

DIN EN 62714-3



ICS 35.240.50

Einsprüche bis 2016-10-12

Entwurf

**Datenaustauschformat für Planungsdaten industrieller
Automatisierungssysteme –
Automation markup language –
Teil 3: Geometrie und Kinematik
(IEC 65E/497/CDV:2016);
Deutsche Fassung FprEN 62714-3:2016**

Engineering data exchange format for use in industrial automation systems engineering –
Automation markup language –
Part 3: Geometry and kinematics
(IEC 65E/497/CDV:2016);
German version FprEN 62714-3:2016

Format d'échange de données techniques pour une utilisation dans l'ingénierie des
systèmes d'automatisation industrielle –
Automation markup language –
Partie 3: Géométrie et cinématique
(IEC 65E/497/CDV:2016);
Version allemande FprEN 62714-3:2016

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2016-08-12 wird der Öffentlichkeit zur Prüfung und
Stellungnahme vorgelegt.

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

Stellungnahmen werden erbeten

- vorzugsweise online im Norm-Entwurfs-Portal des DIN unter www.din.de/go/entwuerfe bzw. für Norm-Entwürfe der DKE auch im Norm-Entwurfs-Portal der DKE unter www.entwuerfe.normenbibliothek.de, sofern dort wiedergegeben;
- oder als Datei per E-Mail an dke@vde.com möglichst in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter www.din.de/go/stellungnahmen-norm-entwuerfe oder für Stellungnahmen zu Norm-Entwürfen der DKE unter www.dke.de/stellungnahme abgerufen werden;
- oder in Papierform an die DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE, 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 184 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE

Nationales Vorwort

Die Deutsche Fassung des europäischen Dokuments FprEN 62714-3:2016 „Datenaustauschformat für Planungsdaten industrieller Automatisierungssysteme – Automation markup language – Teil 3: Geometrie und Kinematik„ (Entwurf in der Umfrage) ist unverändert in diesen Norm-Entwurf übernommen worden.

Die Internationale Elektrotechnische Kommission (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. Dem entsprechend ist das internationale Dokument IEC 65E/497/CDV:2016 „Engineering data exchange format for use in industrial automation systems engineering – Automation markup language – Part 3: Geometry and kinematics„ unverändert in den Entwurf FprEN 62714-3:2016 übernommen worden.

Da die Deutsche Fassung noch nicht endgültig mit der Englischen und Französischen Fassung abgeglichen ist, ist die englische Originalfassung des IEC-CDV entsprechend der diesbezüglich durch die IEC erteilten Erlaubnis beigefügt. Die Nutzungsbedingungen für den deutschen Text des Norm-Entwurfes gelten gleichermaßen auch für den englischen IEC-Text.

Das internationale Dokument wurde vom SC 65E „Devices and integration in enterprise systems“ der Internationalen Elektrotechnischen Kommission (IEC) erarbeitet.

Bei der Abstimmung zu dem Europäischen Schluss-Entwurf bei CENELEC und dem Internationalen Schluss-Entwurf bei IEC [Final Draft International Standard (FDIS)] sind jeweils nur „JA/NEIN“-Entscheidungen möglich, wobei „NEIN“-Entscheidungen fundiert begründet werden müssen. Dokumente, die bei CENELEC als Europäische Norm angenommen und ratifiziert werden, sind unverändert als Deutsche Normen zu übernehmen.

Für dieses Dokument ist das nationale Arbeitsgremium K 941 „Engineering“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE (www.dke.de) zuständig.

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.

Der Zusammenhang der zitierten Dokumente mit den entsprechenden Deutschen Dokumenten 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.

Es wird auf die Möglichkeit hingewiesen, dass einige Elemente dieses Dokuments Patentrechte berühren können. ISO [und/oder] IEC sind nicht dafür verantwortlich, einige oder alle diesbezüglichen Patentrechte zu identifizieren.

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

**Datenaustauschformat für Planungsdaten industrieller Automatisierungssysteme –
Automation markup language –
Teil 3: Geometrie und Kinematik**

Inhalt

	Seite
Einleitung	6
1 Anwendungsbereich	9
2 Normative Verweisungen	9
3 Begriffe und Abkürzungen	9
3.1 Begriffe	9
3.2 Abkürzungen	9
4 Konformität	9
5 Erweiterungen von AML-Bibliotheken für Geometrie und Kinematik	10
5.1 Allgemeines	10
5.2 AutomationMLBaseRoleClassLib	10
5.3 AutomationMLInterfaceClassLib	10
6 Frame-Attribute	11
7 Integration von COLLADA-Dokumenten	12
8 Anbindung von zwei AML-Objekten	13
9 Meta-Informationen über das COLLADA-Quellwerkzeug	14
Anhang A (informativ) Verfahren der Referenzierung für geometrische/kinematische Festlegungen	16
A.1 Integration eines allgemeinen COLLADA-Dokuments mit expliziter Referenzierung	16
A.1.1 Festlegung des Attributs „Frame“	17
A.1.2 Struktur des COLLADA-Dokuments	19
A.1.3 Referenzierung unter Verwendung von URI und Fragmenten ohne Ziel und ID	23
A.1.4 Referenzierung unter Verwendung einer URI und von Fragmenten einschließlich eines Ziels ohne ID	23
A.1.5 Referenzierung unter Verwendung eines URI ohne ein Fragment einschließlich eines Ziels und einer ID	24
A.1.6 Referenzierung unter Verwendung einer URI und von Fragmenten einschließlich eines Ziels und einer ID	25
A.1.7 Referenzierung unter Verwendung einer URI ohne ein Fragment, ein Ziel und einer ID	26
A.2 Implizite Referenzierung auf COLLADA-Elemente	28
A.2.1 Implizite Referenzierung	28
A.2.2 Implizite Referenzierung auf COLLADA-Unterdokumente	30
A.2.3 Veröffentlichung von Elementen eines COLLADA-Dokuments in CAEX	36
A.3 Anbindung zwischen Objekten in CAEX	40
Anhang B (informativ) Modellierung von Kinematik-Systemen und ihre Kombination in AML	46
B.1 Modellierung eines AML-Dokuments einer linearen Einheit in CAEX und COLLADA	46
B.1.1 Festlegung der visuellen Szene	46

	Seite
B.1.2 Festlegung des Gelenks.....	49
B.1.3 Festlegung des Kinematikmodells.....	49
B.1.4 Festlegung des gegliederten Systems	50
B.1.5 Festlegung der kinematischen Szene	52
B.1.6 Zusammenfügen der Szene	53
B.1.7 Kombination von CAEX und COLLADA in AML.....	53
B.2 Modellierung eines AML-Dokuments eines Roboters in CAEX und COLLADA	55
B.2.1 Festlegung der visuellen Szene	56
B.2.2 Festlegung von Gelenken.....	60
B.2.3 Festlegung des Kinematikmodells.....	60
B.2.4 Festlegung des gegliederten Systems	61
B.2.5 Festlegung der kinematischen Szene	66
B.2.6 Zusammenfügen der Szene	67
B.2.7 Kombination von CAEX und COLLADA in AML.....	68
B.3 Modellierung eines AML-Dokuments eines kombinierten Systems einschließlich eines Roboters und einer linearen Achse in CAEX und COLLADA	70
B.4 Modellierung eines AML-Dokuments eines am Roboter angebundene Greifers in CAEX und COLLADA.....	74
B.4.1 Festlegung der visuellen Szene	75
B.4.2 Festlegung des Kinematiksystems.....	77
B.4.3 Zusammenfügen der Szene	86
B.4.4 Kombination von CAEX und COLLADA in AML.....	87
B.5 Modellierung eines AML-Dokuments eines mit einem Greifer verbundenen Werkstücks in CAEX und COLLADA.....	91
B.5.1 Implizite obere Grenze	91
B.5.2 Festlegung des Werkstücks	93
B.5.3 Kombination von CAEX und COLLADA in AML.....	94
Anhang C (informativ) XML-Darstellung von AML-Bibliotheken.....	99
C.1 AutomationMLBaseRoleClassLib.....	99
C.2 AutomationMLInterfaceClassLib	100
Bilder	
Bild 1 – Überblick über das Engineering-Datenaustauschformat AML	7
Bild 2 – Beispiel für erforderlichen XML-Text im Fall von ISO/PAS 17506	15
Bild 3 – Beispiel für erforderlichen XML-Text im Fall von COLLADA 1.4.1.....	15
Bild A.1 – Entscheidungsbaum für verschiedene Referenzierungsverfahren	16
Bild A.2 – Zwei Frames dargestellt in der „InstanceHierarchy“ eines AML-Dokuments.....	17
Bild A.3 – XML-Darstellung des AML-Dokuments.....	18
Bild A.4 – Verschiebung und raumfeste Drehung	19
Bild A.5 – Im vorliegenden Beispiel verwendete COLLADA-Szene	20

	Seite
Bild A.6 – Struktur und Referenzierungen.....	20
Bild A.7 – Inhalt des COLLADA-Dokuments „cube.dae“.....	21
Bild A.8 – Inhalt des COLLADA-Dokuments „red_blue_cubes.dae“.....	22
Bild A.9 – „RedCube“ – Hierarchie des AML-Dokuments.....	23
Bild A.10 – XML-Darstellung des AML-Dokuments.....	23
Bild A.11 – Referenzierung auf den roten Würfel durch ID.....	23
Bild A.12 – „BlueCube“ – Hierarchie des AML-Dokuments.....	24
Bild A.13 – XML-Darstellung des AML-Dokuments.....	24
Bild A.14 – Referenzierung auf den blauen Würfel.....	24
Bild A.15 – „BlueCube“ – Hierarchie des AML-Dokuments.....	25
Bild A.16 – XML-Darstellung des AML-Dokuments.....	25
Bild A.17 – Referenzierung auf den blauen Würfel beginnend mit dem Element „subpart“.....	25
Bild A.18 – Hierarchie des AML-Dokuments.....	26
Bild A.19 – XML-Darstellung des AML-Dokuments.....	26
Bild A.20 – Referenzierung auf den blauen Würfel.....	26
Bild A.21 – Hierarchie des AML-Dokuments.....	27
Bild A.22 – XML-Darstellung des AML-Dokuments.....	27
Bild A.23 – Referenzierung auf die vollständige COLLADA-Szene.....	28
Bild A.24 – Implizite Referenzierung: Hierarchie des AML-Dokuments.....	29
Bild A.25 – XML-Darstellung des AML-Dokuments.....	30
Bild A.26 – Struktur und Beziehungen von referenzierten COLLADA-Unterdokumenten.....	31
Bild A.27 – Inhalt des abgeänderten COLLADA-Dokuments „red_blue_cubes.dae“.....	32
Bild A.28 – Inhalt des COLLADA-Dokuments „red_cube.dae“.....	33
Bild A.29 – Inhalt des COLLADA-Dokuments „blue_cube.dae“.....	34
Bild A.30 – Implizite Referenzierung: Hierarchie des AML-Dokuments.....	35
Bild A.31 – XML-Darstellung eines AML-Dokuments.....	36
Bild A.32 – Abgeänderte COLLADA-Szene mit zusätzlichem Knoten.....	37
Bild A.33 – Zusätzliches Frame-Element im COLLADA-Dokument.....	37
Bild A.34 – Veröffentlichung von Frames: Hierarchie des AML-Dokuments.....	38
Bild A.35 – XML-Darstellung des AML-Dokuments.....	39
Bild A.36 – Struktur für die Anbindung zwischen Objekten in CAEX.....	40
Bild A.37 – Visualisierung des gelben Würfels mit zusätzlichem Frame.....	41
Bild A.38 – COLLADA-Dokument eines gelben Würfels mit zusätzlichem Frame.....	42
Bild A.39 – Hierarchie des AML-Dokuments.....	43
Bild A.40 – XML-Darstellung des AML-Dokuments.....	44
Bild A.41 – Anbindung zwischen geometrischen AML-Objekten.....	45
Bild A.42 – XML-Darstellung des AML-Dokuments.....	45
Bild B.1 – Veranschaulichung der linearen Einheit.....	46

	Seite
Bild B.2 – Festlegung der visuellen Szene	48
Bild B.3 – Festlegung des Gelenks.....	49
Bild B.4 – Festlegung des Kinematikmodells	49
Bild B.5 – Festlegung der Bibliothek des gegliederten Systems	50
Bild B.6 – Festlegung des gegliederten Systems Kinematik	51
Bild B.7 – Festlegung des gegliederten Systems Bewegung	52
Bild B.8 – Festlegung der kinematischen Szene	53
Bild B.9 – Instanziierung der kinematischen Szene	53
Bild B.10 – Hierarchie des AML-Dokuments	54
Bild B.11 – XML-Darstellung des AML-Dokuments.....	55
Bild B.12 – Visualisierung des Roboters	56
Bild B.13 – Festlegung der visuellen Szene	59
Bild B.14 – Festlegung der Gelenke	60
Bild B.15 – Festlegung des Kinematikmodells	61
Bild B.16 – Festlegung der Bibliothek des gegliederten Systems	61
Bild B.17 – Festlegung des gegliederten Systems Kinematik	63
Bild B.18 – Festlegung des gegliederten Systems Bewegung	66
Bild B.19 – Festlegung der kinematischen Szene	67
Bild B.20 – Instanziierung der kinematischen Szene	68
Bild B.21 – Hierarchie des AML-Dokuments	69
Bild B.22 – XML-Darstellung des AML-Dokuments.....	70
Bild B.23 – Struktur der Referenzierung auf COLLADA-Unterdokumente	71
Bild B.24 – Hierarchie des AML-Dokuments	72
Bild B.25 – XML-Darstellung des AML-Dokuments.....	73
Bild B.26 – XML-Darstellung des AML-Dokuments.....	74
Bild B.27 – Visualisierung des an die lineare Einheit angebundenen Roboters	74
Bild B.28 – Hierarchie des COLLADA-Dokuments.....	75
Bild B.29 – Visualisierung des Greifers	75
Bild B.30 – Festlegung der visuellen Szene	77
Bild B.31 – Festlegung der Kinematik.....	78
Bild B.32 – Festlegung von Gelenken	78
Bild B.33 – Festlegung eines Kinematikmodells.....	79
Bild B.34 – Festlegung des gegliederten Systems	80
Bild B.35 – Festlegung des gegliederten Systems	81
Bild B.36 – Festlegung der kinematischen Szene	82
Bild B.37 – Festlegung der Abhängigkeit eines Gelenks unter Verwendung von MathML	82
Bild B.38 – XML-Darstellung des COLLADA-Dokuments „gripper_kinematics.dae“	86
Bild B.39 – XML-Darstellung des COLLADA-Dokuments „gripper.dae“	87

	Seite
Bild B.40 – Hierarchie des AML-Dokuments	88
Bild B.41 – XML-Darstellung des AML-Dokuments	90
Bild B.42 – XML-Darstellung des AML-Dokuments	91
Bild B.43 – Veranschaulichung des Roboters an der linearen Einheit und angebundenem Greifer	91
Bild B.44 – Beispiel für eine implizite obere Grenze	92
Bild B.45 – Struktur für Anbindungen zwischen Objekten in CAEX.....	93
Bild B.46 – Visualisierung des Werkstücks mit zusätzlichem Frame	93
Bild B.47 – COLLADA-Dokument des Werkstücks mit zusätzlichem Frame.....	94
Bild B.48 – Hierarchie des AML-Dokuments	95
Bild B.49 – XML-Darstellung des AML-Dokuments	98
Bild B.50 – Anbindung zwischen geometrischen AML-Objekten	98
Bild B.51 – XML-Darstellung des AML-Dokuments	98
Bild C.52 – XML-Darstellung von AML-Bibliotheken „AutomationMLBaseRoleClassLib“	99
Bild C.53 – XML-Darstellung von AML-Bibliotheken „AutomationMLInterfaceClassLib“	100
Tabellen	
Tabelle 1 – Abkürzungen.....	9
Tabelle 2 – RoleClass „Frame“.....	10
Tabelle 3 – InterfaceClass „COLLADAInterface“	10
Tabelle 4 – InterfaceClass AttachmentInterface	11
Tabelle 5 – Attribut „Frame“	11
Tabelle 6 – Unterattribute des Attributs „Frame“	12
Tabelle 7 – Regeln für das Auflösen eines Dokuments und eines Eintrittspunktes.....	13
Tabelle 8 – Meta-Informationen über das COLLADA-Quellwerkzeug	15

Einleitung

Das in IEC 62714 festgelegte Datenaustauschformat (Automation Markup Language, AML) ist ein auf XML-Schemata beruhendes Datenformat und wurde entwickelt, um den Datenaustausch zwischen Planungswerkzeugen (en: engineering tool) in einer heterogenen Planungswerkzeug-Umgebung zu unterstützen. IEC 62714-1 gibt einen Überblick zu diesem Format.

Das Ziel von AML besteht darin, Planungswerkzeuge aus den bestehenden heterogenen Planungswerkzeug-Umgebungen in ihren Fachrichtungen, z. B. Maschinen- und Anlagenbau, elektrische Konstruktion, Verfahrenstechnik, Prozessleittechnik, HMI-Entwicklung, SPS-Programmierung, Roboterprogrammierung usw. miteinander zu verbinden.

