

## DIN EN 62714-1



ICS 35.240.50

**Datenaustauschformat für Planungsdaten industrieller  
Automatisierungssysteme –  
Automation markup language –  
Teil 1: Architektur und allgemeine Festlegungen  
(IEC 62714-1:2014);  
Deutsche Fassung EN 62714-1:2014**

Engineering data exchange format for use in industrial automation systems engineering –  
Automation markup language –  
Part 1: Architecture and general requirements  
(IEC 62714-1:2014);  
German version EN 62714-1:2014

Format d'échange de données techniques pour une utilisation dans l'ingénierie des  
systèmes d'automatisation industrielle –  
Automation markup language –  
Partie 1: Architecture et exigences générales  
(CEI 62714-1:2014);  
Version allemande EN 62714-1:2014

Gesamtumfang 85 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE

## Anwendungsbeginn

Anwendungsbeginn für die von CENELEC am 2014-07-31 angenommene Europäische Norm als DIN-Norm ist 2015-06-01.

## Nationales Vorwort

*Vorausgegangener Norm-Entwurf: E DIN EN 62714-1:2012-06.*

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

Die enthaltene IEC-Publikation wurde vom SC 65E „Devices and integration in enterprise systems“ erarbeitet.

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

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

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

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

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

Datenaustauschformat für Planungsdaten industrieller  
Automatisierungssysteme –  
Automation markup language –  
Teil 1: Architektur und allgemeine Festlegungen  
(IEC 62714-1:2014)

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

Format d'échange de données techniques pour  
une utilisation dans l'ingénierie des systèmes  
d'automatisation industrielle –  
Automation markup language –  
Partie 1: Architecture et exigences générales  
(CEI 62714-1:2014)

Diese Europäische Norm wurde von CENELEC am 2014-07-31 angenommen. CENELEC-Mitglieder sind gehalten, die CEN/CENELEC-Geschäftsordnung zu erfüllen, in der die Bedingungen festgelegt sind, unter denen dieser Europäischen Norm ohne jede Änderung der Status einer nationalen Norm zu geben ist.

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

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

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

**CENELEC**

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

**CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brüssel**

© 2014 CENELEC – Alle Rechte der Verwertung, gleich in welcher Form und in welchem Verfahren, sind weltweit den Mitgliedern von CENELEC vorbehalten.

Ref. Nr. EN 62714-1:2014 D

## **Vorwort**

Der Text des Dokuments 65E/385/FDIS, zukünftige 1. Ausgabe der IEC 62714-1, erarbeitet vom SC 65E „Devices and integration in enterprise systems“ des IEC/TC 65 „Industrial-process measurement, control and automation“, wurde zur parallelen IEC-CENELEC-Abstimmung vorgelegt und von CENELEC als EN 62714-1:2014 angenommen.

Nachstehende Daten wurden festgelegt:

- spätestes Datum, zu dem dieses Dokument auf nationaler Ebene durch Veröffentlichung einer identischen nationalen Norm oder durch Anerkennung übernommen werden muss (dop): 2015-05-01
- spätestes Datum, zu dem nationale Normen, die diesem Dokument entgegenstehen, zurückgezogen werden müssen (dow): 2017-07-31

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

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

## **Anerkennungsnotiz**

Der Text der Internationalen Norm IEC 62714-1:2014 wurde von CENELEC ohne irgendeine Abänderung als Europäische Norm angenommen.

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

|                   |           |                                    |
|-------------------|-----------|------------------------------------|
| IEC 60027 (Reihe) | ANMERKUNG | Harmonisiert als EN 60027 (Reihe). |
| IEC 62264-1       | ANMERKUNG | Harmonisiert als EN 62264-1.       |
| IEC 62714-2       | ANMERKUNG | Harmonisiert als EN 62714-2.       |
| ISO 80000-1       | ANMERKUNG | Harmonisiert als EN ISO 80000-1.   |

## Inhalt

|                                                                             | Seite |
|-----------------------------------------------------------------------------|-------|
| Vorwort .....                                                               | 2     |
| Einleitung .....                                                            | 7     |
| 1 Anwendungsbereich .....                                                   | 9     |
| 2 Normative Verweisungen .....                                              | 9     |
| 3 Begriffe und Abkürzungen .....                                            | 9     |
| 3.1 Begriffe .....                                                          | 9     |
| 3.2 Abkürzungen .....                                                       | 12    |
| 4 Konformität .....                                                         | 12    |
| 5 Spezifikation der AML-Architektur .....                                   | 13    |
| 5.1 Allgemeines .....                                                       | 13    |
| 5.2 Allgemeine AML-Architektur .....                                        | 13    |
| 5.3 AML-Dokumentversion .....                                               | 14    |
| 5.4 Metainformationen über das AML-Quell-Werkzeug .....                     | 15    |
| 5.5 Objekt-Identifikation .....                                             | 17    |
| 5.6 Festlegungen für AML-Relationen .....                                   | 18    |
| 5.7 Festlegungen für AML-Dokumentreferenzen .....                           | 24    |
| 6 AML-Basis-Bibliotheken .....                                              | 24    |
| 6.1 Allgemeines .....                                                       | 24    |
| 6.2 Allgemeine Festlegungen .....                                           | 24    |
| 6.3 AML-Schnittstellenklassenbibliothek AutomationMLInterfaceClassLib ..... | 25    |
| 6.4 AML-Basis-Rollenklassenbibliothek AutomationMLBaseRoleClassLib .....    | 30    |
| 7 Modellierung benutzerdefinierter Daten .....                              | 38    |
| 7.1 Allgemeines .....                                                       | 38    |
| 7.2 Benutzerdefinierte Attribute .....                                      | 38    |
| 7.3 Benutzerdefinierte Schnittstellenklassen .....                          | 39    |
| 7.4 Benutzerdefinierte Rollenklassen .....                                  | 40    |
| 7.5 Benutzerdefinierte SystemUnit-Klassen .....                             | 41    |
| 7.6 Benutzerdefinierte Instanzhierarchien .....                             | 41    |
| 8 Erweiterte AML-Konzepte .....                                             | 42    |
| 8.1 Allgemeiner Überblick .....                                             | 42    |
| 8.2 AML-Port-Objekt .....                                                   | 43    |
| 8.3 AML-Facetten-Objekt .....                                               | 43    |
| 8.4 AML-Gruppen-Objekt .....                                                | 44    |
| 8.5 AML-PropertySet .....                                                   | 44    |
| 8.6 Unterstützung multipler Rollen .....                                    | 46    |
| 8.7 Splitten von AML-Toplevel-Daten auf verschiedene Dokumente .....        | 47    |
| 8.8 Internationalisierung .....                                             | 47    |
| 8.9 Versionsinformationen von AML-Objekten .....                            | 47    |

