

	DIN EN IEC 62559-3 (VDE 0175-103)	DIN
	Diese Norm ist zugleich eine VDE-Bestimmung im Sinne von VDE 0022. Sie ist nach Durchführung des vom VDE-Präsidium beschlossenen Genehmigungsverfahrens unter der oben angeführten Nummer in das VDE-Vorschriftenwerk aufgenommen und in der „etz Elektrotechnik + Automation“ bekannt gegeben worden.	VDE
<p>ICS 29.020</p> <p>Anwendungsfallmethodik – Teil 3: Definition von Artefakten der Anwendungsfallvorlage in einem in XML-serialisierten Format (IEC 62559-3:2017); Deutsche Fassung EN IEC 62559-3:2018</p> <p>Use case methodology – Part 3: Definition of use case template artefacts into an XML serialized format (IEC 62559-3:2017); German version EN IEC 62559-3:2018</p> <p>Méthodologie des cas d'utilisation – Partie 3: Définition des artefacts de formulaire type de modèle de cas d'utilisation au format sérialisé XML (IEC 62559-3:2017); Version allemande EN IEC 62559-3:2018</p> <p>Gesamtumfang 109 Seiten</p> <p>DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE</p>		

Anwendungsbeginn

Anwendungsbeginn für die von CENELEC am 2018-01-17 angenommene Europäische Norm als DIN-Norm ist 2019-02-01.

Nationales Vorwort

Vorausgegangener Norm-Entwurf: E DIN EN 62559-3 (VDE 0175-103):2015-07.

Für dieses Dokument ist das nationale Arbeitsgremium K 261 „Systemaspekte der elektrischen Energieversorgung“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE (www.dke.de) zuständig.

Die enthaltene IEC-Publikation wurde vom TC 8 „System aspects of electrical energy supply“ erarbeitet.

Das IEC-Komitee hat entschieden, dass der Inhalt dieses Dokuments bis zu dem Datum (stability date) unverändert bleiben soll, das auf der IEC-Website unter „<http://webstore.iec.ch>“ zu diesem Dokument angegeben ist. Zu diesem Zeitpunkt wird entsprechend der Entscheidung des Komitees das Dokument

- bestätigt,
- zurückgezogen,
- durch eine Folgeausgabe ersetzt oder
- geändert.

Das Original-Dokument enthält Bilder und Tabellen in Farbe, die in der Papierversion in einer Graustufen-Darstellung wiedergegeben werden. Elektronische Versionen dieses Dokuments enthalten die Bilder und Tabellen in der originalen Farbdarstellung.

Diese IEC-Norm beinhaltet Codekomponenten, d. h. Komponenten, die dafür vorgesehen sind, direkt von einem Computer verarbeitet zu werden. Derlei Inhalte befinden sich zwischen den Markierungen <CODE BEGINS> und <CODE ENDS> oder sind eindeutig als Codekomponenten gekennzeichnet.

An den Kauf dieser IEC-Norm ist eine Urheberrechtsslizenz gebunden, mit welcher der Käufer Software mit Codekomponenten aus dieser Norm direkt oder über Vertriebs Händler an den Endanwender verkaufen kann, und die den IEC-Softwarelizenzbedingungen unterliegt, welche unter folgender Adresse verfügbar sind: <http://www.iec.ch/webstore/custserv/pdf/CCv1.pdf>.

Nationaler Anhang NA (informativ)

Zusammenhang mit europäischen und internationalen Dokumenten

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf ein Dokument ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils aktuellste Ausgabe des in Bezug genommenen Dokuments.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe des Dokuments.

Eine Information über den Zusammenhang der zitierten Dokumente mit den entsprechenden deutschen Dokumenten ist nachstehend wiedergegeben.

Tabelle NA.1

Europäisches Dokument	Internationales Dokument	Deutsches Dokument	Klassifikation im VDE-Vorschriftenwerk
EN 61968-100:2013	IEC 61968-100:2013	DIN EN 61968-100:2014-08	–
EN 62325-450	IEC 62325-450	DIN EN 62325-450	–
EN 62559 (alle Teile)	IEC 62559 (alle Teile)	DIN EN 62559 (VDE 0175) (alle Teile)	VDE 0175
EN 62559-1 ^a	IEC 62559-1	DIN EN 62559-1 (VDE 0175-101) ^a	VDE 0175-101
EN 62559-2	IEC 62559-2	DIN EN 62559-2 (VDE 0175-102)	VDE 0175-102
^a In Vorbereitung.			

Nationaler Anhang NB (informativ)

Literaturhinweise

DIN EN 61968-100:2014-08, *Integration von Anwendungen in Anlagen der Elektrizitätsversorgung – Systemschnittstellen für Netzführung – Teil 100: Implementations-Profile (IEC 61968-100:2013); Englische Fassung EN 61968-100:2013*

DIN EN 62325-450, *Kommunikation im Energiemarkt – Teil 450: Profil- und Kontext-Modellierungsregeln*

DIN EN 62559 (VDE 0175) (alle Teile), *Anwendungsfallmethodik*

DIN EN 62559-1 (VDE 0175-101) (inVorbereitung), *Anwendungsfallmethodik – Teil 1: Konzept und Prozesse in der Normung*

DIN EN 62559-2 (VDE 0175-102), *Anwendungsfallmethodik – Teil 2: Definition der Anwendungsfallvorlage, Akteurliste und der Anforderungsliste*

Anwendungsfallmethodik –
Teil 3: Definition von Artefakten der Anwendungsfallvorlage in einem in XML-
serialisierten Format
(IEC 62559-3:2017)

Use case methodology –
Part 3: Definition of use case template artefacts
into an XML serialized format
(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
(IEC 62559-3:2017)

Diese Europäische Norm wurde von CENELEC am 2018-01-17 angenommen. CENELEC-Mitglieder sind gehalten, die CEN/CENELEC-Geschäftsordnung zu erfüllen, in der die Bedingungen festgelegt sind, unter denen dieser Europäischen Norm ohne jede Änderung der Status einer nationalen Norm zu geben ist.

Auf dem letzten Stand befindliche Listen dieser nationalen Normen mit ihren bibliographischen Angaben sind beim CEN-CENELEC Management Centre oder bei jedem CENELEC-Mitglied auf Anfrage erhältlich.

Diese Europäische Norm besteht in drei offiziellen Fassungen (Deutsch, Englisch, Französisch). Eine Fassung in einer anderen Sprache, die von einem CENELEC-Mitglied in eigener Verantwortung durch Übersetzung in seine Landessprache gemacht und dem CEN-CENELEC Management Centre mitgeteilt worden ist, hat den gleichen Status wie die offiziellen Fassungen.

CENELEC-Mitglieder sind die nationalen elektrotechnischen Komitees von Belgien, Bulgarien, Dänemark, Deutschland, der ehemaligen jugoslawischen Republik Mazedonien, Estland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Kroatien, Lettland, Litauen, Luxemburg, Malta, den Niederlanden, Norwegen, Österreich, Polen, Portugal, Rumänien, Schweden, der Schweiz, Serbien, der Slowakei, Slowenien, Spanien, der Tschechischen Republik, der Türkei, Ungarn, dem Vereinigten Königreich und Zypern.



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

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

© 2018 CENELEC – Alle Rechte der Verwertung, gleich in welcher Form und in welchem Verfahren, sind weltweit den Mitgliedern von CENELEC vorbehalten.

Ref. Nr. EN IEC 62559-3:2018 D

Europäisches Vorwort

Der Text des Dokuments SyCSmartEnergy/28/CDV, zukünftige 1. Ausgabe der IEC 62559-3:2017, erarbeitet vom SyC Smart Energy „Smart Energy“, wurde zur parallelen IEC-CENELEC-Abstimmung vorgelegt und von CENELEC als EN IEC 62559-3:2018 angenommen.

Nachstehende Daten wurden festgelegt:

- spätestes Datum, zu dem dieses Dokument auf nationaler Ebene durch Veröffentlichung einer identischen nationalen Norm oder durch Anerkennung übernommen werden muss (dop): 2018-10-17
- spätestes Datum, zu dem nationale Normen, die diesem Dokument entgegenstehen, zurückgezogen werden müssen (dow): 2021-01-17

Es wird auf die Möglichkeit hingewiesen, dass einige Elemente dieses Dokuments Patentrechte berühren können. CENELEC ist nicht dafür verantwortlich, einige oder alle diesbezüglichen Patentrechte zu identifizieren.

Dieses Dokument wurde unter einem Mandat erstellt, das von der Europäischen Kommission und der Europäischen Freihandelszone an CENELEC gegeben wurde.

Anerkennungsnotiz

Der Text der Internationalen Norm IEC 62559-3:2017 wurde von CENELEC ohne irgendeine Abänderung als Europäische Norm angenommen.

In der offiziellen Fassung sind unter „Literaturhinweise“ zu den aufgelisteten Normen die nachstehenden Anmerkungen einzutragen:

- | | | |
|---------------|-----------|--------------------------------|
| IEC 61968-100 | ANMERKUNG | Harmonisiert als EN 61968-100. |
| IEC 62325-450 | ANMERKUNG | Harmonisiert als EN 62325-450. |

Anhang ZA (normativ)

Normative Verweisungen auf internationale Publikationen mit ihren entsprechenden europäischen Publikationen

Die folgenden Dokumente werden im Text in solcher Weise in Bezug genommen, dass einige Teile davon oder ihr gesamter Inhalt Anforderungen des vorliegenden Dokuments darstellen. Bei datierten Verweisungen gilt nur die in Bezug genommene Ausgabe. Bei undatierten Verweisungen gilt die letzte Ausgabe des in Bezug genommenen Dokuments (einschließlich aller Änderungen).

ANMERKUNG 1 Ist eine internationale Publikation durch gemeinsame Abänderungen modifiziert worden, gekennzeichnet durch (mod.), dann gilt die entsprechende EN oder das HD.

ANMERKUNG 2 Aktualisierte Informationen über die in diesem Anhang aufgeführten aktuellen Fassungen der Europäischen Normen sind hier verfügbar: www.cenelec.eu.

<u>Publikation</u>	<u>Jahr</u>	<u>Titel</u>	<u>EN/HD</u>	<u>Jahr</u>
IEC 62559-2	–	Usevcase methodology – Part 2: Definition of use case template, actor list and requirement list	EN 62559-2	–

Inhalt

Seite

Europäisches Vorwort.....	2
Anhang ZA (normativ) Normative Verweisungen auf internationale Publikationen mit ihren entsprechenden europäischen Publikationen	3
Einleitung	11
1 Anwendungsbereich	14
2 Normative Verweisungen	15
3 Begriffe	15
4 Methodischer Rahmen für die Erarbeitung dieser Norm	15
4.1 Modellbasierter Ansatz für die Entwicklung des Datenaustauschformats	15
4.2 Beispiel für den modellbasierten Prozess im Kontext von IEC 62559	17
4.3 Entwicklung spezifischer Datenaustauschprofile	21
5 Anwendung des methodischen Rahmens für die Definition einer Reihe von wichtigen syntaktischen Austauschformaten (Anwendungsfälle, Akteure, Anforderungen)	21
6 Detaillierte Inhalte der Modelle (automatische Erzeugung aus UML-Modellen)	22
6.1 Kernkonzepte der UML-Anwendungsfälle (UML-Informationsmodell)	22
6.1.1 Allgemeines	22
6.1.2 Modellpaket „UseCaseCoreConcepts“	22
6.2 Austauschformat der Anwendungsfall-Bibliothek	49
6.2.1 UML-Austauschmodell der Anwendungsfall-Bibliothek (UML-Kontextmodell)	49
6.2.2 Syntaktisches XML-Format der Anwendungsfall-Bibliothek	77
6.3 Austauschformat der Akteur-Bibliothek	95
6.3.1 UML-Austauschmodell der Akteur-Bibliothek (UML-Kontextmodell)	95
6.3.2 Syntaktisches XML-Format der Akteur-Bibliothek	98
6.4 Austauschformat der Anforderungen-Bibliothek	100
6.4.1 UML-Austauschmodell der Anforderungen-Bibliothek (UML-Kontextmodell)	100
6.4.2 Syntaktisches XML-Format der Anforderungen-Bibliothek	103
Literaturhinweise	105
Bilder	
Bild 1 – Normenreihe IEC 62559	11
Bild 2 – Einheitliches XML-Format für den Import/Export von Anwendungsfallinformationen zwischen einer Vielzahl von Modellierungssoftware und Repositories	14
Bild 3 – Modellbasierter Entwicklungsrahmen für die Definition des XML-basierten Datenaustauschformats nach IEC 62559-3	16
Bild 4 – UML-Informationsmodell für das Beispiel der Beziehung zwischen „Scenario“ und „Actor“	18
Bild 5 – UML-Kontextmodell basierend auf dem Beispiel der Beziehung zwischen „Scenario“ und „Actor“	18
Bild 6 – XML-Schema für das Beispiel der Beziehung zwischen „Scenario“ und „Actor“	20
Bild 7 – Entwicklung wichtiger XML-Austauschformate auf der Basis eines einzelnen UML- Informationsmodells nach IEC 62559	22
Bild 8 – Klassendiagramm „UseCaseCoreConcepts::UseCaseCore_Libraries“	23