AML speichert Planungsinformationen entsprechend dem objektorientierten Paradigma und ermöglicht die Modellierung physikalischer und logischer Anlagenkomponenten als Datenobjekte, in denen unterschiedliche Aspekte gekapselt sind. Ein Objekt kann aus Unterobjekten bestehen und kann zugleich selbst Teil einer größeren Komposition bzw. Aggregation sein. Typische Objekte in der Anlagenautomatisierung umfassen Informationen zur Topologie, Geometrie, Kinematik und Logik, wobei die Logik Informationen über Abfolge, Verhalten und Steuerung umfasst.

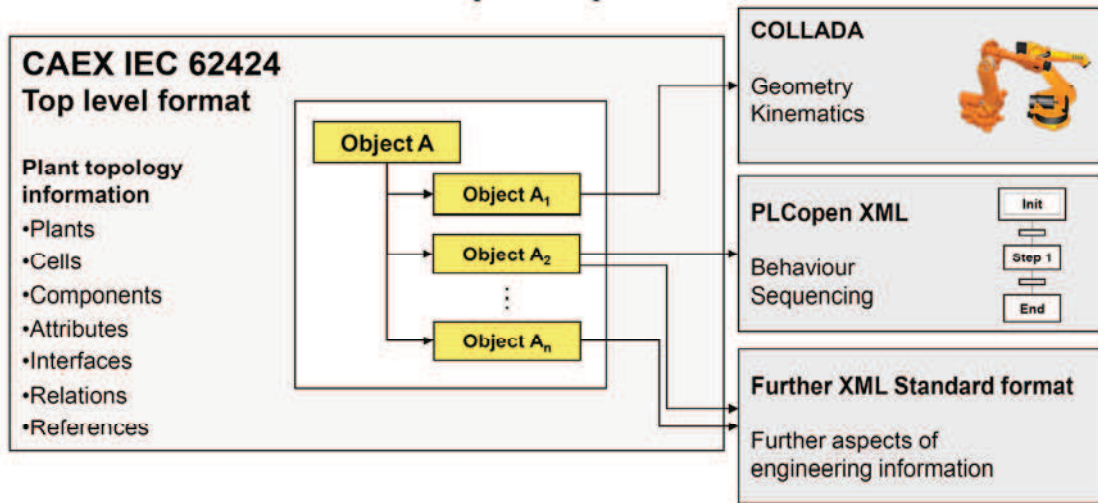
AML kombiniert bestehende Industriedatenformate, die zur Speicherung und zum Austausch von verschiedenen Aspekten der Planungsinformationen entwickelt wurden. Diese Datenformate werden innerhalb ihrer eigenen Spezifikationen „so wie sie sind“ (en: „as is“) verwendet, und es wurden aus ihnen keine speziellen AML-Zweige entwickelt.

Der Kern von AML ist das Top-Level-Datenformat CAEX (Datenformat der höchsten Ebene), welches die verschiedenen Datenformate miteinander verbindet. Zu diesem Zweck verfügt AML über eine inhärent verteilte Dokumentenarchitektur.

Bild 1 veranschaulicht die Grundarchitektur von AML und die Verteilung der Topologie-, Geometrie-, Kinematik- und Logikinformationen.

Automation Markup Language

Engineering data



Engineering Data	Planungsdaten
Top level	Top-Level-Format
Plant topology information	Anlagen-Topologie-Informationen
Plants	Anlagen
Cells	Einheiten
Components	Komponenten
Attributes	Attribute
Interfaces	Schnittstellen
Relations	Beziehungen
References	Referenzen
Object	Objekt
Geometry	Geometrie
Kinematics	Kinematik
Behaviour	Verhalten
Sequencing	Abläufe
Further XML	Weiteres XML-Standard-Format
Further aspects	Weitere Aspekte von Planungsdaten

Bild 1 – Überblick über das Engineering-Datenaustauschformat AML

Aufgrund der verschiedenen Aspekte von AML besteht IEC 62714 aus verschiedenen Teilen, die sich auf unterschiedliche Aspekte von AML konzentrieren:

- IEC 62714-1: Architektur und allgemeine Anforderungen
Dieser Teil legt die grundlegende Architektur von AML, die Modellierung von Planungsdaten, Klassen, Instanzen, Beziehungen, Referenzen, Hierarchien, grundlegende AML-Bibliotheken sowie erweiterte AML-Konzepte fest.
- IEC 62714-2: Rollenklassenbibliotheken
Dieser Teil spezifiziert zusätzliche AML-Bibliotheken.
- IEC 62714-3: Geometrie und Kinematik
Dieser Teil spezifiziert die Modellierung von Geometrie- und Kinematikinformationen.
- IEC 62714-4: Logik
Dieser Teil spezifiziert die Modellierung von Logik, Abläufen, Verhalten und steuerungsbegleite Infor-mationen.

Weitere Teile dürfen zukünftig hinzukommen, um die Verknüpfung weiterer Datenstandards mit AML zu ermöglichen.

Abschnitt 5 legt die geometriebezogenen Erweiterungen der Rollenklassenbibliotheken fest.

Abschnitt 6 legt die Frame-Attribute fest, welche verwendet werden können, um die geometrische Position eines (einer) `InternalElement` (internes Element), `InstanceHierarchy` (Instanzhierarchie), `SystemUnitClass` (Systemeinheitenklasse) oder `SystemUnitClassLibrary` (Systemeinheitenklassenbibliothek) in Bezug auf ein anderes CAEX-Objekt abzubilden.

Abschnitt 7 enthält eine normative Festlegung bezüglich der Referenzierung auf COLLADA-Dokumente.

Abschnitt 8 legt normative Bestimmungen für das Einfügen von zwei geometrischen AML-Objekten fest.

Abschnitt 9 legt fest, wie Meta-Informationen bezüglich des Open-Source-Werkzeugs direkt in das COLLADA-Dokument zu speichern sind.

Anhang A legt die Methoden der Referenzierung für Geometrie- und Kinematik-Modelle fest.

Anhang B stellt ein Beispiel für die Modellierung von Kinematik-Systemen und ihrer Kombination in AML bereit.

Anhang C enthält eine informative XML-Darstellung der in diesem Teil der IEC 62714 festgelegten Bibliotheken.

1 Anwendungsbereich

Dieser Teil von IEC 62714 legt die Integration von Geometrie- und Kinematik-Informationen für den Austausch zwischen Planungswerkzeugen in Bereichen der Anlagenautomatisierung mittels AML fest.

Dieser Teil von IEC 62714 legt keine Einzelheiten für das Verfahren des Datenaustausches fest und beschreibt keine Implementierungsanforderungen für Export-/Importwerkzeuge.

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 62714-1:2014, *Engineering data exchange format for use in industrial automation systems engineering – Automation markup language – Part 1: Architecture and general requirements*

ISO/PAS 17506, *Industrial automation systems and integration – COLLADA digital asset schema specification for 3D visualization of industrial data*

COLLADA 1.4.1: March 2008 COLLADA – Digital Asset Schema Release 1.4.1 (verfügbar unter <http://www.khronos.org/files/collada_spec_1_4.pdf>)

Extensible Markup Language (XML) 1.0:2004, W3C Recommendation (verfügbar unter <<http://www.w3.org/TR/2004/REC-xml-20040204/>>)

3 Begriffe und Abkürzungen

3.1 Begriffe

Für die Anwendung dieses Dokuments gelten die Begriffe nach IEC 62714-1:2014 und von IEC 62714-2:2015 und die folgenden Begriffe.

SID-Pfad

Referenzierung von Typ `sidref_type` auf ein beliebiges Element innerhalb eines COLLADA-Dokuments entsprechend ISO/PAS 17506 oder COLLADA 1.4.1

3.2 Abkürzungen

Für die Anwendung dieses Dokuments gelten die Abkürzungen von IEC 62714-1:2014 und von IEC 62714-2:2015 und zusätzlich die in Tabelle 1 aufgelisteten Abkürzungen.

Tabelle 1 – Abkürzungen

SCARA	Baugruppe eines Roboterarms mit ausgewählter Vereinbarkeit (en: Selective Compliance Assembly Robot Arm)
SID	Bezeichner mit zugewiesenem Bereich (en: Scoped Identifier)

4 Konformität

Um Konformität mit dem Teil 3 der vorliegenden Norm in Bezug auf die Unterstützung von AML zu beanspruchen, müssen die Anforderungen der Abschnitte 5, 6, 7, 8 und 9 erfüllt sein. Im Anwendungsbereich von AML muss ein COLLADA-Dokument den Festlegungen von ISO/PAS 17506 oder COLLADA 1.4.1 entsprechen.

5 Erweiterungen von AML-Bibliotheken für Geometrie und Kinematik

5.1 Allgemeines

Abschnitt 5 legt Erweiterungen von Standard-AML-RoleClasses (Rollenklassen) und Standard-AML-InterfaceClasses (Schnittstellenklassen) fest. Diese Klassen sind Bestandteil der AML-Standardbibliotheken und eine spezifische Erweiterung von IEC 62714-1 für den vorliegenden Teil von IEC 62714. Alle festgelegten Attribute sind Bestandteil der Standardbibliotheken und dürfen in der InstanceHierarchy entfernt werden, wenn sie nicht benötigt werden.

5.2 AutomationMLBaseRoleClassLib

5.2.1 RoleClass „Frame“

Die RoleClass „Frame“ (Rollenklasse Rahmen) muss entsprechend der Festlegung in Tabelle 2 verwendet werden.

Tabelle 2 – RoleClass „Frame“

Klassenname	Frame
Beschreibung	Diese Rolle bezeichnet ein kartesisches rechtwinkliges Koordinatensystem
Elternklasse	AutomationMLBaseRoleClassLib/AutomationMLBaseRole (AML-Basis-Rollenklassenbibliothek/AML-Basis-Rolle)
Pfad für Elementreferenz	AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Frame

ANMERKUNG 1 Ein AML-Objekt, welches auf eine RoleClass „Frame“ referenziert, kann verwendet werden, um Referenzkoordinatensysteme wie z. B. Anbindungspunkte festzulegen.

ANMERKUNG 2 Diese RoleClass kann geerbt werden, um Semantik hinzuzufügen. Beispiele von geerbten Frames sind Werkzeug-Frames, Basis-Frames, Prozess-Frames, Werkstück-Frame, usw.

5.3 AutomationMLInterfaceClassLib

5.3.1 InterfaceClass „COLLADAInterface“

Die InterfaceClass „COLLADAInterface“ muss entsprechend der Festlegung in Tabelle 3 verwendet werden.

Tabelle 3 – InterfaceClass „COLLADAInterface“

Klassenname	COLLADAInterface	
Beschreibung	Die InterfaceClass „COLLADAInterface“ muss verwendet werden, um auf externe COLLADA-Dokumente zu referenzieren und Schnittstellen zu veröffentlichen, welche innerhalb eines COLLADA-Dokuments festgelegt sind.	
Elternklasse	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector (AML-Schnittstellenklassenbibliothek/AML-Basis-Schnittstelle/Externer Datenverbinder)	
Pfad für Elementreferenz	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector/COLLADAInterface	
Attribute	refType (AttributeDataType = „xs:string“))	Das Attribut „refType“ (Art der Referenzierung) legt fest, ob das referenzierte COLLADA-Dokument von expliziter oder impliziter Art ist. Die zulässigen Werte sind „explicit“ (explizit) oder „implicit“ (implizit).
	target (AttributeDataType = „xs:token“)	Das Attribut „target“ (Ziel) legt den SID-Pfad eines COLLADA-Elements innerhalb des referenzierten Dokuments fest. Das Attribut ist wahlfrei.

ANMERKUNG 1 Die InterfaceClass „COLLADAInterface“ erbt zusätzlich das Attribut „refURI“ (Verweisung auf URI) von der Eltern-InterfaceClass „ExternalDataConnector“.

5.3.2 InterfaceClass „AttachmentInterface“

Die InterfaceClass „AttachmentInterface“ (Anbindungsschnittstelle) muss entsprechend der Festlegung in Tabelle 4 verwendet werden.

Tabelle 4 – InterfaceClass AttachmentInterface

Klassenname	AttachmentInterface
Beschreibung	Die InterfaceClass „AttachmentInterface“ legt eine Schnittstelle für Geometrie- und Kinematik-Links zwischen AML-Objekten mit RoleClass-Frame fest.
Elternklasse	AutomationMLInterfaceClassLib/AutomationMLBaseInterface (AML-Schnittstellenklassenbibliothek/AML-Basis-Schnittstelle)
Pfad für Elementreferenz	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/AttachmentInterface

6 Frame-Attribute

Ein InternalElement (internes Element) oder eine SystemUnitClass (Systemeinheitenklasse) erfordert möglicherweise ein Frame-Attribut, welches seine (ihre) geometrische Position in Bezug zu anderen Objekten abbildet. Dafür gelten die folgenden Bestimmungen:

- Jeder Frame muss auf einem dreidimensionalen, orthogonalen, rechtwinkligen Koordinatensystem beruhen.
- Die Elemente InstanceHierarchy und SystemUnitClassLib legen ein dreidimensionales, orthogonales, rechtwinkliges Koordinatensystem mit Standardbasis fest. Die positive Z-Achse wird als nach oben verlaufend angenommen, die positive X-Achse als nach rechts verlaufend und die Y-Achse ist als nach vorn verlaufend festgelegt.
- Sowohl die relativen Verschiebungen (en: translations) x, y und z als auch die Drehungen (en: rotations) rx, ry und rz müssen als Unterattribute in einem AML-Attribut „Frame“ festgelegt werden, welches in Tabelle 5 oder Tabelle 6 festgelegt ist. Die relativen Verschiebungen x, y, und z müssen in Bezug zum Eltern-Koordinatensystem im vorangegangenen Punkt festgelegt werden. Die Drehungen rx, ry und rz müssen in der Reihenfolge rx, ry und rz in Bezug auf die festen Achsen im Eltern-Koordinatensystem ausgeführt werden. Der Ursprung des verschobenen Frames muss seine Position (x, y, z) während der raumfesten Drehung beibehalten. Das Attribut „Frame“ muss sein enthaltenes AML-Objekt und seine Kinder beeinflussen.

ANMERKUNG 1 Dies bedeutet, dass sich der Drehpunkt dreht, während die Frame-Position unverändert bleibt. Dadurch werden doppelte Änderungen vermieden.

- Wenn das Attribut „Frame“ nicht festgelegt ist, müssen die Vorgabewerte von x, y, z, rx, ry und rz gleich 0 sein.
- Wenn das Attribut „Frame“ festgelegt ist, müssen alle Unterattribute x, y, z, rx, ry und rz aufgelistet werden.

Das Attribut „Frame“ muss als komplexes Attribut entsprechend den Festlegungen in Tabelle 5 und Tabelle 6 verwendet werden.

Tabelle 5 – Attribut „Frame“

Attribut	AttributeDataType	Beschreibung
Frame	Dieses Attribut hat kein AttributeDataType, da Attribut keinen Wert hat.	Das Attribut „Frame“ muss als Strukturelement für die Speicherung der in Tabelle 6 festgelegten Unterattribute verwendet werden.

Tabelle 6 – Unterattribute des Attributs „Frame“

Attribut	AttributeDataType	Beschreibung
x	xs:double	Das Attribut „x“ muss verwendet werden, um die relative Position, in Meter, entlang der X-Achse des Koordinatensystems festzulegen, welche vom Elternelement festgelegt wird. Der Wert des Attributs „Unit“ (Einheit) des CAEX-Elements „Attribut“ muss „m“ sein.
y	xs:double	Das Attribut „y“ muss verwendet werden, um die relative Position, in Meter, entlang der X-Achse des Koordinatensystems festzulegen, welche vom Elternelement festgelegt wird. Der Wert des Attributs „Unit“ (Einheit) des CAEX-Elements „Attribut“ muss „m“ sein.
z	xs:double	Das Attribut „z“ muss verwendet werden, um die relative Position, in Meter, entlang der X-Achse des Koordinatensystems festzulegen, welche vom Elternelement festgelegt wird. Der Wert des Attributs „Unit“ (Einheit) des CAEX-Elements „Attribut“ muss „m“ sein.
rx	xs:double	Das Attribut „rx“ muss verwendet werden, um die relative Drehung, in Grad, um die X-Achse des Koordinatensystems festzulegen, welche vom Elternelement festgelegt wird. Der Wert des Attributs „Unit“ (Einheit) des CAEX-Elements „Attribut“ muss „deg“ sein.
ry	xs:double	Das Attribut „ry“ muss verwendet werden, um die relative Drehung, in Grad, um die X-Achse des Koordinatensystems festzulegen, welche vom Elternelement festgelegt wird. Der Wert des Attributs „Unit“ (Einheit) des CAEX-Elements „Attribut“ muss „deg“ sein.
rz	xs:double	Das Attribut „rz“ muss verwendet werden, um die relative Drehung, in Grad, um die X-Achse des Koordinatensystems festzulegen, welche vom Elternelement festgelegt wird. Der Wert des Attributs „Unit“ (Einheit) des CAEX-Elements „Attribut“ muss „deg“ sein.