|                                                                                                                                           | Seite |
|-------------------------------------------------------------------------------------------------------------------------------------------|-------|
| Anhang A (informativ) Allgemeine Einführung in die Automation Markup Language .....                                                       | 48    |
| A.1 Allgemeine Automation Markup Language Konzepte .....                                                                                  | 48    |
| A.1.1 Architektur der Automation Markup Language .....                                                                                    | 48    |
| A.1.2 Modellierung von Informationen zur Anlagentopologie .....                                                                           | 50    |
| A.1.3 Referenzierung von Geometrie- und Kinematikinformationen .....                                                                      | 51    |
| A.1.4 Referenzierung von Logikinformationen .....                                                                                         | 51    |
| A.1.5 Modellierung von Relationen .....                                                                                                   | 52    |
| A.2 Erweiterte AML-Konzepte und Beispiele .....                                                                                           | 54    |
| A.2.1 Allgemeiner Überblick .....                                                                                                         | 54    |
| A.2.2 AML-Port-Konzept .....                                                                                                              | 54    |
| A.2.3 AML-Facettenkonzept .....                                                                                                           | 58    |
| A.2.4 AML-Gruppenkonzept .....                                                                                                            | 60    |
| A.2.5 PropertySet-Konzept .....                                                                                                           | 64    |
| A.2.6 Das Prozess-Produkt-Ressourcen-Konzept .....                                                                                        | 67    |
| A.2.7 Unterstützung multipler Rollen .....                                                                                                | 76    |
| Anhang B (informativ) XML-Text der AML-Bibliotheken .....                                                                                 | 80    |
| B.1 AutomationMLBaseRoleClassLib .....                                                                                                    | 80    |
| B.2 AutomationMLInterfaceClassLib .....                                                                                                   | 81    |
| Literaturhinweise .....                                                                                                                   | 82    |
| Anhang ZA (normativ) Normative Verweisungen auf internationale Publikationen mit ihren<br>entsprechenden europäischen Publikationen ..... | 83    |
| <b><u>Bilder</u></b>                                                                                                                      |       |
| Bild 1 – Überblick über das Engineering-Datenaustauschformat AML .....                                                                    | 7     |
| Bild 2 – AML-Dokumentversions-Information .....                                                                                           | 14    |
| Bild 3 – XML-Text der AML-Information zum Quellwerkzeug .....                                                                             | 16    |
| Bild 4 – Beispiel für die Objekt-Identifizierung einer AML-Klasse .....                                                                   | 18    |
| Bild 5 – Beispiel für die Objekt-Identifizierung einer AML-Objektinstanz .....                                                            | 18    |
| Bild 6 – Beispiel einer Eltern-Kind-Beziehung zwischen AML-Objekten .....                                                                 | 19    |
| Bild 7 – Beispiel einer Eltern-Kind-Beziehung zwischen Klassen .....                                                                      | 19    |
| Bild 8 – Beispiel für eine Vererbungsbeziehung zwischen zwei Klassen .....                                                                | 20    |
| Bild 9 – Beispiel einer Klassen-Instanz-Beziehung .....                                                                                   | 21    |
| Bild 10 – Beispiel einer Beziehung als a) Blockdiagramm und b) als Objektbaum .....                                                       | 22    |
| Bild 11 – Beispielbeziehung zwischen den Objekten „PLC1“ und „Rob1“ .....                                                                 | 23    |
| Bild 12 – AML-Basischnittstellenklassenbibliothek .....                                                                                   | 26    |
| Bild 13 – XML-Beschreibung der AML-Basischnittstellenklassenbibliothek .....                                                              | 27    |
| Bild 14 – AML-Basisrollenklassenbibliothek .....                                                                                          | 31    |
| Bild 15 – AutomationMLBaseRoleClassLib .....                                                                                              | 31    |
| Bild 16 – XML-Text der AutomationMLBaseRoleClassLib .....                                                                                 | 32    |
| Bild 17 – Beispiel eines benutzerdefinierten Attributs .....                                                                              | 39    |
| Bild 18 – Beispiel für eine benutzerdefinierte Schnittstellenklasse in einer benutzerdefinierten                                          |       |

