

INTERNATIONAL STANDARD

NORME INTERNATIONALE



Use case methodology –

Part 3: Definition of use case template artefacts into an XML serialized format

Méthodologie des cas d'utilisation –

**Partie 3: Définition des artefacts de formulaire type de modèle de cas
d'utilisation au format sérialisé XML**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2017 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
info@iec.ch
www.iec.ch

About the IEC

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

About IEC publications

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

IEC publications search - webstore.iec.ch/advsearchform

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 It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

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

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

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

Electropedia - www.electropedia.org

The world's leading online dictionary on electrotechnology, containing more than 22 000 terminological entries in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

67 000 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.

A propos de l'IEC

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

A propos des publications IEC

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

Recherche de publications IEC -

webstore.iec.ch/advsearchform

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

IEC Just Published - webstore.iec.ch/justpublished

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

Service Clients - webstore.iec.ch/csc

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

Electropedia - www.electropedia.org

Le premier dictionnaire d'électrotechnologie en ligne au monde, avec plus de 22 000 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 16 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

67 000 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.

INTERNATIONAL STANDARD

NORME INTERNATIONALE



Use case methodology –

Part 3: Definition of use case template artefacts into an XML serialized format

Méthodologie des cas d'utilisation –

**Partie 3: Définition des artefacts de formulaire type de modèle de cas
d'utilisation au format sérialisé XML**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 29.020

ISBN 978-2-8322-8820-7

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.....	9
INTRODUCTION.....	11
1 Scope.....	13
2 Normative references	14
3 Terms and definitions	14
4 Methodological framework for developing this standard	15
4.1 Model-based approach for developing the data exchange format	15
4.2 Example for the model-based process in the context of IEC 62559	17
4.3 Development of specific data exchange profiles.....	20
5 Application of the methodological framework for defining a set of key syntactic exchange formats (use cases, actors, requirements)	20
6 Models' detailed content (automatic generation from UML models).....	21
6.1 UML use case core concepts (UML Information Model).....	21
6.1.1 General	21
6.1.2 Model package UseCaseCoreConcepts	21
6.2 Use case library exchange format	47
6.2.1 UML use case library exchange model (UML Contextual Model)	47
6.2.2 Use case library XML syntactic format	75
6.3 Actor library exchange Format	93
6.3.1 UML actor library exchange model (UML Contextual Model)	93
6.3.2 Actor library XML syntactic format.....	96
6.4 Requirement library exchange Format.....	98
6.4.1 UML requirement library exchange model (UML Contextual Model)	98
6.4.2 Requirement library XML syntactic format.....	101
Bibliography.....	103
Figure 1 – IEC 62559 standard series	11
Figure 2 – A common XML format for importing/exporting use case information between a variety of modelling software and repositories.....	13
Figure 3 – Model-based development framework for the definition of the IEC 62559-3 standard XML-based data exchange format	16
Figure 4 – UML Information Model for the Scenario-Actor relationship example	17
Figure 5 – UML Contextual Model based on the Scenario-Actor relationship example.....	18
Figure 6 – XML Schema corresponding to the Scenario-Actor relationship example.....	19
Figure 7 – Producing key XML exchange formats based on a unique IEC 62559 UML Information Model.....	21
Figure 8 – Class diagram UseCaseCoreConcepts::UseCaseCore_Libraries	22
Figure 9 – Class diagram UseCaseCoreConcepts::UseCaseCore_ActorLibrary.....	22
Figure 10 – Class diagram UseCaseCoreConcepts::UseCaseCore_UseCaseLibrary.....	23
Figure 11 – Class diagram UseCaseCoreConcepts::UseCaseCore_DetailedActivities.....	23
Figure 12 – Class diagram UseCaseCoreConcepts::UseCaseCore_AreaLibrary	24
Figure 13 – Class diagram UseCaseCoreConcepts::UseCaseCore_BusinessCaseLibrary	24
Figure 14 – Class diagram UseCaseCoreConcepts::UseCaseCore_BusinessObjectLibrary	25

Figure 15 – Class diagram UseCaseCoreConcepts::UseCaseCore_CommonTermLibrary	25
Figure 16 – Class diagram UseCaseCoreConcepts::UseCaseCore_RequirementLibrary	25
Figure 17 – Class diagram Primitives::Primitives	46
Figure 18 – Package diagram UseCase_ExchangeModel::ContextualModel_Dependency	48
Figure 19 – Class diagram UseCase_ExchangeModel::UseCase_Repository	49
Figure 20 – Class diagram UseCase_ExchangeModel::UseCase_ShortDescription	50
Figure 21 – Class diagram UseCase_ExchangeModel::UseCase_DetailedActivities	51
Figure 22 – Class diagram UseCase_ExchangeModel::UseCase_CompleteDescription	52
Figure 23 – Class diagram ActorLibrary_ExchangeModel::ActorLibrary	93
Figure 24 – Package diagram ActorLibrary_ExchangeModel::ActorLibrary	94
Figure 25 – Class diagram RequirementLibrary_ExchangeModel::RequirementLibrary	98
Figure 26 – Package diagram RequirementLibrary_ExchangeModel::RequirementLibrary	99
Table 1 – Attributes of UseCaseCoreConcepts::Activity	26
Table 2 – Association ends of UseCaseCoreConcepts::Activity with other classes	27
Table 3 – Attributes of UseCaseCoreConcepts::Actor	27
Table 4 – Association ends of UseCaseCoreConcepts::Actor with other classes	28
Table 5 – Attributes of UseCaseCoreConcepts::ActorGrouping	28
Table 6 – Association ends of UseCaseCoreConcepts::ActorGrouping with other classes	28
Table 7 – Attributes of UseCaseCoreConcepts::ActorLibrary	29
Table 8 – Association ends of UseCaseCoreConcepts::ActorLibrary with other classes	29
Table 9 – Attributes of UseCaseCoreConcepts::Area	29
Table 10 – Association ends of UseCaseCoreConcepts::Area with other classes	29
Table 11 – Attributes of UseCaseCoreConcepts::AreaLibrary	30
Table 12 – Association ends of UseCaseCoreConcepts::AreaLibrary with other classes	30
Table 13 – Attributes of UseCaseCoreConcepts::Assumption	30
Table 14 – Association ends of UseCaseCoreConcepts::Assumption with other classes	30
Table 15 – Attributes of UseCaseCoreConcepts::Author	31
Table 16 – Association ends of UseCaseCoreConcepts::Author with other classes	31
Table 17 – Attributes of UseCaseCoreConcepts::BusinessCase	31
Table 18 – Association ends of UseCaseCoreConcepts::BusinessCase with other classes	31
Table 19 – Attributes of UseCaseCoreConcepts::BusinessCaseLibrary	32
Table 20 – Association ends of UseCaseCoreConcepts::BusinessCaseLibrary with other classes	32
Table 21 – Attributes of UseCaseCoreConcepts::BusinessObject	32
Table 22 – Association ends of UseCaseCoreConcepts::BusinessObject with other classes	32
Table 23 – Attributes of UseCaseCoreConcepts::BusinessObjectLibrary	33
Table 24 – Association ends of UseCaseCoreConcepts::BusinessObjectLibrary with other classes	33
Table 25 – Attributes of UseCaseCoreConcepts::CommonTerm	33

Table 26 – Association ends of UseCaseCoreConcepts::CommonTerm with other classes	33
Table 27 – Attributes of UseCaseCoreConcepts::CommonTermLibrary	34
Table 28 – Association ends of UseCaseCoreConcepts::CommonTermLibrary with other classes	34
Table 29 – Attributes of UseCaseCoreConcepts::Condition	34
Table 30 – Association ends of UseCaseCoreConcepts::Condition with other classes	34
Table 31 – Attributes of UseCaseCoreConcepts::CustomInformation	35
Table 32 – Association ends of UseCaseCoreConcepts::CustomInformation with other classes	35
Table 33 – Attributes of UseCaseCoreConcepts::Drawing	35
Table 34 – Association ends of UseCaseCoreConcepts::Drawing with other classes	36
Table 35 – Literals of UseCaseCoreConcepts::DrawingClassification	36
Table 36 – Attributes of UseCaseCoreConcepts::FurtherActorInformation	36
Table 37 – Association ends of UseCaseCoreConcepts::FurtherActorInformation with other classes	37
Table 38 – Attributes of UseCaseCoreConcepts::IdentifiedObject	37
Table 39 – Attributes of UseCaseCoreConcepts::KeyPerformanceIndicator	37
Table 40 – Association ends of UseCaseCoreConcepts::KeyPerformanceIndicator with other classes	37
Table 41 – Attributes of UseCaseCoreConcepts::Narrative	38
Table 42 – Association ends of UseCaseCoreConcepts::Narrative with other classes	38
Table 43 – Attributes of UseCaseCoreConcepts::Objective	38
Table 44 – Association ends of UseCaseCoreConcepts::Objective with other classes	39
Table 45 – Attributes of UseCaseCoreConcepts::Reference	39
Table 46 – Association ends of UseCaseCoreConcepts::Reference with other classes	39
Table 47 – Attributes of UseCaseCoreConcepts::Remark	40
Table 48 – Association ends of UseCaseCoreConcepts::Remark with other classes	40
Table 49 – Attributes of UseCaseCoreConcepts::Requirement	40
Table 50 – Association ends of UseCaseCoreConcepts::Requirement with other classes	40
Table 51 – Attributes of UseCaseCoreConcepts::RequirementCategory	41
Table 52 – Association ends of UseCaseCoreConcepts::RequirementCategory with other classes	41
Table 53 – Attributes of UseCaseCoreConcepts::RequirementLibrary	41
Table 54 – Association ends of UseCaseCoreConcepts::RequirementLibrary with other classes	41
Table 55 – Attributes of UseCaseCoreConcepts::Scenario	42
Table 56 – Association ends of UseCaseCoreConcepts::Scenario with other classes	42
Table 57 – Attributes of UseCaseCoreConcepts::TriggeringEvent	42
Table 58 – Association ends of UseCaseCoreConcepts::TriggeringEvent with other classes	42
Table 59 – Attributes of UseCaseCoreConcepts::UseCase	43
Table 60 – Association ends of UseCaseCoreConcepts::UseCase with other classes	44
Table 61 – Attributes of UseCaseCoreConcepts::UseCaseLibrary	44

Table 62 – Association ends of UseCaseCoreConcepts::UseCaseLibrary with other classes	45
Table 63 – Attributes of UseCaseCoreConcepts::UseCaseRepository	45
Table 64 – Association ends of UseCaseCoreConcepts::UseCaseRepository with other classes	45
Table 65 – Attributes of UseCaseCoreConcepts::Version	46
Table 66 – Association ends of UseCaseCoreConcepts::Version with other classes	46
Table 67 – Attributes of UseCase_ExchangeModel::Activity	53
Table 68 – Association ends of UseCase_ExchangeModel::Activity with other classes	54
Table 69 – Attributes of UseCase_ExchangeModel::Actor	54
Table 70 – Association ends of UseCase_ExchangeModel::Actor with other classes	54
Table 71 – Attributes of UseCase_ExchangeModel::ActorGrouping	55
Table 72 – Association ends of UseCase_ExchangeModel::ActorGrouping with other classes	55
Table 73 – Attributes of UseCase_ExchangeModel::ActorLibrary	55
Table 74 – Association ends of UseCase_ExchangeModel::ActorLibrary with other classes	55
Table 75 – Attributes of UseCase_ExchangeModel::Area	56
Table 76 – Association ends of UseCase_ExchangeModel::Area with other classes	56
Table 77 – Attributes of UseCase_ExchangeModel::AreaLibrary	56
Table 78 – Association ends of UseCase_ExchangeModel::AreaLibrary with other classes	56
Table 79 – Attributes of UseCase_ExchangeModel::Assumption	56
Table 80 – Association ends of UseCase_ExchangeModel::Assumption with other classes	57
Table 81 – Attributes of UseCase_ExchangeModel::Author	57
Table 82 – Association ends of UseCase_ExchangeModel::Author with other classes	57
Table 83 – Attributes of UseCase_ExchangeModel::BusinessCase	57
Table 84 – Association ends of UseCase_ExchangeModel::BusinessCase with other classes	58
Table 85 – Attributes of UseCase_ExchangeModel::BusinessCaseLibrary	58
Table 86 – Association ends of UseCase_ExchangeModel::BusinessCaseLibrary with other classes	58
Table 87 – Attributes of UseCase_ExchangeModel::BusinessObject	58
Table 88 – Association ends of UseCase_ExchangeModel::BusinessObject with other classes	59
Table 89 – Attributes of UseCase_ExchangeModel::BusinessObjectLibrary	59
Table 90 – Association ends of UseCase_ExchangeModel::BusinessObjectLibrary with other classes	59
Table 91 – Attributes of UseCase_ExchangeModel::CommonTerm	59
Table 92 – Association ends of UseCase_ExchangeModel::CommonTerm with other classes	60
Table 93 – Attributes of UseCase_ExchangeModel::CommonTermLibrary	60
Table 94 – Association ends of UseCase_ExchangeModel::CommonTermLibrary with other classes	60
Table 95 – Attributes of UseCase_ExchangeModel::Condition	60

Table 96 – Association ends of UseCase_ExchangeModel::Condition with other classes	60
Table 97 – Attributes of UseCase_ExchangeModel::CustomInformation	61
Table 98 – Association ends of UseCase_ExchangeModel::CustomInformation with other classes	61
Table 99 – Attributes of UseCase_ExchangeModel::Drawing	61
Table 100 – Association ends of UseCase_ExchangeModel::Drawing with other classes	62
Table 101 – Literals of UseCase_ExchangeModel::DrawingClassification	62
Table 102 – Attributes of UseCase_ExchangeModel::KeyPerformanceIndicator	62
Table 103 – Association ends of UseCase_ExchangeModel::KeyPerformanceIndicator with other classes	63
Table 104 – Attributes of UseCase_ExchangeModel::Narrative	63
Table 105 – Association ends of UseCase_ExchangeModel::Narrative with other classes	63
Table 106 – Attributes of UseCase_ExchangeModel::Objective	64
Table 107 – Association ends of UseCase_ExchangeModel::Objective with other classes	64
Table 108 – Attributes of UseCase_ExchangeModel::Ref_Actor	64
Table 109 – Association ends of UseCase_ExchangeModel::Ref_Actor with other classes	64
Table 110 – Attributes of UseCase_ExchangeModel::Ref_Area	65
Table 111 – Association ends of UseCase_ExchangeModel::Ref_Area with other classes	65
Table 112 – Attributes of UseCase_ExchangeModel::Ref_BusinessCase	65
Table 113 – Association ends of UseCase_ExchangeModel::Ref_BusinessCase with other classes	65
Table 114 – Attributes of UseCase_ExchangeModel::Ref_BusinessObject	65
Table 115 – Association ends of UseCase_ExchangeModel::Ref_BusinessObject with other classes	66
Table 116 – Attributes of UseCase_ExchangeModel::Ref_CommonTerm	66
Table 117 – Association ends of UseCase_ExchangeModel::Ref_CommonTerm with other classes	66
Table 118 – Attributes of UseCase_ExchangeModel::Ref_Objective	66
Table 119 – Association ends of UseCase_ExchangeModel::Ref_Objective with other classes	66
Table 120 – Attributes of UseCase_ExchangeModel::Ref_Requirement	67
Table 121 – Association ends of UseCase_ExchangeModel::Ref_Requirement with other classes	67
Table 122 – Attributes of UseCase_ExchangeModel::Ref_UseCase	67
Table 123 – Association ends of UseCase_ExchangeModel::Ref_UseCase with other classes	67
Table 124 – Attributes of UseCase_ExchangeModel::Reference	68
Table 125 – Association ends of UseCase_ExchangeModel::Reference with other classes	68
Table 126 – Attributes of UseCase_ExchangeModel::Remark	68
Table 127 – Association ends of UseCase_ExchangeModel::Remark with other classes	68
Table 128 – Attributes of UseCase_ExchangeModel::Requirement	69

Table 129 – Association ends of UseCase_ExchangeModel::Requirement with other classes	69
Table 130 – Attributes of UseCase_ExchangeModel::RequirementCategory	69
Table 131 – Association ends of UseCase_ExchangeModel::RequirementCategory with other classes	69
Table 132 – Attributes of UseCase_ExchangeModel::RequirementLibrary	70
Table 133 – Association ends of UseCase_ExchangeModel::RequirementLibrary with other classes	70
Table 134 – Attributes of UseCase_ExchangeModel::Resource_String	70
Table 135 – Literals of UseCase_ExchangeModel::ResourceType	70
Table 136 – Attributes of UseCase_ExchangeModel::Scenario	70
Table 137 – Association ends of UseCase_ExchangeModel::Scenario with other classes	71
Table 138 – Attributes of UseCase_ExchangeModel::TriggeringEvent	71
Table 139 – Association ends of UseCase_ExchangeModel::TriggeringEvent with other classes	71
Table 140 – Attributes of UseCase_ExchangeModel::UseCase	72
Table 141 – Association ends of UseCase_ExchangeModel::UseCase with other classes	72
Table 142 – Attributes of UseCase_ExchangeModel::UseCaseLibrary	73
Table 143 – Association ends of UseCase_ExchangeModel::UseCaseLibrary with other classes	73
Table 144 – Attributes of UseCase_ExchangeModel::UseCaseRepository	73
Table 145 – Association ends of UseCase_ExchangeModel::UseCaseRepository with other classes	74
Table 146 – Attributes of UseCase_ExchangeModel::Version	74
Table 147 – Association ends of UseCase_ExchangeModel::Version with other classes	74
Table 148 – Attributes of UseCase_ExchangeModel::FurtherActorInformation	75
Table 149 – Association ends of UseCase_ExchangeModel::FurtherActorInformation with other classes	75
Table 150 – Use case library XSD	75
Table 151 – Attributes of ActorLibrary_ExchangeModel::Actor	95
Table 152 – Association ends of ActorLibrary_ExchangeModel::Actor with other classes	95
Table 153 – Attributes of ActorLibrary_ExchangeModel::ActorLibrary	95
Table 154 – Association ends of ActorLibrary_ExchangeModel::ActorLibrary with other classes	95
Table 155 – Attributes of ActorLibrary_ExchangeModel::Ref_Actor	96
Table 156 – Association ends of ActorLibrary_ExchangeModel::Ref_Actor with other classes	96
Table 157 – Association ends of ActorLibrary_ExchangeModel:: UseCaseRepository with other classes	96
Table 158 – Actor library XSD	96
Table 159 – Attributes of RequirementLibrary_ExchangeModel::RequirementLibrary	99
Table 160 – Association ends of RequirementLibrary_ExchangeModel::RequirementLibrary with other classes	100
Table 161 – Attributes of RequirementLibrary_ExchangeModel::RequirementCategory	100

Table 162 – Association ends of RequirementLibrary_ExchangeModel::RequirementCategory with other classes	100
Table 163 – Attributes of RequirementLibrary_ExchangeModel::Ref_RequirementCategory	100
Table 164 – Association ends of RequirementLibrary_ExchangeModel::Ref_RequirementCategory with other classes	101
Table 165 – Attributes of RequirementLibrary_ExchangeModel::Requirement	101
Table 166 – Association ends of RequirementLibrary_ExchangeModel::Requirement with other classes	101
Table 167 – Requirement library XSD	101

IECNORM.COM : Click to view the full PDF of IEC 62559-3:2017

INTERNATIONAL ELECTROTECHNICAL COMMISSION

USE CASE METHODOLOGY –

**Part 3: Definition of use case template artefacts
into an XML serialized format**

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 62559-3 has been prepared by IEC systems committee Smart Energy.

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

CDV	Report on voting
SyCSmartEnergy/28/CDV	SyCSmartEnergy/48/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 document has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts in the IEC 62559 series, published under the general title *Use case methodology*, can be found on the IEC website.

This IEC standard includes Code Components, i.e. components that are intended to be directly processed by a computer. Such content is any text found between the markers <CODE BEGINS> and <CODE ENDS>, or otherwise is clearly labelled in this standard as a Code Component.

The purchase of this IEC standard carries a copyright license for the purchaser to sell software containing Code Components from this standard directly to end users and to end users via distributors, subject to IEC software licensing conditions, which can be found at: <http://www.iec.ch/webstore/custserv/pdf/CCv1.pdf>

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

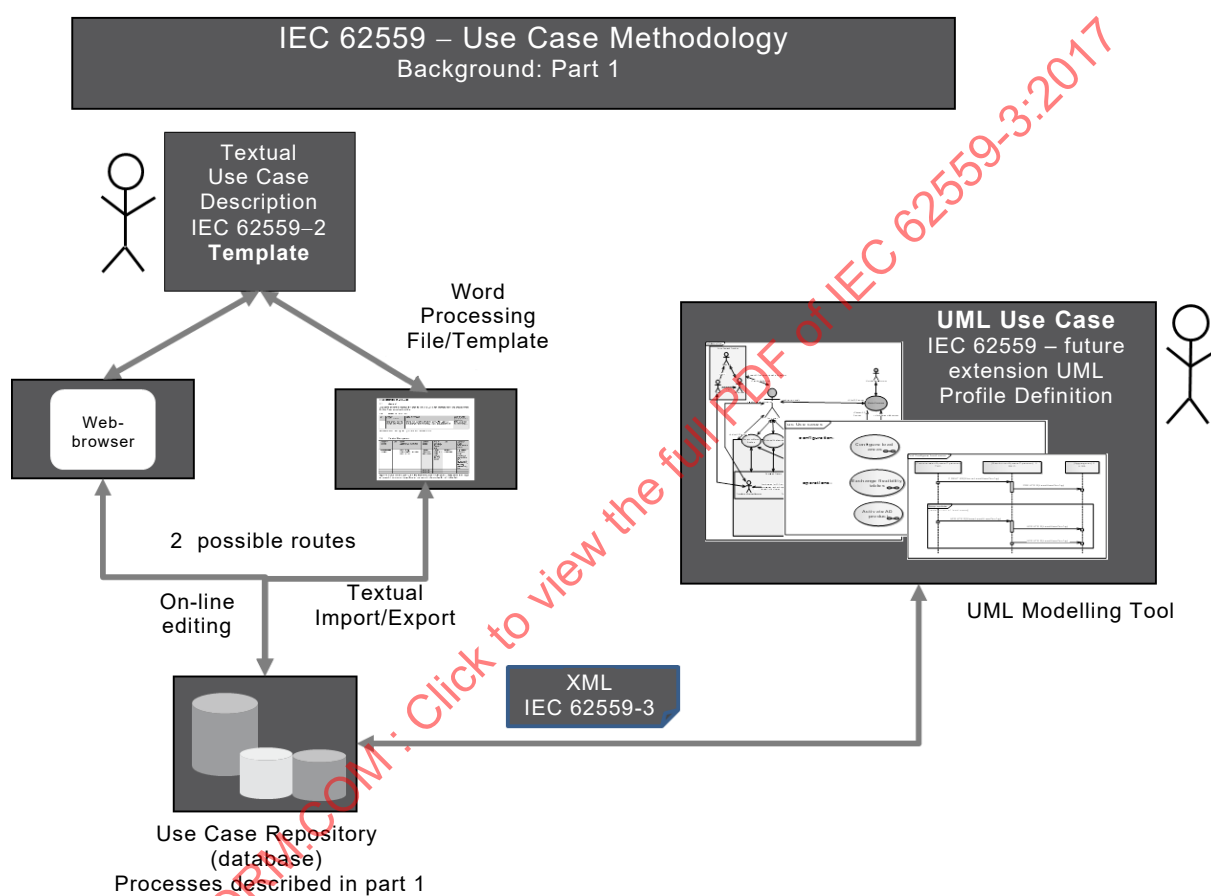
- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

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.

INTRODUCTION

For complex systems, the use case methodology supports a common understanding of functionalities, Actors and processes across different technical committees or even different organizations. Developed as software engineering tool, the methodology can be used to support the development of standards as it facilitates the analysis of requirements in relation to new or existing standards. Further arguments for the use case methodology and background information are available in IEC 62559-1.

Figure 1 provides an overview of the intended first parts of the IEC 62559 series, mainly describing the relation between IEC 62559-2 and IEC 62559-3.



IEC

Figure 1 – IEC 62559 standard series

IEC 62559-1 – Concept and processes in standardization

IEC 62559-1 will be the basis for a common use case repository in order to gather use cases within IEC on a common collaborative platform. This repository will also be used to organize a harmonization of use cases in order to provide broadly accepted generic use cases as a basis for further standardization work. It describes processes and provides basics for the use case methodology like terms or use case types.

IEC 62559-2 – Definition of the templates for use cases, Actor list and requirements list

IEC 62559-2 defines the structure of a use case template, an Actor list and a list for requirements. The document is mainly based on IEC PAS 62559:2008 and shall be read together with IEC 62559-1.

IEC 62559-3 – Definition of use case template artefacts into an XML serialized format

Based on IEC 62559-2, this document defines the required core concepts and their serialization into an XML format of a use case template, an Actor list and a list for detailed requirements. The XML format is used to transfer the content of the template to other engineering systems (e.g. UML modelling tools). These documents are developed using the energy system and Smart Grids as examples, but they are general enough to be transferred to other domains and systems. It is intended to develop a UML profile definition based on this part in the future.

The IEC 62559 series is needed to fulfil the SG3 decision 7 made by the SMB at its February 2010 meeting (SMB/4204/DL, Decision 137/10) requesting the urgent delivery of a generic use case repository for all Smart Grid applications. Nevertheless, the use case methodology described in this document is intended for a broader application within standardization exceeding Smart Grid systems.

More and more complex systems such as Smart Grids or Smart Cities are raising the question of managing system level requirements, which have to be fed by many domains of expertise (in standardization related to different Technical Committees (TCs)), and which have to be broken down further and shared by many TCs in charge of specifying standards to support these system level functions.

One way to handle this transversality efficiently is to set some common methods and terms. The use case methodology is the current state of the art and supports further engineering activities.

The use case methodology offers a unique way for sharing ideas and requirements of new use cases or business cases between many experts/TCs with different backgrounds: for example, domain experts with knowledge about energy systems or business processes on one hand and system-/IT-experts defining exchanged information and communication on the other hand. In the requirement development process, domain experts provide general ideas and functional requirements. The main goal is for system experts to detail down these use cases to a level where they can be used to specify interfaces, dedicated functionality, data and service model exchange. However, safety- or EMC-experts (as examples) may also make use of the described use cases, their terminology and identified requirements.

However the starting point is to set up a framework for consistency within IEC, helping IEC members to provide use cases in a consistent manner – IEC 62559 serves as a basis for use case repositories in order to gather, administrate, maintain, and evaluate use cases.

Within IEC, a use case repository serves as common collaborative platform for use case elaboration and to organize a harmonization of use cases in order to provide broadly accepted generic use cases as a basis for further standardization work.

But the use case template defined in this document may serve not only for the development of standards, but also – as was the original purpose of IEC PAS 62559:2008 – as a helpful means for the realization of projects within the area of complex systems. Also other applications, which need the benefits of a structured requirements development and formalized description of functionality, may make use of the suggested template.

The use case methodology is seen as a process which starts with the definition of business ideas, goals and requirements, detailing these in use case descriptions. This information can be used as a basis to identify/link reference architectures describing the types of components used, and going further down to an analysis for the further standardization process.

Further developments regarding the use case template are expected. These developments are mainly related to information, which is required in the use case description for further analysis, and which can be mapped to other information (e.g. to a reference architecture, IT security methods, standards and data models). Partly this is considered in the suggested template of IEC 62559-2. Further relations will be described separately as they are still under development and they can be considered for the further development of the IEC use case repository.

USE CASE METHODOLOGY –

Part 3: Definition of use case template artefacts into an XML serialized format

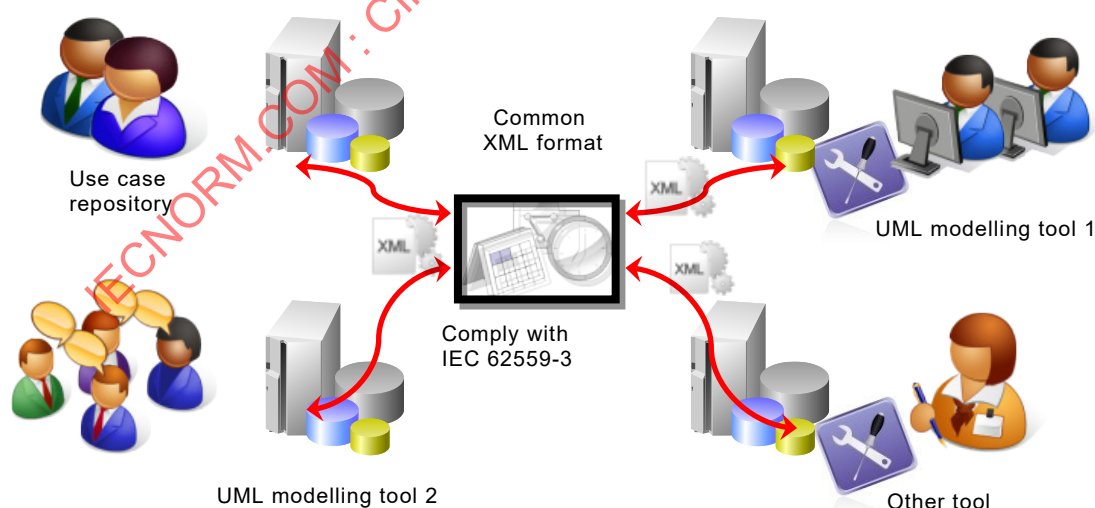
1 Scope

In order to exchange use cases based on the template which is defined in IEC 62559-2, this part of IEC 62559 establishes the interfaces between the different use case repositories and/or UML engineering software tools.

Therefore, this document defines the required core concepts and their serialization into XML syntactic format of a use case template, an Actor list and list for detailed requirements.

As shown in Figure 2, the modelling approach is leveraging the use of UML in order to graphically represent the data contained in a use case based on the IEC 62559 template. Therefore the textual format of the use case template may be in the use case development process just a starting point for business experts or an easy way to modify use case data for non UML experts. As a consequence, it is important for the IEC 62559 series to provide a reliable way for converting this textual format into UML format and reciprocally. As soon as a use case repository is maintained based on the IEC 62559 series, another related need is to be able to import/export between different UML tools and different use case repositories the use case related information based on a tool independent format.

The main purpose of this document is to propose an independent format for transferring the use case information between modelling software. In order to satisfy this goal, the syntactic XML format is chosen to serialize the use case data. This document defines in detail the core concepts of the template into UML and their transformations into XML using the XSD standard.



IEC

**Figure 2 – A common XML format for importing/exporting use case information
between a variety of modelling software and repositories**

This document, as shown in Figure 2, supports the interoperability between modelling software used by different companies over the world.

Once this level of interoperability is achieved, IEC 62559 can provide a reliable mechanism to interpret those XML data in order to represent graphically UML use cases. This need will be covered as well in a future part of IEC 62559 (to be defined).

This document focuses on a methodological framework which is also used by IEC TC 57 standards and which is summarized in Clause 4.

In order to exchange use cases based on the template which is defined in IEC 62559-2, this document establishes the interfaces between the different use case repositories and/or UML tools.

2 Normative references

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

IEC 62559-2, *Use case methodology – Part 2: Definition of the templates for use cases, actor list and requirements list*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62559-2 and the following apply.

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

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

3.1

XML

eXtensible Markup Language

markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable

Note 1 to entry: The XML specification is produced by the World Wide Web Consortium (W3C).

3.2

XSD

XML Schema Definition Language

language used to define the structure, content, and semantics of XML files

Note 1 to entry: XSD is a recommendation by the World Wide Web Consortium (W3C).

4 Methodological framework for developing this standard

4.1 Model-based approach for developing the data exchange format

This document relies on a model-based approach for the definition of exchange formats and therefore defines respective UML models. The approach utilized herein has been developed in IEC TC 57 to develop specific data models (profiles) and is actively used for developing the Common Information Model (CIM) standards of IEC TC 57 for the electricity system communications. Within the approach several types of models, their interrelationships, and rules to derive more specific models from a generic model are described. As the approach is explained in detail in IEC 62325-450, 4.1 only briefly describes the approach and its adaptations to create the framework on which this document is built.

Using the approach enables several XML-based exchange formats to be defined for a particular subject. However, all exchange formats created according to this process are based on a single source model ensuring semantic compatibility. Two interrelated model types are used for different purposes, which play a key role in this process, labelled as Information Model and Contextual Model, respectively. Figure 3 outlines these models in the context of the data exchange format definition process.

The Information Model represents all common core concepts of a use case that are identified in IEC 62559-2 on a rather generic level, including entities, their attributes, and relations. This comprises for instance a use case itself, scenarios of a use case, involved Actors, or information objects exchanged throughout a use case. Concepts defined in the Information Model define the semantics of the data exchange. Consequently, the Information Model provides a “single point of truth” regarding the structuring of use case information and can be seen as strictly normative.

A Contextual Model is always based on the Information Model. Contextual Models are created to cover specific needs of information exchange with regard to the Information Model. Contextual Models include a context-dependent subset of (or even the complete) Information Model that can be further constrained or specialized.

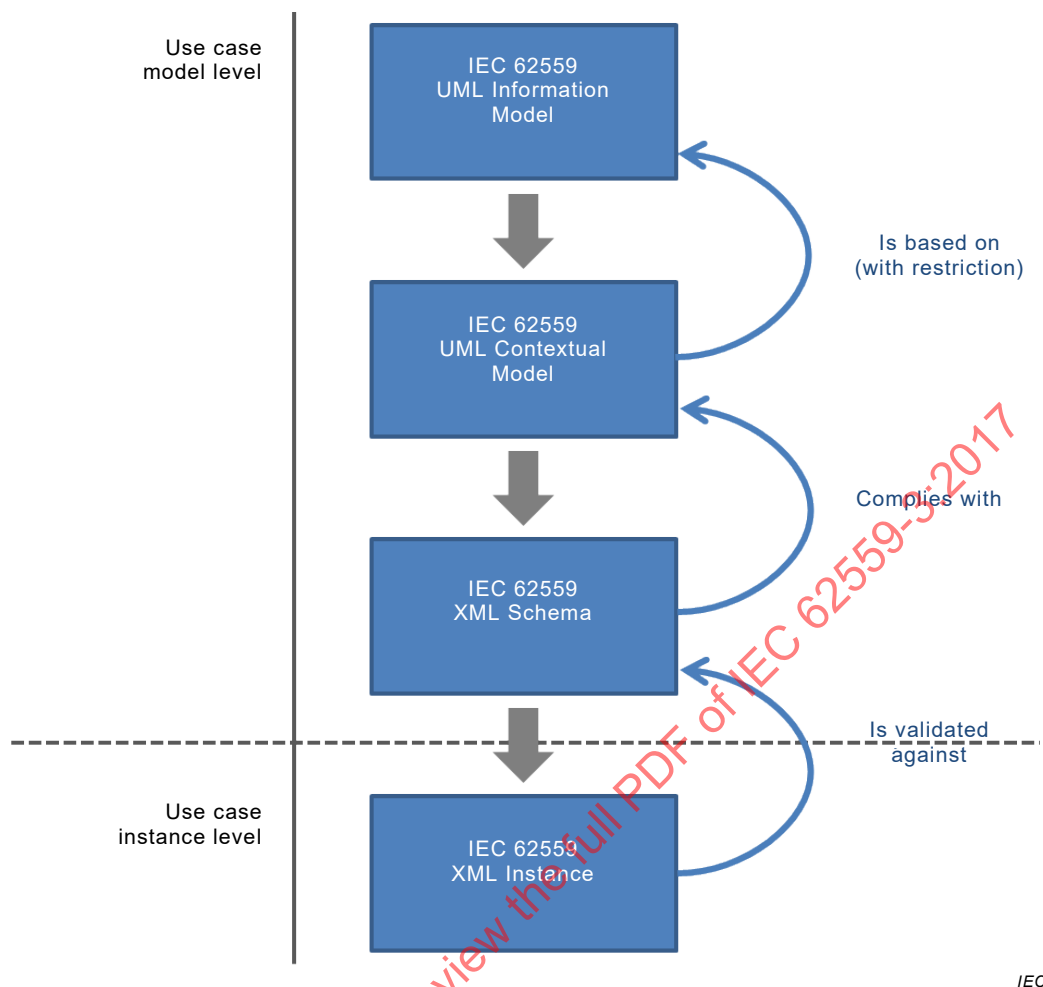


Figure 3 – Model-based development framework for the definition of the IEC 62559-3 standard XML-based data exchange format

Finally further rules can be applied to a Contextual Model creating the final exchange format that then inherently complies with the Contextual Model. In the context of this document, this is an XML schema that serves as the basis for a validation of exchanged use case data. For this purpose, use case instance data shall be defined in an XML document.

The model-based process outlined above was applied to develop respective XML-based data exchange formats. In order to finally produce a standardized XML schema definition (XSD) file according to IEC 62559-2, the model-based approach is carried out in detail as follows.

a) IEC 62559 UML Information Model:

This UML gathers the core concepts described in IEC 62559-2. This is modelled using UML class diagrams including classes, attributes, data types, and associations. This Information Model contains the core concepts of a use case modelled in a generic way in order to fit other reuses of those concepts in other implementations. For example, this Information Model could also be used to define the structure of a data base for use cases. Therefore, this Information Model is made flexible by modelling concepts with optional aspects. For example, many attributes and associations are modelled with flexible cardinalities such as 0..1 for attributes and 0..* for relationships. In order to create an implementation of this generic Information Model, it is necessary to define more precisely how the core concepts are used for the specific purpose of the exchange of use case information.

b) IEC 62559 UML Contextual Model:

This second UML model is based on the UML Information Model by modelling restrictions in order to adapt it for the specific purpose of defining a model specifying the exchange of use case information. Those restrictions set on the Information Model concepts make this Contextual Model accurate for the exchange of use case information. The allowed restrictions are the following ones:

- 1) Restrictions on cardinalities for attributes and associations by narrowing them. For example, an attribute cardinality 0..1 can be set as 1..1 (mandatory attribute) or 0..1 (optional attribute) or 0..0 (not used attribute). An association end cardinality 0..* can be set to an included interval such as 2..6 (sequence of instanced concepts at minimum 2 and maximum 6).
 - 2) Restrictions on attribute value space by adding facets. For example, an attribute typed by a String can have a facet of length 6, meaning that the expected attribute value is a sequence of 6 symbols.
- c) IEC 62559 XML schema:

This XML schema definition (XSD) is automatically generated from the previously developed UML Contextual Model. Every XML use case information exchange shall be validated against this XSD. It thus ensures interoperability between modelling software because the use case information exchange must respect the structure defined in the XSD.

4.2 Example for the model-based process in the context of IEC 62559

Subclause 4.2 illustrates an example based on the relationship between a use case scenario and an Actor playing the PrimaryActor role in this scenario.

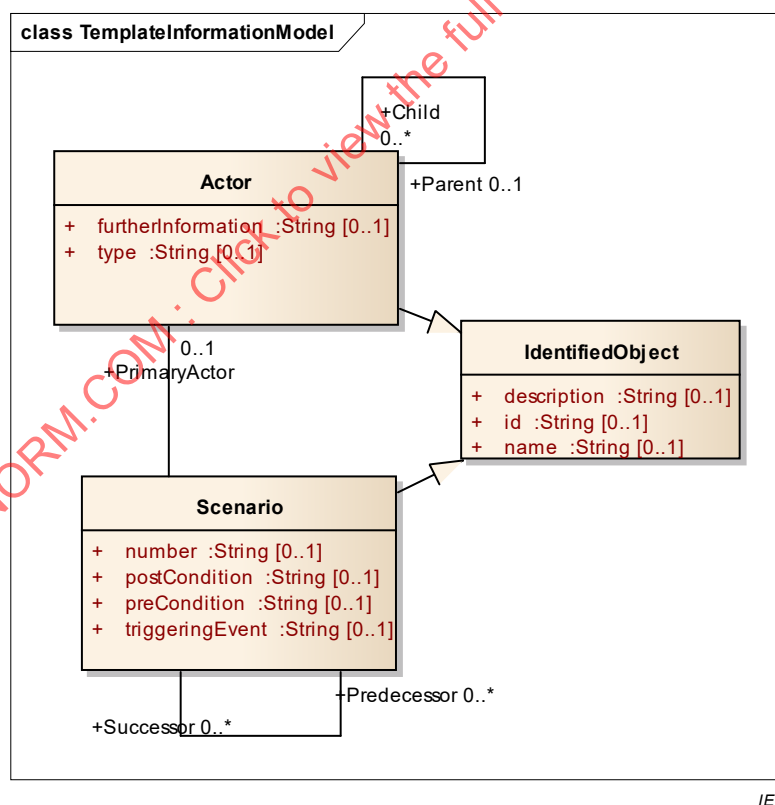


Figure 4 – UML Information Model for the Scenario-Actor relationship example

Figure 4 shows the UML Information Model representing the two core concepts Scenario and Actor (using UML classes) linked with a UML association having the role PrimaryActor and 0..1 cardinality on the Actor side. It means that a scenario can have an optional primary Actor. Supplementary information for both scenario and Actor is modelled with attributes of those two classes. For example, a scenario has a number, a postCondition, a precondition and a triggeringEvent. Because those two concepts have common needs for representing an identifier, a generalization class IdentifiedObject shares those common attributes such as the id, name and description attributes.

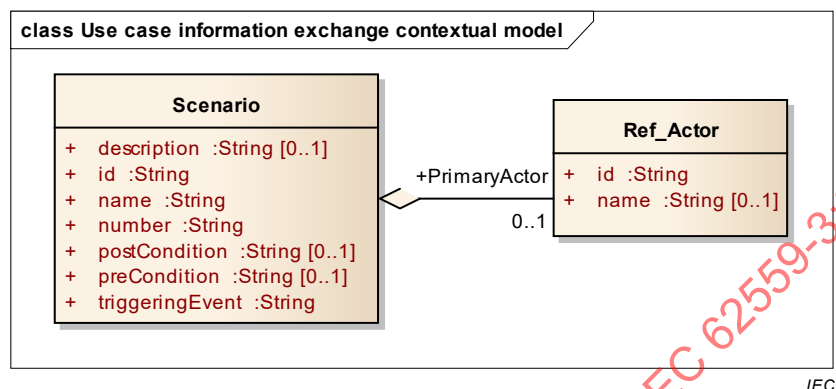


Figure 5 – UML Contextual Model based on the Scenario-Actor relationship example

In the UML Contextual Model (see Figure 5), the two concepts Scenario and Actor are reused and customized by adding restrictions on the Information Model for the specific purpose of exchanging information about a Scenario having a PrimaryActor. In this example, we make some attributes mandatory. Thus for Scenario, the identifier (id), the name, the number and triggeringEvent are mandatory information for the exchange. Moreover, if the information of a PrimaryActor is given, then its identifier is also mandatory. The concept Actor is reused several times with different defined structures in the same Contextual Model. In order to have a unique contextual name definition for those different structures, a qualifier is added to the Information Model class name. In this example, the contextual class based on the Actor concept is named Ref_Actor. When building the Contextual Model, the association between Scenario and Actor is restricted and becomes an aggregation, meaning that in the exchange of data, the scenario information includes information about a PrimaryActor.