7 Integration von COLLADA-Dokumenten

In Bezug auf Referenzierungen auf COLLADA-Dokumente gelten die folgenden Bestimmungen:

- Eine Referenzierung von einem AML-Objekt auf ein COLLADA-Dokument muss mittels eines CAEX-ExternalInterface modelliert werden, welche vom AML-InterfaceClass „COLLADAInterface“, festgelegt in 5.3.1, abgeleitet wird.
- Auf das COLLADA-Dokument muss mittels seiner URI innerhalb des Attributs „refURI“ dieser CAEX-ExternalInterface referenziert werden.
- Der Wert des Attributs „target“ (Ziel), muss auf ein Element innerhalb des COLLADA-Dokuments referenzieren und muss die Syntax des SID-Pfades befolgen.
- Wenn das Attribut „target“ nicht vorhanden ist oder sein Inhalt leer ist und die URI kein Fragment aufweist, muss das Element „scene“ (Szene) des COLLADA-Dokuments berücksichtigt werden.
- Das COLLADA-Dokument und sein Eintrittspunkt müssen wie in Tabelle 7 festgelegt beschlossen werden. Ein Entscheidungsbaum für den Beschluss der verschiedenen Eintrittspunkte ist in Bild A.1 dargestellt.
- Der Wert „explicit“ des Attributs „refType“ muss für AML-Objekte (fortan bezeichnet als „explizites AML-Objekt“) verwendet werden, welche ein Bestandteil der Datendarstellung sind.

ANMERKUNG 1 Ein Beispiel eines expliziten AML-Objekts ist die Geometrie eines Roboterarms und ein Förderband in einer allgemeinen geometrischen Szene.

- Der Wert „implicit“ des Attributs „refType“ muss für AML-Objekte (fortan bezeichnet als „implizites AML-Objekt“) verwendet werden, welche für besondere Zwecke der Referenzierung auf bereits dargestellte Objekte verwendet werden, z. B. für die Veröffentlichung von Frames oder die Referenzierung auf bereits dargestellte Objekte.

ANMERKUNG 2 Ein Beispiel eines impliziten AML-Objekts ist die Geometrie eines Roboterarms, auf welche bereits explizit verwiesen wurde. Die implizite Referenzierung vermeidet doppelte Darstellungen von Geometrien.

- Auf Objekte im COLLADA-Dokument darf nur unter Verwendung des Wertes „implicit“ referenziert werden, wenn sie Bestandteil eines COLLADA-Objekts oder identisch mit einem COLLADA-Objekt sind, welches bereits mit einem Wert „explicit“ dargestellt ist. Implizite AML-Objekte müssen als direkte oder indirekte Kinder eines expliziten AML-Objekts modelliert werden.
- In Fällen, in welchen ein COLLADA-Dokument (fortan als Hauptdokument bezeichnet) entsprechend den Regeln der Instanziierung und externen Referenzierung, wie in ISO/PAS 17506 festgelegt, auf andere COLLADA-Dokumente referenziert (fortan als Unterdokument bezeichnet), dürfen alle Referenzierungen auf Elemente innerhalb der COLLADA-Unterdokumente nur über das COLLADA-Hauptdokument erfolgen. Ein Beispiel ist in A.2.2 angegeben.
- Ein AML-Objekt muss null oder ein „COLLADAInterface“ als direktes Kind haben.

Tabelle 7 – Regeln für das Auflösen eines Dokuments und eines Eintrittspunktes

		Attribut „refURI“	
		URI mit Fragment	URI ohne Fragment
Attribut „target“	SID-Pfad beginnt mit einer ID	URI legt das Dokument fest. SID-Pfad legt den Eintrittspunkt fest.	URI legt das Dokument fest. SID-Pfad legt den Eintrittspunkt fest.
	SID-Pfad beginnt ohne ID	URI legt das Dokument fest, und ein zugehöriges Element für den Eintrittspunkt fest. SID-Pfad legt den Eintrittspunkt fest.	Nicht festgelegter Status. Diese Kombination darf nicht angewendet werden.
	Fehlt oder Wert ist leer	URI legt das Dokument und den Eintrittspunkt fest.	URI legt das Dokument fest. Das Element „scene“ legt den Eintrittspunkt fest.

8 Anbindung von zwei AML-Objekten

Dieser Abschnitt legte die normativen Bestimmungen für die Anbindung von zwei AML-Objekten fest. Ein Beispiel ist in A.3 angegeben.

In Bezug auf die Anbindung von zwei AML-Objekten gelten die folgenden Bestimmungen:

- Ein AML-Objekt muss null oder ein „AttachmentInterface“ als direktes Kind haben.
- Ein Link zwischen zwei „ExternalInterfaces“, welche einer InterfaceClass „AttachmentInterface“ zugeordnet sind muss eine unidirektionale Verbindung mit fester geometrischer Kopplung unter Verwendung eines CAEX-Elements „InternalLink“ sein.
- Die relative Transformation des AML-Objekts, auf welches durch das Attribut „RefPartnerSideA“ (Referenzierung-Partnerseite A) verwiesen wird, muss konstant bleiben, auch wenn das Objekt „RefPartnerSideA“ transformiert wird, woraus sich im Fall einer Transformation von „RefPartnerSideA“ eine Transformation von „RefPartnerSideB“ ergibt. Eine Transformation von „RefPartnerSideB“ ändert nicht die Transformation von „RefPartnerSideA“.

ANMERKUNG 1 „RefPartnerSideA“ und „RefPartnerSideB“ sind Attribute des CAEX-Elements „InternalLink“.

- Da ein Anbindungspunkt ein Bestandteil eines AML-Objekts ist, muss die „AttachmentInterface“ ein Bestandteil des zugehörigen AML-Objekts sein.
- Wenn eine „AttachmentInterface“ das Kind eines AML-Objekts mit der RoleClass „Frame“ oder einer gerbten RoleClass ist, muss das zu verschiebende Objekt das Eltern-AML-Objekt des AML-Objekts mit

der RoleClass „Frame“ oder einer geerbten RoleClass sein (auch wenn das Eltern-AML-Objekt die RoleClass „Frame“ oder eine geerbte RoleClass hat).

- Wenn ein AML-Objekt, das eine „AttachmentInterface“ besitzt, die RoleClass „Frame“ oder eine geerbte RoleClass hat, und sie eine Schnittstelle vom Typ „COLLADAInterface“ aufweist, muss das Eltern-AML-Objekt ebenfalls eine Schnittstelle des Typs „COLLADAInterface“ aufweisen. Die „COLLADAInterface“ des Eltern-AML-Objekts stellt die Grenze der Geometrie der angebundenen Geometrie dar.
- Wenn eine Anbindung auf ein geometrisches Element referenziert, muss eine „COLLADAInterface“ auf einen COLLADA-Knoten der visuellen Szene referenzieren. Die Anbindung muss die gesamte Geometrie (rekursiv in Abhängigkeit vom Knotenattribut eines Elements „bind_kinematics_model“ (Anbinden von Kinematik-Modell)) der angebundenen Geometrie ohne Abänderung des kinematischen Status des angebundenen Objekts bewegen.
- Wenn eine Anbindung auf ein kinematisches Element referenziert, muss ein „COLLADAInterface“ auf ein COLLADA-Element innerhalb eines COLLADA-Elements „kinematic_scene“ (kinematische Szene) referenzieren. Die Anbindung muss sich auf die Kopplungskonfiguration des angebundenen kinematischen Objekts auswirken, wenn sie das Eltern-AML-Objekt des angebundenen Objekts bewegt. Das Importwerkzeug ist verantwortlich für die Berechnung solcher gekoppelten kinematischen Elemente.

ANMERKUNG 2 Zur Schaffung einer bidirektionalen Verbindung sind zwei unidirektionale Links erforderlich, wobei jeder Link in entgegengesetzter Richtung zum anderen verläuft.

ANMERKUNG 3 Ein „AttachmentInterface“ darf mehr als ein Mal verknüpft sein.

9 Meta-Informationen über das COLLADA-Quellwerkzeug

Für eine Vereinfachung des Datenaustausches zwischen einem Quellwerkzeug und einem Zielwerkzeug ist es zweckmäßig, Informationen über das Quellwerkzeug direkt in das COLLADA-Dokument zu speichern. Daraus folgend gelten die folgenden Bestimmungen:

- Jedes COLLADA-Dokument muss Informationen über das Werkzeug bereitstellen, welches das COLLADA-Dokument geschrieben hat.
- In einer Kette von Datenaustausch-Werkzeugen muss das letzte Werkzeug, welches das Dokument bearbeitet, diese Informationen in das COLLADA-Dokument schreiben.
- Diese Informationen müssen im Kindelement „asset“ (Wert) des Elements „COLLADA“ gespeichert werden.
- Die Meta-Informationen müssen enthalten:
 - Anbieter des Schreibwerkzeugs;
 - URL des Schreibwerkzeugs;
 - Bezeichnung des Schreibwerkzeugs;
 - Produktversion des Schreibwerkzeugs;
 - Produktfreigabenummer des Schreibwerkzeugs;
 - Zeitstempel der Erzeugung und Zeitstempel der letzten Änderung des COLLADA-Dokuments;
 - Festlegung von Einheit und Achse nach oben.
- Die Meta-Informationen müssen mittels der in Tabelle 8 angegebenen Elemente gespeichert werden.

Tabelle 8 – Meta-Informationen über das COLLADA-Quellwerkzeug

XML-Tag-Name nach COLLADA 1.4.1	XML-Tag-Name nach ISO/PAS 17506	Typ	Stufe	Beispiel
„author“ (Ersteller)	„author“	xs:string	Vorgeschrieben	ToolX Vendor
„authoring_tool“ (Erstellerwerkzeug)	„authoring_tool“	xs:string	Vorgeschrieben	„ToolX“ „0.2“ „123 prealpha“
„comments“ (Kommentare)	„author_website“ (Webseite des Erstellers)	xs:anyURI	Vorgeschrieben	http://www.ToolX-Vendor.org
„created“ (erstellt)	„created“	xs:dateTime	Vorgeschrieben	2014-04-14T11:37:18.1875000
„modified“ (geändert)	„modified“	xs:dateTime	Vorgeschrieben	2014-04-14T11:37:18.1875000
„unit meter =... name =...“ (Einheit Meter =... Name...)	„unit meter =... name =...“	xs:string	Vorgeschrieben	Name = „meter“ meter = „1.0“
„up_axis“ (Achse nach oben)	„up_axis“	xs:string	Vorgeschrieben	Z_UP

Die XML-Elemente müssen die folgenden Informationen enthalten:

- „author“ gibt den Anbieter des Schreibwerkzeugs an;
- „authoring_tool“ gibt den Namen, die Produktversion und die Produktfreigabenummer des Schreibwerkzeugs, getrennt durch ein Leerzeichen und jedes von ihnen in Anführungszeichen, an;
- im Fall von nach ISO/PAS 17506 gibt „author_website“ die URL des Schreibwerkzeugs an, im Fall von COLLADS 1.4.1 gibt „comments“ die URL des Schreibwerkzeugs an;
- „created“ und „modified“ geben den Zeitstempel der Erstellung und den Zeitstempel der letzten Abänderung entsprechend den in ISO/PAS 17506 festgelegten Regeln an;
- „unit“ und „up_axis“ müssen entsprechend den Festlegungen in ISO/PAS 17506 oder COLLADA 1.4.1 festgelegt werden.

```
<asset>
  <contributor>
    <author>ToolX Vendor</author>
    <author_website>http://www.ToolX-Vendor.org</author_website>
    <authoring_tool>"ToolX" "0.2" "123 prealpha"</authoring_tool>
  </contributor>
  <created>2014-04-14T11:37:18.1875000</created>
  <modified>2014-04-14T11:37:18.1875000</modified>
  <!-- ..... -->
  <unit name="meter" meter="1.0" />
  <up_axis>Z_UP</up_axis>
</asset>
```

Bild 2 – Beispiel für erforderlichen XML-Text im Fall von ISO/PAS 17506

```
<asset>
  <contributor>
    <author>ToolX Vendor</author>
    <comments>http://www.ToolX-Vendor.org</comments>
    <authoring_tool>"ToolX" "0.2" "123 prealpha"</authoring_tool>
  </contributor>
  <created>2014-04-14T11:37:18.1875000</created>
  <modified>2014-04-14T11:37:18.1875000</modified>
  <!-- ..... -->
  <unit name="meter" meter="1.0" />
  <up_axis>Z_UP</up_axis>
</asset>
```

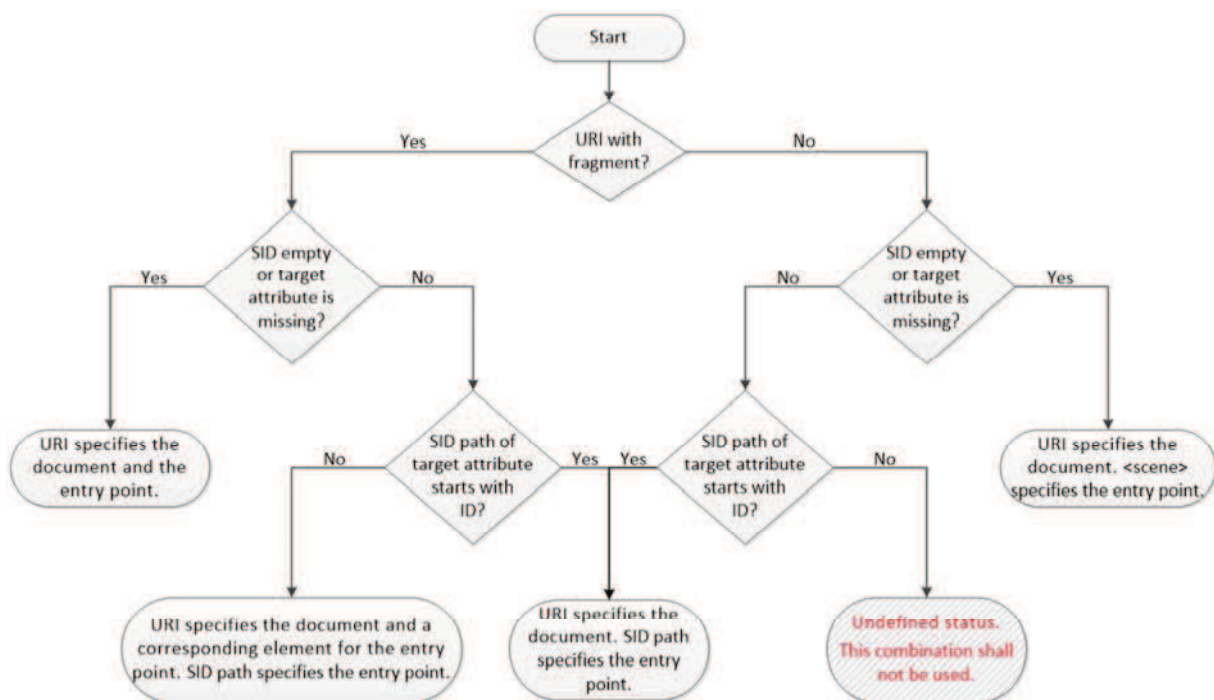
Bild 3 – Beispiel für erforderlichen XML-Text im Fall von COLLADA 1.4.1

Anhang A (informativ)

Verfahren der Referenzierung für geometrische/kinematische Festlegungen

A.1 Integration eines allgemeinen COLLADA-Dokuments mit expliziter Referenzierung

Dieser Abschnitt legt das Verfahren der expliziten Referenzierung für Geometrie- und Kinematik-Modelle fest, welche in einem COLLADA-Dokument angeordnet sind. In Bild A.1 sind die verschiedenen in den folgenden Abschnitten verwendeten Verfahren mittels eines Entscheidungsbaums dargestellt.



Start	Start
Yes	Ja
No	Nein
URI with fragment?	URI mit Fragment?
SID empty or target attribute	SID leer oder Zielt-Attribut fehlt?
URI specifies document entry point	URI legt das Dokument und den Eintrittspunkt fest.
SID path of target attribute	SIP-Pfad des Zielattributs beginnt mit ID?
URI specifies the document and a corresponding	URI legt das Dokument und ein entsprechendes Element für den Eintrittspunkt fest. SID-Pfad legt den Eintrittspunkt fest.
URI specifies SID path specifies the entry point	URI legt das Dokument fest. SID-Pfad legt den Eintrittspunkt fest
URI specifies the document. <scene>	URI legt das Dokument fest. <scene> legt den Eintrittspunkt fest.
Undefined status	Nicht festgelegter Status. Diese Kombination darf nicht angewendet werden.

Bild A.1 – Entscheidungsbaum für verschiedene Referenzierungsverfahren

ANMERKUNG 1 In diesem Dokument werden Zeichenketten von Pfaden oder Unterpfeiden in Kurzform wie „PATH1“ (Pfad 1) oder „PATH2“ usw. dargestellt. Dies dient der Lesbarkeit und steht für einen realen Pfad oder Unterpfad. Gleiche Nummern zeigen den gleichen Pfad an und somit die semantische Gleichwertigkeit an.

ANMERKUNG 2 Für eine bessere Lesbarkeit werden nur maßgebliche Teile von AML/COLLADA-Dokumenten dargestellt. Fehlende Teile sind mit „[...]“ angegeben.

ANMERKUNG 3 Für eine bessere Lesbarkeit wird jedes Attribut „RefBaseClassPath“ (Referenzierung-Basisklassenpfad), welches auf die Klasse „AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector/COLLADAInterface“ referenziert, durch „PATH_CI“ ersetzt.