|                                                                                                                          | Seite |
|--------------------------------------------------------------------------------------------------------------------------|-------|
| Schnittstellenklassen-Bibliothek.....                                                                                    | 40    |
| Bild 19 – Beispiel für eine benutzerdefinierte Rollenklasse in einer benutzerdefinierten Rollenklassenbibliothek.....    | 41    |
| Bild 20 – Beispiele für unterschiedliche benutzerdefinierte SystemUnitClasses .....                                      | 41    |
| Bild 21 – Beispiel für eine benutzerdefinierte Instanzhierarchie .....                                                   | 42    |
| Bild 22 – AML-Darstellung einer benutzerdefinierten Instanzhierarchie .....                                              | 42    |
| Bild 23 – Beispiel zur Erläuterung des PropertySet-Konzeptes .....                                                       | 45    |
| Bild 24 – XML-Text des PropertySet Beispiels .....                                                                       | 46    |
| Bild A.1 – Allgemeine Architektur von AML.....                                                                           | 48    |
| Bild A.2 – Anlagentopologie mit AML .....                                                                                | 50    |
| Bild A.3 – Referenz von CAEX auf ein COLLADA-Dokument .....                                                              | 51    |
| Bild A.4 – Referenz von CAEX auf ein PLCopen XML Dokument.....                                                           | 52    |
| Bild A.5 – Beziehungen in AML.....                                                                                       | 53    |
| Bild A.6 – XML Beschreibung des Beispiels für die Beziehungen .....                                                      | 53    |
| Bild A.7 – XML-Text der SystemUnitClassLib des Beispiels für die Beziehungen .....                                       | 53    |
| Bild A.8 – XML-Text der Instanzhierarchie des Beispiels für die Beziehungen .....                                        | 54    |
| Bild A.9 – Port-Konzept .....                                                                                            | 55    |
| Bild A.10 – Beispiel zur Beschreibung des AML-Port-Konzeptes .....                                                       | 55    |
| Bild A.11 – XML-Beschreibung des AML-Port-Konzeptes .....                                                                | 56    |
| Bild A.12 – XML-Text, der das AML-Port-Konzept beschreibt.....                                                           | 57    |
| Bild A.13 – Definition einer nutzerdefinierten AML-Port-Klasse “myPortClass” .....                                       | 57    |
| Bild A.14 – Beispiel für eine AML-Facette.....                                                                           | 59    |
| Bild A.15 – XML-Text des AML-Facettenbeispiels.....                                                                      | 59    |
| Bild A.16 – Beispiel für AML-Gruppen.....                                                                                | 60    |
| Bild A.17 – XML-Text des AML-Gruppenbeispiels.....                                                                       | 61    |
| Bild A.18 – Kombination aus Gruppen- und Facettenkonzept .....                                                           | 62    |
| Bild A.19 – XML-Text des Beispiels für die Kombination aus Gruppen- und Facettenkonzept .....                            | 63    |
| Bild A.20 – Generische HMI-Vorlage “B”, die eine Prozessvariable “Y” eines Förderbandes visualisiert.....                | 64    |
| Bild A.21 – Generierte Bedienoberfläche “B”, die beide Förderbänder mit individuellen Prozessvariablen visualisiert..... | 64    |
| Bild A.22 – Beispiel für die Verwendung von PropertySets .....                                                           | 65    |
| Bild A.23 – Beispiel für die Verwendung von PropertySets .....                                                           | 65    |
| Bild A.24 – XML-Text der Instanzhierarchie.....                                                                          | 66    |
| Bild A.25 – Beispiel für eine Eigenschaftsmengenbibliothek als XML-Code .....                                            | 67    |
| Bild A.26 – Basiselemente des Produkt-Prozess-Ressourcen-Konzeptes .....                                                 | 68    |
| Bild A.27 – PPRConnector-Schnittstelle .....                                                                             | 69    |
| Bild A.28 – Beispiel für das Produkt-Prozess-Ressourcen-Konzept.....                                                     | 69    |
| Bild A.29 – AML-Rollen für das Prozess-Produkt-Ressourcen-Konzept .....                                                  | 70    |
| Bild A.30 – Elemente des Beispiels.....                                                                                  | 70    |

|                                                                                                    | Seite |
|----------------------------------------------------------------------------------------------------|-------|
| Bild A.31 – Verknüpfungen innerhalb des Beispiels.....                                             | 71    |
| Bild A.32 – Verknüpfungen der ressourcenzentrierten Sichtweise auf das Beispiel.....               | 72    |
| Bild A.33 – Instanzhierarchie des Beispiels in AML .....                                           | 73    |
| Bild A.34 – InternalElements dieses Beispiels .....                                                | 74    |
| Bild A.35 – InternalLinks dieses Beispiels .....                                                   | 74    |
| Bild A.36 – Instanzhierarchie des Beispiels in XML .....                                           | 75    |
| Bild A.37 – Beispiel für eine nutzerdefinierte Instanz, die multiple Rollen unterstützt.....       | 77    |
| Bild A.38 – XML-Text der AML-Darstellung für die Unterstützung multipler Rollen .....              | 78    |
| Bild A.39 – AML-Rollenklassenbibliothek des Beispiels für die Unterstützung multipler Rollen ..... | 78    |
| Bild A.40 – XML-Text der Rollenklassenbibliothek.....                                              | 79    |
| <b><u>Tabellen</u></b>                                                                             |       |
| Tabelle 1 – Abkürzungen.....                                                                       | 12    |
| Tabelle 2 – Metainformationen über das AML-Quellwerkzeug .....                                     | 16    |
| Tabelle 3 – Schnittstellenklassen der AML AutomationMLInterfaceClassLib .....                      | 25    |
| Tabelle 4 – Schnittstellenklasse AutomationMLBaseInterface.....                                    | 27    |
| Tabelle 5 – Schnittstellenklasse Order .....                                                       | 28    |
| Tabelle 6 – Schnittstellenklasse PortConnector .....                                               | 28    |
| Tabelle 7 – Schnittstellenklasse PPRConnector .....                                                | 28    |
| Tabelle 8 – Schnittstellenklasse ExternalDataConnector.....                                        | 29    |
| Tabelle 9 – Schnittstellenklasse COLLADAInterface .....                                            | 29    |
| Tabelle 10 – Schnittstellenklasse PLCopenXMLInterface.....                                         | 29    |
| Tabelle 11 – Schnittstellenklasse Communication .....                                              | 30    |
| Tabelle 12 – Schnittstellenklasse SignalInterface .....                                            | 30    |
| Tabelle 13 – Rollenklasse AutomationMLBaseRole.....                                                | 32    |
| Tabelle 14 – Rollenklasse Group.....                                                               | 33    |
| Tabelle 15 – Rollenklasse Facet.....                                                               | 33    |
| Tabelle 16 – Optionale Attribute für AML-Port-Objekte .....                                        | 34    |
| Tabelle 17 – Unterattribute des Attribut “Cardinality” .....                                       | 34    |
| Tabelle 18 – Schnittstelle der Klasse AML-Port .....                                               | 35    |
| Tabelle 19 – Rollenklasse Resource .....                                                           | 35    |
| Tabelle 20 – Rollenklasse Product .....                                                            | 35    |
| Tabelle 21 – Rollenklasse Process.....                                                             | 36    |
| Tabelle 22 – Rollenklasse Structure .....                                                          | 36    |
| Tabelle 23 – Rollenklasse ProductStructure .....                                                   | 37    |
| Tabelle 24 – Rollenklasse ProcessStructure.....                                                    | 37    |
| Tabelle 25 – Rollenklasse ResourceStructure .....                                                  | 37    |
| Tabelle 26 – Rollenklasse PropertySet .....                                                        | 38    |
| Tabelle A.1 – Überblick über die wesentlichen erweiterten AML-Konzepte.....                        | 54    |

## Einleitung

Die IEC 62714 definiert eine Datenaustauschlösung mit Fokus auf die Anwendungsdomäne der Automatisierungsplanung.

