longueur totale. Par exemple une matrice tridimensionnelle de 2 x 3 x 10 a une longueur de matrice de 60.                                                                                                                                                                                                                                                |
| EngineeringUnits   | O     | Information EU  | Uniquement utilisé pour des <i>Variables de Données</i> ayant un Type de Données "Nombre". Cette <i>Propriété</i> facultative indique les unités techniques pour la valeur de la <i>Variable de Données</i> (par exemple hertz ou secondes). Des détails au sujet de la <i>Propriété</i> et au sujet des unités techniques qu'il convient d'utiliser sont donnés dans l'IEC 62541-8. Le Type de Données "Information UE" est aussi défini dans l'IEC 62541-8.                                                                                                                                                                                                                               |

La *Classe de Nœud* "Variables" hérite des *Attributs* de base de la *Classe de Nœud de Base* définie en 5.2.

La *Classe de Nœud* "Variables" définit également un ensemble d'*Attributs* qui décrit la valeur du temps d'exécution de la *Variable*. L'*Attribut* "Valeur" représente la valeur de la *Variable*. Les *Attributs* *DataType* (Type de Données), *ValueRank* (Rang de Valeur) et *ArrayDimensions* (Dimensions de Matrice) permettent de décrire des valeurs simples et complexes.

L'*Attribut* *AccessLevel* (Niveau d'Accès) indique l'accessibilité de la *Valeur* d'une *Variable* sans tenir compte des droits d'accès de l'utilisateur. Si le *Serveur* OPC UA n'a pas la possibilité d'obtenir les informations de Niveau d'Accès à partir du système sous-jacent, il convient alors de déclarer qu'il est consultable et modifiable. Si une opération de lecture ou d'écriture est invoquée pour la *Variable*, il convient alors que le *Serveur* transfère cette requête et renvoie le *Code de Statut* correspondant même si une telle requête est rejetée. Les *Codes de Statut* sont définis dans l'IEC 62541-4.

Le bit *SemanticChange* (Modifications Sémantiques) de l'*Attribut* *AccessLevel* (Niveau d'Accès) doit être établi lorsque la *Propriété* décrit la sémantique du *Nœud* qui possède la *Propriété* et, en cas de modification de la valeur de la *Propriété*, des "Événements de Modification Sémantique" sont générés. Par exemple, le bit d'une *Propriété* décrivant l'unité technique d'une *DataVariable* est établi tandis que celui d'une *Propriété* contenant une Icône de la *DataVariable* ne l'est pas. Ce comportement est exactement le même que celui décrit par le bit "Modifications Sémantiques" du *Code de Statut* défini dans l'IEC 62541-4. Cependant, si l'utilisateur s'abonne à une *Variable*, il convient alors qu'il examine le *Code de Statut* afin d'identifier une éventuelle modification de la sémantique et recevoir ces informations avant de traiter la valeur de la *Variable*.

L'*Attribut* *UserAccessLevel* (Niveau d'Accès Utilisateur) indique l'accessibilité de la *Valeur* d'une *Variable* en tenant compte des droits d'accès de l'utilisateur. Si le *Serveur* OPC UA n'a pas la possibilité d'obtenir des informations concernant les droits d'accès de l'utilisateur à

partir du système sous-jacent, il convient alors d'utiliser le même masque de bit que dans l'Attribut "Niveau d'Accès". L'Attribut *UserAccessLevel* (Niveau d'Accès Utilisateur) peut limiter l'accessibilité indiquée par l'Attribut "Niveau d'Accès", mais ne peut l'outrepasser.

L'Attribut *MinimumSamplingInterval* (Intervalle d'Échantillonnage Minimal) spécifie la rapidité du *Serveur* pour un échantillonnage raisonnable des modifications de la valeur. L'exactitude de cette valeur (l'aptitude du *Serveur* à atteindre ses performances "optimales") peut être fortement affectée par la charge du système et d'autres facteurs.

L'Attribut *Historizing* (Historisation) indique si le *Serveur* recueille activement les données historiques de la *Variable*. Voir l'IEC 62541-11 pour plus de détails sur les *Variables* d'historisation.

Les *Clients* peuvent lire ou écrire des valeurs de *Variables*, ou surveiller la modification des valeurs, comme spécifié dans l'IEC 62541-4. L'IEC 62541-8 définit des règles supplémentaires d'utilisation des *Services* en automatisation.

Pour spécifier sa *Règle de Modélisation*, une *Variable* peut utiliser au plus une *Référence* "Dispose d'une Règle de Modélisation" pointant vers un *Objet* "Règle de Modélisation". Les *Règles de Modélisation* sont définies en 6.4.4.

Si la *Variable* est créée sur la base d'une *Déclaration d'Instance* (voir 4.5), elle doit avoir le même *Nom de Navigation* que sa *Déclaration d'Instance*.

Les autres *Références* sont décrites séparément pour des *Propriétés* et *Variables de Données* dans le reste de 5.6.

### 5.6.3 Propriétés

Les *Propriétés* sont utilisées pour définir les caractéristiques de *Nœuds*. Les *Propriétés* sont définies en utilisant la *Classe de Nœud* "Variables" spécifiée dans le Tableau 8. Cependant, elles limitent leur utilisation.

Les *Propriétés* sont les feuilles de toute arborescence hiérarchique; par conséquent, elles ne doivent pas être le *SourceNode* d'éventuelles *Références hiérarchiques*. Elles comprennent la *Référence* "HasComponent" ou "HasProperty", ce qui signifie que des *Propriétés* ne contiennent pas elles-mêmes de *Propriétés* et ne peuvent présenter leur structure complexe. Cependant, elles peuvent être le *SourceNode* de toute *Référence non hiérarchique*.

La *Référence* "Dispose d'une Définition de Type" pointe vers le *Type de Variable* de la *Propriété*. Sachant que les *Propriétés* sont identifiées de manière unique par leur *Nom de Navigation*, toutes les *Propriétés* doivent pointer vers le *Type de Propriété* défini dans l'IEC 62541-5.

Les *Propriétés* doivent toujours être définies dans le contexte d'un autre *Nœud* et doivent être le *TargetNode* d'au moins une *Référence* "Dispose d'une Propriété". Pour les différencier des *Variables de Données*, elles ne doivent pas être le *TargetNode* de toute *Référence* "HasComponent". Ainsi, une *Référence* "Dispose d'une Propriété" pointant vers une *Variable Nœud* définit ce *Nœud* comme une *Propriété*.

Le *Nom de Navigation* d'une *Propriété* est toujours unique dans le contexte d'un *Nœud*. Il n'est pas admis qu'un *Nœud* fasse référence à deux *Variables* en utilisant des *Références* "Dispose d'une Propriété" ayant le même *Nom de Navigation*.

### 5.6.4 Variable de Données

Les *Variables de Données* représentent le contenu d'un *Objet*. Les *Variables de Données* sont définies en utilisant la *Classe de Nœud* "Variables" spécifiée dans le Tableau 8.

Les *Variables de Données* identifient leurs *Propriétés* en utilisant des *Références "Dispose d'une Propriété"*. Les *Variables de Données* complexes utilisent des *Références "HasComponent"* pour présenter les *Variables de Données* qui les composent.

La *Propriété NodeVersion (Version de Nœud)* indique la version de la *Variable de Données*. La *Propriété LocalTime (Heure Locale)* indique la différence entre l'Horodatage Source de la valeur et l'heure normale au lieu où la valeur a été obtenue. La *Propriété DataTypeVersion (Version de Type de Données)* est utilisée uniquement pour les *Dictionnaires de Types de Données* et les *Descriptions de Types de Données*, comme défini en 5.8. La *Propriété normalisée DictionaryFragment (Fragment de Dictionnaire)* est utilisée uniquement pour les *Descriptions de Types de Données* comme défini en 5.8. La *Propriété AllowNulls (Permet des Valeurs Nulles)* indique si les valeurs nulles sont admises pour l'Attribut "Valeur". La *Propriété ValueAsText (Valeur comme Texte)* fournit une représentation du texte localisé pour les valeurs d'énumération. La *Propriété MaxStringLength (Longueur de Chaîne Maximale)* indique la longueur maximale admise d'une valeur de chaîne et la *Propriété MaxArrayLength (Longueur de Matrice Maximale)* indique la longueur maximale de matrice admise de la valeur. La *Propriété EngineeringUnits (Unités Techniques)* indique les unités techniques de la valeur. Il n'y a pas d'autres *Propriétés* définies pour les *Variables de Données* dans le présent document. D'autres parties de la série IEC 62541 peuvent définir des *Propriétés* supplémentaires pour les *Variables de Données*. L'IEC 62541-8 définit un ensemble de *Propriétés* qui peuvent être utilisées pour les *Variables de Données*.

Les *Variables de Données* peuvent utiliser des *Références* supplémentaires pour définir des relations avec d'autres *Nœuds*. Aucune contrainte n'est appliquée aux types de *Références* utilisés ou aux *Classes de Nœuds* correspondant aux *Nœuds* pouvant être référencés. Cependant, le *Type de Référence* peut définir des restrictions qui excluent son utilisation pour des *Variables de Données*. Des *Types de Références* normalisés sont décrits à l'Article 7.

Il est prévu qu'une *DataVariable* soit définie dans le contexte d'un *Objet*. Cependant, des *Variables de Données* complexes peuvent présenter d'autres *Variables de Données*, et les *Types d'Objets* et *Types de Variables* complexes peuvent également contenir des *Variables de Données*. De ce fait, chaque *DataVariable* doit être le *TargetNode* d'au moins une *Référence "HasComponent"* provenant d'un *Objet*, d'un *Type d'Objet*, d'une *DataVariable* ou d'un *Type de Variable*. Les *Variables de Données* ne doivent pas être le *TargetNode* d'une quelconque *Référence "Dispose d'une Propriété"*. Par conséquent, une *Référence "HasComponent"* pointant vers une *Variable Nœud* l'identifie comme étant une *Variable de Données*.

La *Référence "Dispose d'une Définition de Type"* pointe vers le *Type de Variable* utilisé comme définition de type de la *Variable de Données*.

Si la *Variable de Données* est utilisée comme *Déclaration d'Instance* (voir 4.5), tous les *Nœuds* référencés avec des *Références hiérarchiques* dans le sens direct doivent avoir des *Noms de Navigation* uniques dans le contexte de cette *Variable de Données*.

### 5.6.5 Classe de nœud "VariableType" (Types de Variables)

Les *Types de Variables* sont utilisés pour fournir des définitions de type pour les *Variables*. Les *Types de Variables* sont définis en utilisant la *Classe de Nœud "VariableType"* spécifiée dans le Tableau 9.

Tableau 9 – Classe de nœud "VariableType"

| Dénomination                 | Usage | Type de Données                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------------------|-------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Attributs</b>             |       |                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Base NodeClass<br>Attributes | M     | --                                | Hérité de la <i>Classe de Nœud de Base</i> . Voir 5.2.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Value                        | O     | Défini par l'attribut "Data-Type" | La <i>valeur</i> par défaut des instances de ce type.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| DataType                     | M     | Identificateur de Nœud            | <i>Identificateur de Nœud</i> de la définition de type de données pour des instances de ce type.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| ValueRank                    | M     | Int32                             | Cet <i>Attribut</i> indique si l'Attribut "Valeur" du <i>Type de Variable</i> est une matrice et le nombre de dimensions de cette matrice. Il peut prendre les valeurs suivantes:<br>$n > 1$ : La valeur est une matrice ayant le nombre spécifié de dimensions.<br>Unidimensionnelle (1): La valeur est une matrice unidimensionnelle.<br>Une Ou Plusieurs Dimensions (0): La valeur est une matrice à une ou plusieurs dimensions.<br>Scalaire (-1): La valeur n'est pas une matrice.<br>Indifférente (-2): La valeur peut être scalaire ou une matrice ayant n'importe quel nombre de dimensions.<br>Scalaire Ou Unidimensionnelle (-3): La valeur peut être scalaire ou une matrice unidimensionnelle.<br>NOTE Tous les Types de Données sont considérés comme scalaires, même s'ils ont une sémantique de type matrice comme une Chaîne d'Octets (ByteString) et une Chaîne (String). |
| ArrayDimensions              | O     | UInt32[]                          | Cet <i>Attribut</i> spécifie la longueur de chaque dimension d'une valeur matricielle. L'Attribut permet de décrire la capacité du <i>Type de Variable</i> et non la taille courante. Le nombre d'éléments doit être égal à la valeur de l'Attribut "Rang de Valeur". Doit être nul si le <i>Rang de Valeur</i> $\leq 0$ . Une valeur 0 pour une dimension particulière indique que la dimension a une longueur variable. Par exemple, si un <i>Type de Variable</i> est défini par la matrice C suivante:<br>Int32 maMatrice[346];<br>ainsi, ce <i>Type de Données</i> de <i>Type de Variable</i> pointerait vers une Int32, le <i>Rang de Valeur</i> du <i>Type de Variable</i> prend la valeur 1 et l'Attribut <i>Dimensions de Matrice</i> indique une matrice dont l'une des entrées a la valeur 346.                                                                                 |
| IsAbstract                   | M     | Booléen                           | Un <i>Attribut</i> booléen prenant les valeurs suivantes:<br>VRAI Il s'agit d'un <i>Type de Variable</i> abstrait, ce qui signifie qu'aucune <i>Variable</i> de ce type ne doit exister, mais uniquement ses sous-types.<br>FAUX Il s'agit d'un <i>Type de Variable</i> qui n'est pas abstrait, ce qui signifie qu'il peut y avoir des <i>Variables</i> de ce type.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Références</b>            |       |                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| HasProperty                  | 0..*  |                                   | Les <i>Références "Dispose d'une Propriété"</i> sont utilisées pour identifier les <i>Propriétés</i> du <i>Type de Variable</i> . Les <i>Nœuds</i> référencés peuvent être instanciés par des instances de ce type, en fonction des <i>Règles de Modélisation</i> définies en 6.4.4.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| HasComponent                 | 0..*  |                                   | Les <i>Références "HasComponent"</i> sont utilisées pour des <i>Types de Variables</i> complexes afin d'identifier les <i>Variables de Données</i> qui les composent. Les <i>Types de Variables</i> complexes ne peuvent être utilisés que pour des <i>Variables de Données</i> . Les <i>Nœuds</i> référencés peuvent être instanciés par des instances de ce type, en fonction des <i>Règles de Modélisation</i> définies en 6.4.4.                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| HasSubtype                   | 0..*  |                                   | Les <i>Références «HasSubtype»</i> indiquent les <i>Types de Variables</i> qui sont des sous-types de ce type. La <i>Référence inverse "Sous-type de"</i> identifie le type parent de ce type.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

| Dénomination                  | Usage | Type de Données | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------------------|-------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GeneratesEvent                | 0..*  |                 | Les <i>Références "Génère un Événement"</i> identifient le type d'instances d' <i>Événements</i> que ce type peut générer.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <other References>            | 0..*  |                 | Les <i>Types de Variables</i> peuvent contenir d'autres <i>Références</i> qui peuvent être instanciées par des <i>Variables</i> définies par ce <i>Type de Variable</i> . Les <i>Règles de Modélisation</i> sont définies en 6.4.4.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Propriétés normalisées</b> |       |                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| NodeVersion                   | O     | Chaîne          | La <i>Propriété "NodeVersion"</i> est utilisée pour indiquer la version d'un <i>Nœud</i> .<br>La <i>Propriété "NodeVersion"</i> est mise à jour à chaque fois qu'une <i>Référence</i> est ajoutée ou retirée du <i>Nœud</i> auquel la <i>Propriété</i> appartient. Sauf pour ce qui concerne l' <i>Attribut "DataType"</i> , les modifications de la valeur de l' <i>Attribut</i> n'entraînent pas une modification de la <i>Version de Nœud</i> . Les <i>Clients</i> peuvent consulter la <i>Propriété "NodeVersion"</i> ou s'y abonner pour établir le moment où la structure d'un <i>Nœud</i> a changé.<br>Bien que la relation entre un <i>Type de Variable</i> et son <i>Type de Données</i> ne soit pas modélisée en utilisant des <i>Références</i> , les modifications apportées à l' <i>Attribut "DataType"</i> d'un <i>Type de Variable</i> entraînent une mise à jour de la <i>Propriété "NodeVersion"</i> . |

La *Classe de Nœud "VariableType"* hérite des *Attributs* de la *Classe de Nœud de Base* définie en 5.2. La *Classe de Nœud "VariableType"* définit également un ensemble d'*Attributs* qui décrit la valeur par défaut ou la valeur initiale de son instance "*Variables*". L'*Attribut "Valeur"* représente la valeur par défaut. Les *Attributs DataType (Type de Données)*, *ValueRank (Rang de Valeur)* et *ArrayDimensions (Dimensions de Matrice)* permettent de décrire des valeurs simples et complexes. L'*Attribut IsAbstract (Est Abstrait)* indique si le type peut être directement instancié.

La *Classe de Nœud "VariableType"* utilise des *Références "Dispose d'une Propriété"* pour définir les *Propriétés* et utilise les *Références "HasComponent"* pour définir les *Variables de Données*. Leur éventuelle instanciation dépend des *Règles de Modélisation* définies en 6.4.4.

La *Propriété "NodeVersion"* indique la version du *Type de Variable*. Il n'y a pas d'autres *Propriétés* définies pour les *Types de Variables* dans le présent document. D'autres parties de la série IEC 62541 peuvent définir des *Propriétés* supplémentaires pour les *Types de Variables*. L'IEC 62541-8 définit un ensemble de *Propriétés* qui peut être utilisé pour les *Types de Variables*.

Les *Références HasSubtype (Dispose d'un Sous-type)* sont utilisées pour définir des sous-types de *Types de Variables*. Les sous-types de *Types de Variables* héritent des caractéristiques sémantiques générales du type parent. Les règles générales de sous-typage sont définies à l'Article 6. Il n'est pas exigé que la *Référence "HasSubtype"* soit fournie pour le supertype; il est cependant exigé que ce sous-type fournisse la *Référence* inverse à son supertype.

Les *Références GeneratesEvent (Génère un Événement)* indiquent que les *Variables* du *Type de Variable* peuvent être la source d'un *Événement* du *Type d'Événement* spécifié ou d'un de ses sous-types. Il convient que les *Serveurs* fassent en sorte que les *Références "Génère un Événement"* soient bidirectionnelles. Il est cependant admis qu'elles soient unidirectionnelles lorsque le *Serveur* n'est pas capable de présenter le sens inverse pointant du *Type d'Événement* vers chaque *Type de Variable* prenant en charge le *Type d'Événement*.

Les *Références "Génère un Événement"* sont facultatives, ce qui signifie que les *Variables* peuvent générer des *Événements* d'un *Type d'Événement* qui n'est pas présenté par ce *Type de Variable*.

Les *Types de Variables* peuvent utiliser des *Références* supplémentaires pour définir des relations avec d'autres *Nœuds*. Aucune contrainte n'est appliquée aux types de *Références*

utilisés ou aux *Classes de Nœuds* correspondant aux *Nœuds* pouvant être référencés. Cependant, le *Type de Référence* peut définir des restrictions qui excluent son utilisation pour des *Types de Variables*. Des *Types de Références* normalisés sont décrits à l'Article 7.

Tous les *Nœuds* référencés avec des *Références hiérarchiques* doivent avoir des *Noms de Navigation* uniques dans le contexte du *Type de Variable* (voir 4.5).

### 5.6.6 Création du côté client de Variables d'un Type de Variable donné

Les *Variables* sont toujours fondées sur un *Type de Variable*; ce qui signifie qu'elles ont une *Référence "Dispose d'une Définition de Type"* pointant vers son *Type de Variable*.

Les *Clients* peuvent créer des *Variables* en utilisant le *Service "Ajouter des Nœuds"* défini dans l'IEC 62541-4. Le *Service* exige que le *Nœud de Définition de Type* de la *Variable* soit spécifié. Une *Variable* créée par le *Service "Ajouter des Nœuds"* contient tous les composants définis par son *Type de Variable* en fonction des *Règles de Modélisation* spécifiées pour les composants. Le *Serveur* peut cependant ajouter des composants supplémentaires et des *Références* à la *Variable* et à ses composants qui ne sont pas définis par le *Type de Variable*. Ce comportement dépend du *Serveur*. Le *Type de Variable* spécifie uniquement l'ensemble minimal de composants qui doit exister pour chaque *Variable* d'un *Type de Variable* donné.

### 5.7 Classe de Nœud "Méthodes"

Les *Méthodes* définissent des fonctions invocables. Les *Méthodes* sont invoquées par le *Service "Appel"* défini dans l'IEC 62541-4. Les invocations des *Méthodes* ne sont pas représentées dans l'*Espace d'Adressage*. Les invocations de *Méthodes* vont toujours jusqu'à achèvement et renvoient toujours des réponses une fois terminées. Les *Méthodes* sont définies en utilisant la *Classe de Nœud "Méthodes"*, spécifiée dans le Tableau 10.