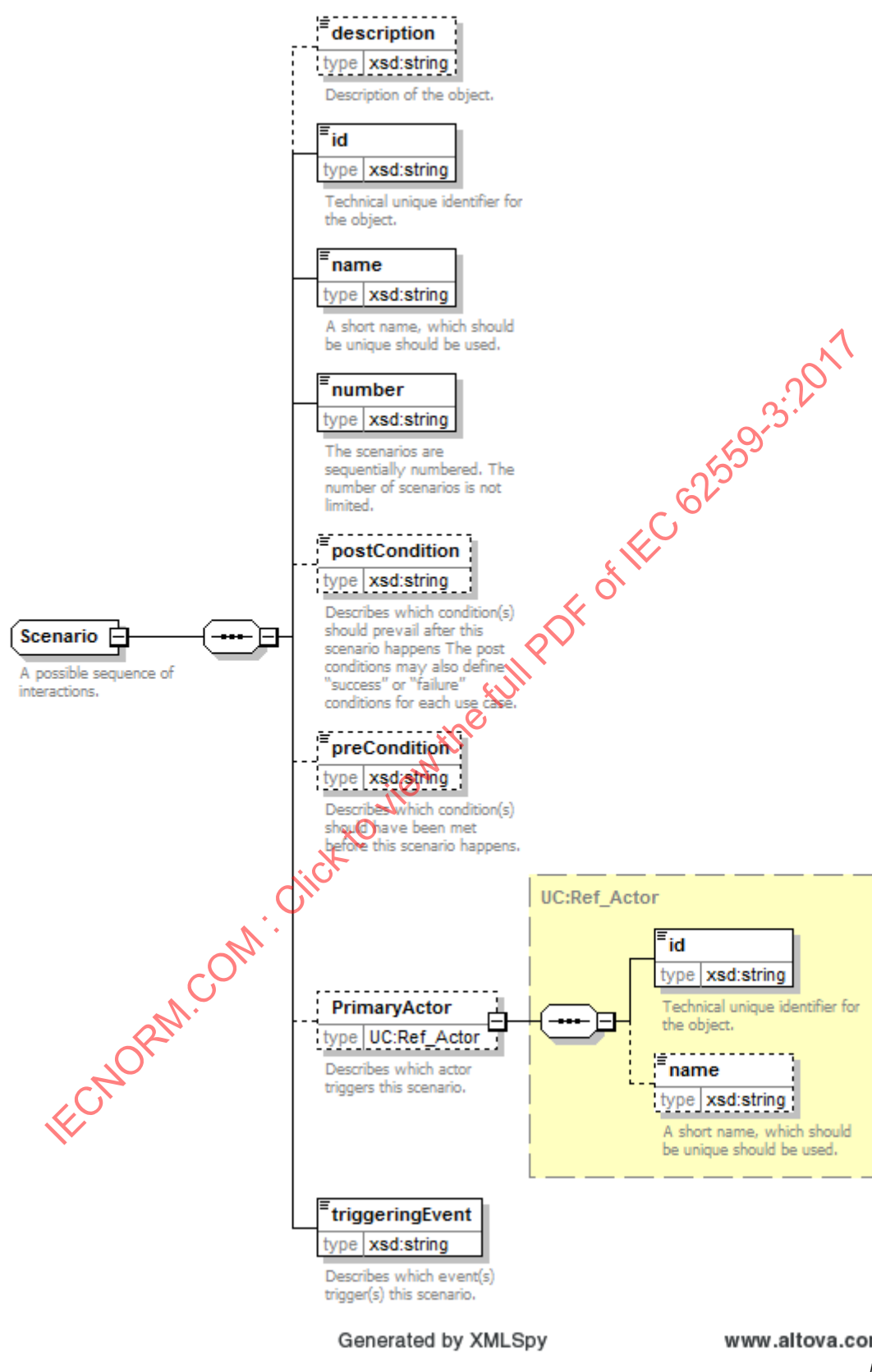


Figure 6 – XML Schema corresponding to the Scenario-Actor relationship example

Figure 6 illustrates graphically the XSD code which is automatically generated from the UML Contextual Model shown in Figure 5.

4.3 Development of specific data exchange profiles

Applying the model-based process described above can result in multiple Contextual Models and accordingly multiple different exchange formats, though usually with the same semantics. This process of deriving a specific model, in terms of a subset of model elements, from a generic model and specializing model elements, is referred to as profiling; the result of the profiling process is called a profile. As multiple profiles may be created based on a single Information Model, not all profiles can be considered normative. However, when data between two applications shall be successfully exchanged, there needs to be a (normative) agreement on a specific exchange format, not only semantically, but also syntactically.

Therefore, this document identifies key scenarios in regard to information exchange resulting in respective profiles that are normative and can be implemented in software. Key information exchange scenarios currently identified include the exchange of all concepts of the Information Model (e.g. for exchanging information between use case repositories), the exchange of a set of Actors (role model), and the exchange of requirements (see Clause 5). While the first data exchange scenario is rather generic, the latter ones enable a more thorough validation of the data in comparison to the exchange model covering an entire repository, as certain restrictions can be assumed, like the presence of an Actor or a requirement.

The normative data exchange format of the first generic exchange scenario can be found in Clause 6. Future editions of IEC 62559-3 may add further standardized information exchange scenarios and respective profiles, like for use cases. In case of specific needs, however, non-normative exchange formats/profiles can be developed using the model-based process.

5 Application of the methodological framework for defining a set of key syntactic exchange formats (use cases, actors, requirements)

With respect to different types of data exchange depending on specific key data exchange scenarios, IEC 62559-3 is intended to include a set of key data exchange formats for use cases, Actors, and requirements in particular. In order to produce those XML syntactic exchange formats, the modelling process described in Clause 4 is applied for designing each exchange format. Figure 7 provides an overview of the profiling of the generic Information Model.

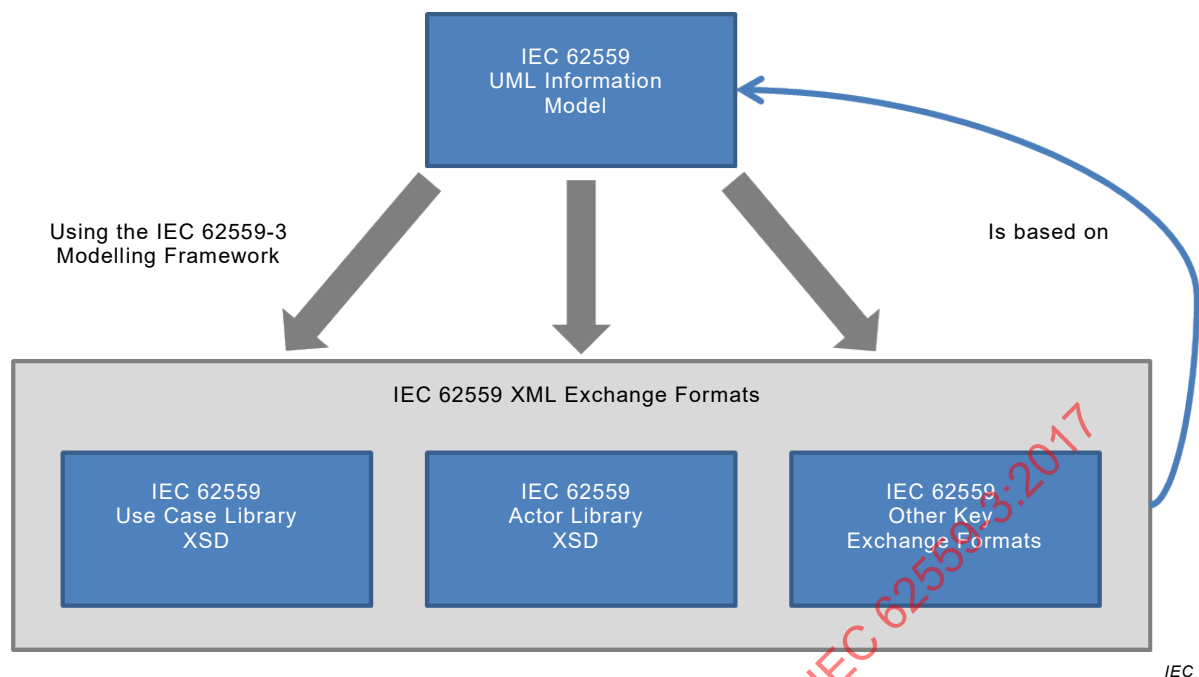


Figure 7 – Producing key XML exchange formats based on a unique IEC 62559 UML Information Model

6 Models' detailed content (automatic generation from UML models)

6.1 UML use case core concepts (UML Information Model)

6.1.1 General

The use case core concepts are described through a UML Information Model which is shared among the various defined exchange formats in this document. The core concepts are classified into a set of UML packages.

6.1.2 Model package UseCaseCoreConcepts

6.1.2.1 General

IEC 62559 UML Information Model for the use case core concepts.

Figure 8 shows class diagram UseCaseCore_Libraries.

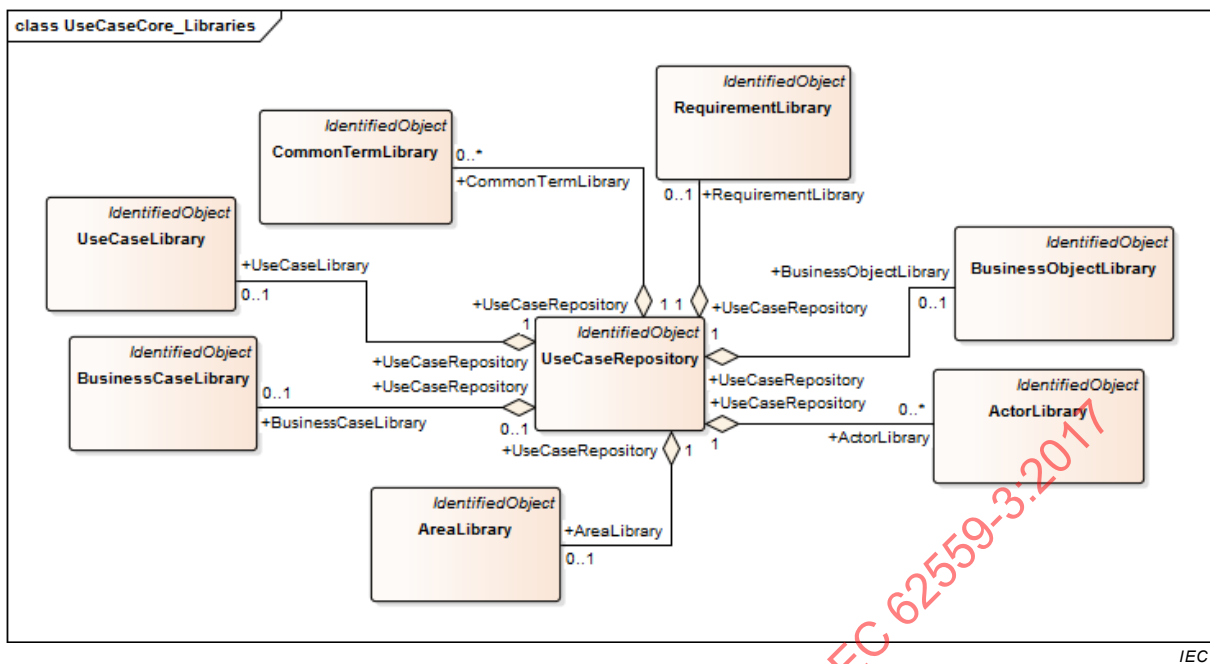


Figure 8 – Class diagram UseCaseCoreConcepts::UseCaseCore_Libraries

Figure 9 shows class diagram UseCaseCore_ActorLibrary.

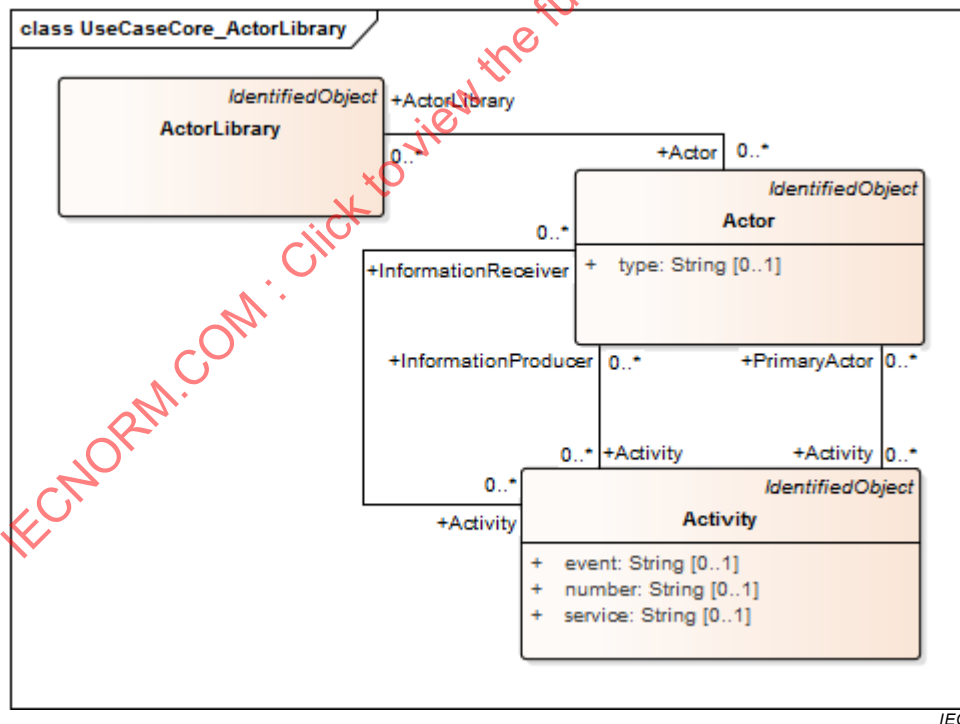


Figure 9 – Class diagram UseCaseCoreConcepts::UseCaseCore_ActorLibrary

Figure 10 shows class diagram UseCaseCore_UseCaseLibrary.

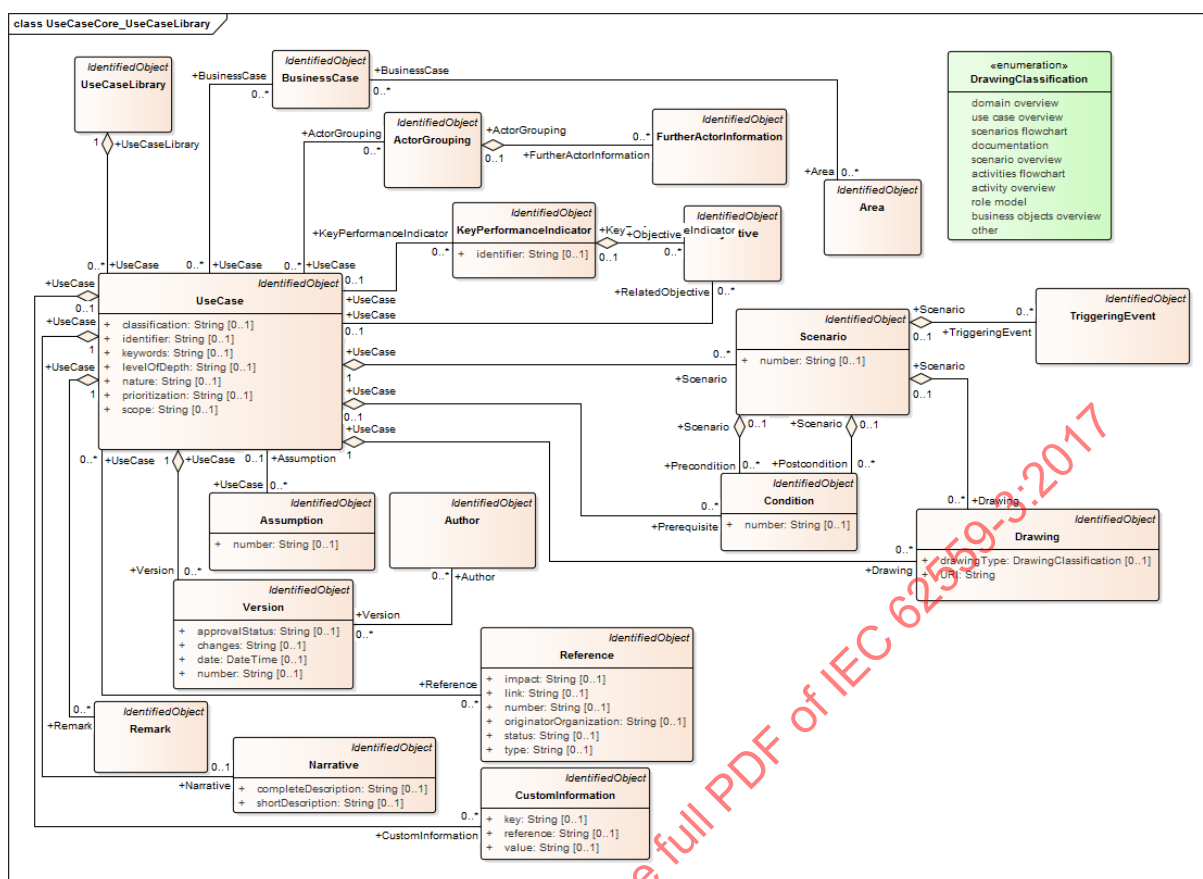


Figure 10 – Class diagram UseCaseCoreConcepts::UseCaseCore_UseCaseLibrary

Figure 11 shows class diagram UseCaseCore DetailedActivities.

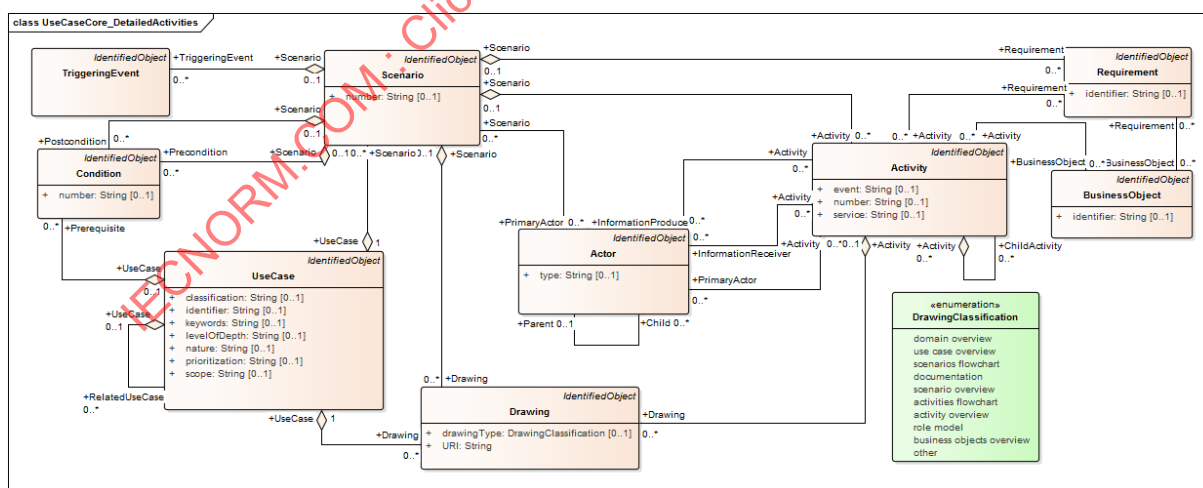


Figure 11 – Class diagram UseCaseCoreConcepts::UseCaseCore_DetailedActivities

Figure 12 shows class diagram UseCaseCore AreaLibrary.

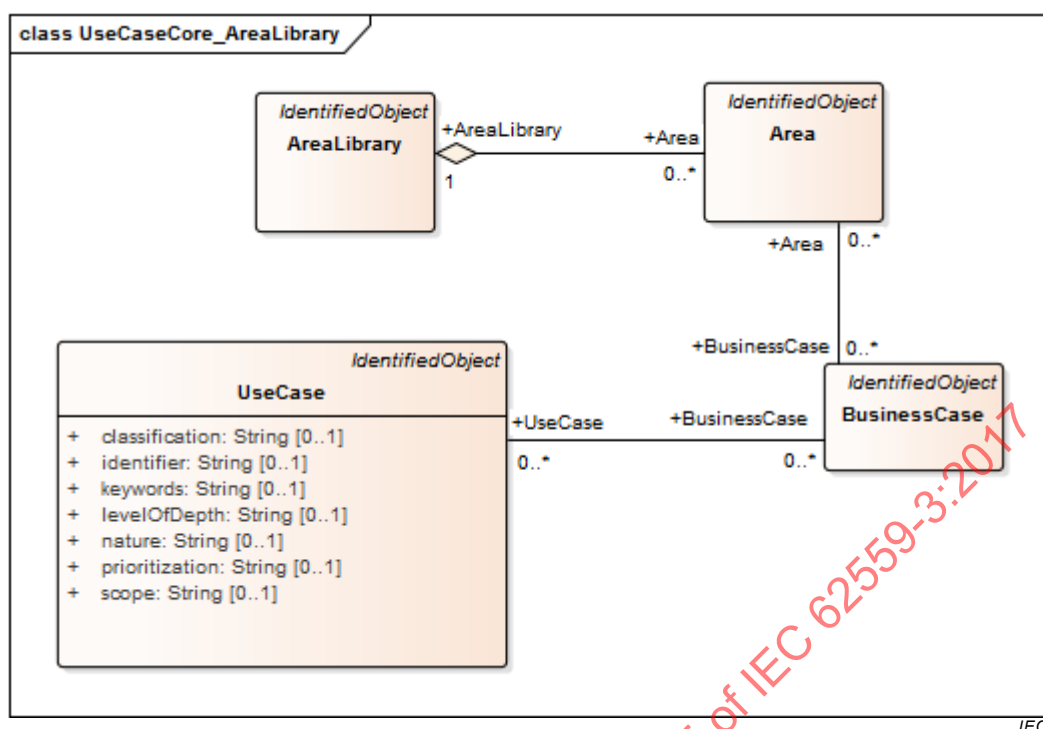


Figure 12 – Class diagram UseCaseCoreConcepts::UseCaseCore_AreaLibrary

Figure 13 shows class diagram UseCaseCore_BusinessCaseLibrary.

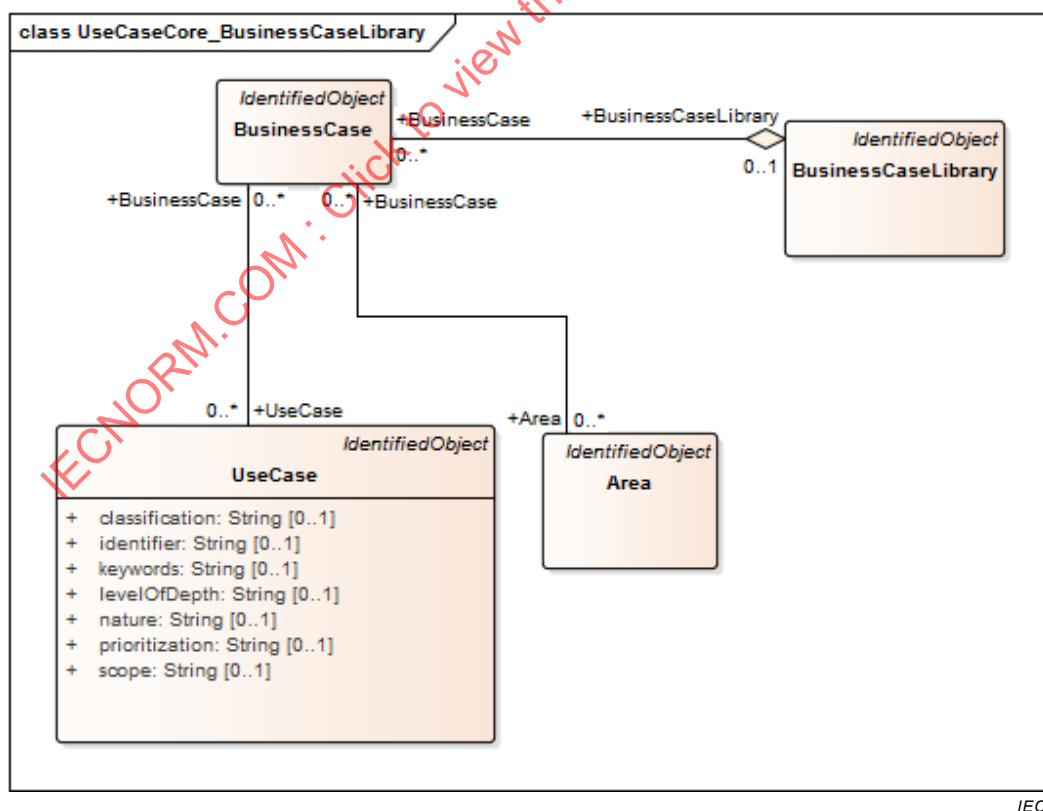
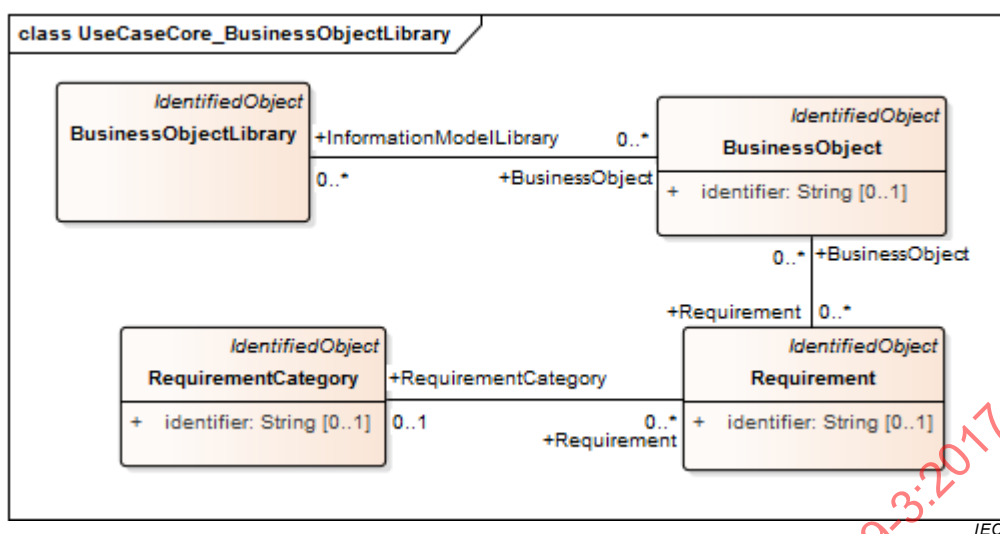


Figure 13 – Class diagram UseCaseCoreConcepts::UseCaseCore_BusinessCaseLibrary

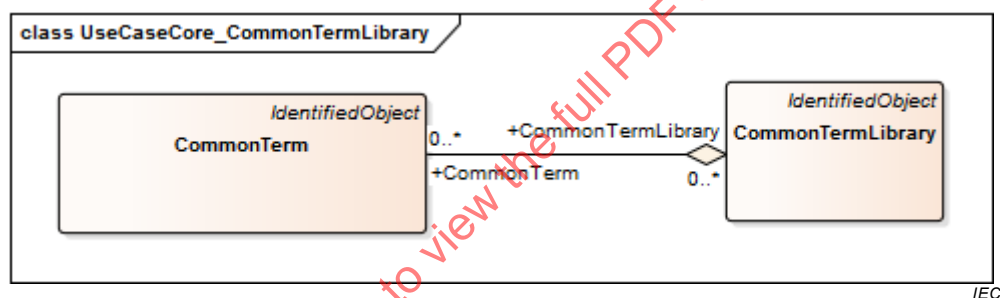
Figure 14 shows class diagram UseCaseCore_BusinessObjectLibrary.



IEC

Figure 14 – Class diagram
UseCaseCoreConcepts::UseCaseCore_BusinessObjectLibrary

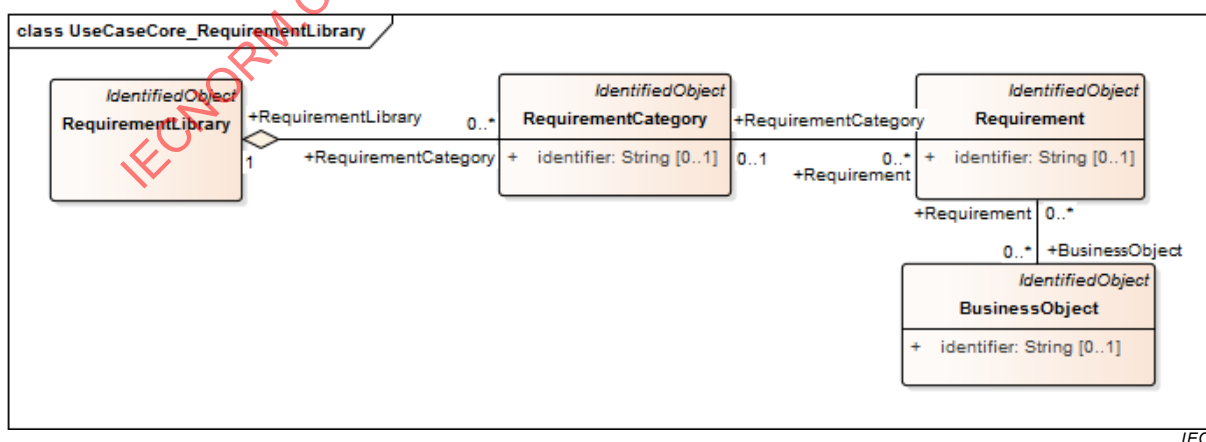
Figure 15 shows class diagram **UseCaseCore_CommonTermLibrary**.



IEC

Figure 15 – Class diagram **UseCaseCoreConcepts::UseCaseCore_CommonTermLibrary**

Figure 16 shows class diagram **UseCaseCore_RequirementLibrary**.



IEC

Figure 16 – Class diagram **UseCaseCoreConcepts::UseCaseCore_RequirementLibrary**

6.1.2.2 Activity

It may be subdivided into other activities or represent an elementary step of the scenario.

Table 1 shows all attributes of Activity.

Table 1 – Attributes of UseCaseCoreConcepts::Activity

name	type	description
event	String	It is the event that triggers the step. This might be completion of previous steps (in that case the event is the comma-separated list of their numbers) or the condition to exit a loop.
number	String	index of the activity
service	String	This column identifies the nature of the information flow and the originator of the information. Available options are CREATE, GET, CHANGE, DELETE, CANCEL, EXECUTE derived from IEC 61968-100:2013. Additionally, REPORT, TIMER and REPEAT are suggested. – CREATE means that an information object is to be created at the Producer. – GET (this is the default value if none is populated) means that the Receiver requests information from the Producer (default). – CHANGE means that information is to be updated. Producer updates the Receiver's information. – DELETE means that information is to be deleted. Producer deletes information from the Receiver. – CANCEL, CLOSE imply actions related to processes, such as the closure of a work order or the cancellation of a control request. – EXECUTE is used when a complex transaction is being conveyed using a service, which potentially contains more than one verb. – REPORT is used to represent transferral of unsolicited information or asynchronous information flows. Producer provides information to the Receiver. – TIMER is used to represent a waiting period. When using the TIMER service, the Information Producer and Information Receiver fields shall refer to the same Actor. – REPEAT is used to indicate that a series of steps is repeated until a condition or trigger event. The condition is specified as the text in the "Event" column for this row or step. Following the word REPEAT, the first and last step numbers of the series to be repeated shall appear, in parenthesis, in the following form: REPEAT(X-Y) where X is the first step and Y is the last step. These common service definitions are related to automation/information or communication systems. In case the use case template is applied in other domains, further services might be used and described.
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 2 shows all association ends of Activity with other classes.

Table 2 – Association ends of UseCaseCoreConcepts::Activity with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] Requirement	Requirement	Several requirements may be listed in one step, comma separated.
[0..*]	[0..*] InformationReceiver	Actor	This identifies the receiver of the information. This should be one of the Actors defined in the repository.
[0..*]	[0..*] InformationProducer	Actor	This identifies the producer or source of the information. This should be one of the Actors defined in the repository.
[0..*]	[0..*] BusinessObject	BusinessObject	This describes briefly the information to be exchanged between the two Actors – information producer and information receiver. It may list several information items in one step, comma separated.
[0..1]	[0..*] Drawing	Drawing	the list of associated drawing
[0..*]	[0..*] PrimaryActor	Actor	reference to the PrimaryActor
[0..*]	[0..*] ChildActivity	Activity	child activities
[0..*]	[0..1] Scenario	Scenario	reference to the scenario
[0..*]	[0..*] Activity	Activity	parent activities

6.1.2.3 Actor

It is the entity that communicates and interacts.

These Actors can include people, software applications, systems, databases, and even the power system itself.

Table 3 shows all attributes of Actor.

Table 3 – Attributes of UseCaseCoreConcepts::Actor

name	type	description
type	String	type of the Actor (business, system, ...)
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 4 shows all association ends of Actor with other classes.

Table 4 – Association ends of UseCaseCoreConcepts::Actor with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] Child	Actor	list of specialized Actors
[0..*]	[0..*] Activity	Activity	the activities related to the Actor
[0..*]	[0..*] Activity	Activity	the activities related to this Actor
[0..*]	[0..*] Activity	Activity	An Actor can be linked to an activity.
[0..*]	[0..*] Scenario	Scenario	The Actor is linked to scenarios.
[0..*]	[0..*] ActorLibrary	ActorLibrary	The Actor is referenced by libraries.
[0..*]	[0..*] UseCase	UseCase	list of use cases in relation with an Actor
[0..*]	[0..*] FurtherActorInformation	FurtherActorInformation	An Actor may have a list of further information shared between different groupings for different use cases.
[0..*]	[0..1] Parent	Actor	the generalized Actor

6.1.2.4 ActorGrouping

Group of Actors used to organize an Actor list

Table 5 shows all attributes of ActorGrouping.

Table 5 – Attributes of UseCaseCoreConcepts::ActorGrouping

name	type	description
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 6 shows all association ends of ActorGrouping with other classes.

Table 6 – Association ends of UseCaseCoreConcepts::ActorGrouping with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] UseCase	UseCase	ActorGrouping is associated with several use cases.
[0..1]	[0..*] FurtherActorInformation	FurtherActorInformation	In a context of one use case, an Actor grouped into an ActorGrouping may have one FurtherActorInformation.

6.1.2.5 ActorLibrary

Actors of the repository

Table 7 shows all attributes of ActorLibrary.

Table 7 – Attributes of UseCaseCoreConcepts::ActorLibrary

name	type	description
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 8 shows all association ends of ActorLibrary with other classes.

Table 8 – Association ends of UseCaseCoreConcepts::ActorLibrary with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] Actor	Actor	the Actors listed in the library
[0..*]	[1..1] UseCaseRepository	UseCaseRepository	the associated use case repository

6.1.2.6 Area

Area of knowledge or activity characterized by a set of concepts and terminology understood by the practitioners in that area

Table 9 shows all attributes of Area.

Table 9 – Attributes of UseCaseCoreConcepts::Area

name	type	description
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 10 shows all association ends of Area with other classes.

Table 10 – Association ends of UseCaseCoreConcepts::Area with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] ParentArea	Area	link with a parent area
[0..*]	[1..1] AreaLibrary	AreaLibrary	the library linking the area
[0..*]	[0..*] BusinessCase	BusinessCase	related business cases associated with the domain
[0..1]	[0..*] SubArea	Area	link with a sub area

6.1.2.7 AreaLibrary

Areas of the repository

Table 11 shows all attributes of AreaLibrary.

Table 11 – Attributes of UseCaseCoreConcepts::AreaLibrary

name	type	description
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 12 shows all association ends of AreaLibrary with other classes.

Table 12 – Association ends of UseCaseCoreConcepts::AreaLibrary with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] Area	Area	the list of areas within the library
[0..1]	[1..1] UseCaseRepository	UseCaseRepository	the associated repository

6.1.2.8 Assumption

It may be used to define a further general assumption for a use case, such as which systems already exist, which contractual relations exist, and which configurations of systems are probably in place. Initial states of information exchanged shall also be identified.

Table 13 shows all attributes of Assumption.

Table 13 – Attributes of UseCaseCoreConcepts::Assumption

name	type	description
number	String	index of the assumption
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 14 shows all association ends of Assumption with other classes.

Table 14 – Association ends of UseCaseCoreConcepts::Assumption with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] UseCase	UseCase	the link toward the related use case

6.1.2.9 Author

This field is used to document who has provided the current version. It can be a person, organization or, for example, standardization committee like a technical committee (TC) or working group (WG).

Table 15 shows all attributes of Author.

Table 15 – Attributes of UseCaseCoreConcepts::Author

name	type	description
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 16 shows all association ends of Author with other classes.

Table 16 – Association ends of UseCaseCoreConcepts::Author with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] Version	Version	the related versions in which the author has participated

6.1.2.10 BusinessCase

It provides a description or reference with some rationale for the suggested use case. Usually the business case is related to several use cases.

Table 17 shows all attributes of BusinessCase.

Table 17 – Attributes of UseCaseCoreConcepts::BusinessCase

name	type	description
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 18 shows all association ends of BusinessCase with other classes.

Table 18 – Association ends of UseCaseCoreConcepts::BusinessCase with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] Area	Area	Use cases can be used in various areas (e.g. energy system). Within these areas different domains are used to define / determine a more specific subgrouping. One or more domains and zones, comma separated, can be specified in the template.
[0..*]	[0..*] UseCase	UseCase	list of use cases covering the business case
[0..*]	[0..1] BusinessCaseLibrary	BusinessCaseLibrary	the library containing the business case

6.1.2.11 BusinessCaseLibrary

Business cases of the repository

Table 19 shows all attributes of BusinessCaseLibrary.

Table 19 – Attributes of UseCaseCoreConcepts::BusinessCaseLibrary

name	type	description
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 20 shows all association ends of BusinessCaseLibrary with other classes.

Table 20 – Association ends of UseCaseCoreConcepts::BusinessCaseLibrary with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] BusinessCase	BusinessCase	the list of business cases
[0..1]	[0..1] UseCaseRepository	UseCaseRepository	the use case repository associated with the library

6.1.2.12 BusinessObject

Business object exchanged in detailed activity steps

Table 21 shows all attributes of BusinessObject.

Table 21 – Attributes of UseCaseCoreConcepts::BusinessObject

name	type	description
identifier	String	identification used in references in the use case description
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 22 shows all association ends of BusinessObject with other classes.

Table 22 – Association ends of UseCaseCoreConcepts::BusinessObject with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] Requirement	Requirement	It can be used to define requirements related to the information and not to the step as in the step by step analysis.
[0..*]	[0..*] Activity	Activity	the activity which uses this exchanged business object
[0..*]	[0..*] InformationModelLibrary	BusinessObjectLibrary	the associated library

6.1.2.13 BusinessObjectLibrary

Business objects which are exchanged in use cases

Table 23 shows all attributes of BusinessObjectLibrary.

Table 23 – Attributes of UseCaseCoreConcepts::BusinessObjectLibrary

name	type	description
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 24 shows all association ends of BusinessObjectLibrary with other classes.

Table 24 – Association ends of UseCaseCoreConcepts::BusinessObjectLibrary with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] BusinessObject	BusinessObject	list of business objects within the library
[0..1]	[1..1] UseCaseRepository	UseCaseRepository	the associated repository

6.1.2.14 CommonTerm

It represents a glossary term and its definition.

Table 25 shows all attributes of CommonTerm.

Table 25 – Attributes of UseCaseCoreConcepts::CommonTerm

name	type	description
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 26 shows all association ends of CommonTerm with other classes.

Table 26 – Association ends of UseCaseCoreConcepts::CommonTerm with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] UseCase	UseCase	the associated use case
[0..*]	[0..*] CommonTermLibrary	CommonTermLibrary	the associated library

6.1.2.15 CommonTermLibrary

Common glossary defining terms for all use cases

Table 27 shows all attributes of CommonTermLibrary.

Table 27 – Attributes of UseCaseCoreConcepts::CommonTermLibrary

name	type	description
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 28 shows all association ends of CommonTermLibrary with other classes.

Table 28 – Association ends of UseCaseCoreConcepts::CommonTermLibrary with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] CommonTerm	CommonTerm	list of common terms
[0..*]	[1..1] UseCaseRepository	UseCaseRepository	the associated repository

6.1.2.16 Condition

It describes either a precondition or a postcondition.

Table 29 shows all attributes of Condition.

Table 29 – Attributes of UseCaseCoreConcepts::Condition

name	type	description
number	String	index of the condition
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 30 shows all association ends of Condition with other classes.

Table 30 – Association ends of UseCaseCoreConcepts::Condition with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] UseCase	UseCase	the associated Use case
[0..*]	[0..1] Scenario	Scenario	reference to the scenario
[0..*]	[0..1] Scenario	Scenario	reference to the scenario

6.1.2.17 CustomInformation

It provides a flexible option to include miscellaneous custom, semi-structured information which does not fit into other parts of the template.

Table 31 shows all attributes of CustomInformation.

Table 31 – Attributes of UseCaseCoreConcepts::CustomInformation

name	type	description
key	String	unique key for identification purposes
value	String	the corresponding information for the provided key
reference	String	This field refers to relevant template section.
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 32 shows all association ends of CustomInformation with other classes.

Table 32 – Association ends of UseCaseCoreConcepts::CustomInformation with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] UseCase	UseCase	the associated use case

6.1.2.18 Drawing

For clarification, in general it is recommended to provide drawing(s) by hand, by a graphic or as UML graphics (preferred in IEC 62559). The drawing should show interactions which identify the steps where possible.

Table 33 shows all attributes of Drawing.

Table 33 – Attributes of UseCaseCoreConcepts::Drawing

name	type	description
drawingType	DrawingClassification	type of drawing
URI	String	resource Path to an image file ex: http://www....../image.jpg or path to an XMI file with a relative diagram path from the root of the UML model. ex: http://www....../Model.xml#Package1/Package2/.../DiagramName
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 34 shows all association ends of Drawing with other classes.

Table 34 – Association ends of UseCaseCoreConcepts::Drawing with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] Activity	Activity	the associated activity
[0..*]	[1..1] UseCase	UseCase	the associated use case
[0..*]	[0..1] Scenario	Scenario	the associated scenario

6.1.2.19 DrawingClassification enumeration

Possible types of drawing

Table 35 shows all literals of DrawingClassification.

Table 35 – Literals of UseCaseCoreConcepts::DrawingClassification

literal	description
domain overview	overview of relationships between domains and use cases
use case overview	overview of use case short template information
scenarios flowchart	flowchart of scenarios in a use case
documentation	additional information about a use case
scenario overview	overview of static information related to a scenario
activities flowchart	flowchart of activities for describing a scenario or another activity (in the case of recursive decomposition)
activity overview	static information related to an activity
role model	overview of role and their relationships
business objects overview	overview of a set of objects used in information exchanges
other	other kind of drawings

6.1.2.20 FurtherActorInformation

Individual or additional information that relates to the use case can be provided.

Table 36 shows all attributes of FurtherActorInformation.

Table 36 – Attributes of UseCaseCoreConcepts::FurtherActorInformation

name	type	description
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 37 shows all association ends of FurtherActorInformation with other classes.

Table 37 – Association ends of UseCaseCoreConcepts::FurtherActorInformation with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] Actor	Actor	information related to an Actor
[0..*]	[0..1] ActorGrouping	ActorGrouping	An Actor is grouped within lists.

6.1.2.21 IdentifiedObject root class

This class is used to gather attributes used for identification purposes. Other classes in this package may inherit from that class.

Table 38 shows all attributes of IdentifiedObject.

Table 38 – Attributes of UseCaseCoreConcepts::IdentifiedObject

name	type	description
description	String	description of the object
mRID	String	master resource identifier
name	String	a short name

6.1.2.22 KeyPerformanceIndicator

Key Performance Indicator (KPI) related to the objectives of the use case

Table 39 shows all attributes of KeyPerformanceIndicator.

Table 39 – Attributes of UseCaseCoreConcepts::KeyPerformanceIndicator

name	type	description
identifier	String	unique identifier for the Key Performance Indicator (KPI)
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 40 shows all association ends of KeyPerformanceIndicator with other classes.

Table 40 – Association ends of UseCaseCoreConcepts::KeyPerformanceIndicator with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] UseCase	UseCase	the associated use case
[0..1]	[0..*] Objective	Objective	the related objective

6.1.2.23 Narrative

It describes the narrative of the use case.

Table 41 shows all attributes of Narrative.

Table 41 – Attributes of UseCaseCoreConcepts::Narrative

name	type	description
shortDescription	String	It is a short text intended to summarize the main idea as a service for the reader who is searching for a use case or looking for an overview. Recommendation: This short description should have not more than 150 words.
completeDescription	String	It provides a complete narrative of the use case from a user's point of view, describing what occurs when, why, with what expectation, and under what conditions. This narrative should be written in plain text so that non-domain experts can understand it. The length of the complete description can range from a few sentences to a few pages, depending on the complexity and/or newness of the use case. This description often helps the domain expert to reflect on the requirements for the use case before getting into the details in the next sections of the use case template. The description may include drawings for explanation.
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 42 shows all association ends of Narrative with other classes.

Table 42 – Association ends of UseCaseCoreConcepts::Narrative with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] UseCase	UseCase	the associated use case

6.1.2.24 Objective

Objective of the use case

Table 43 shows all attributes of Objective.

Table 43 – Attributes of UseCaseCoreConcepts::Objective

name	type	description
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 44 shows all association ends of Objective with other classes.

Table 44 – Association ends of UseCaseCoreConcepts::Objective with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] KeyPerformanceIndicator	KeyPerformanceIndicator	the key performance indicator associated with the objective
[0..*]	[0..1] UseCase	UseCase	the associated use case

6.1.2.25 Reference

Any references that might restrict or affect the design, understanding, and requirements of the use case shall be identified, including contracts, regulations, policies, financial considerations, engineering constraints, pollution constraints, and other environmental quality issues.

Table 45 shows all attributes of Reference.

Table 45 – Attributes of UseCaseCoreConcepts::Reference

name	type	description
impact	String	This information describes the nature of the influence of the document on the use case.
link	String	If available, a public link to the document can be provided.
number	String	index of the reference
originatorOrganization	String	This describes the name of the organization publishing the document.
status	String	status of the referenced document
type	String	There are different reference types (e.g. standard, regulation, contract, others like publications).
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 46 shows all association ends of Reference with other classes.

Table 46 – Association ends of UseCaseCoreConcepts::Reference with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] UseCase	UseCase	the associated use case

6.1.2.26 Remark

It is used for further comments which are not considered elsewhere.