Das in der IEC 62714 festgelegte Datenaustauschformat (Automation Markup Language, AML) basiert auf XML-Schemata und wurde entwickelt, um den Datenaustausch in einer heterogenen Planungswerkzeug-Landschaft zu unterstützen.

Das Ziel von AML besteht darin, Planungswerkzeuge aus unterschiedlichen Disziplinen, z. B. Maschinen- und Anlagenbau, elektrische Konstruktion, Verfahrenstechnik, Prozessleittechnik, HMI-Entwicklung, SPS-Programmierung, Roboterprogrammierung usw. miteinander zu verbinden.

AML speichert Planungsinformationen unter Verwendung des objektorientierten Paradigmas. Es ermöglicht die Abbildung realer Anlagenkomponenten durch Datenobjekte, in denen unterschiedliche Aspekte gekapselt sind. Ein solches Objekt kann aus anderen Unterobjekten bestehen und 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. Deshalb liegt der Schwerpunkt des Datenaustausches in der Anlagenplanung im Austausch von objektorientierten Datenstrukturen, Geometrie, Kinematik und Logik.

AML kombiniert bereits 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 ("as is") verwendet, es wurden aus ihnen keine speziellen AML-Zweige entwickelt.

Der Kern von AML ist das Toplevel-Datenformat CAEX, das die verschiedenen Datenformate miteinander verbindet. Zu diesem Zweck verfügt AML über eine inhärent verteilte Dokumentenarchitektur.

Bild 1 gibt einen Überblick über die Grundarchitektur von AML und die Verteilung der Topologie-, Geometrie, Kinematik- und Logikinformationen.

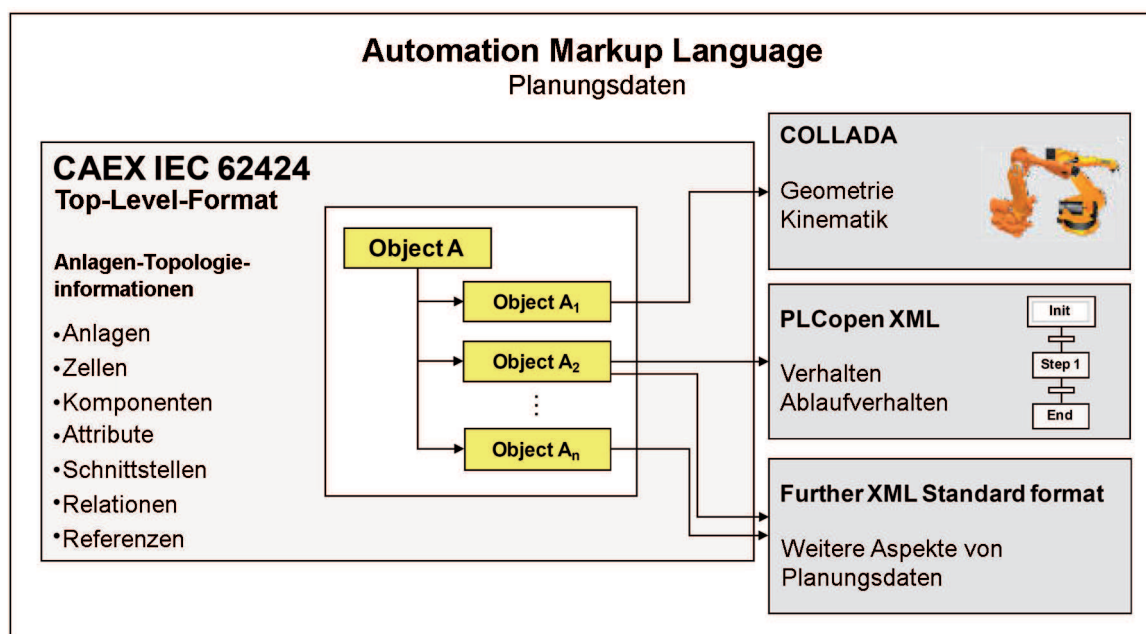


Bild 1 – Überblick über das Engineering-Datenaustauschformat AML

IEC 62714 besteht aus mehreren Teilen, die sich auf unterschiedliche Schwerpunkte von AML konzentrieren:

- IEC 62714-1: Architektur und allgemeine Anforderungen

Dieser Teil beschreibt die grundlegende Architektur von AML, die Modellierung von Planungsdaten, Klassen, Instanzen, Relationen, Referenzen, Hierarchien, grundlegende AML-Bibliotheken sowie erweiterte AML-Konzepte. Er stellt die Basis für alle weiteren bzw. künftigen Teile dar und stellt Mechanismen zur Verfügung, um noch weitere Datenformate zu referenzieren.

- 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 steuerungsrelevanten Informationen.

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

Grundsätzlich wird jedoch empfohlen, vor der Integration weiterer Datenformate die Nutzbarkeit der bereits verfügbaren Datenformate zu prüfen und die Zahl weiterer Datenformaten zu begrenzen - anderenfalls könnte der Datenaustausch durch zu viele unterstützte Datenformate behindert werden.

Anhang A enthält eine informative Einführung sowie entsprechende Anwendungsfälle und Beispiele bezüglich AML.

Anhang B enthält informativ den XML-Text der in diesem Teil der IEC 62714 definierten Bibliotheken.