Bild 9 – Klassendiagramm „seCaseCoreConcepts::UseCaseCore_ActorLibrarys“	23
Bild 10 – Klassendiagramm „UseCaseCoreConcepts::UseCaseCore_UseCaseLibrary“	24
Bild 11 – Klassendiagramm „UseCaseCoreConcepts::UseCaseCore_DetailedActivities“	24
Bild 12 – Klassendiagramm „UseCaseCoreConcepts::UseCaseCore_AreaLibrary“	25
Bild 13 – Klassendiagramm „UseCaseCoreConcepts::UseCaseCore_BusinessCaseLibrary“	25
Bild 14 – Klassendiagramm „UseCaseCoreConcepts::UseCaseCore_BusinessObjectLibrary“	26
Bild 15 – Klassendiagramm „UseCaseCoreConcepts::UseCaseCore_CommonTermLibrary“	26
Bild 16 – Klassendiagramm „UseCaseCoreConcepts::UseCaseCore_RequirementLibrary“	26
Bild 17 – Klassendiagramm „Primitives::Primitives“	48
Bild 18 – Paketdiagramm „UseCase_ExchangeModel::ContextualModel_Dependency“	50
Bild 19 – Klassendiagramm „UseCase_ExchangeModel::UseCase_Repository“	51
Bild 20 – Klassendiagramm „UseCase_ExchangeModel::UseCase_ShortDescription“	52
Bild 21 – Klassendiagramm „UseCase_ExchangeModel::UseCase_DetailedActivities“	53
Bild 22 – Klassendiagramm „UseCase_ExchangeModel::UseCase_CompleteDescription“	54
Bild 23 – Klassendiagramm „ActorLibrary_ExchangeModel::ActorLibrary“	95
Bild 24 – Paketdiagramm „ActorLibrary_ExchangeModel::ActorLibrary“	96
Bild 25 – Klassendiagramm „ActorLibrary_ExchangeModel::RequirementLibrary“	100
Bild 26 – Paketdiagramm „ActorLibrary_ExchangeModel::RequirementLibrary“	101

Tabellen

Tabelle 1 – Attribute von „UseCaseCoreConcepts::Activity“	27
Tabelle 2 – Assoziationsenden von „UseCaseCoreConcepts::Activity“ mit anderen Klassen.....	28
Tabelle 3 – Attribute von „UseCaseCoreConcepts::Actor“	28
Tabelle 4 – Assoziationsenden von „UseCaseCoreConcepts::Actor“ mit anderen Klassen.....	29
Tabelle 5 – Attribute von „UseCaseCoreConcepts::ActorGrouping“	29
Tabelle 6 – Assoziationsenden von „UseCaseCoreConcepts::ActorGrouping“ mit anderen Klassen.....	29
Tabelle 7 – Attribute von „UseCaseCoreConcepts::ActorLibrary“	30
Tabelle 8 – Assoziationsenden von „UseCaseCoreConcepts::ActorLibrary“ mit anderen Klassen.....	30
Tabelle 9 – Attribute von „UseCaseCoreConcepts::Area“	30
Tabelle 10 – Assoziationsenden von „UseCaseCoreConcepts::Area“ mit anderen Klassen	30
Tabelle 11 – Attribute von „UseCaseCoreConcepts::AreaLibrary“	31
Tabelle 12 – Assoziationsenden von „UseCaseCoreConcepts::AreaLibrary“ mit anderen Klassen	31
Tabelle 13 – Attribute von „UseCaseCoreConcepts::Assumption“	31
Tabelle 14 – Assoziationsenden von „UseCaseCoreConcepts::Assumption“ mit anderen Klassen	31
Tabelle 15 – Attribute von „UseCaseCoreConcepts::Author“	32
Tabelle 16 – Assoziationsenden von „UseCaseCoreConcepts::Author“ mit anderen Klassen	32
Tabelle 17 – Attribute von „UseCaseCoreConcepts::BusinessCase“	32
Tabelle 18 – Assoziationsenden von „UseCaseCoreConcepts::BusinessCase“ mit anderen Klassen	33
Tabelle 19 – Attribute von „UseCaseCoreConcepts::BusinessCaseLibrary“	33

Tabelle 20 – Assoziationsenden von „UseCaseCoreConcepts::BusinessCaseLibrary“ mit anderen Klassen	33
Tabelle 21 – Attribute von „UseCaseCoreConcepts::BusinessObject“	34
Tabelle 22 – Assoziationsenden von „UseCaseCoreConcepts::BusinessObject“ mit anderen Klassen	34
Tabelle 23 – Attribute von „UseCaseCoreConcepts::BusinessObjectLibrary“	34
Tabelle 24 – Assoziationsenden von „UseCaseCoreConcepts::BusinessObjectLibrary“ mit anderen Klassen	34
Tabelle 25 – Attribute von „UseCaseCoreConcepts::CommonTerm“	35
Tabelle 26 – Assoziationsenden von „UseCaseCoreConcepts::CommonTerm“ mit anderen Klassen	35
Tabelle 27 – Attribute von „UseCaseCoreConcepts::CommonTermLibrary“	35
Tabelle 28 – Assoziationsenden von „UseCaseCoreConcepts::CommonTermLibrary“ mit anderen Klassen	35
Tabelle 29 – Attribute von „UseCaseCoreConcepts::Condition“	36
Tabelle 30 – Assoziationsenden von „UseCaseCoreConcepts::Condition“ mit anderen Klassen	36
Tabelle 31 – Attribute von „UseCaseCoreConcepts::CustomInformation“	36
Tabelle 32 – Assoziationsenden von „UseCaseCoreConcepts::CustomInformation“ mit anderen Klassen	36
Tabelle 33 – Attribute von „UseCaseCoreConcepts::Drawing“	37
Tabelle 34 – Assoziationsenden von „UseCaseCoreConcepts::Drawing“ mit anderen Klassen	37
Tabelle 35 – Literale von „UseCaseCoreConcepts::DrawingClassification“	38
Tabelle 36 – Attribute von „UseCaseCoreConcepts::FurtherActorInformation“	38
Tabelle 37 – Assoziationsenden von „UseCaseCoreConcepts::FurtherActorInformation“ mit anderen Klassen	38
Tabelle 38 – Attribute von „UseCaseCoreConcepts::IdentifiedObject“	39
Tabelle 39 – Attribute von „UseCaseCoreConcepts::KeyPerformanceIndicator“	39
Tabelle 40 – Assoziationsenden von „UseCaseCoreConcepts::KeyPerformanceIndicator“ mit anderen Klassen	39
Tabelle 41 – Attribute von „UseCaseCoreConcepts::Narrative“	40
Tabelle 42 – Assoziationsenden von „UseCaseCoreConcepts::Narrative“ mit anderen Klassen	40
Tabelle 43 – Attribute von „UseCaseCoreConcepts::Objective“	40
Tabelle 44 – Assoziationsenden von „UseCaseCoreConcepts::Objective“ mit anderen Klassen	40
Tabelle 45 – Attribute von „UseCaseCoreConcepts::Reference“	41
Tabelle 46 – Assoziationsenden von „UseCaseCoreConcepts::Reference“ mit anderen Klassen	41
Tabelle 47 – Attribute von „UseCaseCoreConcepts::Remark“	41
Tabelle 48 – Assoziationsenden von „UseCaseCoreConcepts::Remark“ mit anderen Klassen	42
Tabelle 49 – Attribute von „UseCaseCoreConcepts::Requirement“	42
Tabelle 50 – Assoziationsenden von „UseCaseCoreConcepts::Requirement“ mit anderen Klassen	42
Tabelle 51 – Attribute von „UseCaseCoreConcepts::RequirementCategory“	42
Tabelle 52 – Assoziationsenden von „UseCaseCoreConcepts::RequirementCategory“ mit anderen Klassen	43
Tabelle 53 – Attribute von „UseCaseCoreConcepts::RequirementLibrary“	43

Tabelle 54 – Assoziationsenden von „UseCaseCoreConcepts::RequirementLibrary“ mit anderen Klassen.....	43
Tabelle 55 – Attribute von „UseCaseCoreConcepts::Scenario“	43
Tabelle 56 – Assoziationsenden von „UseCaseCoreConcepts::Scenario“ mit anderen Klassen.....	44
Tabelle 57 – Attribute von „UseCaseCoreConcepts::TriggeringEvent“	44
Tabelle 58 – Assoziationsenden von „UseCaseCoreConcepts::TriggeringEvent“ mit anderen Klassen.....	44
Tabelle 59 – Attribute von „UseCaseCoreConcepts::UseCase“	45
Tabelle 60 – Assoziationsenden von „UseCaseCoreConcepts::UseCase“ mit anderen Klassen	45
Tabelle 61 – Attribute von „UseCaseCoreConcepts::UseCaseLibrary“	46
Tabelle 62 – Assoziationsenden von „UseCaseCoreConcepts::UseCaseLibrary“ mit anderen Klassen.....	46
Tabelle 63 – Attribute von „UseCaseCoreConcepts::UseCaseRepository“	47
Tabelle 64 – Assoziationsenden von „UseCaseCoreConcepts::UseCaseRepository“ mit anderen Klassen.....	47
Tabelle 65 – Attribute von „UseCaseCoreConcepts::Version“	48
Tabelle 66 – Assoziationsenden von „UseCaseCoreConcepts::Version“ mit anderen Klassen.....	48
Tabelle 67 – Attribute von „UseCase_ExchangeModel::Activity“	55
Tabelle 68 – Assoziationsenden von „UseCase_ExchangeModel::Activity“ mit anderen Klassen.....	56
Tabelle 69 – Attribute von „UseCase_ExchangeModel::Actor“	56
Tabelle 70 – Assoziationsenden von „UseCase_ExchangeModel::Actor“ mit anderen Klassen.....	56
Tabelle 71 – Attribute von „UseCase_ExchangeModel::ActorGrouping“	57
Tabelle 72 – Assoziationsenden von „UseCase_ExchangeModel::ActorGrouping“ mit anderen Klassen.....	57
Tabelle 73 – Attribute von „UseCase_ExchangeModel::ActorLibrary“	57
Tabelle 74 – Assoziationsenden von „UseCase_ExchangeModel::ActorLibrary“ mit anderen Klassen.....	57
Tabelle 75 – Attribute von „UseCase_ExchangeModel::Area“	58
Tabelle 76 – Assoziationsenden von „UseCase_ExchangeModel::Area“ mit anderen Klassen.....	58
Tabelle 77 – Attribute von „UseCase_ExchangeModel::AreaLibrary“	58
Tabelle 78 – Assoziationsenden von „UseCase_ExchangeModel::AreaLibrary“ mit anderen Klassen	58
Tabelle 79 – Attribute von „UseCase_ExchangeModel::Assumption“	58
Tabelle 80 – Assoziationsenden von „UseCase_ExchangeModel::Assumption“ mit anderen Klassen	59
Tabelle 81 – Attribute von „UseCase_ExchangeModel::Author“	59
Tabelle 82 – Assoziationsenden von „UseCase_ExchangeModel::Author“ mit anderen Klassen.....	59
Tabelle 83 – Attribute von „UseCase_ExchangeModel::BusinessCase“	59
Tabelle 84 – Assoziationsenden von „UseCase_ExchangeModel::BusinessCase“ mit anderen Klassen.....	60
Tabelle 85 – Attribute von „UseCase_ExchangeModel::BusinessCaseLibrary“	60
Tabelle 86 – Assoziationsenden von „UseCase_ExchangeModel::BusinessCaseLibrary“ mit anderen Klassen	60
Tabelle 87 – Attribute von „UseCase_ExchangeModel::BusinessObject“	60

Tabelle 88 – Assoziationsenden von „UseCase_ExchangeModel::BusinessObject“ mit anderen Klassen	61
Tabelle 89 – Attribute von „UseCase_ExchangeModel::BusinessObjectLibrary“	61
Tabelle 90 – Assoziationsenden von „UseCase_ExchangeModel::BusinessObjectLibrary“ mit anderen Klassen.....	61
Tabelle 91 – Attribute von „UseCase_ExchangeModel::CommonTerm“	61
Tabelle 92 – Assoziationsenden von „UseCase_ExchangeModel::CommonTerm“ mit anderen Klassen	62
Tabelle 93 – Attribute von „UseCase_ExchangeModel::CommonTermLibrary“	62
Tabelle 94 – Assoziationsenden von „UseCase_ExchangeModel::CommonTermLibrary“ mit anderen Klassen.....	62
Tabelle 95 – Attribute von „UseCase_ExchangeModel::Condition“	62
Tabelle 96 – Assoziationsenden von „UseCase_ExchangeModel::Condition“ mit anderen Klassen	62
Tabelle 97 – Attribute von „UseCase_ExchangeModel::CustomInformation“	63
Tabelle 98 – Assoziationsenden von „UseCase_ExchangeModel::CustomInformation“ mit anderen Klassen	63
Tabelle 99 – Attribute von „UseCase_ExchangeModel::Drawing“	63
Tabelle 100 – Assoziationsenden von „UseCase_ExchangeModel::Drawing“ mit anderen Klassen	64
Tabelle 101 – Literale von „UseCase_ExchangeModel::DrawingClassification“	64
Tabelle 102 – Attribute von „UseCase_ExchangeModel::KeyPerformanceIndicator“.....	64
Tabelle 103 – Assoziationsenden von „UseCase_ExchangeModel::KeyPerformanceIndicator“ mit anderen Klassen.....	65
Tabelle 104 – Attribute von „UseCase_ExchangeModel::Narrative“	65
Tabelle 105 – Assoziationsenden von „UseCase_ExchangeModel::Narrative“ mit anderen Klassen	65
Tabelle 106 – Attribute von „UseCase_ExchangeModel::Objective“	66
Tabelle 107 – Assoziationsenden von „UseCase_ExchangeModel::Objective“ mit anderen Klassen	66
Tabelle 108 – Attribute von „UseCase_ExchangeModel::Ref_Actor“	66
Tabelle 109 – Assoziationsenden von „UseCase_ExchangeModel::Ref_Actor“ mit anderen Klassen	66
Tabelle 110 – Attribute von „UseCase_ExchangeModel::Ref_Area“	67
Tabelle 111 – Assoziationsenden von „UseCase_ExchangeModel::Ref_Area“ mit anderen Klassen	67
Tabelle 112 – Attribute von „UseCase_ExchangeModel::Ref_BusinessCase“	67
Tabelle 113 – Assoziationsenden von „UseCase_ExchangeModel::Ref_BusinessCase“ mit anderen Klassen.....	67
Tabelle 114 – Attribute von „UseCase_ExchangeModel::Ref_BusinessObject“	67
Tabelle 115 – Assoziationsenden von „UseCase_ExchangeModel::Ref_BusinessObject“ mit anderen Klassen.....	67
Tabelle 116 – Attribute von „UseCase_ExchangeModel::Ref_CommonTerm“	68
Tabelle 117 – Assoziationsenden von „UseCase_ExchangeModel::Ref_CommonTerm“ mit anderen Klassen	68
Tabelle 118 – Attribute von „UseCase_ExchangeModel::Ref_Objective“	68
Tabelle 119 – Assoziationsenden von „UseCase_ExchangeModel::Ref_Objective“ mit anderen Klassen	68