Table 47 shows all attributes of Remark.

Table 47 – Attributes of UseCaseCoreConcepts::Remark

name	type	description
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 48 shows all association ends of Remark with other classes.

Table 48 – Association ends of UseCaseCoreConcepts::Remark with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] UseCase	UseCase	the associated use case

6.1.2.27 Requirement

Requirement of a part of the use case

Table 49 shows all attributes of Requirement.

Table 49 – Attributes of UseCaseCoreConcepts::Requirement

name	type	description
identifier	String	identification used in references
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 50 shows all association ends of Requirement with other classes.

Table 50 – Association ends of UseCaseCoreConcepts::Requirement with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] Activity	Activity	the related activity
[0..*]	[0..*] BusinessObject	BusinessObject	the business object associated with the requirement
[0..*]	[0..1] RequirementCategory	RequirementCategory	the associated category of requirements
[0..*]	[0..1] Scenario	Scenario	the related scenario

6.1.2.28 RequirementCategory

Requirements can be sorted in categories. For complex categories the category name can be "dot-delimited".

Table 51 shows all attributes of RequirementCategory.

Table 51 – Attributes of UseCaseCoreConcepts::RequirementCategory

name	type	description
identifier	String	identification used in references
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 52 shows all association ends of RequirementCategory with other classes.

Table 52 – Association ends of UseCaseCoreConcepts::RequirementCategory with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] Requirement	Requirement	information about the requirement
[0..1]	[0..*] SubCategory	RequirementCategory	the associated sub categories of requirements
[0..*]	[1..1] RequirementLibrary	RequirementLibrary	the associated library
[0..*]	[0..1] ParentCategory	RequirementCategory	the reference to the parent category of requirement

6.1.2.29 RequirementLibrary

Requirements of the repository

Table 53 shows all attributes of RequirementLibrary.

Table 53 – Attributes of UseCaseCoreConcepts::RequirementLibrary

name	type	description
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 54 shows all association ends of RequirementLibrary with other classes.

Table 54 – Association ends of UseCaseCoreConcepts::RequirementLibrary with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] RequirementCategory	RequirementCategory	the category for the requirement
[0..1]	[1..1] UseCaseRepository	UseCaseRepository	the associated repository

6.1.2.30 Scenario

A possible sequence of interactions

Table 55 shows all attributes of Scenario.

Table 55 – Attributes of UseCaseCoreConcepts::Scenario

name	type	description
number	String	index of the scenario
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 56 shows all association ends of Scenario with other classes.

Table 56 – Association ends of UseCaseCoreConcepts::Scenario with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] Activity	Activity	list of activities for the scenario
[0..*]	[0..*] PrimaryActor	Actor	It describes which Actor triggers this scenario.
[0..1]	[0..*] Precondition	Condition	It describes which condition(s) should have been met before this scenario happens.
[0..1]	[0..*] Postcondition	Condition	It describes which condition(s) should prevail after this scenario happens. The post conditions may also define "success" or "failure" conditions for each use case.
[0..1]	[0..*] Drawing	Drawing	the list of associated drawing
[0..1]	[0..*] Requirement	Requirement	information about the requirement
[0..*]	[1..1] UseCase	UseCase	the associated use case
[0..1]	[0..*] TriggeringEvent	TriggeringEvent	list of triggering events for the scenario

6.1.2.31 TriggeringEvent

It describes which event triggers a scenario.

Table 57 shows all attributes of TriggeringEvent.

Table 57 – Attributes of UseCaseCoreConcepts::TriggeringEvent

name	type	description
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 58 shows all association ends of TriggeringEvent with other classes.

Table 58 – Association ends of UseCaseCoreConcepts::TriggeringEvent with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] Scenario	Scenario	the reference to the scenario having this trigger

6.1.2.32 UseCase

Specification of a set of actions performed by a system, which yields an observable result that is, typically, of value for one or more Actors or other stakeholders of the system

Table 59 shows all attributes of UseCase.

Table 59 – Attributes of UseCaseCoreConcepts::UseCase

name	type	description
classification	String	At the international level the use case description might be generic enough to describe a use case in a more general way independently from the national or regional market design. But use cases might be used to describe regional or national specific circumstances like laws or even project specific details. If the use case reflects those circumstances, it should be characterized accordingly.
identifier	String	The identification number (ID) of a use case is unique within a repository or project and serves for organization/administration of use cases.
keywords	String	Keywords can be defined in order to support extended search functionalities within a use case repository. Multiple keywords should be provided as a comma-separated list.
levelOfDepth	String	Use cases can be described on different levels.
nature	String	It classifies the main focus of the use case.
prioritization	String	If considering a larger number of use cases, it might be interesting to cluster them according to priority. This prioritization might be different from country to country.
scope	String	The scope defines the limits of the use case.
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 60 shows all association ends of UseCase with other classes.

Table 60 – Association ends of UseCaseCoreConcepts::UseCase with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] PrimaryActor	Actor	identification of the PrimaryActor linked to the use case
[0..1]	[0..*] Assumption	Assumption	set of assumptions related to the use case
[0..*]	[0..*] BusinessCase	BusinessCase	list of business cases associated with the use case
[0..*]	[0..*] CommonTerm	CommonTerm	the associated common terms
[1..1]	[0..*] Drawing	Drawing	the list of associated drawings
[1..1]	[0..1] Narrative	Narrative	the information about the narrative part of the use case
[0..1]	[0..*] RelatedObjective	Objective	list of objectives of the use case
[0..*]	[0..*] Reference	Reference	information about referenced documents
[1..1]	[0..*] Remark	Remark	remark associated with the use case
[1..1]	[0..*] Version	Version	list of versions describing changes about the use case
[0..1]	[0..*] RelatedUseCase	UseCase	Known relations to other use cases can be provided here if, for example, the use case is a more detailed one related to a high level use case, or it is an alternative to an existing use case.
[0..*]	[0..*] ActorGrouping	ActorGrouping	A use case references several sets of Actors.
[0..1]	[0..*] Prerequisite	Condition	It describes what condition(s) should have been met before initiation of the use case, such as prior state of the Actors and activities.
[0..1]	[0..*] CustomInformation	CustomInformation	related additional information
[0..1]	[0..*] KeyPerformanceIndicator	KeyPerformanceIndicator	information about the KPI related to the use case
[1..1]	[0..*] Scenario	Scenario	reference to the scenarios belonging to the use case
[0..*]	[1..1] UseCaseLibrary	UseCaseLibrary	reference to the associated library
[0..*]	[0..1] UseCase	UseCase	parent use case

6.1.2.33 UseCaseLibrary

Use cases of the repository

Table 61 shows all attributes of UseCaseLibrary.

Table 61 – Attributes of UseCaseCoreConcepts::UseCaseLibrary

name	type	description
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 62 shows all association ends of UseCaseLibrary with other classes.

Table 62 – Association ends of UseCaseCoreConcepts::UseCaseLibrary with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] UseCase	UseCase	list of use cases within the library
[0..1]	[1..1] UseCaseRepository	UseCaseRepository	the associated repository

6.1.2.34 UseCaseRepository

Database, based on a given use cases template, for editing, maintenance and administration of use cases, Actors and requirements including their interrelations

Table 63 shows all attributes of UseCaseRepository.

Table 63 – Attributes of UseCaseCoreConcepts::UseCaseRepository

name	type	description
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 64 shows all association ends of UseCaseRepository with other classes.

Table 64 – Association ends of UseCaseCoreConcepts::UseCaseRepository with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] ActorLibrary	ActorLibrary	library of Actors
[1..1]	[0..1] AreaLibrary	AreaLibrary	the associated library of areas
[1..1]	[0..1] BusinessObjectLibrary	BusinessObjectLibrary	the library of business objects
[1..1]	[0..1] RequirementLibrary	RequirementLibrary	the associated library
[1..1]	[0..1] UseCaseLibrary	UseCaseLibrary	the associated library of use cases
[0..1]	[0..1] BusinessCaseLibrary	BusinessCaseLibrary	the library associated with the repository
[1..1]	[0..*] CommonTermLibrary	CommonTermLibrary	the associated library of common terms

6.1.2.35 Version

Version of the use case description

Table 65 shows all attributes of Version.

Table 65 – Attributes of UseCaseCoreConcepts::Version

name	type	description
approvalStatus	String	information used within the standardization process
changes	String	It represents the differences with respect to the previous version. Multiple changes are separated with paragraphs.
date	DateTime	date of creation of the version, in the format YYYY-MM-DD
number	String	sequential number to identify the version of the document
description	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 66 shows all association ends of Version with other classes.

Table 66 – Association ends of UseCaseCoreConcepts::Version with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] Author	Author	list of authors participating in the changes of this version
[0..*]	[1..1] UseCase	UseCase	information about the associated use case

6.1.2.36 Top package Primitives

6.1.2.36.1 General

Figure 17 shows class diagram Primitives.

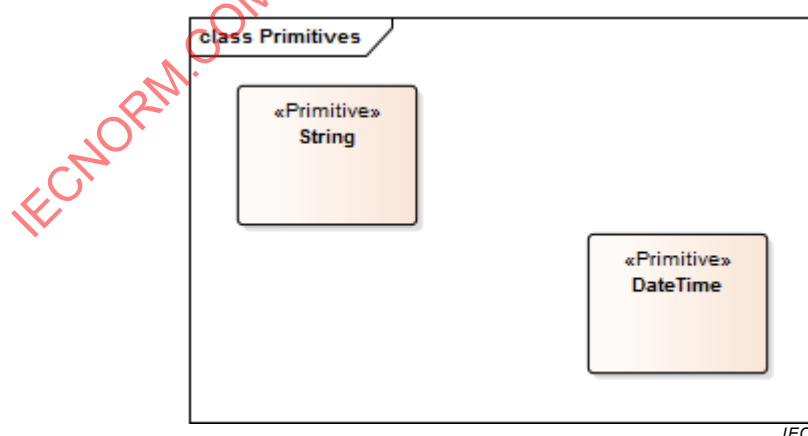


Figure 17 – Class diagram Primitives::Primitives

6.1.2.36.2 DateTime primitive

DateTime Primitive type

6.1.2.36.3 String primitive

String Primitive type

6.2 Use case library exchange format

6.2.1 UML use case library exchange model (UML Contextual Model)

6.2.1.1 General

The use case library exchange model is described through a UML Contextual Model. The XSD content is generated from this model.

6.2.1.2 Model package UseCase_ExchangeModel

6.2.1.2.1 General

IEC 62559 UML Contextual Model for the use case information exchange model

Figure 18 shows package diagram ContextualModel_Dependency.

IECNORM.COM : Click to view the full PDF of IEC 62559-3:2017

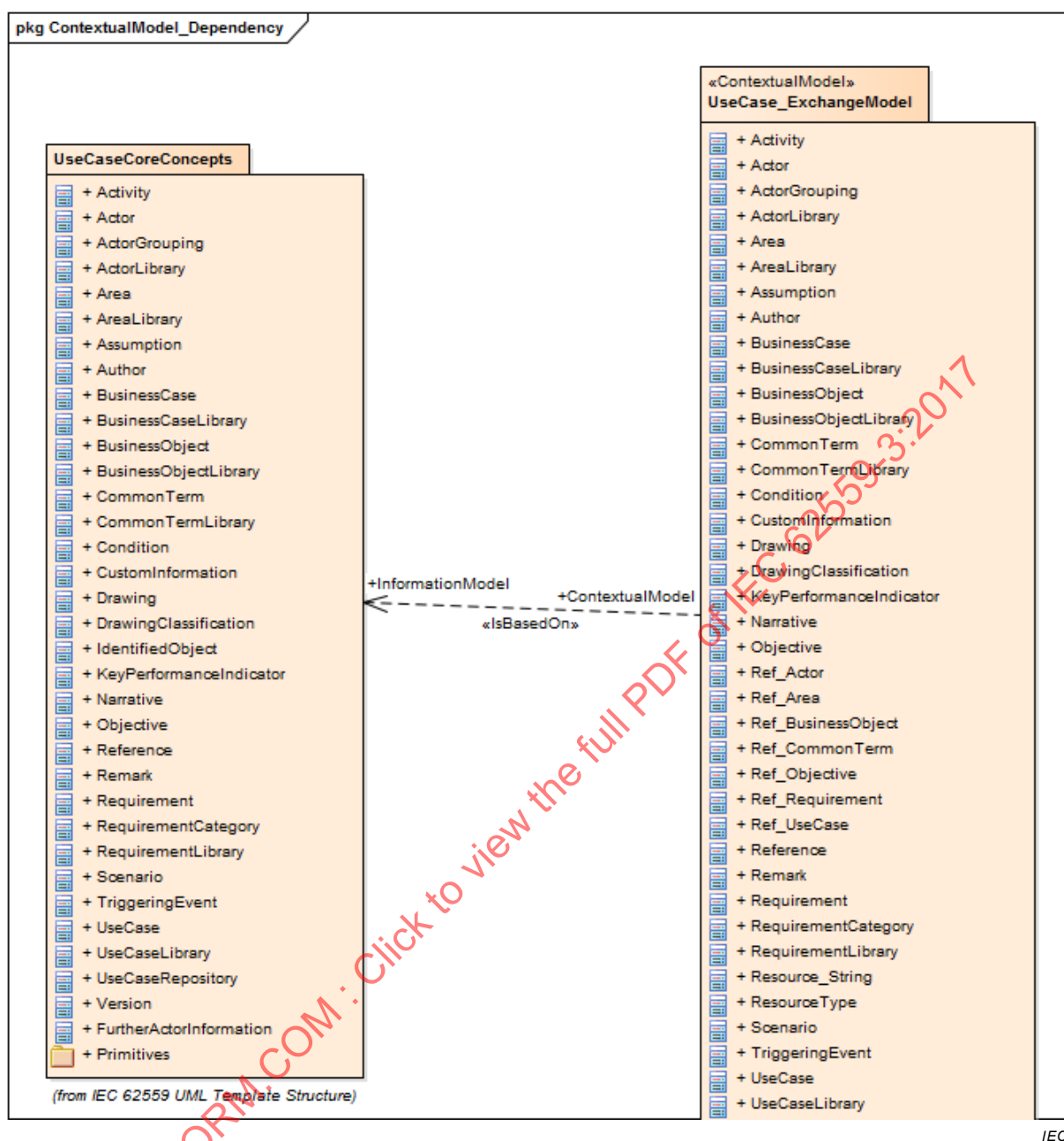


Figure 18 – Package diagram UseCase_ExchangeModel::ContextualModel_Dependency

Figure 19 shows class diagram UseCase_Repository.

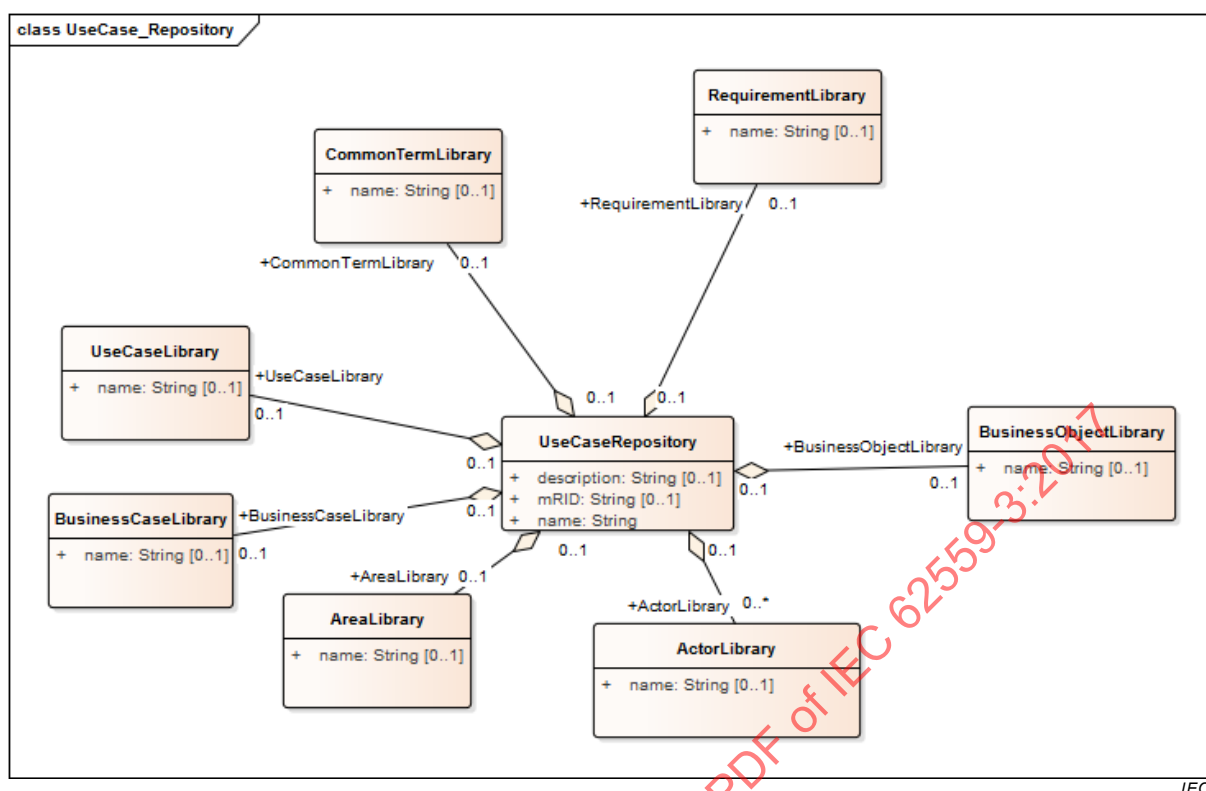


Figure 19 – Class diagram UseCase_ExchangeModel::UseCase_Repository

Figure 20 shows class diagram UseCase_ShortDescription.

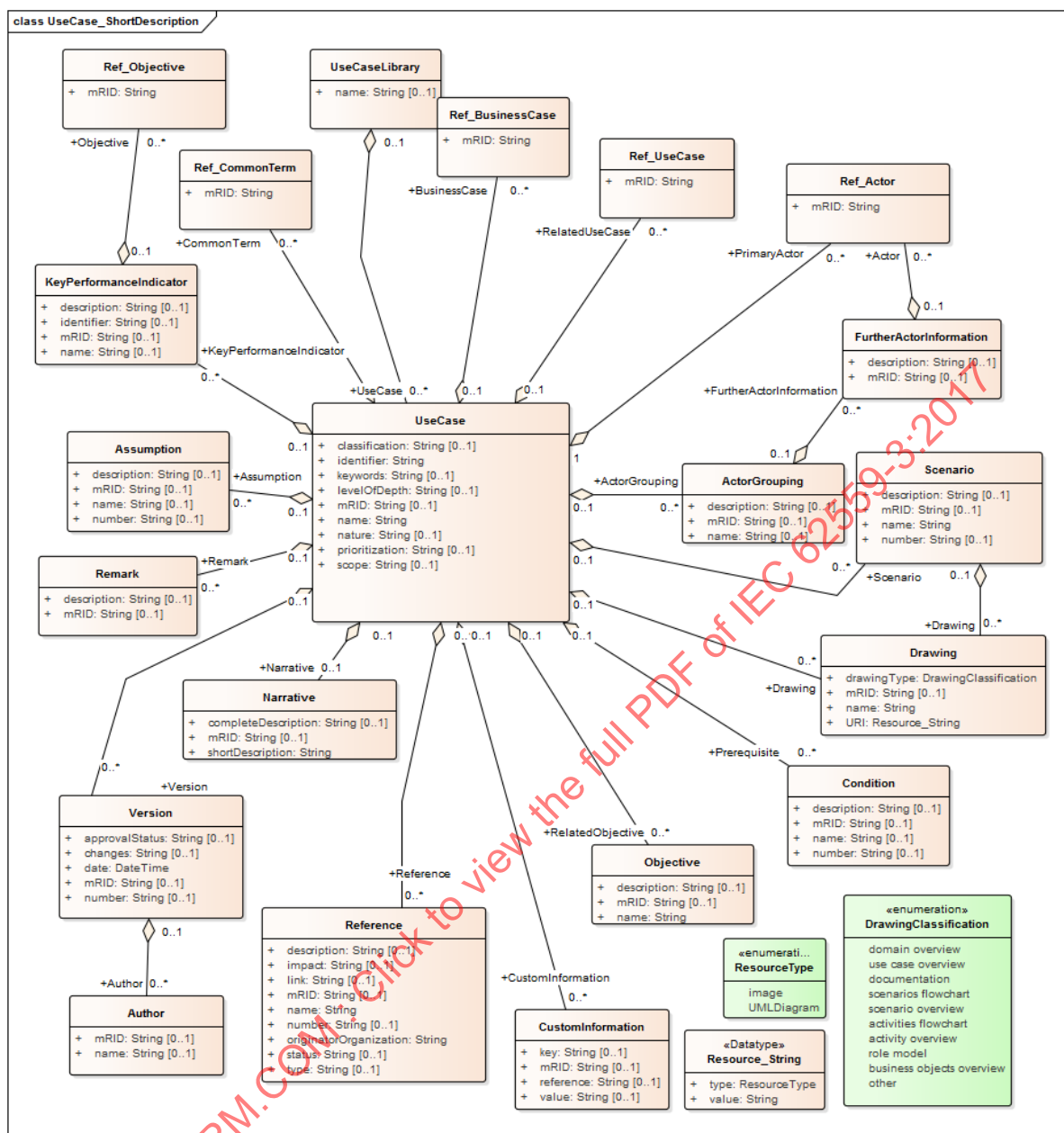


Figure 20 – Class diagram UseCase_ExchangeModel::UseCase_ShortDescription

Figure 21 shows class diagram UseCase_DetailedActivities.

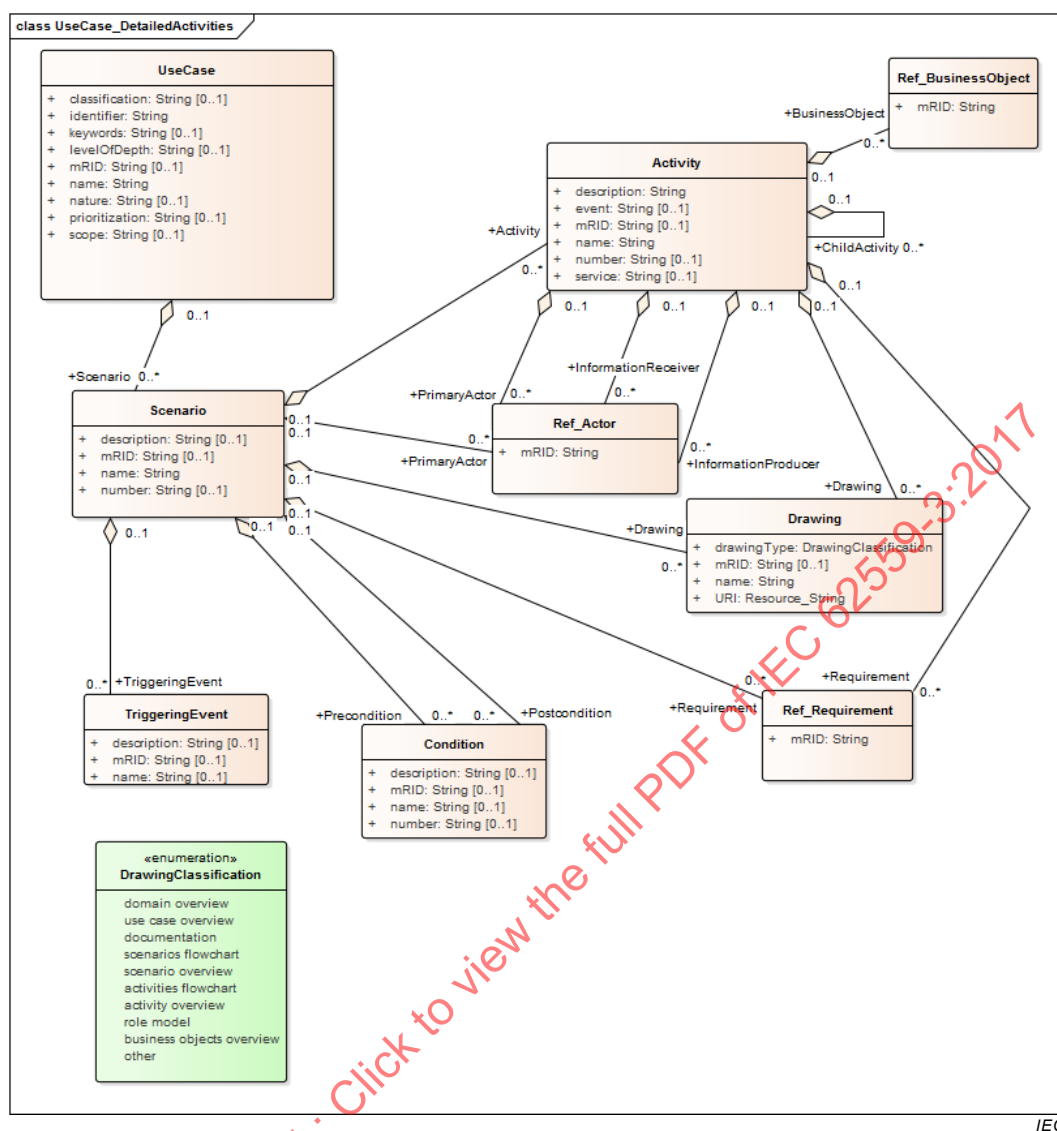
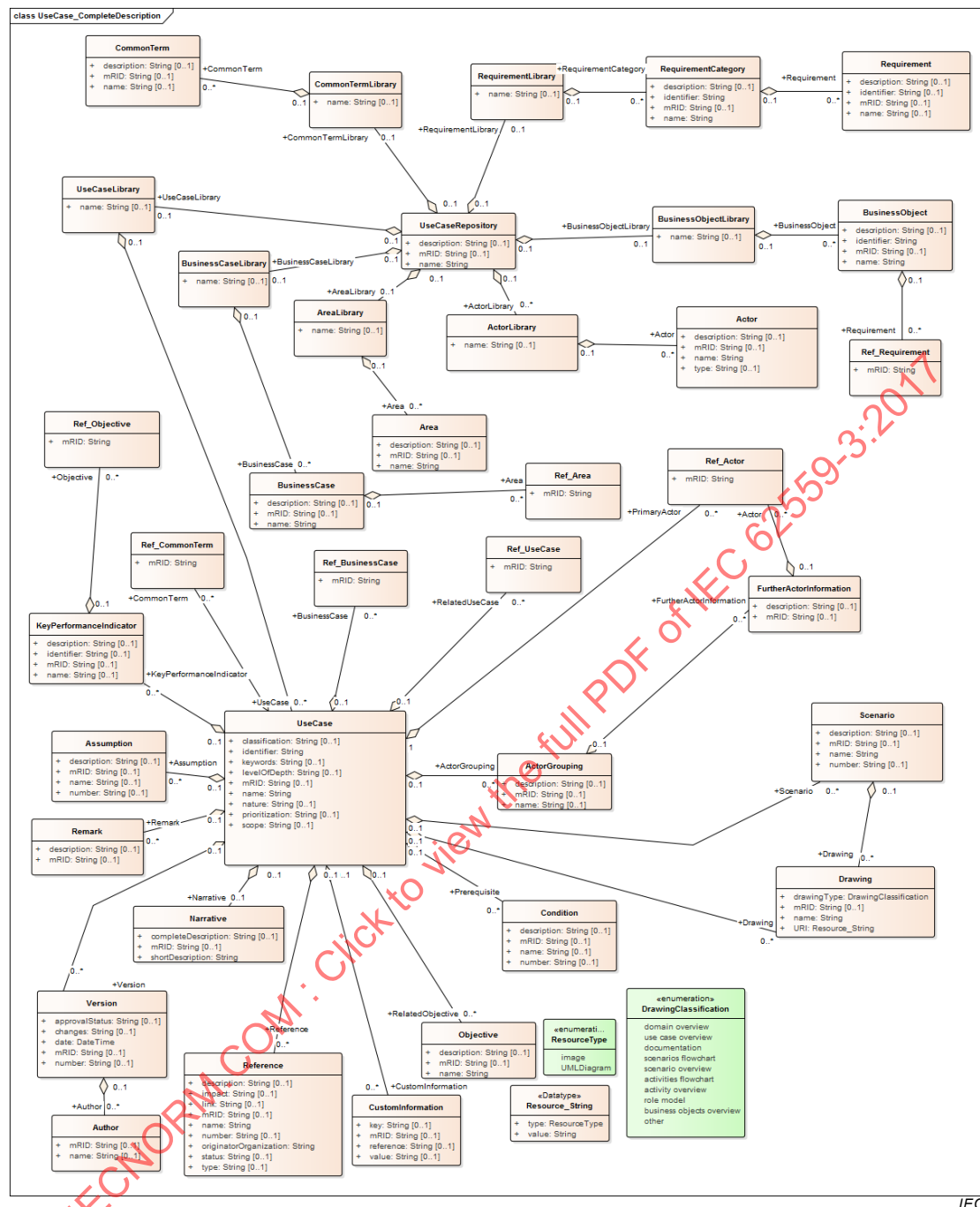


Figure 21 – Class diagram UseCase_ExchangeModel::UseCase_DetailedActivities

Figure 22 shows class diagram UseCase_CompleteDescription.



IEC

Figure 22 – Class diagram UseCase_ExchangeModel::UseCase_CompleteDescription

6.2.1.2.2 Activity root class

It may be subdivided into other activities or represent an elementary step of the scenario.

Table 67 shows all attributes of Activity.

Table 67 – Attributes of UseCase_ExchangeModel::Activity

name	type	description
name	String	a short name
description	String	description of the object
event	String	It is the event that triggers the step. This might be completion of previous steps (in that case the event is the comma-separated list of their numbers) or the condition to exit a loop.
service	String	This column identifies the nature of the information flow and the originator of the information. Available options are CREATE, GET, CHANGE, DELETE, CANCEL, EXECUTE derived from IEC 61968-100:2013. Additionally, REPORT, TIMER and REPEAT are suggested. – CREATE means that an information object is to be created at the Producer. – GET (this is the default value if none is populated) means that the Receiver requests information from the Producer (default). – CHANGE means that information is to be updated. Producer updates the Receiver's information. – DELETE means that information is to be deleted. Producer deletes information from the Receiver. – CANCEL, CLOSE imply actions related to processes, such as the closure of a work order or the cancellation of a control request. – EXECUTE is used when a complex transaction is being conveyed using a service, which potentially contains more than one verb. – REPORT is used to represent transferral of unsolicited information or asynchronous information flows. Producer provides information to the Receiver. – TIMER is used to represent a waiting period. When using the TIMER service, the Information Producer and Information Receiver fields shall refer to the same Actor. – REPEAT is used to indicate that a series of steps is repeated until a condition or trigger event. The condition is specified as the text in the "Event" column for this row or step. Following the word REPEAT, the first and last step numbers of the series to be repeated shall appear, in parenthesis, in the following form: REPEAT(X-Y) where X is the first step and Y is the last step. These common service definitions are related to automation/information or communication systems. In case the use case template is applied in other domains, further services might be used and described.
mRID	String	master resource identifier

Table 68 shows all association ends of Activity with other classes.

Table 68 – Association ends of UseCase_ExchangeModel::Activity with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	Scenario	This association end is an aggregation.
[0..*]	[0..1]	Activity	This association end is an aggregation.
[0..1]	[0..*] InformationReceiver	Ref_Actor	This identifies the receiver of the information. This should be one of the Actors defined in the repository.
[0..1]	[0..*] InformationProducer	Ref_Actor	This identifies the producer or source of the information. This should be one of the Actors defined in the repository.
[0..1]	[0..*] Requirement	Ref_Requirement	Several requirements may be listed in one step, comma separated.
[0..1]	[0..*] BusinessObject	Ref_BusinessObject	This describes briefly the information to be exchanged between the two Actors – information producer and information receiver. It may list several information items in one step, comma separated.
[0..1]	[0..*] Drawing	Drawing	the list of associated drawing
[0..1]	[0..*] PrimaryActor	Ref_Actor	reference to the PrimaryActor
[0..1]	[0..*] ChildActivity	Activity	child activities

6.2.1.2.3 Actor root class

It is the entity that communicates and interacts.

These Actors can include people, software applications, systems, databases, and even the power system itself.

Table 69 shows all attributes of Actor.

Table 69 – Attributes of UseCase_ExchangeModel::Actor

name	type	description
name	String	a short name
type	String	type of the Actor (business, system, ...)
mRID	String	master resource identifier

Table 70 shows all association ends of Actor with other classes.

Table 70 – Association ends of UseCase_ExchangeModel::Actor with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	ActorLibrary	This association end is an aggregation.

6.2.1.2.4 ActorGrouping root class

Group of Actors used to organize an Actor list

Table 71 shows all attributes of ActorGrouping.

Table 71 – Attributes of UseCase_ExchangeModel::ActorGrouping

name	type	description
name	String	a short name
description	String	description of the object
mRID	String	master resource identifier

Table 72 shows all association ends of ActorGrouping with other classes.

Table 72 – Association ends of UseCase_ExchangeModel::ActorGrouping with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] FurtherActorInformation	FurtherActorInformation	In a context of one use case, an Actor grouped into an ActorGrouping may have one FurtherActorInformation.
[0..*]	[0..1]	UseCase	This association end is an aggregation.

6.2.1.2.5 ActorLibrary root class

Actors of the repository

Table 73 shows all attributes of ActorLibrary.

Table 73 – Attributes of UseCase_ExchangeModel::ActorLibrary

name	type	description
name	String	a short name

Table 74 shows all association ends of ActorLibrary with other classes.

Table 74 – Association ends of UseCase_ExchangeModel::ActorLibrary with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	UseCaseRepository	This association end is an aggregation.
[0..1]	[0..*] Actor	Actor	the Actors listed in the library

6.2.1.2.6 Area root class

Area of knowledge or activity characterized by a set of concepts and terminology understood by the practitioners in that area

Table 75 shows all attributes of Area.

Table 75 – Attributes of UseCase_ExchangeModel::Area

name	type	description
name	String	a short name
description	String	description of the object
mRID	String	master resource identifier

Table 76 shows all association ends of Area with other classes.

Table 76 – Association ends of UseCase_ExchangeModel::Area with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	AreaLibrary	This association end is an aggregation.

6.2.1.2.7 AreaLibrary root class

Areas of the repository

Table 77 shows all attributes of AreaLibrary.

Table 77 – Attributes of UseCase_ExchangeModel::AreaLibrary

name	type	description
name	String	a short name

Table 78 shows all association ends of AreaLibrary with other classes.

Table 78 – Association ends of UseCase_ExchangeModel::AreaLibrary with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..1]	UseCaseRepository	This association end is an aggregation.
[0..1]	[0..*] Area	Area	the list of areas within the library

6.2.1.2.8 Assumption root class

It may be used to define a further general assumption for a use case, such as which systems already exist, which contractual relations exist, and which configurations of systems are probably in place. Initial states of information exchanged shall also be identified.

Table 79 shows all attributes of Assumption.

Table 79 – Attributes of UseCase_ExchangeModel::Assumption

name	type	description
name	String	a short name
number	String	index of the assumption
description	String	description of the object
mRID	String	master resource identifier

Table 80 shows all association ends of Assumption with other classes.

Table 80 – Association ends of UseCase_ExchangeModel::Assumption with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	UseCase	This association end is an aggregation.

6.2.1.2.9 Author root class

This field is used to document who has provided the current version. It can be a person, organization or, for example, standardization committee like a technical committee (TC) or working group (WG).

Table 81 shows all attributes of Author.

Table 81 – Attributes of UseCase_ExchangeModel::Author

name	type	description
name	String	a short name
mRID	String	master resource identifier

Table 82 shows all association ends of Author with other classes.

Table 82 – Association ends of UseCase_ExchangeModel::Author with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	Version	This association end is an aggregation.

6.2.1.2.10 BusinessCase root class

It provides a description or reference with some rationale for the suggested use case. Usually the business case is related to several use cases.

Table 83 shows all attributes of BusinessCase.

Table 83 – Attributes of UseCase_ExchangeModel::BusinessCase

name	type	description
name	String	a short name
description	String	description of the object
mRID	String	master resource identifier

Table 84 shows all association ends of BusinessCase with other classes.

Table 84 – Association ends of UseCase_ExchangeModel::BusinessCase with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	BusinessCaseLibrary	This association end is an aggregation.
[0..1]	[0..*] Area	Ref_Area	Use cases can be used in various areas (e.g. energy system). Within these areas different domains are used to define / determine a more specific subgrouping. One or more domains and zones, comma separated, can be specified in the template.

6.2.1.2.11 BusinessCaseLibrary root class

Business cases of the repository

Table 85 shows all attributes of BusinessCaseLibrary.

Table 85 – Attributes of UseCase_ExchangeModel::BusinessCaseLibrary

name	type	description
name	String	a short name

Table 86 shows all association ends of BusinessCaseLibrary with other classes.

Table 86 – Association ends of UseCase_ExchangeModel::BusinessCaseLibrary with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..1]	UseCaseRepository	This association end is an aggregation.
[0..1]	[0..*] BusinessCase	BusinessCase	the list of business cases

6.2.1.2.12 BusinessObject root class

Business object exchanged in detailed activity steps

Table 87 shows all attributes of BusinessObject.

Table 87 – Attributes of UseCase_ExchangeModel::BusinessObject

name	type	description
name	String	a short name
description	String	description of the object
identifier	String	identification used in references in the use case description
mRID	String	master resource identifier

Table 88 shows all association ends of BusinessObject with other classes.

Table 88 – Association ends of UseCase_ExchangeModel::BusinessObject with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	BusinessObjectLibrary	This association end is an aggregation.
[0..1]	[0..*] Requirement	Ref_Requirement	It can be used to define requirements related to the information and not to the step as in the step by step analysis.

6.2.1.2.13 BusinessObjectLibrary root class

Business objects which are exchanged in use cases

Table 89 shows all attributes of BusinessObjectLibrary.

Table 89 – Attributes of UseCase_ExchangeModel::BusinessObjectLibrary

name	type	description
name	String	a short name

Table 90 shows all association ends of BusinessObjectLibrary with other classes.

Table 90 – Association ends of UseCase_ExchangeModel::BusinessObjectLibrary with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..1]	UseCaseRepository	This association end is an aggregation.
[0..1]	[0..*] BusinessObject	BusinessObject	list of business objects within the library

6.2.1.2.14 CommonTerm root class

It represents a glossary term and its definition.

Table 91 shows all attributes of CommonTerm.

Table 91 – Attributes of UseCase_ExchangeModel::CommonTerm

name	type	description
name	String	a short name
description	String	description of the object
mRID	String	master resource identifier

Table 92 shows all association ends of CommonTerm with other classes.

Table 92 – Association ends of UseCase_ExchangeModel::CommonTerm with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	CommonTermLibrary	This association end is an aggregation.

6.2.1.2.15 CommonTermLibrary root class

Common glossary defining terms for all use cases.

Table 93 shows all attributes of CommonTermLibrary.

Table 93 – Attributes of UseCase_ExchangeModel::CommonTermLibrary

name	type	description
name	String	a short name

Table 94 shows all association ends of CommonTermLibrary with other classes.

Table 94 – Association ends of UseCase_ExchangeModel::CommonTermLibrary with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..1]	UseCaseRepository	This association end is an aggregation.
[0..1]	[0..*] CommonTerm	CommonTerm	list of common terms

6.2.1.2.16 Condition root class

It describes either a precondition or a postcondition.

Table 95 shows all attributes of Condition.

Table 95 – Attributes of UseCase_ExchangeModel::Condition

name	type	description
name	String	a short name
number	String	index of the condition
description	String	description of the object
mRID	String	master resource identifier

Table 96 shows all association ends of Condition with other classes.

Table 96 – Association ends of UseCase_ExchangeModel::Condition with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	Scenario	This association end is an aggregation.
[0..*]	[0..1]	Scenario	This association end is an aggregation.
[0..*]	[0..1]	UseCase	This association end is an aggregation.

6.2.1.2.17 CustomInformation root class

It provides a flexible option to include miscellaneous custom, semi-structured information which does not fit into other parts of the template.

Table 97 shows all attributes of CustomInformation.

Table 97 – Attributes of UseCase_ExchangeModel::CustomInformation

name	type	description
key	String	unique key for identification purposes
value	String	the corresponding information for the provided key
reference	String	This field refers to relevant template section.
mRID	String	master resource identifier

Table 98 shows all association ends of CustomInformation with other classes.

Table 98 – Association ends of UseCase_ExchangeModel::CustomInformation with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	UseCase	This association end is an aggregation.

6.2.1.2.18 Drawing root class

For clarification, in general it is recommended to provide drawing(s) by hand, by a graphic or as UML graphics (preferred in IEC 62559). The drawing should show interactions which identify the steps where possible.

Table 99 shows all attributes of Drawing.

Table 99 – Attributes of UseCase_ExchangeModel::Drawing

name	type	description
name	String	a short name
drawingType	DrawingClassification	type of drawing
URI	Resource_String	resource Path to an image file ex: http://www....../image.jpg or path to an XMI file with a relative diagram path from the root of the UML model. ex: http://www.../Model.xmi#Package1/Package2/.../DiagramName
mRID	String	master resource identifier

Table 100 shows all association ends of Drawing with other classes.

Table 100 – Association ends of UseCase_ExchangeModel::Drawing with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	Activity	This association end is an aggregation.
[0..*]	[0..1]	UseCase	This association end is an aggregation.
[0..*]	[0..1]	Scenario	This association end is an aggregation.

6.2.1.2.19 DrawingClassification enumeration

Possible types of drawing.

Table 101 shows all literals of DrawingClassification.

Table 101 – Literals of UseCase_ExchangeModel::DrawingClassification

literal	description
domain overview	overview of relationships between domains and use cases
use case overview	overview of use case short template information
documentation	additional information about a use case
scenarios flowchart	flowchart of scenarios in a use case
scenario overview	overview of static information related to a scenario
activities flowchart	flowchart of activities for describing a scenario or another activity (in the case of recursive decomposition)
activity overview	static information related to an activity
role model	overview of role and their relationships
business objects overview	overview of a set of objects used in information exchanges
other	other kind of drawings

6.2.1.2.20 KeyPerformanceIndicator root class

Key Performance Indicator (KPI) related to the objectives of the use case.

Table 102 shows all attributes of KeyPerformanceIndicator.

Table 102 – Attributes of UseCase_ExchangeModel::KeyPerformanceIndicator

name	type	description
name	String	a short name
description	String	description of the object
identifier	String	unique identifier for the Key Performance Indicator (KPI)
mRID	String	master resource identifier

Table 103 shows all association ends of KeyPerformanceIndicator with other classes.

Table 103 – Association ends of UseCase_ExchangeModel::KeyPerformanceIndicator with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	UseCase	This association end is an aggregation.
[0..1]	[0..*] Objective	Ref_Objective	the related objective

6.2.1.2.21 Narrative root class

It describes the narrative of the use case.

Table 104 shows all attributes of Narrative.

Table 104 – Attributes of UseCase_ExchangeModel::Narrative

name	type	description
shortDescription	String	It is a short text intended to summarize the main idea as a service for the reader who is searching for a use case or looking for an overview. Recommendation: This short description should have not more than 150 words.
completeDescription	String	It provides a complete narrative of the use case from a user's point of view, describing what occurs when, why, with what expectation, and under what conditions. This narrative should be written in plain text so that non-domain experts can understand it. The length of the complete description can range from a few sentences to a few pages, depending on the complexity and/or newness of the use case. This description often helps the domain expert to reflect on the requirements for the use case before getting into the details in the next sections of the use case template. The description may include drawings for explanation.
mRID	String	master resource identifier

Table 105 shows all association ends of Narrative with other classes.

Table 105 – Association ends of UseCase_ExchangeModel::Narrative with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..1]	UseCase	This association end is an aggregation.

6.2.1.2.22 Objective root class

Objective of the use case.

Table 106 shows all attributes of Objective.

Table 106 – Attributes of UseCase_ExchangeModel::Objective

name	type	description
name	String	a short name
description	String	description of the object
mRID	String	master resource identifier

Table 107 shows all association ends of Objective with other classes.

Table 107 – Association ends of UseCase_ExchangeModel::Objective with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	UseCase	This association end is an aggregation.

6.2.1.2.23 Ref_Actor root class

It is the entity that communicates and interacts.

These Actors can include people, software applications, systems, databases, and even the power system itself.

Table 108 shows all attributes of Ref_Actor.

Table 108 – Attributes of UseCase_ExchangeModel::Ref_Actor

name	type	description
mRID	String	master resource identifier

Table 109 shows all association ends of Ref_Actor with other classes.