ANMERKUNG 4 Für eine bessere Lesbarkeit wird jedes Attribut „RefBaseClassPath“ (Referenzierung-Basisklassenpfad), welches auf die Klasse „AutomationMLInterfaceClassLib/AutomationMLBaseInterface/AttachmentInterface“ referenziert, durch „PATH_AI“ ersetzt.

ANMERKUNG 5 Für eine bessere Lesbarkeit wird jedes Attribut „RefBaseRoleClassPath“ (Referenzierung-Basisrollenklassenpfad), welches auf die Klasse „AutomationMLDMIRoleClassLib/DiscManufacturingEquipment/StaticObject“ referenziert, durch „PATH_SO“ ersetzt.

ANMERKUNG 6 Für eine bessere Lesbarkeit wird jedes Attribut „RefBaseRoleClassPath“ (Referenzierung-Basisrollenklassenpfad), „AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Frame“ referenziert, durch „PATH_FRAME“ ersetzt.

A.1.1 Festlegung des Attributs „Frame“

Das folgende Beispiel zeigt, wie die geometrische Position und die Ausrichtung von zwei CAEX-Objekten unter Verwendung des in Abschnitt 6 erwähnten Frame-Attributs festgelegt werden. Das entsprechende AML-Dokument besteht aus einer „InstanceHierarchy“, bezeichnet als „ExampleInstanceHierarchy“ (Beispiel-Instanzhierarchie) mit zwei InternalElements „BaseFrame“ (Basis-Frame) und „ChildFrame“ (Kind-Frame) (Siehe Bild A.2).

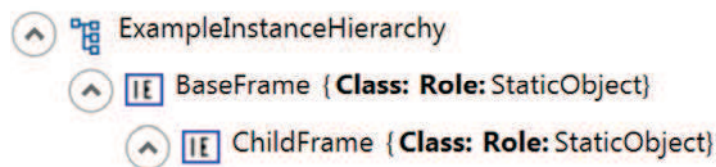


Bild A.2 – Zwei Frames dargestellt in der „InstanceHierarchy“ eines AML-Dokuments

Zur Beschreibung der vollständigen Koordinatentransformation des InternalElement „ChildFrame“ bezüglich seines Eltern-Knotens „BaseFrame“, erbt das vorherige Objekt das Frame-Attribut. In Bild A.3 ist der maßgebliche Teil des AML-Dokuments dargestellt, und wie das Frame-Attribut und seine Unterattribute festzulegen sind.

```
<InstanceHierarchy Name="ExampleInstanceHierarchy">
  <Version>1.0.0</Version>
  <InternalElement Name="BaseFrame" ID="d0235958-bce3-4b48-bec9-08d1033cc49e">
    <InternalElement Name="ChildFrame" ID="07218e99-f4d8-4dd9-ba92-16c6f0dd4f46">
      <Attribute Name="Frame">
        <Attribute Name="x" AttributeDataType="xs:double">
          <Value>1.00</Value>
        </Attribute>
        <Attribute Name="y" AttributeDataType="xs:double">
          <Value>1.00</Value>
        </Attribute>
        <Attribute Name="z" AttributeDataType="xs:double">
          <Value>1.00</Value>
        </Attribute>
        <Attribute Name="rx" AttributeDataType="xs:double">
          <Value>45.00</Value>
        </Attribute>
        <Attribute Name="ry" AttributeDataType="xs:double">
          <Value>45.00</Value>
        </Attribute>
        <Attribute Name="rz" AttributeDataType="xs:double">
          <Value>45.00</Value>
        </Attribute>
      </Attribute>
      <RoleRequirements RefBaseRoleClassPath="AutomationMLDMIRoleClassLib/./StaticObject" />
    </InternalElement>
    <RoleRequirements RefBaseRoleClassPath="AutomationMLDMIRoleClassLib/./StaticObject" />
  </InternalElement>
</InstanceHierarchy>
```

Bild A.3 – XML-Darstellung des AML-Dokuments

In Bild A.4 ist mit der Annahme, dass das Koordinatensystem $K_1(x_1, y_1, z_1)$ zum InternalElement „ChildFrame“ gehört, sowohl seine relative Verschiebung um x , y und z als auch seine raumfeste Drehung um rx , ry und rz in Bezug auf die drei Achsen des Koordinatensystems $K_0(x_0, y_0, z_0)$ dargestellt, welches durch das Elternelement „BaseFrame“ festgelegt wird. Durch Verwendung einer homogenen Transformationsmatrix H_1^0 und der sechs in Tabelle 6 erwähnten Unterattribute kann die relative Verschiebung und Ausrichtung zwischen K_1 und K_0 in einem kompakten Format wie folgt angegeben werden:

$$H_1^0(x, y, z, rx, ry, rz) = \begin{bmatrix} R_1^0 & d_1^0 \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \quad (\text{A.1})$$

Dabei gibt die Drehmatrix R_1^0 die Ausrichtung von K_1 in Bezug zu K_0 an. Abschnitt 6 wird die Reihenfolge der Drehung um rx , ry und rz bestimmt. Im Zusammenhang mit raumfesten Drehungen sei daran erinnert, dass die zweckmäßigen Drehmatrizen von der linken zur rechten Seite multipliziert werden müssen. Dadurch ergibt sich der Drehteil von H_1^0 als:

$$R_1^0 = \text{Rot}_z(rz)\text{Rot}_y(ry)\text{Rot}_x(rx). \quad (\text{A.2})$$

Schritt 1 Verschiebung um x, y und z	Schritt 2 Drehung um rx um x_0	Schritt 3 Drehung um ry um y_0	Schritt 4 Drehung um rz um z_0
$d_1^0(x,y,z)$	$Rot_x(rx)$	$Rot_y(ry)$	$Rot_z(rz)$

Bild A.4 – Verschiebung und raumfeste Drehung

Jede Drehmatrix ist gekennzeichnet als:

$$Rot_x(rx) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(rx) & -\sin(rx) \\ 0 & \sin(rx) & \cos(rx) \end{bmatrix}, \quad Rot_y(ry) = \begin{bmatrix} \cos(ry) & 0 & \sin(ry) \\ 0 & 1 & 0 \\ -\sin(ry) & 0 & \cos(ry) \end{bmatrix} \text{ und}$$

$$Rot_z(rz) = \begin{bmatrix} \cos(rz) & -\sin(rz) & 0 \\ \sin(rz) & \cos(rz) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Der Verschiebungsvektor wird angegeben als

$$d_1^0 = \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

Wie in Bild A.4 dargestellt, behält der verschobene Frame seine kartesischen Koordinaten x, y und z während der raumfesten Drehung. Durch Substitution von $\sin(xi) = sx_i$ und $\cos(xi) = cx_i$ ergeben Gleichung A.2 und die Multiplikation in Gleichung A.1 ergibt das kompakte Format

$$H_1^0 = \begin{bmatrix} cr_z cr_y & cr_z sr_y sr_x - sr_z cr_x & cr_z sr_y sr_x + sr_z sr_x & x \\ sr_z sr_y & sr_z sr_y sr_x + sr_z cr_x & sr_z sr_y cr_x - cr_z sr_x & y \\ -sr_y & cr_y sr_x & cr_y sr_x & z \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

welches die Verschiebung und Ausrichtung des InternalElement „ChildFrame“ in Bezug auf seinen Eltern-Knoten „BaseFrame“ festlegt.

A.1.2 Struktur des COLLADA-Dokuments

Die in den folgenden Beispielen verwendete Szene besteht aus zwei Würfeln, wobei sich einer auf dem anderen befindet. In der visuellen Szene wird ein Knoten für jeden Würfel instanziiert. Jeder Knoten referenziert auf dasselbe geometrische Element, aber verwendet eine(n) andere(n) Namen („RedCube“ (roter

Würfel), „BlueCube“ (blauer Würfel)), Material (rot, blau) und Transformation. In Bild A.5 ist die gerenderte Szene mit dem entsprechenden Graphen der COLLADA-Szene dargestellt.

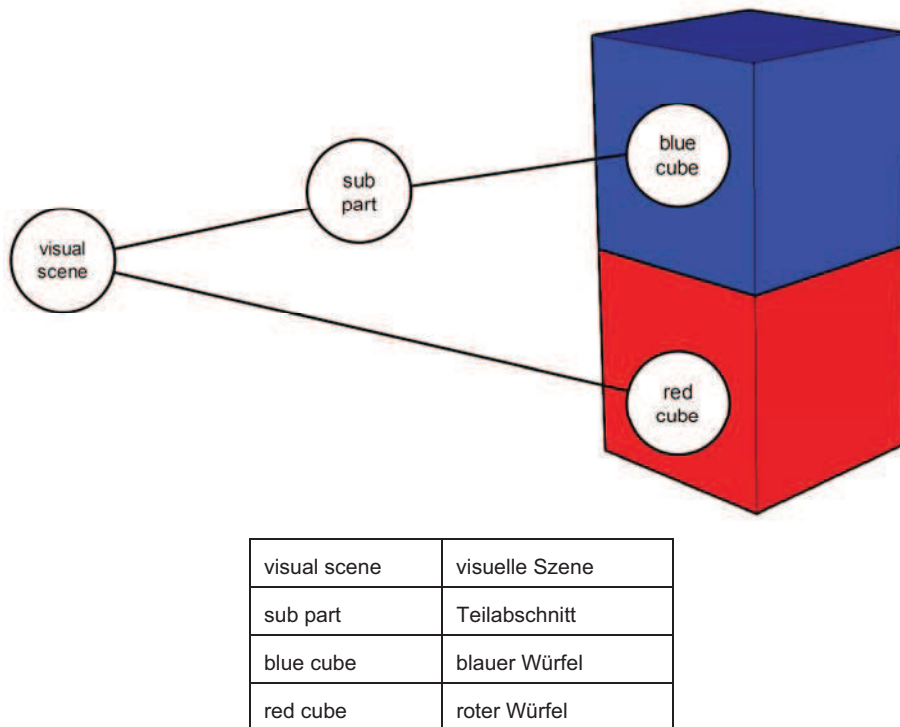


Bild A.5 – Im vorliegenden Beispiel verwendete COLLADA-Szene

Im vorliegenden Beispiel werden zwei COLLADA-Dokumente verwendet, um die vollständige visuelle Szene festzulegen. Die Knotenhierarchie ist im COLLADA-Hauptdokument „red_blue_cubes.dae“ festgelegt. Die Geometrie des Würfels ist in einer Bibliothek innerhalb des separaten COLLADA-Unterdokuments „cube.dae“ festgelegt. Jeder Knoten referenziert auf das Unterdokument „cube.dae“ und instanziiert eine Würfelgeometrie entsprechend den Regeln der Instanziierung und externen Referenzierung, wie sie in ISO/PAS 17506 festgelegt sind. In Bild A.6 sind die im vorliegenden Beispiel verwendete(n) Struktur und Referenzierungen dargestellt.

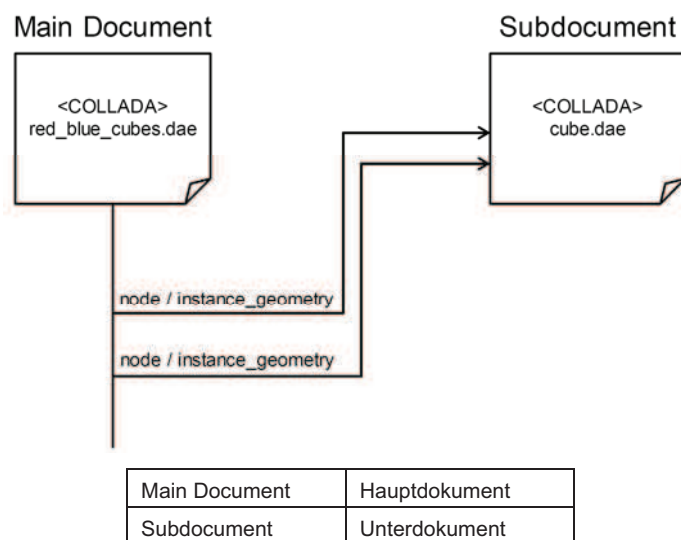


Bild A.6 – Struktur und Referenzierungen

Der Inhalt des COLLADA-Unterdokuments „cube.dae“ ist in Bild A.7 dargestellt.

```
<COLLADA version="1.5.0">
  <asset>
    <contributor>
      <author>WriterVendor</author>
      <author_website>http://www.WriterVendor.com</author_website>
      <authoring_tool>"WriterName" "WriterVersion" "WriterRelease"</authoring_tool>
      <comments>This document defines the geometry of a unit cube stored as reusable library
element</comments>
    </contributor>
    <created>2014-04-14T11:37:18.1875000</created>
    <modified>2014-04-14T11:37:18.1875000</modified>
    <unit meter="1.0" name="meter" />
    <up_axis>Z_UP</up_axis>
  </asset>
  <library_geometries>
    <geometry id="cube-geom" name="cube">
      <mesh>
        <source id="cube-geom-positions">
          <float_array id="cube-geom-positions-array" count="144"> 1 1 0 1 0 1 1 0 0 1 1 1 1 1 1 1 1
0 1 0 1 1 0 0 1 1 0 0 0 0 0 1 0 1 0 0 1 0 0 1 1 0 0 0 0 0 1 0 0 1 1 1 1 0 0 1 0 1 1 1 1 1 1 0 1 1 1
1 0 0 1 0 1 0 1 0 1 1 0 0 1 1 1 1 1 1 1 1 0 1 0 1 1 0 0 1 1 0 0 1 1 0 0 0 1 0 0 0 1 0 0 1 1 0 1 0 0 0
1 0 0 0 1 1 0 0 0 0 0 1 0 1 0 0 1 0 0 1 1 0 0 0 0 0 1 </float_array>
          <technique_common>
            <accessor count="48" source="#cube-geom-positions-array" stride="3">
              <param name="X" type="float" />
              <param name="Y" type="float" />
              <param name="Z" type="float" />
            </accessor>
          </technique_common>
        </source>
        <source id="cube-geom-normals">
          <float_array id="cube-geom-normals-array" count="144"> 1 0 0 1 0 0 1 0 0 1 0 0 -1 0 0 -1 0
0 -1 0 0 -1 0 0 0 0 -1 0 0 -1 0 0 -1 0 0 -1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 -1 0 0
-1 0 0 -1 0 0 -1 0 0 0 1 0 0 1 0 0 1 0 0 1 0 0 -1 0 0 -1 0 0 -1 0 -1 0 0 -1 0 0 -1 0 0 -1 0 0
1 0 0 1 0 0 1 0 0 1 0 -1 0 0 -1 0 0 -1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 </float_array>
          <technique_common>
            <accessor count="48" source="#cube-geom-normals-array" stride="3">
              <param name="X" type="float" />
              <param name="Y" type="float" />
              <param name="Z" type="float" />
            </accessor>
          </technique_common>
        </source>
        <vertices id="cube-geom-vertices">
          <input semantic="POSITION" source="#cube-geom-positions" />
          <input semantic="NORMAL" source="#cube-geom-normals" />
        </vertices>
        <triangles count="24" material="mat">
          <input offset="0" semantic="VERTEX" source="#cube-geom-vertices" />
          <p>0 1 2 1 0 3 4 5 6 7 6 5 8 9 10 9 8 11 12 13 14 15 14 13 16 17 18 17 16 19 20 21 22 23
22 21 24 25 26 25 24 27 28 29 30 31 30 29 32 33 34 33 32 35 36 37 38 39 38 37 40 41 42 41 40 43 44
45 46 47 46 45</p>
        </triangles>
      </mesh>
    </geometry>
  </library_geometries>
</COLLADA>
```

Bild A.7 – Inhalt des COLLADA-Dokuments „cube.dae“

Das Hauptdokument („red_blue_cubes.dae“) einschließlich der Knotenhierarchie und der externen Referenzierungen auf die Würfelgeometrie („cube.dae“) ist in Bild A.8 dargestellt. Jede instanziierte Geometrie überschreibt das Materialsymbol mit seiner eigenen Materialfestlegung.

```
<COLLADA version="1.5.0">
  <asset>
    <contributor>
      <author>WriterVendor</author>
      <author_website>http://www.WriterVendor.com</author_website>
      <authoring_tool>WriterName "WriterVersion" "WriterRelease"</authoring_tool>
      <comments>This document defines the node hierarchy and instantiates cubes from
subdocuments</comments>
    </contributor>
    <created>2014-04-14T11:37:18.1875000</created>
    <modified>2014-04-14T11:37:18.1875000</modified>
    <unit meter="1.0" name="meter" />
    <up_axis>Z_UP</up_axis>
  </asset>
  <library_effects>
    <effect id="red-fx">
      <profile_COMMON>
        <technique sid="COMMON">
          <phong>
            <diffuse>
              <color>1 0 0 1</color>
            </diffuse>
          </phong>
        </technique>
      </profile_COMMON>
    </effect>
    <effect id="blue-fx">
      <profile_COMMON>
        <technique sid="COMMON">
          <phong>
            <diffuse>
              <color>0 0 1 1</color>
            </diffuse>
          </phong>
        </technique>
      </profile_COMMON>
    </effect>
  </library_effects>
  <library_materials>
    <material id="red" name="Red">
      <instance_effect url="#red-fx" />
    </material>
    <material id="blue" name="Blue">
      <instance_effect url="#blue-fx" />
    </material>
  </library_materials>
  <library_visual_scenes>
    <visual_scene id="visualscene">
      <node id="redcube" name="RedCube">
        <instance_geometry url="./cube.dae#cube-geom">
          <bind_material>
            <technique_common>
              <instance_material symbol="mat" target="#red" />
            </technique_common>
          </bind_material>
        </instance_geometry>
      </node>
      <node id="subpart">
        <translate>0 0 1</translate>
        <node name="bluecube" sid="bluecube">
          <instance_geometry url="./cube.dae#cube-geom">
            <bind_material>
              <technique_common>
                <instance_material symbol="mat" target="#blue" />
              </technique_common>
            </bind_material>
          </instance_geometry>
        </node>
      </node>
    </visual_scene>
  </library_visual_scenes>
  <scene>
    <instance_visual_scene url="#visualscene" />
  </scene>
</COLLADA>
```

Bild A.8 – Inhalt des COLLADA-Dokuments „red_blue_cubes.dae“

A.1.3 Referenzierung unter Verwendung von URI und Fragmenten ohne Ziel und ID

Das folgende Beispiel eines AML-Dokuments besteht aus einer InstanceHierarchy mit der Bezeichnung „ExampleInstanceHierarchy“ mit einen einzelnen InternalElement „RedCube“. Letzteres enthält eine CAEX-ExternalInterface des Typs „COLLADAInterface“ mit der Bezeichnung „FileLinkExplicit“ (Datei-Link-Explizit), welche Bestandteil der Standard-InterfaceClassLibrary „AutomationMLInterfaceClassLib“ ist, wie in Bild A.9 festgelegt.