Tabelle 120 – Attribute von „UseCase_ExchangeModel::Ref_Requirement“	68
Tabelle 121 – Assoziationsenden von „UseCase_ExchangeModel::Ref_Requirement“ mit anderen Klassen.....	69
Tabelle 122 – Attribute von „UseCase_ExchangeModel::Ref_UseCase“	69
Tabelle 123 – Assoziationsenden von „UseCase_ExchangeModel::Ref_UseCase“ mit anderen Klassen.....	69
Tabelle 124 – Attribute von „UseCase_ExchangeModel::Reference“	70
Tabelle 125 – Assoziationsenden von „UseCase_ExchangeModel::Reference“ mit anderen Klassen	70
Tabelle 126 – Attribute von „UseCase_ExchangeModel::Remark“	70
Tabelle 127 – Assoziationsenden von „UseCase_ExchangeModel::Remark“ mit anderen Klassen.....	70
Tabelle 128 – Attribute von „UseCase_ExchangeModel::Requirement“	71
Tabelle 129 – Assoziationsenden von „UseCase_ExchangeModel::Requirement“ mit anderen Klassen.....	71
Tabelle 130 – Attribute von „UseCase_ExchangeModel::RequirementCategory“	71
Tabelle 131 – Assoziationsenden von „UseCase_ExchangeModel::RequirementCategory“ mit anderen Klassen	71
Tabelle 132 – Attribute von „UseCase_ExchangeModel::RequirementLibrary“	72
Tabelle 133 – Assoziationsenden von „UseCase_ExchangeModel::RequirementLibrary“ mit anderen Klassen	72
Tabelle 134 – Attribute von „UseCase_ExchangeModel::Resource_String“	72
Tabelle 135 – Literale von „UseCase_ExchangeModel::ResourceType“	72
Tabelle 136 – Attribute von „UseCase_ExchangeModel::Scenario“	73
Tabelle 137 – Assoziationsenden von „UseCase_ExchangeModel::Scenario“ mit anderen Klassen	73
Tabelle 138 – Attribute von „UseCase_ExchangeModel::TriggeringEvent“	73
Tabelle 139 – Assoziationsenden von „UseCase_ExchangeModel::TriggeringEvent“ mit anderen Klassen.....	73
Tabelle 140 – Attribute von „UseCase_ExchangeModel::UseCase“	74
Tabelle 141 – Assoziationsenden von „UseCase_ExchangeModel::UseCase“ mit anderen Klassen	74
Tabelle 142 – Attribute von „UseCase_ExchangeModel::UseCaseLibrary“	75
Tabelle 143 – Assoziationsenden von „UseCase_ExchangeModel::UseCaseLibrary“ mit anderen Klassen.....	75
Tabelle 144 – Attribute von „UseCase_ExchangeModel::UseCaseRepository“	76
Tabelle 145 – Assoziationsenden von „UseCase_ExchangeModel::UseCaseRepository“ mit anderen Klassen	76
Tabelle 146 – Attribute von „UseCase_ExchangeModel::Version“	76
Tabelle 147 – Assoziationsenden von „UseCase_ExchangeModel::Version“ mit anderen Klassen	77
Tabelle 148 – Attribute von „UseCase_ExchangeModel::FurtherActorInformation“	77
Tabelle 149 – Assoziationsenden von „UseCase_ExchangeModel::FurtherActorInformation“ mit anderen Klassen	77
Tabelle 150 – Anwendungsfall-Bibliothek XSD	77
Tabelle 151 – Attribute von „ActorLibrary_ExchangeModel::Actor“	97
Tabelle 152 – Assoziationsenden von „ActorLibrary_ExchangeModel::Actor“ mit anderen Klassen	97

Tabelle 153 – Attribute von „ActorLibrary_ExchangeModel::ActorLibrary“	97
Tabelle 154 – Assoziationsenden von „ActorLibrary_ExchangeModel::ActorLibrary“ mit anderen Klassen	97
Tabelle 155 – Attribute von „ActorLibrary_ExchangeModel::Ref_Actor“	98
Tabelle 156 – Assoziationsenden von „ActorLibrary_ExchangeModel::Ref_Actor“ mit anderen Klassen	98
Tabelle 157 – Assoziationsenden von „ActorLibrary_ExchangeModel:: UseCaseRepository“ mit anderen Klassen.....	98
Tabelle 158 – Akteur-Bibliothek XSD	98
Tabelle 159 – Attribute von „RequirementLibrary_ExchangeModel::RequirementLibrary“	101
Tabelle 160 – Assoziationsenden von „RequirementLibrary_ExchangeModel:: RequirementLibrary“ mit anderen Klassen.....	102
Tabelle 161 – Attribute von „RequirementLibrary_ExchangeModel::RequirementCategory“	102
Tabelle 162 – Assoziationsenden von „RequirementLibrary_ExchangeModel:: RequirementCategory“ mit anderen Klassen	102
Tabelle 163 – Attribute von „RequirementLibrary_ExchangeModel::Ref_RequirementCategory“	102
Tabelle 164 – Assoziationsenden von „RequirementLibrary_ExchangeModel:: Ref_RequirementCategory“ mit anderen Klassen	102
Tabelle 165 – Attribute von „RequirementLibrary_ExchangeModel::Requirement“	103
Tabelle 166 – Assoziationsenden von „RequirementLibrary_ExchangeModel::Requirement“ mit anderen Klassen.....	103
Tabelle 167 – Anforderungen-Bibliothek XSD	103

Einleitung

Die Anwendungsfallmethodik dient dazu, komplexe Systeme hinsichtlich ihrer Funktionalitäten, Akteure und Prozesse über verschiedene technische Komitees oder sogar verschiedene Organisationen hinweg zu verstehen. Als Entwicklungswerkzeug kann die Methodik zur Unterstützung der Erarbeitung von Normen verwendet werden, da sie die Analyse von Anforderungen in Bezug auf neue oder vorhandene Normen vereinfacht. Weitere Vorteile der Anwendungsfallmethodik sowie Hintergrundinformationen sind in IEC 62559-1 enthalten.

Bild 1 gibt einen Überblick über die vorgesehenen ersten Teile von IEC 62559 und beschreibt im Wesentlichen den Zusammenhang zwischen IEC 62559-2 und IEC 62559-3.

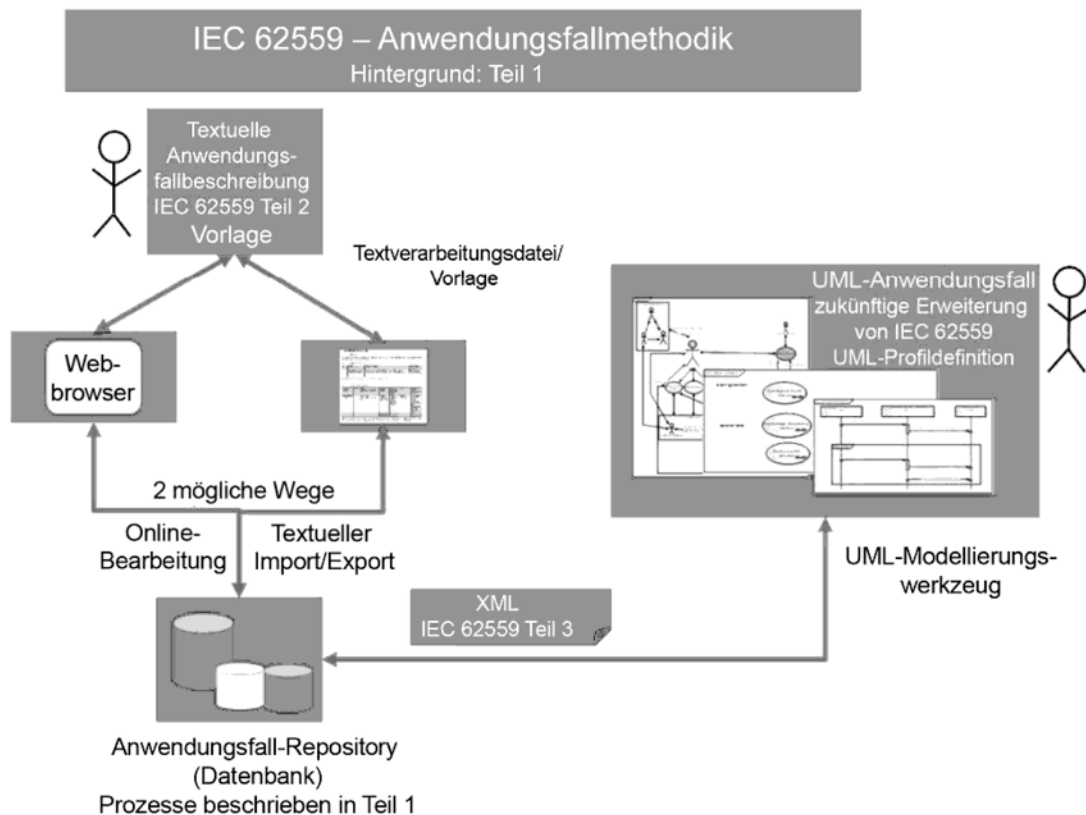


Bild 1 – Normenreihe IEC 62559

IEC 62559-1 – Konzept und Prozesse in der Normung

IEC 62559-1 stellt die Grundlage für ein gemeinsames Anwendungsfall-Repository dar, um Anwendungsfälle innerhalb der IEC auf einer gemeinsamen, kollaborativen Plattform erarbeiten zu können. Mit diesem Repository soll auch eine Harmonisierung von Anwendungsfällen organisiert werden, um weitgehend akzeptierte generische Anwendungsfälle als Basis für die weitere Normungsarbeit bereitzustellen. Dieser Teil beschreibt Prozesse und stellt die Grundlagen für die Anwendungsfallmethodik, wie Begriffe und Anwendungsfalltypen, bereit.

IEC 62559-2 – Definition der Anwendungsfallvorlage, Akteurliste und Anforderungsliste

IEC 62559-2 legt die Struktur einer Anwendungsfallvorlage, einer Akteurliste und einer Liste für Anforderungen fest. Dieses Dokument basiert hauptsächlich auf IEC PAS 62559:2008 und ist zusammen mit IEC 62559-1 anzuwenden.

IEC 62559-3 – Definition von Artefakten der Anwendungsfallvorlage in einem in XML serialisierten Format

Auf der Grundlage von **IEC 62559-2** definiert dieses Dokument die erforderlichen Kernkonzepte und ihre Serialisierung in einem XML-Format für eine Anwendungsfallvorlage, eine Akteurliste und eine Liste für detaillierte Anforderungen. Das XML-Format wird verwendet, um den Inhalt der Vorlage auf andere Entwicklungssysteme übertragen zu können (z. B. UML-Modellierungswerkzeuge). Diese Dokumente sind beispielhaft für Energiesysteme und Smart Grids entwickelt worden, sind aber generisch genug, um auf andere Anwendungsgebiete und Systeme übertragen werden zu können. Auf der Basis dieses Teils soll zukünftig eine UML-Profildefinition erarbeitet werden.

Die Reihe IEC 62559 wird benötigt, um der Entscheidung 7 der SG3 nachzukommen, die vom SMB bei dessen Treffen im Februar 2010 getroffen wurde (SMB/4204/DL, Entscheidung 137/10) und die dringende Bereitstellung eines generischen Repositorys für Anwendungsfälle für alle Smart-Grid-Anwendungen betraf. Nichtsdestotrotz ist die in diesem Dokument beschriebene Anwendungsfallmethodik für eine breitere, über Smart-Grid-Systeme hinausgehende Anwendung innerhalb der Normung vorgesehen.