Table 109 – Association ends of UseCase_ExchangeModel::Ref_Actor with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	Activity	This association end is an aggregation.
[0..*]	[0..1]	Activity	This association end is an aggregation.
[0..*]	[0..1]	Activity	This association end is an aggregation.
[0..*]	[0..1]	Scenario	This association end is an aggregation.
[0..*]	[0..1]	FurtherActorInformation	This association end is an aggregation.
[0..*]	[1..1]	UseCase	This association end is an aggregation.

6.2.1.2.24 Ref_Area root class

Area of knowledge or activity characterized by a set of concepts and terminology understood by the practitioners in that area.

Table 110 shows all attributes of Ref_Area.

Table 110 – Attributes of UseCase_ExchangeModel::Ref_Area

name	type	description
mRID	String	master resource identifier

Table 111 shows all association ends of Ref_Area with other classes.

Table 111 – Association ends of UseCase_ExchangeModel::Ref_Area with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	BusinessCase	This association end is an aggregation.

6.2.1.2.25 Ref_BusinessCase root class

It provides a description or reference with some rationale for the suggested use case. Usually the business case is related to several use cases.

Table 112 shows all attributes of Ref_BusinessCase.

Table 112 – Attributes of UseCase_ExchangeModel::Ref_BusinessCase

name	type	description
mRID	String	master resource identifier

Table 113 shows all association ends of Ref_BusinessCase with other classes.

Table 113 – Association ends of UseCase_ExchangeModel::Ref_BusinessCase with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	UseCase	This association end is an aggregation.

6.2.1.2.26 Ref_BusinessObject root class

Business object exchanged in detailed activity steps.

Table 114 shows all attributes of Ref_BusinessObject.

Table 114 – Attributes of UseCase_ExchangeModel::Ref_BusinessObject

name	type	description
mRID	String	master resource identifier

Table 115 shows all association ends of Ref_BusinessObject with other classes.

Table 115 – Association ends of UseCase_ExchangeModel::Ref_BusinessObject with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	Activity	This association end is an aggregation.

6.2.1.2.27 Ref_CommonTerm root class

It represents a glossary term and its definition.

Table 116 shows all attributes of Ref_CommonTerm.

Table 116 – Attributes of UseCase_ExchangeModel::Ref_CommonTerm

name	type	description
mRID	String	master resource identifier

Table 117 shows all association ends of Ref_CommonTerm with other classes.

Table 117 – Association ends of UseCase_ExchangeModel::Ref_CommonTerm with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	UseCase	This association end is an aggregation.

6.2.1.2.28 Ref_Objective root class

Objective of the use case.

Table 118 shows all attributes of Ref_Objective.

Table 118 – Attributes of UseCase_ExchangeModel::Ref_Objective

name	type	description
mRID	String	master resource identifier

Table 119 shows all association ends of Ref_Objective with other classes.

Table 119 – Association ends of UseCase_ExchangeModel::Ref_Objective with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	KeyPerformanceIndicator	This association end is an aggregation.

6.2.1.2.29 Ref_Requirement root class

Requirement of a part of the use case.

Table 120 shows all attributes of Ref_Requirement.

Table 120 – Attributes of UseCase_ExchangeModel::Ref_Requirement

name	type	description
mRID	String	master resource identifier

Table 121 shows all association ends of Ref_Requirement with other classes.

Table 121 – Association ends of UseCase_ExchangeModel::Ref_Requirement with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	Activity	This association end is an aggregation.
[0..*]	[0..1]	BusinessObject	This association end is an aggregation.
[0..*]	[0..1]	Scenario	This association end is an aggregation.

6.2.1.2.30 Ref_UseCase root class

Specification of a set of actions performed by a system, which yields an observable result that, is, typically, of value for one or more Actors or other stakeholders of the system.

Table 122 shows all attributes of Ref_UseCase.

Table 122 – Attributes of UseCase_ExchangeModel::Ref_UseCase

name	type	description
mRID	String	master resource identifier

Table 123 shows all association ends of Ref_UseCase with other classes.

Table 123 – Association ends of UseCase_ExchangeModel::Ref_UseCase with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	UseCase	This association end is an aggregation.

6.2.1.2.31 Reference root class

Any references that might restrict or affect the design, understanding, and requirements of the use case shall be identified, including contracts, regulations, policies, financial considerations, engineering constraints, pollution constraints, and other environmental quality issues.

Table 124 shows all attributes of Reference.

Table 124 – Attributes of UseCase_ExchangeModel::Reference

name	type	description
name	String	a short name
description	String	description of the object
type	String	There are different reference types (e.g. standard, regulation, contract, others like publications).
impact	String	This information describes the nature of the influence of the document on the use case.
status	String	status of the referenced document
link	String	If available, a public link to the document can be provided.
originatorOrganization	String	This describes the name of the organization publishing the document.
number	String	index of the reference
mRID	String	master resource identifier

Table 125 shows all association ends of Reference with other classes.

Table 125 – Association ends of UseCase_ExchangeModel::Reference with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	UseCase	This association end is an aggregation.

6.2.1.2.32 Remark root class

It is used for further comments which are not considered elsewhere.

Table 126 shows all attributes of Remark.

Table 126 – Attributes of UseCase_ExchangeModel::Remark

name	type	description
description	String	description of the object
mRID	String	master resource identifier

Table 127 shows all association ends of Remark with other classes.

Table 127 – Association ends of UseCase_ExchangeModel::Remark with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	UseCase	This association end is an aggregation.

6.2.1.2.33 Requirement root class

Requirement of a part of the use case.

Table 128 shows all attributes of Requirement.

Table 128 – Attributes of UseCase_ExchangeModel::Requirement

name	type	description
name	String	a short name
identifier	String	identification used in references
description	String	description of the object
mRID	String	master resource identifier

Table 129 shows all association ends of Requirement with other classes.

Table 129 – Association ends of UseCase_ExchangeModel::Requirement with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	RequirementCategory	This association end is an aggregation.

6.2.1.2.34 RequirementCategory root class

Requirements can be sorted in categories. For complex categories the category name can be "dot-delimited".

Table 130 shows all attributes of RequirementCategory.

Table 130 – Attributes of UseCase_ExchangeModel::RequirementCategory

name	type	description
name	String	a short name
identifier	String	identification used in references
description	String	description of the object
mRID	String	master resource identifier

Table 131 shows all association ends of RequirementCategory with other classes.

Table 131 – Association ends of UseCase_ExchangeModel::RequirementCategory with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	RequirementLibrary	This association end is an aggregation.
[0..1]	[0..*] Requirement	Requirement	information about the requirement

6.2.1.2.35 RequirementLibrary root class

Requirements of the repository.

Table 132 shows all attributes of RequirementLibrary.

Table 132 – Attributes of UseCase_ExchangeModel::RequirementLibrary

name	type	description
name	String	a short name

Table 133 shows all association ends of RequirementLibrary with other classes.

Table 133 – Association ends of UseCase_ExchangeModel::RequirementLibrary with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..1]	UseCaseRepository	This association end is an aggregation.
[0..1]	[0..*] RequirementCategory	RequirementCategory	the category for the requirement

6.2.1.2.36 Resource_String datatype

String Primitive type.

Table 134 shows all attributes of Resource_String.

Table 134 – Attributes of UseCase_ExchangeModel::Resource_String

name	type	description
value	String	primary value of the data type
type	ResourceType	supplementary attribute of the data type

6.2.1.2.37 ResourceType enumeration

List of possible resources.

Table 135 shows all literals of ResourceType.

Table 135 – Literals of UseCase_ExchangeModel::ResourceType

literal	description
UMLDiagram	UML kind for a resource

6.2.1.2.38 Scenario root class

A possible sequence of interactions.

Table 136 shows all attributes of Scenario.

Table 136 – Attributes of UseCase_ExchangeModel::Scenario

name	type	description
name	String	a short name
description	String	description of the object
number	String	index of the scenario
mRID	String	master resource identifier

Table 137 shows all association ends of Scenario with other classes.

Table 137 – Association ends of UseCase_ExchangeModel::Scenario with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] Requirement	Ref_Requirement	information about the requirement
[0..*]	[0..1]	UseCase	This association end is an aggregation.
[0..1]	[0..*] Activity	Activity	list of activities for the scenario
[0..1]	[0..*] Precondition	Condition	It describes which condition(s) should have been met before this scenario happens.
[0..1]	[0..*] Postcondition	Condition	It describes which condition(s) should prevail after this scenario happens. The post conditions may also define "success" or "failure" conditions for each use case.
[0..1]	[0..*] Drawing	Drawing	the list of associated drawing
[0..1]	[0..*] PrimaryActor	Ref_Actor	It describes which Actor triggers this scenario.
[0..1]	[0..*] TriggeringEvent	TriggeringEvent	list of triggering events for the scenario

6.2.1.2.39 TriggeringEvent root class

It describes which event triggers a scenario.

Table 138 shows all attributes of TriggeringEvent.

Table 138 – Attributes of UseCase_ExchangeModel::TriggeringEvent

name	type	description
name	String	a short name
description	String	description of the object
mRID	String	master resource identifier

Table 139 shows all association ends of TriggeringEvent with other classes.

Table 139 – Association ends of UseCase_ExchangeModel::TriggeringEvent with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	Scenario	This association end is an aggregation.

6.2.1.2.40 UseCase root class

Specification of a set of actions performed by a system, which yields an observable result that is, typically, of value for one or more Actors or other stakeholders of the system

Table 140 shows all attributes of UseCase.

Table 140 – Attributes of UseCase_ExchangeModel::UseCase

name	type	description
name	String	a short name
identifier	String	The identification number (ID) of a use case is unique within a repository or project and serves for organization/administration of use cases.
nature	String	It classifies the main focus of the use case.
classification	String	At the international level the use case description might be generic enough to describe a use case in a more general way independently from the national or regional market design. But use cases might be used to describe regional or national specific circumstances like laws or even project specific details. If the use case reflects those circumstances, it should be characterized accordingly.
keywords	String	Keywords can be defined in order to support extended search functionalities within a use case repository. Multiple keywords should be provided as a comma-separated list.
prioritization	String	If considering a larger number of use cases, it might be interesting to cluster them according to priority. This prioritization might be different from country to country.
scope	String	The scope defines the limits of the use case.
mRID	String	master resource identifier

Table 141 shows all association ends of UseCase with other classes.

Table 141 – Association ends of UseCase_ExchangeModel::UseCase with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] ActorGrouping	ActorGrouping	A use case references several sets of Actors.
[0..1]	[0..*] Prerequisite	Condition	It describes what condition(s) should have been met before initiation of the use case, such as prior state of the Actors and activities.
[1..1]	[0..*] PrimaryActor	Ref_Actor	identification of the PrimaryActor linked to the use case
[0..1]	[0..*] BusinessCase	Ref_BusinessCase	list of business cases associated with the use case
[0..*]	[0..1]	UseCaseLibrary	This association end is an aggregation.
[0..1]	[0..*] Assumption	Assumption	set of assumptions related to the use case
[0..1]	[0..*] CustomInformation	CustomInformation	related additional information
[0..1]	[0..*] Drawing	Drawing	the list of associated drawings
[0..1]	[0..*] KeyPerformanceIndicator	KeyPerformanceIndicator	information about the KPI related to the use case
[0..1]	[0..1] Narrative	Narrative	the information about the narrative part of the use case
[0..1]	[0..*] RelatedObjective	Objective	list of objectives of the use case
[0..1]	[0..*] CommonTerm	Ref_CommonTerm	the associated common terms
[0..1]	[0..*] RelatedUseCase	Ref_UseCase	Known relations to other use cases can

[mult from]	[mult to] name	type	description
			be provided here if, for example, the use case is a more detailed one related to a high level use case, or it is an alternative to an existing use case.
[0..1]	[0..*] Reference	Reference	information about referenced documents
[0..1]	[0..*] Remark	Remark	remark associated with the use case
[0..1]	[0..*] Scenario	Scenario	reference to the scenarios belonging to the use case
[0..1]	[0..*] Version	Version	list of versions describing changes about the use case

6.2.1.2.41 UseCaseLibrary root class

Use cases of the repository.

Table 142 shows all attributes of UseCaseLibrary.

Table 142 – Attributes of UseCase_ExchangeModel::UseCaseLibrary

name	type	description
name	String	a short name

Table 143 shows all association ends of UseCaseLibrary with other classes.

Table 143 – Association ends of UseCase_ExchangeModel::UseCaseLibrary with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..1]	UseCaseRepository	This association end is an aggregation.
[0..1]	[0..*] UseCase	UseCase	list of use cases within the library

6.2.1.2.42 UseCaseRepository root class

Database, based on a given use cases template, for editing, maintenance and administration of use cases, Actors and requirements including their interrelations.

Table 144 shows all attributes of UseCaseRepository.

Table 144 – Attributes of UseCase_ExchangeModel::UseCaseRepository

name	type	description
name	String	a short name
description	String	description of the object
mRID	String	master resource identifier

Table 145 shows all association ends of UseCaseRepository with other classes.

Table 145 – Association ends of UseCase_ExchangeModel::UseCaseRepository with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] ActorLibrary	ActorLibrary	library of Actors
[0..1]	[0..1] AreaLibrary	AreaLibrary	the associated library of areas
[0..1]	[0..1] BusinessCaseLibrary	BusinessCaseLibrary	the library associated with the repository
[0..1]	[0..1] BusinessObjectLibrary	BusinessObjectLibrary	the library of business objects
[0..1]	[0..1] CommonTermLibrary	CommonTermLibrary	the associated library of common terms
[0..1]	[0..1] RequirementLibrary	RequirementLibrary	the associated library
[0..1]	[0..1] UseCaseLibrary	UseCaseLibrary	the associated library of use cases

6.2.1.2.43 Version root class

Version of the use case description.

Table 146 shows all attributes of Version.

Table 146 – Attributes of UseCase_ExchangeModel::Version

name	type	description
date	DateTime	date of creation of the version, in the format YYYY-MM-DD
changes	String	It represents the differences with respect to the previous version. Multiple changes are separated with paragraphs.
approvalStatus	String	information used within the standardization process
number	String	sequential number to identify the version of the document
mRID	String	master resource identifier

Table 147 shows all association ends of Version with other classes.

Table 147 – Association ends of UseCase_ExchangeModel::Version with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	UseCase	This association end is an aggregation.
[0..1]	[0..*] Author	Author	list of authors participating in the changes of this version

6.2.1.2.44 FurtherActorInformation root class

Individual or additional information that relates to the use case can be provided.

Table 148 shows all attributes of FurtherActorInformation.

Table 148 – Attributes of UseCase_ExchangeModel::FurtherActorInformation

name	type	description
description	String	description of the object
mRID	String	master resource identifier

Table 149 shows all association ends of FurtherActorInformation with other classes.

Table 149 – Association ends of UseCase_ExchangeModel::FurtherActorInformation with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1]	ActorGrouping	This association end is an aggregation.
[0..1]	[0..*] Actor	Ref_Actor	information related to an Actor

6.2.2 Use case library XML syntactic format

Table 150 – Use case library XSD

<CODE BEGINS>

XSD content
<pre> <?xml version="1.0" encoding="utf-8"?> <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:UC="http://www.SYC.org/IEC62559/part3/UC/V01/Build08082016" targetNamespace="http://www.SYC.org/IEC62559/part3/UC/V01/Build08082016"> <xsd:complexType name="Activity"> <xsd:annotation> <xsd:documentation>It may be subdivided into other activities or represent an elementary step of the scenario.</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="BusinessObject" type="UC:Ref_BusinessObject" minOccurs="0" maxOccurs="unbounded"> <xsd:annotation> <xsd:documentation>This describes briefly the information to be exchanged between the two actors – information producer and information receiver. It may list several information items in one step, comma separated.</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="ChildActivity" type="UC:Activity" minOccurs="0" maxOccurs="unbounded"> <xsd:annotation> <xsd:documentation>child activities</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="description" type="xsd:string" minOccurs="1" maxOccurs="1"> <xsd:annotation> <xsd:documentation>description of the object</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="Drawing" type="UC:Drawing" minOccurs="0" maxOccurs="unbounded"> <xsd:annotation> <xsd:documentation>the list of associated drawing</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="event" type="xsd:string" minOccurs="0" maxOccurs="1"> <xsd:annotation> <xsd:documentation>It is the event that triggers the step. This might be completion of previous steps (in that case the event is the comma-separated list of their numbers) or the condition to exit a loop.</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="InformationProducer" type="UC:Ref_Actor" minOccurs="0" maxOccurs="unbounded"> <xsd:annotation> <xsd:documentation>This identifies the producer or source of the information. This </pre>

should be one of the actors defined in the repository.

```

</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="InformationReceiver" type="UC:Ref_Actor" minOccurs="0"
maxOccurs="unbounded">
  <xsd:annotation>
    <xsd:documentation>This identifies the receiver of the information. This should be one
of the actors defined in the repository.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>master resource identifier</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="name" type="xsd:string" minOccurs="1" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>a short name</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="number" type="xsd:string" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>index of the activity</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="PrimaryActor" type="UC:Ref_Actor" minOccurs="0" maxOccurs="unbounded">
  <xsd:annotation>
    <xsd:documentation>reference to the PrimaryActor</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="Requirement" type="UC:Ref_Requirement" minOccurs="0"
maxOccurs="unbounded">
  <xsd:annotation>
    <xsd:documentation>Several requirements may be listed in one step, comma
separated.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="service" type="xsd:string" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>This column identifies the nature of the information flow and the
originator of the information. Available options are CREATE, GET, CHANGE, DELETE, CANCEL, EXECUTE derived from
IEC 61968-100:2013.
Additionally, REPORT, TIMER and REPEAT are suggested.

- CREATE means that an information object is to be created at the Producer.
- GET (this is the default value if none is populated) means that the Receiver requests information from the Producer
(default).
- CHANGE means that information is to be updated. Producer updates the Receiver's information.
- DELETE means that information is to be deleted. Producer deletes information from the Receiver.
- CANCEL, CLOSE imply actions related to processes, such as the closure of a work order or the cancellation of a
control request.
- EXECUTE is used when a complex transaction is being conveyed using a service, which potentially contains more
than one verb.
- REPORT is used to represent transferral of unsolicited information or asynchronous information flows. Producer
provides information to the Receiver.
- TIMER is used to represent a waiting period. When using the TIMER service, the Information Producer and
Information Receiver fields shall refer to the same actor.
- REPEAT is used to indicate that a series of steps is repeated until a condition or trigger event. The condition is
specified as the text in the "Event" column for this row or step. Following the word REPEAT, the first and last step numbers of
the series to be repeated shall appear, in parenthesis, in the following form: REPEAT(X-Y) where X is the first step and Y is the
last step.

These common service definitions are related to automation/information or communication systems. In case the use case
template is applied in other domains, further services might be used and described.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Drawing">
  <xsd:annotation>
    <xsd:documentation>For clarification, in general it is recommended to provide drawing(s) by hand, by
a graphic or as UML graphics (preferred in IEC 62559). The drawing should show interactions which identify the steps where
possible.</xsd:documentation>
  </xsd:annotation>
</xsd:sequence>
<xsd:element name="drawingType" type="UC:DrawingClassification" minOccurs="1" maxOccurs="1">
  <xsd:annotation>

```

```

        <xsd:documentation>type of drawing</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="name" type="xsd:string" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>a short name</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="URI" type="UC:Resource_String" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>resource Path to an image file
ex: http://www..../image.jpg
or
path to an XML file with a relative diagram path from the root of the UML model. ex:
http://www.../Model.xmi#Package1/Package2/.../DiagName</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="DrawingClassification">
  <xsd:annotation>
    <xsd:documentation>possible types of drawing</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="domain overview">
      <xsd:annotation>
        <xsd:documentation>overview of relationships between domains and use
cases</xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="use case overview">
      <xsd:annotation>
        <xsd:documentation>overview of use case short template
information</xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="documentation">
      <xsd:annotation>
        <xsd:documentation>additional information about a use case</xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="scenarios flowchart">
      <xsd:annotation>
        <xsd:documentation>flowchart of scenarios in a use case</xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="scenario overview">
      <xsd:annotation>
        <xsd:documentation>overview of static information related to a
scenario</xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="activities flowchart">
      <xsd:annotation>
        <xsd:documentation>flowchart of activities for describing a scenario or another activity
(in the case of recursive decomposition)</xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="activity overview">
      <xsd:annotation>
        <xsd:documentation>static information related to an activity</xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="business objects overview">
      <xsd:annotation>
        <xsd:documentation>overview of a set of objects used in information
exchanges</xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="other">
      <xsd:annotation>
        <xsd:documentation>other kind of drawings</xsd:documentation>

```

```

        </xsd:annotation>
        </xsd:enumeration>
        <xsd:enumeration value="role model">
        <xsd:annotation>
            <xsd:documentation>overview of role and their relationships</xsd:documentation>
        </xsd:annotation>
        </xsd:enumeration>
    </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="Resource_String">
    <xsd:annotation>
        <xsd:documentation>String Primitive type</xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
        <xsd:extension base="UC:Resource_String_value_ValueType">
            <xsd:attribute name="type" type="UC:ResourceType" use="required">
                <xsd:annotation>
                    <xsd:documentation>supplementary attribute of the data
type</xsd:documentation>
                </xsd:annotation>
            </xsd:attribute>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
<xsd:simpleType name="Resource_String_value_ValueType">
    <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="ResourceType">
    <xsd:annotation>
        <xsd:documentation>list of possible resources</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="image">
            <xsd:annotation>
                <xsd:documentation>any image kind for a resource</xsd:documentation>
            </xsd:annotation>
        </xsd:enumeration>
        <xsd:enumeration value="UMLDiagram">
            <xsd:annotation>
                <xsd:documentation>UML kind for a resource</xsd:documentation>
            </xsd:annotation>
        </xsd:enumeration>
    </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="Ref_Requirement">
    <xsd:annotation>
        <xsd:documentation>requirement of a part of the use case</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="mRID" type="xsd:string" minOccurs="1" maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>master resource identifier</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Ref_Actor">
    <xsd:annotation>
        <xsd:documentation>It is the entity that communicates and interacts.
These actors can include people, software applications, systems, databases, and even the power system
itself.</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="mRID" type="xsd:string" minOccurs="1" maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>master resource identifier</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Ref_BusinessObject">
    <xsd:annotation>
        <xsd:documentation>business object exchanged in detailed activity steps</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="mRID" type="xsd:string" minOccurs="1" maxOccurs="1">
            <xsd:annotation>

```

```

        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Actor">
  <xsd:annotation>
    <xsd:documentation>It is the entity that communicates and interacts.
    These actors can include people, software applications, systems, databases, and even the power system
    itself.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>description of the object</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="name" type="xsd:string" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>a short name</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="type" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>type of the actor (business, system, ...)</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ActorGrouping">
  <xsd:annotation>
    <xsd:documentation>group of actors used to organize an actor list</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>description of the object</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="FurtherActorInformation" type="UC:FurtherActorInformation" minOccurs="0"
    maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>In a context of one use case, an actor grouped into an
        ActorGrouping may have one FurtherActorInformation.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>a short name</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="FurtherActorInformation">
  <xsd:annotation>
    <xsd:documentation>Individual or additional information that relates to the use case can be
    provided.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="Actor" type="UC:Ref_Actor" minOccurs="0" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>information related to an actor</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>description of the object</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```

```

        </xsd:annotation>
      </xsd:element>
      <xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>master resource identifier</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="ActorLibrary">
    <xsd:annotation>
      <xsd:documentation>actors of the repository</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="Actor" type="UC:Actor" minOccurs="0" maxOccurs="unbounded">
        <xsd:annotation>
          <xsd:documentation>the actors listed in the library</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>a short name</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="Area">
    <xsd:annotation>
      <xsd:documentation>area of knowledge or activity characterized by a set of concepts and terminology
understood by the practitioners in that area</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>description of the object</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>master resource identifier</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="name" type="xsd:string" minOccurs="1" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>a short name</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="Author">
    <xsd:annotation>
      <xsd:documentation>This field is used to document who has provided the current version. It can be a
person, organization or, for example, standardization committee like a technical committee (TC) or working group
(WG).</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>master resource identifier</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>a short name</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="BusinessCase">
    <xsd:annotation>
      <xsd:documentation>It provides a description or reference with some rationale for the suggested use
case. Usually the business case is related to several use cases.</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="Area" type="UC:Ref_Area" minOccurs="0" maxOccurs="unbounded">
        <xsd:annotation>
          <xsd:documentation>Use cases can be used in various areas (e.g. energy system).

```

Within these areas different domains are used to define / determine a more specific subgrouping. One or more domains and zones, comma separated, can be specified in the template. </xsd:documentation>

```

</xsd:annotation>
</xsd:element>
<xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>description of the object</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>master resource identifier</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="name" type="xsd:string" minOccurs="1" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>a short name</xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Ref_Area">
  <xsd:annotation>
    <xsd:documentation>area of knowledge or activity characterized by a set of concepts and terminology
understood by the practitioners in that area</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="mRID" type="xsd:string" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="BusinessCaseLibrary">
  <xsd:annotation>
    <xsd:documentation>business cases of the repository</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="BusinessCase" type="UC:BusinessCase" minOccurs="0"
maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>the list of business cases</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>a short name</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="BusinessObject">
  <xsd:annotation>
    <xsd:documentation>business object exchanged in detailed activity steps</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>description of the object</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="identifier" type="xsd:string" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>identification used in references in the use case
description</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="name" type="xsd:string" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>a short name</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```

```

        </xsd:element>
        <xsd:element name="Requirement" type="UC:Ref_Requirement" minOccurs="0"
maxOccurs="unbounded">
            <xsd:annotation>
                <xsd:documentation>It can be used to define requirements related to the information
and not to the step as in the step by step analysis.</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="BusinessObjectLibrary">
    <xsd:annotation>
        <xsd:documentation>business objects which are exchanged in use cases</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="BusinessObject" type="UC:BusinessObject" minOccurs="0"
maxOccurs="unbounded">
            <xsd:annotation>
                <xsd:documentation>list of business objects within the library</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>a short name</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CommonTerm">
    <xsd:annotation>
        <xsd:documentation>It represents a glossary term and its definition.</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>description of the object</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>master resource identifier</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>a short name</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CommonTermLibrary">
    <xsd:annotation>
        <xsd:documentation>common glossary defining terms for all use cases</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="CommonTerm" type="UC:CommonTerm" minOccurs="0"
maxOccurs="unbounded">
            <xsd:annotation>
                <xsd:documentation>list of common terms</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>a short name</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Condition">
    <xsd:annotation>
        <xsd:documentation>It describes either a precondition or a postcondition.</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>description of the object</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>

```

```

</xsd:element>
<xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>master resource identifier</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>a short name</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="number" type="xsd:string" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>index of the condition</xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CustomInformation">
  <xsd:annotation>
    <xsd:documentation>It provides a flexible option to include miscellaneous custom, semi-structured
information which does not fit into other parts of the template.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="key" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>unique key for identification purposes</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="reference" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>This field refers to relevant template
section.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="value" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>the corresponding information for the provided
key</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="KeyPerformanceIndicator">
  <xsd:annotation>
    <xsd:documentation>Key Performance Indicator (KPI) related to the objectives of the use
case</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>description of the object</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="identifier" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>unique identifier for the Key Performance Indicator
(KPI)</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>a short name</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="Objective" type="UC:Ref_Objective" minOccurs="0" maxOccurs="unbounded">
      <xsd:annotation>

```

```

        <xsd:documentation>the related objective</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Ref_Objective">
  <xsd:annotation>
    <xsd:documentation>objective of the use case</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="mRID" type="xsd:string" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Narrative">
  <xsd:annotation>
    <xsd:documentation>It describes the narrative of the use case.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="completeDescription" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>It provides a complete narrative of the use case from a user's
point of view, describing what occurs when, why, with what expectation, and under what conditions. This narrative should be
written in plain text so that non-domain experts can understand it.
The length of the complete description can range from a few sentences to a few pages, depending on the complexity and/or
newness of the use case. This description often helps the domain expert to reflect on the requirements for the use case before
getting into the details in the next sections of the use case template.
The description may include drawings for explanation.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="shortDescription" type="xsd:string" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>It is a short text intended to summarize the main idea as a
service for the reader who is searching for a use case or looking for an overview.
Recommendation: This short description should have not more than 150 words.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Objective">
  <xsd:annotation>
    <xsd:documentation>objective of the use case</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>description of the object</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="name" type="xsd:string" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>a short name</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Ref_BusinessCase">
  <xsd:annotation>
    <xsd:documentation>It provides a description or reference with some rationale for the suggested use
case. Usually the business case is related to several use cases.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="mRID" type="xsd:string" minOccurs="1" maxOccurs="1">
      <xsd:annotation>

```

```

        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Ref_CommonTerm">
  <xsd:annotation>
    <xsd:documentation>It represents a glossary term and its definition.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="mRID" type="xsd:string" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Ref_UseCase">
  <xsd:annotation>
    <xsd:documentation>specification of a set of actions performed by a system, which yields an
observable result that is, typically, of value for one or more actors or other stakeholders of the system </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="mRID" type="xsd:string" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Reference">
  <xsd:annotation>
    <xsd:documentation>Any references that might restrict or affect the design, understanding, and
requirements of the use case shall be identified, including contracts, regulations, policies, financial considerations, engineering
constraints, pollution constraints, and other environmental quality issues.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>description of the object</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="impact" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>This information describes the nature of the influence of the
document on the use case.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="link" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>If available, a public link to the document can be
provided.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="name" type="xsd:string" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>a short name</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="number" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>index of the reference</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="originatorOrganization" type="xsd:string" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>This describes the name of the organization publishing the
document.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="status" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>

```

```

        <xsd:documentation>status of the referenced document</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="type" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>There are different reference types (e.g. standard, regulation,
contract, others like publications).</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Remark">
  <xsd:annotation>
    <xsd:documentation>It is used for further comments which are not considered
elsewhere.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>description of the object</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Requirement">
  <xsd:annotation>
    <xsd:documentation>requirement of a part of the use case</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>description of the object</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="identifier" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>identification used in references</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>a short name</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="RequirementCategory">
  <xsd:annotation>
    <xsd:documentation>Requirements can be sorted in categories. For complex categories the category
name can be "dot-delimited".</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>description of the object</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="identifier" type="xsd:string" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>identification used in references</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>

```

```

        <xsd:element name="name" type="xsd:string" minOccurs="1" maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>a short name</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="Requirement" type="UC:Requirement" minOccurs="0"
maxOccurs="unbounded">
            <xsd:annotation>
                <xsd:documentation>information about the requirement</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="RequirementLibrary">
    <xsd:annotation>
        <xsd:documentation>requirements of the repository</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>a short name</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="RequirementCategory" type="UC:RequirementCategory" minOccurs="0"
maxOccurs="unbounded">
            <xsd:annotation>
                <xsd:documentation>the category for the requirement</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Scenario">
    <xsd:annotation>
        <xsd:documentation>a possible sequence of interactions</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="Activity" type="UC:Activity" minOccurs="0" maxOccurs="unbounded">
            <xsd:annotation>
                <xsd:documentation>list of activities for the scenario</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>description of the object</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="Drawing" type="UC:Drawing" minOccurs="0" maxOccurs="unbounded">
            <xsd:annotation>
                <xsd:documentation>the list of associated drawing</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>master resource identifier</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="name" type="xsd:string" minOccurs="1" maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>a short name</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="number" type="xsd:string" minOccurs="0" maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>index of the scenario</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="Precondition" type="UC:Condition" minOccurs="0" maxOccurs="unbounded">
            <xsd:annotation>
                <xsd:documentation>It describes which condition(s) should have been met before this
scenario happens.</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="Postcondition" type="UC:Condition" minOccurs="0" maxOccurs="unbounded">
            <xsd:annotation>
                <xsd:documentation>It describes which condition(s) should prevail after this scenario
happens. The post conditions may also define "success" or "failure" conditions for each use case.</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>

```

```

</xsd:element>
<xsd:element name="PrimaryActor" type="UC:Ref_Actor" minOccurs="0" maxOccurs="unbounded">
  <xsd:annotation>
    <xsd:documentation>It describes which actor triggers this
scenario.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="Requirement" type="UC:Ref_Requirement" minOccurs="0"
maxOccurs="unbounded">
  <xsd:annotation>
    <xsd:documentation>information about the requirement</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="TriggeringEvent" type="UC:TriggeringEvent" minOccurs="0"
maxOccurs="unbounded">
  <xsd:annotation>
    <xsd:documentation>list of triggering events for the scenario</xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="TriggeringEvent">
  <xsd:annotation>
    <xsd:documentation>It describes which event triggers a scenario.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>description of the object</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>a short name</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="UseCase">
  <xsd:annotation>
    <xsd:documentation>specification of a set of actions performed by a system, which yields an
observable result that is, typically, of value for one or more actors or other stakeholders of the system </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="ActorGrouping" type="UC:ActorGrouping" minOccurs="0"
maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>A use case references several sets of
actors.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="Assumption" type="UC:Assumption" minOccurs="0" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>set of assumptions related to the use case</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="BusinessCase" type="UC:Ref_BusinessCase" minOccurs="0"
maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>list of business cases associated with the use
case</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="classification" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>At the international level the use case description might be
generic enough to describe a use case in a more general way independently from the national or regional market design. But
use cases might be used to describe regional or national specific circumstances like laws or even project specific details. If the
use case reflects those circumstances, it should be characterized accordingly.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="CommonTerm" type="UC:Ref_CommonTerm" minOccurs="0"

```

```

maxOccurs="unbounded">
    <xsd:annotation>
        <xsd:documentation>the associated common terms</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="CustomInformation" type="UC:CustomInformation" minOccurs="0"
maxOccurs="unbounded">
    <xsd:annotation>
        <xsd:documentation>related additional information</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="Drawing" type="UC:Drawing" minOccurs="0" maxOccurs="unbounded">
    <xsd:annotation>
        <xsd:documentation>the list of associated drawings</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="identifier" type="xsd:string" minOccurs="1" maxOccurs="1">
    <xsd:annotation>
        <xsd:documentation>The identification number (ID) of a use case is unique within a
repository or project and serves for organization/administration of use cases.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="KeyPerformanceIndicator" type="UC:KeyPerformanceIndicator" minOccurs="0"
maxOccurs="unbounded">
    <xsd:annotation>
        <xsd:documentation>information about the KPI related to the use
case</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="keywords" type="xsd:string" minOccurs="0" maxOccurs="1">
    <xsd:annotation>
        <xsd:documentation>Keywords can be defined in order to support extended search
functionalities within a use case repository. Multiple keywords should be provided as a comma-separated
list.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="levelOfDepth" type="xsd:string" minOccurs="0" maxOccurs="1">
    <xsd:annotation>
        <xsd:documentation>Use cases can be described on different
levels.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
    <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="name" type="xsd:string" minOccurs="1" maxOccurs="1">
    <xsd:annotation>
        <xsd:documentation>a short name</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="Narrative" type="UC:Narrative" minOccurs="0" maxOccurs="1">
    <xsd:annotation>
        <xsd:documentation>the information about the narrative part of the use
case</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="nature" type="xsd:string" minOccurs="0" maxOccurs="1">
    <xsd:annotation>
        <xsd:documentation>It classifies the main focus of the use case.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="Prerequisite" type="UC:Condition" minOccurs="0" maxOccurs="unbounded">
    <xsd:annotation>
        <xsd:documentation>It describes what condition(s) should have been met before
initiation of the use case, such as prior state of the actors and activities.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="PrimaryActor" type="UC:Ref_Actor" minOccurs="0" maxOccurs="unbounded">
    <xsd:annotation>
        <xsd:documentation>identification of the PrimaryActor linked to the use
case</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="prioritization" type="xsd:string" minOccurs="0" maxOccurs="1">
    <xsd:annotation>

```

```

<xsd:documentation>If considering a larger number of use cases, it might be
interesting to cluster them according to priority.
This prioritization might be different from country to country.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="Reference" type="UC:Reference" minOccurs="0" maxOccurs="unbounded">
  <xsd:annotation>
    <xsd:documentation>information about referenced documents</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="RelatedObjective" type="UC:Objective" minOccurs="0"
maxOccurs="unbounded">
  <xsd:annotation>
    <xsd:documentation>list of objectives of the use case</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="RelatedUseCase" type="UC:Ref_UseCase" minOccurs="0"
maxOccurs="unbounded">
  <xsd:annotation>
    <xsd:documentation>Known relations to other use cases can be provided here if, for
example, the use case is a more detailed one related to a high level use case, or it is an alternative to an existing use
case.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="Remark" type="UC:Remark" minOccurs="0" maxOccurs="unbounded">
  <xsd:annotation>
    <xsd:documentation>remark associated with the use case</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="Scenario" type="UC:Scenario" minOccurs="0" maxOccurs="unbounded">
  <xsd:annotation>
    <xsd:documentation>reference to the scenarios belonging to the use
case</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="scope" type="xsd:string" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>The scope defines the limits of the use
case.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="Version" type="UC:Version" minOccurs="0" maxOccurs="unbounded">
  <xsd:annotation>
    <xsd:documentation>list of versions describing changes about the use
case</xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Version">
  <xsd:annotation>
    <xsd:documentation>version of the use case description</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="approvalStatus" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>information used within the standardization
process</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="Author" type="UC:Author" minOccurs="0" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>list of authors participating in the changes of this
version</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="changes" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>It represents the differences with respect to the previous version.
Multiple changes are separated with paragraphs.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="date" type="xsd:dateTime" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>date of creation of the version, in the format YYYY-MM-
DD</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```

```

</xsd:element>
<xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>master resource identifier</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="number" type="xsd:string" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>sequential number to identify the version of the
document</xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="UseCaseLibrary">
  <xsd:annotation>
    <xsd:documentation>use cases of the repository</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>a short name</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="UseCase" type="UC:UseCase" minOccurs="0" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>list of use cases within the library</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="UseCaseRepository_Type">
  <xsd:annotation>
    <xsd:documentation>database, based on a given use cases template, for editing, maintenance and
administration of use cases, actors and requirements including their interrelations</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="ActorLibrary" type="UC:ActorLibrary" minOccurs="0" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>library of actors</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="AreaLibrary" type="UC:AreaLibrary" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>the associated library of areas</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="BusinessCaseLibrary" type="UC:BusinessCaseLibrary" minOccurs="0"
maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>the library associated with the repository</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="BusinessObjectLibrary" type="UC:BusinessObjectLibrary" minOccurs="0"
maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>the library of business objects</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="CommonTermLibrary" type="UC:CommonTermLibrary" minOccurs="0"
maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>the associated library of common terms</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>description of the object</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="name" type="xsd:string" minOccurs="1" maxOccurs="1">
      <xsd:annotation>

```

```

        <xsd:documentation>a short name</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="RequirementLibrary" type="UC:RequirementLibrary" minOccurs="0"
maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>the associated library</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="UseCaseLibrary" type="UC:UseCaseLibrary" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>the associated library of use cases</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="UseCaseRepository" type="UC:UseCaseRepository_Type"/>
<xsd:complexType name="AreaLibrary">
  <xsd:annotation>
    <xsd:documentation>areas of the repository</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="Area" type="UC:Area" minOccurs="0" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>the list of areas within the library</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>a short name</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Assumption">
  <xsd:annotation>
    <xsd:documentation>It may be used to define a further general assumption for a use case, such as
which systems already exist, which contractual relations exist, and which configurations of systems are probably in place. Initial
states of information exchanged shall also be identified.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>description of the object</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="mRID" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>a short name</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="number" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>index of the assumption</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

<CODE ENDS>

6.3 Actor library exchange Format

6.3.1 UML actor library exchange model (UML Contextual Model)

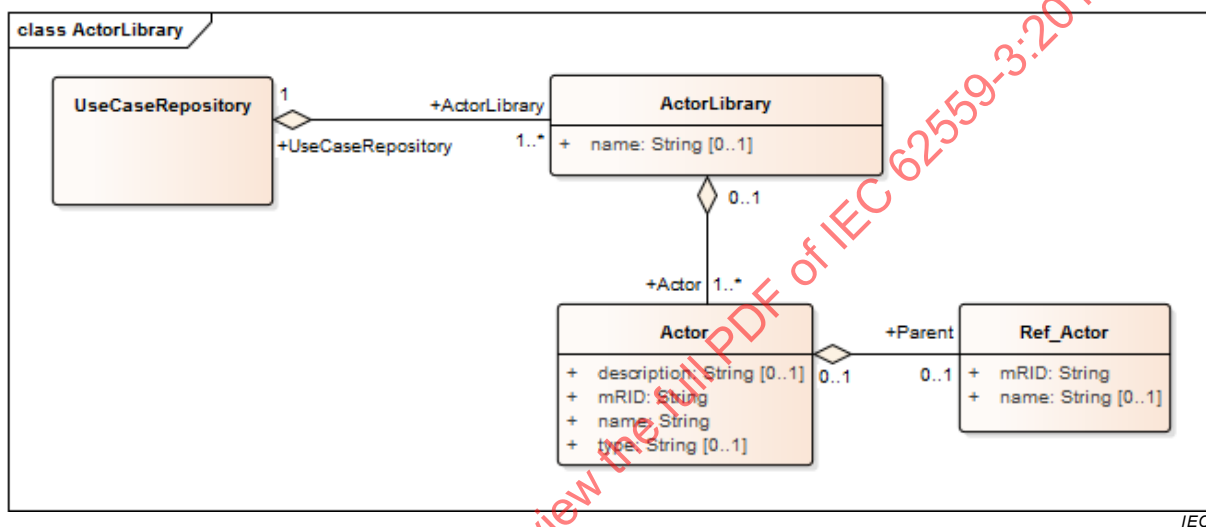
6.3.1.1 General

The Actor library exchange model is described through a UML Contextual Model. The XSD content is generated from this model.

6.3.1.2 Model package ActorLibrary_ExchangeModel

6.3.1.2.1 General

Figure 23 shows class diagram ActorLibrary.



IEC

Figure 23 – Class diagram ActorLibrary_ExchangeModel::ActorLibrary

Figure 24 shows package diagram ActorLibrary.

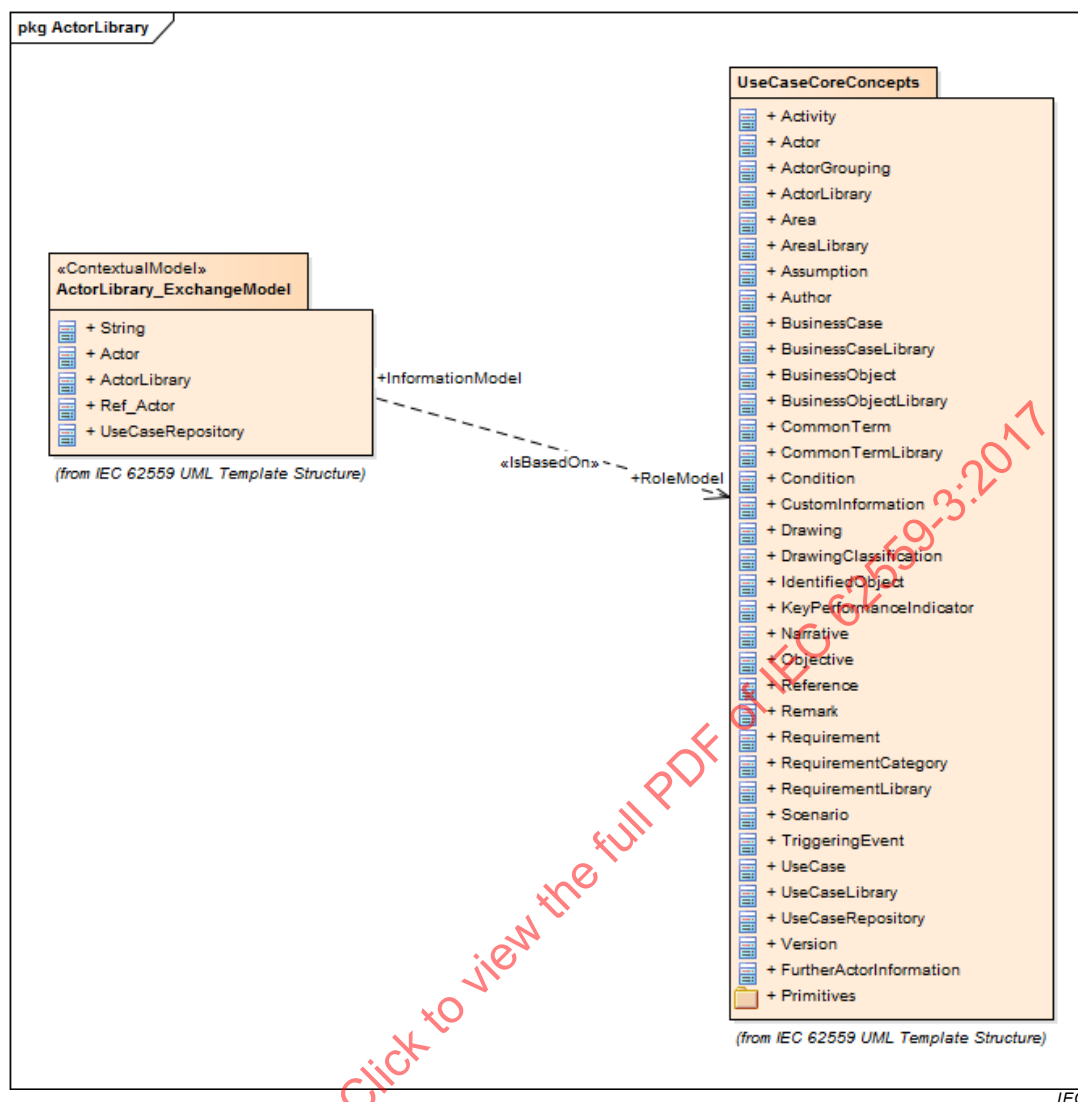


Figure 24 – Package diagram ActorLibrary_ExchangeModel::ActorLibrary

6.3.1.2.2 String primitive

String Primitive type

6.3.1.2.3 Actor root class

It is the entity that communicates and interacts.