## 1 Anwendungsbereich

Dieser Teil der IEC 62714 trifft allgemeine Festlegungen und definiert die Architektur von AML mit dem Ziel der Modellierung von Planungsdaten, die zwischen Planungswerkzeugen industrieller automatisierungs- und Steuerungssystemen ausgetauscht werden. Die hierin enthaltenen Vorgaben gelten für Export-/Importanwendungen von entsprechenden Planungswerkzeugen.

Dieser Teil der IEC 62714 definiert keine Details für die Prozedur des Datenaustausches und beschreibt keine Implementierungsanforderungen für Export-/Importwerkzeuge.

## 2 Normative Verweisungen

Die folgenden Dokumente, die in diesem Dokument teilweise oder als Ganzes zitiert werden, 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 62424:2008, *Representation of process control engineering – Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools*

IEC 62714 (alle Teile), *Engineering data exchange format for use in industrial systems engineering – Automation Markup Language*

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

ISO/ 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](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/>>)

PLCopen XML 2.0:December 3rd 2008 and PLCopen XML 2.0.1:May 8th 2009, *XML formats for IEC 61131-3*  
(verfügbar unter <<http://www.plcopen.org/>>)

## 3 Begriffe und Abkürzungen

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe.

### 3.1 Begriffe

#### 3.1.1

##### **AML**

XML-basiertes Datenaustauschformat für Anlagenplanungsdaten entsprechend IEC 62714

#### 3.1.2

##### **Automatisierungsobjekt**

physische oder logische Einheit im automatisierten System

ANMERKUNG 1 zum Begriff Automatisierungsobjekte sind z. B. Automatisierungskomponente, Ventile und/oder Signale.

### **3.1.3**

#### **AML-Objekt**

Datendarstellung eines Automatisierungsobjekts mit einer Relation zu einer AML-Rollenklasse

ANMERKUNG 1 zum Begriff Die AML-Objekte sind Kernelement der AML. Ein AML-Objekt stellt Instanzen dar und kann Verwaltungselemente, Attribute, Schnittstellen, Relationen und Referenzen enthalten.

### **3.1.4**

#### **AML-Klasse**

vordefinierter AML-Objektyp

ANMERKUNG 1 zum Begriff AML-Klassen werden in AML-Bibliotheken gespeichert.

ANMERKUNG 2 zum Begriff AML-Klassen definieren wiederverwendbare Musterlösungen, die durch Attribute, Schnittstellen und aggregierte Objekte charakterisiert sind.

ANMERKUNG 3 zum Begriff AML-Klassen können für Mehrfachinstanzierungen verwendet werden.

### **3.1.5**

#### **AML-Attribut**

zu einem AML-Objekt gehörende Eigenschaft

ANMERKUNG 1 zum Begriff AML-Attribute werden als XML-Elemente gemäß IEC 62424:2008 (A.2.4) beschrieben.

### **3.1.6**

#### **AML-Dokument**

bestimmtes CAEX-Dokument gemäß IEC 62714 einschließlich aller referenzierten Unterdokumente

ANMERKUNG 1 zum Begriff AML-Dokumente können als Dateien, aber auch z. B. als Zeichenfolgen oder Datenströme gespeichert werden.

### **3.1.7**

#### **AML-Datei**

bestimmte CAEX-Datei gemäß IEC 62714-1 mit der Erweiterung „.aml“, ausgenommen aller referenzierten Unterdateien

### **3.1.8**

#### **AML-Schnittstelle**

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

ANMERKUNG 1 zum Begriff Schnittstellen ermöglichen die Beschreibung von Beziehungen zwischen Objekten durch die Definition von CAEX InternalLinks. Beispiele sind Signalschnittstellen, Geräteschnittstellen oder Stromversorgungsschnittstellen.

### **3.1.9**

#### **AML-Bibliothek**

Bibliothek, die AML-Klassen enthält

### **3.1.10**

#### **AML-Port**

AML-Objekt, das eine Gruppe von Schnittstellen aufnehmen kann und das durch zusätzliche Eigenschaften charakterisiert ist

ANMERKUNG 1 zum Begriff Ports gehören zu einem AML-Elternobjekt und beschreiben komplexe Schnittstellen dieses Objekts. Ports können auf einer höheren Abstraktionsebene miteinander verbunden sein.

### **3.1.11**

#### **AML-Gruppe**

AML-Objekt, das eine alternative Sicht auf AML-Objekte zur Verfügung stellt

### 3.1.12

#### **AML-Facette**

AML-Objekt, das eine alternative Sicht auf AML-Attribute oder Schnittstellen zur Verfügung stellt

### 3.1.13

#### **CAEX**

neutrales XML-basiertes Datenformat

ANMERKUNG 1 zum Begriff Neutrales Datenformat gemäß IEC 62424:2008, Abschnitt 7, Anhang A und Anhang C.

### 3.1.14

#### **Kopie-Instanz-Beziehung**

Beziehung zwischen einer Instanz und der zugehörigen Klasse, wobei die Instanz durch Kopieren der Datenstrukturen der Klasse gebildet wird

ANMERKUNG 1 zum Begriff Die Instanz erhält eine Kopie aller Merkmale und Eigenschaften der AML-Quellenklasse. Änderungen der Klasse führen jedoch nicht zu Änderungen der Instanz. Innerhalb der Instanz werden Klasseneigenschaften individuell ausgeprägt. Weitere Kopien sind mit Kenntnis der AML-Quellenklasse möglich.

### 3.1.15

#### **universeller eindeutiger Bezeichner**

(en: Universal Unique Identifier)

#### **UUID**

eindeutiger Bezeichner für ein AML-Objekt

ANMERKUNG 1 zum Begriff Diese Anmerkung ist nur in französischer Sprache verfügbar.

### 3.1.16

#### **globaler eindeutiger Bezeichner**

(en: global unique identifier)

#### **GUID**

Implementierung einer UUID

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

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

ANMERKUNG 3 zum Begriff Diese Anmerkung ist nur in französischer Sprache verfügbar.