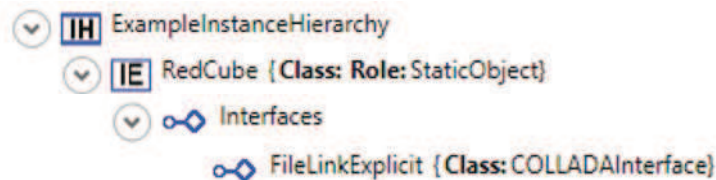


Bild A.9 – „RedCube“ – Hierarchie des AML-Dokuments

In Bild A.10 ist der maßgebliche Teil des AML-Dokuments dargestellt, und wie die Attribute für die Referenzierung festzulegen sind.

```
<InstanceHierarchy Name="ExampleInstanceHierarchy">
  <InternalElement Name="RedCube" ID="GUID1">
    <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID2">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>./red_blue_cubes.dae#redcube</Value>
      </Attribute>
      <Attribute Name="refType" AttributeDataType="xs:string">
        <Value>explicit</Value>
      </Attribute>
    </ExternalInterface>
    <RoleRequirements RefBaseRoleClassPath="PATH_SO" />
  </InternalElement>
</InstanceHierarchy>
```

Bild A.10 – XML-Darstellung des AML-Dokuments

Die ExternalInterface „FileLinkExplicit“ referenziert auf das COLLADA-Dokument, was in Bild A.11 dargestellt ist. Durch die Verwendung der relativen URI „./red_blue_cubes.dae#redcube“ wird eindeutig auf den roten Würfel verwiesen.

```
<visual_scene id="visualscene">
  <node id="redcube" name="RedCube">
    <instance_geometry url="./cube.dae#cube-geom">
      <bind_material>
        <technique_common>
          <instance_material symbol="mat" target="#red" />
        </technique_common>
      </bind_material>
    </instance_geometry>
  </node>
  <node id="subpart"> [...] </node>
</visual_scene>
```

Bild A.11 – Referenzierung auf den roten Würfel durch ID

A.1.4 Referenzierung unter Verwendung einer URI und von Fragmenten einschließlich eines Ziels ohne ID

Das AML-Dokuments besteht aus einer InstanceHierarchy mit der Bezeichnung „ExampleInstanceHierarchy“ mit einem einzelnen InternalElement „BlueCube“. Letzteres enthält eine ExternalInterface

„COLLADAInterface“ mit der Bezeichnung „FileLinkExplicit“, welche Bestandteil der Standard-InterfaceClassLibrary „AutomationMLInterfaceClassLib“ ist, wie in Bild A.12 festgelegt.

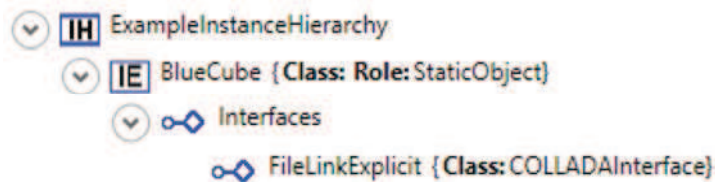


Bild A.12 – „BlueCube“ – Hierarchie des AML-Dokuments

In Bild A.13 ist der maßgebliche Teil des AML-Dokuments dargestellt, und wie die Attribute für die Referenzierung festzulegen sind.

```
<InstanceHierarchy Name="ExampleInstanceHierarchy">
  <InternalElement Name="BlueCube" ID="GUID1">
    <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID2">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>./red_blue_cubes.dae#subpart</Value>
      </Attribute>
      <Attribute Name="refType" AttributeDataType="xs:string">
        <Value>explicit</Value>
      </Attribute>
      <Attribute Name="target" AttributeDataType="xs:token">
        <Value>./bluecube</Value>
      </Attribute>
    </ExternalInterface>
    <RoleRequirements RefBaseRoleClassPath="PATH_SO" />
  </InternalElement>
</InstanceHierarchy>
```

Bild A.13 – XML-Darstellung des AML-Dokuments

Die ExternalInterface „FileLinkExplicit“ referenziert auf das COLLADA-Dokument, was in Bild A.14 dargestellt ist. Zuerst wird der SID-Eintrittspunkt durch die relative URI „./red_blue_cubes.dae#subpart“ festgelegt. Das im COLLADA-Element festgelegte Zielelement (target) mit der Bezeichnung „bluecube“ wird dann mittels Durchsuchen nach dem verwiesenen Element „subpart“ (Teilabschnitt) entsprechend den Regeln einer SID aufgelöst.

```
<visual_scene id="visualscene">
  <node id="redcube" name="RedCube"> [...] </node>
  <node id="subpart">
    <translate>0 0 1</translate>
    <node name="bluecube" sid="bluecube"> [...] </node>
  </node>
</visual_scene>
```

Bild A.14 – Referenzierung auf den blauen Würfel

A.1.5 Referenzierung unter Verwendung eines URI ohne ein Fragment einschließlich eines Ziels und einer ID

Das AML-Dokument besteht aus einer InstanceHierarchy mit der Bezeichnung „ExampleInstanceHierarchy“ mit einem einzelnen InternalElement „BlueCube“. Letzteres enthält eine ExternalInterface „COLLADAInterface“ mit der Bezeichnung „FileLinkExplicit“, welche Bestandteil der Standard-InterfaceClassLibrary „AutomationMLInterfaceClassLib“ ist, wie in Bild A.15 festgelegt.

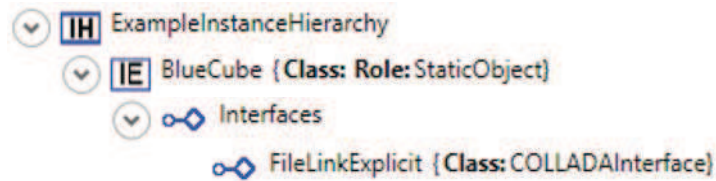


Bild A.15 – „BlueCube“ – Hierarchie des AML-Dokuments

In Bild A.16 ist der maßgebliche Teil des AML-Dokuments dargestellt, und wie die Attribute für die Referenzierung festzulegen sind.

```

<InstanceHierarchy Name="ExampleInstanceHierarchy">
  <InternalElement Name="BlueCube" ID="GUID1">
    <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID2">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>./red_blue_cubes.dae</Value>
      </Attribute>
      <Attribute Name="refType" AttributeDataType="xs:string">
        <Value>explicit</Value>
      </Attribute>
      <Attribute Name="target" AttributeDataType="xs:token">
        <Value>subpart/bluecube</Value>
      </Attribute>
    </ExternalInterface>
    <RoleRequirements RefBaseRoleClassPath="PATH_SO" />
  </InternalElement>
</InstanceHierarchy>
  
```

Bild A.16 – XML-Darstellung des AML-Dokuments

Die ExternalInterface „FileLinkExplicit“ referenziert auf das COLLADA-Dokument, was in Bild A.17 dargestellt ist. Zuerst wird der SID-Eintrittspunkt durch den relativen URI „./red_blue_cubes.dae“ festgelegt. Das im COLLADA-Element festgelegte Zielelement mit der Bezeichnung „bluecube“ wird dann mittels Durchsuchen nach „subpart/bluecube“ ab dem Eintrittspunkt der SID aufgelöst.

```

<visual_scene id="visualscene">
  <node id="redcube" name="RedCube"> [...] </node>
  <node id="subpart">
    <translate>0 0 1</translate>
    <node name="bluecube" sid="bluecube"> [...] </node>
  </node>
</visual_scene>
  
```

Bild A.17 – Referenzierung auf den blauen Würfel beginnend mit dem Element „subpart“

A.1.6 Referenzierung unter Verwendung einer URI und von Fragmenten einschließlich eines Ziels und einer ID

Das AML-Dokuments besteht aus einer InstanceHierarchy mit der Bezeichnung „ExampleInstanceHierarchy“ mit einem einzelnen InternalElement „BlueCube“. Letzteres enthält eine ExternalInterface „COLLADAInterface“ mit der Bezeichnung „FileLinkExplicit“, welche Bestandteil der Standard-InterfaceClassLibrary „AutomationMLInterfaceClassLib“ ist, wie in Bild A.18 festgelegt.

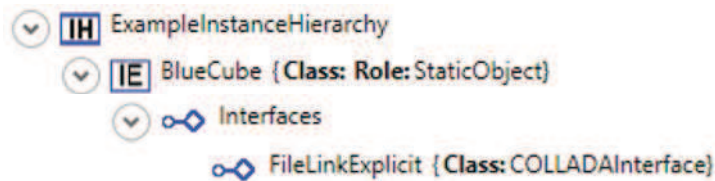


Bild A.18 – Hierarchie des AML-Dokuments

In Bild A.19 ist der maßgebliche Teil des AML-Dokuments dargestellt, und wie die Attribute für die Referenzierung festzulegen sind.

```

<InstanceHierarchy Name="ExampleInstanceHierarchy">
  <InternalElement Name="BlueCube" ID="GUID1">
    <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID2">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>./red_blue_cubes.dae#redcube</Value>
      </Attribute>
      <Attribute Name="refType" AttributeDataType="xs:string">
        <Value>explicit</Value>
      </Attribute>
      <Attribute Name="target" AttributeDataType="xs:token">
        <Value>subpart/bluecube</Value>
      </Attribute>
    </ExternalInterface>
    <RoleRequirements RefBaseRoleClassPath="PATH_SO" />
  </InternalElement>
</InstanceHierarchy>
  
```

Bild A.19 – XML-Darstellung des AML-Dokuments

Die ExternalInterface „FileLinkExplicit“ referenziert auf das COLLADA-Dokument, was in Bild A.20 dargestellt ist. Im vorliegenden Beispiel wird der blaue Würfel wie bereits in A.1.5 dargestellt adressiert. Zuerst wird der SID-Eintrittspunkt durch den relativen URI „./red_blue_cubes.dae#redcube“ festgelegt. An diesem Punkt wird gezeigt, dass das Fragment den relativen URI für die Referenzierung auf den blauen Würfel tatsächlich nicht erforderlich ist. In diesem Fall wird der rote Würfel mittels der Beurteilung der SID beginnend mit der ID aufgelöst. Da die ID jeder SID im gesamten Dokument eindeutig sein muss, ist die vorgeschlagene Referenzierung ebenfalls eindeutig.

```

<visual_scene id="visualscene">
  <node id="redcube" name="RedCube"> [...] </node>
  <node id="subpart">
    <translate>0 0 1</translate>
    <node name="bluecube" sid="bluecube"> [...] </node>
  </node>
</visual_scene>
  
```

Bild A.20 – Referenzierung auf den blauen Würfel

A.1.7 Referenzierung unter Verwendung einer URI ohne ein Fragment, ein Ziel und einer ID

Das AML-Dokument besteht aus einer InstanceHierarchy mit der Bezeichnung „ExampleInstanceHierarchy“ mit einem einzelnen InternalElement „BlueCube“. Letzteres enthält eine ExternalInterface „COLLADAInterface“ mit der Bezeichnung „FileLinkExplicit“, welche Bestandteil der Standard-InterfaceClassLibrary „AutomationMLInterfaceClassLib“ ist, wie in Bild A.21 festgelegt.

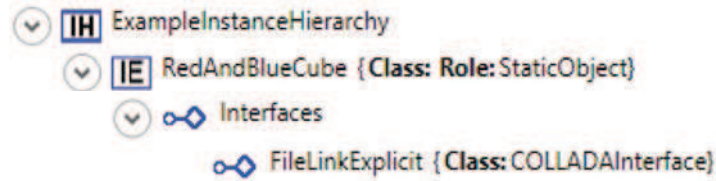


Bild A.21 – Hierarchie des AML-Dokuments

In Bild A.22 ist der maßgebliche Teil des AML-Dokuments dargestellt, und wie die Attribute für die Referenzierung festzulegen sind.

```

<InstanceHierarchy Name="ExampleInstanceHierarchy">
  <InternalElement Name="RedAndBlueCube" ID="GUID1">
    <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID2">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>./red_blue_cubes.dae</Value>
      </Attribute>
      <Attribute Name="refType" AttributeDataType="xs:string">
        <Value>explicit</Value>
      </Attribute>
    </ExternalInterface>
    <RoleRequirements RefBaseRoleClassPath="PATH_SO" />
  </InternalElement>
</InstanceHierarchy>
  
```

Bild A.22 – XML-Darstellung des AML-Dokuments

Die ExternalInterface „FileLinkExplicit“ referenziert auf das COLLADA-Dokument, was in Bild A.23 dargestellt ist. Im vorliegenden Beispiel wird das vollständige COLLADA-Element „scene“ adressiert. Sowohl auf den roten als auch auf den blauen Würfel wird eindeutig mittels der Verwendung der relativen URI „./red_blue_cubes.dae“ ohne jegliches zusätzliche Fragment oder Ziel referenziert.

```
<COLLADA>
[...]
```



```
<library_visual_scenes>
  <visual_scene id="visualscene">
    <node id="redcube" name="RedCube">
      <instance_geometry url="./cube.dae#cube-geom">
        <bind_material>
          <technique_common>
            <instance_material symbol="mat" target="#red" />
          </technique_common>
        </bind_material>
      </instance_geometry>
    </node>
    <node id="subpart">
      <translate>0 0 1</translate>
      <node name="bluecube" sid="bluecube">
        <instance_geometry url="./cube.dae#cube-geom">
          <bind_material>
            <technique_common>
              <instance_material symbol="mat" target="#blue" />
            </technique_common>
          </bind_material>
        </instance_geometry>
      </node>
    </node>
  </visual_scene>
</library_visual_scenes>
<scene>
  <instance_visual_scene url="#visualscene" />
</scene>
</COLLADA>
```

Bild A.23 – Referenzierung auf die vollständige COLLADA-Szene

A.2 Implizite Referenzierung auf COLLADA-Elemente

Dieser Abschnitt legt die impliziten Verfahren der Referenzierung für Geometrie- und Kinematik-Modelle fest, welche sich in einem externen COLLADA-Dokument befinden. Implizite Referenzierung wird verwendet, um von einem CAEX-Objekt auf ein COLLADA-Objekt zu referenzieren und zusätzliche Informationen über das COLLADA-Objekt auf der CAEX-Ebene bereitzustellen, z. B. ein Frame zur Bereitstellung von Anbindungsmöglichkeiten. In diesem Fall ist das COLLADA-Objekt aus einem anderen Zusammenhang bekannt, z. B. von einer expliziten Referenzierung (siehe Abschnitt A.1). In A.2.1 ist die implizite Referenzierung auf ein einzelnes COLLADA-Dokument festgelegt, während in A.2.2 die implizite Referenzierung auf verschachtelte COLLADA-Dokumente festgelegt ist. In A.2.3 ist festgelegt, wie Elemente beziehungsweise Frames eines COLLADA-Dokuments zu veröffentlichen sind.

A.2.1 Implizite Referenzierung

Das folgende Beispiel verwendet die visuelle Szene von Abschnitt A.1.2 (COLLADA-Dokument „red_blue_cubes.dae“). Das AML-Dokument besteht aus einer InstanceHierarchy mit der Bezeichnung „ExampleInstanceHierarchy“ mit drei InternalElements, wie in Bild A.24 dargestellt. Das InternalElement „RedAndBlue_CubeExplicit“ schließt eine ExternalInterface „COLLADAInterface“ mit der Bezeichnung „FileLinkExplicit“ ein. Beide InternalElements „BlueCubeImplicit“ (blauer Würfel implizit) und „RedCubeImplicit“ (roter Würfel implizit) schließen eine ExternalInterface „COLLADAInterface“ mit der Bezeichnung „FileLinkExplicit“ ein, welche ein Bestandteil der Standard-InterfaceClassLibrary „AutomationMLInterfaceClassLib“ sind.