Tableau 10 – Classe de Nœud "Méthodes"

| Dénomination              | Usage | Type de Données | Description                                                                                                                                                                                                                                                                                                                                                                        |
|---------------------------|-------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Attributs</b>          |       |                 |                                                                                                                                                                                                                                                                                                                                                                                    |
| Base NodeClass Attributes | M     | --              | Hérité de la <i>Classe de Nœud de Base</i> . Voir 5.2.                                                                                                                                                                                                                                                                                                                             |
| Executable                | M     | Booléen         | L' <i>Attribut "Exécutable"</i> indique si la <i>Méthode</i> est actuellement exécutable ("Faux" signifie non exécutable, "Vrai" signifie exécutable).<br>L' <i>Attribut "Exécutable"</i> ne tient compte d'aucun droit d'accès utilisateur, ce qui signifie que même si la <i>Méthode</i> est exécutable, elle peut être limitée à certains utilisateurs / groupe d'utilisateurs. |
| UserExecutable            | M     | Booléen         | L' <i>Attribut "Exécutable par l'utilisateur"</i> indique si la <i>Méthode</i> est actuellement exécutable en tenant compte des droits d'accès de l'utilisateur ("Faux" signifie non exécutable, "Vrai" signifie exécutable).                                                                                                                                                      |
| <b>Références</b>         |       |                 |                                                                                                                                                                                                                                                                                                                                                                                    |
| HasProperty               | 0..*  |                 | Les <i>Références "Dispose d'une Propriété"</i> identifient les <i>Propriétés</i> de la <i>Méthode</i> .                                                                                                                                                                                                                                                                           |
| HasModellingRule          | 0..1  |                 | Les <i>Méthodes</i> peuvent pointer vers, au plus, un <i>Objet Règle de Modélisation</i> en utilisant une <i>Référence "Dispose d'une Règle de Modélisation"</i> (pour plus de détails concernant les <i>Règles de Modélisation</i> , voir 6.4.4).                                                                                                                                 |
| GeneratesEvent            | 0..*  |                 | Les <i>Références "Génère un Événement"</i> identifient les types d' <i>Événements</i> qui seront générés à chaque fois que la <i>Méthode</i> est invoquée.                                                                                                                                                                                                                        |
| AlwaysGenerates-Event     | 0..*  |                 | Les <i>Références "Génère Toujours un Événement"</i> identifient les types d' <i>Événements</i> qui seront générés chaque fois que la <i>Méthode</i> est invoquée.                                                                                                                                                                                                                 |
| <other References>        | 0..*  |                 | Les <i>Méthodes</i> peuvent contenir d'autres <i>Références</i> .                                                                                                                                                                                                                                                                                                                  |

| Dénomination                  | Usage | Type de Données | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------------------|-------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Propriétés normalisées</b> |       |                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| NodeVersion                   | O     | Chaîne          | La <i>Propriété</i> "NodeVersion" est utilisée pour indiquer la version d'un <i>Nœud</i> .<br>La <i>Propriété</i> «NodeVersion» est mise à jour à chaque fois qu'une <i>Référence</i> est ajoutée ou retirée du <i>Nœud</i> auquel la <i>Propriété</i> appartient. Les modifications de la valeur de l' <i>Attribut</i> n'entraînent pas une modification de la <i>Version de Nœud</i> . Les <i>Clients</i> peuvent consulter la <i>Propriété</i> «NodeVersion» ou s'y abonner pour établir le moment où la structure d'un <i>Nœud</i> a changé. |
| InputArguments                | O     | Argument[]      | La <i>Propriété</i> "Arguments d'entrée" est utilisée pour spécifier les arguments qui doivent être utilisés par un <i>Cient</i> lorsqu'il invoque la <i>Méthode</i> .                                                                                                                                                                                                                                                                                                                                                                           |
| OutputArguments               | O     | Argument[]      | La <i>Propriété</i> "Arguments de sortie" spécifie le résultat renvoyé par l'invocation de la <i>Méthode</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                   |

La *Classe de Nœud "Méthodes"* hérite des *Attributs* de base de la *Classe de Nœud de Base* définie en 5.2. La *Classe de Nœud "Méthodes"* ne définit pas d'*Attributs* supplémentaires.

L'*Attribut Executable* (*Exécutable*) indique si une *Méthode* est exécutable, sans tenir compte des droits d'accès de l'utilisateur. Si le *Serveur OPC UA* ne peut obtenir du système sous-jacent les informations de caractère *Exécutable*, il convient de déclarer qu'il est exécutable. Si une *Méthode* est invoquée, il convient alors que le *Serveur* transfère cette requête et renvoie le *Code de Statut* correspondant même si une telle requête est rejetée. Les *Codes de Statut* sont définis dans l'IEC 62541-4.

L'*Attribut UserExecutable* (*Exécutable par l'Utilisateur*) indique si la *Méthode* est exécutable, en tenant compte des droits d'accès de l'utilisateur. Si le *Serveur OPC UA* ne peut pas obtenir du système sous-jacent les informations concernant les droits d'accès de l'utilisateur, il convient d'utiliser la même valeur que celle de l'*Attribut "Exécutable"*. L'*Attribut "Exécutable par l'Utilisateur"* peut être mis sur "Faux", même si l'*Attribut "Exécutable"* est mis sur "Vrai"; il doit cependant être mis sur "Faux" si l'*Attribut "Exécutable"* est mis sur "Faux".

Des *Propriétés* peuvent être définies pour les *Méthodes* en utilisant les *Références "Dispose d'une Propriété"*. Les *Propriétés InputArguments* (*Arguments d'entrée*) et *OutputArguments* (*Arguments de sortie*) spécifient les arguments d'entrée et les arguments de sortie de la *Méthode*. Les deux contiennent une matrice du *Type de Données Argument* comme spécifié en 8.6. Une matrice vide ou une *Propriété* qui n'est pas fournie, indique qu'il n'y a pas d'arguments d'entrée ou d'arguments de sortie pour la *Méthode*. La *Propriété «NodeVersion»* indique la version de la *Méthode*. Il n'y a pas d'autres *Propriétés* définies pour les *Méthodes* dans le présent document. D'autres parties de la série IEC 62541 peuvent définir des *Propriétés* supplémentaires pour les *Méthodes*.

Pour spécifier sa *Règle de Modélisation*, une *Variable* peut utiliser au plus une *Méthode "Dispose d'une Règle de Modélisation"* pointant vers un *Objet "Règle de Modélisation"*. Les *Règles de Modélisation* sont définies en 6.4.4.

Les *Références "Génère un Événement"* indiquent que les *Méthodes* peuvent générer un *Événement* du *Type d'Événement* spécifié ou d'un de ses sous-types pour chaque invocation de la *Méthode*. Un *Serveur* peut générer un *Événement* pour chaque *Type d'Événement* référencé lorsqu'une *Méthode* est invoquée avec succès.

Les *Références AlwaysGeneratesEvent* (*Génère Toujours un Événement*) indiquent que les *Méthodes* généreront un *Événement* du *Type d'Événement* spécifié ou de l'un de ses sous-types pour chaque invocation de *Méthode*. Un *Serveur* doit toujours générer un *Événement* pour chaque *Type d'Événement* référencé lorsqu'une *Méthode* est invoquée avec succès.

Il convient que les *Serveurs* fassent en sorte que les *Références "Génère un Événement"* soient bidirectionnelles. Il est cependant admis qu'elles soient unidirectionnelles lorsque le *Serveur* n'est pas capable de présenter le sens inverse, pointant du *Type d'Événement* vers chaque *Méthode* générant le *Type d'Événement*.

Les *Références "Génère un Événement"* sont facultatives, ce qui signifie que l'invocation d'une *Méthode* peut produire des *Événements* d'un *Type d'Événement* qui n'est pas référencé avec une *Référence "Génère un Événement"* par la *Méthode*.

Les *Méthodes* peuvent utiliser des *Références* supplémentaires pour définir des relations avec d'autres *Nœuds*. Aucune contrainte n'est appliquée aux types de *Références* utilisés ou aux *Classes de Nœuds* correspondant aux *Nœuds* pouvant être référencés. Cependant, le *Type de Référence* peut définir des restrictions qui excluent son utilisation pour des *Méthodes*. Des *Types de Références* normalisés sont décrits à l'Article 7.

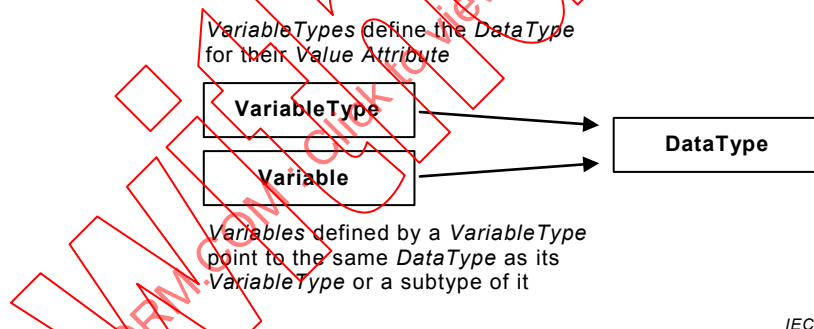
Une *Méthode* doit toujours être le *TargetNode* d'au moins une *Référence "HasComponent"*. Le *SourceNode* de ces *Références "HasComponent"* doit être un *Objet* ou un *Type d'Objet*. Si une *Méthode* est invoquée, alors l'*Identificateur de Nœud* de l'un de ces *Nœuds* doit être placé dans le *Service "Appel"* défini dans l'IEC 62541-4, en tant que paramètre, afin de détecter le contexte d'exécution de la *Méthode*.

Si la *Méthode* est utilisée comme *Déclaration d'Instance* (voir 4.5), tous les *Nœuds* référencés avec des *Références hiérarchiques* dans le sens direct doivent avoir des *Noms de Navigation* uniques dans le contexte de cette *Méthode*.

## 5.8 Types de Données

### 5.8.1 Modèle de Type de Données

Le Modèle de Type de Données est utilisé pour définir des types de données simples et structurés. Les types de données sont utilisés pour décrire la structure de l'*Attribut "Valeur"* de *Variables* et leurs *Types de Variables*. Par conséquent, chaque *Variable* et *Type de Variable* pointe au moyen de son *Attribut "DataType"* (*Type de Données*) vers un *Nœud de la Classe de Nœud "DataType"*, comme illustré en Figure 9.



| Anglais                                                                                                                           | Français                                                                                                                                                               |
|-----------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>VariableTypes</i> define the <i>DataType</i> for their <i>Value Attribute</i>                                                  | Les <i>Types de Variable</i> définissent le <i>Type de Données</i> pour leur <i>Attribut Valeur</i>                                                                    |
| VariableType                                                                                                                      | Type de Variable                                                                                                                                                       |
| Variable                                                                                                                          | Variable                                                                                                                                                               |
| DataType                                                                                                                          | Type de Données                                                                                                                                                        |
| <i>Variables</i> defined by a <i>VariableType</i> point to the same <i>DataType</i> as its <i>VariableType</i> or a subtype of it | Les <i>Variables</i> définies par un <i>Type de Variable</i> pointent vers le même <i>Type de Données</i> que leur <i>Type de Variable</i> ou un sous-type de celui-ci |

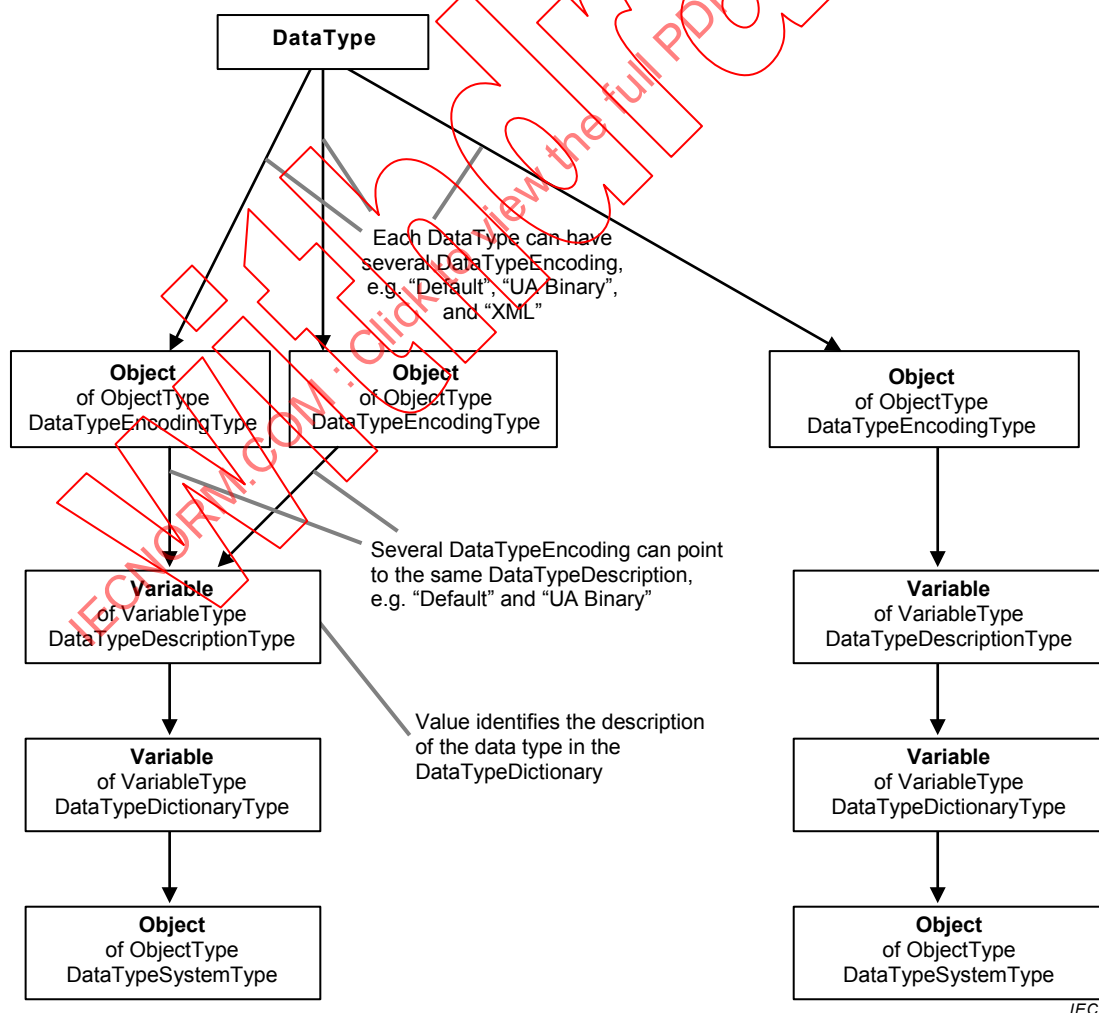
**Figure 9 – Variables, Types de Variables et leurs Types de Données**

Dans de nombreuses situations, l'*Identificateur de Nœud* du *Nœud de Type de Données* – l'*Identificateur de Type de Données* – est bien connu des *Clients* et des *Serveurs*. L'Article 8 définit les *Types de Données* et l'IEC 62541-6 définit leurs *Identificateurs de Type de Données*. Par ailleurs, d'autres organismes peuvent définir des *Types de Données* largement connus dans l'industrie. Les *Identificateurs de Type de Données* largement connus permettent

une généralisation sur les *Serveurs* OPC UA et les *Clients* peuvent de ce fait interpréter des valeurs sans avoir à lire la description de type à partir du *Serveur*. Par conséquent, les *Serveurs* peuvent utiliser des *Identificateurs de Types de Données* largement connus sans avoir à représenter les *Nœuds de Types de Données* correspondants dans leurs *Espaces d'Adressage*.

Dans d'autres situations, les *Types de Données* et leurs *Identificateurs de Type de Données* correspondants peuvent être définis par le fournisseur. Il convient que les *Serveurs* tentent de présenter les *Nœuds de Type de Données* et les informations concernant la structure de ces *Types de Données* pour que les *Clients* puissent les lire, même si ces informations peuvent ne pas toujours être disponibles pour le *Serveur*.

La Figure 10 illustre les *Nœuds* utilisés dans l'*Espace d'Adressage* pour décrire la structure d'un *Type de Données*. Le *Type de Données* pointe vers un *Objet* du type *DataTypeEncodingType* ("Type de Codage de Type de Données". Chaque *Type de Données* peut avoir plusieurs *Codages de Type de Données*, par exemple codage "Par Défaut", "Binaire UA" et "XML". Les services définis dans l'IEC 62541-4 permettent aux *Clients* de demander un codage ou de choisir un codage "Par Défaut". Chaque *Codage de Type de Données* est utilisé précisément par un *Type de Données*, ce qui signifie qu'il n'est pas permis que deux *Types de Données* pointent vers le même *Codage de Type de Données*. L'*Objet* "Codage de Type de Données" pointe précisément vers une *Variable* de type "Type de Description de Type de Données". La *Variable* "Description de Type de Données" appartient à une *Variable* "Dictionnaire de Type de Données".



| Anglais  | Français        |
|----------|-----------------|
| DataType | Type de Données |

| Anglais                                                                                            | Français                                                                                                                                   |
|----------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Each DataType can have several DataTypeEncoding, e.g. "default", "UA Binary", and "XML"            | Chaque Type de Données peut avoir plusieurs Codages de Type de Données, par exemple "par défaut", "Binaire UA" et "XML"                    |
| Object                                                                                             | Objet                                                                                                                                      |
| of ObjectType DataTypeEncodingType                                                                 | du Type d'Objet Type de Codage de Type de Données                                                                                          |
| Several DataTypeEncoding can point to the same DataTypeDescription, e.g. "default" and "UA Binary" | Plusieurs Codages de Type de Données peuvent pointer vers la même Description de Type de Données, par exemple "par défaut" et "Binaire UA" |
| Variable                                                                                           | Variable                                                                                                                                   |
| of VariableType DataTypeDescriptionType                                                            | du Type de Description de Type de Données                                                                                                  |
| Value identifies the description of the data type in the DataTypeDictionary                        | La valeur identifie la description du type de données dans le Dictionnaire de Type de Données                                              |
| of VariableType DataTypeDictionaryType                                                             | du Type de Variable Type de Dictionnaire de Type de Données                                                                                |
| of ObjectType DataTypeSystemType                                                                   | du Type d'Objet Type de Système de Type de données                                                                                         |

Figure 10 – Modèle de Type de Données

Sachant que l'*Identificateur de Nœud* du *Codage de Type de Données* est utilisé dans certaines correspondances pour identifier le *Type de Données* et son codage, comme défini dans l'IEC 62541-6, ces *Identificateurs de Nœuds* peuvent également être bien connus lorsqu'ils correspondent à des *Identificateurs de Type de Données* bien connus.

Le *Dictionnaire de Type de Données* décrit un jeu de *Types de Données* de manière suffisamment détaillée pour que les *Clients* puissent analyser/interpréter les *valeurs* de *Variables* qu'ils reçoivent et composer les *valeurs* qu'ils envoient. Le *Dictionnaire de Type de Données* est représenté comme une *Variable* de type "*Type de Dictionnaire de Type de Données*" dans l'*Espace d'Adressage*; la description relative aux *Types de Données* est contenue dans son *Attribut "Valeur"*. Tous les *Types de Données* contenant, présentés dans l'*Espace d'Adressage*, sont représentés comme des *Variables* du type "*Type de Description de Type de Données*". La *Valeur* de ces *Variables* donne la description d'un *Type de Données* dans l'*Attribut "Valeur"* du *Dictionnaire de Type de Données*.

Le *Type de Données* d'une *Variable "Dictionnaire de Type de Données"* est toujours une Chaîne d'Octets. Le format et les conventions utilisés pour définir les *Types de Données* dans cette Chaîne d'Octets sont définis par des *Systèmes de Type de Données*. Les *Systèmes de Type de Données* sont identifiés par des *Identificateurs de Nœuds*. Ils sont représentés dans l'*Espace d'Adressage* comme des *Objets* du Type d'Objet "*Type de Système de Type de Données*". Chaque *Variable* représentant un *Dictionnaire de Type de Données* référence un Objet "*Système de Type de Données*" pour identifier son *Système de Type de Données*.

Un Client doit reconnaître le *Système de Type de Données* pour analyser d'éventuelles informations de description de type. Les *Clients* OPC UA qui ne reconnaissent pas un *Système de Type de Données* sont incapables d'interpréter ses descriptions de type, et par conséquent, les *valeurs* qu'elles décrivent. Dans ce cas, les *Clients* interprètent ces *valeurs* comme une Chaîne d'Octets opaque.

Les schémas XML OPC binaire et W3C sont des exemples de *Systèmes de Type de Données*. Le *Système de Type de Données* OPC binaire est défini à l'Annexe C. Le système OPC Binaire utilise le langage XML pour décrire des *valeurs* de données binaires. Le Schéma XML du W3C est spécifié dans la Partie 1 du Schéma XML et dans la Partie 2 du Schéma XML.

### 5.8.2 Règles de codage pour différentes sortes de Types de Données

Les différentes sortes de *Types de Données* sont traitées de manière différente en termes de codage et selon que ce codage est ou non représenté dans l'*Espace d'Adressage*.

Les *Types de Données Intégrés* sont un jeu fixe de *Types de Données* (voir l'IEC 62541-6 pour une liste complète des *Types de Données Intégrés*). Ces types n'ont pas de codages visibles dans l'*Espace d'Adressage* car il convient que le codage soit connu de l'ensemble des produits OPC UA. Des exemples de *Types de Données Intégrés* sont *Int32* (voir 8.26) et *Double* (voir 8.12).

Les *Types de Données Simples* sont des sous-types de *Types de Données Intégrés*. Ils sont traités à la volée comme les *Types de Données Intégrés*, ce qui signifie qu'il n'est pas possible, à l'utilisation, de les distinguer de leurs supertypes *Intégrés*. Etant traités, en termes de codage, comme des *Types de Données Intégrés*, leurs codages ne peuvent pas être définis dans l'*Espace d'Adressage*. Les *Clients* peuvent lire l'*Attribut "DataType"* (*Type de Données*) d'une *Variable* ou d'un *Type de Variable* pour identifier le *Type de Données Simple* de l'*Attribut "Valeur"*. Un exemple de *Type de Données Simple* est la *Durée*. Ce type de données est traité à la volée comme un *Double* mais le *Client* peut lire l'*Attribut "DataType"* et ainsi interpréter la valeur, comme définie par la *Durée* (voir 8.13).

Les *Types de Données Structurés* sont des *Types de Données* qui représentent des données structurées et ne sont pas définis comme des *Types de Données Intégrés*. Les *Types de Données Structurés* héritent directement ou indirectement de la *Structure du Type de Données* défini en 8.33. Les *Types de Données Structurés* peuvent avoir plusieurs codages et ces derniers sont présentés dans l'*Espace d'Adressage*. La manière dont le codage des *Types de Données Structurés* est traité à la volée est définie dans l'IEC 62541-6. Le codage de *Type de Données Structuré* est transmis avec chaque valeur et ainsi les *Clients* sont informés du *Type de Données* sans avoir à lire l'*Attribut "DataType"*. Le codage est à transmettre, de façon à ce que le *Client* soit en mesure d'interpréter les données. Un exemple de *Type de Données Structuré* est l'*Argument* (voir 8.6).

