

DIN EN 62714-1

ICS 35.240.50

Einsprüche bis 2012-08-04

Entwurf

**Datenaustauschformat für Planungsdaten industrieller
Automatisierungssysteme (AutomationML) –
Teil 1: Architektur und allgemeine Festlegungen
(IEC 65E/223/CD:2011)**

Engineering data exchange format for use in industrial automation systems engineering
(AutomationML) –
Part 1: architecture and general requirements
(IEC 65E/223/CD:2011)

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2012-05-29 wird der Öffentlichkeit zur Prüfung und
Stellungnahme vorgelegt.

Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses
Entwurfes besonders zu vereinbaren.

Stellungnahmen werden erbeten

- vorzugsweise als Datei per E-Mail an dke@din.de in Form einer Tabelle. Die Vorlage dieser Tabelle kann
im Internet unter www.din.de/stellungnahme oder für Stellungnahmen zu Norm-Entwürfen der DKE unter
www.dke.de/stellungnahme abgerufen werden;
- oder online im Norm-Entwurfs-Portal des DIN unter www.entwuerfe.din.de, sofern dort wiedergegeben;
- oder in Papierform an die DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im
DIN und VDE (Hausanschrift: Stresemannallee 15, 60596 Frankfurt am Main).

Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevanten
Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen.

Gesamtumfang 79 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE

Inhalt

	Seite
Einleitung	7
1 Anwendungsbereich	8
2 Normative Verweisungen	8
3 Begriffe und Abkürzungen	8
3.1 Begriffe	8
3.2 Abkürzungen	11
4 Konformität	11
5 Spezifikation der AutomationML-Architektur	12
5.1 Einleitung	12
5.2 Allgemeine AutomationML-Architektur	12
5.3 AutomationML-Dokumentversionen	13
5.4 Metainformationen über das AutomationML-Quell-Werkzeug	14
5.5 Objektkennung	15
5.6 Spezifikation der AutomationML-Beziehungen	17
5.7 Spezifikation von AutomationML-Dokumentverweisungen	22
6 AutomationML-Basisbibliotheken	23
6.1 Einleitung	23
6.2 Allgemeine Festlegungen	23
6.3 AutomationML-Schnittstellenklassenbibliothek	24
6.4 AutomationML-Basisrollenklassenbibliothek	29
7 Modellierung von benutzerdefinierten Daten	36
7.1 Allgemeines	36
7.2 Benutzerdefinierte Attribute	36
7.3 Benutzerdefinierte Schnittstellenklassen	36
7.4 Benutzerdefinierte Rollenklassen	37
7.5 Benutzerdefinierte Anlagenkomponentenklassen	38
7.6 Benutzerdefinierte Instanzhierarchien	39
8 Erweiterte AutomationML-Konzepte	40
8.1 Allgemeiner Überblick	40
8.2 AutomationML-Port-Objekt	40
8.3 AutomationML-Facettenobjekt	40
8.4 AutomationML-Gruppenobjekt	41
8.5 AutomationML-Eigenschaftsmenge	42
8.6 Unterstützung multipler Rollen	44
8.7 Aufteilung von AutomationML-Toplevel-Daten auf verschiedene Dokumente	45
8.8 Internationalisierung	45

	Seite
8.9 Versionsinformationen von AutomationML-Objekten.....	45
9 Literaturhinweise	46
Anhang A (informativ) Allgemeine Einführung in AutomationML	47
A.1 Allgemeine AutomationML-Konzepte.....	47
A.1.1 Architektur von AutomationML	47
A.1.2 Modellierung von Informationen zur Anlagentopologie.....	49
A.1.3 Referenzierung von Geometrie- und Kinematikinformationen.....	50
A.1.4 Referenzierung von Logikinformationen	51
A.1.5 Modellierung von Beziehungen.....	51
A.2 Erweiterte AutomationML-Konzepte und Beispiele.....	54
A.2.1 Allgemeiner Überblick	54
A.2.2 AutomationML-Port-Konzept.....	55
A.2.3 AutomationML-Facettenkonzept	58
A.2.4 AutomationML-Gruppenkonzept	60
A.2.5 Eigenschaftsmengenkonzep.....	64
A.2.6 Prozess-Produkt-Ressourcen-Konzept.....	67
A.2.7 Unterstützung multipler Rollen	76
Bilder	
Bild 1 – Überblick über das Datenaustauschformat für Planungsdaten (AutomationML).....	7
Bild 2 – Versionsangabe eines AutomationML-Dokuments	13
Bild 3 – XML-Text der AutomationML-Information zum Quellwerkzeug	15
Bild 4 – Beispiel für die Objektkennung einer AutomationML-Klasse	16
Bild 5 – Beispiel für die Objektkennung einer AutomationML-Objektinstanz	17
Bild 6 – Beispiel einer Eltern-Kind-Beziehung zwischen AutomationML-Objekten.....	18
Bild 7 – Beispiel einer Eltern-Kind-Beziehung zwischen Klassen	18
Bild 8 – Beispiel für eine Vererbungsbeziehung zwischen zwei Klassen	19
Bild 9 – Beispiel einer Klasse-Instanz-Beziehung	20
Bild 10 – Beispiel einer Beziehung als a) Blockdiagramm und b) Objektbaum	21
Bild 11 – Beispielbeziehung zwischen den Objekten „PLC1“ und „Rob1“	22
Bild 12 – AutomationML-Basisschnittstellenklassenbibliothek.....	25
Bild 13 – XML-Beschreibung der AutomationML-Basisschnittstellenklassenbibliothek.....	26
Bild 14 – AutomationML-Basisrollenklassenbibliothek.....	29
Bild 15 – AutomationMLBaseRoleClassLib.....	30
Bild 16 – XML-Text der AutomationML-Basisrollenklassenbibliothek.....	31
Bild 17 – Beispiel eines benutzerdefinierten Attributs	36
Bild 18 – Beispiel für eine benutzerdefinierte Schnittstellenklasse in einer benutzerdefinierten Schnittstellenklassenbibliothek	37
Bild 19 – Beispiel für eine benutzerdefinierte Rollenklasse in einer benutzerdefinierten Rollenklassenbibliothek.....	38
Bild 20 – Beispiele für unterschiedliche benutzerdefinierte Anlagenkomponentenklassen	39

	Seite
Bild 21 – Beispiel für eine benutzerdefinierte Instanzhierarchie	39
Bild 22 – AutomationML-Darstellung einer benutzerdefinierten Instanzhierarchie	40
Bild 23 – Beispiel zur Erläuterung des Konzepts der Eigenschaftsmenge (PropertySet)	43
Bild 24 – XML-Text des Eigenschaftsmengenbeispiels	44
Bild 25 – Allgemeine Architektur von AutomationML	48
Bild 26 – Anlagentopologie mit AutomationML	49
Bild 27 – Verweisung von CAEX auf ein COLLADA-Dokument	51
Bild 28 – Verweisung von CAEX auf ein PLCopen-XML-Dokument	51
Bild 29 – Beziehungen in AutomationML	52
Bild 30 – XML-Beschreibung des Beispiels für die Beziehungen	53
Bild 31 – XML-Text der Anlagenkomponentenklassenbibliothek des Beispiels für die Beziehungen	53
Bild 32 – XML-Text der Instanzhierarchie des Beispiels für die Beziehungen	54
Bild 33 – Port-Konzept	55
Bild 34 – Beispiel, das das AutomationML-Port-Konzept beschreibt	55
Bild 35 – XML-Beschreibung des AutomationML-Port-Konzepts	56
Bild 36 – XML-Text, der das AutomationML-Port-Konzept beschreibt	57
Bild 37 – Definition einer benutzerdefinierten AutomationML-Port-Klasse „myPortClass“	58
Bild 38 – Beispiel für eine AutomationML-Facette	59
Bild 39 – XML-Text des AutomationML-Facettenbeispiels	59
Bild 40 – Beispiel für eine AutomationML-Gruppe	60
Bild 41 – XML-Text des AutomationML-Gruppenbeispiels	61
Bild 42 – Vereinigung von Facetten- und Gruppenkonzept	62
Bild 43 – XML-Textansicht des Beispiels für die Vereinigung von Facetten- und Gruppenkonzept	63
Bild 44 – Allgemeine HMI-Schablone „B“, die eine Prozessvariable „Y“ eines Förderbandes visualisiert	64
Bild 45 – Erzeugtes HMI-Ergebnis „B“, das beide Förderbänder mit einzelnen Prozessvariablen visualisiert	64
Bild 46 – Beispiel für die Eigenschaftsmenge (PropertySet)	65
Bild 47 – Beispiel für die Eigenschaftsmenge (PropertySet)	65
Bild 48 – XML-Text für die Instanzhierarchie	66
Bild 49 – Beispiel für eine Eigenschaftsmengenbibliothek als XML-Code	67
Bild 50 – Basiselemente des Produkt-Prozess-Ressourcen-Konzepts	68
Bild 51 – PPRConnector-Schnittstelle	69
Bild 52 – Beispiel für das Produkt-Prozess-Ressourcen-Konzept	69
Bild 53 – AutomationML-Rollen, die für das Prozess-Produkt-Ressourcen-Konzept erforderlich sind	70
Bild 54 – Elemente des Beispiels	70
Bild 55 – Verknüpfungen innerhalb des Beispiels	71
Bild 56 – Verknüpfungen der ressourcenzentrierten Sichtweise auf das Beispiel	72
Bild 57 – Instanzhierarchie des Beispiels in AutomationML	73
Bild 58 – Interne Elemente dieses Beispiels	74