Bei zunehmend komplexeren Systemen, wie Smart Grids und Smart Cities, stellt sich die Frage, wie Systemanforderungen verwaltet werden können, die von vielen Fachbereichen (die in der Normung durch verschiedene Technische Komitees, TCs, vertreten werden) konkretisiert und genauer definiert werden müssen, um zwischen für die Spezifikation von Normen verantwortlichen TCs ausgetauscht werden zu können, damit diese Systemfunktionen unterstützt werden.

Die Festlegung allgemeiner Methoden und Begriffe ist eine Möglichkeit, dieser Transversalität wirksam zu begegnen. Die Anwendungsfallmethodik stellt den aktuellen Stand der Technik dar und unterstützt weitere Entwicklungsaktivitäten.

Die Anwendungsfallmethodik bietet einen einzigartigen Weg, um Ideen und Anforderungen neuer Anwendungsfälle oder Geschäftsfälle zwischen einer Vielzahl von Experten/TCs mit unterschiedlichen Hintergründen austauschen zu können: beispielsweise zwischen Fachexperten, die sich mit Energiesystemen oder Geschäftsprozessen auskennen, und IT-Experten, welche die ausgetauschten Informationen sowie die Kommunikation definieren. Im Anforderungsentwicklungsprozess bringen Fachexperten allgemeine Ideen und funktionale Anforderungen ein. Das Hauptziel von Systemexperten ist es, diese Anwendungsfälle soweit aufzuschlüsseln, dass sie verwendet werden können, um Schnittstellen, zugehörige Funktionalität, Daten und den Dienstmodellaustausch festzulegen. Aber auch Sicherheits- oder EMV-Experten (als Beispiele) können die beschriebenen Anwendungsfälle, deren Terminologie und ermittelten Anforderungen verwenden.

Zunächst muss jedoch ein Rahmen innerhalb der IEC geschaffen werden, der den IEC-Mitgliedern dabei hilft, Anwendungsfälle in einer konsistenten Art und Weise bereitzustellen – IEC 62559 dient als Grundlage für Anwendungsfall-Repositorys, um Anwendungsfälle sammeln, verwalten, pflegen und bewerten zu können.

Innerhalb der IEC soll ein Anwendungsfall-Repository als eine gemeinsame gemeinschaftliche Plattform für die Erarbeitung von Anwendungsfällen sowie zur Organisation der Harmonisierung von Anwendungsfällen verwendet werden, um weitgehend akzeptierte generische Anwendungsfälle als Basis für die weitere Normungsarbeit bereitzustellen.

Die in diesem Dokument definierte Anwendungsfallvorlage kann nicht nur für die Entwicklung von Normen, sondern auch – wie es die ursprüngliche Absicht der vorherigen IEC/PAS 62559:2008 war – als hilfreiches Mittel für die Realisierung von Projekten in komplexen Systemen dienen. Ebenso können andere Anwendungen, die die Vorteile einer strukturierten Entwicklung von Anforderungen und einer formalisierten Beschreibung der Funktionalität benötigen, Gebrauch von dieser vorgeschlagenen Vorlage machen.

Die Anwendungsfallmethodik muss als Prozess angesehen werden, der mit der Definition von Geschäftsideen, Zielen und Anforderungen beginnt und diese in Anwendungsfallbeschreibungen umsetzt. Diese Informationen können als Grundlage dienen, um Referenzarchitekturen zu ermitteln bzw. zu verknüpfen, welche die Typen der verwendeten Komponenten beschreiben, und für eine fortführende Analyse für den weiteren Normungsprozess verwendet werden.

Es werden Weiterentwicklungen der Anwendungsfallvorlage erwartet. Diese Entwicklungen beziehen sich hauptsächlich auf die Informationen, die in einer Anwendungsfallbeschreibung für die weitere Analyse benötigt werden, und die auf andere Informationen (z. B. eine Referenzarchitektur, IT-Sicherheitsmethoden,

Normen und Datenmodelle) abgebildet werden können. Dies ist teilweise in der vorgeschlagenen Vorlage von **IEC 62559-2** berücksichtigt worden. Weitere Zusammenhänge werden separat beschrieben, da sie sich noch in der Entwicklung befinden, und könnten in der weiteren Entwicklung des IEC-Anwendungsfall-Repositorys Berücksichtigung finden.

1 Anwendungsbereich

Um Anwendungsfälle auf der Grundlage der in IEC 62559-2 definierten Vorlage austauschen zu können, werden in diesem Teil von IEC 62559 die Schnittstellen zwischen den verschiedenen Anwendungsfall-Repositories und/oder UML-Entwicklungssoftwarewerkzeugen beschrieben.

Dafür definiert dieses Dokument die erforderlichen Kernkonzepte einer Anwendungsfallvorlage, einer Akteurliste und einer Liste für detaillierte Anforderungen sowie deren Serialisierung in dem Syntaxformat XML.

Wie in Bild 2 gezeigt, verwendet dieser Modellierungsansatz UML, um die in einem auf der IEC 62559-Vorlage basierten Anwendungsfall enthaltenen Daten graphisch darstellen zu können. Deshalb kann das Textformat der Anwendungsfallvorlage im Anwendungsfallentwicklungsprozess auch einfach als Ausgangspunkt für Geschäftsexperten oder als einfacher Weg zur Änderung von Anwendungsfalldaten für unerfahrene UML-Anwender dienen. Infolgedessen muss die Normenreihe IEC 62559 eine verlässliche Methode zur Umwandlung dieses Textformats in UML und umgekehrt zur Verfügung stellen. Sobald ein Anwendungsfall-Repository auf der Grundlage der Normenreihe IEC 62559 im Einsatz ist, ist es notwendig, dass die anwendungsfall-relevanten Informationen basierend auf einem werkzeugunabhängigen Format zwischen verschiedenen UML-Werkzeugen und verschiedenen Anwendungsfall-Repositories importiert und exportiert werden können.

Hauptziel dieses Dokuments ist es, ein unabhängiges Format für den Austausch der Anwendungsfallinformationen zwischen Modellierungssoftware vorzuschlagen. Um dieses Ziel zu erreichen, wird das Syntaxformat XML gewählt, mithilfe dessen die Anwendungsfalldaten serialisiert werden. Dieses Dokument beschreibt detailliert die Kernkonzepte der Vorlage in UML sowie deren Umwandlungen in XML unter Anwendung des XSD-Standards.

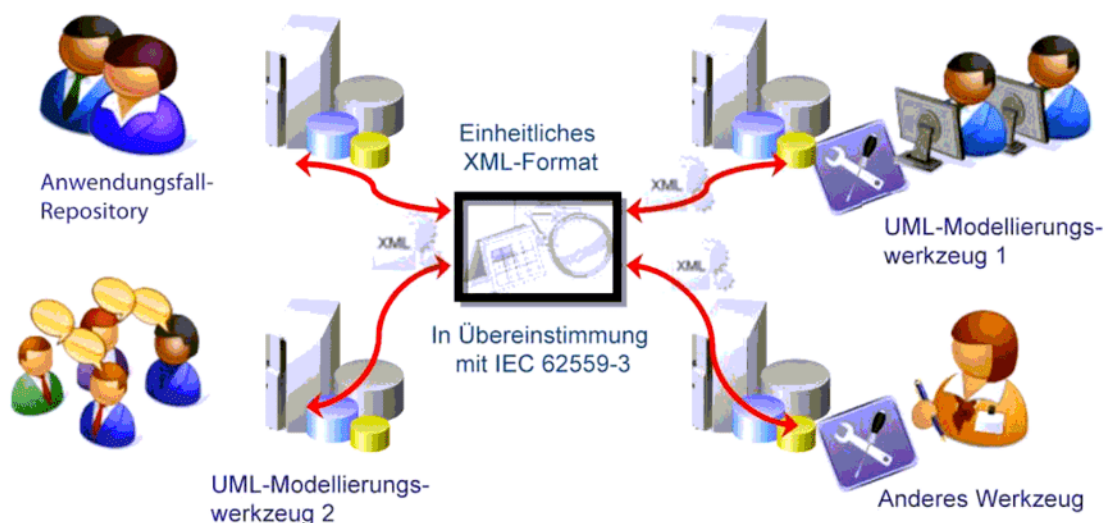


Bild 2 – Einheitliches XML-Format für den Import/Export von Anwendungsfallinformationen zwischen einer Vielzahl von Modellierungssoftware und Repositories

Wie Bild 2 zu entnehmen ist, unterstützt dieser Teil der Normenreihe die Interoperabilität zwischen Modellierungssoftware, die von verschiedenen Unternehmen auf der ganzen Welt eingesetzt wird.

Sobald diese Stufe der Interoperabilität erreicht ist, kann IEC 62559 auch verlässliche Mechanismen zur Interpretation der XML-Daten bereitstellen, damit UML-Anwendungsfälle graphisch dargestellt werden können. Diese Anforderung wird auch von einem anderen Teil von IEC 62559 (noch festzulegen) abgedeckt werden.

Das vorliegende Dokument legt den Schwerpunkt auf den methodischen Rahmen, welcher auch in Normen des IEC/TC 57 verwendet wird und der in Abschnitt 4 zusammengefasst ist.

Um Anwendungsfälle auf der Grundlage der in IEC 62559-2 definierten Vorlage austauschen zu können, werden in diesem Dokument die Schnittstellen zwischen den verschiedenen Anwendungsfall-Repositorys und/oder UML-Werkzeugen beschrieben.

2 Normative Verweisungen

Die folgenden Dokumente werden im Text in solcher Weise in Bezug genommen, dass einige Teile davon oder ihr gesamter Inhalt Anforderungen des vorliegenden Dokuments darstellen. Bei datierten Verweisungen gilt nur die in Bezug genommene Ausgabe. Bei undatierten Verweisungen gilt die letzte Ausgabe des in Bezug genommenen Dokuments (einschließlich aller Änderungen).

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

3 Begriffe

Für die Anwendung dieses Dokuments gelten die Begriffe nach IEC 62559-2 und die folgenden Begriffe.

ISO und IEC stellen terminologische Datenbanken für die Verwendung in der Normung unter den folgenden Adressen bereit:

- IEC Electropedia: verfügbar unter <http://www.electropedia.org/>;
- ISO Online Browsing Platform: verfügbar unter .

3.1

XML

erweiterbare Auszeichnungssprache

Auszeichnungssprache, die einen Satz von Regeln für die Codierung von Dokumenten in einem Format definiert, das sowohl von Menschen als auch von Maschinen gelesen werden kann

Anmerkung 1 zum Begriff: Die XML-Spezifikation wird vom World Wide Web Consortium (W3C) erarbeitet.

3.2

XSD

Definitionssprache des XML-Schemas

Sprache, die verwendet wird, um die Struktur, den Inhalt und die Semantik von XML-Dateien zu definieren

Anmerkung 1 zum Begriff: XSD ist eine Empfehlung des World Wide Web Consortium (W3C).

4 Methodischer Rahmen für die Erarbeitung dieser Norm

4.1 Modellbasierter Ansatz für die Entwicklung des Datenaustauschformats

Dieses Dokument baut auf einem modellbasierten Ansatz für die Definition von Austauschformaten auf und legt deshalb entsprechende UML-Modelle fest. Der verwendete Ansatz wurde in IEC TC 57 entwickelt, um spezifische Datenmodelle (Profile) zu entwickeln, und wird aktiv für die Erarbeitung der CIM-Normen des IEC TC 57 für die Kommunikation von Stromversorgungssystemen eingesetzt. Innerhalb des Ansatzes werden mehrere Modelltypen, deren Zusammenhänge sowie Regeln zur Ableitung spezifischer Modelle von einem generischen Modell beschrieben. Da der Ansatz in IEC 62325-450 ausführlich erklärt wird, wird er in 4.1 nur kurz und hinsichtlich seiner Anpassungen zur Schaffung des Rahmens, auf dem diese Norm beruht, beschrieben.

Unter Verwendung des Ansatzes können mehrere XML-basierte Austauschformate für einen bestimmten Bereich definiert werden. Alle Austauschformate, die anhand dieses Prozesses entwickelt werden, basieren jedoch auf einem einzigen Quellmodell, das die semantische Kompatibilität sicherstellt. Zwei zusammenhängende Modelle, die in diesem Prozess eine Schlüsselrolle spielen, sind das Informationsmodell und das

Kontextmodell, die für verschiedene Belange eingesetzt werden. In Bild 3 sind diese Modelle im Kontext des Prozesses der Definition des Datenaustauschformats dargestellt.

Das Informationsmodell bildet alle in IEC 62559-2 erarbeiteten allgemeinen Kernkonzepte eines Anwendungsfalls auf einer eher generischen Ebene ab, einschließlich Entitäten, deren Attribute und Beziehungen. Dies umfasst zum Beispiel einen Anwendungsfall selbst, Szenarien eines Anwendungsfalls, beteiligte Akteure, oder Informationsobjekte, die innerhalb eines Anwendungsfalls ausgetauscht werden. Die im Informationsmodell definierten Konzepte legen die Semantik des Datenaustausches fest. In Bezug auf die Strukturierung von Anwendungsfallinformationen kann das Informationsmodell demzufolge als streng normativ angesehen werden.

Ein Kontextmodell basiert immer auf dem Informationsmodell. Kontextmodelle werden entwickelt, um spezifische Anforderungen an den Informationsaustausch in Bezug auf das Informationsmodell abzubilden. Kontextmodelle enthalten eine kontextabhängige Teilmenge des Informationsmodells (oder aber das vollständige Informationsmodell), die (das) weiter eingeschränkt oder spezialisiert werden kann.