These Actors can include people, software applications, systems, databases, and even the power system itself.

Table 151 shows all attributes of Actor.

Table 151 – Attributes of ActorLibrary_ExchangeModel::Actor

name	type	description
type	String	type of the Actor (business, system, ...)
description	String	description of the object
mRID	String	master resource identifier
name	String	a short name

Table 152 shows all association ends of Actor with other classes.

Table 152 – Association ends of ActorLibrary_ExchangeModel::Actor with other classes

[mult from]	[mult to] name	type	description
[1..*]	[0..1]	ActorLibrary	This association end is an aggregation.
[0..1]	[0..1] Parent	Ref_Actor	the generalized Actor

6.3.1.2.4 ActorLibrary root class

Actors of the repository.

Table 153 shows all attributes of ActorLibrary.

Table 153 – Attributes of ActorLibrary_ExchangeModel::ActorLibrary

name	type	description
name	String	a short name

Table 154 shows all association ends of ActorLibrary with other classes.

Table 154 – Association ends of ActorLibrary_ExchangeModel::ActorLibrary with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..*] Actor	Actor	the Actors listed in the library
[1..*]	[1..1] UseCaseRepository	UseCaseRepository	This association end is an aggregation.

6.3.1.2.5 Ref_Actor root class

It is the entity that communicates and interacts.

These Actors can include people, software applications, systems, databases, and even the power system itself.

Table 155 shows all attributes of Ref_Actor.

Table 155 – Attributes of ActorLibrary_ExchangeModel::Ref_Actor

name	type	description
mRID	String	master resource identifier
name	String	a short name

Table 156 shows all association ends of Ref_Actor with other classes.

Table 156 – Association ends of ActorLibrary_ExchangeModel::Ref_Actor with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..1]	Actor	This association end is an aggregation.

6.3.1.2.6 UseCaseRepository root class

Database, based on a given use cases template, for editing, maintenance and administration of use cases, Actors and requirements including their interrelations.

Table 157 shows all association ends of UseCaseRepository with other classes.

Table 157 – Association ends of ActorLibrary_ExchangeModel::UseCaseRepository with other classes

[mult from]	[mult to] name	type	description
[1..1]	[1..*] ActorLibrary	ActorLibrary	library of Actors

6.3.2 Actor library XML syntactic format

Table 158 – Actor library XSD

<CODE BEGINS>

XSD content
<pre> <?xml version="1.0" encoding="utf-8"?> <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:AL="http://www.SYC.org/IEC62559/part3/AL/V01/Build08082016" targetNamespace="http://www.SYC.org/IEC62559/part3/AL/V01/Build08082016"> <xsd:complexType name="Actor"> <xsd:annotation> <xsd:documentation>It is the entity that communicates and interacts. These actors can include people, software applications, systems, databases, and even the power system itself.</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1"> <xsd:annotation> <xsd:documentation>description of the object</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="mRID" type="xsd:string" minOccurs="1" maxOccurs="1"> <xsd:annotation> <xsd:documentation>master resource identifier</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"> <xsd:annotation> <xsd:documentation>a short name</xsd:documentation> </xsd:annotation> </xsd:element> </xsd:sequence> </xsd:complexType> </xsd:schema> </pre>

```

</xsd:element>
<xsd:element name="Parent" type="AL:Ref_Actor" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>the generalized actor</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="type" type="xsd:string" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>type of the actor (business, system, ...)</xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Ref_Actor">
  <xsd:annotation>
    <xsd:documentation>It is the entity that communicates and interacts.
    These actors can include people, software applications, systems, databases, and even the power system
    itself.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="mRID" type="xsd:string" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>a short name</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ActorLibrary">
  <xsd:annotation>
    <xsd:documentation>actors of the repository</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="Actor" type="AL:Actor" minOccurs="1" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>the actors listed in the library</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>a short name</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="UseCaseRepository_Type">
  <xsd:annotation>
    <xsd:documentation>database, based on a given use cases template, for editing, maintenance and
    administration of use cases, actors and requirements including their interrelations</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="ActorLibrary" type="AL:ActorLibrary" minOccurs="1" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>library of actors</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="UseCaseRepository" type="AL:UseCaseRepository_Type"/>
</xsd:schema>

```

<CODE ENDS>

6.4 Requirement library exchange Format

6.4.1 UML requirement library exchange model (UML Contextual Model)

6.4.1.1 General

The requirement library exchange model is described through a UML Contextual Model. The XSD content is generated from this model.

6.4.1.2 Model package RequirementLibrary_ExchangeModel

6.4.1.2.1 General

Figure 25 shows class diagram RequirementLibrary.

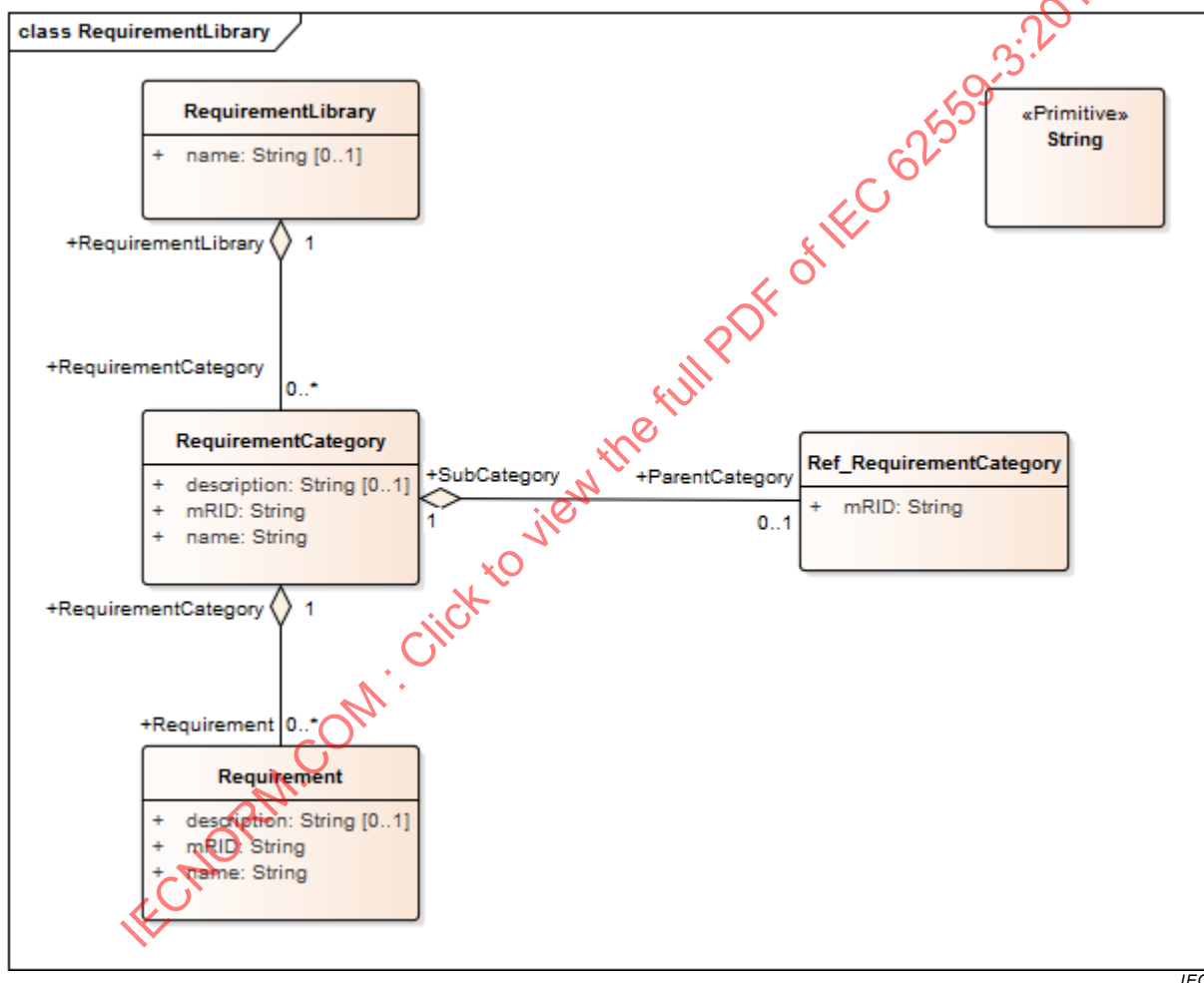
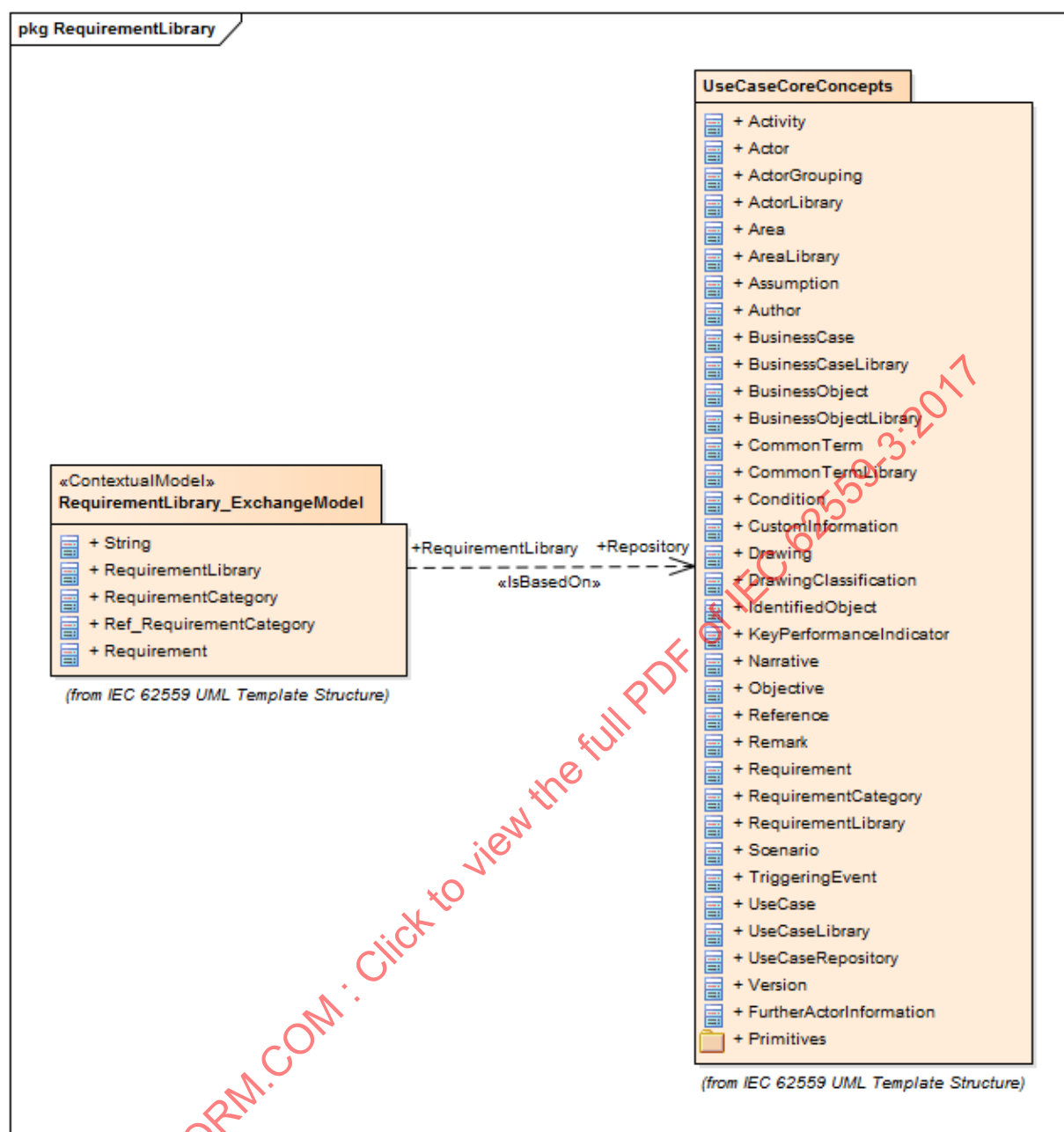


Figure 25 – Class diagram RequirementLibrary_ExchangeModel::RequirementLibrary

Figure 26 shows package diagram RequirementLibrary.



IEC

Figure 26 – Package diagram RequirementLibrary_ExchangeModel::RequirementLibrary

6.4.1.2.2 String primitive

String Primitive type

6.4.1.2.3 RequirementLibrary root class

Requirements of the repository.

Table 159 shows all attributes of RequirementLibrary.

Table 159 – Attributes of RequirementLibrary_ExchangeModel::RequirementLibrary

name	type	description
name	String	a short name

Table 160 shows all association ends of RequirementLibrary with other classes.

Table 160 – Association ends of RequirementLibrary_ExchangeModel::RequirementLibrary with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] RequirementCategory	RequirementCategory	the category for the requirement

6.4.1.2.4 RequirementCategory root class

Requirements can be sorted in categories. For complex categories the category name can be "dot-delimited".

Table 161 shows all attributes of RequirementCategory.

Table 161 – Attributes of RequirementLibrary_ExchangeModel::RequirementCategory

name	type	description
description	String	description of the object
mRID	String	master resource identifier
name	String	a short name

Table 162 shows all association ends of RequirementCategory with other classes.

Table 162 – Association ends of RequirementLibrary_ExchangeModel::RequirementCategory with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] RequirementLibrary	RequirementLibrary	This association end is an aggregation.
[1..1]	[0..1] ParentCategory	Ref_RequirementCategory	the reference to the parent category of requirement
[1..1]	[0..*] Requirement	Requirement	information about the requirement

6.4.1.2.5 Ref_RequirementCategory root class

Requirements can be sorted in categories. For complex categories the category name can be "dot-delimited".

Table 163 shows all attributes of Ref_RequirementCategory.

Table 163 – Attributes of RequirementLibrary_ExchangeModel::Ref_RequirementCategory

name	type	description
mRID	String	master resource identifier

Table 164 shows all association ends of Ref_RequirementCategory with other classes.

Table 164 – Association ends of RequirementLibrary_ExchangeModel::Ref_RequirementCategory with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] SubCategory	RequirementCategory	This association end is an aggregation.

6.4.1.2.6 Requirement root class

Requirement of a part of the use case.

Table 165 shows all attributes of Requirement.

Table 165 – Attributes of RequirementLibrary_ExchangeModel::Requirement

name	type	description
description	String	description of the object
mRID	String	master resource identifier
name	String	a short name

Table 166 shows all association ends of Requirement with other classes.

Table 166 – Association ends of RequirementLibrary_ExchangeModel::Requirement with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] RequirementCategory	RequirementCategory	This association end is an aggregation.

6.4.2 Requirement library XML syntactic format**Table 167 – Requirement library XSD**

<CODE BEGINS>

XSD content
<pre> <?xml version="1.0" encoding="utf-8"?> <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:RL="http://www.SYC.org/IEC62559/part3/RL/V01/Build08082016" targetNamespace="http://www.SYC.org/IEC62559/part3/RL/V01/Build08082016"> <xsd:complexType name="RequirementLibrary_Type"> <xsd:annotation> <xsd:documentation>requirements of the repository</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1"> <xsd:annotation> <xsd:documentation>a short name</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="RequirementCategory" type="RL:RequirementCategory" minOccurs="0" maxOccurs="unbounded"> <xsd:annotation> <xsd:documentation>the category for the requirement</xsd:documentation> </xsd:annotation> </xsd:element> </xsd:sequence> </xsd:complexType> <xsd:complexType name="RequirementCategory"> <xsd:annotation> <xsd:documentation>Requirements can be sorted in categories. For complex categories the category </pre>

```

name can be "dot-delimited".</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
  <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1">
    <xsd:annotation>
      <xsd:documentation>description of the object</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="mRID" type="xsd:string" minOccurs="1" maxOccurs="1">
    <xsd:annotation>
      <xsd:documentation>master resource identifier</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="name" type="xsd:string" minOccurs="1" maxOccurs="1">
    <xsd:annotation>
      <xsd:documentation>a short name</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="ParentCategory" type="RL:Ref_RequirementCategory" minOccurs="0"
maxOccurs="1">
    <xsd:annotation>
      <xsd:documentation>the reference to the parent category of
requirement</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="Requirement" type="RL:Requirement" minOccurs="0"
maxOccurs="unbounded">
    <xsd:annotation>
      <xsd:documentation>information about the requirement</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Ref_RequirementCategory">
  <xsd:annotation>
    <xsd:documentation>Requirements can be sorted in categories. For complex categories the category
name can be "dot-delimited".</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="mRID" type="xsd:string" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Requirement">
  <xsd:annotation>
    <xsd:documentation>requirement of a part of the use case</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>description of the object</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="mRID" type="xsd:string" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>master resource identifier</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="name" type="xsd:string" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>a short name</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="RequirementLibrary" type="RL:RequirementLibrary_Type"/>
</xsd:schema>

```

<CODE ENDS>

Bibliography

- [1] SMB Smart Grid Strategic Group (SG3), *IEC smart grid standardization roadmap* (edition 1.0), June 2010
 - [2] CEN-CENELEC-ETSI Smart Grid Coordination Group – Working Group Sustainable Processes, Draft Report: *Use Case Collection, Management, Repository, Analysis and Harmonization*
 - [3] IEC 61968-100, *Application integration at electric utilities – System interfaces for distribution management – Part 100: Implementation profiles*
 - [4] IEC PAS 62559:2008, *IntelliGrid methodology for developing requirements for energy systems*
 - [5] IEC 62325-450, *Framework for energy market communications – Part 450: Profile and context modelling rules*
 - [6] ISO/IEC 19505-1:2012, *Information technology – Object Management Group Unified Modeling Language (OMG UML) – Part 1: Infrastructure*
 - [7] ISO/IEC 19505-2:2012, *Information technology – Object Management Group Unified Modeling Language (OMG UML) – Part 2: Superstructure*
 - [8] W3C XML Schema Definition Language (XSD) 1.1 – Part 1: Structures (<http://www.w3.org/TR/xmlschema11-1/>)
 - [9] W3C XML Schema Definition Language (XSD) 1.1 – Part 2: Datatypes (<http://www.w3.org/TR/xmlschema11-2/>)
-

SOMMAIRE

AVANT-PROPOS	111
INTRODUCTION.....	113
1 Domaine d'application	116
2 Références normatives	117
3 Termes et définitions	117
4 Cadre méthodologique pour le développement de la présente Norme	118
4.1 Approche basée sur des modèles pour le développement du format d'échange de données	118
4.2 Exemple de processus basé sur des modèles dans le contexte de l'IEC 62559	120
4.3 Développement de profils particuliers d'échange de données.....	123
5 Application du cadre méthodologique de définition d'un ensemble de formats d'échange syntaxiques clés (cas d'utilisation, acteurs, exigences).....	124
6 Contenu détaillé des modèles (génération automatique à partir de modèles UML).....	124
6.1 Concepts clés de cas d'utilisation UML (modèle d'information UML)	124
6.1.1 Généralités	124
6.1.2 Paquetage de modèle UseCaseCoreConcepts.....	124
6.2 Format d'échange de bibliothèque de cas d'utilisation.....	151
6.2.1 Modèle d'échange de bibliothèque de cas d'utilisation UML (modèle contextuel UML)	151
6.2.2 Format syntaxique XML de bibliothèque de cas d'utilisation.....	181
6.3 Format d'échange de bibliothèque d'acteurs	200
6.3.1 Modèle d'échange de bibliothèque d'acteurs UML (modèle contextuel UML)	200
6.3.2 Format syntaxique XML de bibliothèque d'acteurs	203
6.4 Format d'échange de bibliothèque d'exigences	205
6.4.1 Modèle d'échange de bibliothèque d'exigences UML (modèle contextuel UML)	205
6.4.2 Format syntaxique XML de bibliothèque d'exigences	209
Bibliographie.....	211
Figure 1 – Série de normes IEC 62559	113
Figure 2 – Format XML commun d'import/export d'informations de cas d'utilisation entre différents logiciels de modélisation et référentiels	116
Figure 3 – Cadre de développement basé sur des modèles pour la définition du format d'échange de données XML de la norme IEC 62559-3	119
Figure 4 – Modèles d'information UML pour le scénario, exemple de relation Actor.....	120
Figure 5 – Modèle contextuel UML basé sur l'exemple de relation Scenario-Actor	121
Figure 6 – Schéma XML correspondant à l'exemple de relation Scenario-Actor	123
Figure 7 – Production de formats d'échange XML clés basés sur un modèle d'information UML IEC 62559 unique	124
Figure 8 – Diagramme de classe UseCaseCoreConcepts::UseCaseCore_Libraries.....	125
Figure 9 – Diagramme de classe UseCaseCoreConcepts::UseCaseCore_ActorLibrary	125
Figure 10 – Diagramme de classe UseCaseCoreConcepts::UseCaseCore_UseCaseLibrary	126
Figure 11 – Diagramme de classe UseCaseCoreConcepts::UseCaseCore_DetailedActivities.....	127

Figure 12 – Diagramme de classe UseCaseCoreConcepts::UseCaseCore_AreaLibrary	127
Figure 13 – Diagramme de classe UseCaseCoreConcepts::UseCaseCore_BusinessCaseLibrary	128
Figure 14 – Diagramme de classe UseCaseCoreConcepts::UseCaseCore_BusinessObjectLibrary	128
Figure 15 – Diagramme de classe UseCaseCoreConcepts::UseCaseCore_CommonTermLibrary	129
Figure 16 – Diagramme de classe UseCaseCoreConcepts::UseCaseCore_RequirementLibrary	129
Figure 17 – Diagramme de classe Primitives::Primitives	150
Figure 18 – Diagramme de paquetage UseCase_ExchangeModel::ContextualModel_Dependency	152
Figure 19 – Diagramme de classe UseCase_ExchangeModel::UseCase_Repository	153
Figure 20 – Diagramme de classe UseCase_ExchangeModel::UseCase_ShortDescription	154
Figure 21 – Diagramme de classe UseCase_ExchangeModel::UseCase_DetailedActivities	155
Figure 22 – Diagramme de classe UseCase_ExchangeModel::UseCase_CompleteDescription	156
Figure 23 – Diagramme de classe ActorLibrary_ExchangeModel::ActorLibrary	200
Figure 24 – Diagramme de paquetage ActorLibrary_ExchangeModel::ActorLibrary	201
Figure 25 – Diagramme de classe RequirementLibrary_ExchangeModel::RequirementLibrary	205
Figure 26 – Diagramme de paquetage RequirementLibrary_ExchangeModel::RequirementLibrary	206
Tableau 1 – Attributs de UseCaseCoreConcepts::Activity	130
Tableau 2 – Extrémités d'association de UseCaseCoreConcepts:: Activity avec d'autres classes	131
Tableau 3 – Attributs de UseCaseCoreConcepts::Actor	131
Tableau 4 – Extrémités d'association de UseCaseCoreConcepts:: Actor avec d'autres classes	132
Tableau 5 – Attributs de UseCaseCoreConcepts::ActorGrouping	132
Tableau 6 – Extrémités d'association de UseCaseCoreConcepts:: ActorGrouping avec d'autres classes	132
Tableau 7 – Attributs de UseCaseCoreConcepts::ActorLibrary	133
Tableau 8 – Extrémités d'association de UseCaseCoreConcepts:: ActorLibrary avec d'autres classes	133
Tableau 9 – Attributs de UseCaseCoreConcepts::Area	133
Tableau 10 – Extrémités d'association de UseCaseCoreConcepts:: Area avec d'autres classes	133
Tableau 11 – Attributs de UseCaseCoreConcepts::AreaLibrary	134
Tableau 12 – Extrémités d'association de UseCaseCoreConcepts:: AreaLibrary avec d'autres classes	134
Tableau 13 – Attributs de UseCaseCoreConcepts::Assumption	134
Tableau 14 – Extrémités d'association de UseCaseCoreConcepts:: Assumption avec d'autres classes	134
Tableau 15 – Attributs de UseCaseCoreConcepts::Author	135

Tableau 16 – Extrémités d'association de UseCaseCoreConcepts:: Author avec d'autres classes	135
Tableau 17 – Attributs de UseCaseCoreConcepts::BusinessCase.....	135
Tableau 18 – Extrémités d'association de UseCaseCoreConcepts::BusinessCase avec d'autres classes	135
Tableau 19 – Attributs de UseCaseCoreConcepts::BusinessCaseLibrary	136
Tableau 20 – Extrémités d'association de UseCaseCoreConcepts:: BusinessCaseLibrary avec d'autres classes	136
Tableau 21 – Attributs de UseCaseCoreConcepts::BusinessObject.....	136
Tableau 22 – Extrémités d'association de UseCaseCoreConcepts:: BusinessObject avec d'autres classes.....	136
Tableau 23 – Attributs de UseCaseCoreConcepts::BusinessObjectLibrary	137
Tableau 24 – Extrémités d'association de UseCaseCoreConcepts:: BusinessObjectLibrary avec d'autres classes	137
Tableau 25 – Attributs de UseCaseCoreConcepts::CommonTerm	137
Tableau 26 – Extrémités d'association de UseCaseCoreConcepts:: CommonTerm avec d'autres classes	137
Tableau 27 – Attributs de UseCaseCoreConcepts::CommonTermLibrary	138
Tableau 28 – Extrémités d'association de UseCaseCoreConcepts:: CommonTermLibrary avec d'autres classes	138
Tableau 29 – Attributs de UseCaseCoreConcepts::Condition	138
Tableau 30 – Extrémités d'association de UseCaseCoreConcepts:: Condition avec d'autres classes	138
Tableau 31 – Attributs de UseCaseCoreConcepts::CustomInformation	139
Tableau 32 – Extrémités d'association de UseCaseCoreConcepts:: CustomInformation avec d'autres classes.....	139
Tableau 33 – Attributs de UseCaseCoreConcepts::Drawing	139
Tableau 34 – Extrémités d'association de UseCaseCoreConcepts:: Drawing avec d'autres classes	140
Tableau 35 – Littéraux de UseCaseCoreConcepts::DrawingClassification	140
Tableau 36 – Attributs de UseCaseCoreConcepts::FurtherActorInformation	140
Tableau 37 – Extrémités d'association de UseCaseCoreConcepts:: FurtherActorInformation avec d'autres classes	141
Tableau 38 – Attributs de UseCaseCoreConcepts::IdentifiedObject	141
Tableau 39 – Attributs de UseCaseCoreConcepts::KeyPerformanceIndicator.....	141
Tableau 40 – Extrémités d'association de UseCaseCoreConcepts:: KeyPerformanceIndicator avec d'autres classes	141
Tableau 41 – Attributs de UseCaseCoreConcepts::Narrative	142
Tableau 42 – Extrémités d'association de UseCaseCoreConcepts:: Narrative avec d'autres classes	142
Tableau 43 – Attributs de UseCaseCoreConcepts::Objective	142
Tableau 44 – Extrémités d'association de UseCaseCoreConcepts:: Objective avec d'autres classes	143
Tableau 45 – Attributs de UseCaseCoreConcepts::Reference.....	143
Tableau 46 – Extrémités d'association de UseCaseCoreConcepts:: Reference avec d'autres classes	143
Tableau 47 – Attributs de UseCaseCoreConcepts::Remark.....	144

Tableau 48 – Extrémités d'association de UseCaseCoreConcepts:: Remark avec d'autres classes	144
Tableau 49 – Attributs de UseCaseCoreConcepts::Requirement	144
Tableau 50 – Extrémités d'association de UseCaseCoreConcepts:: Requirement avec d'autres classes	144
Tableau 51 – Attributs de UseCaseCoreConcepts::RequirementCategory	145
Tableau 52 – Extrémités d'association de UseCaseCoreConcepts:: RequirementCategory avec d'autres classes	145
Tableau 53 – Attributs de UseCaseCoreConcepts::RequirementLibrary	145
Tableau 54 – Extrémités d'association de UseCaseCoreConcepts:: RequirementLibrary avec d'autres classes	145
Tableau 55 – Attributs de UseCaseCoreConcepts::Scenario	146
Tableau 56 – Extrémités d'association de UseCaseCoreConcepts:: Scenario avec d'autres classes	146
Tableau 57 – Attributs de UseCaseCoreConcepts::TriggeringEvent	146
Tableau 58 – Extrémités d'association de UseCaseCoreConcepts:: TriggeringEvent avec d'autres classes	147
Tableau 59 – Attributs de UseCaseCoreConcepts::UseCase	147
Tableau 60 – Extrémités d'association de UseCaseCoreConcepts:: UseCase avec d'autres classes	148
Tableau 61 – Attributs de UseCaseCoreConcepts::UseCaseLibrary	148
Tableau 62 – Extrémités d'association de UseCaseCoreConcepts:: UseCaseLibrary avec d'autres classes	149
Tableau 63 – Attributs de UseCaseCoreConcepts::UseCaseRepository	149
Tableau 64 – Extrémités d'association de UseCaseCoreConcepts:: UseCaseRepository avec d'autres classes	149
Tableau 65 – Attributs de UseCaseCoreConcepts::Version	150
Tableau 66 – Extrémités d'association de UseCaseCoreConcepts:: Version avec d'autres classes	150
Tableau 67 – Attributs de UseCase_ExchangeModel::Activity	157
Tableau 68 – Extrémités d'association de UseCase_ExchangeModel:: Activity avec d'autres classes	158
Tableau 69 – Attributs de UseCase_ExchangeModel::Actor	158
Tableau 70 – Extrémités d'association de UseCase_ExchangeModel:: Actor avec d'autres classes	159
Tableau 71 – Attributs de UseCase_ExchangeModel::ActorGrouping	159
Tableau 72 – Extrémités d'association de UseCase_ExchangeModel:: ActorGrouping avec d'autres classes	159
Tableau 73 – Attributs de UseCase_ExchangeModel::ActorLibrary	159
Tableau 74 – Extrémités d'association de UseCase_ExchangeModel:: ActorLibrary avec d'autres classes	160
Tableau 75 – Attributs de UseCase_ExchangeModel::Area	160
Tableau 76 – Extrémités d'association de UseCase_ExchangeModel:: Area avec d'autres classes	160
Tableau 77 – Attributs de UseCase_ExchangeModel::AreaLibrary	160
Tableau 78 – Extrémités d'association de UseCase_ExchangeModel:: AreaLibrary avec d'autres classes	161
Tableau 79 – Attributs de UseCase_ExchangeModel::Assumption	161

Tableau 80 – Extrémités d'association de UseCase_ExchangeModel:: Assumption avec d'autres classes.....	161
Tableau 81 – Attributs de UseCase_ExchangeModel::Author.....	161
Tableau 82 – Extrémités d'association de UseCase_ExchangeModel:: Author avec d'autres classes.....	162
Tableau 83 – Attributs de UseCase_ExchangeModel::BusinessCase.....	162
Tableau 84 – Extrémités d'association de UseCase_ExchangeModel:: BusinessCase avec d'autres classes.....	162
Tableau 85 – Attributs de UseCase_ExchangeModel::BusinessCaseLibrary.....	162
Tableau 86 – Extrémités d'association de UseCase_ExchangeModel:: BusinessCaseLibrary avec d'autres classes.....	163
Tableau 87 – Attributs de UseCase_ExchangeModel::BusinessObject.....	163
Tableau 88 – Extrémités d'association de UseCase_ExchangeModel:: BusinessObject avec d'autres classes.....	163
Tableau 89 – Attributs de UseCase_ExchangeModel::BusinessObjectLibrary.....	163
Tableau 90 – Extrémités d'association de UseCase_ExchangeModel:: BusinessObjectLibrary avec d'autres classes.....	164
Tableau 91 – Attributs de UseCase_ExchangeModel::CommonTerm.....	164
Tableau 92 – Extrémités d'association de UseCase_ExchangeModel:: CommonTerm avec d'autres classes.....	164
Tableau 93 – Attributs de UseCase_ExchangeModel::CommonTermLibrary.....	164
Tableau 94 – Extrémités d'association de UseCase_ExchangeModel:: CommonTermLibrary avec d'autres classes.....	165
Tableau 95 – Attributs de UseCase_ExchangeModel::Condition.....	165
Tableau 96 – Extrémités d'association de UseCase_ExchangeModel:: Condition avec d'autres classes.....	165
Tableau 97 – Attributs de UseCase_ExchangeModel::CustomInformation.....	165
Tableau 98 – Extrémités d'association de UseCase_ExchangeModel:: CustomInformation avec d'autres classes.....	166
Tableau 99 – Attributs de UseCase_ExchangeModel::Drawing.....	166
Tableau 100 – Extrémités d'association de UseCase_ExchangeModel:: Drawing avec d'autres classes.....	166
Tableau 101 – Littéraux de UseCase_ExchangeModel::DrawingClassification.....	167
Tableau 102 – Attributs de UseCase_ExchangeModel::KeyPerformanceIndicator.....	167
Tableau 103 – Extrémités d'association de UseCase_ExchangeModel:: KeyPerformanceIndicator avec d'autres classes.....	167
Tableau 104 – Attributs de UseCase_ExchangeModel::Narrative.....	168
Tableau 105 – Extrémités d'association de UseCase_ExchangeModel:: Narrative avec d'autres classes.....	168
Tableau 106 – Attributs de UseCase_ExchangeModel::Objective.....	168
Tableau 107 – Extrémités d'association de UseCase_ExchangeModel:: Objective avec d'autres classes.....	169
Tableau 108 – Attributs de UseCase_ExchangeModel::Ref_Actor.....	169
Tableau 109 – Extrémités d'association de UseCase_ExchangeModel:: Ref_Actor avec d'autres classes.....	169
Tableau 110 – Attributs de UseCase_ExchangeModel::Ref_Area.....	170
Tableau 111 – Extrémités d'association de UseCase_ExchangeModel:: Ref_Area avec d'autres classes.....	170

Tableau 112 – Attributs de UseCase_ExchangeModel::Ref_BusinessCase	170
Tableau 113 – Extrémités d'association de UseCase_ExchangeModel:: Ref_BusinessCase avec d'autres classes	170
Tableau 114 – Attributs de UseCase_ExchangeModel::Ref_BusinessObject	170
Tableau 115 – Extrémités d'association de UseCase_ExchangeModel:: Ref_BusinessObject avec d'autres classes	171
Tableau 116 – Attributs de UseCase_ExchangeModel::Ref_CommonTerm	171
Tableau 117 – Extrémités d'association de UseCase_ExchangeModel:: Ref_CommonTerm avec d'autres classes	171
Tableau 118 – Attributs de UseCase_ExchangeModel::Ref_Objective	171
Tableau 119 – Extrémités d'association de UseCase_ExchangeModel:: Ref_Objective avec d'autres classes	171
Tableau 120 – Attributs de UseCase_ExchangeModel::Ref_Requirement	172
Tableau 121 – Extrémités d'association de UseCase_ExchangeModel:: Ref_Requirement avec d'autres classes	172
Tableau 122 – Attributs de UseCase_ExchangeModel::Ref_UseCase	172
Tableau 123 – Extrémités d'association de UseCase_ExchangeModel:: Ref_UseCase avec d'autres classes	172
Tableau 124 – Attributs de UseCase_ExchangeModel::Reference	173
Tableau 125 – Extrémités d'association de UseCase_ExchangeModel:: Reference avec d'autres classes	173
Tableau 126 – Attributs de UseCase_ExchangeModel::Remark	173
Tableau 127 – Extrémités d'association de UseCase_ExchangeModel:: Remark avec d'autres classes	174
Tableau 128 – Attributs de UseCase_ExchangeModel::Requirement	174
Tableau 129 – Extrémités d'association de UseCase_ExchangeModel:: Requirement avec d'autres classes	174
Tableau 130 – Attributs de UseCase_ExchangeModel::RequirementCategory	174
Tableau 131 – Extrémités d'association de UseCase_ExchangeModel:: RequirementCategory avec d'autres classes	175
Tableau 132 – Attributs de UseCase_ExchangeModel::RequirementLibrary	175
Tableau 133 – Extrémités d'association de UseCase_ExchangeModel:: RequirementLibrary avec d'autres classes	175
Tableau 134 – Attributs de UseCase_ExchangeModel::Resource_String	175
Tableau 135 – Littéraux de UseCase_ExchangeModel::ResourceType	176
Tableau 136 – Attributs de UseCase_ExchangeModel::Scenario	176
Tableau 137 – Extrémités d'association de UseCase_ExchangeModel:: Scenario avec d'autres classes	176
Tableau 138 – Attributs de UseCase_ExchangeModel::TriggeringEvent	177
Tableau 139 – Extrémités d'association de UseCase_ExchangeModel:: TriggeringEvent avec d'autres classes	177
Tableau 140 – Attributs de UseCase_ExchangeModel::UseCase	177
Tableau 141 – Extrémités d'association de UseCase_ExchangeModel:: UseCase avec d'autres classes	178
Tableau 142 – Attributs de UseCase_ExchangeModel::UseCaseLibrary	178
Tableau 143 – Extrémités d'association de UseCase_ExchangeModel:: UseCaseLibrary avec d'autres classes	179
Tableau 144 – Attributs de UseCase_ExchangeModel::UseCaseRepository	179

Tableau 145 – Extrémités d'association de UseCase_ExchangeModel::UseCaseRepository avec d'autres classes	179
Tableau 146 – Attributs de UseCase_ExchangeModel::Version	180
Tableau 147 – Extrémités d'association de UseCase_ExchangeModel::Version avec d'autres classes	180
Tableau 148 – Attributs de UseCase_ExchangeModel::FurtherActorInformation	180
Tableau 149 – Extrémités d'association de UseCase_ExchangeModel::FurtherActorInformation avec d'autres classes	180
Tableau 150 – XSD de bibliothèque de cas d'utilisation	181
Tableau 151 – Attributs de ActorLibrary_ExchangeModel::Actor	202
Tableau 152 – Extrémités d'association de ActorLibrary_ExchangeModel::Actor avec d'autres classes	202
Tableau 153 – Attributs de ActorLibrary_ExchangeModel::ActorLibrary	202
Tableau 154 – Extrémités d'association de ActorLibrary_ExchangeModel::ActorLibrary avec d'autres classes	202
Tableau 155 – Attributs de ActorLibrary_ExchangeModel::Ref_Actor	203
Tableau 156 – Extrémités d'association de ActorLibrary_ExchangeModel::Ref_Actor avec d'autres classes	203
Tableau 157 – Extrémités d'association de ActorLibrary_ExchangeModel::UseCaseRepository avec d'autres classes	203
Tableau 158 – XSD de bibliothèque d'acteurs	203
Tableau 159 – Attributs de RequirementLibrary_ExchangeModel::RequirementLibrary	207
Tableau 160 – Extrémités d'association de RequirementLibrary_ExchangeModel::RequirementLibrary avec d'autres classes	207
Tableau 161 – Attributs de RequirementLibrary_ExchangeModel::RequirementCategory	207
Tableau 162 – Extrémités d'association de RequirementLibrary_ExchangeModel::RequirementCategory avec d'autres classes	207
Tableau 163 – Attributs de RequirementLibrary_ExchangeModel::Ref_RequirementCategory	208
Tableau 164 – Extrémités d'association de RequirementLibrary_ExchangeModel::Ref_RequirementCategory avec d'autres classes	208
Tableau 165 – Attributs de RequirementLibrary_ExchangeModel::Requirement	208
Tableau 166 – Extrémités d'association de RequirementLibrary_ExchangeModel::Requirement avec d'autres classes	208
Tableau 167 – XSD de bibliothèque d'exigences	209

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

MÉTHODOLOGIE DES CAS D'UTILISATION –

**Partie 3: Définition des artefacts de formulaire type
de modèle de cas d'utilisation au format sérialisé XML**

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. A 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 62559-3 a été établie par le groupe "Gestion intelligente de l'énergie électrique", du comité d'études 8 de l'IEC: Aspects système de la fourniture d'énergie électrique.

La présente version bilingue (2020-09) correspond à la version anglaise monolingue publiée en 2017-12.

La version française de cette norme n'a pas été soumise au vote.

Ce document a été rédigé selon les Directives ISO/IEC, Partie 2.

Une liste de toutes les parties de la série IEC 62559, publiées sous le titre général *Méthodologie des cas d'utilisation*, peut être consultée sur le site web de l'IEC.

La présente norme IEC inclut les éléments de code, c'est-à-dire les éléments destinés à être traités directement par un ordinateur. Ce type de contenu est un texte placé entre les marqueurs <CODE BEGINS> et <CODE ENDS> ou est clairement étiqueté dans la présente norme en tant qu'élément de code.

L'achat de la présente norme IEC fait l'objet d'une licence de droits d'auteur permettant à l'acheteur de vendre le logiciel contenant des éléments de code issus de la présente norme à des utilisateurs, soit directement soit par l'intermédiaire de distributeurs, soumis aux conditions de licence logicielle IEC, qui peuvent être consultées à l'adresse ci-dessous: <http://www.iec.ch/webstore/custserv/pdf/CCv1.pdf>.

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. A cette date, la publication sera

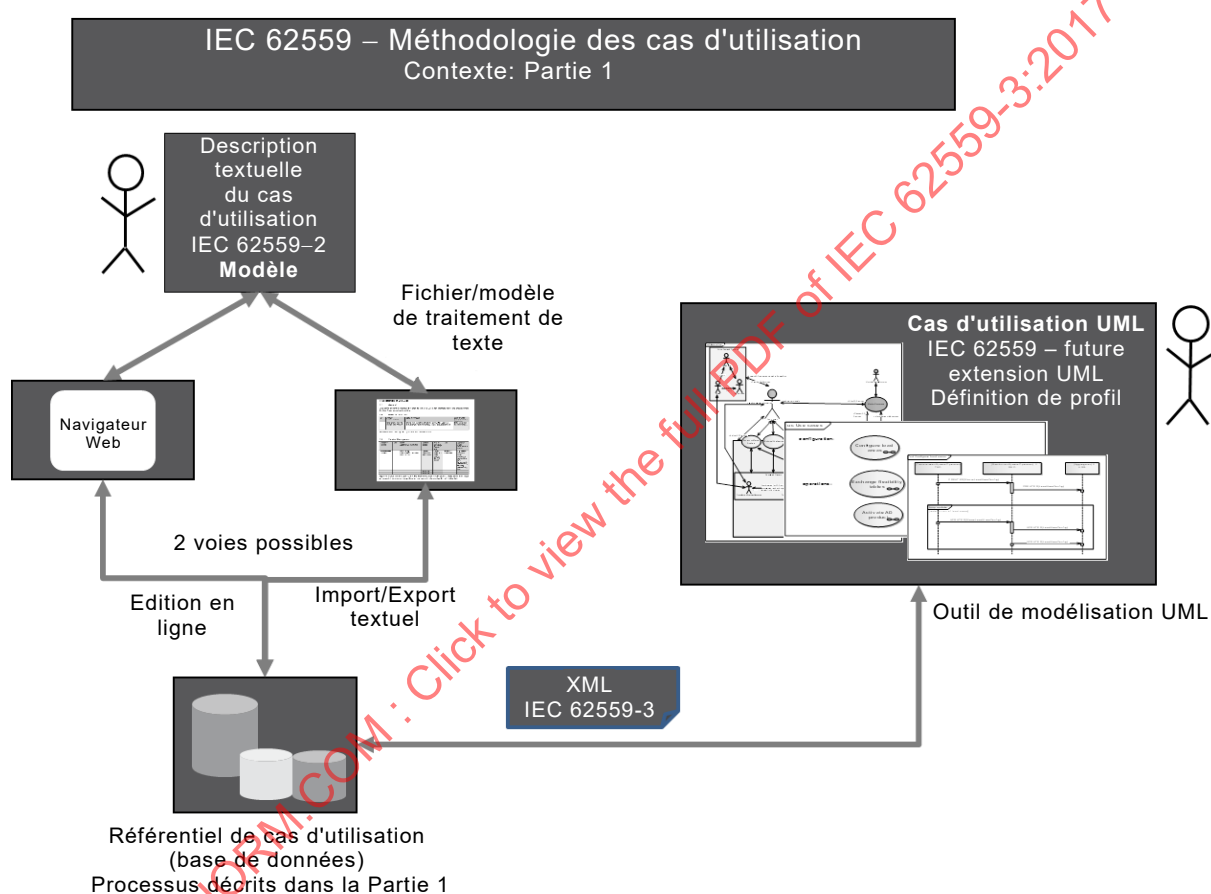
- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

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.