Les *Types de Données "Énumération"* sont des *Types de Données* qui représentent des jeux discrets de valeurs définies. Les *Énumérations* sont toujours codées en *Int32* à la volée, comme défini dans l'IEC 62541-6. Les *Types de Données Énumération* héritent directement ou indirectement du *Type de Données Énumération* défini en 8.14. Les *Énumérations* n'ont pas de codages présentés dans l'*Espace d'Adressage*. Pour exposer la représentation interprétable par l'utilisateur d'une valeur énumérée, le *Nœud de Type de Données* peut disposer d'une *Propriété "Chaîne d'Énumération"* contenant une matrice de "*Texte Localisé*". La représentation Entier de la valeur d'énumération pointe vers une position à l'intérieur de cette matrice. La *Propriété "Valeurs d'Énumération"* peut être utilisée au lieu des *Chaînes d'Énumération* pour prendre en charge la représentation Entier des énumérations qui ne sont pas basées sur zéro ou qui présentent des espaces. Elle contient une matrice d'un *Type de Données Structuré* contenant la représentation Entier ainsi que la représentation interprétable par l'utilisateur. La *Classe de Nœud* qui est définie en 8.30 est un exemple de *Type de Données* d'énumération contenant une liste clairsemée d'Entiers.

Outre les *Types de Données* décrit ci-dessus, les *Types de Données* abstraits sont également pris en charge; ces derniers n'ont aucun codage et ne peuvent pas être échangés à la volée. Les *Variables* et *Types de Variables* utilisent des *Types de Données* abstraits pour indiquer que leur *Valeur* peut être n'importe lequel des sous-types du *Type de Données* abstrait. Un exemple de *Type de Données* abstrait est l'*Entier* qui est défini en 8.24.

### 5.8.3 Classe de nœud "DataType" ("*Types de Données*")

La *Classe de nœud "Datatype"* décrit la syntaxe d'une *Variable de Valeur*. Les *Types de Données* peuvent être simples ou complexes, en fonction du *Système de Type de Données*. Les *Types de Données* sont définis en utilisant la *Classe de nœud "Datatype"*, spécifiée dans le Tableau 11.

Tableau 11 – Classe de nœud "DataType"

| Dénomination                  | Usage | Type de Données  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------------------|-------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Attributs</b>              |       |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Base NodeClass Attributes     | M     | --               | Hérité de la <i>Classe de Nœud de Base</i> . Voir 5.2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| IsAbstract                    | M     | Booléen          | Un <i>Attribut</i> booléen prenant les valeurs suivantes:<br>VRAI il s'agit d'un <i>Type de Données</i> abstrait.<br>FAUX il s'agit d'un <i>Type de Données</i> qui n'est pas abstrait.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Références</b>             |       |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| HasProperty                   | 0..*  |                  | Les <i>Références "Dispose d'une Propriété"</i> identifient les <i>Propriétés</i> du <i>Type de Données</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| HasSubtype                    | 0..*  |                  | Les <i>Références "Dispose d'un sous-type"</i> peuvent être utilisées pour couvrir une hiérarchie de type de données.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| HasEncoding                   | 0..*  |                  | Les <i>Références "Dispose d'un Codage"</i> indiquent les codages du <i>Type de Données</i> représentés comme des <i>Objets</i> de type <i>"Type de Codage de Type de Données"</i> .<br>Seuls des <i>Types de Données Structurés</i> concrets peuvent utiliser les <i>Références "Dispose d'un Codage"</i> . Il n'est pas admis que des <i>Types de Données Abstraits</i> , <i>Intégrés</i> , <i>Énumération</i> , et <i>Simple</i> soient le <i>SourceNode</i> d'une <i>Référence "Dispose d'un Codage"</i> .<br>Chaque <i>Type de Données Structuré</i> concret doit pointer vers au moins un <i>Objet "Codage de Type de Données"</i> , avec le <i>Nom de Navigation</i> "Binaire par Défaut" ou "XML par Défaut" ayant l' <i>Indice de l'espace de Nom</i> 0. Le <i>Nom de Navigation</i> des <i>Objets "Codage de Type de Données"</i> doit être unique dans le contexte d'un <i>Type de Données</i> , ce qui signifie qu'un <i>Type de Données</i> ne doit pas pointer vers deux <i>Codages de Type de Données</i> ayant le même <i>Nom de Navigation</i> . |
| <b>Propriétés normalisées</b> |       |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| NodeVersion                   | O     | Chaîne           | La <i>Propriété "NodeVersion"</i> est utilisée pour indiquer la version d'un <i>Nœud</i> .<br>La <i>Propriété "NodeVersion"</i> est mise à jour à chaque fois qu'une <i>Référence</i> est ajoutée ou retirée du <i>Nœud</i> auquel la <i>Propriété</i> appartient. Les modifications de la valeur de l' <i>Attribut</i> n'entraînent pas une modification de la <i>Version de Nœud</i> . Les <i>Clients</i> peuvent consulter la <i>Propriété «NodeVersion»</i> ou s'y abonner pour établir le moment où la structure d'un <i>Nœud</i> a changé.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| EnumStrings                   | O     | Texte localisé[] | La <i>Propriété "Chaîne d'Énumération"</i> s'applique uniquement aux <i>Types de Données "Énumération"</i> . Elle ne doit pas être appliquée à d'autres <i>Types de Données</i> . Si la <i>Propriété "Valeur d'Énumération"</i> est fournie, la <i>Propriété "Chaîne d'Énumération"</i> ne doit pas être fournie.<br>Chaque entrée de la matrice de <i>Texte Localisé</i> dans cette <i>Propriété</i> constitue la représentation interprétable par l'utilisateur d'une valeur énumérée. La représentation Entier de la valeur d'énumération pointe vers une position de la matrice.                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| EnumValues                    | O     | EnumValueType [] | La <i>Propriété "Valeur d'Énumération"</i> s'applique uniquement aux <i>Types de Données "Énumération"</i> . Elle ne doit pas être appliquée à d'autres <i>Types de Données</i> . Si la <i>Propriété "Chaîne d'Énumération"</i> est fournie, la <i>Propriété "Valeur d'Énumération"</i> ne doit pas être fournie.<br>A l'aide de la <i>Propriété "Valeur d'Énumération"</i> , il est possible de représenter des <i>Énumérations</i> avec des entiers qui ne sont pas basées sur zéro ou qui ont des espaces (par exemple, 1, 2, 4, 8, 16).<br>Chaque entrée de la matrice de <i>Type de Valeur d'Énumération</i> dans cette <i>Propriété</i> représente une valeur d'énumération avec sa notation Entier, une représentation interprétable par l'utilisateur et des informations d'aide.                                                                                                                                                                                                                                                                         |

La *Classe de nœud "Datatype"* hérite des *Attributs* de base de la *Classe de Nœud de Base* définie en 5.2. L'*Attribut* supplémentaire *IsAbstract (Est Abstrait)* spécifie si le *Type de Données* est ou non abstrait. Les *Types de Données Abstraits* peuvent être utilisés dans l'*Espace d'Adressage*, ce qui signifie que les *Variables* et *Types de Variables* peuvent pointer, au moyen de leur *Attribut "DataType"*, vers un *Type de Données Abstrait*. Cependant, des

valeurs concrètes ne peuvent jamais être un *Type de Données* Abstrait et doivent toujours être du sous-type concret du *Type de Données* Abstrait.

Les *Références HasProperty* (*Dispose d'une Propriété*) sont utilisées pour identifier les *Propriétés* d'un *Type de Données*. La *Propriété "NodeVersion"* est utilisée pour indiquer la version du *Type de Données*. La Version n'est pas affectée par la *Propriété "Version de Type de Données"* des *"Dictionnaires de Types de Données"* et des *"Descriptions de Types de Données"*. La *Propriété "EnumStrings"* (*Chaînes d'Énumération*) contient des représentations interprétables par l'utilisateur des valeurs d'énumération et est uniquement appliquée aux *Types de Données "Énumération"*. Au lieu de la *Propriété "Chaînes d'Énumération"*, un *Type de Données "Énumération"* peut aussi utiliser la *Propriété "Valeurs d'Énumération"* pour représenter des *Énumérations* avec des valeurs d'entier qui ne sont pas basées sur zéro ou contenant des espaces. Il n'y a pas d'autres *Propriétés* définies pour les *Types de Données* dans la présente norme. D'autres parties de la série IEC 62541 peuvent définir des *Propriétés* supplémentaires pour les *Types de Données*.

Les *Références HasSubtype* (*Dispose d'un Sous-type*) peuvent être utilisées pour présenter une hiérarchie de type de données dans l'*Espace d'Adressage*. Cette hiérarchie doit refléter l'arborescence spécifiée dans le *Dictionnaire de Type de Données*. La sémantique de sous-typage dépend du *Système de Type de Données*. Il n'est pas nécessaire que les *Serveurs* fournissent les *Références "HasSubtype"*, même si leurs *Types de Données* couvrent une hiérarchie de *Types de Données*. Il convient que les *Clients* ne fassent aucune hypothèse relative à une éventuelle sémantique utilisant ces informations autres que celles fournies dans le *Dictionnaire de Type de Données*. Par exemple, il peut s'avérer impossible de relier la valeur d'un type de données à son type de données de base. Certaines restrictions s'appliquent pour le sous-typage des *Types de Données* d'énumération tels que définis en 8.14.

Les *Références HasEncoding* (*Dispose d'un Codage*) pointent du *Type de Données* vers son *Codage de Type de Données*. En suivant cette *Référence*, le *Client* peut explorer le *Dictionnaire de Type de Données* décrivant la structure du *Type de Données* pour le codage utilisé. Chaque *Type de Données Structuré* concret peut pointer vers plusieurs *Codages de Types de Données*, mais chaque *Codage de Type de Données* doit appartenir à un *Type de Données*; en d'autres termes, il n'est pas admis que deux *Nœuds de Type de Données* pointent vers le même *Objet "Codage de Type de Données"* au moyen de *Références "Dispose d'un Codage"*.

Un *Type de Données* Abstrait n'est pas le *SourceNode* d'une *Référence "Dispose d'un Codage"*. Le *Codage de Type de Données* d'un *Type de Données* Abstrait est fourni par ses sous-types concrets.

Les *Nœuds de Type de Données* ne doivent pas être le *SourceNode* d'autres *Types de Références*. Ils peuvent cependant être le *TargetNode* d'autres *Références*.

#### **5.8.4 Dictionnaire de Type de Données, Description de Type de Données, Codage de Type de Données et Système de Type de Données**

Un *Dictionnaire de Type de Données* est une entité qui contient un ensemble de descriptions de type, par exemple un schéma XML. Les *Dictionnaires de Types de Données* sont définis comme des *Variables* du *Type de Variable "Type de Dictionnaire de Type de Données"*.

Un *Système de Type de Données* spécifie le format et les conventions utilisés pour définir les *Types de Données* dans des *Dictionnaires de Types de Données*. Les *Systèmes de Type de Données* sont définis comme des *Objets* du *Type d'Objet "Type de Système de Type de Données"*.

Le *Type de Référence* utilisé pour relier les *Objets* du *Type d'Objet "Type de Système de Type de Données"* aux *Variables* du *Type de Variable "Type de Dictionnaire de Type de Données"* est le *Type de Référence "Dispose d'un Composant"*. Ainsi, la *Variable* est toujours

le *TargetNode* d'une *Référence "Dispose d'un Composant"*; il s'agit là d'une exigence pour les *Variables*. Cependant, pour les *Dictionnaires de Types de Données*, le *Serveur* doit toujours fournir la *Référence* inverse, sachant qu'il est nécessaire de connaître le *Système de Type de Données* lors du traitement du *Dictionnaire de Type de Données*.

Un exemple de *Dictionnaire de Type de Données* est un document XML contenant un schéma XML. Dans ce cas, le *Système de Type de Données* est le Schéma XML du W3C et les déclarations d'éléments de niveau supérieur dans le document schéma sont des descriptions de type de données. Chacune de ces descriptions est définie dans différentes versions d'un schéma XML en utilisant le même espace de nom cible XML. L'espace de nom cible est utilisé comme le composant espace de nom de l'*Identificateur de Type de Données* dans l'*Espace d'Adressage* du *Serveur*. Étant donné que le même espace de nom cible peut être utilisé dans d'autres schémas XML, les *Clients* doivent savoir que deux *Identificateurs de Type de Données* ayant le même espace de nom ne sont pas nécessairement définis dans le même *Dictionnaire de Type de Données*.

Une modification d'une description de type peut entraîner d'autres modifications mais il est plus probable que des modifications du dictionnaire résultent de la somme ou de la suppression de descriptions de type. Ceci comprend les modifications apportées à un moment où le *Serveur* est hors ligne, de sorte que la nouvelle version soit disponible lorsque le *Serveur* redémarre. Les *Clients* peuvent s'abonner à la *Propriété "Version de Type de Données"* pour déterminer une éventuelle modification au *Dictionnaire de Type de Données* depuis la dernière consultation.

Le *Serveur* peut, sans que cela ne soit exigé, mettre le contenu du *Dictionnaire de Type de Données* à la disposition des *Clients* par le biais de l'*Attribut "Valeur"*. Il convient dans ce cas que les *Clients* considèrent que le contenu du *Dictionnaire de Type de Données* est relativement important et qu'ils seront confrontés à des problèmes de performance s'ils consultent automatiquement le contenu du *Dictionnaire de Type de Données* à chaque fois qu'ils rencontrent une instance d'un *Type de Données* spécifique. Il convient que le *Client* utilise la *Propriété Version de Type de Données* pour établir si la copie contenue dans la mémoire cache locale est encore valide. Si le *Client* détecte une modification dans la *Version de Type de Données*, il doit consulter à nouveau le *Dictionnaire de Type de Données*. Ceci implique que la *Version de Type de Données* doit être mise à jour par un *Serveur*, même après redémarrage, étant donné que les *Clients* peuvent avoir gardé une copie persistante dans la mémoire cache locale.

L'*Attribut "Valeur"* du *Dictionnaire de Type de Données* qui contient les descriptions de type est une Chaîne d'Octets dont le formatage est défini par le *Système de Type de Données*. Pour le *Système de Type de Données* "schéma XML", la Chaîne d'Octets contient un document de schéma XML valide. Pour le *Système de Type de Données* "OPC Binaire", la Chaîne d'Octets contient une chaîne qui est un document XML valide. Le *Serveur* doit assurer que toute modification au contenu de la Chaîne d'Octets correspond à une modification de la *Propriété "Version de Type de Données"*. En d'autres termes, le *Client* peut utiliser en toute sécurité une copie en cache du *Dictionnaire de Type de Données*, tant que la *Version de Type de Données* demeure identique.

Les *Dictionnaires de Types de Données* sont des *Variables* complexes qui présentent leurs *Descriptions de Types de Données* comme des *Variables* en utilisant des *Références "Dispose d'un Composant"*. Une *Description de Type de Données* fournit les informations nécessaires à l'obtention de la description formelle d'un *Type de Données* dans le *Dictionnaire de Type de Données*. La *Valeur* d'une *Description de Type de Données* dépend du *Système de Type de Données* du *Dictionnaire de Type de Données*. Lorsque des dictionnaires "OPC Binaires" sont utilisés, la *Valeur* doit être le nom de la *Description de Type*. Lorsque des dictionnaires à "schéma XML" sont utilisés, la *Valeur* doit être une expression Xpath (Voir XPATH (CHEMIN X)) pointant vers un élément XML dans le document du schéma.

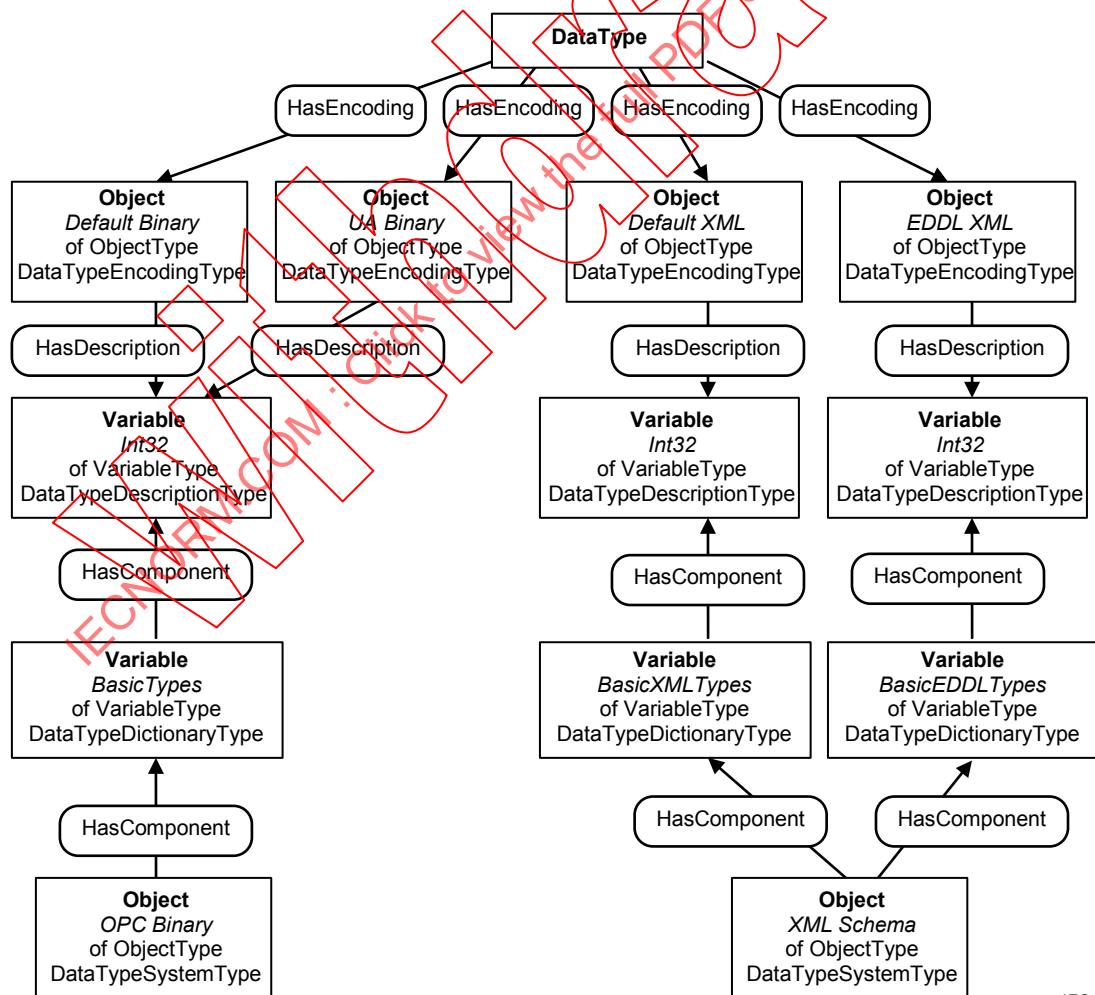
Comme les *Dictionnaires de Types de Données*, chaque *Description de Type de Données* fournit la *Propriété Version de Type de Données* indiquant si la description de type du *Type*

de Données a changé. Les modifications de la *Version de Type de Données* peuvent avoir un effet sur le fonctionnement des *Abonnements*. Si la *Version de Type de Données* est modifiée en une *Variable* qui fait l'objet d'une surveillance pour un *Abonnement* donné et qui utilise la *Description de Type de Données*, la prochaine *Notification* de modification des données envoyée pour ce qui concerne la *Variable*, contiendra un statut qui indique la modification apportée à la *Description de Type de Données*.

Les *Objets "Codage de Type de Données"* des *Types de Données* référencent leurs *Descriptions de Types de Données* des *Dictionnaires de Types de Données* en utilisant des *Références "Dispose d'une Description"*. Cependant, il n'est pas exigé que les *Serveurs* fournissent des *Références* inverses qui relient en retour les *Descriptions de Types de Données* aux *Objets "Codage de Type de Données"*. Si un *Nœud de Type de Données* est présenté dans l'*Espace d'Adressage*, il doit fournir son *Codage de Type de Données* et si un *Dictionnaire de Type de Données* est présenté, il convient alors qu'il présente l'ensemble de ses *Descriptions de Types de Données*. Ces deux *Références* doivent être bidirectionnelles.

Les *Types de Variables "Type de Dictionnaire de Type de Données"* et "*Type de Description de Type de Données*" ainsi que les *Types d'Objets "Type de Système de Type de Données"* et "*Type de Codage de Type de Données*" sont définis de manière formelle dans l'IEC 62541-5.

La Figure 11 donne un exemple de modélisation de *Types de Données* dans l'*Espace d'Adressage*.



IEC

| Anglais  | Français        |
|----------|-----------------|
| DataType | Type de Données |

| Anglais                                 | Français                                                    |
|-----------------------------------------|-------------------------------------------------------------|
| HasEncoding                             | Dispose d'un Codage                                         |
| Object                                  | Objet                                                       |
| <i>Default Binary</i>                   | <i>Binaire par défaut</i>                                   |
| of ObjectType DataTypeEncodingType      | du Type d'Objet Type de Codage de Type de données           |
| <i>UA Binary</i>                        | <i>Binaire UA</i>                                           |
| <i>Default XML</i>                      | <i>XML par défaut</i>                                       |
| <i>EDDL XML</i>                         | <i>XML EDDL</i>                                             |
| HasDescription                          | Dispose d'une description                                   |
| Variable                                | Variable                                                    |
| <i>Int32</i>                            | <i>Int32 (Entier de 32 bits)</i>                            |
| of VariableType DataTypeDescriptionType | du Type de Variable Type de Description de Type de Données  |
| HasComponent                            | Dispose d'un composant                                      |
| <i>BasicTypes</i>                       | <i>Types de Base</i>                                        |
| <i>BasicXMLTypes</i>                    | <i>Types XML de Base</i>                                    |
| <i>BasicEDDLTypes</i>                   | <i>Types EDDL de Base</i>                                   |
| of VariableType DataTypeDictionaryType  | du Type de Variable Type de Dictionnaire de Type de Données |
| <i>OPC Binary</i>                       | <i>Binaire OPC</i>                                          |
| <i>XML Schema</i>                       | <i>Schéma XML</i>                                           |
| of ObjectType DataTypeSystemType        | du Type d'Objet Type de Système de Type de Données          |

**Figure 11 – Exemple de modélisation de Type de Données**

Dans certains scénarios, un *Serveur* OPC UA peut avoir des ressources limitées et il s'avère peu pratique de présenter des *Dictionnaires de Types de Données* de taille importante. Dans de tels scénarios, le *Serveur* peut fournir un accès à des descriptions de Types de Données particuliers, même si l'ensemble du dictionnaire n'est pas consultable. De ce fait, la présente norme définit une *Propriété* pour la *Description de Type de Données* appelée *Fragment de Dictionnaire* (voir 5.6.2). Cette *Propriété* est une Chaîne d'Octets qui contient un sous-ensemble de *Dictionnaire de Type de Données* qui décrit le format de *Type de Données* associé à la *Description de Type de Données*. Ainsi, le *Serveur* découpe le *Dictionnaire de Type de Données* de grande taille en plusieurs petites parties et les *Clients* peuvent y accéder sans affecter la performance globale du système.