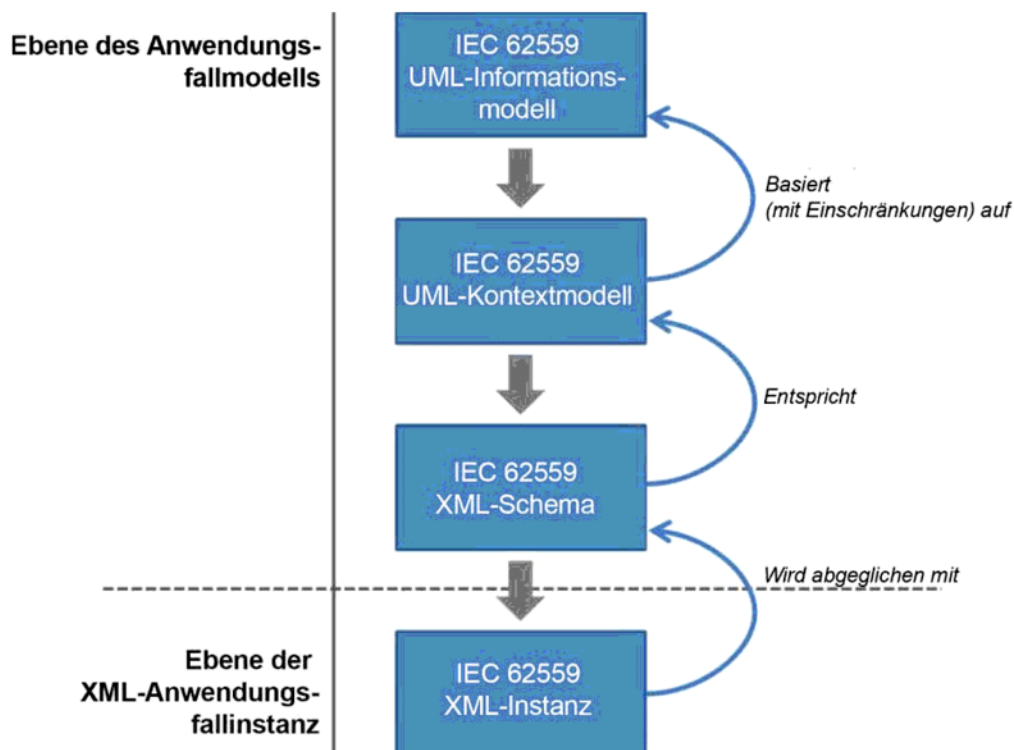


Bild 3 – Modellbasierter Entwicklungsrahmen für die Definition des XML-basierten Datenaustauschformats nach IEC 62559-3

Schlussendlich können auf ein Kontextmodell weitere Regeln angewendet werden, wodurch das endgültige Austauschformat festgelegt wird, das folglich dem Kontextmodell entspricht. Im Rahmen dieses Dokuments handelt es sich hierbei um ein XML-Schema, das als Grundlage für einen Abgleich von ausgetauschten Anwendungsfalldaten dient. Zu diesem Zweck müssen Anwendungsfallinstanzdaten in einem XML-Dokument definiert werden.

Der oben kurz beschriebene modellbasierte Prozess wurde angewendet, um entsprechende XML-basierte Datenaustauschformate zu entwickeln. Um letztlich eine genormte XML-Schemadefinitionsdatei (XSD-Datei) in Übereinstimmung mit IEC 62559-2 zu erhalten, wird der modellbasierte Ansatz wie im Folgenden ausführlich beschrieben angewendet.

a) UML-Informationsmodell nach IEC 62559:

Dieses UML-Modell vereint die in IEC 62559-2 beschriebenen Kernkonzepte. Es ist mithilfe von UML-Klassendiagrammen, einschließlich Klassen, Attributen, Datentypen und Assoziationen, modelliert. Dieses Informationsmodell enthält die Kernkonzepte eines Anwendungsfalls, der so allgemein modelliert ist, dass die enthaltenen Konzepte in anderen Implementationen wiederverwendet werden können.

Dieses Informationsmodell könnte beispielsweise auch zur Festlegung der Struktur einer Datenbank für Anwendungsfälle verwendet werden. Um das Informationsmodell flexibel zu gestalten, wurden die Konzepte mit optionalen Aspekten modelliert. Viele Attribute und Assoziationen wurden z. B. mit flexiblen Kardinalitäten, wie „0..1“ für Attribute und „0..*“ für Beziehungen, modelliert. Um dieses generische Informationsmodell implementieren zu können, muss genauer festgelegt werden, wie die Kernkonzepte für den bestimmten Zweck des Austausches von Anwendungsfallinformationen zu verwenden sind.

b) UML-Kontextmodell nach IEC 62559:

Dieses zweite UML-Modell basiert auf dem UML-Informationsmodell, umfasst aber Einschränkungen, die dazu dienen, es auf einen bestimmten Zweck der Definition eines Modells für den Austausch von Anwendungsfallinformationen anzupassen. Durch diese Einschränkungen, die für die Konzepte des Informationsmodells festgelegt werden, wird das Kontextmodell für den Austausch von Anwendungsfallinformationen spezialisiert. Folgende Einschränkungen sind zulässig:

- 1) Einschränkungen für Kardinalitäten von Attributen und Assoziationen durch Eingrenzung dieser. Beispiel: Die Kardinalität eines Attributs „0..1“ kann auf „1..1“ (obligatorisches Attribut) oder „0..1“ (optionales Attribut) oder „0..0“ (nicht verwendetes Attribut) festgelegt werden. Die Kardinalität eines Assoziationsendes „0..*“ kann auf ein eingeschlossenes Intervall festgelegt werden, z. B. „2..6“ (Folge von mindestens 2 und maximal 6 instanziierten Konzepten).
- 2) Einschränkungen des Attributwertaums durch Hinzufügen von Wertebereichen. Beispiel: Ein Attribut, das eine Zeichenfolge ist, kann einen Wertebereich der Länge 6 aufweisen, d. h. der erwartete Attributwert ist eine Folge von 6 Symbolen.

c) XML-Schema nach IEC 62559:

Diese XML-Schemadefinition (XSD) wird automatisch aus dem zuvor entwickelten UML-Kontextmodell generiert. Jeder XML-basierte Austausch von Anwendungsfallinformationen muss mit dieser XSD abgeglichen werden. Da für jeden Austausch von Anwendungsfallinformationen die in der XSD festgelegte Struktur berücksichtigt werden muss, wird die Interoperabilität zwischen Modellierungssoftware sichergestellt.

4.2 Beispiel für den modellbasierten Prozess im Kontext von IEC 62559

Unterabschnitt 4.2 beschreibt beispielhaft die Beziehung zwischen einem Anwendungsfallszenario und einem Akteur, der in diesem Szenario die Rolle „PrimaryActor“ (de: primärer Akteur) spielt.

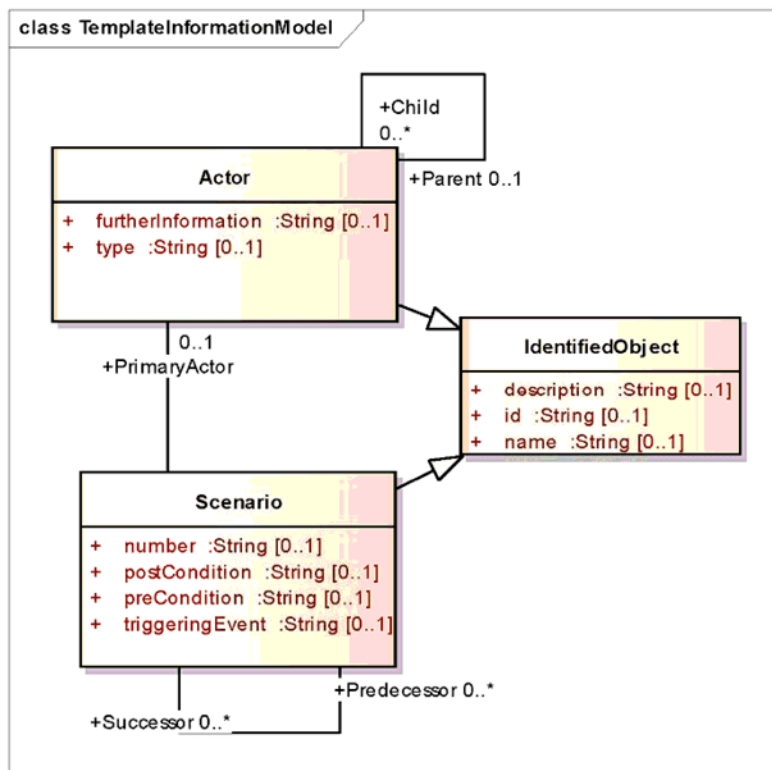


Bild 4 – UML-Informationsmodell für das Beispiel der Beziehung zwischen „Scenario“ und „Actor“

Bild 4 zeigt das UML-Informationsmodell mit den zwei Kernkonzepten „Scenario“ und „Actor“ (unter Verwendung von UML-Klassen), die über eine UML-Assoziation miteinander verbunden sind, welche die Rolle „PrimaryActor“ und die Kardinalität „0..1“ auf „Actor“-Seite aufweist. Das bedeutet, dass einem Szenario ein optionaler primärer Akteur zugewiesen werden darf. Zusatzinformationen für „Scenario“ und „Actor“ werden mithilfe von Attributen dieser beiden Klassen modelliert. Ein Szenario weist beispielsweise eine Nummer („number“), eine Nachbedingung („postCondition“), eine Vorbedingung („preCondition“) und ein Auslöseereignis („triggeringEvent“) auf. Da beide Konzepte gleiche Anforderungen an die Darstellung einer Identifikation haben, enthält eine Generalisierungsklasse „IdentifiedObject“ diese gemeinsamen Attribute, nämlich die Kennung („id“), den Namen („name“) und die Beschreibung („description“).

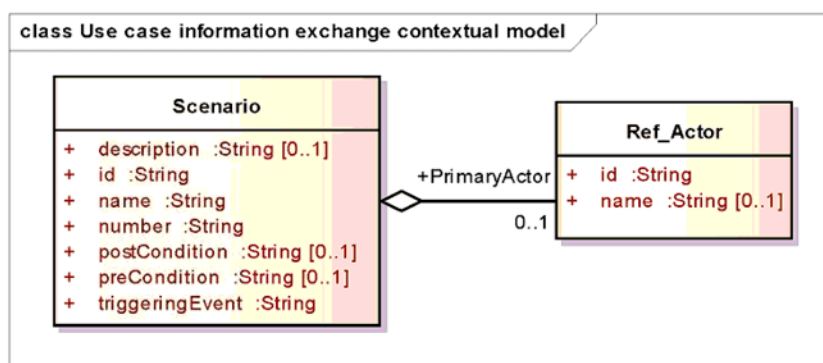
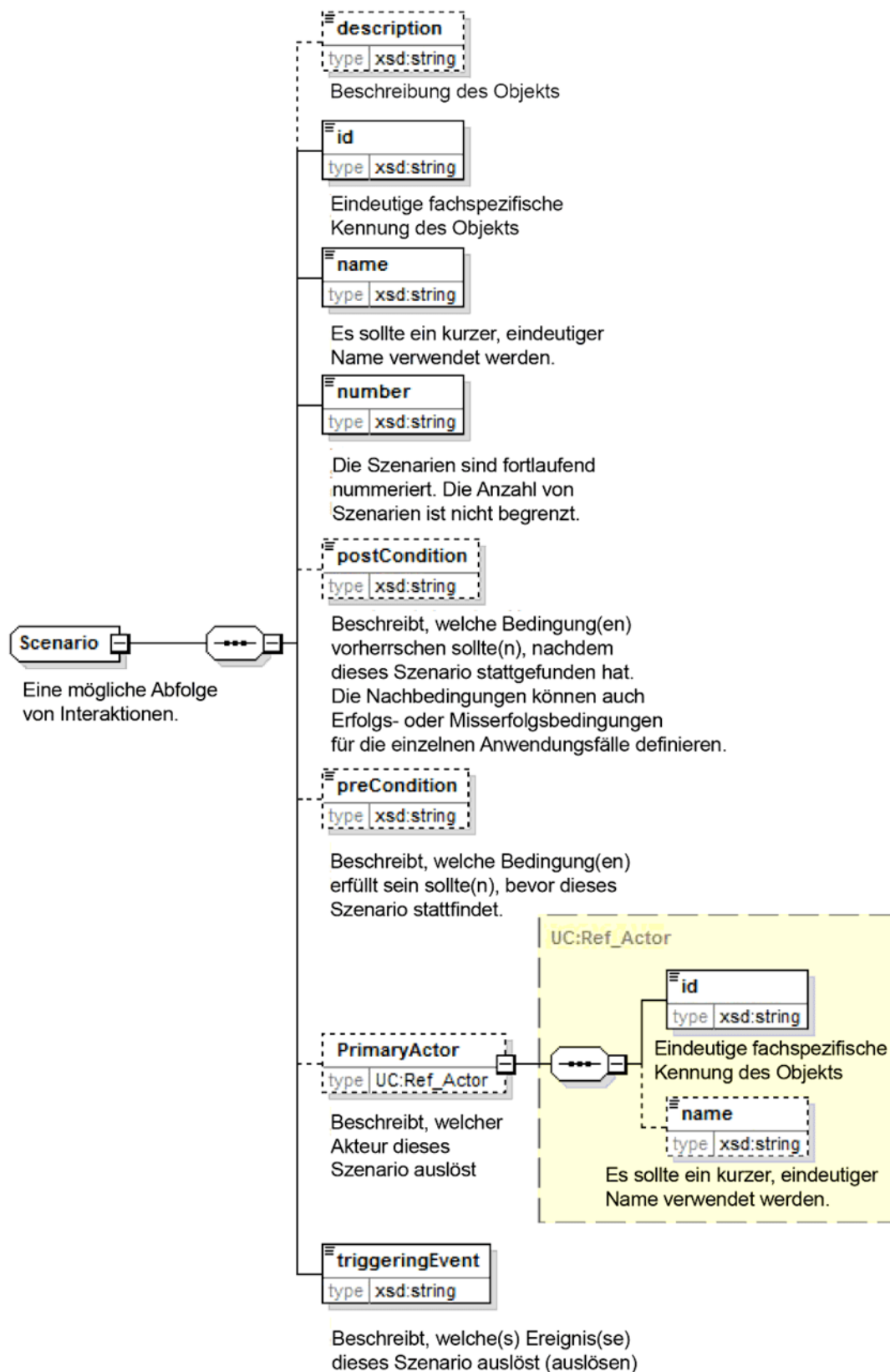