	Seite
Bild 59 – Interne Verknüpfungen (InternalLinks) des Beispiels.....	74
Bild 60 – Instanzhierarchie des Beispiels in XML	75
Bild 61 – Beispiel für eine benutzerdefinierte Instanz, die multiple Rollen unterstützt.....	77
Bild 62 – XML-Text der AutomationML-Darstellung für die Unterstützung multipler Rollen	78
Bild 63 – Rollenklassenbibliothek, die dem Beispiel für die Unterstützung multipler Rollen entspricht.....	78
Bild 64 – XML-Text der Rollenklassenbibliothek	79
<u>Tabellen</u>	
Tabelle 1 – Abkürzungen.....	11
Tabelle 2 – Metainformationen über das AutomationML-Quell-Werkzeug	15
Tabelle 3 – Schnittstellenklassen der AutomationML-Schnittstellenklassenbibliothek (AutomationMLInterfaceClassLib).....	24
Tabelle 4 – Schnittstellenklasse „AutomationMLBaseInterface“	26
Tabelle 5 – Schnittstellenklasse „Order“	26
Tabelle 6 – Schnittstellenklasse „PortConnector“	27
Tabelle 7 – Schnittstellenklasse „PPRConnector“	27
Tabelle 8 – Schnittstellenklasse „ExternalDataConnector“	27
Tabelle 9 – Schnittstellenklasse „COLLADAInterface“	27
Tabelle 10 – Schnittstellenklasse „PLCopenXMLInterface“	28
Tabelle 11 – Schnittstellenklasse „Communication“	28
Tabelle 12 – Schnittstellenklasse „SignalInterface“	28
Tabelle 13 – Rollenklasse „AutomationMLBaseRole“	31
Tabelle 14 – Rollenklasse „Group“	31
Tabelle 15 – Rollenklasse „Facet“	32
Tabelle 16 – Optionale Attribute für AutomationML-Port-Objekte.....	32
Tabelle 17 – Unterattribute des Attributs „Cardinality“	33
Tabelle 18 – Schnittstellen der AutomationML-Port-Klasse.....	33
Tabelle 19 – Rollenklasse „Resource“	33
Tabelle 20 – Rollenklasse „Product“	34
Tabelle 21 – Rollenklasse „Process“	34
Tabelle 22 – Rollenklasse „Structure“	34
Tabelle 23 – Rollenklasse „ProductStructure“	35
Tabelle 24 – Rollenklasse „ProcessStructure“	35
Tabelle 25 – Rollenklasse „ResourceStructure“	35
Tabelle 26 – Rollenklasse „PropertySet“	35
Tabelle 27 – Überblick über wesentliche erweiterte AutomationML-Konzepte.....	54

Nationales Vorwort

Das internationale Dokument IEC 65E/223/CD:2011 „Engineering data exchange format for use in industrial automation systems engineering (AutomationML) – Part 1: architecture and general requirements“ (CD, en: Committee Draft) ist unverändert in diesen Norm-Entwurf übernommen worden. Dieser Norm-Entwurf enthält eine autorisierte deutsche Übersetzung.

Das internationale Dokument wurde vom SC 65E „Devices and integration in enterprise systems“ der Internationalen Elektrotechnischen Kommission (IEC) erarbeitet und den nationalen Komitees zur Stellungnahme vorgelegt.

Die IEC und das Europäische Komitee für Elektrotechnische Normung (CENELEC) haben vereinbart, dass ein auf IEC-Ebene erarbeiteter Entwurf für eine Internationale Norm zeitgleich (parallel) bei IEC und CENELEC zur Umfrage (CDV-Stadium) und Abstimmung als FDIS (en: Final Draft International Standard) bzw. Schluss-Entwurf für eine Europäische Norm gestellt wird, um eine Beschleunigung und Straffung der Normungsarbeit zu erreichen. Dokumente, die bei CENELEC als Europäische Norm angenommen und ratifiziert werden, sind unverändert als Deutsche Normen zu übernehmen.

Da der Abstimmungszeitraum für einen FDIS bzw. Schluss-Entwurf prEN nur 2 Monate beträgt, und dann keine sachlichen Stellungnahmen mehr abgegeben werden können, sondern nur noch eine „JA/NEIN“-Entscheidung möglich ist, wobei eine „NEIN“-Entscheidung fundiert begründet werden muss, wird bereits der CD als DIN-Norm-Entwurf veröffentlicht, um die Stellungnahmen aus der Öffentlichkeit frühzeitig berücksichtigen zu können.

Für diesen Norm-Entwurf ist das nationale Arbeitsgremium K 941 „Engineering“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE (www.dke.de) zuständig.

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf eine Norm ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnichtsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils neueste gültige Ausgabe der in Bezug genommenen Norm.

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

Der Zusammenhang der zitierten Normen mit den entsprechenden Deutschen Normen ergibt sich, soweit ein Zusammenhang besteht, grundsätzlich über die Nummer der entsprechenden IEC-Publikation. Beispiel: IEC 60068 ist als EN 60068 als Europäische Norm durch CENELEC übernommen und als DIN EN 60068 ins Deutsche Normenwerk aufgenommen.

Datenaustauschformat für Planungsdaten industrieller Automatisierungssysteme (AutomationML) – Teil 1: Architektur und allgemeine Festlegungen

Einleitung

Bei dem in dieser Norm festgelegten Datenaustauschformat (AutomationML) handelt es sich um ein auf einem XML-Schema basierendes Datenformat, das mit dem Ziel, den Datenaustausch zwischen Planungswerkzeugen bzw. -tools (en: engineering tools) in einer heterogenen Planungswerkzeug-Landschaft zu unterstützen, entwickelt wurde. Ein informativer Überblick über AutomationML findet sich in Anhang A.

Das Ziel von AutomationML ist, Planungswerkzeuge der bestehenden heterogenen Werkzeug-Landschaft in ihren unterschiedlichen Disziplinen, z. B. Maschinen- und Anlagenbau, elektrische Konstruktion, Verfahrenstechnik, Prozessleittechnik, HMI-Entwicklung, SPS-Programmierung, Roboterprogrammierung usw., miteinander zu verbinden.

AutomationML speichert Planungsinformationen unter Befolgung des objektorientierten Paradigmas und ermöglicht die Modellierung von realen Anlagenkomponenten als Datenobjekte, in denen unterschiedliche Aspekte gekapselt sind. Ein Objekt kann aus anderen Unterobjekten bestehen und aber auch selbst Teil einer größeren Zusammenstellung oder Aggregation sein. Typische Objekte in der Anlagenautomatisierung umfassen Informationen zur Topologie, Geometrie, Kinematik und Logik, während die Logik Abfolge, Verhalten und Steuerung umfasst.

AutomationML vereinigt vorhandene Industriedatenformate, die zur Speicherung und zum Austausch von verschiedenen Aspekten der Planungsinformationen dienen. Diese Standards werden innerhalb ihrer eigenen Spezifikationen so, wie sie sind ("as is"), verwendet und nicht für im Zusammenhang mit AutomationML stehende Erfordernisse verzweigt.

Der Kern von AutomationML ist das Toplevel-Datenformat CAEX, das die verschiedenen Datenformate miteinander verbindet. Aus diesem Grunde hat AutomationML eine inhärent verteilte Dokumentarchitektur.

Bild 1 verdeutlicht den Grundaufbau von AutomationML und die Verteilung der Topologie-, Geometrie-, Kinematik- und Logikinformationen.

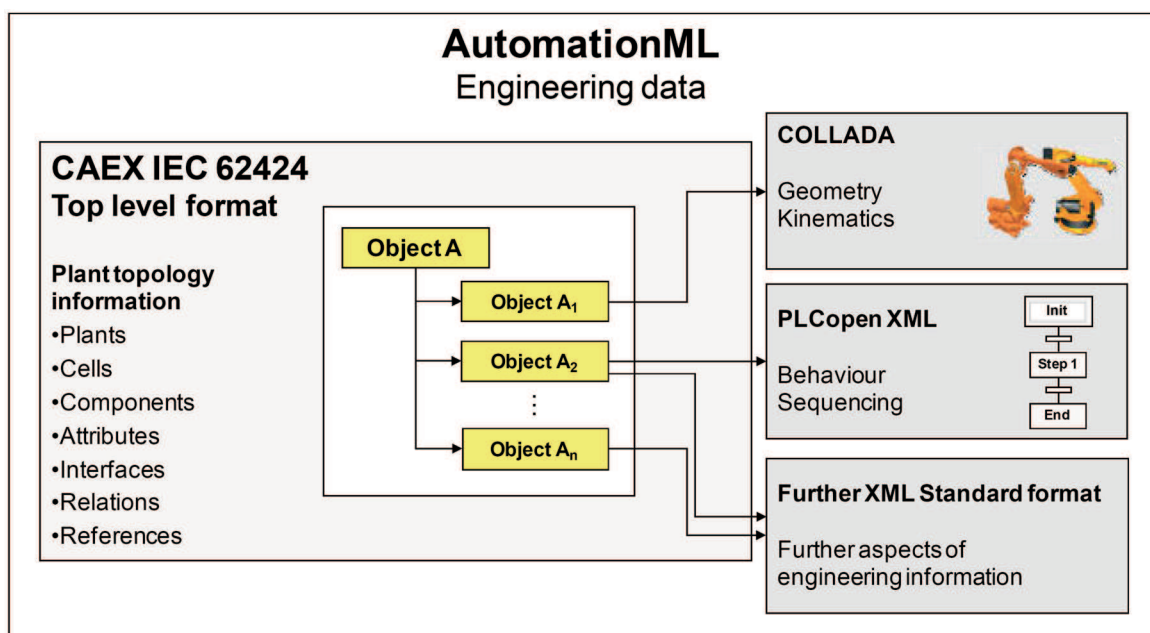


Bild 1 – Überblick über das Datenaustauschformat für Planungsdaten (AutomationML)

Wegen der bei AutomationML zu berücksichtigenden unterschiedlichen Gesichtspunkte besteht diese Norm aus mehreren Teilen, die sich auf verschiedene Aspekte konzentrieren.