Cependant, il convient que le *Serveur* fournisse l'ensemble du *Dictionnaire de Type de Données* en une seule fois si cela est possible. Il est en général plus efficace de lire l'ensemble du *Dictionnaire de Type de Données* en une seule fois plutôt que de consulter ses différentes parties.

## 5.9 Synthèse des Attributs de NodeClasses (Classes de Nœuds)

Le Tableau 12 récapitule tous les *Attributs* définis dans le présent document et précise les *Classes de Nœuds* qui les utilisent, que ce soit de manière facultative (O pour Optional) ou obligatoire (M pour Mandatory).

**Tableau 12 – Vue d'ensemble des Attributs**

| Attribut                | Variable | Type de Variable | Objet | Type d'Objet | Type de Référence | Type de Données | Méthode | Vue |
|-------------------------|----------|------------------|-------|--------------|-------------------|-----------------|---------|-----|
| AccessLevel             | M        |                  |       |              |                   |                 |         |     |
| ArrayDimensions         | O        | O                |       |              |                   |                 |         |     |
| BrowseName              | M        | M                | M     | M            | M                 | M               | M       | M   |
| ContainsNoLoops         |          |                  |       |              |                   |                 |         | M   |
| DataType                | M        | M                |       |              |                   |                 |         |     |
| Description             | O        | O                | O     | O            | O                 | O               | O       | O   |
| DisplayName             | M        | M                | M     | M            | M                 | M               | M       | M   |
| EventNotifier           |          |                  | M     |              |                   |                 |         | M   |
| Executable              |          |                  |       |              |                   |                 | M       |     |
| Historizing             | M        |                  |       |              |                   |                 |         |     |
| InverseName             |          |                  |       |              | O                 |                 |         |     |
| IsAbstract              |          | M                |       | M            | M                 | M               |         |     |
| MinimumSamplingInterval | O        |                  |       |              |                   |                 |         |     |
| NodeClass               | M        | M                | M     | M            | M                 | M               | M       | M   |
| NodeId                  | M        | M                | M     | M            | M                 | M               | M       | M   |
| Symmetric               |          |                  |       |              | M                 |                 |         |     |
| UserAccessLevel         | M        |                  |       |              |                   |                 |         |     |
| UserExecutable          |          |                  |       |              |                   |                 | M       |     |
| UserWriteMask           | O        | O                | O     | O            | O                 | O               | O       | O   |
| Value                   | M        | O                |       |              |                   |                 |         |     |
| ValueRank               | M        | M                |       |              |                   |                 |         |     |
| WriteMask               | O        | O                | O     | O            | O                 | O               | O       | O   |

## 6 Modèle Type de Types d'Objets et de Types de Variables

### 6.1 Vue d'ensemble

Dans le reste de l'Article 6, le modèle type de *Types d'Objets* et de *Types de Variables* est défini en termes de sous-typage et d'instanciation.

### 6.2 Définitions

#### 6.2.1 Déclaration d'Instance

Une *Déclaration d'Instance* est un *Objet*, une *Variable* ou une *Méthode* qui référence une *Règle de Modélisation* au moyen d'une *Référence "Dispose d'une Règle de Modélisation"* et qui est le *TargetNode* d'une *Référence hiérarchique* à partir d'une ou d'une autre *Déclaration d'Instance*.

#### 6.2.2 Instances sans Règles de Modélisation

S'il n'existe aucune *Règle de Modélisation*, le *Nœud* n'est pas pris en compte pour l'instanciation de type ou pour le sous-typage.

Si un *Nœud* référencé par un *TypeDefinitionNode* (*Nœud de Définition de Type*) ne référence pas une *Règle de Modélisation*, cela signifie que ce *Nœud* appartient uniquement au *TypeDefinitionNode* et non aux instances. Par exemple, un *Nœud de Type d'Objet* peut contenir une *Propriété* qui décrit des scénarios qui peuvent utiliser le type. Cette *Propriété* ne serait pas prise en considération lors de la création d'instances de ce type. Ceci est également vrai pour le sous-typage, ce qui signifie que les sous-types de la définition de type n'hériteraient pas du *Nœud* référencé.

### 6.2.3 Hiérarchie de Déclaration d'Instance

La *Hiérarchie de Déclaration d'Instance* d'un *Nœud de Définition de Type* contient le *Nœud de Définition de Type* ainsi que toutes les *Déclarations d'Instance* directement ou indirectement référencées à partir d'un *Nœud de Définition de Type* au moyen des *Références hiérarchiques* dans le sens direct.

### 6.2.4 Nœud de Déclaration d'Instance similaire

Un *Nœud* similaire d'une *Déclaration d'Instance* est un *Nœud* qui a le même *Nom de Navigation* et la même *Classe de Nœud* que la *Déclaration d'Instance* et, dans le cas de *Variables* et d'*Objets*, le même *Nœud de Définition de Type* ou un sous-type correspondant.

### 6.2.5 Chemin de Navigation (BrowsePath)

Toutes les cibles de *Références hiérarchiques* directes à partir d'un *Nœud de Définition de Type* doivent avoir un *Nom de Navigation* qui est unique au sein du *Nœud de Définition de Type*. Cette même restriction s'applique aux cibles de *Références hiérarchiques* dans le sens direct à partir d'une quelconque *Déclaration d'Instance*. Ceci signifie que toute *Déclaration d'Instance* dans la *Hiérarchie de Déclaration d'Instance* peut être identifiée de manière unique par une séquence de *Noms de Navigation*. Cette séquence de *Noms de Navigation* est appelée *Chemin de Navigation*.

### 6.2.6 Traitement des Attributs de Déclarations d'Instance

Il existe certaines restrictions concernant les *Attributs de Déclarations d'Instance* lorsque la *Déclaration d'Instance* est remplacée ou instanciée. Le *Nom de Navigation* et la *Classe de Nœud* ne doivent jamais changer et toujours demeurer les mêmes que ceux de la *Déclaration d'Instance* initiale.

En outre, les règles définies en 6.2.7 s'appliquent aux *Déclarations d'Instance* de la *Classe de Nœud "Variables"*.

### 6.2.7 Traitement des Attributs de Variables et Types de Variables

Il existe certaines restrictions concernant les *Attributs* d'un *Type de Variable* ou d'une *Variable* utilisée comme *Déclaration d'Instance* pour ce qui concerne le type de données de l'*Attribut "Valeur"*.

Lorsqu'une *Variable* utilisée comme *Déclaration d'Instance* ou qu'un *Type de Variable* est remplacé ou instancié, les règles suivantes s'appliquent:

- a) L'*Attribut "Type de Données"* ne peut être modifié qu'en un nouveau *Type de Données* si le nouveau *Type de Données* est un sous-type du *Type de Données* initialement utilisé.
- b) L'*Attribut Rang de Valeur* peut seulement être plus strict
  - 1) 'Toute valeur' peut être mise pour toute autre valeur;
  - 2) 'Scalaire Ou Unidimensionnelle' peut être mis sur 'Scalaire' ou sur 'Unidimensionnelle';
  - 3) 'Une Ou Plusieurs Dimensions' peut être mis sur un nombre concret de dimensions (valeur > 0);
  - 4) Toutes les autres valeurs de cet *Attribut* doivent demeurer inchangées.
- c) L'*Attribut "Dimensions de Matrice"* peut être ajouté s'il n'a pas été fourni lors de la modification de la valeur 0 d'une entrée de la matrice en une valeur différente. Toutes les autres valeurs de la matrice doivent demeurer inchangées.

### 6.2.8 Identificateurs de Nœuds de Déclarations d'Instance

Les *Déclarations d'Instance* sont identifiées par leur *Chemin de Navigation*. Différents *Serveurs* peuvent utiliser différents *Identificateurs de Nœuds* pour les *Déclarations d'Instance*

des *Nœuds de Définition de Type* communs, à moins que le *Nœud de Définition de Type* définisse déjà un *Identificateur de Nœud* pour la *Déclaration d'Instance*. Tous les *Nœuds de Définition de Type* définis dans l'IEC 62541-5 définissent déjà les *Identificateurs de Nœuds* pour leurs *Déclarations d'Instance* et doivent donc être utilisés dans tous les *Serveurs*.

## 6.3 Sous-typage de Types d'Objets et de Types de Variables

### 6.3.1 Vue d'ensemble

Le *Type de Référence «HasSubtype»* définit des sous-types de types. Le sous-typage ne peut avoir lieu qu'entre *Nœuds* de la même *Classe de Nœud*. Les règles de sous-typage des *Types de Références* sont décrites en 5.3.3.3. Il n'y a aucune définition commune pour le sous-typage des *Types de Données*, comme décrit en 5.8.3. Le reste de 6.3 spécifie les règles de sous-typage pour un héritage simple portant sur les *Types d'Objets* et les *Types de Variables*.

### 6.3.2 Attributs

Les sous-types héritent des valeurs d'*Attribut* des types de parents, sauf l'*Identificateur de Nœud*. Les valeurs d'*Attribut* héritées peuvent être remplacées par le sous-type; il convient de remplacer les valeurs du *Nom de Navigation* et du *Nom d'Affichage*. Des règles particulières s'appliquent à certains *Attributs* de *Types de Variables* comme défini en 6.2.7. Des *Attributs* facultatifs, non prévus par le type parent, peuvent être ajoutés au sous-type.

### 6.3.3 Déclarations d'Instance

#### 6.3.3.1 Vue d'ensemble

Les sous-types héritent intégralement des *Déclarations d'Instance* de type parent héritées.

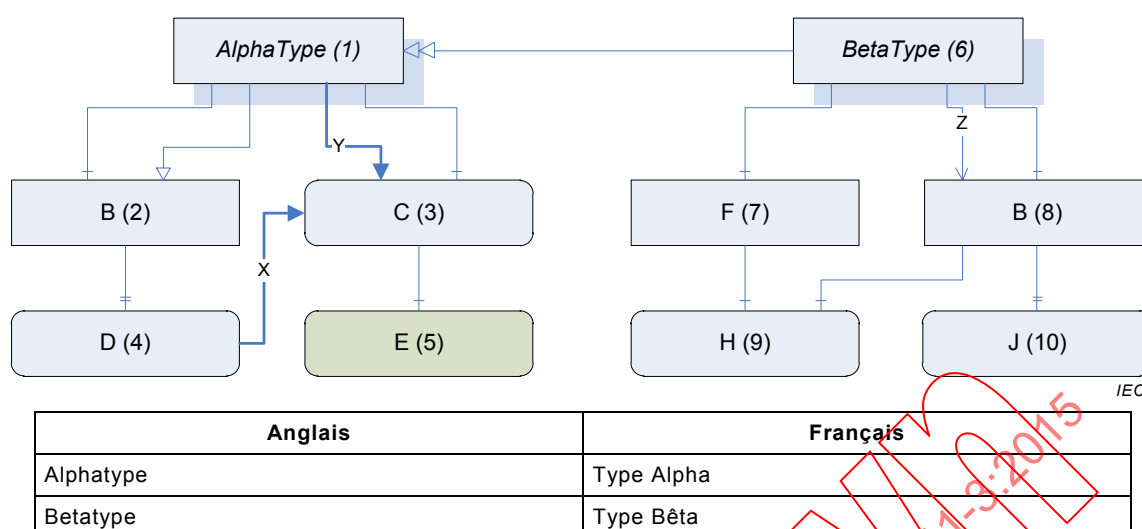
Tant que ces *Déclarations d'Instance* ne sont pas remplacées, elles ne sont pas référencées par le sous-type. Des *Déclarations d'Instance* peuvent être remplacées en ajoutant des *Références*, en modifiant les *Références* portant sur les différents *Nœuds*, en modifiant les *Références* en sous-types du *Type de Référence* initial, en modifiant les valeurs des *Attributs* ou en ajoutant des *Attributs* facultatifs. Pour obtenir l'ensemble des informations concernant un sous-type, les *Déclarations d'Instance* héritées sont à recueillir à partir de tous les types qui peuvent être obtenus en suivant de manière récursive les *Références* inverses "*HasSubtype*" à partir du sous-type. Ce recueil de *Déclarations d'Instance* est appelé la *Hiérarchie de Déclaration d'Instance* intégralement héritée d'un sous-type donné.

Le reste de 6.3.3 définit la méthode de constitution de la *Hiérarchie de Déclaration d'Instance* intégralement héritée et la manière dont les *Déclarations d'Instance* peuvent être remplacées.

#### 6.3.3.2 Hiérarchie de Déclaration d'Instance intégralement héritée

Une instance de *Nœud de Définition de Type* est décrite par la *Hiérarchie de Déclaration d'Instance* intégralement héritée du *Nœud de Définition de Type*. La *Hiérarchie de Déclaration d'Instance* intégralement héritée peut être créée en partant de la *Hiérarchie de Déclaration d'Instance* du *Nœud de Définition de Type* et en fusionnant la *Hiérarchie de Déclaration d'Instance* intégralement héritée de son type parent.

Le processus de fusion des *Hiérarchies de Déclaration d'Instance* est direct. Il est illustré par l'exemple présenté en Figure 12 où est spécifié un *Nœud de Définition de Type* "Type Bêta" qui est un sous-type de "Type Alpha". Dans chaque case, la lettre est le *Nom de Navigation* et le chiffre est l'*Identificateur de Nœud*.



**Figure 12 – Sous-typage des Nœuds de Définition de Type**

Une *Hiérarchie de Déclaration d'Instance* peut être pleinement décrite comme un tableau de Nœuds identifiés par leurs *Chemins de Navigation* avec un tableau de *Références* correspondant. La *Hiérarchie de Déclaration d'Instance* pour le "Type Bêta" est décrite dans le Tableau 13; la moitié supérieure du tableau est le tableau des Nœuds et la moitié inférieure est le tableau des *Références* (les références "*Dispose d'une Règle de Modélisation*" ont été omises du tableau pour plus de clarté; tous les Nœuds, à l'exception des Nœuds 1, 6 et 5 ont des *Règles de Modélisation*). Toutes les *Déclarations d'Instance* de la *Hiérarchie de Déclaration d'Instance* et tous les Nœuds références avec une *Référence* non hiérarchique à partir d'une telle *Déclaration d'Instance* sont ajoutés au tableau. Les *Références Hiérarchiques* aux Nœuds sans *Règle de Modélisation* ne sont pas prises en compte.

**Tableau 13 – Hiérarchie de Déclaration d'Instance pour un Type Bêta**

| Chemin de Navigation | Identificateur de Nœud |
|----------------------|------------------------|
| /                    | 6                      |
| /F                   | 7                      |
| /B                   | 8                      |
| /F/H                 | 9                      |
| /B/J                 | 10                     |
| /B/H                 | 9                      |

| Chemin Source | Type de Référence | Chemin Cible | Identificateur de Nœud Cible        |
|---------------|-------------------|--------------|-------------------------------------|
| /             | HasComponent      | /F           | -                                   |
| /             | HasComponent      | /B           | -                                   |
| /             | Z                 | /B           | -                                   |
| /             | HasTypeDefinition | -            | Type Bêta                           |
| /F            | HasComponent      | /F/H         | -                                   |
| /F            | HasTypeDefinition | -            | Type d'Objet de Base                |
| /B            | HasProperty       | /B/J         | -                                   |
| /B            | HasTypeDefinition | -            | Type d'Objet de Base                |
| /F/H          | HasTypeDefinition | -            | Type de Propriété                   |
| /B/J          | HasTypeDefinition | -            | Type de Propriété                   |
| /B            | HasComponent      | /B/H         | -                                   |
| /B/H          | HasTypeDefinition | -            | Type de Variable de données de Base |

Plusieurs *Chemins de Navigation* vers le même Nœud doivent être traités comme des Nœuds séparés. Une *Instance* peut fournir différents Nœuds pour chaque *Chemin de Navigation*.

La *Hiérarchie de Déclaration d'Instance* intégralement héritée pour un “Type Bêta” peut à présent être constituée par fusion avec la *Hiérarchie de Déclaration d'Instance* pour le “Type Alpha”. Le résultat est présenté dans le Tableau 14; les entrées ajoutées à partir du “Type Alpha” sont grisées.

**Tableau 14 – Hiérarchie de Déclaration d'Instance intégralement héritée pour un Type Bêta**

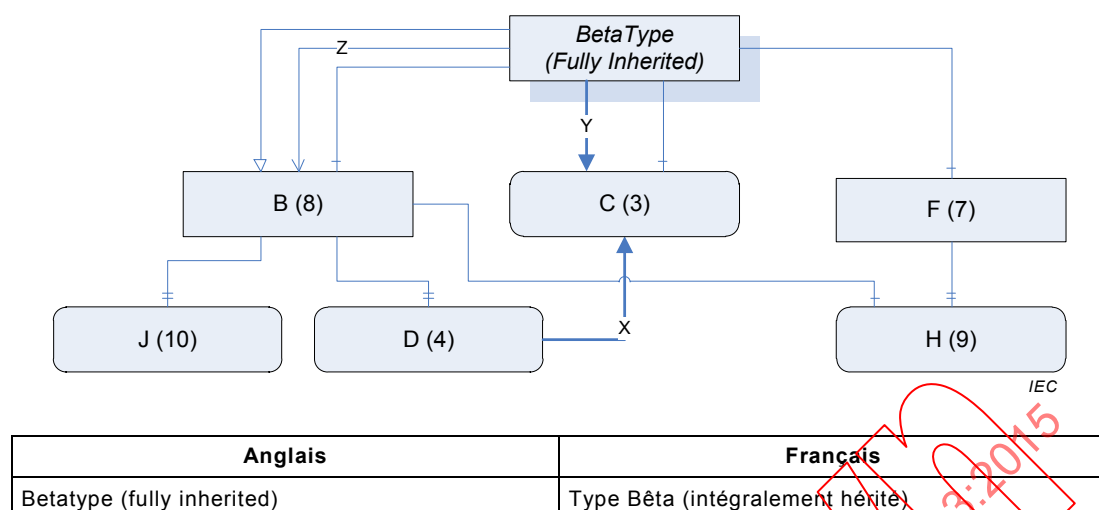
| Chemin de Navigation | Identificateur de Nœud |
|----------------------|------------------------|
| /                    | 6                      |
| /F                   | 7                      |
| /B                   | 8                      |
| /F/H                 | 9                      |
| /B/J                 | 10                     |
| /B/H                 | 9                      |
| /B/D                 | 4                      |
| /C                   | 3                      |

| Chemin Source | Type de Référence | Chemin Cible | Identificateur de Nœud Cible        |
|---------------|-------------------|--------------|-------------------------------------|
| /             | HasComponent      | /F           | -                                   |
| /             | HasComponent      | /B           | -                                   |
| /             | Z                 | /B           | -                                   |
| /             | HasTypeDefinition | -            | Type Bêta                           |
| /F            | HasComponent      | /F/H         | -                                   |
| /F            | HasTypeDefinition | -            | Type d'Objet de Base                |
| /B            | HasProperty       | /B/J         | -                                   |
| /B            | HasTypeDefinition | -            | Type d'Objet de Base                |
| /F/H          | HasTypeDefinition | -            | Type de Propriété                   |
| /B/J          | HasTypeDefinition | -            | Type de Propriété                   |
| /B            | HasComponent      | /B/H         | -                                   |
| /B/H          | HasTypeDefinition | -            | Type de Variable de Données de Base |
| /             | HasNotifier       | /B           | -                                   |
| /B            | HasProperty       | /B/D         | -                                   |
| /             | HasComponent      | /C           | -                                   |
| /             | Y                 | /C           | -                                   |
| /C            | HasTypeDefinition | -            | Type de Variable de Données de Base |
| /B/D          | HasTypeDefinition | -            | Type de Propriété                   |
| /B/D          | X                 | /C           | -                                   |

Le *Chemin de Navigation* “/B” existe déjà dans le tableau, de sorte qu'il n'est pas nécessaire de l'ajouter. Cependant, la référence *HasNotifier* (*Dispose d'un Notificateur*) de “/” vers “/B” n'existe pas et a été ajoutée.

Les *Nœuds* et *Références* définis dans le Tableau 14 peuvent être utilisés pour générer la *Hiérarchie de Déclaration d'Instance* intégralement héritée illustrée en Figure 13. La *Hiérarchie de Déclaration d'Instance* intégralement héritée contient toutes les informations nécessaires à un *Nœud de Définition de Type* pour ce qui concerne sa structure complexe, sans qu'il ne soit nécessaire d'ajouter d'éventuelles informations supplémentaires à partir de ses supertypes.



**Figure 13 – Hiérarchie de Déclaration d'Instance intégralement héritée pour le Type Bêta**

### 6.3.3.3 Remplacement des Déclarations d'Instance

Un sous-type remplace une *Déclaration d'Instance* en spécifiant une *Déclaration d'Instance* ayant le même *Chemin de Navigation*. Une *Déclaration d'Instance* remplacée doit avoir la même *Classe de Nœud* et le même *Nom de Navigation*. Le *Nœud de Définition de Type* de la *Déclaration d'Instance* remplacée doit être le même ou un sous-type du *Nœud de Définition de Type* doit être spécifié dans le supertype.

Lorsqu'une *Déclaration d'Instance* est remplacée, il est nécessaire de fournir les *Références hiérarchiques* qui relient le nouveau *Nœud* au sous-type (les *Références* sont utilisées pour déterminer le *Chemin de Navigation* du *Nœud*).

Il n'est possible de remplacer que des *Déclarations d'Instance* directement référencées à partir du *Nœud de Définition de Type*. Si une *Déclaration d'Instance* indirectement référencée, comme "J" dans la Figure 13, est à remplacer, les *Déclarations d'Instance* directement référencées qui incluent "J", en l'occurrence "B", sont à remplacer en premier lieu et ensuite "J" peut être remplacé dans une seconde phase.

Une *Référence* est remplacée si elle s'inscrit entre deux *Nœuds* remplacés et qu'elle a le même *Type de Référence* qu'une *Référence* définie dans le supertype. La *Référence* spécifiée dans le sous-type peut être un sous-type du *Type de Référence* utilisé dans le type parent.