Bild 5 – UML-Kontextmodell basierend auf dem Beispiel der Beziehung zwischen „Scenario“ und „Actor“

Im UML-Kontextmodell (siehe Bild 5) werden die beiden Konzepte „Scenario“ und „Actor“ wiederverwendet und angepasst, indem Einschränkungen für das Informationsmodell im Hinblick auf den spezifischen Zweck des Informationsaustausches über ein „Scenario“ mit einem „PrimaryActor“ hinzugefügt werden. In diesem Beispiel werden einige Attribute als obligatorisch erklärt. Für „Scenario“ sind tatsächlich die Kennung („id“), der Name („name“), die Nummer („number“) und das Auslöseereignis („triggeringEvent“) obligatorische In-

formationen für den Austausch. Darüber hinaus wird auch die Kennung zum obligatorischen Attribut, wenn Informationen zu einem „PrimaryActor“ angegeben sind. Das Konzept „Actor“ wird in demselben Kontextmodell mehrere Male mit verschiedenen definierten Strukturen wiederverwendet. Um eine eindeutige Definition von Kontext-namen für diese verschiedenen Strukturen zu erhalten, wird dem Klassennamen des Informationsmodells ein Bezeichner hinzugefügt. In diesem Beispiel wird die Kontextklasse, die auf dem Konzept „Actor“ basiert, als „Ref_Actor“ bezeichnet. Bei der Modellierung des Kontextmodells erhält die Assoziation zwischen „Scenario“ und „Actor“ Einschränkungen und wird zur Aggregation, d. h., dass die Informationen zu „Scenario“ beim Datenaustausch Informationen zu einem „PrimaryActor“ einschließen.



Generated by XMLSpy

www.altova.com

Bild 6 – XML-Schema für das Beispiel der Beziehung zwischen „Scenario“ und „Actor“

[Bild 6](#) stellt den XSD-Code graphisch dar, der automatisch aus dem in [Bild 5](#) gezeigten UML-Kontextmodell generiert wird.

4.3 Entwicklung spezifischer Datenaustauschprofile

Durch die Anwendung des oben beschriebenen modellbasierten Prozesses können mehrere Kontextmodelle und entsprechend mehrere verschiedene Austauschformate entstehen, die allerdings üblicherweise die gleiche Semantik aufweisen. Dieser Prozess des Ableitens eines spezifischen Modells mit einer Teilmenge von Modellelementen von einem generischen Modell und des Spezialisierens von Modellelementen wird als „Profilierung“ bezeichnet; das Ergebnis des Profilierungsprozesses ist ein Profil. Da mehrere Profile auf der Basis eines einzigen Informationsmodells erstellt werden können, können nicht alle Profile als normativ angesehen werden. Wenn jedoch Daten zwischen zwei Anwendungen erfolgreich ausgetauscht werden sollen, muss eine (normative) Vereinbarung über ein bestimmtes Austauschformat, nicht nur hinsichtlich der Semantik, sondern auch hinsichtlich der Syntax getroffen werden.

Deshalb beschreibt diese Norm Schlüsselszenarien des Informationsaustausches, die zu entsprechenden normativen Profilen führen, die in Software implementiert werden können. Zu den aktuell ermittelten Schlüsselszenarien des Informationsaustausches gehört der Austausch aller Konzepte des Informationsmodells (z. B. für den Austausch von Informationen zwischen Anwendungsfall-Repositorys), der Austausch einer Gruppe von Akteuren (Rollenmodell) und der Austausch von Anforderungen (siehe Abschnitt 5). Während das erste Datenaustauschszenario eher allgemeiner Natur ist, lassen die letzteren einen detaillierteren Abgleich von Daten im Vergleich zu dem Austauschmodell, das ein gesamtes Repository abdeckt, zu, da bestimmte Einschränkungen, wie das Vorhandensein eines Akteurs bzw. einer Anforderung, angenommen werden können.

Das normative Datenaustauschformat des ersten generischen Austauschschemas ist in [Abschnitt 6](#) enthalten. Im Rahmen zukünftiger Erweiterungen von IEC 62559-3 werden möglicherweise weitere genormte Informationsaustauschszenarien mit entsprechenden Profilen, z. B. für Anwendungsfälle, hinzugefügt. Im Falle spezifischer Anforderungen können mithilfe des modellbasierten Prozesses auch nicht normative Austauschformate/Profile entwickelt werden.

5 Anwendung des methodischen Rahmens für die Definition einer Reihe von wichtigen syntaktischen Austauschformaten (Anwendungsfälle, Akteure, Anforderungen)

Im Hinblick auf die verschiedenen Arten des Datenaustausches basierend auf bestimmten Schlüsselszenarien des Datenaustausches zielt IEC 62559-3 darauf ab, eine Reihe von wichtigen Datenaustauschformaten im Besonderen für Anwendungsfälle, Akteure und Anforderungen bereitzustellen. Um diese syntaktischen XML-Austauschformate zu erhalten, wird der in [Abschnitt 4](#) beschriebene Modellierungsprozess für die Entwicklung der einzelnen Austauschformate angewendet. [Bild 7](#) gibt einen Überblick über die Profilierung des generischen Informationsmodells.

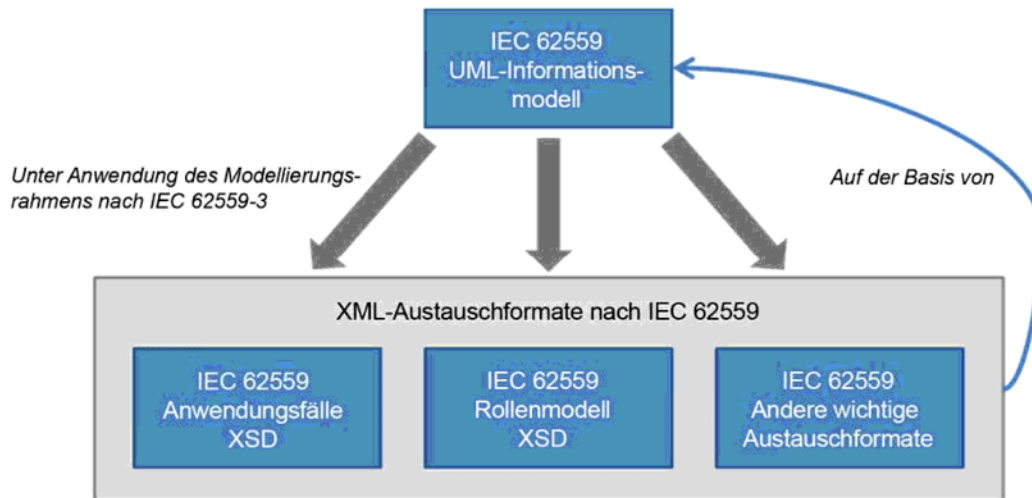


Bild 7 – Entwicklung wichtiger XML-Austauschformate auf der Basis eines einzelnen UML-Informationsmodells nach IEC 62559

6 Detaillierte Inhalte der Modelle (automatische Erzeugung aus UML-Modellen)

6.1 Kernkonzepte der UML-Anwendungsfälle (UML-Informationsmodell)

6.1.1 Allgemeines

Die Kernkonzepte der Anwendungsfälle werden durch ein UML-Informationsmodell beschrieben, das zwischen den verschiedenen festgelegten Austauschformaten in diesem Dokument geteilt wird. Die Kernkonzepte werden in einen Satz UML-Pakete eingeteilt.

6.1.2 Modellpaket „UseCaseCoreConcepts“

6.1.2.1 Allgemeines

UML-Informationsmodell nach IEC 62559 für die Kernkonzepte der Anwendungsfälle.

Bild 8 zeigt das Klassendiagramm „UseCaseCoreConcepts_Libraries“.

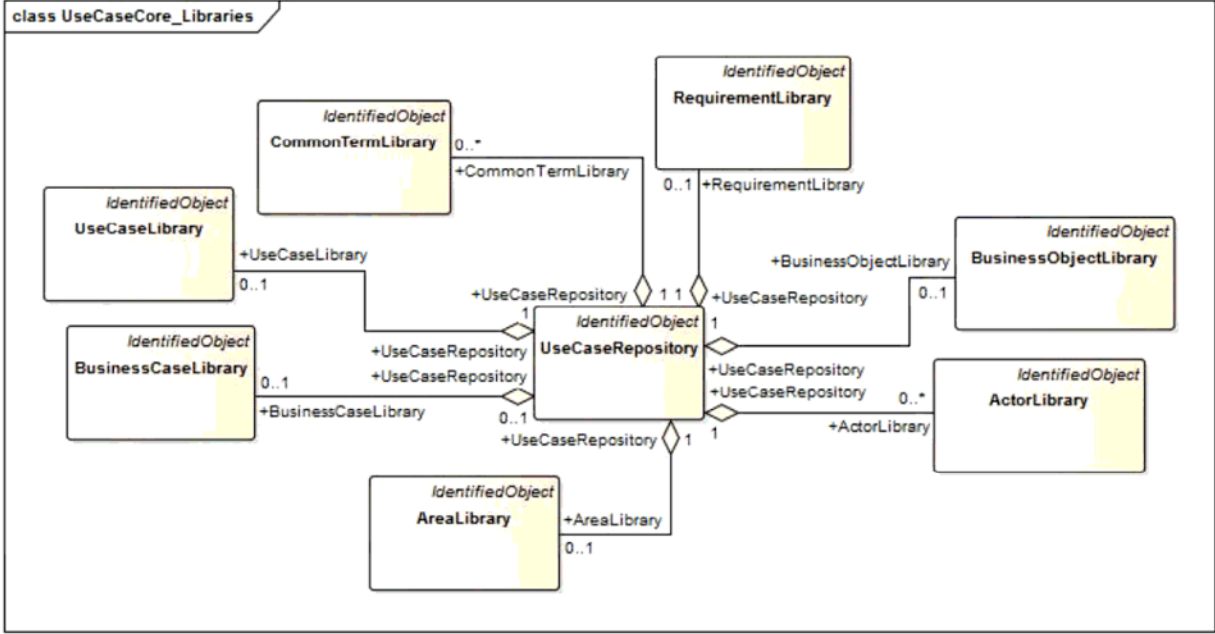


Bild 8 – Klassendiagramm „UseCaseCoreConcepts::UseCaseCore_Libraries“

Bild 9 zeigt das Klassendiagramm „UseCaseCore_ActorLibrary“.

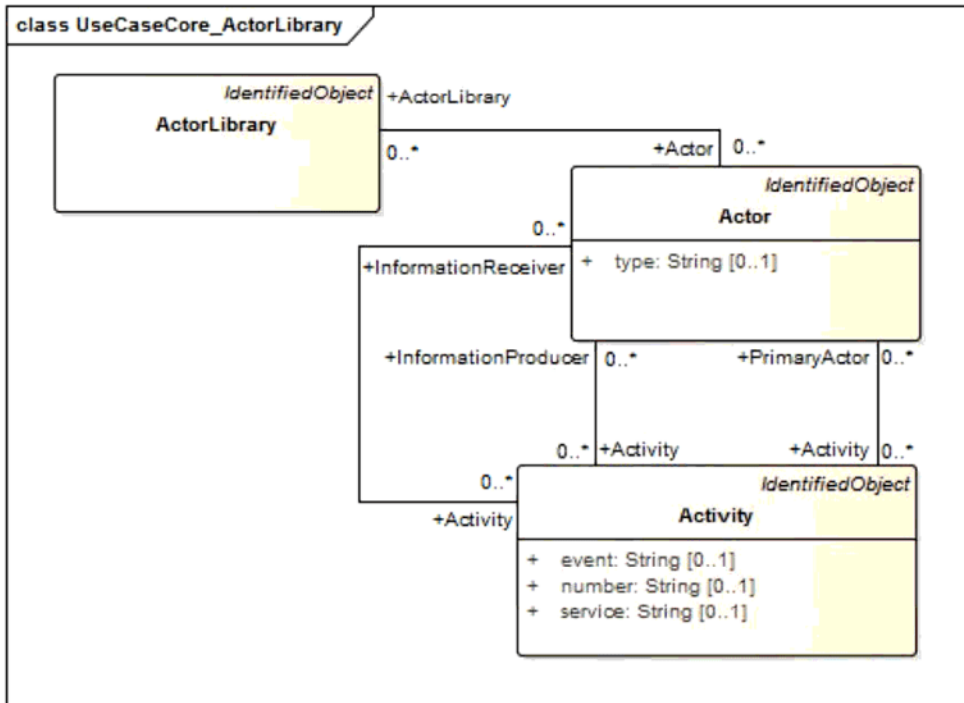


Bild 9 – Klassendiagramm „seCaseCoreConcepts::UseCaseCore_ActorLibrarys“

Bild 10 zeigt das Klassendiagramm „UseCaseCore_UseCaseLibrary“.