### 3.1.17

#### **Vererbungsbeziehung**

Beziehung zwischen zwei AML-Klassen

ANMERKUNG 1 zum Begriff Die abgeleitete Klasse erbt alle Attribute und Merkmale der Basisklasse.

### 3.1.18

#### **Instanz**

Datendarstellung einer individuellen physischen oder logischen Einheit

ANMERKUNG 1 zum Begriff Instanzen können erweitert werden, z. B. durch aggregierte Objekte oder Attribute.

### 3.1.19

#### **PropertySet**

AML-Standard-Rollenklasse, die einen Satz semantisch vordefinierter Attribute beinhaltet

### 3.1.20

#### **Topologie**

hierarchischer Aufbau eines Systems, das sich als Objektbaum darstellen lässt

ANMERKUNG 1 zum Begriff Dies schließt multiple Hierarchien, überkreuzte Strukturen und Objektnetzwerke ein.

### 3.1.21

#### Anlagen-Topologie

hierarchische Struktur einer Anlage, die sich als Objektbaum darstellen lässt

### 3.1.22

#### Publizieren

(en: publish)

Abbildung einer Datenstruktur eines externen Dokuments zur Verwendung innerhalb von CAEX

ANMERKUNG 1 zum Begriff Dies erlaubt die Festlegung von Verknüpfungen zwischen Datenstrukturen aus unabhängigen externen Dokumenten.

### 3.1.23

#### Beziehung

(en: relation)

Assoziation zwischen Objekten in CAEX

ANMERKUNG 1 zum Begriff Beispiele für Beziehungen sind Eltern-Kind-Beziehungen und Klassen-Instanz-Beziehungen

### 3.1.24

#### Link

Verbindung zwischen Objekten des Typs CAEX ExternalInterface

ANMERKUNG 1 zum Begriff Ein Link wird mithilfe eines CAEX InternalLink abgebildet.

### 3.1.25

#### Referenz

(en: reference)

Assoziation zwischen einem CAEX InternalElement und extern gespeicherten Informationen

## 3.2 Abkürzungen

**Tabelle 1 – Abkürzungen**

| Abkürzung | Englisch                            | Übersetzung/Beschreibung                                           |
|-----------|-------------------------------------|--------------------------------------------------------------------|
| AML       | Automation Markup Language          | Auszeichnungssprache für die Automation, XML-basiertes Datenformat |
| CAE       | Computer Aided Engineering          | computergestützte Planung                                          |
| CAEX      | Computer Aided Engineering eXchange | Datenformat für den rechnergestützten Datenaustausch               |
| COLLADA   | Collaborative design activity       | Geometriedatenformat der Khronos-Gruppe                            |
| GUID      | Global unique identifier            | globaler eindeutiger Bezeichner                                    |
| HMI       | Human machine interface             | Mensch-Maschine-Schnittstelle                                      |
| ID        | Identifier                          | Bezeichner, Identifikator                                          |
| PLC       | Programmable logic controller       | speicherprogrammierbare Steuerung                                  |
| URL       | Uniform resource locator            | einheitlicher Quellenanzeiger                                      |
| URI       | Uniform resource identifier         | einheitlicher Bezeichner für Ressourcen                            |
| UUID      | Universal unique identifier         | universeller eindeutiger Bezeichner                                |
| XML       | Extensible Markup Language          | erweiterbare Auszeichnungssprache, ein ASCII-basiertes Datenformat |

## 4 Konformität

Um Konformität mit diesem Teil der IEC 62714 in Bezug auf die Unterstützung von AML zu beanspruchen, müssen die Anforderungen der Abschnitte 5, 6, 7 und 8 erfüllt sein.

## 5 Spezifikation der AML-Architektur

### 5.1 Allgemeines

Den Kern von AML bildet das Toplevel-Datenformat CAEX, ein neutrales Datenformat gemäß IEC 62424:2008, Abschnitt 7, Anhang A und Anhang C, mit dessen Hilfe etablierte Datenformate für die Planungsaspekte Topologie, Geometrie, Kinematik, Verhalten und Abfolgeinformationen miteinander verknüpft werden. Deshalb ist ein wesentliches Merkmal von AML eine inhärente verteilte Dokumentenarchitektur, die auf die o. g. Planungsaspekte fokussiert.

Abbildungen dienen nur der Erläuterung. Die grafischen Darstellungen sind nicht normativ.

### 5.2 Allgemeine AML-Architektur

Bezüglich der allgemeinen AML-Architektur gelten die folgenden Festlegungen:

**Informationen zur Anlagentopologie:** Die Anlagentopologie fungiert als die Toplevel-Datenstruktur der Information der Anlagenplanung und muss mit dem Datenformat CAEX gemäß IEC 62424:2008, Abschnitt 7, Anhang A und Anhang C, modelliert werden. Eventuelle semantische Erweiterungen von CAEX werden gesondert beschrieben. Multiple und überkreuzte Hierarchiestrukturen müssen im Sinne des Konzepts des Spiegelobjekts (en: mirror object) gemäß IEC 62424:2008, A.2.14, verwendet werden. Spiegelobjekte dürfen nicht verändert werden, sämtliche Änderungen müssen am Master-Objekt vorgenommen werden.

ANMERKUNG 1 Gemäß IEC 62424:2008, A.2.14, wird ein AML-Objekt mit einem Verweis zu einem anderen AML-Objekt als "Spiegelobjekt" bezeichnet, während das referenzierte Objekt als "Master-Objekt" bezeichnet wird. Das Spiegel-Objekt gilt als identisch mit dem Master-Objekt. Dies ermöglicht die Platzierung einer Objektinstanz in verschiedenen Anlagenhierarchien und damit die Modellierung von komplexen Objektnetzwerken mit überkreuzten Strukturen.

ANMERKUNG 2 Die IEC 62714 ändert nicht die Syntax des CAEX-Datenformats. Ein informativer Überblick und weitere Beispiele zur Anlagentopologie finden sich in A.1.2 und in IEC 62424:2008, Anhang D.