- Teil 1: Architektur und allgemeine Festlegungen
Dieser Teil legt die Architektur von AutomationML im Allgemeinen sowie die Modellierung von Planungsdaten, Klassen, Instanzen, Beziehungen, Verweisungen, Hierarchien, AutomationML-Basisbibliotheken und erweiterte AutomationML-Konzepte fest.
- Teil 2: Rollenbibliotheken
Dieser Teil legt zusätzliche AutomationML-Bibliotheken fest.
- Teil 3: Geometrie und Kinematik
Dieser Teil legt die Modellierung von Geometrie- und Kinematikinformationen fest.
- Teil 4: Logik
Dieser Teil legt die Modellierung von Informationen bezüglich Logik, Abfolge, Verhalten und/oder Steuerung fest.

Es können noch weitere Teile hinzukommen, um weitere Datenstandards mit AutomationML zu verbinden. Anhang A enthält eine informative Einleitung in AutomationML sowie entsprechende Anwendungsfälle und Beispiele.

1 Anwendungsbereich

Diese Norm legt das Datenaustauschformat für Planungsdaten AutomationML¹ fest. Teil 1 dieser Norm legt allgemeine Anforderungen für die Modellierung von Planungsinformationen für den Austausch zwischen Planungswerkzeugen im Bereich der Anlagenautomatisierung mittels AutomationML fest. Die hierin enthaltenen Vorgaben gelten für die Export-/Importanwendungen von verwandten Werkzeugen.

ANMERKUNG 1 Die Benennung „AutomationML“ steht für „Datenaustauschformat für Planungsdaten“.

ANMERKUNG 2 Anhang A enthält eine Einführung in AutomationML.

2 Normative Verweisungen

Die folgenden zitierten Dokumente sind für die Anwendung dieses Dokuments erforderlich. 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 60027, *Letter symbols to be used in electrical technology*

IEC 62424:2008, *Representation of process control engineering – Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools*

ISO/IEC 9834-8, *Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities: Generation and registration of Universally Unique Identifiers (UUIDs) and their use as ASN.1 Object Identifier components*

ISO 1000, *SI units and recommendations for the use of their multiples and of certain other units*

3 Begriffe und Abkürzungen

3.1 Begriffe

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe.

¹ AutomationML[®] ist eine Marke des AutomationML e.V. Diese Angabe dient nur zur Information der Anwender dieser Internationalen Norm und stellt keine Anerkennung des Markeninhabers oder irgendeines seiner Produkte durch die IEC dar. Für die Übereinstimmung mit diesem Profil ist die Verwendung der eingetragenen Marke nicht erforderlich. Für die Verwendung der Marke AutomationML ist jedoch die Erlaubnis des Markeninhabers einzuholen.

3.1.1

AutomationML

XML-basiertes Datenaustauschformat für Anlagenplanungsdaten

3.1.2

Automatisierungsobjekt

Einheit im automatisierten System

ANMERKUNG Automatisierungsobjekte sind z. B. Automatisierungskomponenten, Ventile und/oder Signale.

3.1.3

AutomationML-Objekt

Datendarstellung eines Automatisierungsobjekts oder einer Gruppe von Automatisierungsobjekten

ANMERKUNG 1 Hierbei handelt es sich um ein Kernelement von AutomationML. Ein AutomationML-Objekt kann Verwaltungselemente, Attribute, Schnittstellen, Beziehungen und Verweisungen enthalten. Ein AutomationML-Objekt ist eine eigene Instanz und wird aus AutomationML-Standardklassen abgeleitet.

ANMERKUNG 2 AutomationML-Objekte haben eine Beziehung zu ihrer entsprechenden AutomationML-Klasse.

ANMERKUNG 3 Beispiele für Beziehungen sind Typ-Instanz- und Kopie-Instanz-Beziehungen. Einzelne Instanzen können erweitert werden, z. B. durch aggregierte Objekte oder Attribute.

ANMERKUNG 4 AutomationML-Objekte haben eine Kopie-Instanz-Beziehung zu ihrer AutomationML-Klasse.

3.1.4

AutomationML-Klasse

vordefinierter AutomationML-Objektyp

ANMERKUNG 1 AutomationML-Klassen werden in AutomationML-Bibliotheken gespeichert.

ANMERKUNG 2 AutomationML-Klassen definieren wiederverwendbare Musterlösungen, die durch Attribute, Schnittstellen und aggregierte Objekte beschrieben werden.

ANMERKUNG 3 AutomationML-Klassen können für Mehrfachinstanziierungen verwendet werden.

3.1.5

AutomationML-Attribut

Eigenschaft, die zu einem AutomationML-Objekt gehört

ANMERKUNG AutomationML-Attribute werden als XML-Elemente nach IEC 62424:2008 beschrieben.

3.1.6

AutomationML-Dokument

bestimmtes CAEX-Dokument, das einschließlich aller referenzierten Unterdokumente IEC 62714 entspricht

ANMERKUNG AutomationML-Dokumente können als Dateien, aber auch z. B. als Zeichenfolgen oder Datenströme gespeichert werden.

3.1.7

AutomationML-Datei

bestimmte CAEX-Datei, die mit der Erweiterung „.aml“ und ohne referenzierte Unterdateien IEC 62714 entspricht

3.1.8

AutomationML-Schnittstelle

einzelner Verbindungspunkt, der zu einem AutomationML-Objekt gehört und mit einer anderen Schnittstelle verknüpft sein kann

ANMERKUNG Schnittstellen ermöglichen die Beschreibung von Beziehungen zwischen Objekten durch die Definition von internen CAEX-Schnittstellen (CAEX InternalLink). Beispiele sind Signalschnittstellen, Produktschnittstellen oder Stromversorgungsschnittstellen.

3.1.9

AutomationML-Bibliothek

Bibliothek, die AutomationML-Klassen enthält

3.1.10

AutomationML-Port

AutomationML-Objekt, das einen Behälter für eine Gruppe von Schnittstellen darstellt, die durch zusätzliche Eigenschaften charakterisiert sind

ANMERKUNG Ports gehören zu einem AutomationML-Elternobjekt und beschreiben komplexe Schnittstellen dieses Objekts. Ports können auf einer höheren Abstraktionsebene miteinander verbunden sein.

3.1.11

AutomationML-Gruppe

AutomationML-Objekt, das eine bestimmte Ansicht von AutomationML-Objekten zur Verfügung stellt

3.1.12

AutomationML-Facette

AutomationML-Objekt, das eine bestimmte Ansicht von AutomationML-Attributen oder -Schnittstellen eines AutomationML-Objekts zur Verfügung stellt

3.1.13

Kopie-Instanz-Beziehung

Beziehung zwischen einer AutomationML-Klasse und einem AutomationML-Objekt

ANMERKUNG Die Instanz empfängt eine Kopie aller Funktionen und Eigenschaften der AutomationML-Quellenklasse. Änderungen der Klasse führen nicht zu Änderungen der Instanz. Innerhalb der Instanz werden Klasseneigenschaften individualisiert. Weitere Kopien sind aufgrund der Kenntnis der AutomationML-Quellenklasse möglich.

3.1.14

globaler eindeutiger Bezeichner

(en: Global Unique Identifier, GUID)

Implementierung eines UUID

ANMERKUNG 1 Reales GUID-Beispiel: „{AC76BA86-7AD7-1033-7B44-A70000000000}“.

ANMERKUNG 2 In IEC 62714 werden GUID auch in Kurzform dargestellt, z. B. als „GUID1“, „GUID2“ usw. Dies dient der Lesbarkeit und steht an der Stelle eines realen GUID.

3.1.15

Vererbung

Beziehung zwischen zwei AutomationML-Klassen

ANMERKUNG Die abgeleitete Klasse erbt alle Attribute und Merkmale der Basisklasse.

3.1.16

Vererbungshierarchie

Hierarchie von Vererbungsbeziehungen

3.1.17

Instanz

Synonym für das AutomationML-Objekt

3.1.18

Instanzhierarchie

Hierarchie von AutomationML-Objekten

3.1.19

Topologie

hierarchischer Aufbau eines Systems, der sich als Objektbaum darstellen lässt

ANMERKUNG Dies schließt multiple Hierarchien, überkreuzte Strukturen und Objektnetzwerke ein.

3.1.20

Beziehung

(en: relation)

Zuordnung zwischen Objekten des Toplevel-Formats CAEX

3.1.21

Verweisung

(en: reference)

Zuordnung zwischen einem AutomationML-Objekt und extern gespeicherten Informationen

3.1.22

universell eindeutiger Bezeichner

(en: Universal Unique Identifier, UUID)

eindeutiger Bezeichner für AutomationML-Objekte

3.2 Abkürzungen

Tabelle 1 – Abkürzungen

Abkürzung	Deutsche Benennung	Englische Benennung
AutomationML	Automatisierungsauszeichnungssprache	Automation Markup Language
CAE	Rechnerunterstützter Entwurf	Computer Aided Engineering
CAEX	– (Softwaretechnisch unterstützter Austausch für Ingenieurarbeit, neutrales Datenformat nach IEC 62424:2008)	Computer Aided Engineering eXchange, neutral data format according to IEC 62424:2008
COLLADA	– (etwa: gemeinschaftliche Entwurfsaktion)	Collaborative Design Activity
CSG	Konstruktive Festkörpergeometrie für die verlustlose 3D-Geometriebeschreibung	Constructive Solid Geometry for lossless 3D geometry description
DCC	Erstellung von digitalen Inhalten	Digital Content Creation
GUID	Globaler eindeutiger Bezeichner	Global Unique Identifier
HMI	Mensch-Maschine-Schnittstelle	Human Machine Interface
ID	Bezeichner	Identifier
IR	Industrieroboter	Industrial Robot
SPS	Speicherprogrammierbare Steuerung	Programmable Logic Controller (PLC)
RPY	Roll-Nick-Gier-Winkel	Roll Pitch Yaw
URL	Einheitlicher Quellenweiser (Internetadresse)	Uniform Resource Locator
UUID	Universell eindeutiger Bezeichner	Universal Unique Identifier
XML	Erweiterbare Auszeichnungssprache	Extensible Markup Language

4 Konformität

Um Konformität mit Teil 1 von IEC 62714 in Bezug auf die Unterstützung von AutomationML zu beanspruchen, müssen die Anforderungen der Abschnitte 5, 6, 7 und 8 erfüllt sein. Teil 1 dieser Norm legt keine Einzelheiten des Datenaustauschverfahrens und auch keine Anforderungen an die Implementierung von Import-/Exportwerkzeugen fest.