INTRODUCTION

Pour les systèmes complexes, la méthodologie des cas d'utilisation favorise une compréhension commune des fonctionnalités, acteurs et processus parmi différents comités d'études, voire différentes organisations. Développée comme outil logiciel, la méthodologie peut être utilisée pour soutenir le développement de normes, car elle facilite l'analyse des exigences par rapport aux nouvelles normes ou aux normes existantes. D'autres arguments en faveur de la méthodologie des cas d'utilisation et des informations générales figurent dans l'IEC 62559-1.

La Figure 1 fournit une vue d'ensemble des premières parties prévues de la série de normes IEC 62559, décrivant principalement la relation entre l'IEC 62559-2 et l'IEC 62559-3.



IEC

Figure 1 – Série de normes IEC 62559

IEC 62559-1 – Concept et processus de normalisation

L'IEC 62559-1 sert de base à un référentiel commun des cas d'utilisation pour regrouper les cas d'utilisation au sein de l'IEC sur une plateforme collaborative commune. Ce référentiel est aussi utilisé pour organiser une harmonisation des cas d'utilisation afin de fournir des cas d'utilisation génériques largement acceptés comme base pour le futur travail de normalisation. Cette partie décrit les processus et fournit les bases de la méthodologie des cas d'utilisation, comme les termes ou types de cas d'utilisation.

IEC 62559-2 – Définition de modèles de cas d'utilisation, de la liste d'acteurs et de la liste d'exigences

L'IEC 62559-2 définit la structure d'un formulaire type de modèle de cas d'utilisation, d'une liste d'acteurs et d'une liste d'exigences. Le document repose principalement sur l'IEC PAS 62559:2008 et doit être lu conjointement avec l'IEC 62559-1.

IEC 62559-3 – Définition des artefacts de formulaire type de modèle de cas d'utilisation au format sérialisé XML

Basé sur l'IEC 62559-2, ce document définit les concepts clés exigés et leur sérialisation en format XML d'un formulaire type de modèle de cas d'utilisation, d'une liste d'acteurs et d'une liste d'exigences détaillées. Le format XML est utilisé pour transférer le contenu du modèle aux autres systèmes techniques (par exemple, des outils de modélisation UML [Unified Modeling Language – langage de modélisation unifié]). Ces documents sont développés avec le système énergétique et les réseaux intelligents (Smart Grids) comme exemples, mais ils sont généralement suffisants pour être transférés vers d'autres domaines et systèmes. Le développement d'une définition de profil UML basée sur cette partie est prévu.

La série IEC 62559 est nécessaire pour répondre à la décision 7 du SG3 (Smart Grid Strategic Group – Groupe stratégique des réseaux intelligents) prise par le SMB (Standardization Management Board – Bureau de gestion de la normalisation) lors de la réunion de février 2010 (SMB/4204/DL, Décision 137/10) demandant la fourniture urgente d'un référentiel de cas d'utilisation génériques pour toutes les applications de réseaux intelligents. Néanmoins, la méthodologie des cas d'utilisation décrite dans le présent document est destinée à une plus large application dans la normalisation, allant au-delà des systèmes de réseaux intelligents.

L'apparition de systèmes de plus en plus complexes tels que les réseaux intelligents ou les villes intelligentes (Smart Cities) soulève la question de la gestion des exigences au niveau système, qui doivent être alimentées par de nombreux domaines d'expertise (dans la normalisation relative aux différents comités d'études) et qui doivent être davantage décomposées et partagées entre un grand nombre de comités d'études chargés de spécifier des normes pour soutenir ces fonctions au niveau système.

Une manière de traiter efficacement cette transversalité consiste à définir des méthodes et termes communs. La méthodologie des cas d'utilisation correspond à l'état de l'art actuel et vient à l'appui d'autres activités techniques.

La méthodologie des cas d'utilisation offre un moyen unique de partager des idées et exigences concernant de nouveaux cas d'utilisation ou cas d'affaire entre de nombreux experts/comités d'études dans différents domaines (par exemple, experts de domaine avec des connaissances des systèmes énergétiques ou processus métier d'une part et experts système/en informatique définissant les informations échangées et la communication d'autre part). Dans le processus de développement des exigences, les experts de domaine fournissent les idées générales et les exigences fonctionnelles. Pour les experts système, le principal objectif consiste à détailler ces cas d'utilisation à un niveau auquel ils peuvent être utilisés pour spécifier les interfaces, la fonctionnalité dédiée, ainsi que l'échange de modèles de données et de services. Toutefois, les experts en sécurité ou CEM (par exemple) peuvent également utiliser les cas d'utilisation décrits, leur terminologie et les exigences identifiées.

Le point de départ consiste cependant à définir un cadre pour assurer une certaine uniformité au sein de l'IEC dans le but d'aider les membres de l'IEC à fournir des cas d'utilisation de manière cohérente. L'IEC 62559 sert de base pour les référentiels de cas d'utilisation afin de collecter, d'administrer, de maintenir et d'évaluer les cas d'utilisation.

Au sein de l'IEC, un référentiel des cas d'utilisation fait office de plateforme collaborative commune pour l'élaboration des cas d'utilisation et pour organiser une harmonisation des cas d'utilisation afin de fournir des cas d'utilisation génériques largement acceptés comme base pour le travail de normalisation futur.

Par ailleurs, le formulaire type de modèle de cas d'utilisation défini dans le présent document peut servir non seulement au développement de normes, mais également – but original de l'IEC PAS 62559:2008 – comme moyen utile pour la réalisation de projets dans le domaine des systèmes complexes. De même, les autres applications qui ont besoin des avantages liés à un développement d'exigences structuré et à une description de fonctionnalités formalisée peuvent utiliser le modèle suggéré.

La méthodologie des cas d'utilisation est considérée comme un processus qui commence par la définition d'idées, d'objectifs et d'exigences métier, détaillés dans les descriptions des cas d'utilisation. Ces informations peuvent être utilisées comme base pour identifier/associer les architectures de référence décrivant les types de composants utilisés et approfondissant l'analyse du processus de normalisation.

D'autres développements concernant le formulaire type de modèle de cas d'utilisation sont prévus. Ces développements portent principalement sur les informations qui sont exigées pour la description des cas d'utilisation pour une analyse ultérieure, et qui peuvent être associées à d'autres informations (par exemple, à une architecture de référence, aux méthodes de sécurité informatique, aux normes et aux modèles de données). Cela est en partie pris en compte dans le modèle suggéré de l'IEC 62559-2. D'autres relations sont décrites séparément, car elles sont en cours de développement et peuvent être prises en considération pour le développement futur du référentiel des cas d'utilisation de l'IEC.

IECNORM.COM : Click to view the full PDF of IEC 62559-3:2017

MÉTHODOLOGIE DES CAS D'UTILISATION –

Partie 3: Définition des artefacts de formulaire type de modèle de cas d'utilisation au format sérialisé XML

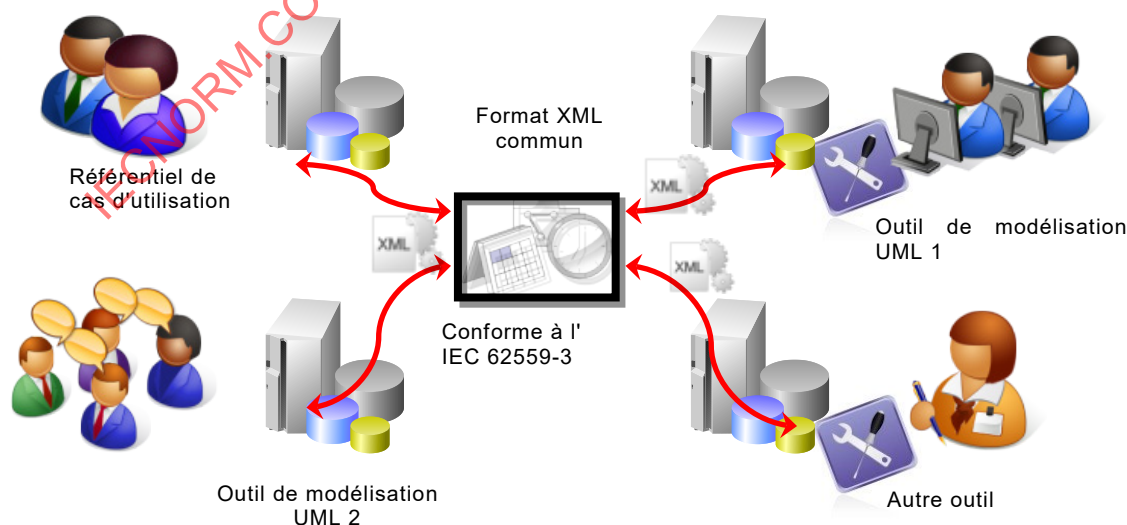
1 Domaine d'application

Pour échanger des cas d'utilisation s'appuyant sur le modèle défini dans l'IEC 62559-2, la présente partie de l'IEC 62559 établit les interfaces entre les différents référentiels de cas d'utilisation et/ou les outils logiciels UML.

Par conséquent, le présent document définit les concepts clés exigés et leur sérialisation en format syntaxique XML d'un formulaire type de modèle de cas d'utilisation, d'une liste d'acteurs et d'une liste d'exigences détaillées.

Comme le montre la Figure 2, la modélisation s'appuie sur l'utilisation du langage UML afin de représenter graphiquement les données contenues dans un cas d'utilisation reposant sur le modèle de l'IEC 62559. Par conséquent, le format textuel du formulaire type de modèle de cas d'utilisation peut constituer, dans le processus de développement du cas d'utilisation, un simple point de départ pour les experts métier ou un moyen aisé de modifier les données de cas d'utilisation pour les experts non UML. Il est donc important que la série IEC 62559 propose un moyen fiable de conversion de ce format textuel au format UML, et inversement. Dès qu'un référentiel de cas d'utilisation est maintenu conformément à la série IEC 62559, pouvoir importer/exporter les informations relatives au cas d'utilisation entre les différents outils UML et les différents référentiels de cas d'utilisation en utilisant un format indépendant constitue une autre nécessité associée.

Le présent document a pour principal objet de proposer un format indépendant de transfert des informations de cas d'utilisation entre des logiciels de modélisation. Pour atteindre cet objectif, le format syntaxique XML est choisi afin de sérialiser les données de cas d'utilisation. Le présent document définit en détail les concepts clés du modèle en langage UML et leurs transformations en langage XML à l'aide de la norme XSD.



IEC

**Figure 2 – Format XML commun d'import/export d'informations de cas d'utilisation
entre différents logiciels de modélisation et référentiels**

Le présent document assure l'interopérabilité entre les logiciels de modélisation utilisés par différentes sociétés dans le monde (voir la Figure 2).

Lorsque ce niveau d'interopérabilité est atteint, l'IEC 62559 peut fournir un mécanisme fiable d'interprétation de ces données XML afin de représenter graphiquement les cas d'utilisation UML. Ce besoin est également couvert dans une partie à venir de l'IEC 62559 (à définir).

Le présent document met l'accent sur un cadre méthodologique également utilisé par les normes du comité d'études 57 de l'IEC et qui est récapitulé à l'Article 4.

Pour échanger des cas d'utilisation s'appuyant sur le modèle défini dans l'IEC 62559-2, le présent document établit les interfaces entre les différents référentiels de cas d'utilisation et/ou outils UML.

2 Références normatives

Les documents suivants sont cités dans le texte de sorte qu'ils constituent, pour tout ou partie de leur contenu, des exigences du présent document. 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 62559-2, *Méthodologie des cas d'utilisation – Partie 2: Définition du formulaire type de modèle de cas d'utilisation, de la liste d'acteurs et de la liste d'exigences*

3 Termes et définitions

Pour les besoins du présent document, les termes et les définitions de l'IEC 62559-2, ainsi que les suivants, s'appliquent.

L'ISO et l'IEC tiennent à jour des bases de données terminologiques destinées à être utilisées en normalisation, consultables aux adresses suivantes:

- IEC Electropedia: disponible à l'adresse <http://www.electropedia.org/>
- ISO Online browsing platform: disponible à l'adresse <http://www.iso.org/obp>

3.1

XML

eXtensible Markup Language

langage de marquage qui définit un ensemble de règles pour le codage des documents dans un format lisible à la fois par l'homme et la machine

Note 1 à l'article: La spécification XML est produite par le World Wide Web Consortium (W3C).

3.2

XSD

langage XML Schema Definition

langage utilisé pour définir la structure, le contenu et la sémantique des fichiers XML

Note 1 à l'article: XSD est une recommandation du World Wide Web Consortium (W3C).

4 Cadre méthodologique pour le développement de la présente Norme

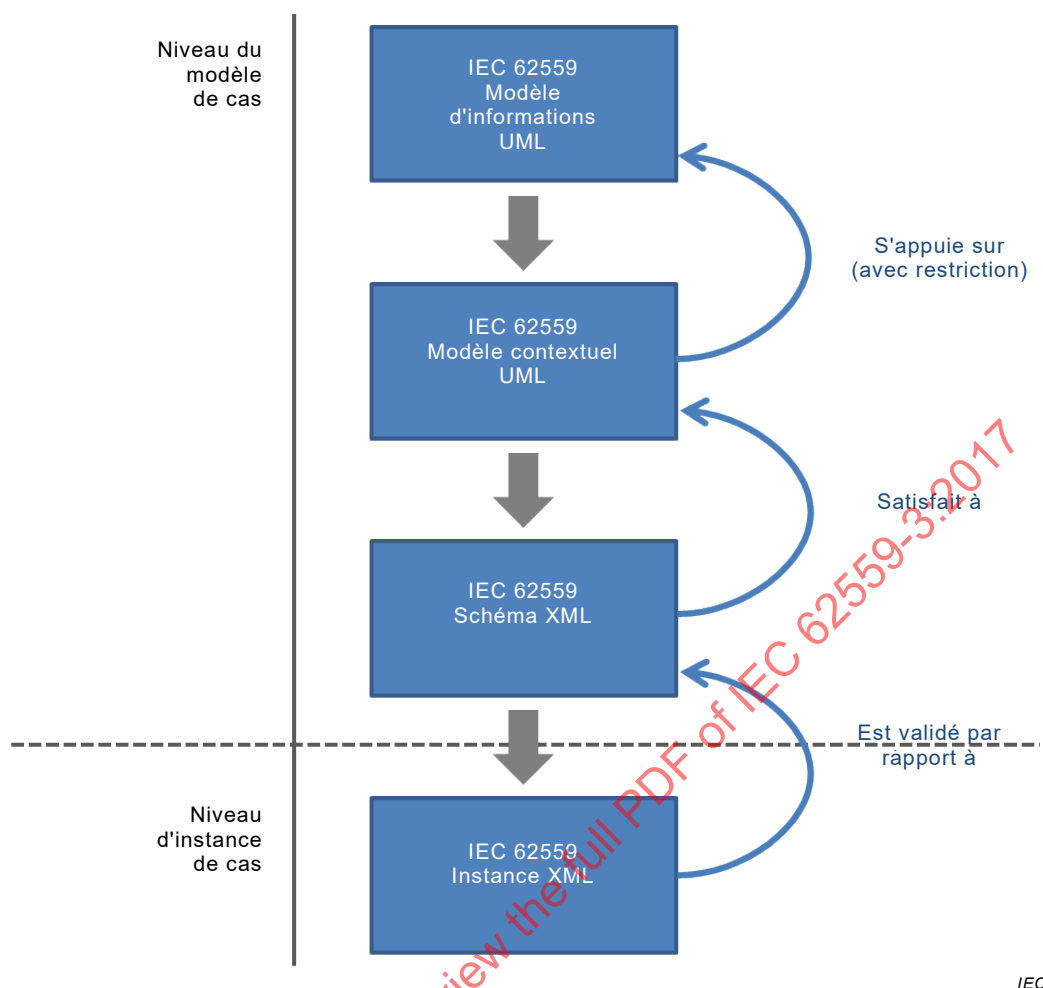
4.1 Approche basée sur des modèles pour le développement du format d'échange de données

Le présent document s'appuie sur une approche basée sur des modèles pour la définition des formats d'échange et, par conséquent, définit les modèles UML respectifs. L'approche utilisée ici a été établie par le comité d'études 57 de l'IEC afin de développer des modèles de données (profils) spécifiques. Elle est activement utilisée pour développer des normes CIM (Common Information Model – Modèle d'informations commun) du comité d'études 57 de l'IEC pour les communications des réseaux d'électricité. Cette approche décrit plusieurs types de modèles, leurs relations et les règles de déduction de modèles plus spécifiques à partir d'un modèle générique. L'approche étant expliquée en détail dans l'IEC 62325-450, l'Article 4.1 ne décrit l'approche et ses adaptations que brièvement pour créer le cadre dans lequel s'inscrit le présent document.

L'approche permet de définir plusieurs formats d'échange XML pour un sujet particulier. Toutefois, tous les formats d'échange créés selon ce procédé reposent sur un seul modèle source assurant la compatibilité sémantique. Deux types de modèles reliés sont utilisés pour différents objectifs, jouant un rôle essentiel dans ce procédé. Il s'agit respectivement du modèle d'information et du modèle contextuel. La Figure 3 présente ces modèles dans le contexte du processus de définition du format d'échange de données.

Le modèle d'information représente tous les concepts clés communs d'un cas d'utilisation identifiés dans l'IEC 62559-2 à un niveau plutôt générique, y compris les entités, leurs attributs et leurs relations. Cela correspond, par exemple, au cas d'utilisation lui-même, aux scénarii d'un cas d'utilisation, aux acteurs concernés ou aux objets d'information échangés dans l'ensemble d'un cas d'utilisation. Les concepts définis dans le modèle d'information définissent la sémantique de l'échange de données. Par conséquent, le modèle d'information constitue une "source unique de vérité" concernant la structure des informations de cas d'utilisation et peut être considéré comme strictement normatif.

Un modèle contextuel s'appuie toujours sur le modèle d'information. Les modèles contextuels sont créés pour répondre aux besoins spécifiques en matière d'échange d'informations concernant le modèle d'information. Les modèles contextuels incluent un sous-ensemble (voire même l'ensemble) du modèle d'information dépendant du contexte et pouvant être limité ou spécialisé dans une plus large mesure.



IEC

Figure 3 – Cadre de développement basé sur des modèles pour la définition du format d'échange de données XML de la norme IEC 62559-3

Enfin, d'autres règles peuvent être appliquées à un modèle contextuel en créant le format d'échange final qui satisfait ensuite, par nature, au modèle contextuel. Dans le contexte du présent document, il s'agit d'un schéma XML qui sert de point de départ pour la validation des données de cas d'utilisation échangées. A cet effet, les données d'instance de cas d'utilisation doivent être définies dans un document XML.

Le processus basé sur des modèles présenté ci-dessus a été appliqué afin de développer les formats d'échange de données XML correspondants. Pour enfin générer un fichier normalisé de définition de schéma XML (XSD) conformément à l'IEC 62559-2, l'approche basée sur des modèles est réalisée en détail comme suit:

a) Modèle d'information UML IEC 62559:

Ce langage UML rassemble les concepts clés décrits dans l'IEC 62559-2. Il est modélisé par des diagrammes de classe UML comprenant des classes, des attributs, des types de données et des associations. Ce modèle d'information contient les concepts clés d'un cas d'utilisation modélisé de manière générique pour s'adapter à d'autres utilisations de ces concepts dans d'autres mises en œuvre. Par exemple, il pourrait également être utilisé pour définir la structure d'une base de données pour des cas d'utilisation. Par conséquent, ce modèle d'information est assoupli en modélisant les concepts avec des aspects facultatifs. Par exemple, de nombreux attributs et de nombreuses associations sont modélisés avec des cardinalités flexibles (par exemple, 0..1 pour les attributs et 0..* pour les relations). Pour permettre une mise en œuvre de ce modèle d'information générique, il est nécessaire de définir plus précisément la manière dont les concepts clés sont utilisés pour les besoins particuliers de l'échange d'informations de cas d'utilisation.

b) Modèle contextuel UML IEC 62559:

Ce deuxième modèle UML s'appuie sur le modèle d'information UML en modélisant les restrictions afin de l'adapter aux besoins particuliers de définition d'un modèle spécifiant l'échange d'informations de cas d'utilisation. Ces restrictions définies sur les concepts du modèle d'information précisent ce modèle contextuel pour l'échange d'informations de cas d'utilisation. Les restrictions admises sont les suivantes:

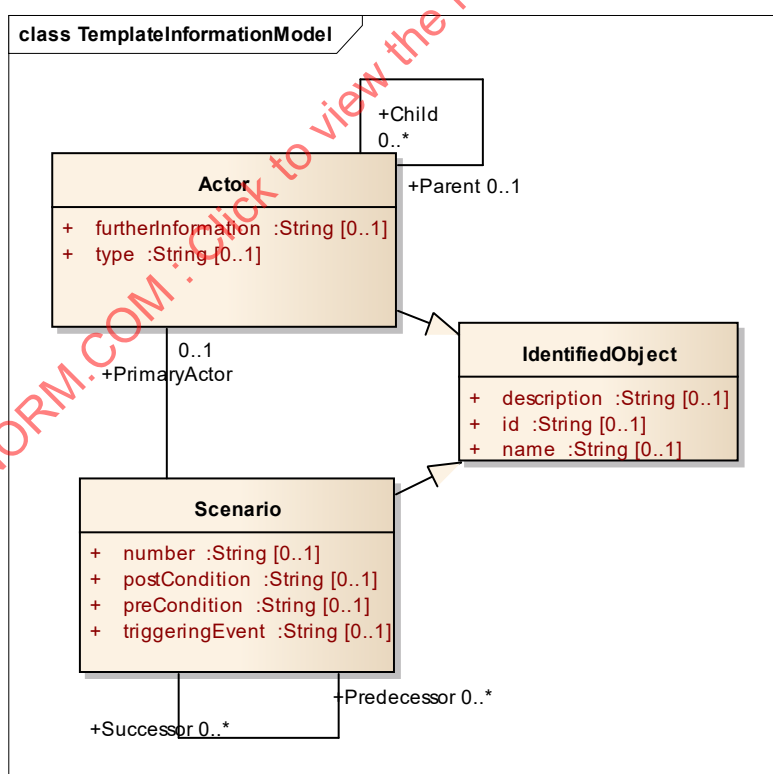
- 1) Restrictions sur les cardinalités des attributs et des associations en les réduisant. Par exemple, la cardinalité 0..1 d'un attribut peut être définie sous la forme 1..1 (attribut obligatoire), 0..1 (attribut facultatif) ou 0..0 (attribut non utilisé). Une cardinalité de fin d'association 0..* peut être définie à intervalle inclus comme 2..6 (séquence de concepts instanciés à 2 au minimum et 6 au maximum).
- 2) Restrictions sur la valeur d'attribut en ajoutant des facettes. Par exemple, un attribut typé par une chaîne peut avoir une facette de longueur 6, ce qui signifie que la valeur d'attribut prévue est une séquence de 6 symboles.

c) Schéma XML IEC 62559:

Cette définition du schéma XML (XSD) est générée automatiquement à partir du modèle contextuel UML précédemment développé. Chaque échange d'informations de cas d'utilisation XML doit être validé en fonction de ce XSD. Elle permet donc d'assurer l'interopérabilité entre les logiciels de modélisation, l'échange d'informations de cas d'utilisation devant respecter la structure définie dans le XSD.

4.2 Exemple de processus basé sur des modèles dans le contexte de l'IEC 62559

Le paragraphe 4.2 donne un exemple s'appuyant sur la relation entre un scénario de cas d'utilisation et un acteur jouant le rôle PrimaryActor dans ce même scénario.

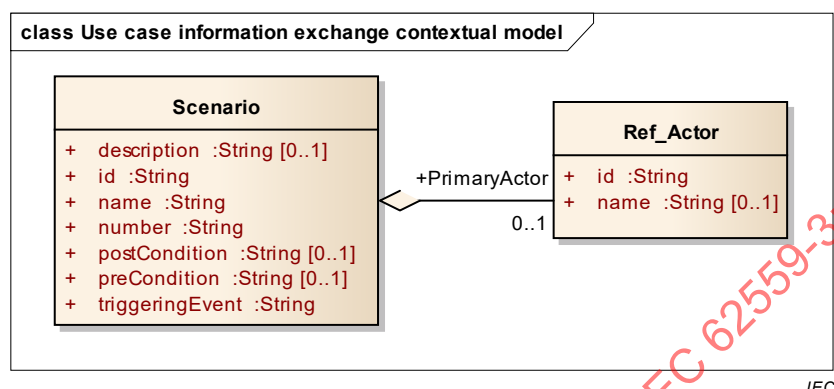


IEC

Anglais	Français
Class TemplateInformationModel	Classe TemplateInformationModel

Figure 4 – Modèles d'information UML pour le scénario, exemple de relation Actor

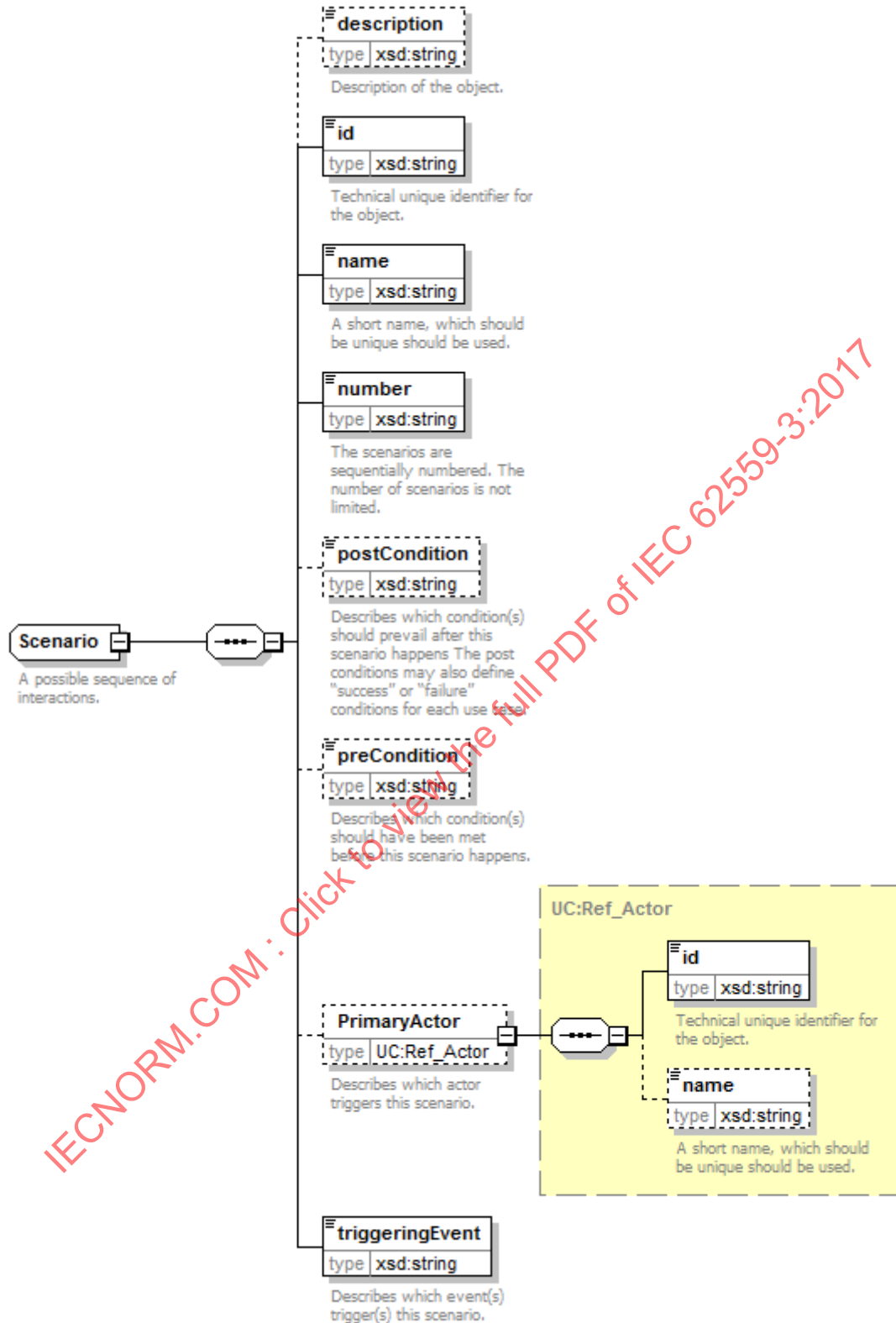
La Figure 4 présente le modèle d'information UML représentant les deux concepts clés Scenario et Actor (à l'aide de classes UML) liés à une association UML jouant le rôle PrimaryActor et la cardinalité 0..1 côté Actor. Cela signifie que l'acteur principal d'un scénario peut être facultatif. Les informations supplémentaires pour le scénario et l'acteur sont modélisées avec des attributs de ces deux classes. Par exemple, un scénario est composé d'un numéro, d'une postCondition, d'une preCondition et d'un triggeringEvent. Ces deux concepts ayant le besoin commun de représenter un identificateur, une classe de généralisation IdentifiedObject partage ces attributs communs tels que les attributs id, name et description.



Anglais	Français
class Use case information exchange contextual model	Classe Use case information exchange contextual model

Figure 5 – Modèle contextuel UML basé sur l'exemple de relation Scenario-Actor

Dans le modèle contextuel UML (voir la Figure 5), les deux concepts Scenario et Actor sont réutilisés et personnalisés en ajoutant des restrictions sur le modèle d'information pour répondre au besoin particulier d'échange d'informations concernant un scénario comportant un PrimaryActor. Dans cet exemple, certains attributs ont été rendus obligatoires. Ainsi, pour le scénario, l'identificateur (ID), le nom, le numéro et l'évènement déclencheur (triggeringEvent) sont des informations obligatoires pour l'échange. De plus, si les informations d'un PrimaryActor sont données, son identifiant est également obligatoire. Le concept Actor est réutilisé plusieurs fois avec différentes structures définies dans le même modèle contextuel. Pour obtenir une définition de nom contextuelle unique pour ces différentes structures, un qualifiant est ajouté au nom de classe du modèle d'information. Dans cet exemple, la classe contextuelle reposant sur le concept Actor est appelée Ref_Actor. Lors de la construction du modèle contextuel, l'association entre Scenario et Actor est restreinte et devient une agrégation, ce qui signifie que dans l'échange de données, les informations du scénario contiennent des informations relatives à un PrimaryActor.



Anglais	Français
Type	Type
Description of the object.	Description de l'objet.
Technical unique identifier for the object.	Identificateur technique unique de l'objet.
A short name, which should be unique should be used.	Il convient d'utiliser un nom court, et il convient que ce nom soit unique.
The scenarios are sequentially numbered. The number of scenarios is not limited.	Les scénarii sont numérotés de manière séquentielle. Le nombre de scénarii n'est pas limité.
Describes which condition(s) should prevail after this scenario happens. The post conditions may also define "success" or "failure" conditions for each use case.	Décrit quelle(s) condition(s) il convient de faire prévaloir après que ce scénario s'est produit. Les postconditions peuvent également définir les conditions de "réussite" ou de "défaillance" pour chaque cas d'utilisation.
Describes which condition(s) should have been met before this scenario happens.	Décrit à quelle(s) condition(s) il convient de satisfaire avant que ce scénario ne se produise.
A possible sequence of interactions.	Séquence possible d'interactions.
Describes which actor triggers this scenario	Décrit l'acteur qui déclenche ce scénario
Describes which even(s) trigger(s) this scenario	Décrit l'évènement ou les évènements qui déclenche(nt) ce scénario

Figure 6 – Schéma XML correspondant à l'exemple de relation Scenario-Actor

Figure 6 présente de manière graphique le code XSD qui est automatiquement généré à partir du modèle contextuel UML de la Figure 5.

4.3 Développement de profils particuliers d'échange de données

L'application du processus basé sur des modèles décrit ci-dessus peut générer plusieurs modèles contextuels et, par conséquent, plusieurs formats d'échange différents, présentant toutefois souvent la même sémantique. Ce processus de déduction d'un modèle spécifique, en ce qui concerne un sous-ensemble d'éléments de modèle, à partir d'un modèle générique et de spécialisation des éléments de modèle, est appelé "profilage", dont le résultat est appelé "profil". Plusieurs profils pouvant être créés à partir d'un seul modèle d'information, tous les profils ne peuvent pas être considérés comme étant normatifs. Toutefois, si les données entre deux applications doivent être échangées avec succès, un format d'échange particulier doit faire l'objet d'un accord (normatif), tant du point de vue sémantique que syntaxique.

Le présent document identifie donc les principaux scénarii en matière d'échange d'informations, donnant lieu aux profils correspondants qui sont normatifs et qui peuvent être mis en œuvre dans un logiciel. Les principaux scénarii d'échange d'informations actuellement identifiés incluent l'échange de tous les concepts du modèle d'information (par exemple, pour l'échange d'informations entre des référentiels de cas d'utilisation), l'échange d'un ensemble d'acteurs (modèle de rôles) et l'échange d'exigences (voir l'Article 5). Le premier scénario d'échange de données étant plutôt générique, les autres permettent une validation plus rigoureuse des données par rapport au modèle d'échange couvrant un référentiel complet, étant donné que certaines restrictions peuvent être prises pour hypothèse, comme la présence d'un acteur ou d'une exigence.

Le format normatif d'échange de données du premier scénario d'échange générique peut être consulté à l'Article 6. Les éditions ultérieures de l'IEC 62559-3 peuvent ajouter d'autres scénarii normalisés d'échange d'informations et leurs profils respectifs, comme pour les cas d'utilisation. Toutefois, en cas de besoins particuliers, des formats/profils d'échange non normatifs peuvent être développés à l'aide du processus basé sur des modèles.

5 Application du cadre méthodologique de définition d'un ensemble de formats d'échange syntaxiques clés (cas d'utilisation, acteurs, exigences)

Par rapport aux différents types d'échanges de données dépendant des scénarii d'échange de données clés spécifiques, l'IEC 62559-3 vise à inclure un ensemble de formats clés d'échange de données pour des cas d'utilisation, des acteurs et des exigences. Pour générer ces formats d'échange syntaxiques XML, le processus de modélisation décrit à l'Article 4 est appliqué pour concevoir chaque format d'échange. La Figure 7 donne un aperçu du profilage du modèle d'information générique.

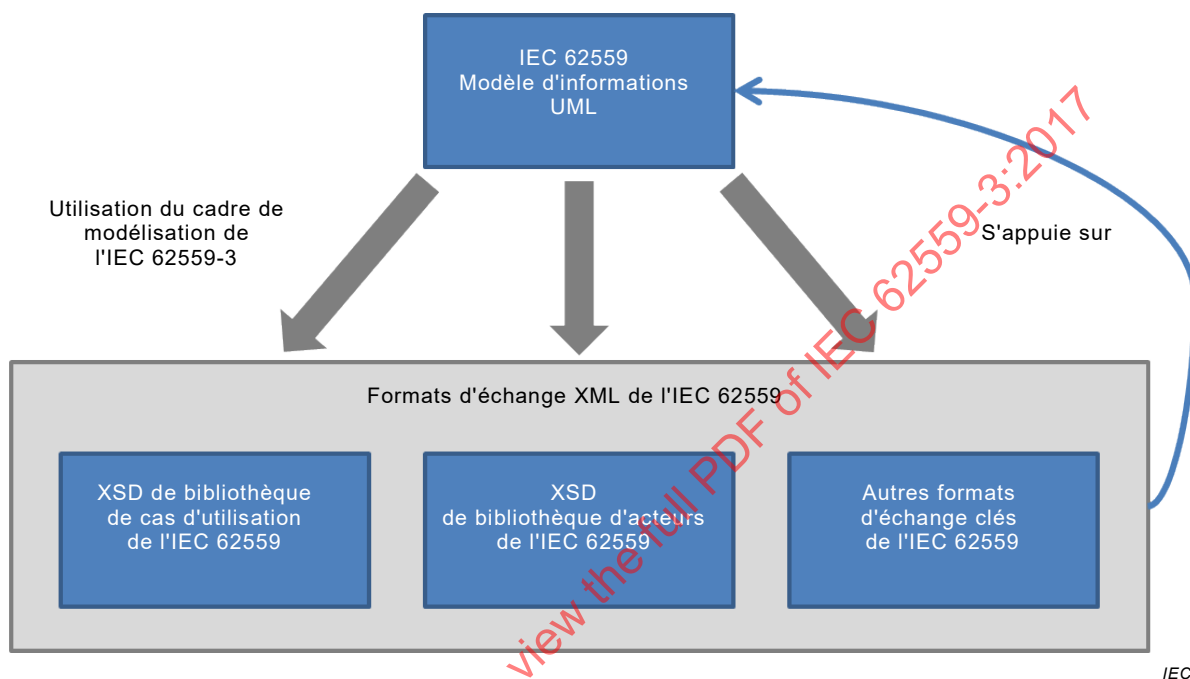


Figure 7 – Production de formats d'échange XML clés basés sur un modèle d'information UML IEC 62559 unique

6 Contenu détaillé des modèles (génération automatique à partir de modèles UML)

6.1 Concepts clés de cas d'utilisation UML (modèle d'information UML)

6.1.1 Généralités

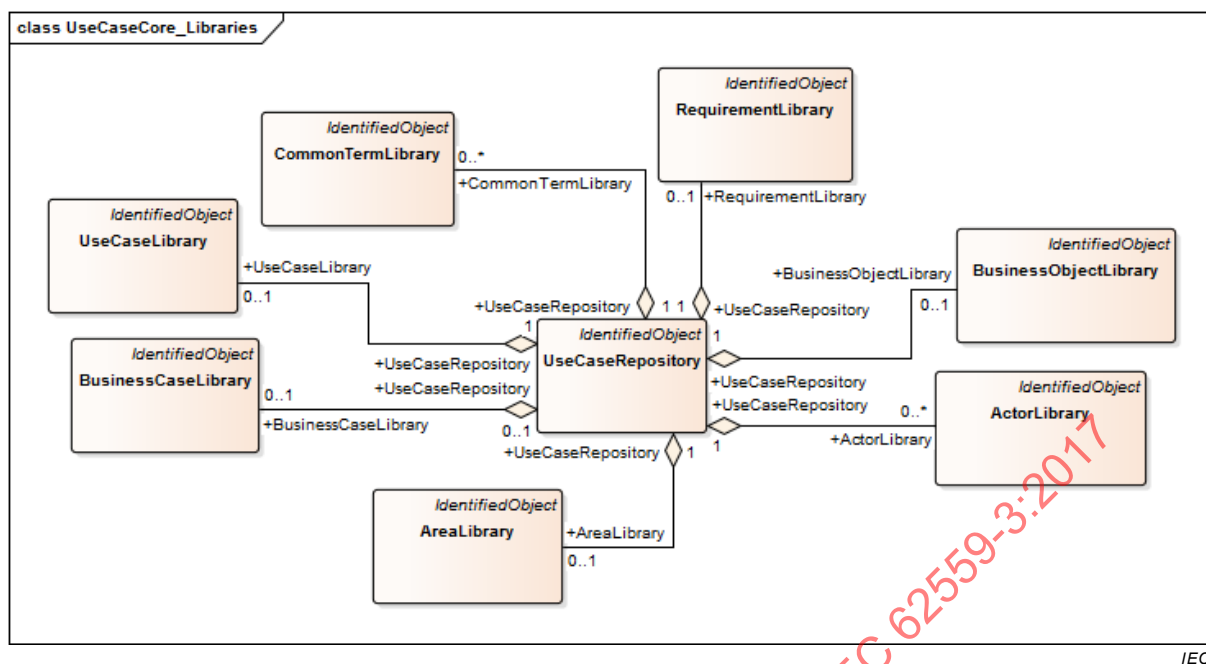
Les concepts clés de cas d'utilisation sont décrits dans un modèle d'information UML partagé entre les différents formats d'échange du présent document. Les concepts clés sont classés en ensemble de paquetages UML.

6.1.2 Paquetage de modèle UseCaseCoreConcepts

6.1.2.1 Généralités

Modèle d'information UML IEC 62559 pour les concepts clés de cas d'utilisation.

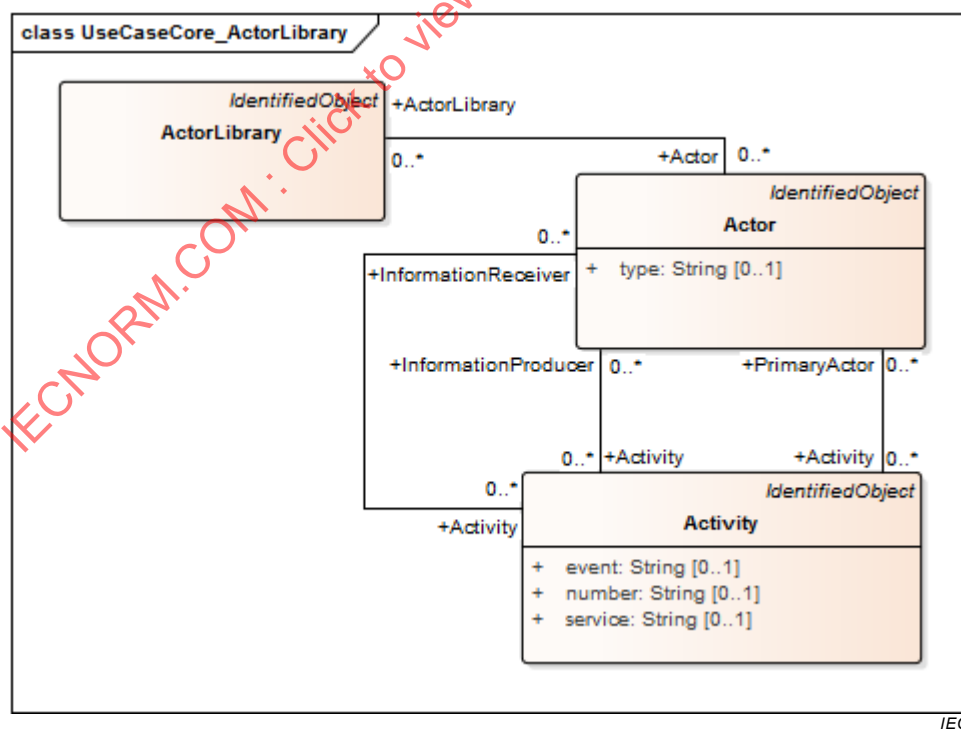
La Figure 8 présente le diagramme de classe UseCaseCore_Libraries.



Anglais	Français
class UseCaseCore_Libraries	classe UseCaseCore_Libraries

Figure 8 – Diagramme de classe UseCaseCoreConcepts::UseCaseCore_Libraries

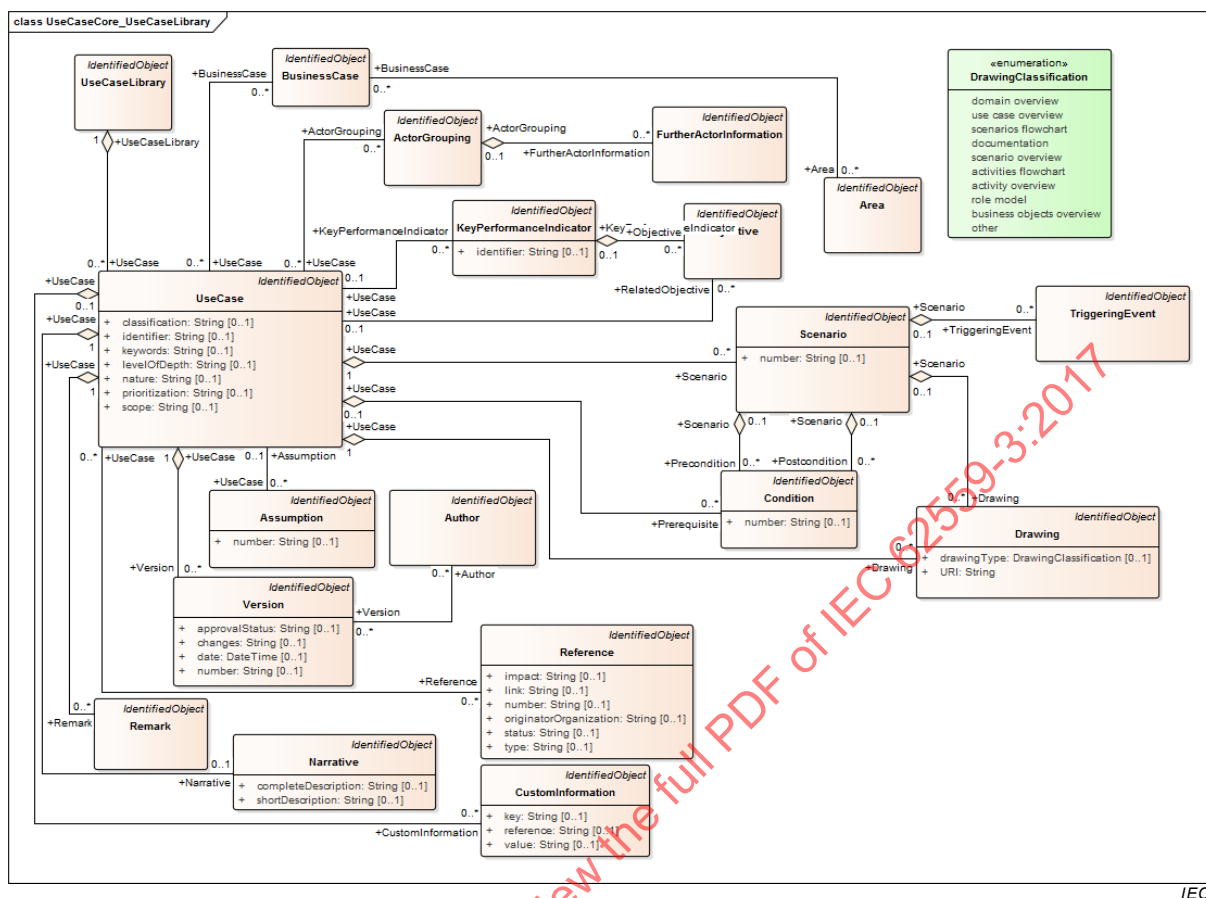
La Figure 9 présente le diagramme de classe UseCaseCore_ActorLibrary.