**Referenz- und Relationsinformationen:** Referenzen (en: references) und Relationen (en: relations) müssen gemäß 5.6 und 5.7 gespeichert werden. Relationen zwischen extern gespeicherten Informationen müssen unter Anwendung von CAEX-Mechanismen gespeichert werden. Falls erforderlich, müssen dazu die entsprechenden „Verknüpfungspartner“ in der CAEX-Anlagentopologiebeschreibung mit Hilfe von CAEX ExternalInterfaces publiziert werden. Diese müssen aus den in 6.3 festgelegten AML-Standardschnittstellenklassen abgeleitet werden.

ANMERKUNG 3 Referenzen beschreiben Verknüpfungen von CAEX-Objekten mit extern gespeicherten Informationen. Ein informativer Überblick über Relationen findet sich in A.1.5. Referenzen und die Publikation von Schnittstellen sind in zusätzlichen Teilen der IEC 62714 beschrieben.

ANMERKUNG 4 Relationen beschreiben Zuordnungen zwischen CAEX-Objekten.

**Geometrie- und Kinematikinformationen:** Informationen bezüglich Geometrie und Kinematik müssen unter Verwendung des Datenformats COLLADA™<sup>1</sup> gespeichert werden. COLLADA-Schnittstellen, die innerhalb des Toplevel-Formats miteinander verbunden werden sollen, müssen als CAEX ExternalInterfaces publiziert werden.

ANMERKUNG 5 Die IEC 62714 ändert nicht die Syntax des COLLADA-Datenformats. Ein überblickartiges Beispiel zur Referenzierung von COLLADA findet sich in A.1.3. Einzelheiten dazu sind in der IEC 62713-3 vorgesehen.

---

<sup>1</sup> COLLADA ist eingetragenes Warenzeichen eines Produktes der Khronos Gruppe. Diese Information wird für die Unterrichtung der Nutzer des Dokuments gegeben und bedeutet keine Anerkennung des genannten Produktes durch die IEC. Ähnliche Produkte können, wenn sie zu gleichen Ergebnissen führen, ebenso verwendet werden.

ANMERKUNG 6 Aus den COLLADA-Geometrieinformationen verschiedener Objekte kann automatisch eine vollständige Szene abgeleitet werden. Auf diese Dateien kann von CAEX aus verwiesen werden, und sie können mit Hilfe von CAEX-Verknüpfungsmechanismen miteinander verknüpft sein bzw. werden.

**Logikinformationen:** Logikinformationen müssen unter Verwendung des Datenformats PLCopen XML gespeichert werden. Falls Logikelemente, wie z. B. Variablen oder Signale, innerhalb des Toplevel-Formats miteinander verbunden werden sollen, müssen sie als CAEX ExternalInterface publiziert werden. Alle Elemente von PLCopen XML, die innerhalb des Toplevel-Formats veröffentlicht werden, müssen innerhalb von PLCopen XML über eine eindeutige Kennung (ID) verfügen.

ANMERKUNG 7 Logikinformationen beschreiben Abfolgen von Aktionen und das interne Verhalten von Objekten, einschließlich E/A-Verbindungen und logischer Variablen. Die IEC 62714 modifiziert nicht das PLCopen XML-Format. Ein informativer Überblick zur Referenzierung von Logikinformation findet sich in A.1.4. Einzelheiten dazu sind in der IEC 62714-4 vorgesehen.

**Referenzierung anderer Datenformate:** IEC 62714 kann künftig um weitere Teile ergänzt werden, die die Einbindung weiterer XML-Datenformate unter Verwendung der AML-Referenzmechanismen festlegen. Einzelheiten können in weiteren Teilen dieser Norm festgelegt werden.

Das Datenformat AML stellt weder Konsistenzüberprüfungen bezüglich CAEX Constraints, Attributwerten, Relationen oder Referenzen noch Überprüfungen der semantischen Richtigkeit enthaltener Daten zur Verfügung: Diese Überprüfungen fallen vielmehr in den Aufgabenbereich des Quell- oder Zielwerkzeugs oder der entsprechenden Import-/Exportanwendung. AML ermöglicht lediglich eine syntaktische Prüfung des Dokuments gegen das zugehörige Schema.

### 5.3 AML-Dokumentversion

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

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

Entsprechend gelten die folgenden Festlegungen:

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

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

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

#### Bild 2 – AML-Dokumentversions-Information

- Jede AML-Standardbibliothek und jede benutzerdefinierte Bibliothek muss ihre jeweilige Versionsnummer unter Verwendung des CAEX-Elements „Version“ definieren. Die Syntax des Werts der Versionsnummer wird im vorliegenden Teil der IEC 62714 nicht festgelegt.

- Falls erforderlich, müssen CAEX-Klassen ihre Versionsnummer unter Verwendung des CAEX-Elements „Version“ definieren. Die Syntax und Semantik der Versionsnummer von Klassen innerhalb einer Bibliothek wird im vorliegenden Teil der IEC 62714 nicht festgelegt.
- Es ist verboten, gleiche Bibliotheken unterschiedlicher Versionen in derselben AML-Datei zu speichern.

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

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

Die IEC 62714 basiert auf den folgenden Datenformaten:

- CAEX Version 2.15,
- PLCopenXML 2.0 und 2.0.1,
- COLLADA 1.5.0 gemäß ISO/PAS 17506 und COLLADA 1.4.1,
- AML-Standardbibliotheken gemäß Teil 1 und anderen Teilen der IEC 62714.