5 Spezifikation der AutomationML-Architektur

5.1 Einleitung

Ein informativer Überblick über AutomationML ist in Anhang A enthalten. Im Zentrum von AutomationML steht das Toplevel-Datenformat CAEX, ein neutrales Datenformat, das in Übereinstimmung mit IEC 62424:2008 bereits eingeführte Datenformate für die Planungsaspekte Topologie, Geometrie, Kinematik, Verhalten und Abfolgeinformationen miteinander verbindet. Deshalb ist ein wesentliches Merkmal von AutomationML eine inhärente verteilte Dokumentarchitektur, die sich auf die erwähnten Planungsaspekte konzentriert.

Die Bilder dienen nur zur Erläuterung. Die grafische Darstellung ist nicht normativ.

5.2 Allgemeine AutomationML-Architektur

Bezüglich der allgemeinen AutomationML-Architektur gelten die folgenden Festlegungen:

Informationen zur Anlagentopologie: Die Anlagentopologie fungiert als die Toplevel-Datenstruktur von anlagenplanungstechnischen Informationen und ist mittels des Datenformats CAEX nach IEC 62424:2008 zu modellieren. Eventuelle Erweiterungen der CAEX-Semantik werden gesondert beschrieben. Multiple und überkreuzte Hierarchiestrukturen sind unter Anwendung des Konzepts des Spiegelobjekts zu verwenden. Spiegelobjekte dürfen nicht verändert werden, und sämtliche Änderungen sind am Master-Objekt vorzunehmen.

ANMERKUNG 1 Dies entspricht dem Spiegel-Konzept nach IEC 62424:2008, A.2.14. Die Instanz wird „Spiegelobjekt“ genannt, während die angezeigte Objektinstanz als „Master-Objekt“ bezeichnet wird. Das Spiegelobjekt gilt als identisch mit dem Master-Objekt. Dies ermöglicht die Einordnung einer Objektinstanz in verschiedene Anlagenhierarchien und damit die Modellierung von komplexen Objektnetzwerken mit überkreuzten Strukturen.

ANMERKUNG 2 AutomationML ändert nicht die Syntax des CAEX-Datenformats. Ein informativer Überblick und weitere Beispiele zur Anlagentopologie finden sich in Anhang A.1.2 und in IEC 62424:2008.

Verweisungs- und Beziehungsinformationen: Verweisungen (en: references) und Beziehungen (en: relations) sind nach 5.6 und 5.7 zu speichern. Beziehungen zwischen extern gespeicherten Informationen sind unter Anwendung von CAEX-Mechanismen zu speichern. Falls erforderlich, sind die miteinander in Beziehung stehenden „Verknüpfungspartner“ in der CAEX-Anlagentopologiebeschreibung mit Hilfe von externen CAEX-Schnittstellen zu veröffentlichen. Sie sind aus den in 6.3 festgelegten AutomationML-Standardschnittstellenklassen abzuleiten.

ANMERKUNG 1 Verweisungen beschreiben Verknüpfungen von CAEX-Objekten mit extern gespeicherten Informationen. Ein informativer Überblick über die möglichen Beziehungen findet sich in Anhang A.1.5. Verweisungen und die Veröffentlichung von Schnittstellen sind in anderen Teilen dieser Norm beschrieben.

ANMERKUNG 2 Beziehungen beschreiben Zuordnungen zwischen CAEX-Objekten.

Geometrie- und Kinematikinformationen: Informationen, die für Geometrie und Kinematik von Bedeutung sind, sind unter Verwendung des Datenformats COLLADA zu speichern. COLLADA-Schnittstellen, die innerhalb des Toplevel-Formats miteinander zu verbinden sind, müssen als externe CAEX-Schnittstellen „veröffentlicht“ werden.

ANMERKUNG 1 AutomationML ändert nicht die Syntax des COLLADA-Datenformats. Ein überblicksartiges Beispiel zur Referenzierung von COLLADA findet sich in Anhang A.1.3. Einzelheiten sind in Teil 3 dieser Norm festgelegt.

ANMERKUNG 2 Aus den COLLADA-Geometrieinformationen verschiedener Objekte kann automatisch eine vollständige Szene abgeleitet werden. Auf diese Dateien kann von CAEX aus verwiesen werden, und sie können mit Hilfe von CAEX-Verknüpfungsmechanismen miteinander verknüpft sein bzw. werden.

Logikinformationen: Logikinformationen sind unter Verwendung des Datenformats PLCopen XML zu speichern. Falls Logikelemente, wie z. B. Variable oder Signale, innerhalb des Toplevel-Formats miteinander

verbunden werden müssen, sind sie als externe CAEX-Schnittstellen zu „veröffentlichen“. Alle Elemente von PLCopen XML, die innerhalb des Toplevel-Formats veröffentlicht werden, müssen innerhalb von PLCopen XML eine eindeutige Kennung (ID) haben.

ANMERKUNG Logikinformationen beschreiben Abfolgen von Aktionen und das interne Verhalten von Objekten, einschließlich E/A-Verbindungen und logischer Variablen. AutomationML ändert nicht das PLCopen XML-Format. Ein informativer Überblick zur Referenzierung von Logikinformation findet sich in Anhang A.1.4. Einzelheiten sind in Teil 4 dieser Norm festgelegt.

Referenzierung anderer Datenformate: Diese Norm kann später um weitere Teile erweitert werden, die die Einbindung weiterer XML-Datenformate unter Verwendung der AutomationML-Verweisungsmechanismen festlegen. Einzelheiten können in weiteren Teilen dieser Norm festgelegt werden.

Das Datenformat AutomationML stellt weder Konsistenzüberprüfungen von Einschränkungen, Attributwerten, Beziehungen und Verweisungen noch Überprüfungen von enthaltenen Daten auf semantische Richtigkeit zur Verfügung: Diese Überprüfungen fallen vielmehr in den Aufgabenbereich des Quell- oder Zielwerkzeugs oder der entsprechenden Import-/Exportanwendung. Nur AutomationML ermöglicht eine syntaktische Prüfung des Dokuments in Bezug auf die entsprechenden Schemata.

5.3 AutomationML-Dokumentversionen

Jedes AutomationML-XML-Dokument muss die zugrunde liegende AutomationML-Version speichern, der dieses Dokument folgt.

ANMERKUNG Abschnitt 8.9 enthält normative Festlegungen im Hinblick auf Versionsinformationen, die sich auf AutomationML-Objektinstanzen beziehen. Die Speicherung von werkzeugspezifischen Metainformationen ist in Abschnitt 5.4 festgelegt.

Entsprechend gelten die folgenden Festlegungen:

- Das CAEX-Wurzelement „CAEXFile“ jedes AutomationML-Toplevel-Dokuments muss das CAEX-Kind-Element „AdditionalInformation“ enthalten.
- Das Element „AdditionalInformation“ muss ein Attribut „AutomationMLVersion“ enthalten.
- Der Wert dieses Attributs „AutomationMLVersion“ muss im XML-Dokument gespeichert sein. Nach der vorliegenden Norm muss dies „2.0“ sein.
- Jedes referenzierte CAEX-Dokument muss derselben AutomationML-Version des Wurzeldokuments folgen. Jegliche Vermischung von Dokumenten verschiedener AutomationML-Versionen ist ausdrücklich verboten.
- Jedes externe referenzierte Dokument muss ebenfalls den in der obigen AutomationML-Versionsfestlegung angegebenen benannten Schemaversionen folgen. Jegliche Vermischung von externen Dokumentversionen über die betreffende AutomationML-Versionsspezifikation hinaus ist ausdrücklich verboten.

Bild 2 erläutert den XML-Text für ein CAEX-Dokument, das der AutomationML-Version 2.0 folgt.

```
<CAEXFile xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="
CAEX_ClassModel.xsd" FileName="AutomationMLStandardLibrary2010-01-14_v1.99.aml" SchemaVersion="2.15">
  <AdditionalInformation AutomationMLVersion="2.0"/>
```

Bild 2 – Versionsangabe eines AutomationML-Dokuments

- Jede AutomationML-Standardbibliothek und jede benutzerdefinierte Bibliothek muss ihre jeweilige Versionsnummer unter Verwendung des CAEX-Elements „Version“ definieren. Die Syntax des Werts der Versionsnummer wird im vorliegenden Teil dieser Norm nicht festgelegt.
- Falls gefordert, müssen CAEX-Klassen ihre Versionsnummer unter Verwendung des CAEX-Elements „Version“ definieren. Die Syntax und Semantik der Versionsnummer von Klassen innerhalb einer Bibliothek wird im vorliegenden Teil dieser Norm nicht festgelegt.
- Es ist verboten, gleiche Bibliotheken unterschiedlicher Versionen in derselben AutomationML-Datei zu speichern.

ANMERKUNG Auf diese Weise wird die Eindeutigkeit der Bibliotheksnamen innerhalb einer AutomationML-Datei sichergestellt.

- Der Ersteller eines AutomationML-Dokuments muss sicherstellen, dass ausschließlich mit der jeweiligen Version kompatible Klassen und externe Dokumente referenziert werden.

Die vorliegende Norm basiert auf den folgenden Dokumentformaten:

- CAEX Version 2.15;
- PLCopenXML 2.0 und 2.0.1;
- COLLADA 1.4.1 und 1.5.0;
- AutomationML-Standardbibliotheken nach Teil 1 und anderen Teilen dieser Norm.