Anglais	Français
class UseCaseCore_ActorLibrary	classe UseCaseCore_ActorLibrary

Figure 9 – Diagramme de classe UseCaseCoreConcepts::UseCaseCore_ActorLibrary

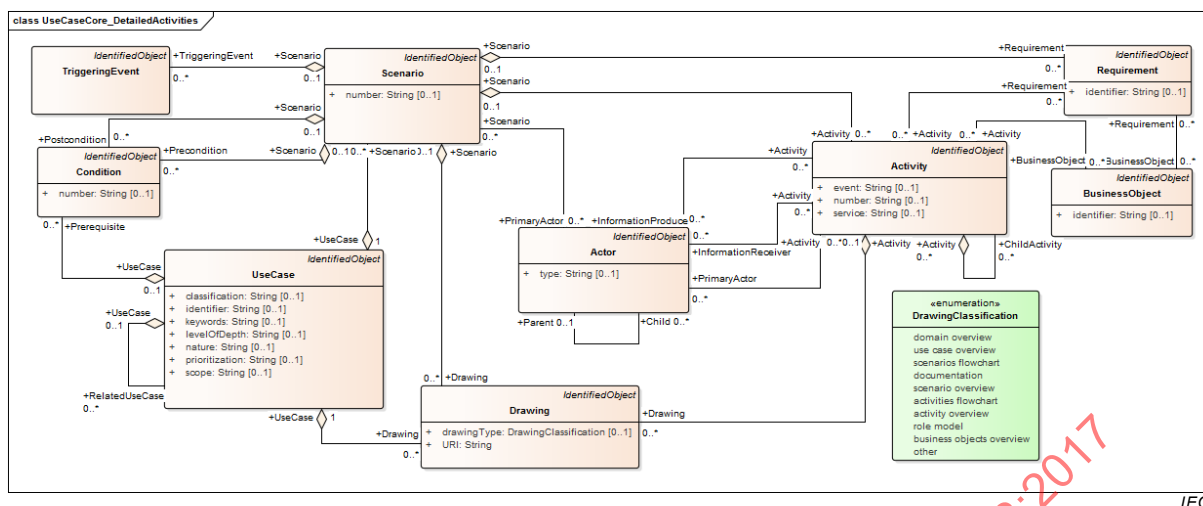
La Figure 10 présente le diagramme de classe UseCaseCore_UseCaseLibrary.



Anglais	Français
class UseCaseCore_UseCaseLibrary	classe UseCaseCore_UseCaseLibrary

**Figure 10 – Diagramme de classe
UseCaseCoreConcepts::UseCaseCore_UseCaseLibrary**

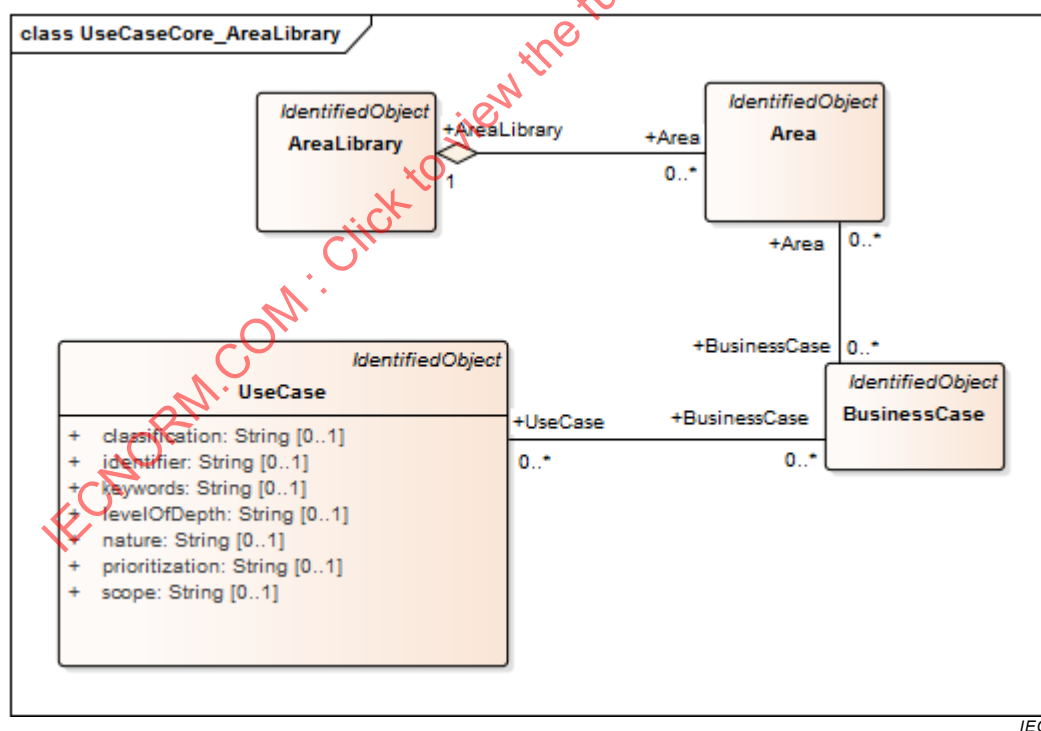
La Figure 11 présente le diagramme de classe UseCaseCore_DetailedActivities.



Anglais	Français
class UseCaseCore_Detailed Activities	classe UseCaseCore_Detailed Activities

**Figure 11 – Diagramme de classe
UseCaseCoreConcepts::UseCaseCore_DetailedActivities**

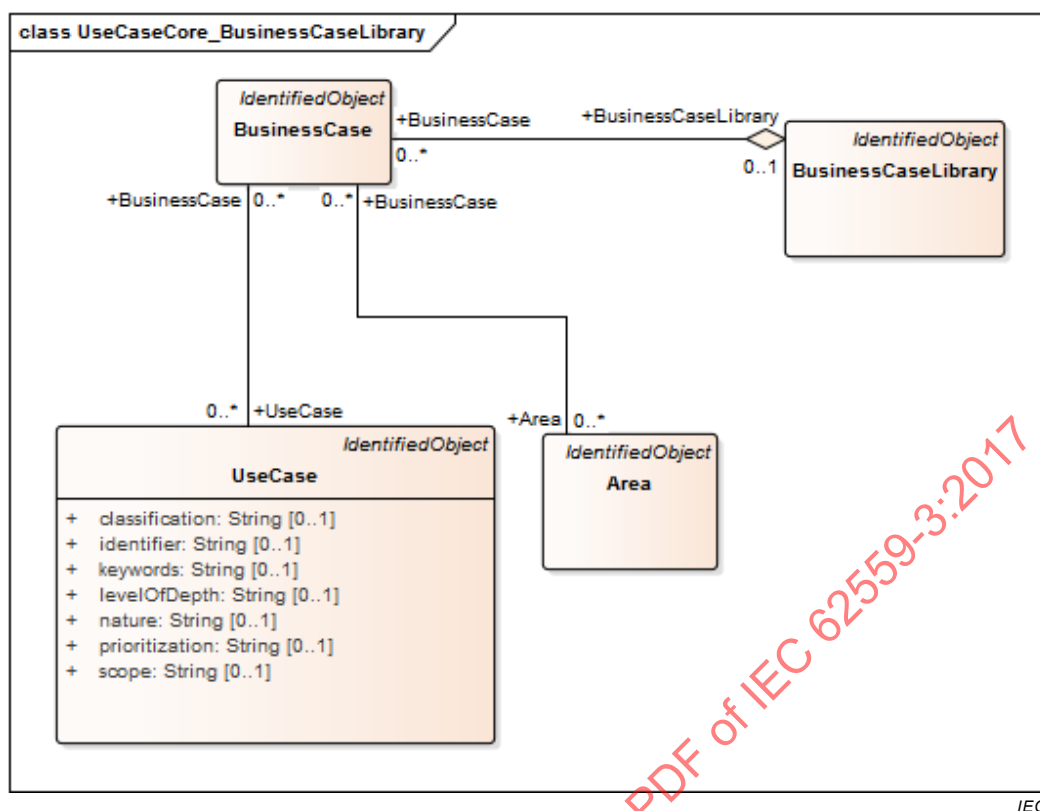
La Figure 12 présente le diagramme de classe UseCaseCore AreaLibrary.



Anglais	Français
Classe UseCaseCore_AreaLibrary	Classe UseCaseCore_AreaLibrary

Figure 12 – Diagramme de classe UseCaseCoreConcepts::UseCaseCore_AreaLibrary

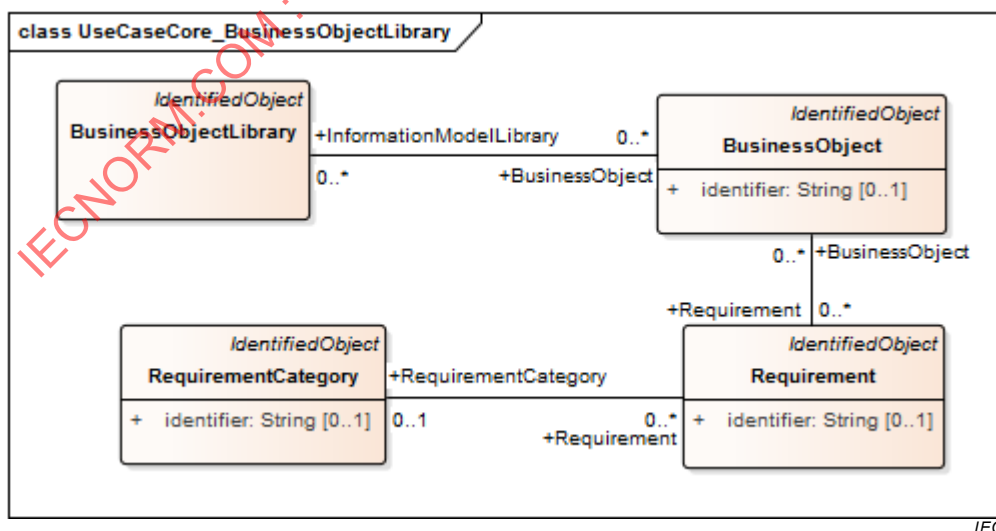
La Figure 13 présente le diagramme de classe UseCaseCore_BusinessCaseLibrary.



Anglais	Français
class UseCaseCore_BusinessCaseLibrary	classe UseCaseCore_BusinessCaseLibrary

Figure 13 – Diagramme de classe
UseCaseCoreConcepts::UseCaseCore_BusinessCaseLibrary

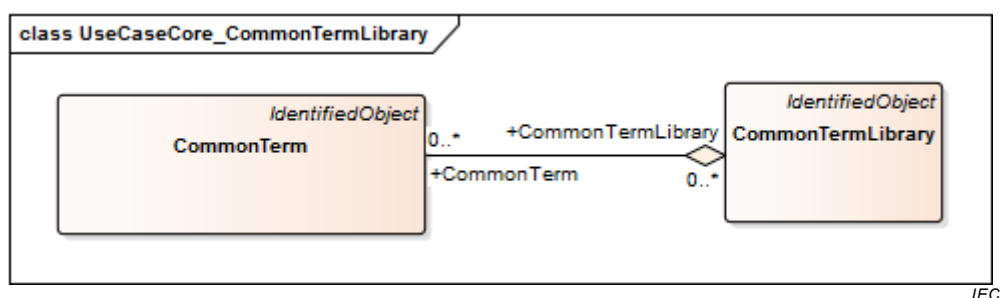
La Figure 14 présente le diagramme de classe UseCaseCore_BusinessObjectLibrary.



Anglais	Français
class UseCaseCore_BusinessObjectLibrary	classe UseCaseCore_BusinessObjectLibrary

Figure 14 – Diagramme de classe
UseCaseCoreConcepts::UseCaseCore_BusinessObjectLibrary

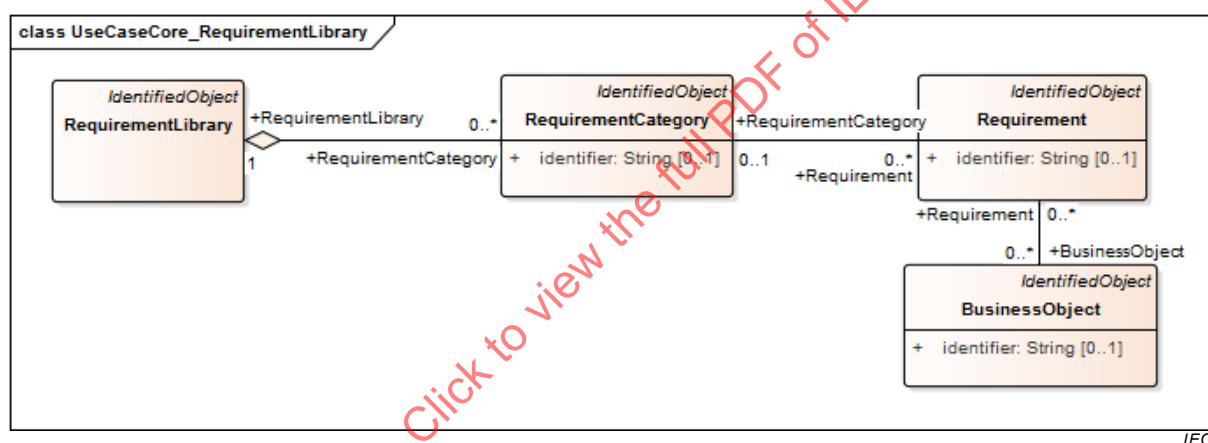
La Figure 15 présente le diagramme de classe UseCaseCore_CommonTermLibrary.



Anglais	Français
class UseCaseCore_CommonTermLibrary	classe UseCaseCore_CommonTermLibrary

**Figure 15 – Diagramme de classe
UseCaseCoreConcepts::UseCaseCore_CommonTermLibrary**

La Figure 16 présente le diagramme de classe UseCaseCore_RequirementLibrary.



Anglais	Français
class UseCaseCore_RequirementLibrary	classe UseCaseCore_RequirementLibrary

**Figure 16 – Diagramme de classe
UseCaseCoreConcepts::UseCaseCore_RequirementLibrary**

6.1.2.2 Activity

Elle peut être divisée en d'autres activités ou représenter une étape élémentaire du scénario.

Le Tableau 1 présente tous les attributs d'Activity.

Tableau 1 – Attributs de UseCaseCoreConcepts::Activity

name	type	description
event	String	Il s'agit de l'évènement qui déclenche l'étape. Il pourrait s'agir de l'achèvement d'étapes précédentes (auquel cas l'évènement est la liste séparée par une virgule de leurs nombres) ou de la condition de sortie d'une boucle.
number	String	indice de l'activité
service	String	Cette colonne identifie la nature du flux d'informations et l'auteur des informations. Les options disponibles sont CREATE, GET, CHANGE, DELETE, CANCEL, EXECUTE déduites de l'IEC 61968-100:2013. De plus, REPORT, TIMER et REPEAT sont suggérés. – CREATE signifie qu'un objet d'information doit être créé au niveau du producteur. – GET (il s'agit de la valeur par défaut si rien n'est renseigné) signifie que le récepteur demande des informations au producteur (par défaut). – CHANGE signifie que les informations doivent être mises à jour. Le producteur met à jour les informations du récepteur. – DELETE signifie que les informations doivent être supprimées. Le producteur supprime les informations du récepteur. – CANCEL et CLOSE impliquent des actions relatives aux processus, telles que la fermeture d'un ordre d'exécution ou l'annulation d'une demande de contrôle. – EXECUTE est utilisé lorsqu'une transaction complexe est en cours de transmission au moyen d'un service, qui contient éventuellement plusieurs verbes. – REPORT est utilisé pour représenter le transfert d'informations non sollicitées ou de flux d'informations asynchrones. Le producteur donne des informations au récepteur. – TIMER est utilisé pour représenter une période d'attente. Lorsque le service TIMER est utilisé, les champs Producteur d'informations et Récepteur d'informations doivent faire référence au même acteur. – REPEAT est utilisé pour indiquer qu'une série d'étapes est répétée jusqu'à une condition ou un évènement déclencheur. La condition est spécifiée sous forme de texte dans la colonne "Event" pour cette ligne ou étape. Après le mot REPEAT doivent apparaître entre parenthèses le premier et le dernier numéro d'étape de la série à répéter, au format suivant: REPEAT(X-Y) où X est la première étape et Y la dernière étape. Ces définitions de services communs font référence aux systèmes d'automatisation/ d'information ou de communication. Si le formulaire type de modèle de cas d'utilisation est appliqué dans d'autres domaines, d'autres services peuvent être utilisés et décrits.
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 2 présente toutes les extrémités d'association d'Activity avec d'autres classes.

Tableau 2 – Extrémités d'association de UseCaseCoreConcepts::Activity avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..*] Requirement	Requirement	Plusieurs exigences peuvent être énumérées dans une étape, séparées par une virgule.
[0..*]	[0..*] InformationReceiver	Actor	Identifie le récepteur des informations. Il convient qu'il s'agisse de l'un des acteurs définis dans le référentiel.
[0..*]	[0..*] InformationProducer	Actor	Identifie le producteur ou la source des informations. Il convient qu'il s'agisse de l'un des acteurs définis dans le référentiel.
[0..*]	[0..*] BusinessObject	BusinessObject	Décrit brièvement les informations à échanger entre les deux acteurs – producteur d'informations et récepteur d'informations. Plusieurs éléments d'informations peuvent être énumérés dans une étape, séparés par une virgule.
[0..1]	[0..*] Drawing	Drawing	liste de dessins associés
[0..*]	[0..*] PrimaryActor	Actor	référence au PrimaryActor
[0..*]	[0..*] ChildActivity	Activity	activités enfants
[0..*]	[0..1] Scenario	Scenario	référence au scénario
[0..*]	[0..*] Activity	Activity	activités parentes

6.1.2.3 Actor

Il s'agit de l'entité qui communique et interagit.

Ces acteurs peuvent inclure les personnes, applications logicielles, systèmes, bases de données, voire le système d'alimentation lui-même.

Le Tableau 3 présente tous les attributs d'Actor.

Tableau 3 – Attributs de UseCaseCoreConcepts::Actor

name	type	description
type	String	type de l'acteur (professionnel, système, ...)
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 4 présente toutes les extrémités d'association d'Actor avec d'autres classes.

Tableau 4 – Extrémités d'association de UseCaseCoreConcepts::Actor avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..1]	[0..*] Child	Actor	liste des acteurs spécialisés
[0..*]	[0..*] Activity	Activity	les activités liées à l'acteur
[0..*]	[0..*] Activity	Activity	les activités liées à cet acteur
[0..*]	[0..*] Activity	Activity	Un acteur peut être lié à une activité.
[0..*]	[0..*] Scenario	Scenario	L'acteur est lié à des scénarii.
[0..*]	[0..*] ActorLibrary	ActorLibrary	L'acteur est référencé par des bibliothèques.
[0..*]	[0..*] UseCase	UseCase	liste des cas d'utilisation liés à un acteur
[0..*]	[0..*] FurtherActorInformation	FurtherActorInformation	Un acteur peut disposer d'une liste d'informations complémentaires partagée entre des regroupements de différents cas d'utilisation.
[0..*]	[0..1] Parent	Actor	l'acteur généralisé

6.1.2.4 ActorGrouping

Groupe d'acteurs utilisé pour organiser une liste d'acteurs

Le Tableau 5 présente tous les attributs d'ActorGrouping.

Tableau 5 – Attributs de UseCaseCoreConcepts::ActorGrouping

name	type	description
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 6 présente toutes les extrémités d'association d'ActorGrouping avec d'autres classes.

Tableau 6 – Extrémités d'association de UseCaseCoreConcepts::ActorGrouping avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..*] UseCase	UseCase	ActorGrouping est associé à plusieurs cas d'utilisation.
[0..1]	[0..*] FurtherActorInformation	FurtherActorInformation	Dans le contexte d'un seul cas d'utilisation, un acteur placé dans un ActorGrouping peut avoir un FurtherActorInformation.

6.1.2.5 ActorLibrary

Acteurs du référentiel

Le Tableau 7 présente tous les attributs d'ActorLibrary.

Tableau 7 – Attributs de UseCaseCoreConcepts::ActorLibrary

name	type	description
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 8 présente toutes les extrémités d'association d'ActorLibrary avec d'autres classes.

Tableau 8 – Extrémités d'association de UseCaseCoreConcepts::ActorLibrary avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..*] Actor	Actor	les acteurs figurant dans la bibliothèque
[0..*]	[1..1] UseCaseRepository	UseCaseRepository	référentiel de cas d'utilisation associé

6.1.2.6 Area

Domaine de connaissances ou d'activité caractérisé par un ensemble de concepts et termes compris par les praticiens dans ce secteur.

Le Tableau 9 présente tous les attributs d'Area.

Tableau 9 – Attributs de UseCaseCoreConcepts::Area

name	type	description
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 10 présente toutes les extrémités d'association d'Area avec d'autres classes.

Tableau 10 – Extrémités d'association de UseCaseCoreConcepts::Area avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1] ParentArea	Area	lien avec un secteur parent
[0..*]	[1..1] AreaLibrary	AreaLibrary	bibliothèque associée au secteur
[0..*]	[0..*] BusinessCase	BusinessCase	cas d'affaire connexes associés au domaine
[0..1]	[0..*] SubArea	Area	lien avec un sous-secteur

6.1.2.7 AreaLibrary

Domaines du référentiel

Le Tableau 11 présente tous les attributs d'AreaLibrary.

Tableau 11 – Attributs de UseCaseCoreConcepts::AreaLibrary

name	type	description
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 12 présente toutes les extrémités d'association d'AreaLibrary avec d'autres classes.

Tableau 12 – Extrémités d'association de UseCaseCoreConcepts::AreaLibrary avec d'autres classes

[mult from]	[mult to] nom	type	description
[1..1]	[0..*] Area	Area	liste des domaines à l'intérieur de la bibliothèque
[0..1]	[1..1] UseCaseRepository	UseCaseRepository	le référentiel associé

6.1.2.8 Assumption

Elle peut être utilisée pour définir une hypothèse générale pour un cas d'utilisation, par exemple, quels systèmes existent déjà, quelles relations contractuelles existent et quelles configurations de systèmes sont probablement en place. Les états initiaux des informations échangées doivent également être identifiés.

Le Tableau 13 présente tous les attributs d'Assumption.

Tableau 13 – Attributs de UseCaseCoreConcepts::Assumption

name	type	description
number	String	indice de l'hypothèse
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 14 présente toutes les extrémités d'association d'Assumption avec d'autres classes.

Tableau 14 – Extrémités d'association de UseCaseCoreConcepts::Assumption avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1] UseCase	UseCase	lien vers le cas d'utilisation connexe

6.1.2.9 Author

Ce champ est utilisé pour documenter la personne qui a fourni la version actuelle. Il peut s'agir d'une personne, d'une organisation ou par exemple d'un comité de normalisation comme un comité d'études (TC) ou un groupe de travail (WG).

Le Tableau 15 présente tous les attributs d'Author.

Tableau 15 – Attributs de UseCaseCoreConcepts::Author

name	type	description
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 16 présente toutes les extrémités d'association d'Author avec d'autres classes.

Tableau 16 – Extrémités d'association de UseCaseCoreConcepts::Author avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..*] Version	Version	les versions associées auxquelles l'auteur a participé

6.1.2.10 BusinessCase

Donne une description ou référence avec certaines justifications pour le cas d'utilisation suggéré. Le cas d'affaire est généralement associé à plusieurs cas d'utilisation.

Le Tableau 17 présente tous les attributs de BusinessCase.

Tableau 17 – Attributs de UseCaseCoreConcepts::BusinessCase

name	type	description
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 18 présente toutes les extrémités d'association de BusinessCase avec d'autres classes.

Tableau 18 – Extrémités d'association de UseCaseCoreConcepts::BusinessCase avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..*] Area	Area	Les cas d'utilisation peuvent être utilisés dans différents secteurs (par exemple, système énergétique). Dans ces secteurs, différents domaines sont utilisés pour définir/déterminer un sous-groupement plus spécifique. Un ou plusieurs domaines et zones, séparés par des virgules, peuvent être spécifiés dans le modèle.
[0..*]	[0..*] UseCase	UseCase	liste des cas d'utilisation couvrant le cas d'affaire
[0..*]	[0..1] BusinessCaseLibrary	BusinessCaseLibrary	la bibliothèque contenant le cas d'affaire

6.1.2.11 BusinessCaseLibrary

Cas d'affaire du référentiel

Le Tableau 19 présente tous les attributs de BusinessCaseLibrary.

Tableau 19 – Attributs de UseCaseCoreConcepts::BusinessCaseLibrary

name	type	description
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 20 présente toutes les extrémités d'association de BusinessCaseLibrary avec d'autres classes.

Tableau 20 – Extrémités d'association de UseCaseCoreConcepts::BusinessCaseLibrary avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..1]	[0..*] BusinessCase	BusinessCase	la liste des cas d'affaire
[0..1]	[0..1] UseCaseRepository	UseCaseRepository	le référentiel de cas d'utilisation associé à la bibliothèque

6.1.2.12 BusinessObject

Objet métier échangé dans les étapes d'activité détaillées

Le Tableau 21 présente tous les attributs de BusinessObject.

Tableau 21 – Attributs de UseCaseCoreConcepts::BusinessObject

name	type	description
identifiant	String	identification utilisée dans les références de la description de cas d'utilisation
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 22 présente toutes les extrémités d'association de BusinessObject avec d'autres classes.

Tableau 22 – Extrémités d'association de UseCaseCoreConcepts::BusinessObject avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..*] Requirement	Requirement	Il peut être utilisé pour définir les exigences relatives aux informations et non à l'étape comme dans l'analyse étape par étape.
[0..*]	[0..*] Activity	Activity	activité qui utilise cet objet métier échangé
[0..*]	[0..*] InformationModelLibrary	BusinessObjectLibrary	la bibliothèque associée

6.1.2.13 BusinessObjectLibrary

Objets métier échangés dans les cas d'utilisation

Le Tableau 23 présente tous les attributs de BusinessObjectLibrary.

Tableau 23 – Attributs de UseCaseCoreConcepts::BusinessObjectLibrary

name	type	description
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 24 présente toutes les extrémités d'association de BusinessObjectLibrary avec d'autres classes.

Tableau 24 – Extrémités d'association de UseCaseCoreConcepts::BusinessObjectLibrary avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..*] BusinessObject	BusinessObject	liste des objets métier à l'intérieur de la bibliothèque
[0..1]	[1..1] UseCaseRepository	UseCaseRepository	le référentiel associé

6.1.2.14 CommonTerm

Il représente un terme du glossaire et sa définition.

Le Tableau 25 présente tous les attributs de CommonTerm.

Tableau 25 – Attributs de UseCaseCoreConcepts::CommonTerm

name	type	description
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 26 présente toutes les extrémités d'association de CommonTerm avec d'autres classes.

Tableau 26 – Extrémités d'association de UseCaseCoreConcepts::CommonTerm avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..*] UseCase	UseCase	le cas d'utilisation associé
[0..*]	[0..*] CommonTermLibrary	CommonTermLibrary	la bibliothèque associée

6.1.2.15 CommonTermLibrary

Glossaire commun qui définit les termes pour tous les cas d'utilisation

Le Tableau 27 présente tous les attributs de CommonTermLibrary.

Tableau 27 – Attributs de UseCaseCoreConcepts::CommonTermLibrary

name	type	description
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 28 présente toutes les extrémités d'association de CommonTermLibrary avec d'autres classes.

Tableau 28 – Extrémités d'association de UseCaseCoreConcepts::CommonTermLibrary avec d'autres classes

[mult from]	[mult to] nom	type	déscription
[0..*]	[0..*] CommonTerm	CommonTerm	liste des termes communs
[0..*]	[1..1] UseCaseRepository	UseCaseRepository	le référentiel associé

6.1.2.16 Condition

Elle décrit une précondition ou une postcondition.

Le Tableau 29 présente tous les attributs de Condition.

Tableau 29 – Attributs de UseCaseCoreConcepts::Condition

name	type	description
number	String	indice de la condition
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 30 présente toutes les extrémités d'association de Condition avec d'autres classes.

Tableau 30 – Extrémités d'association de UseCaseCoreConcepts::Condition avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1] UseCase	UseCase	le cas d'utilisation associé
[0..*]	[0..1] Scenario	Scenario	référence au scénario
[0..*]	[0..1] Scenario	Scenario	référence au scénario

6.1.2.17 CustomInformation

Il offre une option flexible permettant d'inclure les diverses informations personnalisées et semi-structurées qui n'entrent pas dans les autres parties du modèle.

Le Tableau 31 présente tous les attributs de CustomInformation.

Tableau 31 – Attributs de UseCaseCoreConcepts::CustomInformation

name	type	description
key	String	clé unique utilisée pour les besoins de l'identification
value	String	informations correspondantes pour la clé fournie
reference	String	Ce champ fait référence à la section de modèle correspondante.
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 32 présente toutes les extrémités d'association de CustomInformation avec d'autres classes.

Tableau 32 – Extrémités d'association de UseCaseCoreConcepts::CustomInformation avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1] UseCase	UseCase	le cas d'utilisation associé

6.1.2.18 Drawing

A des fins de clarification, il est généralement recommandé de fournir des dessins manuscrits, graphiques ou sous forme de graphiques UML (option préférentielle dans l'IEC 62559). Dans la mesure du possible, il convient que le dessin montre les interactions qui identifient les étapes.

Le Tableau 33 présente tous les attributs de Drawing.

Tableau 33 – Attributs de UseCaseCoreConcepts::Drawing

name	type	description
drawingType	DrawingClassification	type de dessin
URI	String	Chemin de ressource vers un fichier image ex.: http://www...../image.jpg ou chemin vers un fichier XML avec un chemin de diagramme relatif à partir de la racine du modèle UML (par exemple, http://www.../Model.xml#Package1/Package2/.../DiagName)
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 34 présente toutes les extrémités d'association de Drawing avec d'autres classes.

**Tableau 34 – Extrémités d'association de UseCaseCoreConcepts::
Drawing avec d'autres classes**

[mult from]	[mult to] nom	type	description
[0..*]	[0..1] Activity	Activity	l'activité associée
[0..*]	[1..1] UseCase	UseCase	le cas d'utilisation associé
[0..*]	[0..1] Scenario	Scenario	le scénario associé

6.1.2.19 Enumération DrawingClassification

Types de dessins possibles

Le Tableau 35 présente tous les littéraux de DrawingClassification.

Tableau 35 – Littéraux de UseCaseCoreConcepts::DrawingClassification

littéral	description
domain overview	présentation des relations entre les domaines et les cas d'utilisation
use case overview	présentation des informations de modèle court de cas d'utilisation
scenarios flowchart	diagramme des scénarii dans un cas d'utilisation
documentation	informations supplémentaires relatives à un cas d'utilisation
scenario overview	présentation des informations statiques liées à un scénario
activities flowchart	diagramme des activités de description d'un scénario ou d'une autre activité (dans le cas d'une décomposition récursive)
activity overview	informations statiques liées à une activité
role model	présentation du rôle et des relations
business objects overview	présentation d'un ensemble d'objets utilisés dans les échanges d'informations
other	autre type de dessins

6.1.2.20 FurtherActorInformation

Des informations individuelles ou supplémentaires qui font référence au cas d'utilisation peuvent être fournies.

Le Tableau 36 présente tous les attributs de FurtherActorInformation.

Tableau 36 – Attributs de UseCaseCoreConcepts::FurtherActorInformation

name	type	description
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 37 présente toutes les extrémités d'association de FurtherActorInformation avec d'autres classes.

Tableau 37 – Extrémités d'association de UseCaseCoreConcepts::FurtherActorInformation avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..*] Actor	Actor	informations relatives à un acteur
[0..*]	[0..1] ActorGrouping	ActorGrouping	Un acteur est groupé dans des listes.

6.1.2.21 Classe racine IdentifiedObject

Cette classe permet de rassembler les attributs utilisés pour les besoins de l'identification. Les autres classes de ce paquetage peuvent hériter de cette classe.

Le Tableau 38 présente tous les attributs d'IdentifiedObject.

Tableau 38 – Attributs de UseCaseCoreConcepts::IdentifiedObject

name	type	description
description	String	description de l'objet
mRID	String	identificateur de la ressource principale
name	String	nom court

6.1.2.22 KeyPerformanceIndicator

Indicateur de rendement clé (KPI) lié aux objectifs du cas d'utilisation

Le Tableau 39 présente tous les attributs de KeyPerformanceIndicator.

Tableau 39 – Attributs de UseCaseCoreConcepts::KeyPerformanceIndicator

name	type	description
identifiant	String	identificateur unique pour l'indicateur de rendement clé (KPI)
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 40 présente toutes les extrémités d'association de KeyPerformanceIndicator avec d'autres classes.

Tableau 40 – Extrémités d'association de UseCaseCoreConcepts::KeyPerformanceIndicator avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1] UseCase	UseCase	le cas d'utilisation associé
[0..1]	[0..*] Objective	Objective	l'objectif associé

6.1.2.23 Narrative

Il décrit le texte descriptif du cas d'utilisation.

Le Tableau 41 présente tous les attributs de Narrative.

Tableau 41 – Attributs de UseCaseCoreConcepts::Narrative

name	type	description
shortDescription	String	Il s'agit d'un texte court destiné à résumer l'idée principale pour aider le lecteur qui cherche un cas d'utilisation ou consulte une présentation. Recommandation: Il convient que cette brève description ne comporte pas plus de 150 mots.
completeDescription	String	Fournit un texte descriptif complet du cas d'utilisation du point de vue d'un utilisateur, décrivant ce qui se produit, quand, pourquoi, avec quelle attente et dans quelles conditions. Il convient que ce texte descriptif soit écrit en texte en clair de manière à ce que les non-experts du domaine puissent le comprendre. La longueur de cette description complète peut aller de quelques phrases à quelques pages, selon la complexité et/ou la nouveauté du cas d'utilisation. Cette description aide souvent l'expert du domaine à réfléchir aux exigences du cas d'utilisation avant d'entrer dans le détail dans les sections suivantes du formulaire type de modèle de cas d'utilisation. La description peut inclure des schémas explicatifs.
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 42 présente toutes les extrémités d'association de Narrative avec d'autres classes.

Tableau 42 – Extrémités d'association de UseCaseCoreConcepts::Narrative avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..1]	[1..1] UseCase	UseCase	le cas d'utilisation associé

6.1.2.24 Objective

Objectif du cas d'utilisation

Le Tableau 43 présente tous les attributs d'Objective.

Tableau 43 – Attributs de UseCaseCoreConcepts::Objective

name	type	description
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 44 présente toutes les extrémités d'association d'Objective avec d'autres classes.

Tableau 44 – Extrémités d'association de UseCaseCoreConcepts:: Objective avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1] KeyPerformanceIndicator	KeyPerformanceIndicator	l'indicateur de rendement clé associé à l'objectif
[0..*]	[0..1] UseCase	UseCase	le cas d'utilisation associé

6.1.2.25 Reference

Les références qui peuvent limiter ou affecter la conception, la compréhension et les exigences du cas d'utilisation doivent être identifiées, notamment les contrats, règlements, politiques, aspects financiers, contraintes techniques, contraintes de pollution et autres problèmes de qualité environnementale.

Le Tableau 45 présente tous les attributs de Reference.

Tableau 45 – Attributs de UseCaseCoreConcepts::Reference

name	type	description
impact	String	Cette information décrit la nature de l'influence du document sur le cas d'utilisation.
link	String	Le cas échéant, un lien public vers le document peut être fourni.
number	String	indice de la référence
originatorOrganization	String	Décrit le nom de l'organisme de publication du document.
status	String	statut du document référencé
type	String	Il existe différents types de références (par exemple, norme, règlement, contrat, autres comme les publications).
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 46 présente toutes les extrémités d'association de Reference avec d'autres classes.

Tableau 46 – Extrémités d'association de UseCaseCoreConcepts:: Reference avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..*] UseCase	UseCase	le cas d'utilisation associé

6.1.2.26 Remark

Utilisé pour les commentaires qui ne sont pas pris en considération ailleurs.

Le Tableau 47 présente tous les attributs de Remark.

Tableau 47 – Attributs de UseCaseCoreConcepts::Remark

name	type	description
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 48 présente toutes les extrémités d'association de Remark avec d'autres classes.

Tableau 48 – Extrémités d'association de UseCaseCoreConcepts::Remark avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[1..1] UseCase	UseCase	le cas d'utilisation associé

6.1.2.27 Requirement

Exigence d'une partie du cas d'utilisation

Le Tableau 49 présente tous les attributs de Requirement.

Tableau 49 – Attributs de UseCaseCoreConcepts::Requirement

name	type	description
identifiant	String	identification utilisée dans les références
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 50 présente toutes les extrémités d'association de Requirement avec d'autres classes.

Tableau 50 – Extrémités d'association de UseCaseCoreConcepts::Requirement avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..*] Activity	Activity	l'activité associée
[0..*]	[0..*] BusinessObject	BusinessObject	l'objet métier associé à l'exigence
[0..*]	[0..1] RequirementCategory	RequirementCategory	la catégorie d'exigences associée
[0..*]	[0..1] Scenario	Scenario	Le scénario associé

6.1.2.28 RequirementCategory

Les exigences peuvent être triées en catégories. Pour les catégories complexes, le nom de catégorie peut être "délimité par un point".

Le Tableau 51 présente tous les attributs de RequirementCategory.

Tableau 51 – Attributs de UseCaseCoreConcepts::RequirementCategory

name	type	description
identifiant	String	identification utilisée dans les références
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 52 présente toutes les extrémités d'association de RequirementCategory avec d'autres classes.

Tableau 52 – Extrémités d'association de UseCaseCoreConcepts::RequirementCategory avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..1]	[0..*] Requirement	Requirement	Informations relatives à l'exigence
[0..1]	[0..*] SubCategory	RequirementCategory	les sous-catégories d'exigences associées
[0..*]	[1..1] RequirementLibrary	RequirementLibrary	la bibliothèque associée
[0..*]	[0..1] ParentCategory	RequirementCategory	La référence à la catégorie parente de l'exigence

6.1.2.29 RequirementLibrary

Exigences du référentiel

Le Tableau 53 présente tous les attributs de RequirementLibrary.

Tableau 53 – Attributs de UseCaseCoreConcepts::RequirementLibrary

name	type	description
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 54 présente toutes les extrémités d'association de RequirementLibrary avec d'autres classes.

Tableau 54 – Extrémités d'association de UseCaseCoreConcepts::RequirementLibrary avec d'autres classes

[mult from]	[mult to] nom	type	description
[1..1]	[0..*] RequirementCategory	RequirementCategory	la catégorie de l'exigence
[0..1]	[1..1] UseCaseRepository	UseCaseRepository	le référentiel associé

6.1.2.30 Scenario

Séquence possible d'interactions

Le Tableau 55 présente tous les attributs de Scenario.

Tableau 55 – Attributs de UseCaseCoreConcepts::Scenario

name	type	description
number	String	indice du scénario
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 56 présente toutes les extrémités d'association de Scenario avec d'autres classes.

Tableau 56 – Extrémités d'association de UseCaseCoreConcepts::Scenario avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..1]	[0..*] Activity	Activity	Liste des activités pour le scénario
[0..*]	[0..*] PrimaryActor	Actor	Décrit l'acteur qui déclenche ce scénario.
[0..1]	[0..*] Precondition	Condition	Décrit à quelle(s) condition(s) il convient de satisfaire avant que ce scénario ne se produise.
[0..1]	[0..*] Postcondition	Condition	Décrit quelle(s) condition(s) il convient de faire prévaloir après que ce scénario s'est produit. Les postconditions peuvent également définir les conditions de "réussite" ou de "défaillance" pour chaque cas d'utilisation.
[0..1]	[0..*] Drawing	Drawing	liste de dessins associés
[0..1]	[0..*] Requirement	Requirement	Informations relatives à l'exigence
[0..*]	[1..1] UseCase	UseCase	le cas d'utilisation associé
[0..1]	[0..*] TriggeringEvent	TriggeringEvent	Liste des événements déclencheurs pour le scénario

6.1.2.31 TriggeringEvent

Décrit l'évènement qui déclenche ce scénario.

Le Tableau 57 présente tous les attributs de TriggeringEvent.

Tableau 57 – Attributs de UseCaseCoreConcepts::TriggeringEvent

name	type	description
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 58 présente toutes les extrémités d'association de TriggeringEvent avec d'autres classes.

Tableau 58 – Extrémités d'association de UseCaseCoreConcepts::TriggeringEvent avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1] Scenario	Scenario	Référence au scénario présentant ce déclencheur

6.1.2.32 UseCase

Spécification d'un ensemble d'actions effectuées par un système, qui obtient un résultat observable qui est généralement intéressant pour un ou plusieurs acteurs ou autres intervenants du système

Le Tableau 59 présente tous les attributs de UseCase.

Tableau 59 – Attributs de UseCaseCoreConcepts::UseCase

name	type	description
classification	String	Au niveau international, la description du cas d'utilisation pourrait être suffisamment générique pour décrire un cas d'utilisation de manière plus générale, indépendamment du marché national ou régional. Mais les cas d'utilisation pourraient être utilisés pour décrire les circonstances spécifiques régionales ou nationales comme les lois ou les détails spécifiques au projet. Si le cas d'utilisation reflète ces circonstances, il convient de le caractériser en conséquence.
identifiant	String	Le numéro d'identification (ID) d'un cas d'utilisation est unique dans un référentiel ou projet et sert à l'organisation/l'administration des cas d'utilisation.
keywords	String	Les mots-clés peuvent être définis pour soutenir les fonctionnalités de recherche avancées dans un référentiel de cas d'utilisation. Il convient de fournir plusieurs mots-clés sous forme de liste séparée par une virgule.
levelOfDepth	String	Les cas d'utilisation peuvent être décrits à différents niveaux.
nature	String	Classe l'objectif principal du cas d'utilisation.
prioritization	String	Si un plus grand nombre de cas d'utilisation est pris en compte, il pourrait être intéressant de les regrouper par priorité. Cette définition de l'ordre de priorité peut être différente d'un pays à l'autre.
scope	String	Le périmètre définit les limites du cas d'utilisation.
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 60 présente toutes les extrémités d'association de UseCase avec d'autres classes.