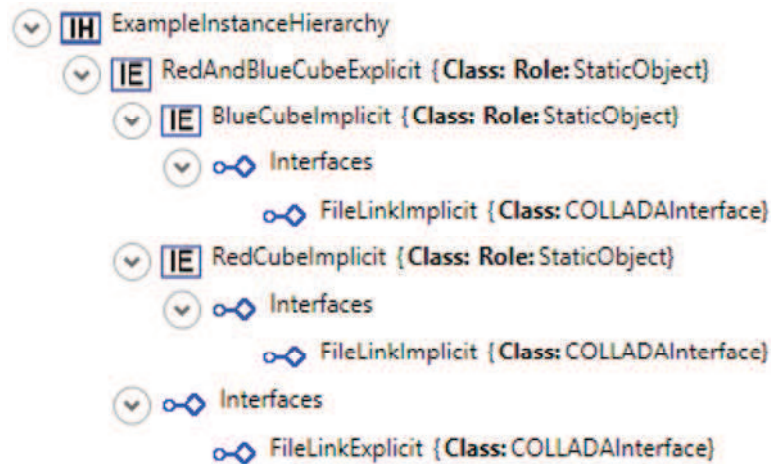


Bild A.24 – Implizite Referenzierung: Hierarchie des AML-Dokuments

Die ExternalInterface „FileLinkExplicit“ des InternalElement „RedAndBlueCubeExplicit“ referenziert auf das COLLADA-Dokument und adressiert explizit die vollständige COLLADA-Szene (siehe Abschnitt A.1.7). Sowohl der rote als auch der blaue Würfel werden eindeutig mittels der Verwendung des relativen URI „./red_blue_cubes.dae“ ohne jegliches zusätzliche Fragment oder Ziel referenziert.

Die InternalElements „RedCubeImplicit“ und „BlueCubeImplicit“ referenzieren implizit auf den roten und blauen Würfel. In Bild A.25 ist der maßgebliche Teil des AML-Dokuments dargestellt, und wie die Attribute für die Referenzierung festzulegen sind.

```
<InstanceHierarchy Name="ExampleInstanceHierarchy">
  <InternalElement Name="RedAndBlueCubeExplicit" ID="GUID1">
    <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID2">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>./red_blue_cubes.dae</Value>
      </Attribute>
      <Attribute Name="refType" AttributeDataType="xs:string">
        <Value>explicit</Value>
      </Attribute>
    </ExternalInterface>
  <InternalElement Name="BlueCubeImplicit" ID="GUID5">
    <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID6">
      <Attribute Name="refType" AttributeDataType="xs:string">
        <Value>implicit</Value>
      </Attribute>
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>./red_blue_cubes.dae#bluecube</Value>
      </Attribute>
    </ExternalInterface>
    <RoleRequirements RefBaseRoleClassPath="PATH_SO" />
  </InternalElement>
  <InternalElement Name="RedCubeImplicit" ID="GUID3">
    <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID4">
      <Attribute Name="refType" AttributeDataType="xs:string">
        <Value>implicit</Value>
      </Attribute>
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>./red_blue_cubes.dae#redcube</Value>
      </Attribute>
    </ExternalInterface>
    <RoleRequirements RefBaseRoleClassPath="PATH_SO" />
  </InternalElement>
  <RoleRequirements RefBaseRoleClassPath="PATH_SO" />
</InternalElement>
</InstanceHierarchy>
```

Bild A.25 – XML-Darstellung des AML-Dokuments

Im Fall der Visualisierung ist es nicht erforderlich, sowohl den roten als auch den blauen Würfel nochmal zu rendern, weil beide bereits über die explizite Referenz auf das Element „RedAndBlueCubeExplicit“ visualisiert wurden. In diesem Fall wird die explizit referenzierte Szene („RedAndBlueCubeExplicit“) visualisiert.

A.2.2 Implizite Referenzierung auf COLLADA-Unterdokumente

Entsprechend der externen Referenzierung, wie in ISO/PAS 17506 und COLLADA 1.4.1 festgelegt, darf die geometrische Festlegung einer Szene in separate COLLADA-Dokumente aufgeteilt werden. In einem solchen Fall referenziert ein COLLADA-Hauptdokument auf Elemente, welche in referenzierten COLLADA-Unterdokumenten gespeichert sind. In Bild A.26 sind die im vorliegenden Beispiel verwendete(n) Struktur und die Beziehungen dargestellt.

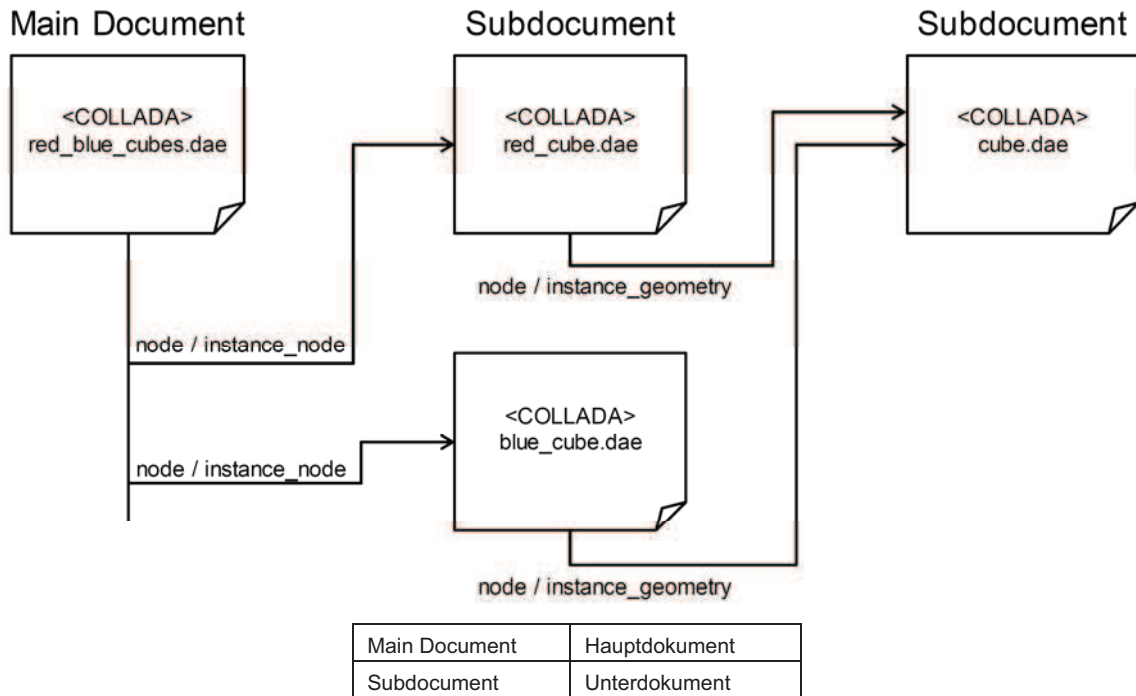


Bild A.26 – Struktur und Beziehungen von referenzierten COLLADA-Unterdokumenten

Das folgende Beispiel verwendet die visuelle Szene aus Abschnitt A.1.2 mit einer abgeänderten Struktur von COLLADA-Dokumenten. Das COLLADA-Hauptdokument („red_blue_cubes.dae“) besteht aus zwei Knoten, welche jeweils einen Knoten instanziierten, der in einem COLLADA-Unterdokument („red_cube.dae“ und „blue_cube.dae“) bereitgestellt wird. Letzteres COLLADA-Unterdokument besteht aus einer Bibliothek, welche einen einzelnen Knoten und ein Material festlegt. Jeder Knoten beider Dokumente instanziiert eine Würfelgeometrie, welche in einem COLLADA-Unterdokument („cube.dae“) festgelegt ist. Während die Festlegung der Geometrie dieselbe bleibt, wie die in Abschnitt A.1.2 verwendete, wird das COLLADA-Hauptdokument („red_blue_cubes.dae“) abgeändert und es werden im vorliegenden Beispiel zwei zusätzliche COLLADA-Unterdokumente („red_cube.dae“ und „blue_cube.dae“) eingefügt. In Bild A.27 ist der Inhalt des abgeänderten COLLADA-Dokuments („red_blue_cubes.dae“) dargestellt.

```
<COLLADA version="1.5.0">
  <asset>
    <contributor>
      <author>WriterVendor</author>
      <author_website>http://www.WriterVendor.com</author_website>
      <authoring_tool>"WriterName" "WriterVersion" "WriterRelease"</authoring_tool>
      <comments>This document defines the node hierarchy and instantiates nodes from
subdocuments</comments>
    </contributor>
    <created>2014-04-14T11:37:18.1875000</created>
    <modified>2014-04-14T11:37:18.1875000</modified>
    <unit meter="1.0" name="meter" />
    <up_axis>Z_UP</up_axis>
  </asset>
  <library_visual_scenes>
    <visual_scene id="visualscene">
      <node id="redcube" name="RedCubeInstance">
        <instance_node sid="redcuberef" url="./red_cube.dae#redcube" />
      </node>
      <node id="bluecube" name="blueCubeInstance">
        <translate>0 0 1</translate>
        <instance_node sid="bluecuberef" url="./blue_cube.dae#bluecube" />
      </node>
    </visual_scene>
  </library_visual_scenes>
  <scene>
    <instance_visual_scene url="#visualscene" />
  </scene>
</COLLADA>
```

Bild A.27 – Inhalt des abgeänderten COLLADA-Dokuments „red_blue_cubes.dae“

Die visuelle Szene besteht aus zwei Knotenelementen (redcube und bluecube), welche jeweils einen Knoten in einem Unterdokument („red_cube.dae“ und „blue_cube.dae“) instanziiieren. Der Inhalt beider COLLADA-Unterdokumente ist in Bild A.28 und Bild A.29 dargestellt.

```
<COLLADA version="1.5.0">
  <asset>
    <contributor>
      <author>WriterVendor</author>
      <author_website>http://www.WriterVendor.com</author_website>
      <authoring_tool>"WriterName" "WriterVersion" "WriterRelease"</authoring_tool>
      <comments>This document defines a node which instantiates a cube geometry from a COLLADA
subdocument and applies a red material</comments>
    </contributor>
    <created>2014-04-14T11:37:18.1875000</created>
    <modified>2014-04-14T11:37:18.1875000</modified>
    <unit meter="1.0" name="meter" />
    <up_axis>Z_UP</up_axis>
  </asset>
  <library_effects>
    <effect id="red-fx">
      <profile_COMMON>
        <technique sid="COMMON">
          <phong>
            <diffuse>
              <color>1 0 0 1</color>
            </diffuse>
          </phong>
        </technique>
      </profile_COMMON>
    </effect>
  </library_effects>
  <library_materials>
    <material id="red" name="Red">
      <instance_effect url="#red-fx" />
    </material>
  </library_materials>
  <library_nodes>
    <node id="redcube" name="RedCube">
      <instance_geometry url="./cube.dae#cube-geom">
        <bind_material>
          <technique_common>
            <instance_material symbol="mat" target="#red" />
          </technique_common>
        </bind_material>
      </instance_geometry>
    </node>
  </library_nodes>
</COLLADA>
```

Bild A.28 – Inhalt des COLLADA-Dokuments „red_cube.dae“

Das COLLADA-Dokument „red_cube.dae“ legt eine Knoten „RedCube“ innerhalb einer Bibliothek fest, welche referenziert und instanziiert werden dürfen. Zusätzlich wird ein rotes Material festgelegt, um eine instanziierte Geometrie in das COLLADA-Unterdokument „cube.dae“ einzubinden. In vergleichbarer Weise werden der Knoten und die Geometrie des blauen Würfels mit der Bezeichnung „BlueCube“ im COLLADA-Unterdokument „blue_cube.dae“ festgelegt, wie in Bild A.29 dargestellt.

```
<COLLADA version="1.5.0">
  <asset>
    <contributor>
      <author>WriterVendor</author>
      <author_website>http://www.WriterVendor.com</author_website>
      <authoring_tool>"WriterName" "WriterVersion" "WriterRelease"</authoring_tool>
      <comments>This document defines a node which instantiates a cube geometry from a COLLADA
subdocument and applies a blue material</comments>
    </contributor>
    <created>2014-04-14T11:37:18.1875000</created>
    <modified>2014-04-14T11:37:18.1875000</modified>
    <unit meter="1.0" name="meter" />
    <up_axis>Z_UP</up_axis>
  </asset>
  <library_effects>
    <effect id="blue-fx">
      <profile_COMMON>
        <technique sid="COMMON">
          <phong>
            <diffuse>
              <color>0 0 1 1 </color>
            </diffuse>
          </phong>
        </technique>
      </profile_COMMON>
    </effect>
  </library_effects>
  <library_materials>
    <material id="blue" name="Blue">
      <instance_effect url="#blue-fx" />
    </material>
  </library_materials>
  <library_nodes>
    <node id="bluecube" name="BlueCube">
      <instance_geometry url="../cube.dae#cube-geom">
        <bind_material>
          <technique_common>
            <instance_material symbol="mat" target="#blue" />
          </technique_common>
        </bind_material>
      </instance_geometry>
    </node>
  </library_nodes>
</COLLADA>
```

Bild A.29 – Inhalt des COLLADA-Dokuments „blue_cube.dae“

Um von einem AML-Dokument auf den blauen und den roten Würfel in einem COLLADA-Dokument zu referenzieren, muss der vollständige Pfad beginnend vom COLLADA-Wurzeldokument festgelegt werden. Das folgende Beispiel zeigt einen Auszug aus einem AML-Dokument, um die implizite Referenzierung auf Elemente aufzuzeigen, welche sich in verschiedenen COLLADA-Unterdokumenten befinden.

Das AML-Dokument besteht aus einer InstanceHierarchy mit der Bezeichnung „ExampleInstanceHierarchy“ mit drei InternalElements, wie in Bild A.30 dargestellt. Alle InternalElements schließen eine ExternalInterface des Typs „COLLADAInterface“ ein, welche Bestandteil der Standard-InterfaceClassLibrary „AutomationMLInterfaceClassLib“ ist.

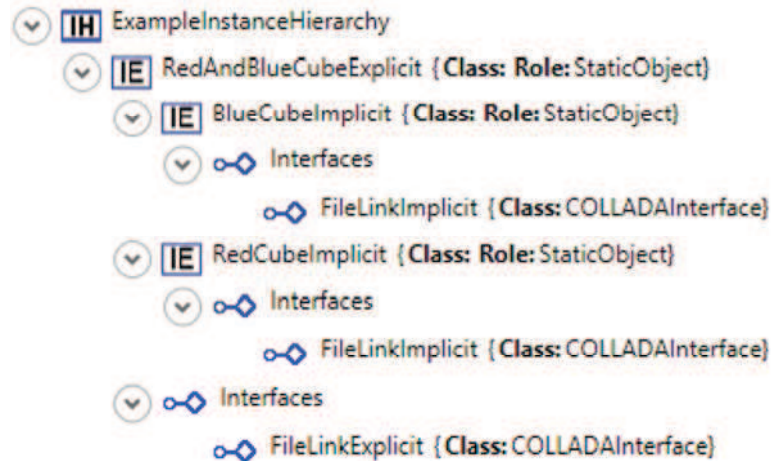


Bild A.30 – Implizite Referenzierung: Hierarchie des AML-Dokuments

Die ExternalInterface „FileLinkExplicit“ des InternalElement „RedAndBlueCubeExplicit“ referenziert auf das COLLADA-Dokument, und adressiert explizit die vollständige COLLADA-Szene (siehe Abschnitt A.1.7). Sowohl der rote als auch auf der blaue Würfel wird eindeutig mittels der Verwendung der relativen URI „./red_blue_cubes.dae“ ohne jegliches zusätzliche Fragment oder Ziel referenziert. Die InternalElements „RedCubeImplicit“ und „BlueCubeImplicit“ referenzieren implizit auf den roten und den blauen Würfel, welche sich beide in einem COLLADA-Unterdokument befinden. In Bild A.31 ist der maßgebliche Teil des AML-Dokuments für implizite Referenzierung dargestellt.

```
<InstanceHierarchy Name="ExampleInstanceHierarchy">
  <InternalElement Name="RedAndBlueCubeExplicit" ID="GUID1">
    <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID2">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>./red_blue_cubes.dae</Value>
      </Attribute>
      <Attribute Name="refType" AttributeDataType="xs:string">
        <Value>explicit</Value>
      </Attribute>
    </ExternalInterface>
  </InternalElement>
  <InternalElement Name="RedCubeImplicit" ID="GUID3">
    <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID4">
      <Attribute Name="refType" AttributeDataType="xs:string">
        <Value>implicit</Value>
      </Attribute>
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>./red_cube.dae</Value>
      </Attribute>
      <Attribute Name="target" AttributeDataType="xs:token">
        <Value>redcube/redcuberef</Value>
      </Attribute>
    </ExternalInterface>
    <RoleRequirements RefBaseRoleClassPath="PATH_SO" />
  </InternalElement>
  <InternalElement Name="BlueCubeImplicit" ID="GUID5">
    <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID6">
      <Attribute Name="refType" AttributeDataType="xs:string">
        <Value>implicit</Value>
      </Attribute>
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>./blue_cube.dae</Value>
      </Attribute>
      <Attribute Name="target" AttributeDataType="xs:token">
        <Value>bluecube/bluecuberef</Value>
      </Attribute>
    </ExternalInterface>
    <RoleRequirements RefBaseRoleClassPath="PATH_SO" />
  </InternalElement>
  <RoleRequirements RefBaseRoleClassPath="PATH_SO" />
</InstanceHierarchy>
```

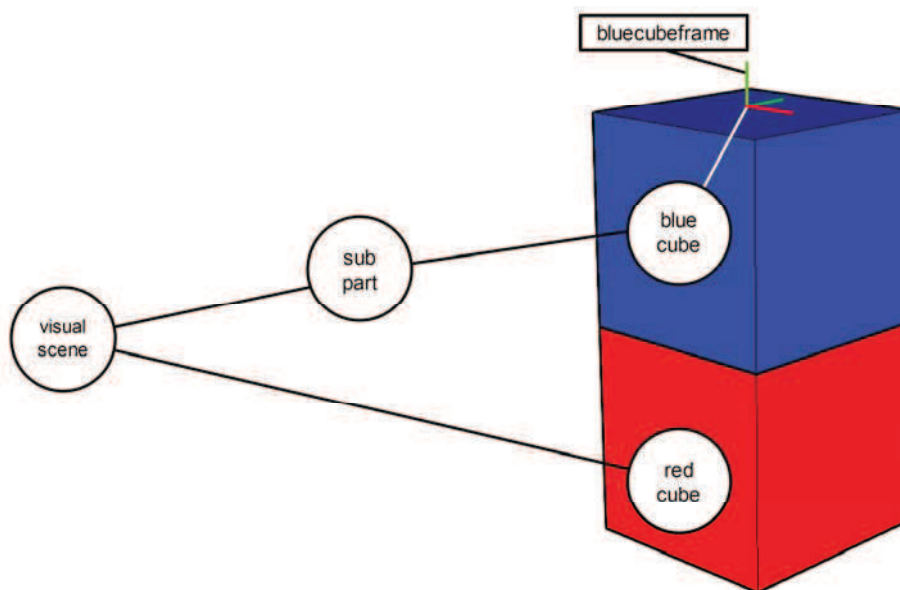
Bild A.31 – XML-Darstellung eines AML-Dokuments

A.2.3 Veröffentlichung von Elementen eines COLLADA-Dokuments in CAEX

In diesem Abschnitt wird ein Anwendungsfall gezeigt, in welchem beide (explizites und implizites) Verfahren der Referenzierung maßgeblich sind. Dieser Abschnitt legt fest, wie Frames von COLLADA-Dokumenten zu veröffentlichen sind und ihre Referenzierung innerhalb eines AML-Dokuments.