5.4 Metainformationen über das AutomationML-Quell-Werkzeug

Falls benutzerdefinierte Daten von einem Quellwerkzeug an ein Zielwerkzeug übermittelt werden müssen, ist es notwendig, Informationen über das Quellwerkzeug direkt im AutomationML-Dokument zu speichern. Entsprechend gelten die folgenden Festlegungen:

- Jedes AutomationML-Dokument muss Informationen über das Werkzeug zur Verfügung stellen, das das AutomationML-Dokument erstellt hat.
- In einer Datenaustausch-Werkzeugkette müssen alle beteiligten Werkzeuge diese Informationen in derselben Weise im CAEX-Dokument speichern. Daher kann das Dokument Informationen über mehrere Werkzeuge einer Datenaustausch-Werkzeugkette enthalten.

ANMERKUNG Ein Werkzeug kann die Erstellereinformation anderer Werkzeuge entfernen. Dies kann jedoch den iterativen Datenaustausch mit den anderen Werkzeugen behindern; daher wird davon abgeraten, die Erstellereinformation anderer Werkzeuge zu entfernen.

- Diese Information muss als Bestandteil der CAEX-Zusatzinformation (AdditionalInformation) des Wurzelobjekts des CAEX-Dokuments gespeichert werden.
- Der AdditionalInformation-Block ist mit dem Namen „WriterHeader“ zu versehen.
- Die Metainformationen müssen folgende Angaben enthalten:
 - Name des exportierenden Programms, des Erstellerwerkzeugs;
 - ID des Erstellerwerkzeugs, diese muss unverändert bleiben;
 - Lieferant des Erstellerwerkzeugs;
 - URL des Erstellerwerkzeugs;
 - Produktversion des Erstellerwerkzeugs;
 - Produktversionsnummer des Erstellerwerkzeugs;
 - Datum der letzten Änderung durch den Ersteller;
 - Projekttitel des Quellenprojekts;
 - Projektkennung des Quellenprojekts.
- Der Inhalt der Metainformationen muss durch das Erstellerwerkzeug festgelegt werden und dem Typ xs:string entsprechen.
- Die geforderten Informationen sind unter Verwendung der in Tabelle 2 angegebenen Attribute zu speichern:

Tabelle 2 – Metainformationen über das AutomationML-Quell-Werkzeug

XML-Tag-Name	Typ	Stufe	Beispiel
WriterName	xs:string	Verbindlich vorgeschrieben	„ToolX AutomationML Exporter“
WriterID	xs:string	Verbindlich vorgeschrieben	„ToolXToAutomationML123“
WriterVendor	xs:string	Verbindlich vorgeschrieben	„ToolX Vendor“
WriterVendorURL	xs:string	Verbindlich vorgeschrieben	„http://www.ToolX-Vendor.org“
WriterVersion	xs:string	Verbindlich vorgeschrieben	„0.2“
WriterRelease	xs:string	Verbindlich vorgeschrieben	„123 prealpha“
LastWritingDateTime	xs:DateTime	Verbindlich vorgeschrieben	„2011-05-25T09:30:47“
WriterProjectTitle	xs:string	Wahlfrei	„eCarproduction“
WriterProjectID	xs:string	Wahlfrei	„eCarproduction_LinePLC.prj“

Für die XML-Darstellung der Metainformation gelten die folgenden Festlegungen:

- Das Element „WriterHeader“ muss ein XML-Kind-Element des CAEX-Elements „AdditionalInformation“ des CAEX-Wurzelements sein.
- Jede der in Tabelle 2 aufgeführten Metainformationen muss als XML-Kind-Element des „WriterHeaders“ beschrieben werden.
- Mehrere Metainformationen mit demselben Namen dürfen nicht im selben „WriterHeader“-Element enthalten sein.
- Die Reihenfolge der Metainformationen muss Tabelle 2 entsprechen.

Bild 3 zeigt den geforderten XML-Text anhand eines Beispiels.

```

<AdditionalInformation>
  <WriterHeader>
    <WriterName>ToolX AutomationML Exporter</WriterName>
    <ToolWriterID>ToolXToAutomationML123</ToolWriterID>
    <WriterVendor>ToolX Vendor</WriterVendor>
    <WriterVendorURL>http://www.ToolX-Vendor.org</WriterVendorURL>
    <WriterVersion>0.1</WriterVersion>
    <WriterRelease>123 prealpha</WriterRelease>
    <LastWritingDateTime>2011-05-25T09:30:47</LastWritingDateTime>
    <WriterProjectTitle>eCarproduction</WriterProjectTitle>
    <WriterProjectID>eCarproduction_LinePLC.prj</WriterProjectID>
  </WriterHeader>
</AdditionalInformation>

```

Bild 3 – XML-Text der AutomationML-Information zum Quellwerkzeug

5.5 Objektkennung

AutomationML folgt dem objektorientierten Paradigma. Alle Planungsinformationen werden entweder als Objekt modelliert oder gehören zu einem Objekt. In einer heterogenen Werkzeug-Landschaft verwenden unterschiedliche Planungswerkzeuge jedoch unterschiedliche Konzepte für die Identifikation von Objekten, z. B. einen eindeutigen Namen, einen eindeutigen Bezeichner oder einen eindeutigen Pfad. Einige Werkzeuge gestatten eine oder mehrere Änderungen der Bezeichner während der Lebenszeit, während andere dies nicht zulassen. AutomationML ermöglicht den Datenaustausch zwischen unterschiedlichen Planungswerkzeugen mittels solch individueller Objektkennungskonzepte. Aufgrund der beschriebenen Eigenschaften hebt AutomationML diese Möglichkeit zur Auswahl auf und definiert ein verbindlich vorgeschriebenes Objektkennungskonzept.

Im Hinblick auf die Objektidentifikation gelten die folgenden Festlegungen:

- Nach IEC 62424:2008 müssen AutomationML-Klassen (Rollenklassen (RoleClasses), Schnittstellenklassen (InterfaceClasses) und Anlagenkomponentenklassen (SystemUnitClasses)) durch ihre CAEX-Auszeichnung „Name“, d. h. den „Name“-Tag identifiziert werden.
- Dieser Name muss innerhalb der Hierarchiestufe der entsprechenden Bibliothek über die gesamte Lebenszeit der Klasse eindeutig sein.
- Nach IEC 62424:2008 muss die Referenzierung von Klassen über vollständige Pfade unter Verwendung der entsprechenden Pfadtrennzeichen erfolgen.
- Alle AutomationML-Objektinstanzen (interne CAEX-Elemente) müssen durch ihren CAEX-Tag „ID“ (Kennung) identifiziert werden. Dieser Bezeichner muss ein UUID nach ISO/IEC 9834-8 sein.

ANMERKUNG 1 Eine mögliche Implementierung des UUID ist der GUID.

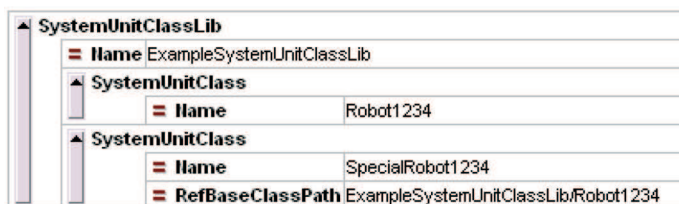
ANMERKUNG 2 Im Gegensatz zu AutomationML ist der „ID“-Tag in IEC 62424:2008 nicht verbindlich vorgeschrieben.

ANMERKUNG 3 In der vorliegenden Norm werden UUID in Kurzform dargestellt, z. B. als „GUID1“, „GUID2“ usw. Dies dient der Lesbarkeit und steht für eine reale UUID-Zeichenfolge.

- Nach seiner Erstellung darf sich dieser UUID während der gesamten Lebensdauer des betreffenden Objekts innerhalb aller beteiligten Werkzeuge nicht mehr ändern.
- Referenzierende Instanzen müssen den „ID“-Wert verwenden.
- Die Referenzierung von CAEX-Schnittstellen oder CAEX-Attributen muss mit Hilfe von UUID anstelle von Pfaden, ergänzt durch den Unterpfad zur Schnittstelle oder zum Attribut erfolgen.

ANMERKUNG Der CAEX-„Name“-Tag ist ein Anzeigenname; er ist nur informativ und kann sich mit der Zeit oder je nach Werkzeug ändern.

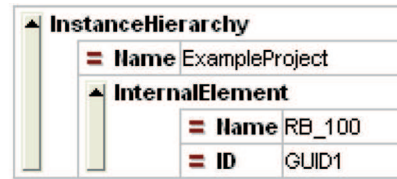
Bild 4 zeigt ein Beispiel für eine Anlagenkomponentenklassenbibliothek (SystemUnitClassLib) mit der Anlagenkomponentenklasse „Robot1234“ und einer weiteren Anlagenkomponentenklasse „SpecialRobot1234“, die von „Robot1234“ abgeleitet ist.



```
<SystemUnitClassLib Name="ExampleSystemUnitClassLib">
  <SystemUnitClass Name="Robot1234"/>
  <SystemUnitClass Name="SpecialRobot1234" RefBaseClassPath="ExampleSystemUnitClassLib/Robot1234"/>
</SystemUnitClassLib>
```

Bild 4 – Beispiel für die Objektkennung einer AutomationML-Klasse

Bild 5 zeigt ein Beispiel für eine Instanzhierarchie (InstanceHierarchy) mit einem Objekt „RB_100“, das eine eindeutige Kennung (ID) hat, die durch die Zeichenfolge „GUID1“ dargestellt wird.



```
<InstanceHierarchy Name="ExampleProject">
  ...
  <InternalElement Name="RB_100" ID="GUID1"/>
</InstanceHierarchy>
```

Bild 5 – Beispiel für die Objektkennung einer AutomationML-Objektinstanz

5.6 Spezifikation der AutomationML-Beziehungen

5.6.1 Einleitung

Die Ausrichtung auf Objekte macht es notwendig, Mechanismen festzulegen, mit deren Hilfe sich Objekte in Beziehung zueinander setzen lassen. IEC 62714 unterscheidet zwei Mechanismen für die Speicherung dieser Informationen: Verweisungen und Beziehungen. Dieser Abschnitt behandelt die Beziehungen, während die Verweisungen in 5.7 behandelt werden. Ein informativer Überblick über die Beziehungen und Verweisungen findet sich in Anhang A.1.5.