Toute *Référence non hiérarchique* spécifiée pour la *Déclaration d'Instance* remplacée est traitée comme une nouvelle *Référence*, à moins que le *Type de Référence* ne permette qu'une seule *Référence par Nœud Source*. Dans ce cas, le sous-type peut modifier la cible de la *Référence* mais la nouvelle cible doit avoir la même *Classe de Nœud* et, pour les *Objets* et les *Variables* avoir également le même type ou un sous-type du type spécifié dans le type parent.

Le *Nœud* remplaçant peut spécifier de nouvelles valeurs pour les *Attributs de Nœud* autres que la *Classe de Nœud* ou le *Nom de Navigation*; toutefois les restrictions relatives aux *Attributs* spécifiées en 6.2.6 s'appliquent. Tout *Attribut* fourni par la *Déclaration d'Instance* remplacée est tenu d'être fourni par la *Déclaration d'Instance* remplaçante, il est admis d'ajouter des *Attributs* facultatifs supplémentaires.

La *Règle de Modélisation* de la *Déclaration d'Instance* remplaçante peut être modifiée comme défini en 6.4.4.3.

Chaque *Déclaration d'Instance* remplaçante nécessite ses propres *Références HasModellingRule* (*Dispose d'une Règle de Modélisation*) et *HasTypeDefinition* (*Dispose d'une Définition de Type*), même si elles n'ont pas été modifiées.

Il convient de ne pas remplacer un *Nœud* par un sous-type, à moins que ce sous-type ait à le modifier.

La sémantique de certains *Nœuds de Définition de Type* et de *Types de Références* peut imposer des restrictions supplémentaires concernant les *Nœuds* remplaçants.

## 6.4 Instances de Types d'Objets et de Types de Variables

### 6.4.1 Vue d'ensemble

Toute *Instance* d'un *Nœud de Définition de Type* peut être la racine d'une hiérarchie qui reflète la *Hiérarchie de Déclaration d'Instance* du *Nœud de Définition de Type*. Chaque *Nœud* de la hiérarchie de l'*Instance* a un *Chemin de Navigation* qui peut être le même que celui de l'une des *Déclarations d'Instance* de la hiérarchie du *Nœud de Définition de Type*. La *Déclaration d'Instance* ayant le même *Chemin de Navigation* est la *Déclaration d'Instance* invoquée pour le *Nœud*. Si un *Nœud* a une *Déclaration d'Instance*, il doit avoir le même *Nom de Navigation* et la même *Classe de Nœud* que la *Déclaration d'Instance* et, dans le cas des *Variables* et des *Objets*, le même *Nœud de Définition de Type* ou un sous-type correspondant.

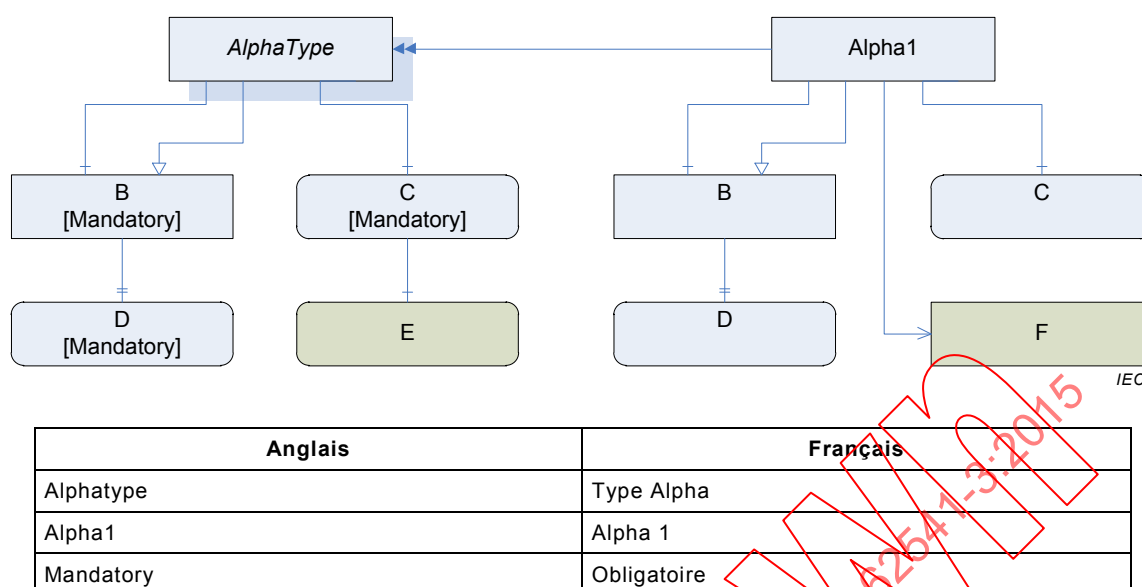
Les *Instances* peuvent référencer plusieurs *Nœuds* ayant le même *Chemin de Navigation*. Les *Clients* qui ont à faire la distinction entre les *Nœuds* fondés sur la *Hiérarchie de Déclaration d'Instance* et les *Nœuds* qui ne sont pas fondés sur la *Hiérarchie de Déclaration d'Instance* peuvent effectuer cette distinction en utilisant le service "Traduire Chemins de Navigation en Identificateurs de Nœuds" (*TranslateBrowsePathsToNodeIds*), défini dans l'IEC 62541-4

### 6.4.2 Création d'une Instance

Les *Instances* héritent des valeurs initiales des *Attributs* qu'ils partagent avec le *Nœud de Définition de Type* à partir duquel ils sont instanciés, à l'exception de la *Classe de Nœud* et de l'*Identificateur de Nœud*.

Lorsqu'un *Serveur* crée une instance de *Nœud de Définition de Type*, il doit créer la même hiérarchie de *Nœuds* sous le nouvel *Objet* ou la nouvelle *Variable*, en fonction de la *Règle de Modélisation* de chaque *Déclaration d'Instance*. Des *Règles de Modélisation* normalisées sont définies en 6.4.4.5. Les *Nœuds* dans la hiérarchie nouvellement créée peuvent être des copies des *Déclarations d'Instance*, de la *Déclaration d'Instance* proprement dite ou d'un autre *Nœud* dans l'*Espace d'Adressage* ayant le même *Nœud de Définition de Type* et le même *Nom de Navigation*. Si de nouvelles copies sont créées, alors les valeurs d'*Attribut* des *Déclarations d'Instance* sont utilisées comme valeurs initiales.

Un exemple simple de *Nœud de Définition de Type* et d'une *Instance* est fourni en Figure 14. Les *Nœuds* référencés par le *Nœud de Définition de Type* sans *Règle de Modélisation* n'apparaissent pas dans l'instance. Les *Instances* peuvent avoir des enfants ayant des *Noms de Navigation* dupliqués; cependant, l'un de ces enfants correspondra à la *Déclaration d'Instance*.



**Figure 14 – Instance et son Nœud de Définition de Type**

Il incombe au *Serveur* de décider quelles *Déclarations d'Instance* apparaissent dans toute instance unique. Dans certains cas, le *Serveur* ne définit pas l'ensemble de l'instance et fournit des références distantes aux *Nœuds*, dans un autre *Serveur*. Les *Règles de Modélisation* décrites en 6.4.4.5 permettent aux *Serveurs* d'indiquer que certains *Nœuds* sont toujours présents; les *Clients* doivent toutefois se préparer à des situations où le *Nœud* existe dans un *Serveur* différent.

Un *Client* peut utiliser les informations de *Nœuds de Définition de Type* pour accéder à des *Nœuds* qui sont dans la hiérarchie de l'instance. Il doit transmettre l'*Identificateur de Nœud* de l'instance et le *Chemin de Navigation* des *Nœuds* enfants, fondés sur le *Nœud de Définition de Type*, au service "*Traduire Chemins de Navigation en Identificateurs de Nœuds*" (*TranslateBrowsePathsToNodeIds*) (voir l'IEC 62541-4). Le *Service* renvoie l'*Identificateur de Nœud* pour chacun des *Nœuds* enfants. Si un *Nœud* existe, le *Nom de Navigation* et la *Classe de Nœud* doivent correspondre à la *Déclaration d'Instance*. Dans le cas d'*Objets* ou de *Variables*, le *Nœud de Définition de Type* doit également correspondre au *Nœud de Définition de Type* initial ou en être un sous-type.

#### 6.4.3 Contraintes applicables à une Instance

Les *Objets* et *Variables* peuvent modifier leurs valeurs d'*Attribut* après création. Des règles particulières s'appliquent pour certains *Attributs* comme défini en 6.2.6.

Des *Références* supplémentaires peuvent être ajoutées aux *Nœuds* et les *Références* peuvent être supprimées tant que les *Règles de Modélisation* définies sur des *Déclarations d'Instance* du *Nœud de Définition de Type* demeurent satisfaites.

Pour les *Variables* et *Objets*, la *Référence HasTypeDefinition* (*Dispose d'une Définition de Type*) doit toujours pointer vers le même *Nœud de Définition de Type* que la *Déclaration d'Instance* ou un sous-type correspondant.

Si deux *Déclarations d'Instance* de la *Hiérarchie de Déclaration d'Instance* intégralement héritée ont été directement reliées par plusieurs *Références*, toutes ces *Références* doivent relier les mêmes *Nœuds*. Un exemple est fourni en Figure 15. Les instances A1 et A2 sont admises, étant donné que B1 référence le même *Nœud* en utilisant les deux *Références*, tandis qu'A3 n'est pas admis puisque deux *Nœuds* différents sont référencés. Il est à noter que cette restriction s'applique uniquement aux *Nœuds* directement reliés. Par exemple, A2 référence une C1 directement et une C1 différente via B1.

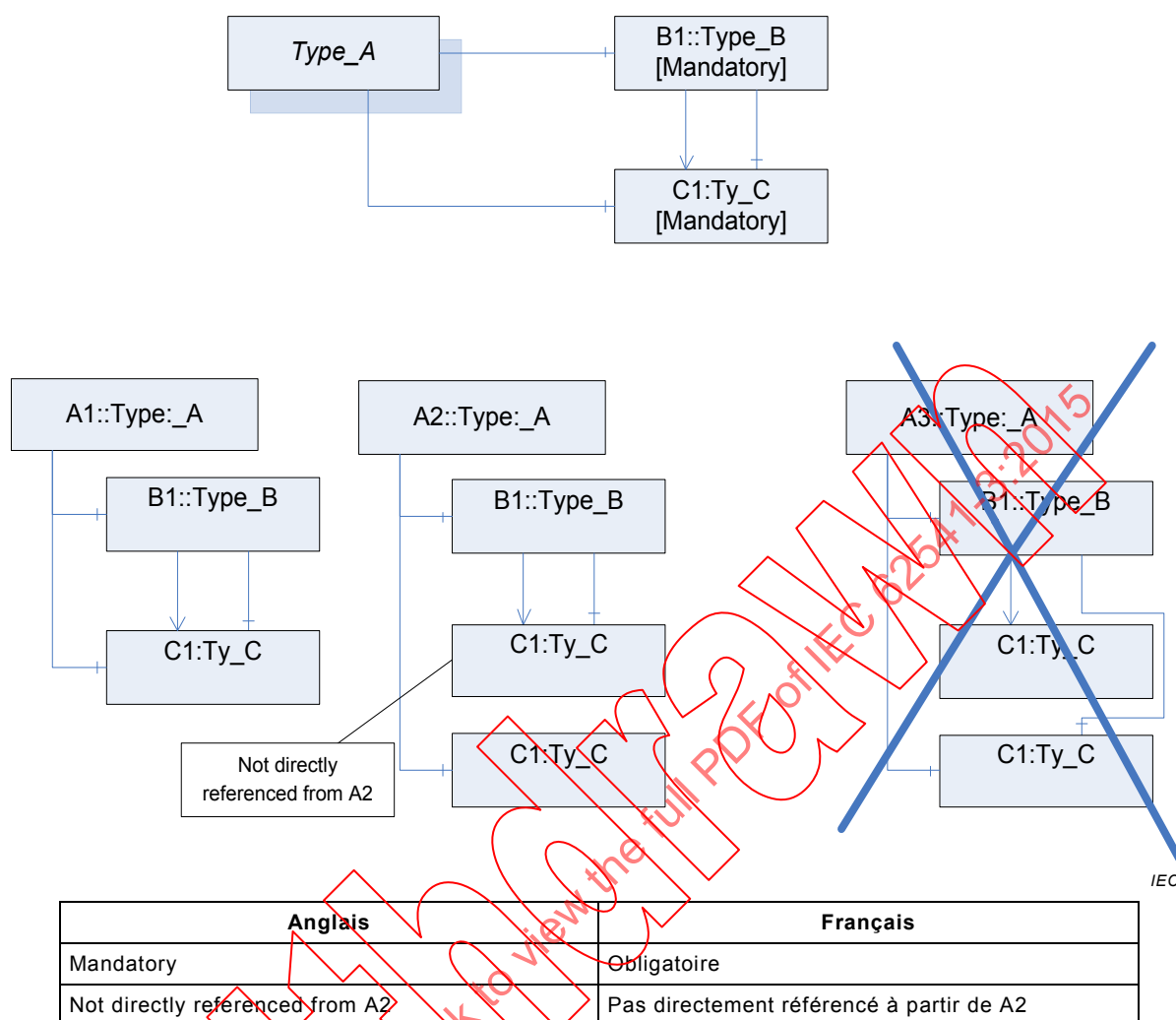


Figure 15 – Exemple de plusieurs Références entre Déclarations d'Instance

#### 6.4.4 Règles de Modélisation

##### 6.4.4.1 Généralités

Pour une définition des *Règles de Modélisation*, voir 6.4.4.5. D'autres parties de la série IEC 62541 peuvent définir des *Règles de Modélisation* supplémentaires. Dans OPC UA, les *Règles de Modélisation* sont un concept extensible; par conséquent les fournisseurs peuvent définir leurs propres *Règles de Modélisation*.

Il est à noter que les *Règles de Modélisation* définies dans la présente norme ne spécifient pas la manière de traiter des *Références* non hiérarchiques entre *Déclarations d'Instance*, c'est-à-dire qu'il incombe au *Serveur* d'établir si ces *Références* existent ou non dans une hiérarchie d'instance. D'autres *Règles de Modélisation* peuvent également définir le comportement de *Références* non hiérarchiques entre *Déclarations d'Instance*.

Les *Règles de Modélisation* sont représentées dans l'*Espace d'Adressage* comme des *Objets* du *Type d'Objet* "Type de Règle de Modélisation". Quelques *Propriétés* définissent la sémantique commune aux *Règles de Modélisation*. La présente édition de la norme spécifie uniquement une *Propriété* pour les *Règles de Modélisation*. Il est admis que des éditions futures définissent des *Propriétés* supplémentaires pour les *Règles de Modélisation*. L'IEC 62541-5 spécifie la représentation des *Objets* "Règle de Modélisation", leurs *Propriétés* et leurs types dans l'*Espace d'Adressage*. La sémantique des *Propriétés* pour les *Règles de Modélisation* est définie en 6.4.4.2.

Le paragraphe 6.4.4.4 définit la méthode de modification de la *Règle de Modélisation* lorsqu'il s'agit d'instancier des *Déclarations d'Instance* en termes de *Propriétés*. Le paragraphe 6.4.4.3 définit la méthode de modification de la *Règle de Modélisation* lorsqu'il s'agit de remplacer des *Déclarations d'Instance* par des sous-types en termes de *Propriétés*.

#### 6.4.4.2 Propriétés décrivant des Règles de Modélisation – Règle de Nommage (NamingRule)

La *Règle de Nommage* est une *Propriété* obligatoire d'une *Règle de Modélisation*. Elle spécifie le but d'une *Déclaration d'Instance*. Chaque *Déclaration d'Instance* référence une *Règle de Modélisation* et ainsi la *Règle de Nommage* est définie pour chaque *Déclaration d'Instance*.

Il est admis trois valeurs pour la *Règle de Nommage* d'une *Règle de Modélisation*: *Facultative*, *Obligatoire* et *Contrainte*.

La sémantique suivante s'applique pendant toute la durée de vie d'une instance fondée sur un *Nœud de Définition de Type* ayant une *Déclaration d'Instance*.

Pour une instance A1 d'un *Nœud de Définition de Type* "Type Alpha" ayant une *Déclaration d'Instance* B1 disposant d'une *Règle de Modélisation* qui utilise la *Règle de Nommage* "Facultative", la règle suivante s'applique: Pour chaque *Chemin de Navigation* du Type Alpha à B1, l'instance A1 peut avoir ou ne pas avoir un *Nœud* similaire (voir 6.2.4) pour B1 avec le même *Chemin de Navigation*. Si ce *Nœud* existe, alors le service "Traduire Chemins de Navigation en Identificateurs de Nœuds" (TranslateBrowsePathsToNodeIds) (voir l'IEC 62541-4) renvoie ce *Nœud* comme première entrée de la liste.

Pour une instance A1 d'un *Nœud de Définition de Type* "Type Alpha" ayant une *Déclaration d'Instance* B1 disposant d'une *Règle de Modélisation* qui utilise la *Règle de Nommage* "Obligatoire", la règle suivante s'applique. Pour chaque *Chemin de Navigation* du Type Alpha à B1, l'instance A1 doit avoir un *Nœud* similaire (voir 6.2.4) pour B1 avec le même *Chemin de Navigation* si tous les *Nœuds* du *Chemin de Navigation* existent. Par exemple, si un *Nœud* dans le *Chemin de Navigation* a une *Règle de Nommage* "Facultative" et qu'il est omis dans l'instance, tous les enfants de ce *Nœud* sont également omis, indépendamment de leurs *Règles de Modélisation*.

Si une *Déclaration d'Instance* a une *Règle de Modélisation* utilisant la *Règle de Nommage* "Contrainte", elle indique que le *Nom de Navigation* de la *Déclaration d'Instance* est sans importance mais qu'une autre sémantique est définie avec la *Règle de Modélisation*. Le Service "Traduire Chemins de Navigation en Identificateurs de Nœuds" (TranslateBrowsePathsToNodeIds) (voir l'IEC 62541-4) ne peut pas en général être utilisé pour accéder à des instances fondées sur ces *Déclarations d'Instance*.

#### 6.4.4.3 Règles de sous-typage pour des Propriétés de Règles de Modélisation

Il est admis que des sous-types remplacent des *Règles de Modélisation* en ce qui concerne leurs *Déclarations d'Instance*. De manière générale, les contraintes de sous-typage doivent être rendues plus contraignantes et non pas le contraire. Par conséquent, il n'est pas admis pour le supertype de spécifier qu'il doit exister une instance avec le nom (*Règle de Nommage* "Obligatoire") et, pour le sous-type, rendre la règle facultative (*Règle de Nommage* "Facultative"). Le Tableau 15 précise les modifications autorisées pour les *Propriétés* lorsque les *Règles de Modélisation* sont changées dans le sous-type.

**Tableau 15 – Règles applicables aux Propriétés des Règles de Modélisation en cas de Sous-typage**

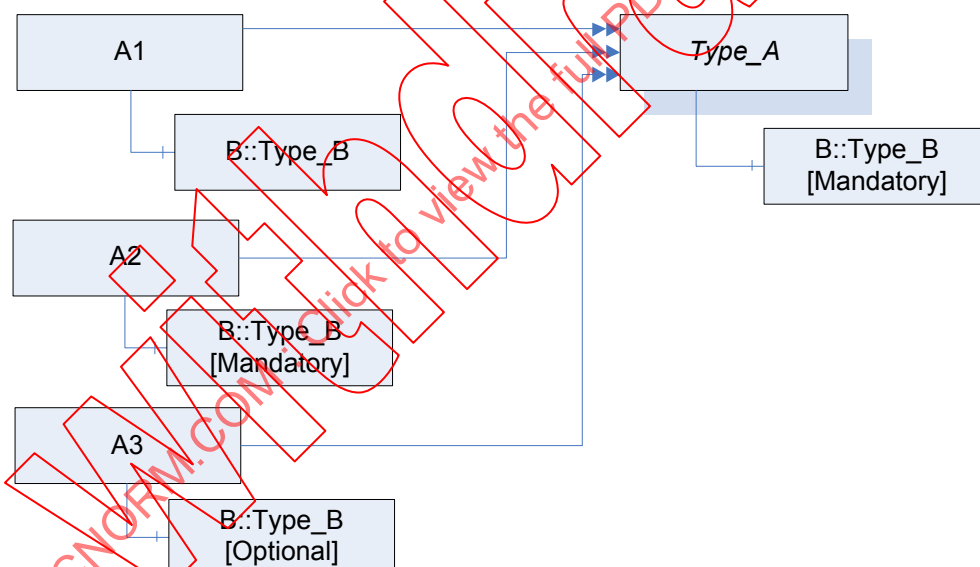
|                  | Valeur pour le supertype | Valeur pour le sous-type   |
|------------------|--------------------------|----------------------------|
| Règle de Nommage | Obligatoire              | Obligatoire                |
| Règle de Nommage | Facultative              | Obligatoire ou Facultative |
| Règle de Nommage | Contrainte               | Contrainte                 |

#### 6.4.4.4 Règles d'instanciation pour les Propriétés des Règles de Modélisation

Il existe deux différents cas d'utilisation lors de la création d'une instance 'A' fondée sur un *Nœud de Définition de Type* "Type\_A". 'A' est utilisé comme une instance normale ou comme *Déclaration d'Instance* d'un autre *Nœud de Définition de Type*.

Dans le premier cas, il n'est pas exigé que des instances nouvellement créées ou référencées, fondées sur des *Déclarations d'Instance*, aient une *Règle de Modélisation*; cependant, il est permis qu'elles aient une *Règle de Modélisation* indépendante de la *Règle de Modélisation* de leur *Déclaration d'Instance*.

Un exemple est fourni en Figure 16. Les instances A1, A2 et A3 sont des instances valides de Type\_A, même si B de A1 n'a aucune *Règle de Modélisation* et B de A3 a une *Règle de Modélisation* différente de B de Type\_A.