Tableau 60 – Extrémités d'association de UseCaseCoreConcepts::UseCase avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..*] PrimaryActor	Actor	identification du PrimaryActor lié au cas d'utilisation
[0..1]	[0..*] Assumption	Assumption	ensemble des hypothèses relatives au cas d'utilisation
[0..*]	[0..*] BusinessCase	BusinessCase	liste des cas d'affaire associés au cas d'utilisation
[0..*]	[0..*] CommonTerm	CommonTerm	termes communs associés
[1..1]	[0..*] Drawing	Drawing	liste de dessins associés
[1..1]	[0..1] Narrative	Narrative	informations relatives à la partie narrative du cas d'utilisation
[0..1]	[0..*] RelatedObjective	Objective	liste des objectifs du cas d'utilisation
[0..*]	[0..*] Reference	Reference	informations relatives aux documents référencés
[1..1]	[0..*] Remark	Remark	remarque associée au cas d'utilisation
[1..1]	[0..*] Version	Version	liste des versions décrivant les modifications du cas d'utilisation
[0..1]	[0..*] RelatedUseCase	UseCase	Des relations connues avec d'autres cas d'utilisation peuvent être fournies ici. Par exemple, si le cas d'utilisation est un cas d'utilisation plus détaillé par rapport à un cas d'utilisation de haut niveau ou s'il s'agit d'une variante à un cas d'utilisation existant.
[0..*]	[0..*] ActorGrouping	ActorGrouping	Un cas d'utilisation fait référence à plusieurs ensembles d'acteurs.
[0..1]	[0..*] Prerequisite	Condition	Décrit la ou les condition(s) qu'il convient de satisfaire avant l'initiation du cas d'utilisation (par exemple, l'état préalable des acteurs et activités).
[0..1]	[0..*] CustomInformation	CustomInformation	informations supplémentaires connexes
[0..1]	[0..*] KeyPerformanceIndicator	KeyPerformanceIndicator	informations relatives à l'indicateur de rendement clé (KPI) lié au cas d'utilisation
[1..1]	[0..*] Scenario	Scenario	référence aux scénarii appartenant au cas d'utilisation
[0..*]	[1..1] UseCaseLibrary	UseCaseLibrary	référence à la bibliothèque associée
[0..*]	[0..1] UseCase	UseCase	cas d'utilisation parent

6.1.2.33 UseCaseLibrary

Cas d'utilisation du référentiel

Le Tableau 61 présente tous les attributs de UseCaseLibrary.

Tableau 61 – Attributs de UseCaseCoreConcepts::UseCaseLibrary

name	type	description
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 62 présente toutes les extrémités d'association de UseCaseLibrary avec d'autres classes.

Tableau 62 – Extrémités d'association de UseCaseCoreConcepts::UseCaseLibrary avec d'autres classes

[mult from]	[mult to] nom	type	description
[1..1]	[0..*] UseCase	UseCase	liste des cas d'utilisation à l'intérieur de la bibliothèque
[0..1]	[1..1] UseCaseRepository	UseCaseRepository	le référentiel associé

6.1.2.34 UseCaseRepository

Base de données, en fonction d'un formulaire type de modèle de cas d'utilisation donné, pour l'édition, la maintenance et l'administration de cas d'utilisation, d'acteurs et d'exigences, notamment leurs relations respectives

Le Tableau 63 présente tous les attributs de UseCaseRepository.

Tableau 63 – Attributs de UseCaseCoreConcepts::UseCaseRepository

name	type	description
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 64 présente toutes les extrémités d'association de UseCaseRepository avec d'autres classes.

Tableau 64 – Extrémités d'association de UseCaseCoreConcepts::UseCaseRepository avec d'autres classes

[mult from]	[mult to] nom	type	description
[1..1]	[0..*] ActorLibrary	ActorLibrary	bibliothèques d'acteurs
[1..1]	[0..1] AreaLibrary	AreaLibrary	la bibliothèque de domaines associée
[1..1]	[0..1] BusinessObjectLibrary	BusinessObjectLibrary	bibliothèque d'objets métier
[1..1]	[0..1] RequirementLibrary	RequirementLibrary	la bibliothèque associée
[1..1]	[0..1] UseCaseLibrary	UseCaseLibrary	bibliothèque de cas d'utilisation associée
[0..1]	[0..1] BusinessCaseLibrary	BusinessCaseLibrary	bibliothèque associée au référentiel
[1..1]	[0..*] CommonTermLibrary	CommonTermLibrary	bibliothèque de termes communs associée

6.1.2.35 Version

version de la description du cas d'utilisation

Le Tableau 65 présente tous les attributs de Version.

Tableau 65 – Attributs de UseCaseCoreConcepts::Version

name	type	description
approvalStatus	String	informations utilisées dans le processus de normalisation
changes	String	Représente les différences par rapport à la version précédente. Les multiples modifications sont séparées par des alinéas.
date	DateTime	date de création de la version, au format AAAA-MM-JJ.
number	String	numéro séquentiel pour identifier la version du document
description	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 66 présente toutes les extrémités d'association de Version avec d'autres classes.

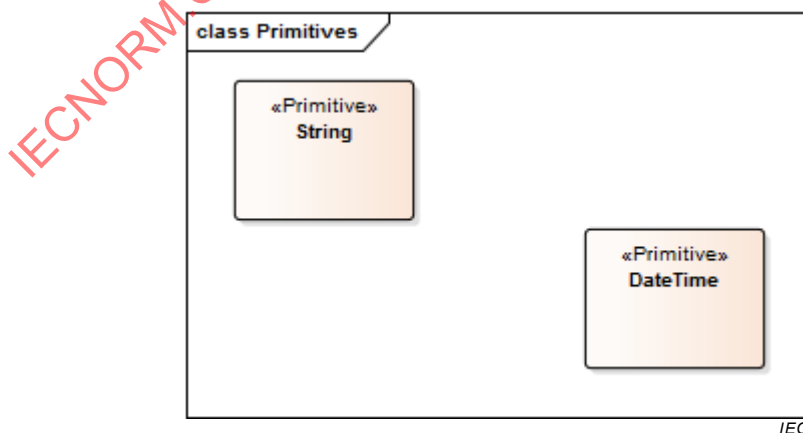
Tableau 66 – Extrémités d'association de UseCaseCoreConcepts::Version avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..*] Author	Author	liste des auteurs participant aux modifications de cette version
[0..*]	[1..1] UseCase	UseCase	informations relatives au cas d'utilisation associé

6.1.2.36 Primitives de paquetage supérieur

6.1.2.36.1 Généralités

La Figure 17 présente le diagramme de classe Primitives.



Anglais	Français
Class Primitives	Classe Primitives

Figure 17 – Diagramme de classe Primitives::Primitives

6.1.2.36.2 Primitive DateTime

Type de primitive DateTime

6.1.2.36.3 Primitive String

Type de primitive String

6.2 Format d'échange de bibliothèque de cas d'utilisation**6.2.1 Modèle d'échange de bibliothèque de cas d'utilisation UML (modèle contextuel UML)****6.2.1.1 Généralités**

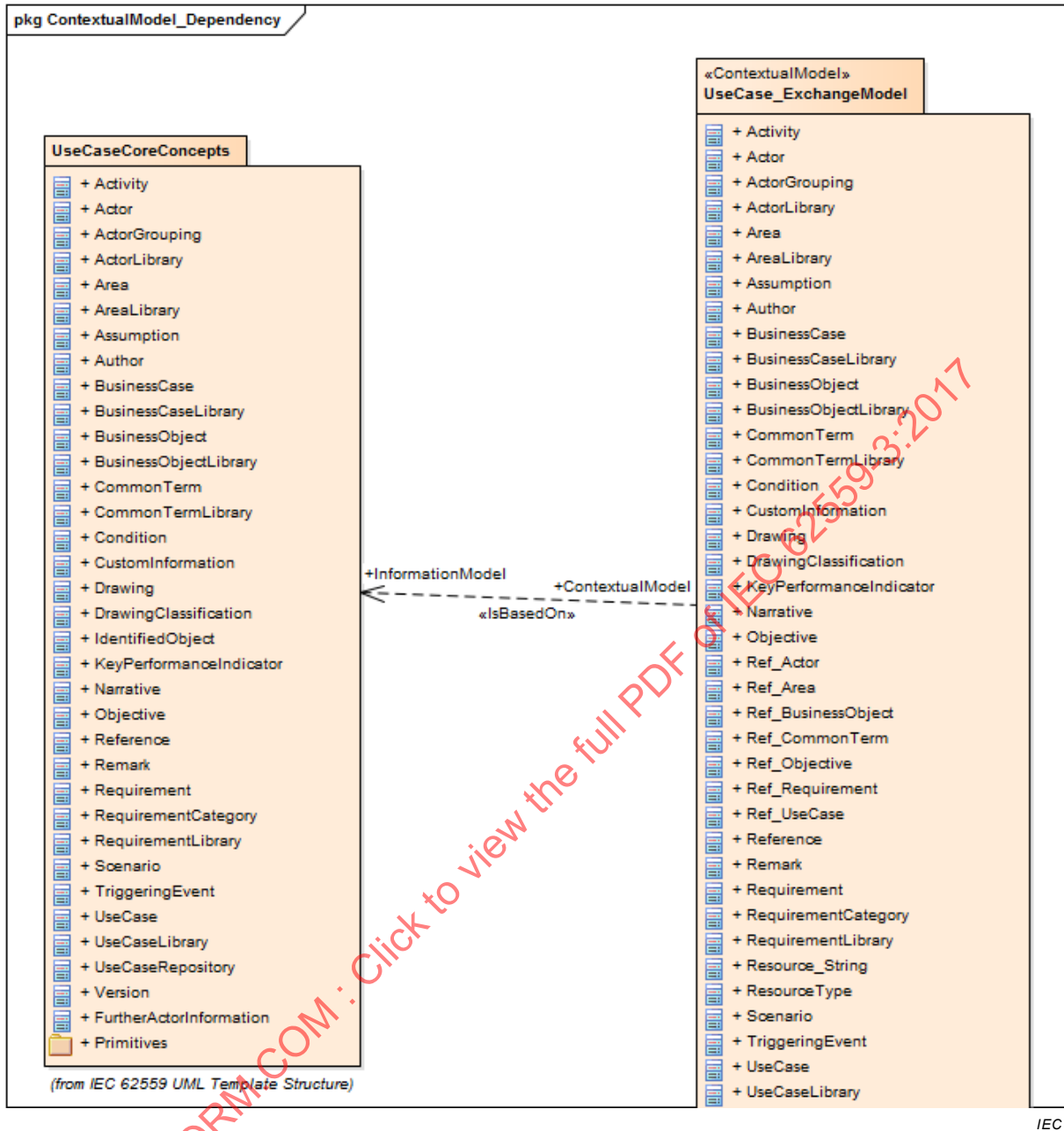
Le modèle d'échange de bibliothèque de cas d'utilisation est décrit à l'aide d'un modèle contextuel UML. Le contenu XSD est généré à partir de ce modèle.

6.2.1.2 Paquetage de modèle UseCase_ExchangeModel**6.2.1.2.1 Généralités**

Modèle contextuel UML de l'IEC 62559 pour le modèle d'échange d'informations de cas d'utilisation.

La Figure 18 présente le diagramme de paquetage ContextualModel_Dependency.

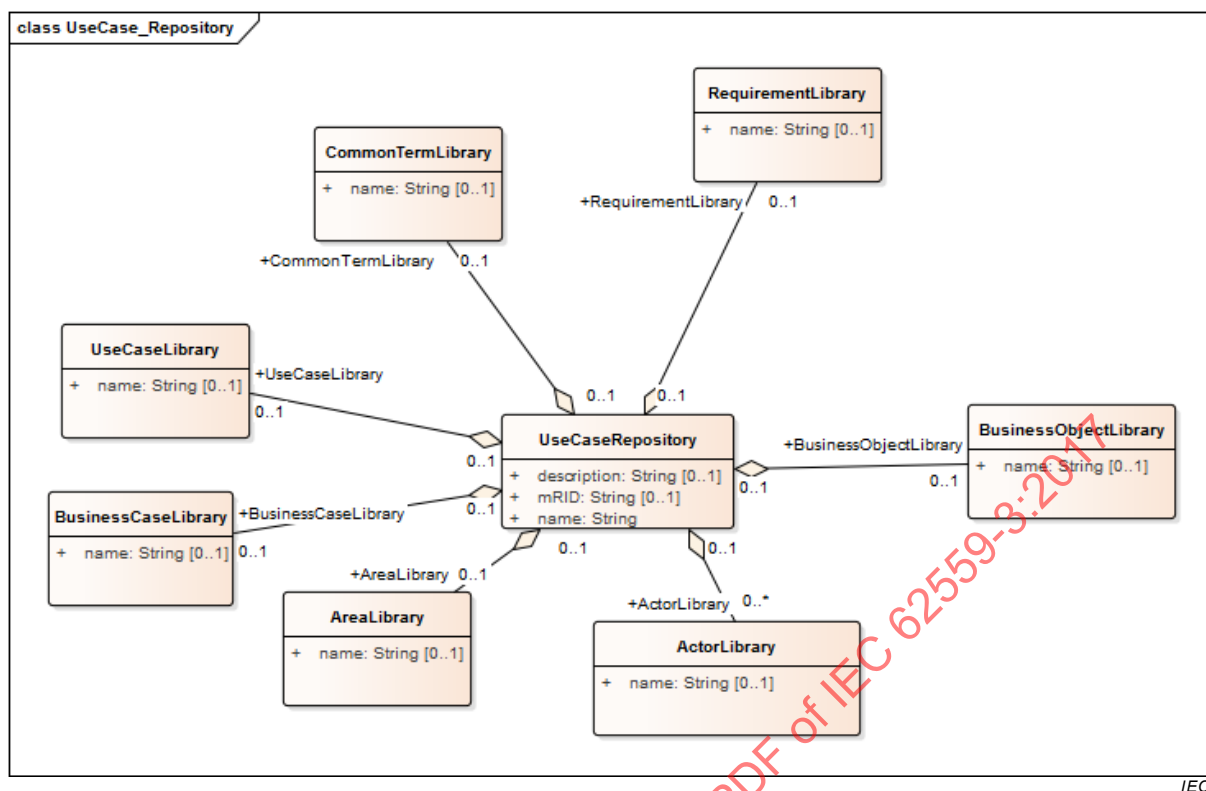
IECNORM.COM : Click to view the full PDF of IEC 62559-3:2017



Anglais	Français
pkg ContextualModel_Dependency	paquetage ContextualModel_Dependency

**Figure 18 – Diagramme de paquetage
UseCase_ExchangeModel::ContextualModel_Dependency**

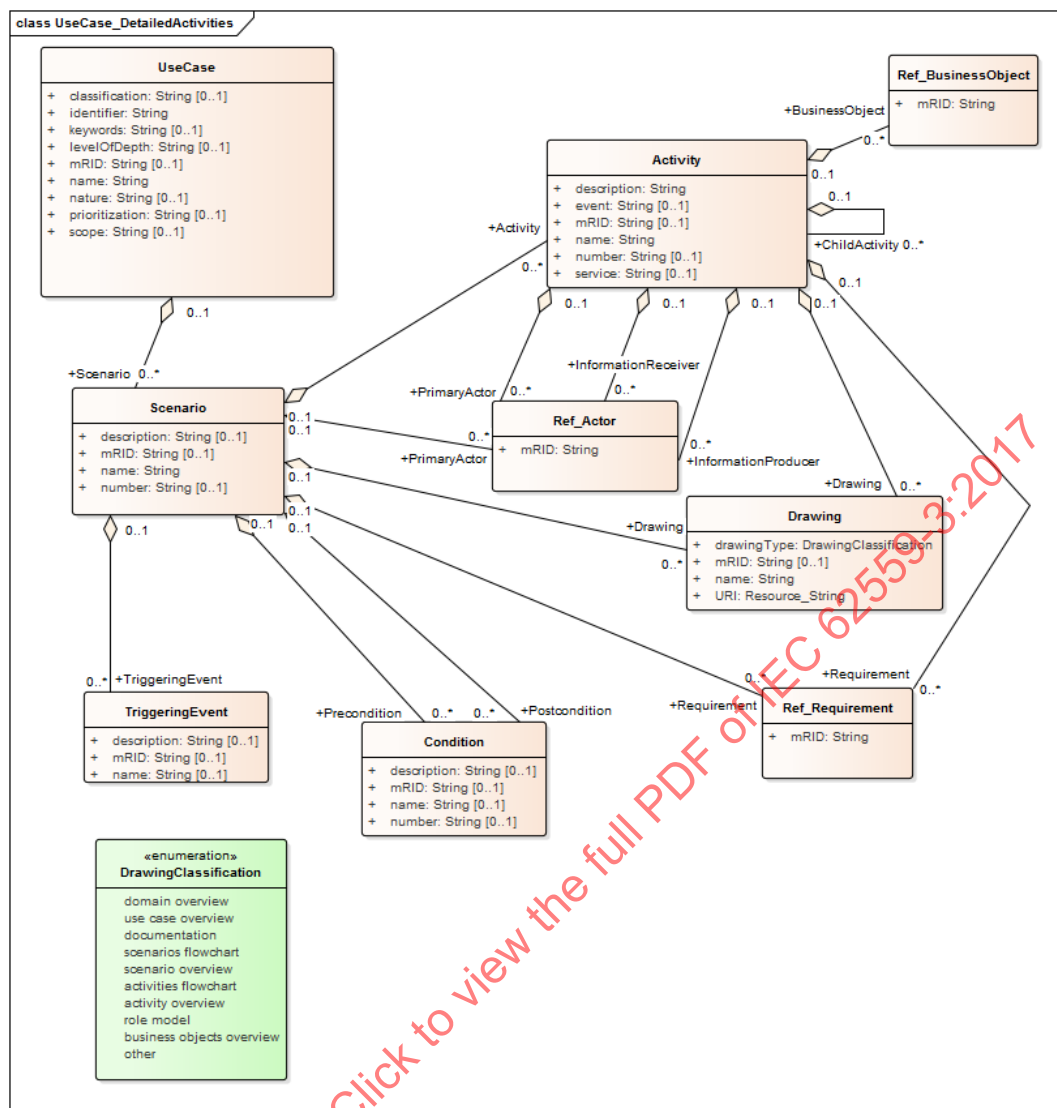
La Figure 19 présente le diagramme de classe UseCase_Repository.



Anglais	Français
Class UseCase_Repository	Classe UseCase_Repository

Figure 19 – Diagramme de classe UseCase_ExchangeModel::UseCase_Repository

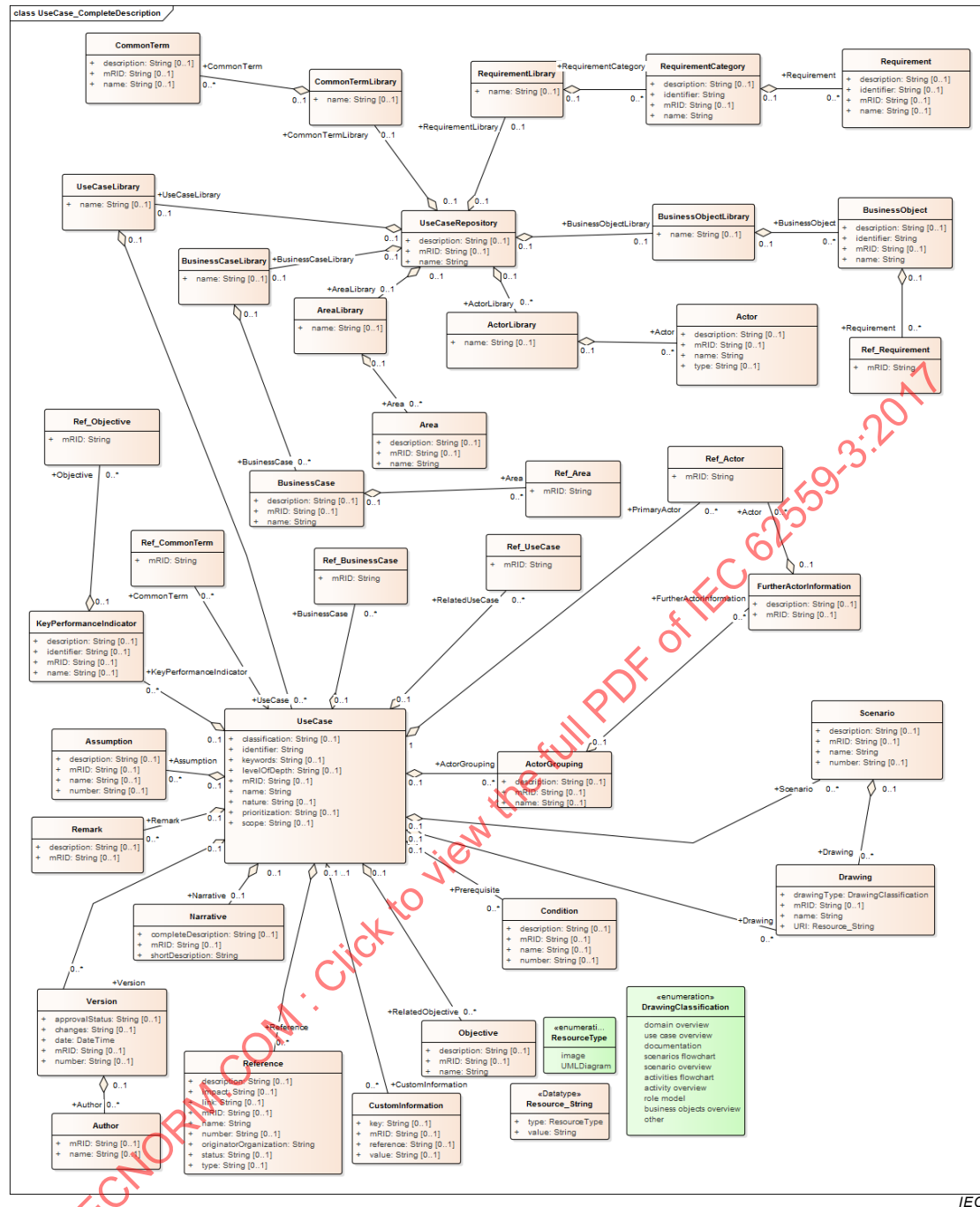
La Figure 20 présente le diagramme de classe UseCase_ShortDescription.



Anglais	Français
class UseCase_Detailed Activities	classe UseCase_Detailed Activities
other	autre

**Figure 21 – Diagramme de classe
UseCase_ExchangeModel::UseCase_DetailedActivities**

La Figure 22 présente le diagramme de classe UseCase_CompleteDescription.



IEC

Anglais	Français
class UseCase_CompleteDescription	classe UseCase_CompleteDescription
other	autre

Figure 22 – Diagramme de classe
UseCase_ExchangeModel::UseCase_CompleteDescription

6.2.1.2.2 Classe racine Activity

Elle peut être divisée en d'autres activités ou représenter une étape élémentaire du scénario.

Le Tableau 67 présente tous les attributs d'Activity.

Tableau 67 – Attributs de UseCase_ExchangeModel::Activity

name	type	description
name	String	nom court
description	String	description de l'objet
event	String	Il s'agit de l'évènement qui déclenche l'étape. Il pourrait s'agir de l'achèvement d'étapes précédentes (auquel cas l'évènement est la liste séparée par une virgule de leurs nombres) ou de la condition de sortie d'une boucle.
service	String	Cette colonne identifie la nature du flux d'informations et l'auteur des informations. Les options disponibles sont CREATE, GET, CHANGE, DELETE, CANCEL, EXECUTE déduites de l'IEC 61968-100:2013. De plus, REPORT, TIMER et REPEAT sont suggérés. – CREATE signifie qu'un objet d'information doit être créé au niveau du producteur. – GET (il s'agit de la valeur par défaut si rien n'est renseigné) signifie que le récepteur demande des informations au producteur (par défaut). – CHANGE signifie que les informations doivent être mises à jour. Le producteur met à jour les informations du récepteur. – DELETE signifie que les informations doivent être supprimées. Le producteur supprime les informations du récepteur. – CANCEL et CLOSE impliquent des actions relatives aux processus, telles que la fermeture d'un ordre d'exécution ou l'annulation d'une demande de contrôle. – EXECUTE est utilisé lorsqu'une transaction complexe est en cours de transmission au moyen d'un service, qui contient éventuellement plusieurs verbes. – REPORT est utilisé pour représenter le transfert d'informations non sollicitées ou de flux d'informations asynchrones. Le producteur donne des informations au récepteur. – TIMER est utilisé pour représenter une période d'attente. Lorsque le service TIMER est utilisé, les champs Producteur d'informations et Récepteur d'informations doivent faire référence au même acteur. – REPEAT est utilisé pour indiquer qu'une série d'étapes est répétée jusqu'à une condition ou un évènement déclencheur. La condition est spécifiée sous forme de texte dans la colonne "Event" pour cette ligne ou étape. Après le mot REPEAT doivent apparaître entre parenthèses le premier et le dernier numéro d'étape de la série à répéter, au format suivant: REPEAT(X-Y) où X est la première étape et Y la dernière étape. Ces définitions de services communs font référence aux systèmes d'automatisation/ d'information ou de communication. Si le formulaire type de modèle de cas d'utilisation est appliqué dans d'autres domaines, d'autres services peuvent être utilisés et décrits.
mRID	String	identificateur de la ressource principale

Le Tableau 68 présente toutes les extrémités d'association d'Activity avec d'autres classes.

Tableau 68 – Extrémités d'association de UseCase_ExchangeModel::Activity avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1]	Scenario	Cette extrémité d'association est une agrégation.
[0..*]	[0..1]	Activity	Cette extrémité d'association est une agrégation.
[0..1]	[0..*] InformationReceiver	Ref_Actor	Identifie le récepteur des informations. Il convient qu'il s'agisse de l'un des acteurs définis dans le référentiel.
[0..1]	[0..*] InformationProducer	Ref_Actor	Identifie le producteur ou la source des informations. Il convient qu'il s'agisse de l'un des acteurs définis dans le référentiel.
[0..1]	[0..*] Requirement	Ref_Requirement	Plusieurs exigences peuvent être énumérées dans une étape, séparées par une virgule.
[0..1]	[0..*] BusinessObject	Ref_BusinessObject	Décrit brièvement les informations à échanger entre les deux acteurs – producteur d'informations et récepteur d'informations. Plusieurs éléments d'informations peuvent être énumérés dans une étape, séparés par une virgule.
[0..1]	[0..*] Drawing	Drawing	liste de dessins associés
[0..1]	[0..*] PrimaryActor	Ref_Actor	référence au PrimaryActor
[0..1]	[0..*] ChildActivity	Activity	activités enfants

6.2.1.2.3 Classe racine Actor

Il s'agit de l'entité qui communique et interagit.

Ces acteurs peuvent inclure les personnes, applications logicielles, systèmes, bases de données, voire le système d'alimentation lui-même.

Le Tableau 69 présente tous les attributs d'Actor.

Tableau 69 – Attributs de UseCase_ExchangeModel::Actor

name	type	description
name	String	nom court
type	String	type de l'acteur (professionnel, système, ...)
mRID	String	identificateur de la ressource principale

Le Tableau 70 présente toutes les extrémités d'association d'Actor avec d'autres classes.

Tableau 70 – Extrémités d'association de UseCase_ExchangeModel::Actor avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1]	ActorLibrary	Cette extrémité d'association est une agrégation.

6.2.1.2.4 Classe racine ActorGrouping

Groupe d'acteurs utilisé pour organiser une liste d'acteurs

Le Tableau 71 présente tous les attributs d'ActorGrouping.

Tableau 71 – Attributs de UseCase_ExchangeModel::ActorGrouping

name	type	description
name	String	nom court
description	String	description de l'objet
mRID	String	identificateur de la ressource principale

Le Tableau 72 présente toutes les extrémités d'association d'ActorGrouping avec d'autres classes.

Tableau 72 – Extrémités d'association de UseCase_ExchangeModel::ActorGrouping avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..1]	[0..*] FurtherActorInformation	FurtherActorInformation	Dans le contexte d'un seul cas d'utilisation, un acteur placé dans un ActorGrouping peut avoir un FurtherActorInformation.
[0..*]	[0..1]	UseCase	Cette extrémité d'association est une agrégation.

6.2.1.2.5 Classe racine ActorLibrary

Acteurs du référentiel

Le Tableau 73 présente tous les attributs d'ActorLibrary.

Tableau 73 – Attributs de UseCase_ExchangeModel::ActorLibrary

name	type	description
name	String	nom court

Le Tableau 74 présente toutes les extrémités d'association d'ActorLibrary avec d'autres classes.

Tableau 74 – Extrémités d'association de UseCase_ExchangeModel::ActorLibrary avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1]	UseCaseRepository	Cette extrémité d'association est une agrégation.
[0..1]	[0..*] Actor	Actor	les acteurs figurant dans la bibliothèque

6.2.1.2.6 Classe racine Area

Domaine de connaissances ou d'activité caractérisé par un ensemble de concepts et termes compris par les praticiens dans ce secteur.

Le Tableau 75 présente tous les attributs d'Area.

Tableau 75 – Attributs de UseCase_ExchangeModel::Area

name	type	description
name	String	nom court
description	String	description de l'objet
mRID	String	identificateur de la ressource principale

Le Tableau 76 présente toutes les extrémités d'association d'Area avec d'autres classes.

Tableau 76 – Extrémités d'association de UseCase_ExchangeModel::Area avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1]	AreaLibrary	Cette extrémité d'association est une agrégation.

6.2.1.2.7 Classe racine AreaLibrary

Domaines du référentiel

Le Tableau 77 présente tous les attributs d'AreaLibrary.

Tableau 77 – Attributs de UseCase_ExchangeModel::AreaLibrary

name	type	description
name	String	nom court

Le Tableau 78 présente toutes les extrémités d'association d'AreaLibrary avec d'autres classes.

Tableau 78 – Extrémités d'association de UseCase_ExchangeModel::AreaLibrary avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..1]	[0..1]	UseCaseRepository	Cette extrémité d'association est une agrégation.
[0..1]	[0..*] Area	Area	liste des domaines à l'intérieur de la bibliothèque

6.2.1.2.8 Classe racine Assumption

Elle peut être utilisée pour définir une hypothèse générale pour un cas d'utilisation, par exemple, quels systèmes existent déjà, quelles relations contractuelles existent et quelles configurations de systèmes sont probablement en place. Les états initiaux des informations échangées doivent également être identifiés.

Le Tableau 79 présente tous les attributs d'Assumption.

Tableau 79 – Attributs de UseCase_ExchangeModel::Assumption

name	type	description
name	String	nom court
number	String	indice de l'hypothèse
description	String	description de l'objet
mRID	String	identificateur de la ressource principale

Le Tableau 80 présente toutes les extrémités d'association d'Assumption avec d'autres classes.

Tableau 80 – Extrémités d'association de UseCase_ExchangeModel::Assumption avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1]	UseCase	Cette extrémité d'association est une agrégation.

6.2.1.2.9 Classe racine Author

Ce champ est utilisé pour documenter la personne qui a fourni la version actuelle. Il peut s'agir d'une personne, d'une organisation ou par exemple d'un comité de normalisation comme un comité d'études (TC) ou un groupe de travail (WG).

Le Tableau 81 présente tous les attributs d'Author.

Tableau 81 – Attributs de UseCase_ExchangeModel::Author

name	type	description
name	String	nom court
mRID	String	identificateur de la ressource principale

Le Tableau 82 présente toutes les extrémités d'association d'Author avec d'autres classes.

Tableau 82 – Extrémités d'association de UseCase_ExchangeModel::Author avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1]	Version	Cette extrémité d'association est une agrégation.

6.2.1.2.10 Classe racine BusinessCase

Donne une description ou référence avec certaines justifications pour le cas d'utilisation suggéré. Le cas d'affaire est généralement associé à plusieurs cas d'utilisation.

Le Tableau 83 présente tous les attributs de BusinessCase.

Tableau 83 – Attributs de UseCase_ExchangeModel::BusinessCase

name	type	description
name	String	nom court
description	String	description de l'objet
mRID	String	identificateur de la ressource principale

Le Tableau 84 présente toutes les extrémités d'association de BusinessCase avec d'autres classes.

Tableau 84 – Extrémités d'association de UseCase_ExchangeModel::BusinessCase avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1]	BusinessCaseLibrary	Cette extrémité d'association est une agrégation.
[0..1]	[0..*] Area	Ref_Area	Les cas d'utilisation peuvent être utilisés dans différents secteurs (par exemple, système énergétique). Dans ces secteurs, différents domaines sont utilisés pour définir/déterminer un sous-groupement plus spécifique. Un ou plusieurs domaines et zones, séparés par des virgules, peuvent être spécifiés dans le modèle.

6.2.1.2.11 Classe racine BusinessCaseLibrary

Cas d'affaire du référentiel

Le Tableau 85 présente tous les attributs de BusinessCaseLibrary.

Tableau 85 – Attributs de UseCase_ExchangeModel::BusinessCaseLibrary

name	type	description
name	String	nom court

Le Tableau 86 présente toutes les extrémités d'association de BusinessCaseLibrary avec d'autres classes.

Tableau 86 – Extrémités d'association de UseCase_ExchangeModel::BusinessCaseLibrary avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..1]	[0..1]	UseCaseRepository	Cette extrémité d'association est une agrégation.
[0..1]	[0..*] BusinessCase	BusinessCase	la liste des cas d'affaire

6.2.1.2.12 Classe racine BusinessObject

Objet métier échangé dans les étapes d'activité détaillées

Le Tableau 87 présente tous les attributs de BusinessObject.

Tableau 87 – Attributs de UseCase_ExchangeModel::BusinessObject

name	type	description
name	String	nom court
description	String	description de l'objet
identifier	String	identification utilisée dans les références de la description de cas d'utilisation
mRID	String	identificateur de la ressource principale

Le Tableau 88 présente toutes les extrémités d'association de BusinessObject avec d'autres classes.

Tableau 88 – Extrémités d'association de UseCase_ExchangeModel::BusinessObject avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1]	BusinessObjectLibrary	Cette extrémité d'association est une agrégation.
[0..1]	[0..*] Requirement	Ref_Requirement	Il peut être utilisé pour définir les exigences relatives aux informations et non à l'étape comme dans l'analyse étape par étape.

6.2.1.2.13 Classe racine BusinessObjectLibrary

Objets métier échangés dans les cas d'utilisation

Le Tableau 89 présente tous les attributs de BusinessObjectLibrary.

Tableau 89 – Attributs de UseCase_ExchangeModel::BusinessObjectLibrary

name	type	description
name	String	nom court

Le Tableau 90 présente toutes les extrémités d'association de BusinessObjectLibrary avec d'autres classes.

Tableau 90 – Extrémités d'association de UseCase_ExchangeModel::BusinessObjectLibrary avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..1]	[0..1]	UseCaseRepository	Cette extrémité d'association est une agrégation.
[0..1]	[0..*] BusinessObject	BusinessObject	liste des objets métier à l'intérieur de la bibliothèque

6.2.1.2.14 Classe racine CommonTerm

Il représente un terme du glossaire et sa définition.

Le Tableau 91 présente tous les attributs de CommonTerm.

Tableau 91 – Attributs de UseCase_ExchangeModel::CommonTerm

name	type	description
name	String	nom court
description	String	description de l'objet
mRID	String	identificateur de la ressource principale

Le Tableau 92 présente toutes les extrémités d'association de CommonTerm avec d'autres classes.

Tableau 92 – Extrémités d'association de UseCase_ExchangeModel::CommonTerm avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1]	CommonTermLibrary	Cette extrémité d'association est une agrégation.

6.2.1.2.15 Classe racine CommonTermLibrary

Glossaire commun qui définit les termes pour tous les cas d'utilisation.

Le Tableau 93 présente tous les attributs de CommonTermLibrary.

Tableau 93 – Attributs de UseCase_ExchangeModel::CommonTermLibrary

name	type	description
name	String	nom court

Le Tableau 94 présente toutes les extrémités d'association de CommonTermLibrary avec d'autres classes.

Tableau 94 – Extrémités d'association de UseCase_ExchangeModel::CommonTermLibrary avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..1]	[0..1]	UseCaseRepository	Cette extrémité d'association est une agrégation.
[0..1]	[0..*] CommonTerm	CommonTerm	liste des termes communs

6.2.1.2.16 Classe racine Condition

Elle décrit une précondition ou une postcondition.

Le Tableau 95 présente tous les attributs de Condition.

Tableau 95 – Attributs de UseCase_ExchangeModel::Condition

name	type	description
name	String	nom court
number	String	indice de la condition
description	String	description de l'objet
mRID	String	identificateur de la ressource principale

Le Tableau 96 présente toutes les extrémités d'association de Condition avec d'autres classes.

Tableau 96 – Extrémités d'association de UseCase_ExchangeModel::Condition avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1]	Scenario	Cette extrémité d'association est une agrégation.
[0..*]	[0..1]	Scenario	Cette extrémité d'association est une agrégation.
[0..*]	[0..1]	UseCase	Cette extrémité d'association est une agrégation.

6.2.1.2.17 Classe racine CustomInformation

Il offre une option flexible permettant d'inclure les diverses informations personnalisées et semi-structurées qui n'entrent pas dans les autres parties du modèle.

Le Tableau 97 présente tous les attributs de CustomInformation.

Tableau 97 – Attributs de UseCase_ExchangeModel::CustomInformation

name	type	description
key	String	clé unique utilisée pour les besoins de l'identification
value	String	informations correspondantes pour la clé fournie
reference	String	Ce champ fait référence à la section de modèle correspondante.
mRID	String	identificateur de la ressource principale

Le Tableau 98 présente toutes les extrémités d'association de CustomInformation avec d'autres classes.

Tableau 98 – Extrémités d'association de UseCase_ExchangeModel::CustomInformation avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1]	UseCase	Cette extrémité d'association est une agrégation.

6.2.1.2.18 Classe racine Drawing

A des fins de clarification, il est généralement recommandé de fournir des dessins manuscrits, graphiques ou sous forme de graphiques UML (option préférentielle dans l'IEC 62559). Dans la mesure du possible, il convient que le dessin montre les interactions qui identifient les étapes.

Le Tableau 99 présente tous les attributs de Drawing.

Tableau 99 – Attributs de UseCase_ExchangeModel::Drawing

name	type	description
name	String	nom court
drawingType	DrawingClassification	type de dessin
URI	Resource_String	Chemin de ressource vers un fichier image ex: http://www..../image.jpg ou chemin vers un fichier XML avec un chemin de diagramme relatif à partir de la racine du modèle UML (par exemple, http://www.../Model.xmi#Package1/Package2/.../DiagName)
mRID	String	identificateur de la ressource principale

Le Tableau 100 présente toutes les extrémités d'association de Drawing avec d'autres classes.

Tableau 100 – Extrémités d'association de UseCase_ExchangeModel::Drawing avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1]	Activity	Cette extrémité d'association est une agrégation.
[0..*]	[0..1]	UseCase	Cette extrémité d'association est une agrégation.
[0..*]	[0..1]	Scenario	Cette extrémité d'association est une agrégation.

6.2.1.2.19 Enumération DrawingClassification

Types de dessins possibles.

Le Tableau 101 présente tous les littéraux de DrawingClassification.

Tableau 101 – Littéraux de UseCase_ExchangeModel::DrawingClassification

littéral	description
domain overview	présentation des relations entre les domaines et les cas d'utilisation
use case overview	présentation des informations de modèle court de cas d'utilisation
documentation	informations supplémentaires relatives à un cas d'utilisation
scenarios flowchart	diagramme des scénarii dans un cas d'utilisation
scenario overview	présentation des informations statiques liées à un scénario
activities flowchart	diagramme des activités de description d'un scénario ou d'une autre activité (dans le cas d'une décomposition récursive)
activity overview	informations statiques liées à une activité
role model	présentation du rôle et des relations
business objects overview	présentation d'un ensemble d'objets utilisés dans les échanges d'informations
other	autre type de dessins

6.2.1.2.20 Classe racine KeyPerformanceIndicator

Indicateur de rendement clé (KPI) lié aux objectifs du cas d'utilisation.

Le Tableau 102 présente tous les attributs de KeyPerformanceIndicator.

Tableau 102 – Attributs de UseCase_ExchangeModel::KeyPerformanceIndicator

name	type	description
name	String	nom court
description	String	description de l'objet
identifiant	String	identificateur unique pour l'indicateur de rendement clé (KPI)
mRID	String	identificateur de la ressource principale

Le Tableau 103 présente toutes les extrémités d'association de KeyPerformanceIndicator avec d'autres classes.

Tableau 103 – Extrémités d'association de UseCase_ExchangeModel::KeyPerformanceIndicator avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1]	UseCase	Cette extrémité d'association est une agrégation.
[0..1]	[0..*] Objective	Ref_Objective	l'objectif associé

6.2.1.2.21 Classe racine Narrative

Il décrit le texte descriptif du cas d'utilisation.

Le Tableau 104 présente tous les attributs de Narrative.

Tableau 104 – Attributs de UseCase_ExchangeModel::Narrative

name	type	description
shortDescription	String	Il s'agit d'un texte court destiné à résumer l'idée principale pour aider le lecteur qui cherche un cas d'utilisation ou consulte une présentation. Recommandation: Il convient que cette brève description ne comporte pas plus de 150 mots.
completeDescription	String	Fournit un texte descriptif complet du cas d'utilisation du point de vue d'un utilisateur, décrivant ce qui se produit, quand, pourquoi, avec quelle attente et dans quelles conditions. Il convient que ce texte descriptif soit écrit en texte en clair de manière à ce que les non-experts du domaine puissent le comprendre. La longueur de cette description complète peut aller de quelques phrases à quelques pages, selon la complexité et/ou la nouveauté du cas d'utilisation. Cette description aide souvent l'expert du domaine à réfléchir aux exigences du cas d'utilisation avant d'entrer dans le détail dans les sections suivantes du formulaire type de modèle de cas d'utilisation. La description peut inclure des schémas explicatifs.
mRID	String	identificateur de la ressource principale

Le Tableau 105 présente toutes les extrémités d'association de Narrative avec d'autres classes.

Tableau 105 – Extrémités d'association de UseCase_ExchangeModel::Narrative avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..1]	[0..1]	UseCase	Cette extrémité d'association est une agrégation.

6.2.1.2.22 Classe racine Objective

Objectif du cas d'utilisation.

Le Tableau 106 présente tous les attributs d'Objective.

Tableau 106 – Attributs de UseCase_ExchangeModel::Objective

name	type	description
name	String	nom court
description	String	description de l'objet
mRID	String	identificateur de la ressource principale

Le Tableau 107 présente toutes les extrémités d'association d'Objective avec d'autres classes.

Tableau 107 – Extrémités d'association de UseCase_ExchangeModel:: Objective avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1]	UseCase	Cette extrémité d'association est une agrégation.

6.2.1.2.23 Classe racine Ref_Actor

Il s'agit de l'entité qui communique et interagit.

Ces acteurs peuvent inclure les personnes, applications logicielles, systèmes, bases de données, voire le système d'alimentation lui-même.

Le Tableau 108 présente tous les attributs de Ref_Actor.

Tableau 108 – Attributs de UseCase_ExchangeModel::Ref_Actor

name	type	description
mRID	String	identificateur de la ressource principale

Le Tableau 109 présente toutes les extrémités d'association de Ref_Actor avec d'autres classes.

Tableau 109 – Extrémités d'association de UseCase_ExchangeModel:: Ref_Actor avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1]	Activity	Cette extrémité d'association est une agrégation.
[0..*]	[0..1]	Activity	Cette extrémité d'association est une agrégation.
[0..*]	[0..1]	Activity	Cette extrémité d'association est une agrégation.
[0..*]	[0..1]	Scenario	Cette extrémité d'association est une agrégation.
[0..*]	[0..1]	FurtherActorInformation	Cette extrémité d'association est une agrégation.
[0..*]	[1..1]	UseCase	Cette extrémité d'association est une agrégation.

6.2.1.2.24 Classe racine Ref_Area

Domaine de connaissances ou activité caractérisé(e) par un ensemble de concepts et termes compris par les praticiens dans ce secteur.

Le Tableau 110 présente tous les attributs de Ref_Area.

Tableau 110 – Attributs de UseCase_ExchangeModel::Ref_Area

name	type	description
mRID	String	identificateur de la ressource principale

Le Tableau 111 présente toutes les extrémités d'association de Ref_Area avec d'autres classes.

Tableau 111 – Extrémités d'association de UseCase_ExchangeModel::Ref_Area avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1]	BusinessCase	Cette extrémité d'association est une agrégation.

6.2.1.2.25 Classe racine Ref_BusinessCase

Donne une description ou référence avec certaines justifications pour le cas d'utilisation suggéré. Le cas d'affaire est généralement associé à plusieurs cas d'utilisation.

Le Tableau 112 présente tous les attributs de Ref_BusinessCase.

Tableau 112 – Attributs de UseCase_ExchangeModel::Ref_BusinessCase

name	type	description
mRID	String	identificateur de la ressource principale

Le Tableau 113 présente toutes les extrémités d'association de Ref_BusinessCase avec d'autres classes.

Tableau 113 – Extrémités d'association de UseCase_ExchangeModel::Ref_BusinessCase avec d'autres classes

[mult from]	[mult to] nom	type	description
[0..*]	[0..1]	UseCase	Cette extrémité d'association est une agrégation.

6.2.1.2.26 Classe racine Ref_BusinessObject

Objet métier échangé dans les étapes d'activité détaillées.

Le Tableau 114 présente tous les attributs de Ref_BusinessObject.

Tableau 114 – Attributs de UseCase_ExchangeModel::Ref_BusinessObject

name	type	description
mRID	String	identificateur de la ressource principale

Le Tableau 115 présente toutes les extrémités d'association de Ref_BusinessObject avec d'autres classes.