5.6.2 Eltern-Kind-Beziehungen zwischen AutomationML-Objekten

Eltern-Kind-Beziehungen zwischen Objektinstanzen werden zur Darstellung von hierarchischen Objektstrukturen verwendet und beschreiben eine „bestehen-aus“-Beziehung (en: consist-of-relation).

Im Hinblick auf Eltern-Kind-Beziehungen zwischen AutomationML-Objekten gelten die folgenden Festlegungen:

- Die Speicherung von Hierarchien in AutomationML muss nach IEC 62424:2008 erfolgen.

ANMERKUNG Neben einfachen Hierarchien können nach IEC 62424:2008 auch überkreuzte Hierarchien (Objektnetze) gespeichert werden.

Bild 6 zeigt ein Beispiel für eine Objekthierarchie und deren Speicherung in AutomationML.

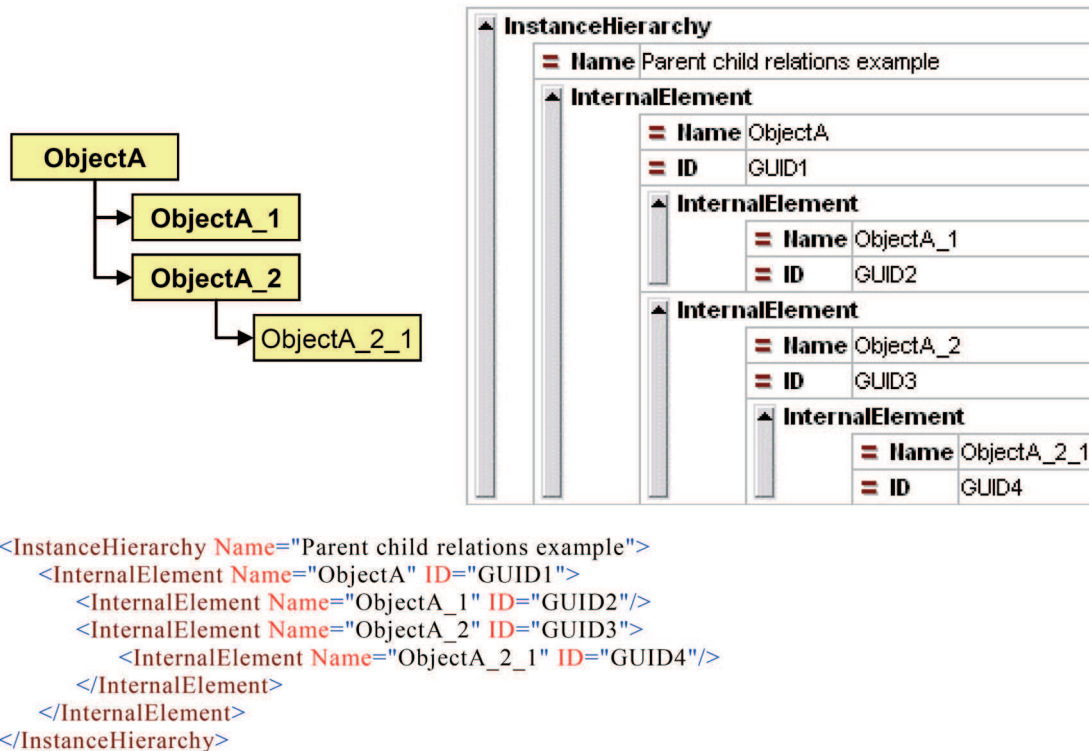


Bild 6 – Beispiel einer Eltern-Kind-Beziehung zwischen AutomationML-Objekten

5.6.3 Eltern-Kind-Beziehungen zwischen AutomationML-Klassen

Im Hinblick auf Eltern-Kind-Beziehungen zwischen AutomationML-Klassen gilt die folgende Festlegung:

- Nach IEC 62424:2008 darf eine Eltern-Kind-Beziehung zwischen CAEX-Klassen keine Vererbungsbeziehung, sondern nur eine benutzerdefinierte hierarchische Struktur ohne weitere Semantik darstellen.

Bild 7 zeigt ein Beispiel einer Eltern-Kind-Beziehung zwischen den Klassen „ParentClass“ (Elternklasse) und „ChildClass“ (Kindklasse).

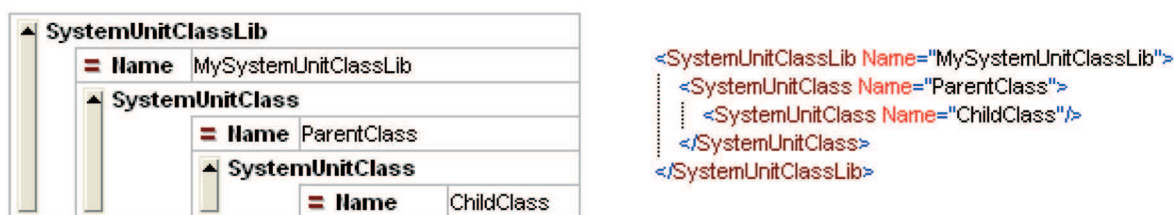


Bild 7 – Beispiel einer Eltern-Kind-Beziehung zwischen Klassen

5.6.4 Vererbungsbeziehungen

Im Hinblick auf Vererbungsbeziehungen gelten die folgenden Festlegungen:

- Vererbungen zwischen Klassen müssen nach IEC 62424:2008 definiert werden.
- Falls Vererbung erforderlich ist, muss die Elternklasse unter Verwendung des CAEX-Tags „RefBaseClassPath“ (Referenz-Basisklassenpfad) festgelegt werden, der den vollständigen Pfad der Klasse in Übereinstimmung mit IEC 62424:2008 umfasst.
- Wenn sich die gewünschte Elternklasse eine Hierarchieebene über der Kindebene befindet, kann die Elternklasse durch Speicherung des Namens der Elternklasse im CAEX-Tag „RefBaseClassPath“ festgelegt werden, ohne den vollständigen Pfad anzugeben.

Bild 8 zeigt ein Beispiel für die Klasse „Robot1234“ und eine weitere Klasse „SpecialRobot1234“, die von „Robot1234“ erbt. Zusätzlich zu diesem Beispiel kann der CAEX-Tag „RefBaseClassPath“ sowohl „InheritanceExampleLib/Robot1234“ als auch „Robot1234“ sein, da sich die Elternklasse eine Hierarchieebene über der Klasse „SpecialRobot1234“ befindet.

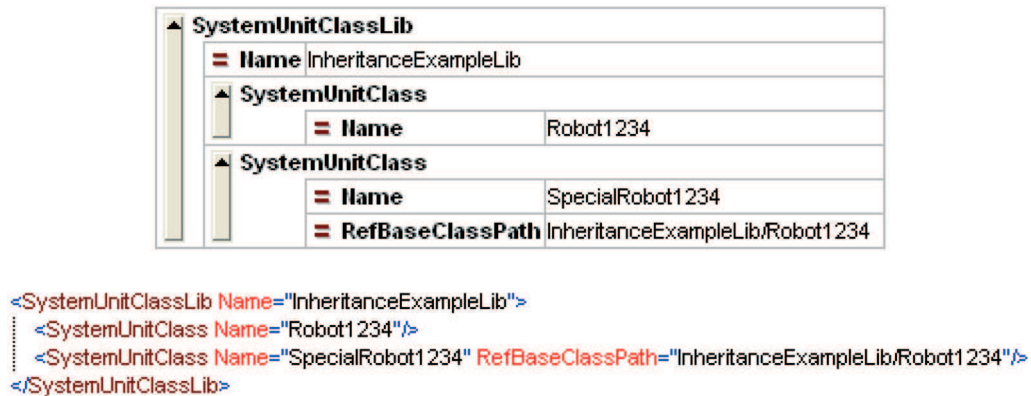


Bild 8 – Beispiel für eine Vererbungsbeziehung zwischen zwei Klassen

5.6.5 Klasse-Instanz-Beziehungen

Instanzen stellen einzelne Objekte dar und werden durch eine eindeutige Menge von Bezeichnern und Parametern beschrieben. Es gelten die folgenden Festlegungen:

- Ein AutomationML-Objekt ist als internes CAEX-Element (CAEX InternalElement) als Teil der CAEX-Instanzhierarchie oder einer Anlagenkomponentenklasse (SystemUnitClass) zu modellieren.
- Ein AutomationML-Objekt kann eine einelementige Menge ohne jede Beziehung zu irgendeiner Anlagenkomponentenklasse sein.

ANMERKUNG 1 Ein AutomationML-Objekt hat jedoch eine Beziehung zu einer AutomationML-Standardrollenklasse.

ANMERKUNG 2 Instanzen ohne jede Beziehung zur AutomationML-Basisrolle (AutomationMLBaseRole) sind zulässig, es handelt sich dabei aber um benutzerdefinierte Objekte. Es sind dies keine AutomationML-Objekte.

- Hat ein AutomationML-Objekt eine Klasse-Instanz-Beziehung zu einer Anlagenkomponentenklasse, so muss es als eine Kopie dieser Anlagenkomponentenklasse erstellt werden, was auch die interne Architektur der Klasse und alle ererbten Informationen einschließt.