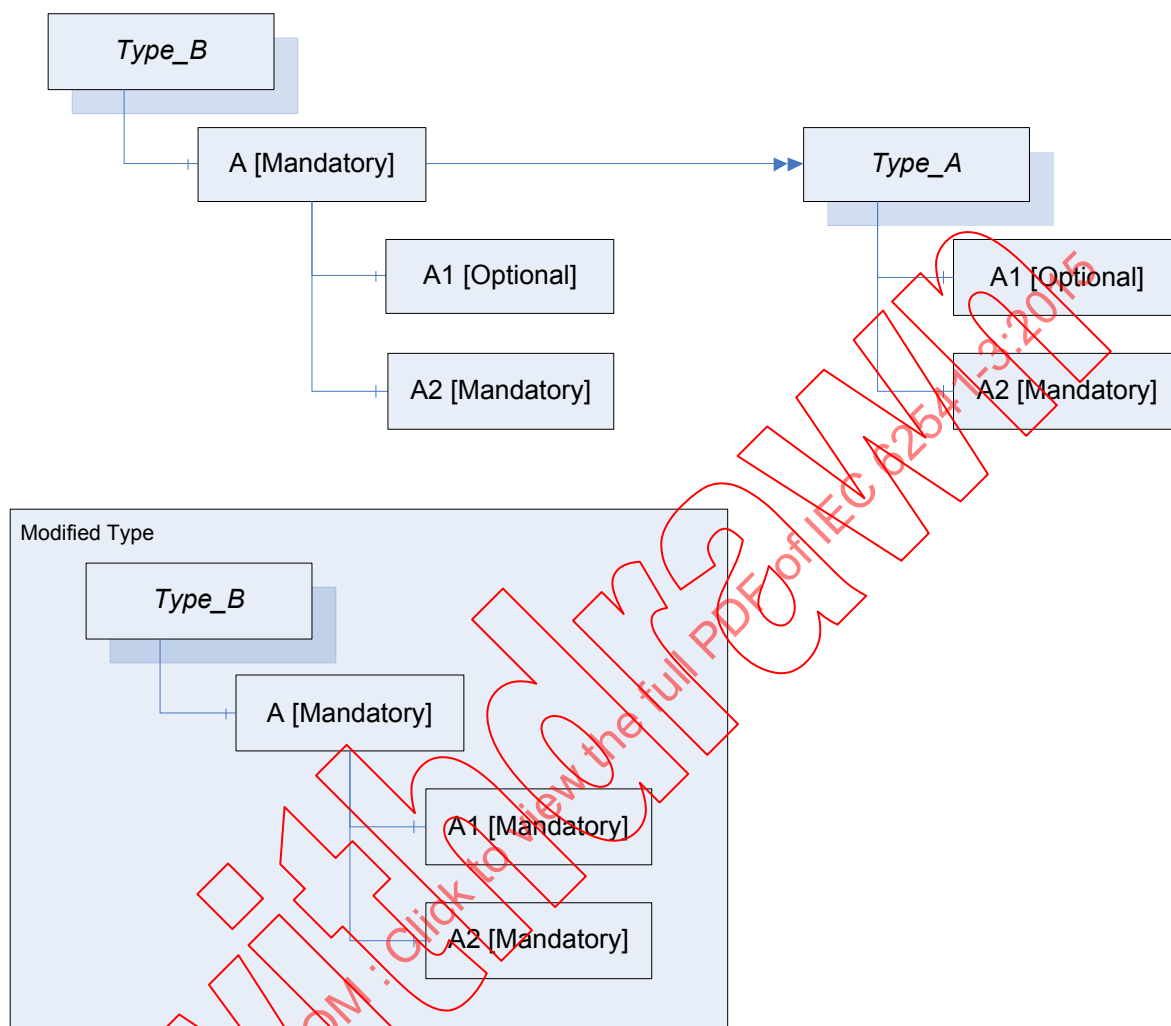
IEC

| Anglais   | Français    |
|-----------|-------------|
| Mandatory | Obligatoire |
| Optional  | Facultatif  |

**Figure 16 – Exemple de modification d'instances sur la base des Déclarations d'Instance**

Dans le second cas, toutes les instances qui sont directement ou indirectement référencées à partir de 'A', fondées sur des *Déclarations d'Instance* de 'Type\_A', maintiennent initialement la même *Règle de Modélisation* que leurs *Déclarations d'Instance*. Les *Règles de Modélisation* peuvent être mises à jour; les modifications autorisées aux *Règles de Modélisation* de ces *Nœuds* sont les mêmes que celles définies pour le sous-typage en 6.4.4.3.

La Figure 17 donne un exemple de ce scénario. Le Type\_B utilise une *Déclaration d'Instance* fondée sur le Type\_A (partie supérieure de la Figure). Par la suite, la *Règle de Modélisation* de la *Déclaration d'Instance* A1 est modifiée (partie inférieure de la Figure). La *Règle de Nommage* de A1 est devenue *obligatoire* (modifiée à partir de "Facultative").



IEC

| Anglais       | Français     |
|---------------|--------------|
| Mandatory     | Obligatoire  |
| Optional      | Facultative  |
| Modified type | Type modifié |

**Figure 17 – Exemple de modification de Déclarations d'Instance fondées sur une Déclaration d'Instance**

#### 6.4.4.5 Règles de Modélisation normalisées

##### 6.4.4.5.1 Intitulés des Règles de Modélisation normalisées

Le reste de 6.4.4.5 définit les *Règles de Modélisation*. Le Tableau 16 résume les *Propriétés* de ces *Règles de Modélisation*.

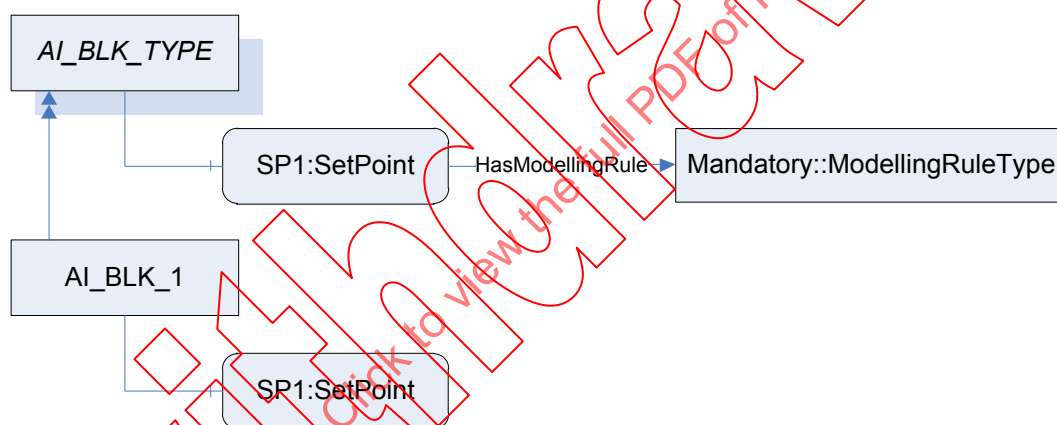
**Tableau 16 – Propriétés des Règles de Modélisation**

| Intitulé                                          | Règle de Nommage |
|---------------------------------------------------|------------------|
| Obligatoire                                       | Obligatoire      |
| Facultative                                       | Facultative      |
| ExposesItsArray (Expose sa Matrice)               | Contrainte       |
| OptionalPlaceholder (Espace Réserve Facultatif)   | Contrainte       |
| MandatoryPlaceholder (Espace Réserve Obligatoire) | Contrainte       |

#### 6.4.4.5.2 Obligatoire

Une *Déclaration d'Instance* marquée d'une *Règle de Modélisation "Obligatoire"* satisfait précisément à la sémantique définie pour la *Règle de Nommage "Obligatoire"*. Cela signifie que pour chaque *Chemin de Navigation* existant sur l'instance, il doit exister un *Nœud* similaire, mais il n'est pas défini si un nouveau *Nœud* est créé ou si un *Nœud* existant est référencé.

Par exemple, le *Nœud de Définition de Type* d'un bloc fonctionnel "AI\_BLK\_TYPE" aura un point de consigne "PC1". Une instance de ce type "AI\_BLK\_1" aura un point de consigne nouvellement créé "PC1" comme *Nœud* similaire à la *Déclaration d'Instance* PC1. Cet exemple est illustré en Figure 18.



IEC

| Anglais                      | Français                                   |
|------------------------------|--------------------------------------------|
| SP1 SetPoint                 | Point de consigne PC1                      |
| HasModellingRule             | Dispose de Règle de Modélisation           |
| Mandatory::ModellingRuleType | Obligatoire: Type de Règle de Modélisation |

**Figure 18 – Utilisation de la Règle de Modélisation normalisée "Nouveau"**

Un exemple complexe, provenant des *Règles de Modélisation "Obligatoire"* et *"Facultative"*, est donné en 6.4.4.5.3.

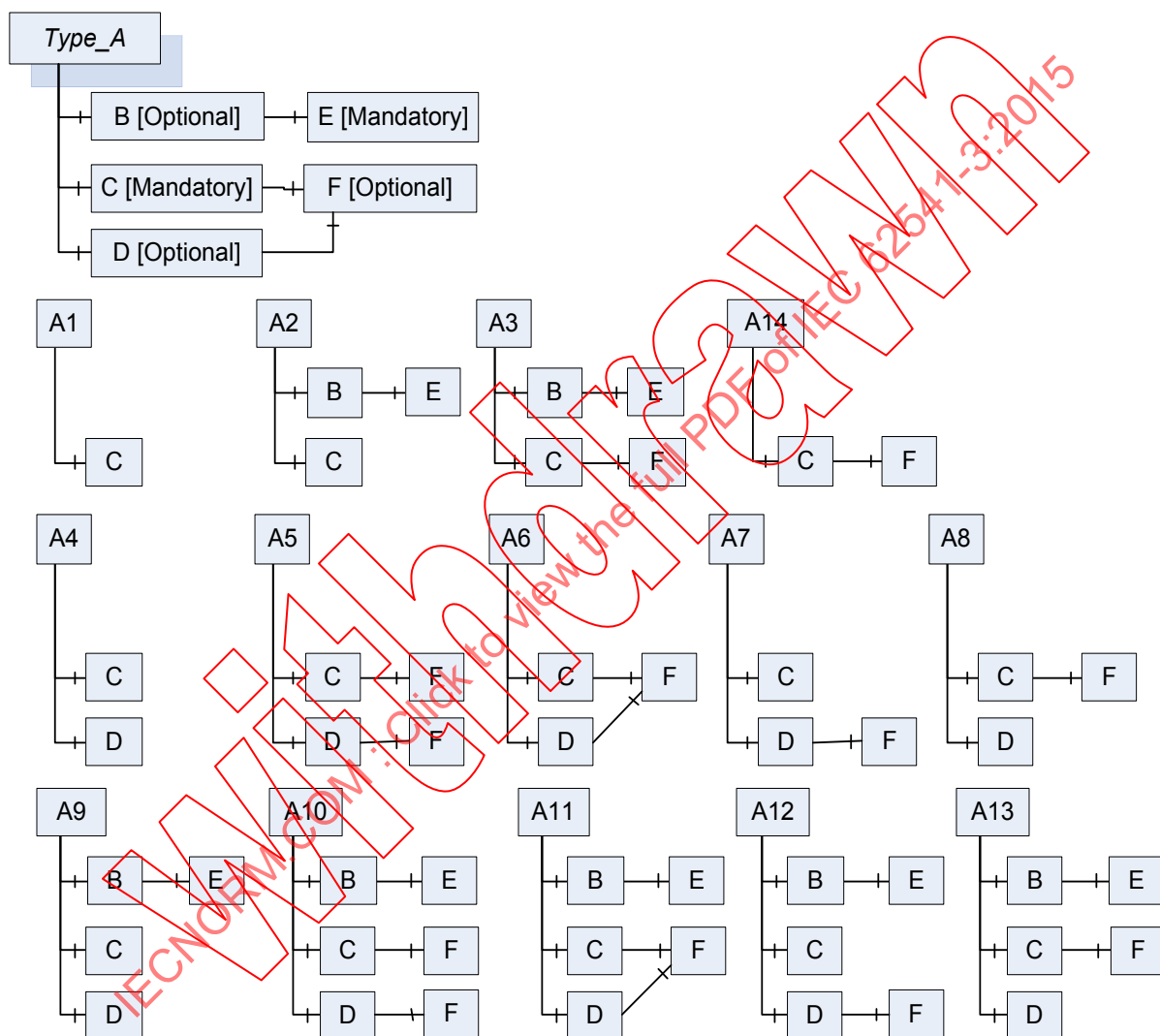
#### 6.4.4.5.3 Facultatif

Une *Déclaration d'Instance* marquée d'une *Règle de Modélisation "Facultative"* satisfait précisément à la sémantique définie pour la *Règle de Nommage "Facultative"*. Cela signifie que pour chaque *Chemin de Navigation* existant sur l'instance, il peut exister un *Nœud* similaire, mais il n'est pas défini si un nouveau *Nœud* est créé ou si un *Nœud* existant est référencé.

Un exemple d'utilisation des *Règles de Modélisation "Facultative"* et *"Obligatoire"* est illustré en Figure 19. L'exemple contient un *Type d'Objet* "Type\_A" et toutes les combinaisons valides d'instances nommées A1 à A13. À noter que si le B "facultatif" est fourni, le "E" obligatoire est

également à fournir; dans le cas contraire il ne l'est pas. F est référencé par C et D. Dans l'instance, il peut s'agir du même *Nœud* ou de deux *Nœuds* différents ayant le même *Nom de Navigation* (*Nœud* similaire à la *Déclaration d'Instance* F). L'exemple ne tient pas compte du fait que les instances ont ou n'ont pas des *Règles de Modélisation*. On suppose que chaque F est similaire à la *Déclaration d'Instance* F, etc.

Dans le cas où il y aurait une *Référence* non hiérarchique entre E et F dans la *Hierarchie de Déclaration d'Instance*, il n'est pas spécifié si elle a lieu ou non dans la hiérarchie d'instance. Dans le cas de A10, il pourrait y avoir une référence à partir de E à un F et non à l'autre F, ou aux deux ou à aucun d'entre eux.



IEC

| Anglais   | Français    |
|-----------|-------------|
| Mandatory | Obligatoire |
| Optional  | Facultative |

**Figure 19 – Exemple d'utilisation des Règles de Modélisation normalisées "Facultative" et "Obligatoire"**

#### 6.4.4.5.4 ExposesItsArray (Expose Sa Matrice)

La *Règle de Modélisation ExposesItsArray (Expose Sa Matrice)* présente une sémantique particulière pour les *Types de Variables* ayant comme type de données une matrice unidimensionnelle ou multidimensionnelle. Elle indique que chaque valeur de la matrice sera également présentée comme une *Variable* dans l'*Espace d'Adressage*.

La *Règle de Modélisation "ExposesItsArray"* ne peut être appliquée qu'aux *Déclarations d'Instance de Classe de Nœud "Variable"* qui font partie d'un *Type de Variable* ayant comme type de données une matrice unidimensionnelle ou multidimensionnelle.

La *Variable A* ayant cette *Règle de Modélisation* doit être référencée par une *Référence hiérarchique* dans un sens direct à partir d'un *Type de Variable B*. B doit avoir une valeur "*Rang de Valeur*" supérieure ou égale à zéro. Il convient que A ait un type de données qui reflète au moins des parties des données qui sont gérées dans la matrice de B. Chaque instance de B doit référencer une instance de A pour chacun de ses éléments de matrice. La *Référence* utilisée doit être du même type que la *Référence hiérarchique* qui relie B à A ou un sous-type correspondant. S'il y a plusieurs *Références hiérarchiques* dans le sens direct entre A et B, toutes les instances fondées sur B doivent être référencées avec ces *Références*.

Un exemple est donné en Figure 20. A est une instance de Type\_A ayant deux entrées dans sa matrice de valeurs. Par conséquent, elle référence deux instances du même type que la *Déclaration d'Instance ArrayExpose* (Expose Matrice). Les *Noms de Navigation* de ces instances ne sont pas définis par la *Règle de Modélisation*. En général, il n'est pas possible d'obtenir une *Variable* représentant une entrée spécifique de la matrice (par exemple la seconde). Les *Clients* tenteront d'obtenir la matrice ou d'accéder directement aux *Variables*, de sorte qu'il n'est pas nécessaire que cette information soit fournie.



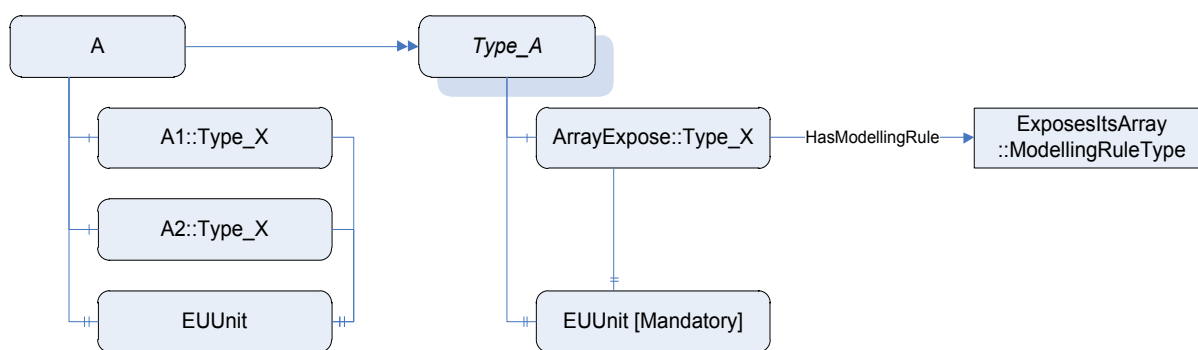
IEC

| Anglais                            | Français                                         |
|------------------------------------|--------------------------------------------------|
| ArrayExpose                        | Expose Matrice                                   |
| HasModellingType                   | Dispose d'un Type de Modélisation                |
| ExposesItsArray::ModellingRuleType | Expose Sa Matrice: Type de Règle de Modélisation |

**Figure 20 – Exemple d'utilisation de la Règle de Modélisation "ExposesItsArray"**

Il est également admis de référencer A par d'autres *Déclarations d'Instance*. Ces *Références* ont à être reflétées en chaque instance fondée sur A.

Un exemple est donné en Figure 21. La *Propriété EUUnit* (Unité EU) est référencée par *ArrayExpose* (Expose Matrice) et par conséquent chaque instance fondée sur "ArrayExpose" référence l'instance fondée sur la *Déclaration d'Instance "EUUnit"*.



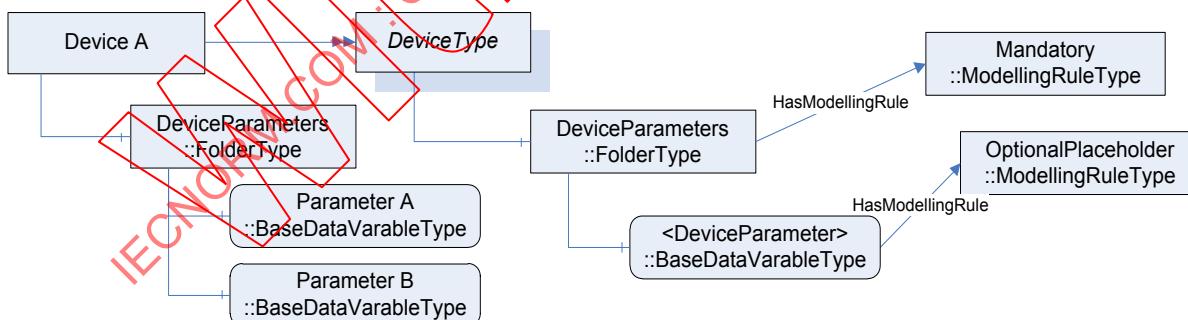
IEC

| Anglais                            | Français                                         |
|------------------------------------|--------------------------------------------------|
| ArrayExpose                        | Expose Matrice                                   |
| ExposesItsArray: ModellingRuleType | Expose Sa Matrice: Type de Règle de Modélisation |
| HasModellingRule                   | Dispose d'une Règle de Modélisation              |
| EUUnit                             | Unité EU                                         |
| EUUnit (Mandatory)                 | Unité EU (Obligatoire)                           |

**Figure 21 – Exemple complexe d'utilisation de la Règle de Modélisation "ExposesItsArray"**

#### 6.4.4.5.5 OptionalPlaceholder (Espace Réservé Facultatif)

La Règle de Modélisation "OptionalPlaceholder" vise à présenter l'information qu'une Définition de Type (TypeDefinition) complexe attend des instances de la Définition de Type pour ajouter des instances avec des Références spécifiques sans définir des Noms de Navigation pour les instances. Par exemple, un Appareil peut avoir un Dossier pour les Paramètres d'Appareil (DeviceParameters), et il convient que les Paramètres d'Appareil soient reliés à la Référence "Dispose de composant". Cependant, les noms des Paramètres d'Appareil sont spécifiques aux instances. L'exemple est présenté à la Figure 22, où une instance Appareil A ajoute deux Paramètres d'Appareil dans le Dossier.



IEC

| Anglais                                | Français                                                  |
|----------------------------------------|-----------------------------------------------------------|
| Device; DeviceType                     | Appareil; Type d'Appareil                                 |
| DeviceParameters::FolderType           | Paramètres d'Appareil:: Type de Dossier                   |
| Parameter:: BaseDataVariableType       | Paramètre::Type de Variable de Données de Base            |
| HasModellingRule                       | Dispose d'une Règle de Modélisation                       |
| Mandatory:: ModellingRuleType          | Obligatoire:: Type de Règle de Modélisation               |
| OptionalPlaceholder::ModellingRuleType | Espace Réservé facultatif:: Type de Règle de Modélisation |

**Figure 22 – Exemple d'utilisation de la Règle de Modélisation "OptionalPlaceholder"**

Il est recommandé que le *Nom de Navigation* et le *Nom d’Affichage des Déclarations d’Instance* ayant la *Règle de Modélisation "OptionalPlaceholder"* soient compris entre parenthèses en chevron.