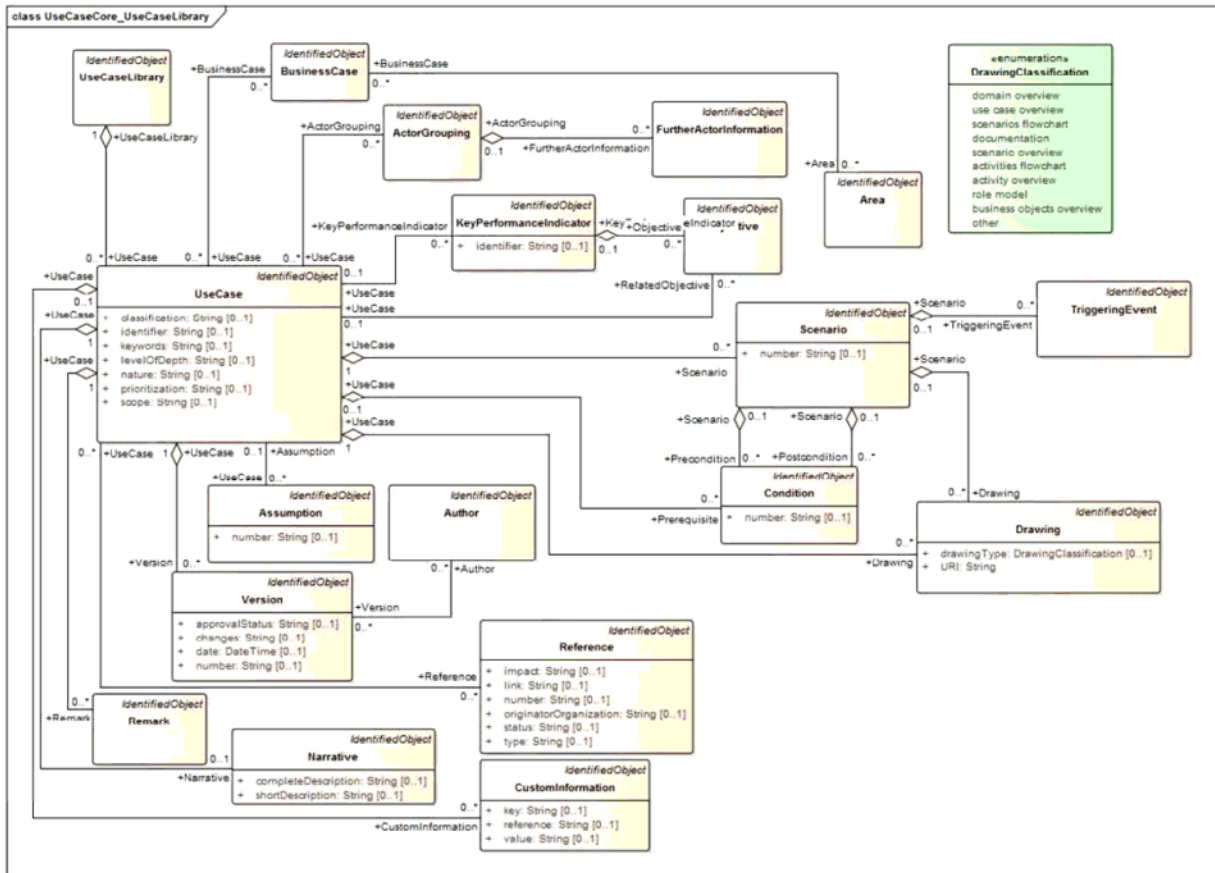


Bild 10 – Klassendiagramm „UseCaseCoreConcepts::UseCaseCore_UseCaseLibrary“

Bild 11 zeigt das Klassendiagramm „UseCaseCore_DetailedActivities“.

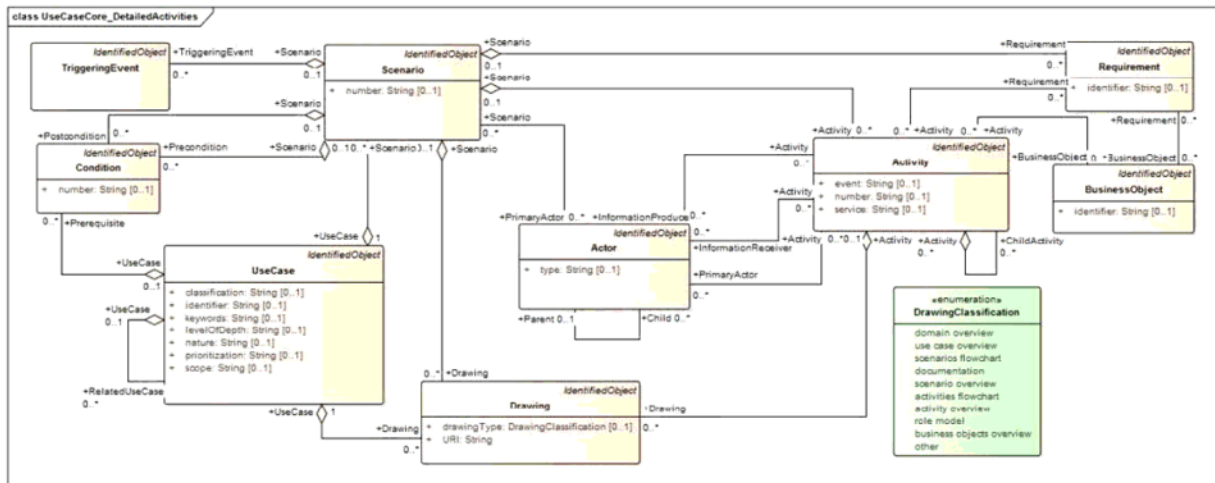


Bild 11 – Klassendiagramm „UseCaseCoreConcepts::UseCaseCore_DetailedActivities“

Bild 12 zeigt das Klassendiagramm „UseCaseCore_AreaLibrary“.

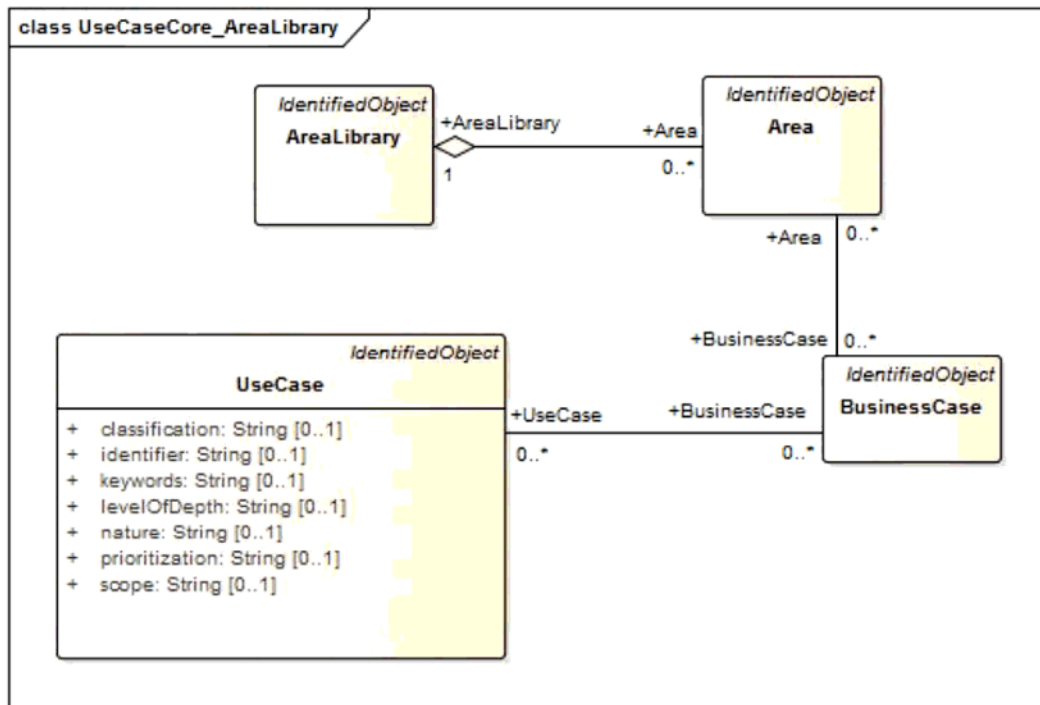


Bild 12 – Klassendiagramm „UseCaseCoreConcepts::UseCaseCore_AreaLibrary“

Bild 13 zeigt das Klassendiagramm „UseCaseCore_BusinessCaseLibrary“.

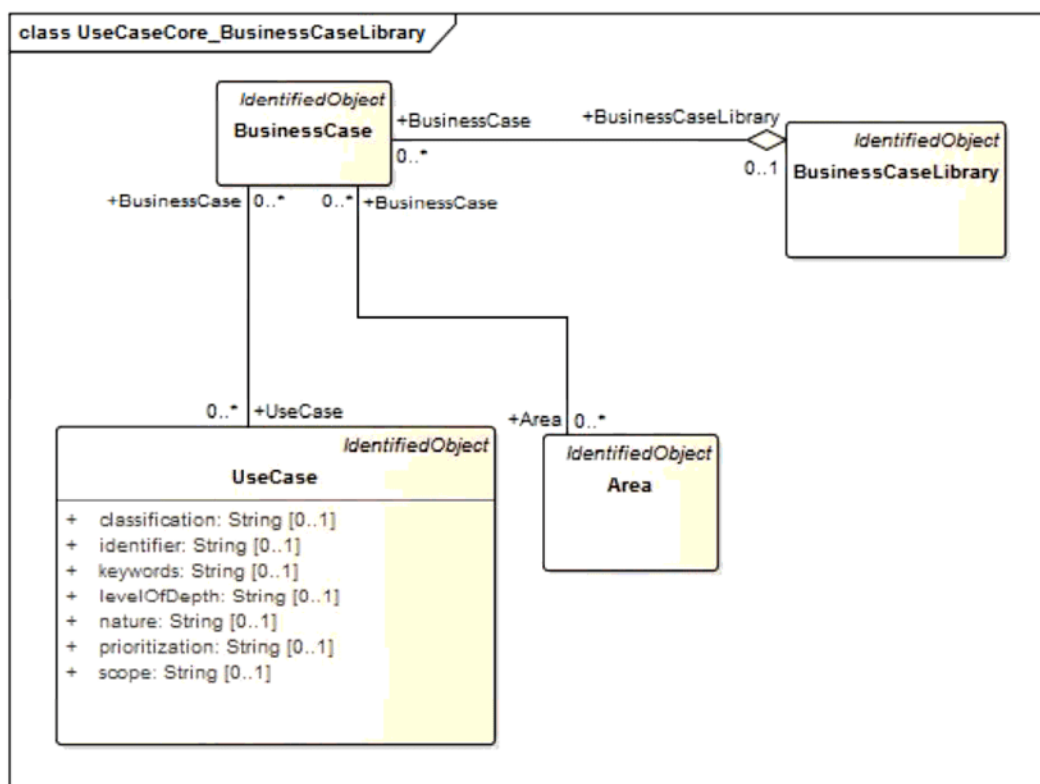


Bild 13 – Klassendiagramm „UseCaseCoreConcepts::UseCaseCore_BusinessCaseLibrary“

Bild 14 zeigt das Klassendiagramm „UseCaseCore_BusinessObjectLibrary“.

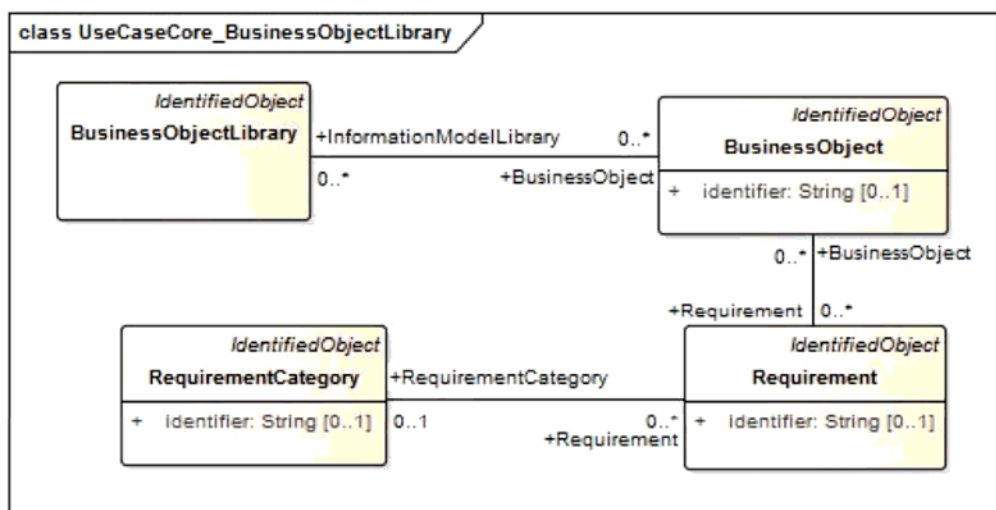


Bild 14 – Klassendiagramm „UseCaseCoreConcepts::UseCaseCore_BusinessObjectLibrary“

Bild 15 zeigt das Klassendiagramm „UseCaseCore_CommonTermLibrary“.