ANMERKUNG Somit dient eine Anlagenkomponentenklasse als Schablone (Template). Änderungen an der Anlagenkomponentenklasse spiegeln sich nicht automatisch in den entsprechenden Automatisierungsobjekten wider. Außerdem kann das Automatisierungsobjekt ohne die Klasseninformationen übermittelt werden, es enthält selbst alle Informationen dazu, was wozu gehört.

- Die Erweiterung oder Reduktion von Instanzdaten im Vergleich mit der Quellklasse ist zulässig.

ANMERKUNG Die Quellklasse hat den Zweck, als geeigneter Ausgangspunkt für das Instanzmodell zu dienen.

- Die kopierte Quellklasse muss für die weitere Verwendung im CAEX-Tag „RefBaseSystemUnitPath“ (Referenz-Basisanlagenkomponentenpfad) der Instanz angezeigt werden. Dieser Tag muss den vollständigen Pfad und den Namen der Quellklasse umfassen.

ANMERKUNG Wenn sich die Quellklasse einer Instanz ändert, hat dies keine Änderung der Instanz zur Folge. Die Aktualisierung von Instanzen ist eine mögliche Funktionalität des Werkzeugs, die jedoch außerhalb des Anwendungsbereichs der vorliegenden Norm liegt.

- Änderungen an einer Quellklasse sollten zu einer neuen Version der Klasse mit einem anderen Namen führen. Innerhalb der neuen Klasse sollte der vollständige Pfad der alten Version der Klasse im CAEX-Tag „OldVersion“ (alte Version) gespeichert werden.

ANMERKUNG Diese Festlegung unterstützt die Verfolgung von Änderungen über unterschiedliche Versionen einer Klasse.

- Vererbungen zwischen Anlagenkomponentenklassen und Objektinstanzen sind nicht zulässig.

ANMERKUNG Eine Instanz kann nur eine Kopie ihrer Klasse sein. Hierbei handelt es sich um eine Einschränkung, die im Gegensatz zu IEC 62424:2008 steht. Die Vererbung zwischen Klassen und Instanzen kann Bestandteil einer zukünftigen AutomationML-Erweiterung sein.

- Die Beziehung zwischen einer Instanz und einer Rollenklasse ist nach IEC 62424:2008 durch das Attribut „RefBaseRoleClassPath“ (Referenz-Basisrollenklassenpad) anzuzeigen. Im Gegensatz zu IEC 62424:2008 ist keine Vererbung zulässig; es müssen vielmehr alle Rollenklassenspezifikationen in das entsprechende AutomationML-Objekt kopiert werden.
- Die Beziehung zwischen einer externen CAEX-Schnittstelle (CAEX ExternalInterface) und einer Schnittstellenklasse (InterfaceClass) muss in Übereinstimmung mit IEC 62424:2008 angezeigt werden. Im Gegensatz zu IEC 62424:2008 ist jedoch keine Vererbung zulässig; es müssen vielmehr alle Schnittstellenklassenspezifikationen in das entsprechende AutomationML-Objekt kopiert werden.

Bild 9 zeigt ein Beispiel einer Klasse-Instanz-Beziehung zwischen dem Objekt „ObjectA“ und einer benutzerdefinierten Anlagenkomponentenklasse „generic_Valve“ (allgemeines Ventil).

▲ InstanceHierarchy	
■ Name	ClassInstanceRelation Example
▲ InternalElement	
■ Name	ObjectA
■ ID	GUID1
■ RefBaseSystemUnitPath	mySystemUnitClassLib/generic_Valve

```
<InstanceHierarchy Name="ClassInstanceRelation Example">
  ...
  <InternalElement Name="ObjectA" ID="GUID1" RefBaseSystemUnitPath="mySystemUnitClassLib/generic_Valve"/>
</InstanceHierarchy>
```

Bild 9 – Beispiel einer Klasse-Instanz-Beziehung

Zusätzlich zu den Klasse-Instanz-Standardbeziehungsfestlegungen gilt nach dem CAEX-Spiegel-Konzept die folgende spezielle Festlegung:

- Der Tag „RefBaseSystemUnitPath“ kann nach dem Spiegel-Konzept von IEC 62424:2008 anstelle einer Anlagenkomponentenklasse (SystemUnitClass) eine andere Objektinstanz anzeigen.

5.6.6 Instanz-Instanz-Beziehungen

Instanz-Instanz-Beziehungen sind Beziehungen zwischen zwei Schnittstellen beliebiger AutomationML-Objekte.

Im Hinblick auf Instanz-Instanz-Beziehungen gelten die folgenden Festlegungen:

- Instanz-Instanz-Beziehungen müssen nach IEC 62424:2008 mittels der CAEX-InternalLink-Funktionalität gespeichert werden.
- Interne CAEX-Verknüpfungen (CAEX InternalLinks) sollten bei dem internen CAEX-Element (CAEX InternalElement) gespeichert werden, das das am weitesten zurückliegende gemeinsame Elternteil der entsprechenden miteinander verbundenen CAEX-Objekte ist.
- Instanz-Instanz-Beziehungen dürfen nur zwischen externen CAEX-Schnittstellen (CAEX ExternalInterfaces) definiert werden, die zu den entsprechenden AutomationML-Objekten gehören. Dies entspricht IEC 62424:2008.
- Die externen Schnittstellen sollten direkt oder indirekt von einer der AutomationML-Standard-Schnittstellenklassen abgeleitet werden.

ANMERKUNG Die AutomationML-Standard-Schnittstellenklassenbibliothek ist in 6.3 festgelegt. Die Schnittstellenklassen legen die Semantik der Schnittstelle und damit die Semantik der Verknüpfung fest. Eine Verknüpfung zwischen Schnittstellen ohne Verweisung auf eine Schnittstellenklasse hat keine Semantik.

- COLLADA-Dokumente können miteinander verknüpft sein. Die entsprechenden COLLADA-Schnittstellen können beliebige Elemente sein, die über einen gültigen URI (Uniform Resource Identifier, d.h. einen einheitlichen Bezeichner für Ressourcen) verfügen. Falls diese Knoten in CAEX miteinander verknüpft sein müssen, sind sie in CAEX durch Hinzufügung einer externen CAEX-Schnittstelle (CAEX External-Interface) zu dem betreffenden Objekt zu veröffentlichen. Diese externe Schnittstelle muss von der AutomationML-Standard-Schnittstellenklasse „COLLADAInterface“ oder einer ihrer Ableitungen abgeleitet werden.

ANMERKUNG Die Standard-Schnittstellenklasse „COLLADAInterface“ ist in 6.3.6 festgelegt. Einzelheiten hierzu sind Teil 3 dieser Norm zu entnehmen.

- PLCopen-XML-Dokumente können durch Verwendung entsprechender PLCopen-XML-Schnittstellen miteinander verknüpft werden. Falls PLCopen-XML-Elemente in CAEX miteinander verknüpft sein müssen, sind sie durch Hinzufügung einer externen CAEX-Schnittstelle (CAEX External-Interface) zu dem betreffenden Objekt zu veröffentlichen. Diese externe Schnittstelle muss von der AutomationML-Standard-Schnittstellenklasse „PLCopenXMLInterface“ oder einer ihrer Ableitungen abgeleitet werden.

ANMERKUNG Die Standard-Schnittstellenklasse „PLCopenXMLInterface“ ist in 6.3.7 festgelegt. Einzelheiten hierzu sind Teil 4 dieser Norm zu entnehmen.

Bild 10 a) zeigt ein Beispiel eines Roboters „Rob1“ und einer SPS „PLC1“, jeweils mit einer Signalschnittstelle, die angeschlossen sein muss. Bild 10 b) stellt dieses Beispiel als Objekthierarchie dar.

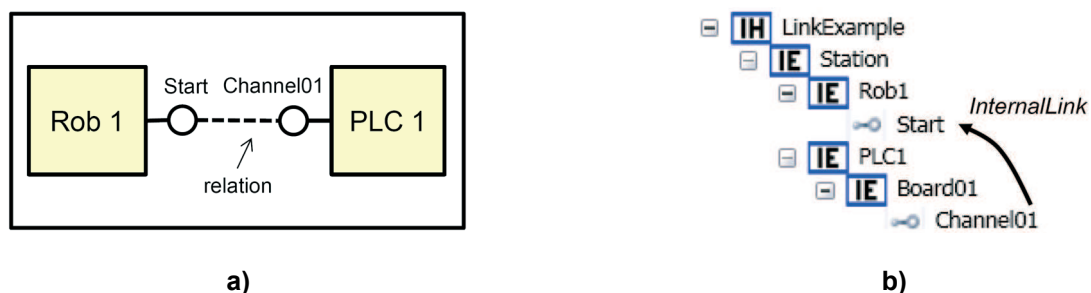
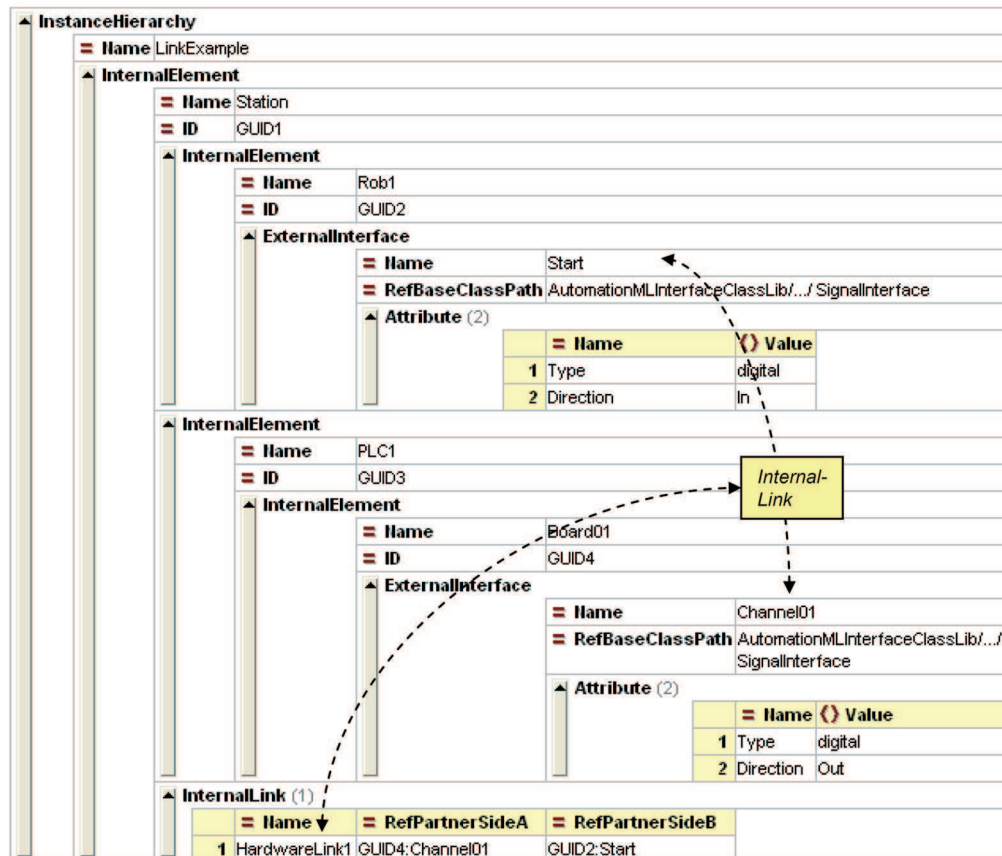