Geometrische und kinematische COLLADA-Elemente können auf der CAEX-Ebene veröffentlicht werden, um sie für eine Anwendung verfügbar zu machen. Dies ermöglicht es, auf sie zu referenzieren und weitere Informationen zuzuweisen. Im vorliegenden Beispiel werden das COLLADA-Dokument und seine visuelle Szene, entnommen von A.1.2 (COLLADA-Dokument „red_blue_cubes.dae“) mit einem zusätzlichen COLLADA-Element verwendet. Der zusätzliche Knoten mit der Bezeichnung „bluecubeframe“ wird an den blauen Würfel angebunden und stellt einen Frame dar.

In Bild A.32 ist die abgeänderte visuelle Szene mit dem entsprechenden Graphen der COLLADA-Szene dargestellt.



visual scene	visuelle Szene
sub part	Teilabschnitt
bluecubeframe	Frame des blauen Würfels
blue cube	blauer Würfel
red cube	roter Würfel

Bild A.32 – Abgeänderte COLLADA-Szene mit zusätzlichem Knoten

Der maßgebliche Teil des COLLADA-Dokuments, insbesondere der zusätzliche Frame-Knoten, ist in Bild A.33 dargestellt.

```
<visual_scene id="visualscene">
  <node id="redcube" name="RedCube">
    <instance_geometry url="./cube.dae#cube-geom">
      <bind_material>
        <technique_common>
          <instance_material symbol="mat" target="#red" />
        </technique_common>
      </bind_material>
    </instance_geometry>
  </node>
  <node id="subpart">
    <translate>0 0 1</translate>
    <node name="bluecube" sid="bluecube">
      <node sid="bluecubeframe">
        <translate>0.5 0.5 1.0</translate>
      </node>
      <instance_geometry url="./cube.dae#cube-geom">
        <bind_material>
          <technique_common>
            <instance_material symbol="mat" target="#blue" />
          </technique_common>
        </bind_material>
      </instance_geometry>
    </node>
  </node>
</visual_scene>
```

Bild A.33 – Zusätzliches Frame-Element im COLLADA-Dokument

Im vorliegenden Beispiel besteht das AML-Dokument aus einer InternalHierarchy mit der Bezeichnung „ExampleInstanceHierarchy“ mit einem InternalElement „RedAndBlueCubeExplicit“, welches eine Externalinterface „COLLADAInterface“ mit der Bezeichnung „FileLinkExplicit“ einschließt. Diese Externalinterface wird verwendet, um explizit auf die vollständige Szene zu referenzieren, wie in Bild A.34 dargestellt.

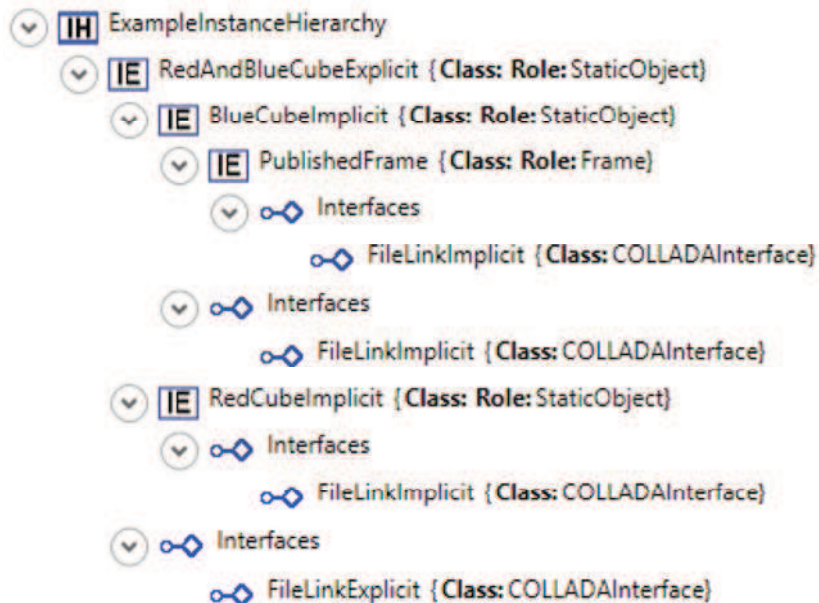


Bild A.34 – Veröffentlichung von Frames: Hierarchie des AML-Dokuments

Das InternalElement „RedAndBlueCubeExplicit“ hat zwei InternalElements mit der Bezeichnung „BlueCubeExplicit“ und „RedCubeExplicit“. Das InternalElement „BlueCubeExplicit“ hat ein InternalElement „PublishedFrame“ (veröffentlichter Frame) mit einer zugewiesenen RoleClass „Frame“. Das ExternalElement „FileLinkImplicit“ des InternalElements „PublishedFrame“ referenziert implizit auf den COLLADA-Knoten „bluecubeframe“ im Knoten des COLLADA-Dokuments.

In Bild A.35 ist der maßgebliche Teil des AML-Dokuments dargestellt, und wie die Attribute für die implizite Referenzierung festzulegen sind.

```
<InstanceHierarchy Name="ExampleInstanceHierarchy">
  <InternalElement Name="RedAndBlueCubeExplicit" ID="GUID1">
    <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID2">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>./red_blue_cubes.dae</Value>
      </Attribute>
      <Attribute Name="refType" AttributeDataType="xs:string">
        <Value>explicit</Value>
      </Attribute>
    </ExternalInterface>
  <InternalElement Name="BlueCubeImplicit" ID="GUID3">
    <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID4">
      <Attribute Name="refType" AttributeDataType="xs:string">
        <Value>implicit</Value>
      </Attribute>
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>./red_blue_cubes.dae#subpart</Value>
      </Attribute>
      <Attribute Name="target" AttributeDataType="xs:token">
        <Value>./bluecube</Value>
      </Attribute>
    </ExternalInterface>
  <InternalElement Name="PublishedFrame" ID="GUID5">
    <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID6">
      <Attribute Name="refType" AttributeDataType="xs:string">
        <Value>implicit</Value>
      </Attribute>
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>./red_blue_cubes.dae#subpart</Value>
      </Attribute>
      <Attribute Name="target" AttributeDataType="xs:token">
        <Value>./bluecube/bluecubeframe</Value>
      </Attribute>
      <Attribute Name="AdditionalInformation1" AttributeDataType="xs:token">
        <Value>AdditionalAttributeValue</Value>
      </Attribute>
      <Attribute Name="AdditionalInformation2" AttributeDataType="xs:token">
        <Value>AdditionalAttributeValue</Value>
      </Attribute>
    </ExternalInterface>
    <RoleRequirements RefBaseRoleClassPath="PATH_FRAME" />
  </InternalElement>
  <RoleRequirements RefBaseRoleClassPath="PATH_SO" />
</InternalElement>
<InternalElement Name="RedCubeImplicit" ID="GUID7">
  <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID8">
    <Attribute Name="refType" AttributeDataType="xs:string">
      <Value>implicit</Value>
    </Attribute>
    <Attribute Name="refURI" AttributeDataType="xs:anyURI">
      <Value>./red_blue_cubes.dae#redcube</Value>
    </Attribute>
  </ExternalInterface>
  <RoleRequirements RefBaseRoleClassPath="PATH_SO" />
</InternalElement>
<RoleRequirements RefBaseRoleClassPath="PATH_SO" />
</InternalElement>
</InstanceHierarchy>
```

Bild A.35 – XML-Darstellung des AML-Dokuments

ANMERKUNG 1 Ein veröffentlichtes CAEX-Objekt mit der zugewiesenen RoleClass „Frame“ muss nicht auf ein COLLADA-Geometrie-Objekt verweisen (es darf auf ein COLLADA-Geometrie-Objekt ohne Geometrie verweisen). Im ersteren Fall wird die Transformationsinformation vom COLLADA-Objekt genommen und auf das veröffentlichte CAEX-Objekt mit der zugewiesenen RoleClass „Frame“ angewendet, auch dann, wenn das CAEX-Frame-Objekt ein CAEX-Frameattribut bereitstellt. Wenn das COLLADA-Objekt keinerlei Transformationsinformation bereitstellt, wird diejenige vom Frameattribut des CAEX-Objekts mit der zugewiesenen RoleClass „Frame“ angewendet.

A.3 Anbindung zwischen Objekten in CAEX

Dieser Abschnitt legt fest, wie ein geometrisches AML-Objekt an ein anderes geometrisches AML-Objekt (beide vom Typ InternalElement) angebunden werden kann. Beide AML-Objekte weisen durch die Verwendung expliziter Referenzen eine Referenzierung auf alle COLLADA-Geometrie-Objekte auf. Außerdem haben beide AML-Objekte ein Kind-InternalElement „Frame“, welches der RoleClass „Frame“ zugeordnet ist. Jedes InternalElement „Frame“ hat eine implizite ExternalInterface, die von der InterfaceClass „COLLADAInterface“ abgeleitet ist, welche auf den im COLLADA-Dokument festgelegten Knoten referenziert. Zusätzlich hat jedes InternalElement „Frame“ eine ExternalInterface, die von der InterfaceClass „AttachmentInterface“ abgeleitet ist, welche verbunden oder verknüpft sein kann. Die Struktur ist in Bild A.36 dargestellt.

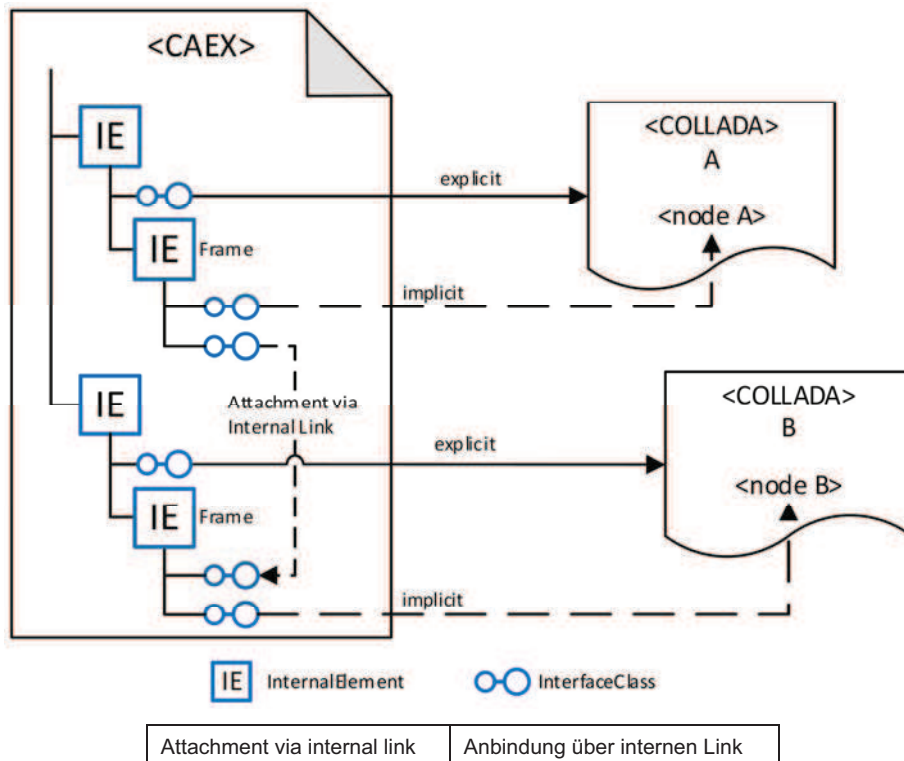


Bild A.36 – Struktur für die Anbindung zwischen Objekten in CAEX

Im vorliegenden Beispiel sind das COLLADA-Dokument und seine visuelle Szene aus dem vorgenannten Abschnitt A.2.2 entnommen, wobei ein zusätzlicher Knoten verwendet wird. Der zusätzliche Knoten, welcher an den blauen Würfel angebunden ist, stellt den veröffentlichten Frame dar. Ein zusätzlicher gelber Würfel ist in einem separaten COLLADA-Dokument („yellow_cube.dae“) mit einem zusätzlichen Knoten festgelegt, welches veröffentlicht wird. Die Verbindung zwischen beiden zusätzlichen Knoten (Anbindungs-Frames) wird innerhalb des AML-Dokuments geschaffen. Bild A.37 zeigt eine Visualisierung des gelben Würfels („YelloCube“) und seinen zusätzlichen Frame.

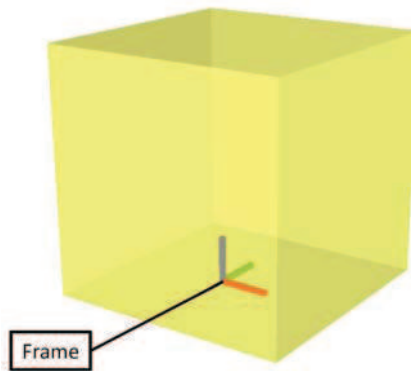


Bild A.37 – Visualisierung des gelben Würfels mit zusätzlichem Frame

Das vollständige COLLADA-Dokument des gelben Würfels („yellow_cube.dae“) ist in Bild A.38 dargestellt. Es verwendet die Festlegung der Geometrie vom externen COLLADA-Dokument „cube.dae“, welches aus A.1.2 entnommen wurde.

```
<COLLADA version="1.5.0">
  <asset>
    <contributor>
      <author>WriterVendor</author>
      <author_website>http://www.WriterVendor.com</author_website>
      <authoring_tool>"WriterName" "WriterVersion" "WriterRelease"</authoring_tool>
      <comments>This document defines a node which instantiates a cube geometry from a COLLADA
subdocument and applies a yellow material. An additional node is provided.</comments>
    </contributor>
    <created>2014-04-14T11:37:18.1875000</created>
    <modified>2014-04-14T11:37:18.1875000</modified>
    <unit meter="1.0" name="meter" />
    <up_axis>Z_UP</up_axis>
  </asset>
  <library_effects>
    <effect id="yellow-fx">
      <profile_COMMON>
        <technique sid="COMMON">
          <phong>
            <diffuse>
              <color>1 1 0 0.5 </color>
            </diffuse>
          </phong>
        </technique>
      </profile_COMMON>
    </effect>
  </library_effects>
  <library_materials>
    <material id="yellow" name="Yellow">
      <instance_effect url="#yellow-fx" />
    </material>
  </library_materials>
  <library_visual_scenes>
    <visual_scene id="visualscene">
      <node id="yellowcube" name="YellowCube">
        <translate>0 0 2</translate>
        <node sid="yellowcubeframe">
          <translate>0.5 0.5 0</translate>
        </node>
        <instance_geometry url="./cube.dae#cube-geom">
          <bind_material>
            <technique_common>
              <instance_material symbol="mat" target="#yellow" />
            </technique_common>
          </bind_material>
        </instance_geometry>
      </node>
    </visual_scene>
  </library_visual_scenes>
  <scene>
    <instance_visual_scene url="#visualscene" />
  </scene>
</COLLADA>
```

Bild A.38 – COLLADA-Dokument eines gelben Würfels mit zusätzlichem Frame

Das AML-Dokument besteht aus einer InstanceHierarchy mit der Bezeichnung „ExampleInstanceHierarchy“ mit zwei InternalElements („RedAndBlueCubeExplicit“ und „YelloCubeExplicit“). Im vorliegenden Beispiel ist der gelbe Würfel („YelloCube“) aus dem separaten COLLADA-Dokument an den blauen Würfel („BlueCube“) angebunden. Beide InternalElements schließen eine ExternalInterface „COLLADAInterface“ mit der Bezeichnung „FileLinkExplicit“ ein, welche ein Bestandteil der Standard-InterfaceClassLibrary „AutomationMLInterfaceClassLib“ ist.