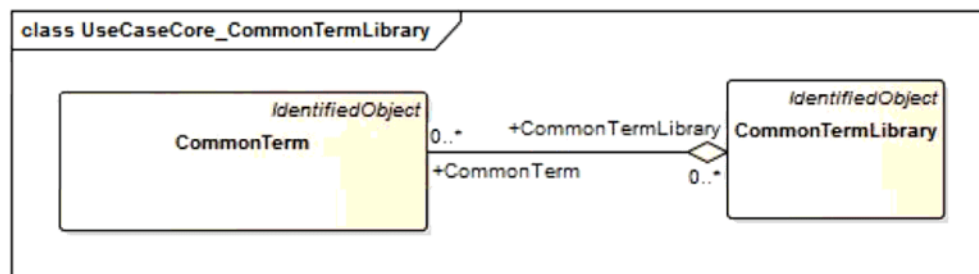


Bild 15 – Klassendiagramm „UseCaseCoreConcepts::UseCaseCore_CommonTermLibrary“

Bild 16 zeigt das Klassendiagramm „UseCaseCore_RequirementLibrary“.

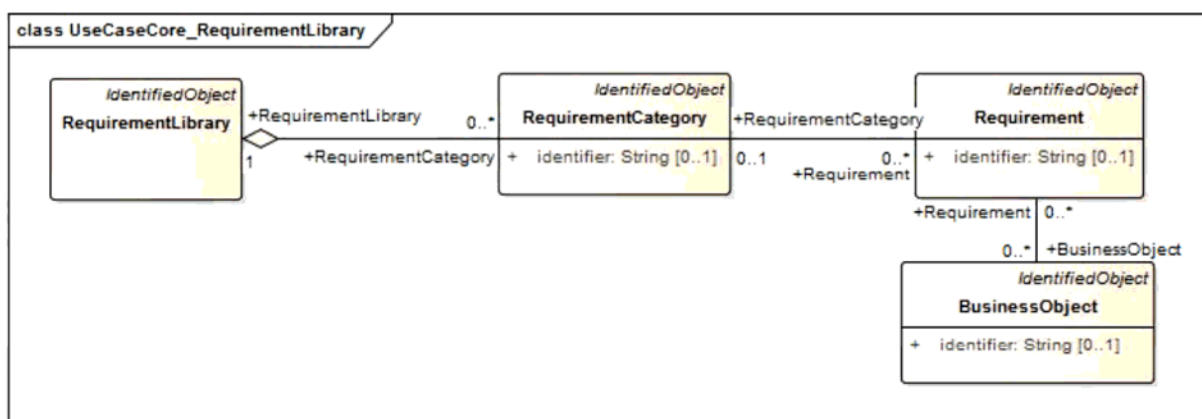


Bild 16 – Klassendiagramm „UseCaseCoreConcepts::UseCaseCore_RequirementLibrary“

6.1.2.2 Activity

Kann in weitere Aktivitäten unterteilt sein oder einen elementaren Schritt eines Szenarios darstellen.

Tabelle 1 zeigt alle Attribute von „Activity“.

Tabelle 1 – Attribute von „UseCaseCoreConcepts::Activity“

Name	Typ	Beschreibung
event	String	Ist das Ereignis, das den Schritt auslöst. Dies kann der Abschluss vorhergehender Schritte (in diesem Fall entspricht das Ereignis der durch Kommata getrennten Liste ihrer Nummern) oder die Bedingung des Austritts aus einer Schleife sein.
number	String	Index der Aktivität
service	String	<p>Diese Spalte legt die Natur des Informationsflusses und den Urheber der Informationen dar. Nach IEC 61968-100:2013 verfügbare Optionen sind: CREATE, GET, CHANGE, DELETE, CANCEL, EXECUTE.</p> <p>Zusätzlich werden vorgeschlagen: REPORT, TIMER, REPEAT.</p> <ul style="list-style-type: none"> – CREATE bedeutet, dass ein Informationsobjekt am Erzeuger erstellt werden muss. – GET (dies ist der Standardwert, wenn nichts angegeben wurde) bedeutet, dass der Empfänger Informationen vom Erzeuger anfordert (Standard). – CHANGE bedeutet, dass Informationen aktualisiert werden müssen. Der Erzeuger aktualisiert die Informationen des Empfängers. – DELETE bedeutet, dass Informationen gelöscht werden müssen. Der Erzeuger löscht die Informationen vom Empfänger. – CANCEL, CLOSE beinhalten prozessbezogenen Aktionen, wie den Abschluss eines Arbeitsauftrags oder den Abbruch eines Steuerungsauftrags. – EXECUTE wird verwendet, wenn eine komplexe Transaktion mithilfe eines Services übermittelt wird, der potentiell mehr als ein Verb enthält. – REPORT wird für die Repräsentation des Transfers von ungebetenen Informationen oder asynchronen Informationsflüssen verwendet. Der Erzeuger liefert Informationen an den Empfänger. – TIMER wird für die Anzeige einer Warteperiode verwendet. Bei Verwendung des Service TIMER müssen sich der Erzeuger der Informationen und der Empfänger der Informationen auf den gleichen Akteur beziehen. – REPEAT zeigt an, dass eine Reihe von Schritten wiederholt wird, bis ein bestimmter Zustand oder ein Triggerereignis eintritt. Der Zustand für diese Reihe oder diesen Schritt wird in der Spalte „Event“ in Textform erläutert. Nach dem Wort REPEAT, müssen die Nummern des ersten und des letzten Schritts der zu wiederholenden Reihe auf folgende Weise in Klammern angegeben werden: REPEAT(X-Y), wobei X der erste Schritt und Y der letzte Schritt ist. <p>Diese üblichen Servicedefinitionen beziehen sich auf Automatisierungs-/Informations- oder Kommunikationssysteme. Falls die Anwendungsfall-Vorlage in anderen Domänen angewendet wird, können weitere Services verwendet und beschrieben werden.</p>
Beschreibung	String	Geerbt von: IdentifiedObject
mRID	String	Geerbt von: IdentifiedObject
name	String	Geerbt von: IdentifiedObject

In [Tabelle 2](#) sind alle Assoziationsenden von „Activity“ mit anderen Klassen aufgeführt.

Tabelle 2 – Assoziationsenden von „UseCaseCoreConcepts::Activity“ mit anderen Klassen

[Mult von]	[Mult bis] Name	Typ	Beschreibung
[0..*]	[0..*] Anforderung	Anforderung	Es können auch mehrere Anforderungen in einem Schritt aufgeführt werden, indem sie durch Kommata getrennt werden.
[0..*]	[0..*] InformationReceiver	Actor	Dies gibt den Empfänger der Informationen an. Dies sollte einer der im Repository festgelegten Akteure sein.
[0..*]	[0..*] InformationProducer	Actor	Dies gibt den Erzeuger oder die Quelle der Informationen an. Dies sollte einer der im Repository festgelegten Akteure sein.
[0..*]	[0..*] BusinessObject	BusinessObject	Dies beschreibt kurz die zwischen den Akteuren – Erzeuger der Informationen und Empfänger der Informationen – auszutauschenden Informationen. Es können auch mehrere Informationselemente in einem Schritt aufgeführt werden, indem sie durch Kommata getrennt werden.
[0..1]	[0..*] Drawing	Drawing	die Liste zugehöriger Zeichnungen
[0..*]	[0..*] PrimaryActor	Actor	Referenz auf den PrimaryActor
[0..*]	[0..*] ChildActivity	Activity	Unteraktivitäten
[0..*]	[0..1] Scenario	Scenario	Referenz auf das Szenario
[0..*]	[0..*] Activity	Activity	Hauptaktivitäten

6.1.2.3 Actor

Entität, die kommuniziert und interagiert.

Akteure können Personen, Softwareanwendungen, Systeme, Datenbanken und sogar Stromversorgungssysteme selbst sein.

In Tabelle 3 sind alle Attribute von „Actor“ aufgeführt.

Tabelle 3 – Attribute von „UseCaseCoreConcepts::Actor“

name	Typ	Beschreibung
type	String	Art des Akteurs (Unternehmen, System, ...)
Beschreibung	String	Geerbt von: IdentifiedObject
mRID	String	Geerbt von: IdentifiedObject
name	String	Geerbt von: IdentifiedObject

In [Tabelle 4](#) sind alle Assoziationsenden von „Actor“ mit anderen Klassen aufgeführt.

Tabelle 4 – Assoziationsenden von „UseCaseCoreConcepts::Actor“ mit anderen Klassen

[Mult von]	[Mult bis] Name	Typ	Beschreibung
[0..1]	[0..*] Child	Actor	Liste spezialisierter Akteure
[0..*]	[0..*] Activity	Activity	die zum Akteur gehörenden Aktivitäten
[0..*]	[0..*] Activity	Activity	die zu diesem Akteur gehörenden Aktivitäten
[0..*]	[0..*] Activity	Activity	Ein Akteur kann mit einer Aktivität verbunden sein.
[0..*]	[0..*] Scenario	Scenario	Der Akteur ist mit Szenarien verbunden.
[0..*]	[0..*] ActorLibrary	ActorLibrary	Auf den Akteur wird durch Bibliotheken Bezug genommen.
[0..*]	[0..*] UseCase	UseCase	Liste zu einem Akteur gehörender Anwendungsfälle
[0..*]	[0..*] FurtherActorInformation	FurtherActorInformation	Ein Akteur kann über eine Liste mit weiteren Informationen verfügen, die zwischen verschiedenen Gruppierungen für unterschiedliche Anwendungsfälle geteilt wird.
[0..*]	[0..1] Parent	Actor	der generalisierte Akteur

6.1.2.4 ActorGrouping

Gruppe von Akteuren zur Organisation einer Akteurliste

In Tabelle 5 sind alle Attribute von „ActorGrouping“ aufgeführt.

Tabelle 5 – Attribute von „UseCaseCoreConcepts::ActorGrouping“

name	Typ	Beschreibung
Beschreibung	String	Geerbt von: IdentifiedObject
mRID	String	Geerbt von: IdentifiedObject
name	String	Geerbt von: IdentifiedObject

In Tabelle 6 – sind alle Assoziationsenden von „ActorGrouping“ mit anderen Klassen aufgeführt.

Tabelle 6 – Assoziationsenden von „UseCaseCoreConcepts::ActorGrouping“ mit anderen Klassen

[Mult von]	[Mult bis] Name	Typ	Beschreibung
[0..*]	[0..*] UseCase	UseCase	ActorGrouping gehört zu verschiedenen Anwendungsfällen.
[0..1]	[0..*] FurtherActorInformation	FurtherActorInformation	Im Kontext eines Anwendungsfalles, kann ein in ein ActorGrouping eingruppierter Akteur über eine FurtherActorInformation verfügen.

6.1.2.5 ActorLibrary

Akteure des Repositorys

In Tabelle 7 sind alle Attribute von „ActorLibrary“ aufgeführt.

Tabelle 7 – Attribute von „UseCaseCoreConcepts::ActorLibrary“

Name	Typ	Beschreibung
Beschreibung	String	Geerbt von: IdentifiedObject
mRID	String	Geerbt von: IdentifiedObject
name	String	Geerbt von: IdentifiedObject

In Tabelle 8 – sind alle Assoziationsenden von „ActorLibrary“ mit anderen Klassen aufgeführt.

Tabelle 8 – Assoziationsenden von „UseCaseCoreConcepts::ActorLibrary“ mit anderen Klassen

[Mult von]	[Mult bis] Name	Typ	Beschreibung
[0..*]	[0..*] Actor	Actor	die in der Bibliothek gelisteten Akteure
[0..*]	[1..1] UseCaseRepository	UseCaseRepository	das zugehörige Anwendungsfall-Repository

6.1.2.6 Area

Fachgebiet oder Aktivität, die durch eine Menge von Konzepten und durch Terminologie gekennzeichnet ist, die von den Fachleuten dieses Gebiets verstanden wird

In Tabelle 9 sind alle Attribute von „Area“ aufgeführt.

Tabelle 9 – Attribute von „UseCaseCoreConcepts::Area“

Name	Typ	Beschreibung
Beschreibung	String	Geerbt von: IdentifiedObject
mRID	String	Geerbt von: IdentifiedObject
name	String	Geerbt von: IdentifiedObject

In Tabelle 10 sind alle Assoziationsenden von „Area“ mit anderen Klassen aufgeführt.

Tabelle 10 – Assoziationsenden von „UseCaseCoreConcepts::Area“ mit anderen Klassen

[Mult von]	[Mult bis] Name	Typ	Beschreibung
[0..*]	[0..1] ParentArea	Area	Verknüpfung mit einem Hauptbereich
[0..*]	[1..1] AreaLibrary	AreaLibrary	die den Bereich verknüpfende Bibliothek
[0..*]	[0..*] BusinessCase	BusinessCase	zugehörige Geschäftsfälle, die zur Domäne gehören
[0..1]	[0..*] SubArea	Area	Verknüpfung mit einem Unterbereich