#### 6.4.4.5.6 MandatoryPlaceholder (Espace Réservé Obligatoire)

Par exemple, lorsque le Type d'Appareil exige qu'au moins un Paramètre d'Appareil doit exister sans préciser son *Nom de navigation*, il utilise "*MandatoryPlaceHolder*" tel qu'illustré à la Figure 23. L'Appareil A est une instance valide car elle dispose du Paramètre d'Appareil exigé. L'Appareil B n'est pas valide puisqu'il utilise le *Type de Référence* erroné pour référencer un Paramètre d'Appareil ("*Organizes*" au lieu de "*HasComponent*") et l'Appareil C n'est pas valide parce qu'il ne fournit pas du tout de Paramètre d'Appareil.



| Anglais                                 | Français                                                   |
|-----------------------------------------|------------------------------------------------------------|
| DeviceType                              | Type d'Appareil                                            |
| DeviceParameters::FolderType            | Paramètres d'Appareil:: Type de Dossier                    |
| DeviceParameter:: BaseDataVariableType  | Paramètre d'Appareil::Type de Variable de Données de Base  |
| HasModellingRule                        | Dispose d'une Règle de Modélisation                        |
| Mandatory:: ModellingRuleType           | Obligatoire:: Type de Règle de Modélisation                |
| MandatoryPlaceHolder::ModellingRuleType | Espace Réservé obligatoire:: Type de Règle de Modélisation |

**Figure 23 – Exemple d'utilisation de la Règle de Modélisation "MandatoryPlaceholder"**

La *Règle de Modélisation "MandatoryPlaceholder"* exige que chaque instance fournisse au moins une instance avec la *Définition de Type* de la *Déclaration d'Instance* ou un sous-type, et soit référencée avec le même *Type de Référence* ou un sous-type comme la *Déclaration d'Instance*. Elle n'exige pas un *Nom de Navigation* spécifique et ne peut donc pas être utilisée pour le *Service TranslateBrowsePathsToNodeIds* (voir l'IEC 62541-4).

Lorsque la *Déclaration d'Instance* est complexe, ce qui signifie qu'elle référence d'autres *Déclarations d'Instance* avec des *Références* hiérarchiques, ces *Déclarations d'Instance* ne sont plus par la suite prises en compte pour l'instanciation de la *Définition de Type*.

Il est recommandé que le *Nom de Navigation* et le *Nom d'Affichage des Déclarations d'Instance* ayant la *Règle de Modélisation "MandatoryPlaceholder"* soient compris entre parenthèses en chevron.

Lors du remplacement de la *Déclaration d'Instance*, la *Règle de Modélisation* doit demeurer celle de *"MandatoryPlaceholder"*.

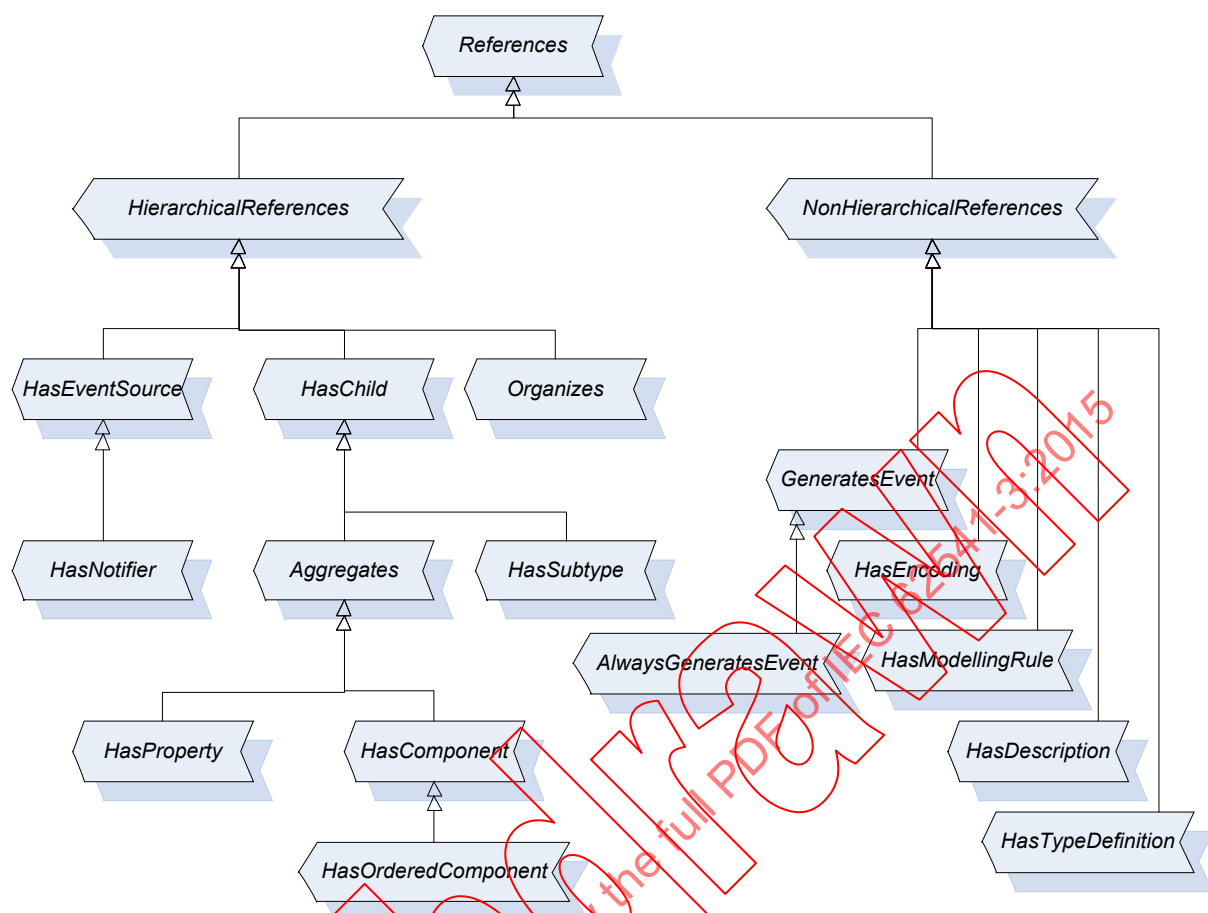
## 6.5 Modifications de définitions de type déjà utilisées

Aucun comportement n'est spécifié concernant les sous-types et les instances lorsque les *Types d'Objets* et les *Types de Variables* sont modifiés. Il incombe au *Serveur* d'établir si ces modifications sont reflétées sur les sous-types et les instances de types. Cependant, toutes les contraintes définies pour les sous-types et les instances sont à satisfaire. Par exemple, il n'est pas admis qu'une *Propriété* soit ajoutée en utilisant la *Règle de Modélisation "Obligatoire"* sur un type donné s'il existe des instances de ce type sans la *Propriété*. Dans ce cas, le *Serveur* est tenu soit d'ajouter la *Propriété* à toutes les instances de type, soit de rejeter l'ajout de la *Propriété* ajoutée.

## 7 Types de Références normalisés

### 7.1 Généralités

La présente norme définit les *Types de Références* comme partie inhérente du Modèle de l'Espace d'Adressage d'OPC UA. La Figure 24 fournit une description informelle de la hiérarchie de ces *Types de Références*. D'autres parties de la série IEC 62541 peuvent spécifier des *Types de Références* supplémentaires. Le reste de l'Article 7 définit les *Types de Références*. L'IEC 62541-5 définit leur représentation dans l'*Espace d'Adressage*.



IEC

| Anglais                   | Français                            |
|---------------------------|-------------------------------------|
| References                | Références                          |
| HierarchicalReferences    | Références Hiérarchiques            |
| NonHierarchicalReferences | Références Non Hiérarchiques        |
| HasEventSource            | Dispose d'une Source d'Événements   |
| HasChild                  | Dispose d'une Référence Enfant      |
| Organizes                 | Organise                            |
| GeneratesEvent            | Génère un Événement                 |
| HasNotifier               | Dispose d'un Notificateur           |
| Aggregates                | Regroupe                            |
| HasSubtype                | Dispose d'un Sous-Type              |
| HasEncoding               | Dispose d'un Codage                 |
| HasModellingRule          | Dispose d'une Règle de Modélisation |
| AlwaysGeneratesEvent      | Génère toujours des événements      |
| HasProperty               | Dispose d'une Propriété             |
| HasComponent              | Dispose d'un Composant              |
| HasDescription            | Dispose d'une Description           |
| HasTypeDefinition         | Dispose d'une Définition de Type    |
| HasOrderedComponent       | Dispose de Composants Ordonnés      |

Figure 24 – Hiérarchie d'un Type de Référence normalisé

## 7.2 Type de Référence "Références"

Le *Type de Référence "Références"* est un *Type de Référence* abstrait; seuls ses sous-types peuvent être utilisés.

Aucune sémantique n'est associée à ce *Type de Référence*. Il s'agit du type de base de tous les *Types de Références*. Tous les *Types de Références* doivent être un sous-type de ce *Type de Référence* de base – directement ou indirectement. Le principal objectif de ce *Type de Référence* est de permettre l'utilisation de filtres et de requêtes simples dans les *Services* correspondants de l'IEC 62541-5.

Il n'y a pas de contraintes définies pour ce *Type de Référence* abstrait.

## 7.3 Type de Référence "HierarchicalReferences" (Références Hiérarchiques)

Le *Type de Référence "HierarchicalReferences"* (*Références Hiérarchiques*) est un *Type de Référence* abstrait; seuls ses sous-types peuvent être utilisés.

La sémantique des *Références Hiérarchiques* a pour but d'indiquer que les *Références* des *Références Hiérarchiques* couvrent une hiérarchie. Ceci signifie qu'il peut être utile de présenter des *Nœuds* liés à des *Références* de ce type de manière hiérarchique. Les *Références Hiérarchiques* n'interdisent pas les boucles. Par exemple, en partant du *Nœud* "A" et en suivant les *Références Hiérarchiques*, on peut être amené à explorer de nouveau le *Nœud* "A".

Il n'est pas admis d'avoir une *Propriété*, comme *SourceNode* d'une *Référence* de quelque sous-type que ce soit de ce *Type de Référence* abstrait.

Il n'est pas admis que le *SourceNode* et le *TargetNode* d'une *Référence* de *Type de Référence Références Hiérarchiques* soient les mêmes; en d'autres termes, il n'est pas admis d'avoir des autoréférences utilisant des *Références Hiérarchiques*.

## 7.4 Type de Référence "NonHierarchicalReferences" (Références Non Hiérarchiques)

Le *Type de Référence "NonHierarchicalReferences"* (*Références Non Hiérarchiques*) est un *Type de Référence* abstrait; seuls ses sous-types peuvent être utilisés.

La sémantique des *Références Non Hiérarchiques* a pour but de montrer que ses sous-types ne couvrent pas une hiérarchie et qu'il convient de ne pas les suivre pour tenter de présenter une hiérarchie. Pour faire la distinction entre les *Références* hiérarchiques et non hiérarchiques, tous les *Types de Références* concrets doivent hériter soit des *Références hiérarchiques*, soit des *Références non hiérarchiques*, directement ou indirectement.

Il n'y a pas de contraintes définies pour ce *Type de Référence* abstrait.

## 7.5 Type de Référence "HasChild" (Dispose d'une Référence Enfant)

Le *Type de Référence HasChild* (*Dispose d'une Référence Enfant*) est un *Type de Référence* abstrait; seuls ses sous-types peuvent être utilisés. Il s'agit d'un sous-type des *Références Hiérarchiques*.

La sémantique a pour but d'indiquer que les *Références* de ce type couvrent une hiérarchie sans boucles.

En partant du *Nœud* "A" et en suivant uniquement les *Références* des sous-types du *Type de Référence "Dispose d'une Référence Enfant"*, on ne doit jamais pouvoir revenir à "A". Il est cependant admis qu'en suivant les *Références*, il peut y avoir plusieurs chemins menant vers un autre *Nœud* "B".

## 7.6 Type de Référence "Aggregates" (Regroupe)

Le *Type de Référence Aggregates (Regroupe)* est un *Type de Référence* abstrait; seuls ses sous-types peuvent être utilisés. Il s'agit d'un sous-type du type "*Dispose d'une Référence Enfant*".

La sémantique a pour but d'indiquer qu'une partie (le *TargetNode*) appartient au *SourceNode*. Elle ne spécifie pas le propriétaire du *TargetNode*.

Il n'y a pas de contraintes définies pour ce *Type de Référence* abstrait.

## 7.7 Type de Référence "HasComponent" (Dispose d'un composant)

Le *Type de Référence HasComponent (Dispose d'un Composant)* est un *Type de Référence* concret qui peut être utilisé directement. Il s'agit d'un sous-type du *Type de Référence Regroupe*.

La sémantique indique une relation "fait partie de". Le *TargetNode* (*Nœud Cible*) d'une *Référence* de *Type de Référence "Dispose d'un Composant"* fait partie du *SourceNode* (*Nœud Source*). Ce *Type de Référence* est utilisé pour relier des *Objets* ou des *Types d'Objets* avec des *Objets*, *Variables de Données* et *Méthodes* qu'ils contiennent, ainsi que des *Variables* complexes ou des *Types de Variables* avec leurs *Variables de Données*.

Comme tous les autres *Types de Références*, ce *Type de Référence* ne spécifie rien concernant le propriétaire des parties, même s'il représente une sémantique relationnelle "fait partie de". En d'autres termes, il n'est pas spécifié si le *TargetNode* d'une *Référence* du *Type de Référence "Dispose d'un Composant"* est supprimé lorsque le *SourceNode* est supprimé.

Le *TargetNode* de ce *Type de Référence* doit être une *Variable*, un *Objet* ou une *Méthode*.

Si le *TargetNode* est une *Variable*, le *SourceNode* doit être un *Objet*, un *Type d'Objet*, une *Variable de Données* ou un *Type de Variable*. En utilisant la *Référence "Dispose d'un Composant"*, la *Variable* est définie comme *Variable de Données*.

Si le *TargetNode* est un *Objet* ou une *Méthode*, le *SourceNode* doit être un *Objet* ou un *Type d'Objet*.

## 7.8 Type de Référence "HasProperty" (Dispose d'une Propriété)

Le *Type de Référence HasProperty (Dispose d'une Propriété)* est un *Type de Référence* concret qui peut être utilisé directement. Il s'agit d'un sous-type du *Type de Référence Regroupe*.

La sémantique a pour but d'identifier les *Propriétés* d'un *Nœud*. Les *Propriétés* sont décrites en 4.4.2.

Le *SourceNode* de ce *Type de Référence* peut être toute *Classe de Nœud*. Le *TargetNode* doit être *Variable*. En utilisant la *Référence "Dispose d'une Propriété"*, la *Variable* est définie comme *Propriété*. Étant donné que les *Propriétés* ne doivent pas avoir de *Propriétés*, une *Propriété* ne doit jamais être le *SourceNode* d'une *Référence "Dispose d'une Propriété"*.

## 7.9 Type de Référence "HasOrderedComponent " (Dispose de Composants Ordonnés)

Le *Type de Référence HasOrderedComponent (Dispose de Composants Ordonnés)* est un *Type de Référence* concret qui peut être utilisé directement. Il s'agit d'un sous-type du *Type de Référence Dispose d'un Composant*.

La sémantique du *Type de Référence "Dispose de Composants Ordonnés"* – outre celle *Type de Référence "Dispose d'un Composant"* – est d'indiquer que lorsque la navigation est effectuée à partir d'un *Nœud* et suivant des *Références* de ce type ou de son sous-type, toutes les *Références* sont renvoyées, dans un ordre bien défini, au *Service de Navigation* défini dans l'IEC 62541-4. L'ordre est fonction du *Serveur*, mais le *Client* peut considérer que le *Serveur* renvoie toujours les composants dans le même ordre.

Il n'y a pas d'autres contraintes définies pour ce *Type de Référence*.

### 7.10 Type de Référence "HasSubtype" (Dispose d'un Sous-type)

Le *Type de Référence HasSubtype (Dispose d'un Sous-type)* est un *Type de Référence* concret qui peut être utilisé directement. Il s'agit d'un sous-type du *Type de Référence "Dispose d'une Référence Enfant"*.

La sémantique de ce *Type de Référence* vise à exprimer une relation entre des sous-types et des types. Elle est utilisée pour couvrir la hiérarchie du *Type de Référence*, dont la sémantique est spécifiée en 5.3.3.3; la hiérarchie de *Type de Données* est spécifiée en 5.8.3, et d'autres hiérarchies du sous-type sont spécifiées à l'Article 6.

Le *SourceNode* des *Références* de ce type doit être un *Type d'Objet*, un *Type de Variable*, un *Type de Données* ou un *Type de Référence* et le *TargetNode* doit être de la même *Classe de Nœud* que le *SourceNode*. Chaque *Type de Référence* doit être le *TargetNode* d'au plus une *Référence* du type «*HasSubtype*».

### 7.11 Type de référence "Organizes" (Organise)

Le *Type de référence Organizes (Organise)* est un *Type de Référence* concret qui peut être utilisé directement. Il s'agit d'un sous-type des *Références Hiérarchiques*.

La sémantique de ce *Type de Référence* vise à organiser les *Nœuds* dans l'*Espace d'Adressage*. Elle peut être utilisée pour couvrir de multiples hiérarchies indépendamment de toute hiérarchie créée avec les *Références Regroupe* sans boucles.

Le *SourceNode* des *Références* de ce type doit être un *Objet* ou une *Vue*. S'il s'agit d'un *Objet*, il convient alors qu'il soit du *Type d'Objet "Type de Dossier"* ou de l'un de ses sous-types (voir 5.5.3).

Le *TargetNode* de ce *Type de Référence* peut être toute *Classe de Nœud*.

### 7.12 Type de Référence "HasModellingRule" (Dispose d'une Règle de Modélisation)

Le *Type de Référence HasModellingRule (Dispose d'une Règle de Modélisation)* est un *Type de Référence* concret qui peut être utilisé directement. Il s'agit d'un sous-type des *Références Non Hiérarchiques*.

La sémantique de ce *Type de Référence* vise à lier la *Règle de Modélisation* à un *Objet*, une *Variable* ou une *Méthode*. Les mécanismes des *Règles de Modélisation* sont décrits en 6.4.4.

Le *SourceNode* de ce *Type de Référence* doit être un *Objet*, une *Variable* ou une *Méthode*. Le *TargetNode* doit être un *Objet* de *Type d'Objet "Règle de Modélisation"* ou l'un de ses sous-types.

Chaque *Nœud* doit être un *SourceNode* d'au plus une *Référence "Dispose d'une Règle de Modélisation"*.

### 7.13 Type de Référence "HasTypeDefinition" (Dispose d'une Définition de Type)

Le *Type de Référence HasTypeDefinition (Dispose d'une Définition de Type)* est un *Type de Référence* concret qui peut être utilisé directement. Il s'agit d'un sous-type des *Références Non Hiérarchiques*.

La sémantique de ce *Type de Référence* vise à lier un *Objet* ou une *Variable*, respectivement, à son *Type d'Objet* ou à son *Type de Variable*. Les relations entre types et instances sont décrites en 4.5.

Le *SourceNode* de ce *Type de Référence* doit être un *Objet* ou une *Variable*. Si le *SourceNode* est un *Objet*, alors le *TargetNode* doit être un *Type d'Objet*. Si le *SourceNode* est une *Variable*, alors le *TargetNode* doit être un *Type de Variable*.

Chaque *Variable* et chaque *Objet* doit être le *SourceNode* de précisément une *Référence "Dispose d'une Définition de Type"*.

### 7.14 Type de Référence "HasEncoding" (Dispose d'un Codage)

Le *Type de Référence HasEncoding (Dispose d'un codage)* est un *Type de Référence* concret qui peut être utilisé directement. Il s'agit d'un sous-type des *Références Non Hiérarchiques*.

La sémantique de ce *Type de Référence* vise à référencer les *Codages de Type de Données* d'un *Type de Données*.

Le *SourceNode* des *Références* de ce type doit être un *Type de Données*.

Le *TargetNode* de ce *Type de Référence* doit être un *Objet* du *Type d'Objet Type de Codage de Type de Données* ou de l'un de ses sous-types (voir 5.8.4).

### 7.15 Type de Référence "HasDescription" (Dispose d'une Description)

Le *Type de Référence HasDescription (Dispose d'une Description)* est un *Type de Référence* concret qui peut être utilisé directement. Il s'agit d'un sous-type des *Références Non Hiérarchiques*.

La sémantique de ce *Type de Référence* vise à référencer la *Description de Type de Données* d'un *Codage de Type de Données*.

Le *SourceNode* des *Références* de ce type doit être un *Objet* du *Type d'Objet Type de Codage de Type de Données* ou de l'un de ses sous-types.

Le *TargetNode* de ce *Type de Référence* doit être une *Variable* du *Type de Variable "Type de Description de Type de Données"* ou de l'un de ses sous-types (voir 5.8.4).

### 7.16 Type de Référence "GeneratesEvent" (Génère un Événement)

Le *Type de Référence GeneratesEvent (Génère un Événement)* est un *Type de Référence* concret qui peut être utilisé directement. Il s'agit d'un sous-type des *Références Non Hiérarchiques*.

La sémantique de ce *Type de Référence* vise à identifier les types d'*Événements* que des instances de *Types d'Objets* ou de *Types de Variables* peuvent générer ainsi que celles que les *Méthodes* peuvent générer à chaque invocation de *Méthode*.

Le *SourceNode* des *Références* de ce type doit être un *Type d'Objet*, un *Type de Variable* ou une *Méthode*.

Le *TargetNode* de ce *Type de Référence* doit être un *Type d'Objet* représentant les *Types d'Événements*, c'est-à-dire un *Type d'Événement de Base* ou l'un de ses sous-types.

### 7.17 Type de Référence "AlwaysGeneratesEvent" (Génère Toujours un Événement)

Le *Type de Référence AlwaysGeneratesEvent* (*Génère Toujours un Événement*) est un *Type de Référence* concret qui peut être utilisé directement. Il s'agit d'un sous-type du *Type de Référence "Génère un Événement"*.

La sémantique de ce *Type de Référence* vise à identifier les types d'*Événements* que les *Méthodes* ont à générer à chaque invocation de *Méthode*.

Le *SourceNode* des *Références* de ce type doit être une *Méthode*.

Le *TargetNode* de ce *Type de Référence* doit être un *Type d'Objet* représentant les *Types d'Événements*, c'est-à-dire un *Type d'Événement de Base* ou l'un de ses sous-types.

### 7.18 Type de Référence "HasEventSource" (Dispose d'une Source d'Événement)

Le *Type de Référence HasEventSource* (*Dispose d'une Source d'Événement*) est un *Type de Référence* concret qui peut être utilisé directement. Il s'agit d'un sous-type des *Références Hiérarchiques*.