Jede ExternalInterface mit der Bezeichnung „FileLinkExplicit“ von beiden InternalElements verweist auf separate COLLADA-Dokumente und adressiert explizit die vollständige COLLADA-Szene (siehe auch A.1.7). Zusätzlich haben beide InternalElements ein Kind-InternalElement („PublishedFrame“) mit einer angewen-

deten RoleClass „Frame“, welche auch ein Bestandteil der Standard-RoleClassLibrary ist. Jedes InternalElement „PublishedFrame“ hat zwei ExternalInterfaces: eine ExternalInterface, welche von der InterfaceClass „COLLADAInterface“ abgeleitet ist, um implizit auf einen COLLADA-Knoten zu referenzieren („FileLinkImplicit“) und eine ExternalInterface, welche von der InterfaceClass „AttachmentInterface“ abgeleitet ist, um eine Anbindung („Attachment“) festzulegen. Die Struktur ist in Bild A.39 dargestellt.

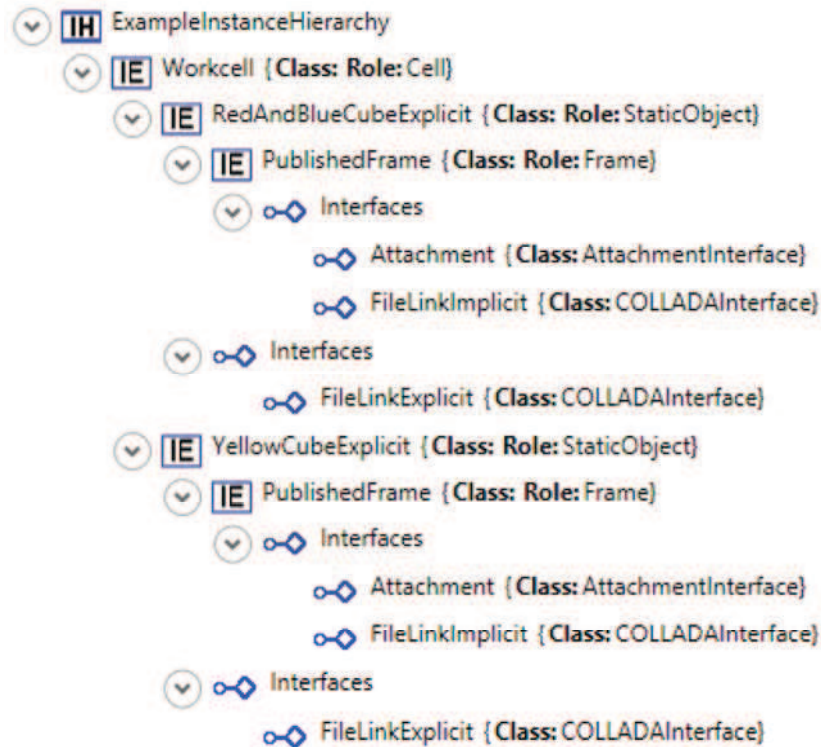


Bild A.39 – Hierarchie des AML-Dokuments

In Bild A.40 ist der maßgebliche Teil des AML-Dokuments dargestellt, und wie Frames für die Veröffentlichung von Objekten zu veröffentlichen sind, welche in AML festgelegt sind.

```
<InstanceHierarchy Name="ExampleInstanceHierarchy">
  <InternalElement Name="Workcell" ID="GUID0">
    <InternalElement Name="RedAndBlueCubeExplicit" ID="GUID1">
      <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID2">
        <Attribute Name="refURI" AttributeDataType="xs:anyURI">
          <Value>./red_blue_cubes.dae</Value>
        </Attribute>
        <Attribute Name="refType" AttributeDataType="xs:string">
          <Value>explicit</Value>
        </Attribute>
      </ExternalInterface>
    <InternalElement Name="PublishedFrame" ID="GUID3">
      <ExternalInterface Name="Attachment" RefBaseClassPath="PATH_AI" ID="GUID4" />
      <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID5">
        <Attribute Name="refType" AttributeDataType="xs:string">
          <Value>implicit</Value>
        </Attribute>
        <Attribute Name="refURI" AttributeDataType="xs:anyURI">
          <Value>./red_blue_cubes.dae#subpart</Value>
        </Attribute>
        <Attribute Name="target" AttributeDataType="xs:token">
          <Value>./bluecube/bluecubeframe</Value>
        </Attribute>
      </ExternalInterface>
      <RoleRequirements RefBaseRoleClassPath="PATH_FRAME" />
    </InternalElement>
    <RoleRequirements RefBaseRoleClassPath="PATH_SO" />
  </InternalElement>
  <InternalElement Name="YellowCubeExplicit" ID="GUID6">
    <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID7">
      <Attribute Name="refType" AttributeDataType="xs:string">
        <Value>explicit</Value>
      </Attribute>
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>./yellow_cube.dae</Value>
      </Attribute>
    </ExternalInterface>
    <InternalElement Name="PublishedFrame" ID="GUID8">
      <ExternalInterface Name="Attachment" RefBaseClassPath="PATH_AI" ID="GUID9" />
      <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID10">
        <Attribute Name="refType" AttributeDataType="xs:string">
          <Value>implicit</Value>
        </Attribute>
        <Attribute Name="refURI" AttributeDataType="xs:anyURI">
          <Value>./yellow_cube.dae#yellowcube</Value>
        </Attribute>
        <Attribute Name="target" AttributeDataType="xs:token">
          <Value>./yellowcubeframe</Value>
        </Attribute>
      </ExternalInterface>
      <RoleRequirements RefBaseRoleClassPath="PATH_FRAME" />
    </InternalElement>
    <RoleRequirements RefBaseRoleClassPath="PATH_SO" />
  </InternalElement>
  <InternalLink Name="AttachmentBlueMovesYellow" RefPartnerSideA="GUID3:Attachment" RefPartnerSideB="GUID7:Attachment" />
  <RoleRequirements RefBaseRoleClassPath="PATH5/Cell" />
</InternalElement>
</InstanceHierarchy>
```

Bild A.40 – XML-Darstellung des AML-Dokuments

Als Ergebnis wird der gelbe Würfel an den blauen angebunden, wie in Bild A.41 dargestellt. Im Gegensatz zum Einbau umfasst die Anbindung keine zusätzlichen Transformationen des angebundenen Objekts ein. Die Verbindung ist unidirektional. Wenn der blaue Würfel verschoben oder gedreht wird, wirkt sich die Transformation des blauen Würfels auch auf den gelben Würfel aus.

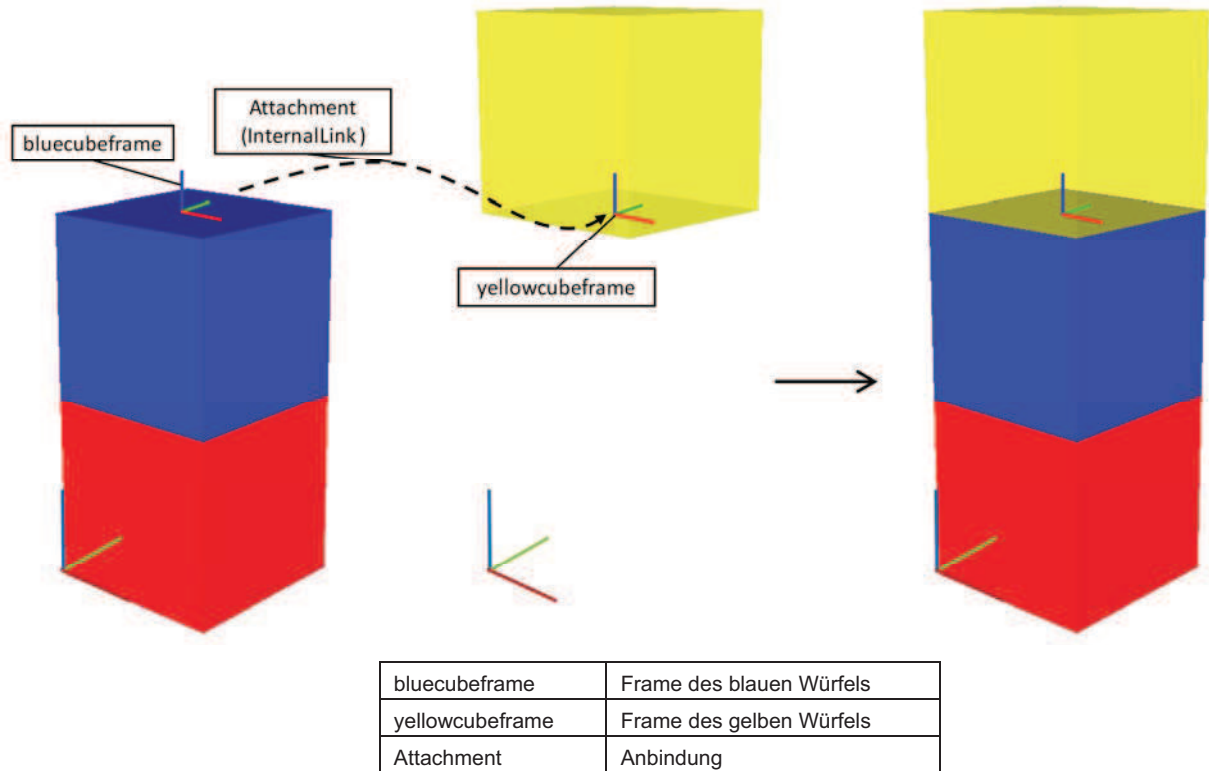


Bild A.41 – Anbindung zwischen geometrischen AML-Objekten

Schließlich wird die Verbindung durch die Verwendung eines InternalLink geschaffen, welcher beide veröffentlichten Frames verbindet. In Bild A.42 ist der maßgebliche Teil des AML-Dokuments dargestellt, und wie ein InternalLink festzulegen ist. Der Link wird im CAEX-InternalElement „Workcell“ (Fertigungseinheit) gespeichert, welches das niedrigste allgemeine Eltern-Element des verbundenen CAEX-Objekts ist. Dies entspricht IEC 62714-1.

```
<InstanceHierarchy Name="ExampleInstanceHierarchy">
  <InternalElement Name="Workcell" ID="GUID0">
    [...]
    <InternalLink Name="AttachmentBlueRedMovesYellowCube"
      RefPartnerSideA="GUID3:Attachment"
      RefPartnerSideB="GUID8:Attachment" />
    </InternalElement>
  </InstanceHierarchy>
```

Bild A.42 – XML-Darstellung des AML-Dokuments

Anhang B (informativ)

Modellierung von Kinematik-Systemen und ihre Kombination in AML

Der folgende Abschnitt legt die Modellierung von Kinematik-Systemen und die Kombination in AML fest. In diesem Abschnitt wird schrittweise eine Szene aufgebaut, welche eine lineare Einheit (B.1) und einen Roboter (B.2) enthält, welcher an der linearen Einheit befestigt ist (B.3). Zusätzlich wird ein Greifer festgelegt und am Roboter angebaut (B.4). Schließlich wird ein Werksstück am Greifer angebunden (B.5).

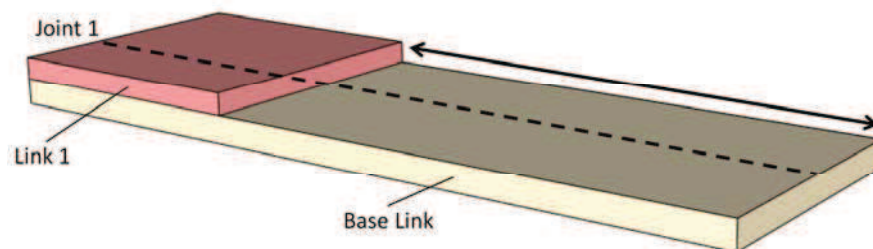
ANMERKUNG 1 Zur besseren Lesbarkeit gelten alle Ersetzungen für „RefBaseRoleClassPath“ von A.1.

ANMERKUNG 2 Zur besseren Lesbarkeit werden alle Attribute „RefBaseRoleClassPath“, die auf die Klasse „AutomationMLMIRoleClassLib/ManufacturingEquipment/Robot“ durch „PATH_ROB“ (Pfad Roboter) ersetzt.

ANMERKUNG 3 Zur besseren Lesbarkeit werden alle Attribute „RefBaseRoleClassPath“, die auf die Klasse „AutomationMLMIRoleClassLib/ManufacturingEquipment/Carrier“ durch „PATH_CAR“ (Pfad Träger) ersetzt.

B.1 Modellierung eines AML-Dokuments einer linearen Einheit in CAEX und COLLADA

Dieser Abschnitt legt die Modellierung einer linearen Einheit fest. In B.1.1 ist die visuelle Szene festgelegt. In den folgenden Abschnitten B.1.2 – B.1.6 wird die Szene schrittweise mit kinematischen Elementen vervollständigt. Schließlich wird das AML-Dokument festgelegt und in B.1.7 mit dem Kinematik-System kombiniert.



Joint	Gelenk
Link	Link
Base Link	Basis-Link

Bild B.1 – Veranschaulichung der linearen Einheit

B.1.1 Festlegung der visuellen Szene

Die visuelle Szene der linearen Einheit im vorliegenden Beispiel besteht aus zwei geometrischen Elementen, wobei sich eines auf dem anderen befindet, wie in Bild B.1 dargestellt. Um den COLLADA-Inhalt auf einem Minimum zu halten, werden die Geometrien der linearen Einheit durch einfache rechteckige Quader abgebildet, welche innerhalb des COLLADA-Dokuments „cube.dae“, wie in Bild A.7 eingeführt, festgelegt sind. Jeder Quader hat eine unterschiedliche Größe und stellt ein eigenes Material bereit. Der untere Quader stellt den Basis-Link der linearen Einheit dar, während der obere Quader den Gleiter darstellt.

In Bild B.2 ist der Inhalt des COLLADA-Dokuments „linear_unit.dae“ dargestellt, welches die geometrische Darstellung der Einheit festlegt.

```
<COLLADA version="1.5.0">
  <asset>
    <contributor>
      <author>WriterVendor</author>
      <author_website>http://www.WriterVendor.com</author_website>
      <authoring_tool>"WriterName" "WriterVersion" "WriterRelease"</authoring_tool>
      <comments>This document defines the geometry and kinematic of a linear unit</comments>
    </contributor>
    <created>2014-04-14T11:37:18.1875000</created>
    <modified>2014-04-14T11:37:18.1875000</modified>
    <unit meter="1.0" name="meter" />
    <up_axis>Z_UP</up_axis>
  </asset>
  <library_effects>
```

```
<effect id="base-fx">
  <profile_COMMON>
    <technique sid="COMMON">
      <phong>
        <diffuse>
          <color>1 0.9 0.8 1</color>
        </diffuse>
      </phong>
    </technique>
  </profile_COMMON>
</effect>
<effect id="link-fx">
  <profile_COMMON>
    <technique sid="COMMON">
      <phong>
        <diffuse>
          <color>1 0.6 0.6 1</color>
        </diffuse>
      </phong>
    </technique>
  </profile_COMMON>
</effect>
</library_effects>
<library_materials>
  <material id="base-mat" name="BaseMat">
    <instance_effect url="#base-fx" />
  </material>
  <material id="link-mat" name="LinkMat">
    <instance_effect url="#link-fx" />
  </material>
</library_materials>
<library_visual_scenes>
  <visual_scene id="visualscene">
    <node id="linearunit" name="LinearUnit">
      <node id="geombase" name="GeomBase">
        <scale>3.0 1.0 0.1</scale>
        <instance_geometry url="./cube.dae#cube-geom">
          <bind_material>
            <technique_common>
              <instance_material symbol="mat" target="#base-mat" />
            </technique_common>
          </bind_material>
        </instance_geometry>
        <node id="geomlink1" name="GeomLink1">
          <translate>0 0 1</translate>
          <scale>0.33 1 0.8</scale>
          <rotate sid="transform">0 0 0</rotate>
          <instance_geometry url="./cube.dae#cube-geom">
            <bind_material>
              <technique_common>
                <instance_material symbol="mat" target="#link-mat" />
              </technique_common>
            </bind_material>
          </instance_geometry>
        </node>
      </node>
    </node>
  </visual_scene>
</library_visual_scenes>
<!-- Step 1: Definition of joint elements -->
<!-- Step 2: Definition of kinematic model element -->
<!-- Step 3: Definition of articulated system (kinematics) element -->
<!-- Step 4: Definition of kinematics scene element -->
<scene>
  <instance_visual_scene url="#visualscene" />
  <!-- Step 5: Instantiation of kinematics scene -->
</scene>
</COLLADA>
```

Bild B.2 – Festlegung der visuellen Szene