Bild 10 – Beispiel einer Beziehung als a) Blockdiagramm und b) Objektbaum

Bild 11 zeigt die AutomationML-Darstellung dieses Beispiels. Der vollständige XML-Text für die Instanzhierarchie dieses Beispiels, der sämtliche AutomationML-Objekte „Station“, „Rob1“, „PLC1“ und „Board01“ einschließlich ihrer Schnittstellen umfasst, ist nachstehend angegeben.

ANMERKUNG Die in diesem Beispiel enthaltenen Pfadzeichenfolgen wurden der besseren Lesbarkeit wegen unter Verwendung von „/.../“ verkürzt.



```

<InstanceHierarchy Name="LinkExample">
  <InternalElement Name="Station" ID="GUID1">
    <InternalElement Name="Rob1" ID="GUID2">
      <ExternalInterface Name="Start" RefBaseClassPath="AutomationMLInterfaceClassLib/.../SignalInterface">
        <Attribute Name="Type">
          <Value>digital</Value>
        </Attribute>
        <Attribute Name="Direction">
          <Value>In</Value>
        </Attribute>
      </ExternalInterface>
    </InternalElement>
    <InternalElement Name="PLC1" ID="GUID3">
      <InternalElement Name="Board01" ID="GUID4">
        <ExternalInterface Name="Channel01" RefBaseClassPath="AutomationMLInterfaceClassLib/.../SignalInterface">
          <Attribute Name="Type">
            <Value>digital</Value>
          </Attribute>
          <Attribute Name="Direction">
            <Value>Out</Value>
          </Attribute>
        </ExternalInterface>
      </InternalElement>
    </InternalElement>
    <InternalLink Name="HardwareLink1" RefPartnerSideA="GUID4:Channel01" RefPartnerSideB="GUID2:Start"/>
  </InternalElement>
</InstanceHierarchy>

```

Bild 11 – Beispielbeziehung zwischen den Objekten „PLC1“ und „Rob1“

5.7 Spezifikation von AutomationML-Dokumentverweisungen

5.7.1 Einleitung

Eine Dokumentverweisung dient zur Verknüpfung eines AutomationML-Objekts mit einem externen Dokument, das z. B. Geometrie-, Kinematik- oder Abfolgeinformationen enthalten kann. Der Verweisungsmecha-

nismus basiert auf der AutomationML-Standschnittstelle „ExternalDataConnector“ (externer Datenverbinder) oder einer ihrer Ableitungen.

5.7.2 Referenzierung von COLLADA-Dokumenten

Die Referenzierung von COLLADA-Dokumenten muss unter Zugrundelegung der AutomationML-Standard-Schnittstellenklasse „COLLADAInterface“ oder einer ihrer Ableitungen erfolgen. Diese Klasse ist in Abschnitt 6.3.6 festgelegt. Einzelheiten hierzu sind Teil 3 dieser Norm zu entnehmen.

5.7.3 Referenzierung von PLCopen-XML-Dokumenten

Die Referenzierung von PLCopen-XML muss unter Zugrundelegung der AutomationML-Standard-Schnittstelle „PLCopenXMLInterface“ oder einer ihrer Ableitungen erfolgen. Diese Klasse ist in Abschnitt 6.3.7 festgelegt. Einzelheiten hierzu sind Teil 4 dieser Norm zu entnehmen.

5.7.4 Referenzierung weiterer Dokumente

In zukünftige Erweiterungen von AutomationML können weitere Schnittstellentypen für die Referenzierung weiterer Arten von Dokumenten aufgenommen werden. Diese sind in gesonderten Teilen dieser Norm festgelegt und fallen nicht in den Anwendungsbereich des vorliegenden Teils. Für diese Erweiterungen gelten die folgenden Festlegungen:

- Falls weitere Arten von Dokumenten in AutomationML aufgenommen werden müssen, sind sie mit zusätzlichen Schnittstellenklassen zu modellieren.
- Diese zusätzlichen Schnittstellen sind als Erweiterung der AutomationML-Schnittstellenklassenbibliothek zu modellieren und direkt oder indirekt aus der Standard-Schnittstellenklasse „ExternalDataConnector“ abzuleiten.
- Die Speicherung von Verweisungen sollte unter Verwendung derselben Standardattribute erfolgen, die von den Standard-Schnittstellenklassen zur Verfügung gestellt werden.
- Die Standard-Schnittstellenklasse „ExternalDataConnector“ darf nur für die in dieser Norm enthaltenen Arten von Dokumenten verwendet werden.

6 AutomationML-Basisbibliotheken

6.1 Einleitung

In diesem Abschnitt sind wesentliche AutomationML-Standardrollenklassen und AutomationML-Schnittstellenklassen festgelegt, die zur Modellierung von AutomationML-Kernkonzepten benötigt werden. Diese Klassen sind Bestandteil der AutomationML-Standardbibliothek. Alle beschriebenen Attribute sind Teil der Standardbibliothek und dürfen, wenn sie nicht benötigt werden, aus der Instanzhierarchie entfernt werden.

ANMERKUNG Domainspezifische Bibliotheken fallen in den Anwendungsbereich anderer Teile dieser Norm.

6.2 Allgemeine Festlegungen

- Alle AutomationML-Objekte müssen entweder direkt oder indirekt der Rollenklasse „AutomationMLBaseRole“ (AutomationML-Basisrolle) zugeordnet sein bzw. werden.
- Alle Schnittstellen müssen entweder direkt oder indirekt einer AutomationML-Schnittstellenklasse zugeordnet sein bzw. werden.
- Im Bedarfsfall sind AutomationML-Attribute zu verwenden; diese dürfen, wenn sie nicht benötigt werden, aus den AutomationML-Objekten entfernt werden.

6.3 AutomationML-Schnittstellenklassenbibliothek

Die nachstehende AutomationML-Schnittstellenklassenbibliothek (AutomationMLInterfaceClassLib) wurde nach IEC 62424:2008 modelliert. Diese Norm verwendet das CAEX-Schnittstellenkonzept. Benutzerdefinierte Erweiterungen dieser Bibliothek sind zulässig, wie in 7.3 festgelegt.

Jede Schnittstelle muss direkt oder indirekt aus einer Klasse der in der nachstehenden Tabelle 3 angegebenen AutomationML-Standard-Schnittstellenklassenbibliotheken (standard AutomationMLInterfaceClassLib) abgeleitet sein. In den Abschnitten 6.3.1 bis 6.3.9 werden die Schnittstellenklassen im Einzelnen spezifiziert.

Tabelle 3 – Schnittstellenklassen der AutomationML-Schnittstellenklassenbibliothek (AutomationMLInterfaceClassLib)

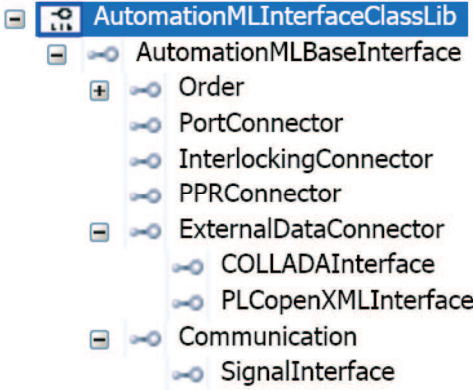
Schnittstellenklassenbibliothek	Schnittstellenklasse	Beschreibung
	AutomationMLBaseInterface	Abstrakter Schnittstellentyp
	Order	Schnittstelle zur Beschreibung von Reihenfolgen
	PortConnector	Schnittstelle zur Beschreibung von Ports
	PPRConnector	Verbinder zur Verknüpfung von Produkten, Prozessen und Ressourcen
	ExternalDataConnector	Allgemeine Verbindungsschnittstelle zu externen Daten
	COLLADAInterface	Schnittstelle zu einem COLLADA-Dokument
	PLCopenXMLInterface	Schnittstelle zu einem PLCopen-XML-Dokument
	Communication	Allgemeine Kommunikationsschnittstelle
	SignalInterface	Allgemeine Signalschnittstelle

Bild 12 zeigt eine tabellarische Ansicht und Bild 13 den XML-Text der AutomationML-Standard-Schnittstellenklassenbibliothek. Die folgenden Unterabschnitte enthalten detaillierte Informationen über die Klassen.