La sémantique de ce *Type de Référence* vise à relier des sources d'événements dans une organisation hiérarchique sans boucles. Ce *Type de Référence* et ses éventuels sous-types sont destinés à être utilisés pour découvrir la génération d'*Événements* dans un *Serveur*. Il n'est pas exigé qu'ils soient présents pour qu'un *Serveur* génère un *Événement* à partir de sa source (provoquant l'*Événement*) vers ses *Nœuds* notificateurs. En particulier, le notificateur racine d'un *Serveur*, l'*Objet "Serveur"* défini dans l'IEC 62541-5, est toujours capable de fournir l'ensemble des *Événements* à partir du *Serveur* et, en tant que tel, il a des *Références* implicites "*Dispose d'une Source d'Événement*" vers chaque source d'événement dans un *Serveur* donné.

Le *SourceNode* de ce *Type de Référence* doit être un *Objet* qui est une source d'abonnements à des événements. Une source d'abonnements à des événements est un *Objet* dont le bit *SubscribeToEvents* ("S'Abonner à des Événements") est établi dans l'*Attribut "Notificateur d'Événements"*.

Le *TargetNode* de ce *Type de Référence* peut être un *Nœud* de toute *Classe de Nœud* qui peut générer des notifications d'événements par le biais d'un abonnement à la source de référence.

En partant du *Nœud "A"* et en suivant uniquement les *Références* du *Type de Référence "Dispose d'une Source d'Événement"* ou de ses sous-types, on ne doit jamais pouvoir revenir à "A". Il est cependant admis qu'en suivant les *Références*, il peut y avoir plusieurs chemins menant vers un autre *Nœud "B"*.

### 7.19 Type de Référence "HasNotifier" (Dispose d'un Notificateur)

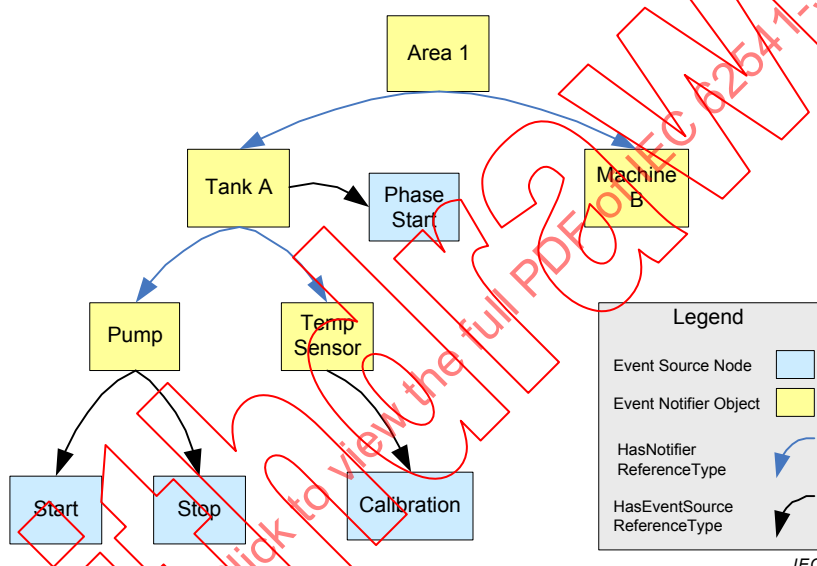
Le *Type de Référence HasNotifier* (*Dispose d'un Notificateur*) est un *Type de Référence* concret qui peut être directement utilisé. Il s'agit d'un sous-type du *Type de Référence "Dispose d'une Source d'Événement"*.

La sémantique de ce *Type de Référence* a pour objectif de relier des *Nœuds Objet* notificateurs entre eux. Le *Type de Référence* est utilisé pour établir une organisation hiérarchique des *Objets* de notification d'événements. Il s'agit d'un sous-type du *Type de Référence "Dispose d'une Source d'Événement"* défini en 7.17.

Le *SourceNode* de ce *Type de Référence* doit être un *Objet* ou une *Vue* qui est une source d'abonnement à des événements. Le *TargetNode* de ce *Type de Référence* doit être un *Objet* qui est une source d'abonnement à des événements. Une source d'abonnement à des événements est un *Objet* dont le bit "S'Abonner à des Événements" est établi dans l'*Attribut "Notificateur d'Événements"*.

Si le *TargetNode* d'une *Référence* de ce type génère un *Événement*, alors cet *Événement* doit également être fourni dans le *SourceNode* de la *Référence*.

La Figure 25 présente un exemple d'organisation possible de *Références d'Événements*. Dans cet exemple, un abonnement à des événements non filtré, dirigé vers l'*Objet* "Pompe" fournit les sources d'*Événements* "Démarrage" et "Arrêt" à l'abonné. Un abonnement à des événements non filtré, dirigé vers l'*Objet* "Zone 1", fournit les sources d'*Événements* à partir de la "Machine B", du "Réservoir A" et de toutes les sources notificatives sous le "Réservoir A".

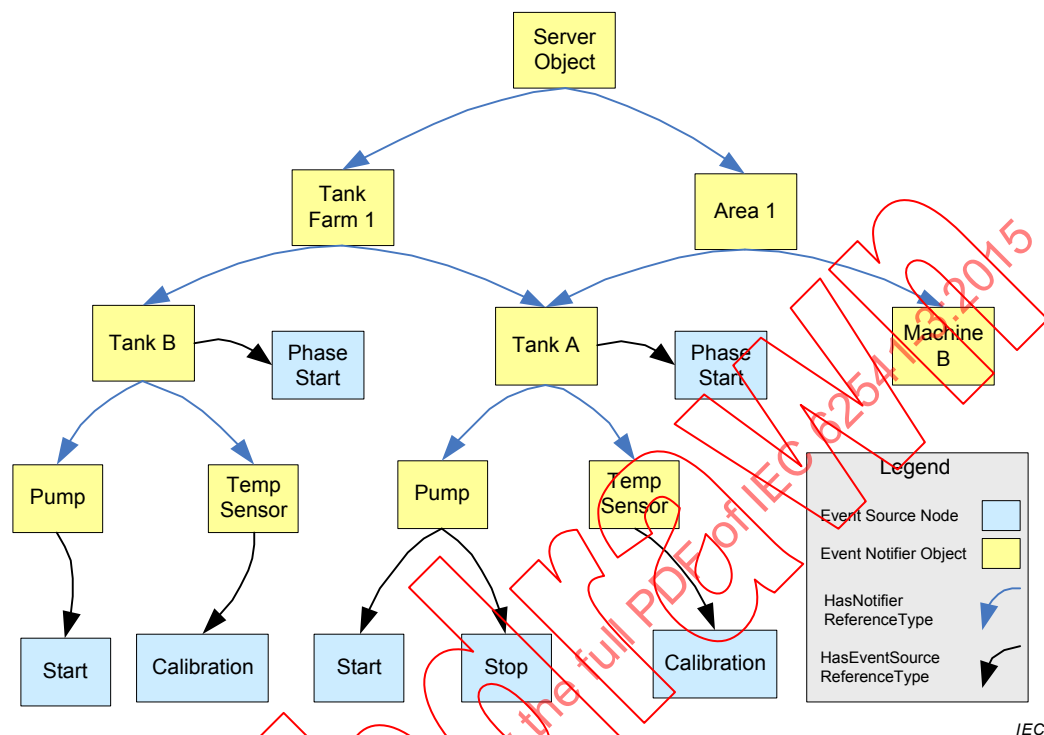


| Anglais                      | Français                                             |
|------------------------------|------------------------------------------------------|
| Area                         | Zone                                                 |
| Tank                         | Réservoir                                            |
| Phase start                  | Début de phase                                       |
| Machine                      | Machine                                              |
| Pump                         | Pompe                                                |
| Temp sensor                  | Capteur de temp.                                     |
| Start                        | Démarrage                                            |
| Stop                         | Arrêt                                                |
| Calibration                  | Etalonnage                                           |
| Legend                       | Légende                                              |
| Event Source Node            | SourceNode d'Événement                               |
| Event Notifier Object        | Objet Notificateur d'Événement                       |
| HasNotifier ReferenceType    | Type de Référence "Dispose d'un notificateur"        |
| HasEventSource ReferenceType | Type de Référence "Dispose d'une Source d'Événement" |

**Figure 25 – Exemple de Référence d'Événement**

La Figure 26 présente un autre exemple d'une organisation plus complexe de *Références d'Événements*. Dans cet exemple, les *Références* explicites sont incluses à partir de l'*Objet*

"Serveur" du *Serveur*, qui est une source de tous les *Événements* du *Serveur*. Il a été introduit une seconde organisation des *Événements* permettant de recueillir les *Événements* relatifs au "Parc de Réservoirs 1". Un abonnement à des *Événements* non filtré, dirigé vers l'*Objet* "Parc de Réservoirs 1", fournit les sources d'*Événements* à partir du "Réservoir B", du "Réservoir A" et de toutes les sources notifiables sous le "Réservoir B" et le "Réservoir A".



| Anglais                      | Français                                             |
|------------------------------|------------------------------------------------------|
| Server Object                | objet "Serveur"                                      |
| Tank farm                    | Parc de réservoirs                                   |
| Area                         | Zone                                                 |
| Tank                         | Réservoir                                            |
| Phase start                  | Début de phase                                       |
| Machine                      | Machine                                              |
| Pump                         | Pompe                                                |
| Temp sensor                  | Capteur de temp.                                     |
| Start                        | Démarrage                                            |
| Stop                         | Arrêt                                                |
| Calibration                  | Étalonnage                                           |
| Legend                       | Légende                                              |
| Event Source Node            | SourceNode d'Événement                               |
| Event Notifier Object        | Objet Notificateur d'Événement                       |
| HasNotifier ReferenceType    | Type de Référence "Dispose d'un notificateur"        |
| HasEventSource ReferenceType | Type de Référence "Dispose d'une Source d'Événement" |

**Figure 26 – Exemple de Référence à des Événements complexes**

## 8 Types de Données Normalisés

### 8.1 Généralités

Le reste de l'Article 8 définit les *Types de Données*. Leur représentation dans l'*Espace d'Adressage* et la hiérarchie du *Type de Données* sont spécifiées dans l'IEC 62541-5. D'autres parties la série IEC 62541 peuvent spécifier des *Types de Données* supplémentaires.

### 8.2 Identificateur de Nœud

#### 8.2.1 Généralités

Le *Type de Données intégré* est composé de trois éléments qui identifient un *Nœud* au sein d'un *Serveur*. Ils sont définis dans le Tableau 17.

**Tableau 17 – Définition d'un Identificateur de Nœud**

| Dénomination           | Type                                       | Description                                                                                                             |
|------------------------|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| Identificateur de Nœud | Structure                                  |                                                                                                                         |
| Indice d'Espace de Nom | UInt16 (Entier Non Signé codé sur 16 bits) | L'indice pour un URI d'espace de nom (voir 8.2.2).                                                                      |
| Type d'Identificateur  | Enum                                       | Le format et le type de données de l'Identificateur (voir 8.2.3).                                                       |
| Identificateur         | *                                          | L'identificateur utilisé pour un <i>Nœud</i> dans l' <i>Espace d'Adressage</i> d'un <i>Serveur</i> OPC UA (voir 8.2.4). |

Pour une description du codage de l'identificateur dans des Messages OPC UA, voir l'IEC 62541-6.

#### 8.2.2 Indice d'Espace de Nom

L'espace de nom est un URI qui identifie l'autorité de nommage chargée de l'attribution de l'élément identificateur de l'*Identificateur de Nœud*. Les autorités de nommage comprennent le *Serveur* local, le système sous-jacent ainsi que les organismes et consortiums de normalisation. Il est prévu que la plupart des *Nœuds* utiliseront l'URI du *Serveur* ou du système sous-jacent.

L'utilisation d'un URI d'espace de nom permet de relier plusieurs *serveurs* OPC UA au même système sous-jacent afin d'utiliser le même identificateur pour désigner le même *Objet*. Ainsi, les *Clients* qui se connectent à ces *Serveurs* peuvent reconnaître les *Objets* qu'ils ont en commun.

Les URI d'espaces de nom, tels que les noms de *Serveurs*, sont identifiés par les valeurs numériques dans les *Services* OPC UA afin de garantir une meilleure efficacité de transfert et de traitement (par exemple, les consultations de tables). Les valeurs numériques utilisées pour identifier les espaces de noms correspondent à l'indice dans la *Matrice d'Espace de Nom*. La *Matrice d'Espace de Nom* est une *Variable* qui fait partie de l'*Objet Serveur* dans l'*Espace d'Adressage* (la définition est donnée dans l'IEC 62541-5).

L'URI pour l'espace de nom OPC UA est le suivant:

"http://opcfoundation.org/UA/"

Son indice correspondant dans la table d'Espace de Nom est 0.

L'URI d'espace de nom est sensible à la casse.

### 8.2.3 Type d'Identificateur

L'élément "Type d'Identificateur" permet d'identifier le type d'*Identificateur de Nœud*, son format et son champ d'application. Ses valeurs sont définies dans le Tableau 18.

**Tableau 18 – Valeurs du Type d'Identificateur**

| Valeur    | Description                          |
|-----------|--------------------------------------|
| NUMERIC_0 | Valeur numérique                     |
| STRING_1  | Valeur de la Chaîne                  |
| GUID_2    | Identificateur Globalement Unique    |
| OPAQUE_3  | Format spécifique de l'Espace de Nom |

En général, le champ d'application des *Identificateurs de Nœuds* est le *Serveur* dans lequel ils sont définis. Certains types d'*Identificateurs de Nœuds* peuvent identifier de manière unique un *Nœud* dans un système donné ou sur plusieurs systèmes (par exemple des Identificateurs Globalement Uniques – GUID). Des identificateurs valables dans l'ensemble du système et globalement uniques permettent à des *Clients* de suivre des *Nœuds*, tels que des commandes de travaux, au fur et à mesure qu'ils se déplacent entre *Serveurs* OPC UA et qu'ils progressent dans le système.

Les identificateurs opaques sont constitués par des chaînes d'octets de format libre qui peuvent ou non être interprétables par l'utilisateur.

Les identificateurs de chaînes sont sensibles à la casse, ce qui signifie que les *Clients* doivent les considérer comme sensibles à la casse. Les *Serveurs* sont autorisés à fournir d'autres *Identificateurs de Nœuds* (voir 5.2.2). En utilisant ce mécanisme, les *Serveurs* peuvent traiter les *Identificateurs de Nœuds* comme étant non-sensibles à la casse.

### 8.2.4 Valeur d'Identificateur

L'élément valeur d'identificateur est utilisé dans le contexte des trois premiers éléments pour identifier le *Nœud*. Son type de données et son format sont définis par le Type d'Identificateur.

Les valeurs d'identificateur de Type d'Identificateur STRING\_1 sont limitées à 4 096 caractères. Les valeurs d'identificateur de Type d'Identificateur OPAQUE\_3 sont limitées à 4 096 octets.

Un *Identificateur de Nœud* nul n'a pas de signification particulière. Par exemple, de nombreux services définis dans l'IEC 62541-4 spécifient des comportements particuliers si un *Identificateur de Nœud* nul est transmis comme paramètre. Chaque Type d'Identificateur dispose d'un jeu de valeurs d'Identificateurs qui représentent un *Identificateur de Nœud* nul. Ces valeurs sont résumées dans le Tableau 19.

**Tableau 19 – Valeurs Nulles d'Identificateur de Nœud**

| Type d'Identificateur | Identificateur                                                                                                                                    |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| NUMERIC_0             | 0                                                                                                                                                 |
| STRING_1              | Une Chaîne nulle ou Vide ("")                                                                                                                     |
| GUID_2                | Un Guid (Globally Unique Identifier – Identificateur Globalement Unique) initialisé avec des zéros (par exemple 00000000-0000-0000-0000-00000000) |
| OPAQUE_3              | Une Chaîne d'Octets de Longueur = 0                                                                                                               |

Un *Identificateur de Nœud* nul a toujours un Indice d'Espace de Nom égal à 0.

Un *Nœud* dans l'*Espace d'Adressage* ne doit pas avoir une valeur nulle comme *Identificateur de Nœud*.

### 8.3 Nom Qualifié

Le *Type de Données Intégré* contient un nom qualifié. Il est utilisé par exemple comme *Nom de Navigation*. Ses éléments sont définis dans le Tableau 20. La partie nom du *Nom Qualifié* est limitée à 512 caractères.

**Tableau 20 – Définition de Nom Qualifié**

| Dénomination           | Type      | Description                                                                                                                                                                                                                                                                                                           |
|------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nom Qualifié           | Structure |                                                                                                                                                                                                                                                                                                                       |
| Indice d'Espace de Nom | UInt16    | Indice identificateur de l'espace de nom qui définit le nom.<br>L'indice est celui de l'espace de nom dans la <i>Matrice d'Espace de Nom</i> du <i>Serveur</i> local.<br>Le <i>Client</i> peut lire la <i>Variable "Matrice d'Espace de Nom"</i> pour accéder à la valeur de chaîne de caractères de l'espace de nom. |
| Nom                    | Chaîne    | La partie texte du Nom Qualifié.                                                                                                                                                                                                                                                                                      |

### 8.4 Identificateur de Paramètre de lieu

Ce *Type de Données Simple* est spécifié comme une chaîne constituée d'un composant langue et d'un composant pays/région, comme spécifié par la norme IETF RFC 3066. Le composant <pays/région> est toujours précédé d'un tiret. Le format de la chaîne de caractères de l'*Identificateur de Paramètre de lieu* est présenté ci-dessous:

<langue>[-<pays/région>], où  
 <langue> est le code ISO 639 à deux lettres identifiant une langue,  
 <pays/région> est le code ISO 3166 à deux lettres identifiant le pays/région.

Les règles de construction des *Identificateurs de Paramètre de lieu* définies par la norme IETF RFC 3066 sont limitées de la manière suivante:

- a) la spécification permet uniquement l'utilisation de zéro ou d'un composant <pays/région> à la suite d'un composant <langue> ;
- b) la spécification permet également les codes <pays/région> à trois lettres "-CHS" et "-CHT" pour les paramètres de lieu chinois "Simplifiés" et "Traditionnels";
- c) la spécification permet également l'utilisation d'autres codes <pays/région> considérés nécessaires par le *Client* ou le *Serveur*.

Des exemples d'*Identificateurs de Paramètre de lieu* utilisés dans l'OPC UA sont donnés dans le Tableau 21. Les *Clients* et les *Serveurs* fournissent toujours des *Identificateurs de Paramètre de lieu* qui identifient explicitement la langue et le pays/région.

**Tableau 21 – Exemples d'Identificateurs de Paramètre de lieu**

| Paramètre de lieu     | Identificateur de Paramètre de lieu d'OPC UA |
|-----------------------|----------------------------------------------|
| Anglais               | En                                           |
| Anglais (États-Unis)  | en-US                                        |
| Allemand              | De                                           |
| Allemand (Allemagne)  | de-DE                                        |
| Allemand (Autrichien) | de-AT                                        |

Une chaîne de caractères vide ou nulle indique que l'*Identificateur de Paramètre de lieu* est inconnu.

### 8.5 Texte Localisé

Ce *Type de Données Intégré* définit une structure contenant une chaîne dans une traduction spécifique à un paramètre de lieu donné, défini dans l'*Identificateur de paramètre de lieu*. Ses éléments sont définis dans le Tableau 22.

**Tableau 22 – Définition de Texte Localisé**

| Dénomination      | Type                                | Description                                                  |
|-------------------|-------------------------------------|--------------------------------------------------------------|
| Texte Localisé    | Structure                           |                                                              |
| Texte             | Chaîne                              | Le texte localisé.                                           |
| paramètre de lieu | Identificateur de Paramètre de lieu | L'Identificateur du paramètre de lieu (par exemple "en-US"). |

## 8.6 Argument

Ce *Type de Données Structuré* définit une spécification pour les arguments d'entrée ou de sortie d'une *Méthode*. Il est par exemple utilisé dans l'argument d'entrée et de sortie *Propriétés* pour des *Méthodes*. Ses éléments sont décrits dans le Tableau 23.

**Tableau 23 – Définition des Arguments**

| Dénomination          | Type                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Argument              | Structure              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Nom                   | Chaîne                 | La dénomination de l'argument.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Type de Données       | Identificateur de Nœud | L' <i>Identificateur de Nœud</i> du <i>Type de Données</i> de cet argument.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Rang de Valeur        | Int32                  | Indique si le <i>Type de Données</i> est une matrice et indique le nombre de dimensions de cette matrice.<br>Il peut prendre les valeurs suivantes:<br>$n > 1$ : Le <i>Type de Données</i> est une matrice ayant le nombre spécifié de dimensions.<br>Unidimensionnelle (1): Le <i>Type de Données</i> est une matrice unidimensionnelle.<br>Une Ou Plusieurs Dimensions (0): Le <i>Type de Données</i> est une matrice à une ou plusieurs dimensions.<br>Scalaire (-1): Le <i>Type de Données</i> n'est pas une matrice.<br>Indifférente (-2): Le <i>Type de Données</i> peut être scalaire ou une matrice ayant n'importe quel nombre de dimensions.<br>Scalaire Ou Unidimensionnelle (-3): Le <i>Type de Données</i> peut être scalaire ou une matrice unidimensionnelle.<br>NOTE Tous les Types de Données sont considérés comme scalaires, même s'ils ont une sémantique de type matrice comme une Chaîne d'Octets (ByteString) et une Chaîne (String). |
| Dimensions de Matrice | UInt32[]               | Spécifie la longueur de chaque dimension d'un <i>Type de Données</i> matriciel. Il permet de décrire la capacité du <i>Type de Données</i> et non la taille courante.<br>Le nombre d'éléments doit être égal à la valeur du " <i>Rang de Valeur</i> ". Doit être nul si le <i>Rang de Valeur</i> $\leq 0$ .<br>Une valeur 0 pour une dimension particulière indique que la dimension a une longueur variable.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Description           | Texte Localisé         | Une description localisée de l'argument.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

## 8.7 Type de Données de Base (BaseDataType)

Ce *Type de Données* abstrait définit une valeur qui peut avoir n'importe quel *Type de Données* valide.

Il définit une valeur spéciale nulle indiquant qu'une valeur n'est pas présente.

## 8.8 Booléen (Boolean)

Le *Type de Données Intégré* définit une valeur VRAI ou FAUX.

## 8.9 Octet (Byte)

Ce *Type de Données Intégré* définit une valeur dans une plage de 0 à 255.

## 8.10 Chaîne d'octets (ByteString)

Ce *Type de Données Intégré* définit une valeur qui est une suite de valeurs d'Octets.