#### **5.4 Metainformationen über das AML-Quell-Werkzeug**

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

- Jedes AML-Dokument muss Informationen über das Werkzeug zur Verfügung stellen, welches das AML-Dokument erstellt hat.
- In einer Datenaustausch-Werkzeugkette müssen alle beteiligten Werkzeuge diese Informationen in derselben Weise im CAEX-Dokument speichern. Daher kann das Dokument Informationen über mehrere Werkzeuge einer Datenaustausch-Werkzeugkette enthalten. Ein Werkzeug kann die Erstellereinformation anderer Werkzeuge entfernen. Dies kann jedoch den iterativen Datenaustausch mit den anderen Werkzeugen behindern, daher wird davon abgeraten, die Erstellereininformation anderer Werkzeuge zu entfernen.
- Diese Information muss als Teil der CAEX AdditionalInformation des Wurzelobjekts des CAEX-Dokuments gespeichert werden.
- Der AdditionalInformation-Block muss „WriterHeader“ heißen.
- Die Metainformationen müssen folgende Angaben enthalten:
  - Name des exportierenden Programms, das Erstellerwerkzeug,
  - ID des Erstellerwerkzeugs (diese muss unverändert bleiben),
  - Hersteller (en: vendor) des Erstellerwerkzeugs,
  - URL des Erstellerwerkzeugs,
  - Produktversion des Erstellerwerkzeugs,
  - Produkt-Release-Nummer des Erstellerwerkzeugs,
  - Datum der letzten Änderung durch den Ersteller,
  - Projekttitel des Quellprojekts,
  - Projektkennung (ID) des Quellenprojekts.
- Der Inhalt der Metainformationen muss durch das Erstellerwerkzeug festgelegt werden und dem Typ xs:string entsprechen.
- Die geforderten Informationen sind unter Verwendung der in Tabelle 2 angegebenen Attribute zu speichern.

**Tabelle 2 – Metainformationen über das AML-Quellwerkzeug**

| XML-Tag-Name        | Typ         | Stufe              | Beispiel                      |
|---------------------|-------------|--------------------|-------------------------------|
| WriterName          | xs:string   | vorgeschrieben     | "ToolX AML Exporter"          |
| WriterID            | xs:string   | vorgeschrieben     | "ToolXToAML123"               |
| WriterVendor        | xs:string   | vorgeschrieben     | "ToolX Vendor"                |
| WriterVendorURL     | xs:string   | vorgeschrieben     | "http://www.ToolX-Vendor.org" |
| WriterVersion       | xs:string   | vorgeschrieben     | "0.2"                         |
| WriterRelease       | xs:string   | vorgeschrieben     | "123 prealpha"                |
| LastWritingDateTime | xs:DateTime | vorgeschrieben     | "2011-05-25T09:30:47"         |
| WriterProjectTitle  | xs:string   | zusätzlich möglich | "eCarproduction"              |
| WriterProjectID     | xs:string   | zusätzlich möglich | "eCarproduction_LinePLC.prj"  |

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

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

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

```

<AdditionalInformation>
  <WriterHeader>
    <WriterName>ToolX AML Exporter</WriterName>
    <ToolWriterID>ToolXToAML123</ToolWriterID>
    <WriterVendor>ToolX Vendor</WriterVendor>
    <WriterVendorURL>http://www.ToolX-Vendor.org</WriterVendorURL>
    <WriterVersion>0.1</WriterVersion>
    <WriterRelease>123 prealpha</WriterRelease>
    <LastWritingDateTime>2013-12-31T12:00:00</LastWritingDateTime>
    <WriterProjectTitle>eCarproduction</WriterProjectTitle>
    <WriterProjectID>eCarproduction_LinePLC.prj</WriterProjectID>
  </WriterHeader>
</AdditionalInformation>

```

**Bild 3 – XML-Text der AML-Information zum Quellwerkzeug**

## 5.5 Objekt-Identifikation

AML folgt dem objektorientierten Paradigma. Alle Planungsinformationen werden entweder als Objekt modelliert oder gehören zu einem Objekt. In einer heterogenen Werkzeug-Landschaft verwenden unterschiedliche Planungswerkzeuge jedoch unterschiedliche Konzepte für die Identifikation von Objekten, z. B. einen eindeutigen Namen, einen eindeutige Identifikator (en: ID) oder einen eindeutigen Pfad. Einige Werkzeuge gestatten eine oder mehrere Änderungen der Identifikatoren während der Lebenszeit, während andere dies nicht zulassen. Die IEC 62714 ermöglicht den Datenaustausch zwischen verschiedenen Planungswerkzeugen mit derartigen individuellen Objektkennungskonzepten. AML neutralisiert die beschriebene Vielfalt und definiert ein einheitliches und verbindlich vorgeschriebenes Objekt-Identifikationskonzept.

Im Hinblick auf die Objekt-Identifikation gelten die folgenden Festlegungen:

- In Übereinstimmung mit IEC 62424:2008, A.2.2.1, müssen AML-Klassen (CAEX RoleClasses, CAEX InterfaceClasses, CAEX SystemUnitClasses) durch ihren CAEX-Tag „Name“ identifiziert werden.
- Dieser Name muss innerhalb einer Hierarchiestufe der entsprechenden Bibliothek über die gesamte Lebenszeit der Klasse eindeutig sein.
- In Übereinstimmung mit IEC 62424:2008, A.2.8, muss die Referenzierung von Klassen über vollständige Pfade unter Verwendung der entsprechenden Pfad-Separatoren erfolgen.
- Alle AML-Objektinstanzen (CAEX InternalElements sowie CAEX ExternalInterfaces) müssen durch ihren CAEX-Tag „ID“ identifiziert werden. Dieser Identifikator muss eine UUID gemäß ISO/IEC 9834-8 sein.

ANMERKUNG 1 Eine mögliche Implementierung der UUID ist der globale eindeutige Bezeichner (GUID).

ANMERKUNG 2 Im Gegensatz zu diesem Teil der IEC 62714 ist der „ID“-Tag in IEC 62424:2008, A.3.15, nicht verbindlich vorgeschrieben.

ANMERKUNG 3 In diesem Teil der IEC 62714 werden UUIDs in Kurzform dargestellt, z. B. als „GUID1“, „GUID2“ usw.

ANMERKUNG 4 Der CAEX-Tag „Name“ ist nur ein Anzeigenamen, er besitzt lediglich informativen Charakter und kann sich mit der Zeit oder über Werkzeuge hinweg ändern.

- Nach seiner Erzeugung darf sich die UUID während der gesamten Lebensdauer des betreffenden Objekts innerhalb aller beteiligten Werkzeuge nicht mehr ändern.
- Beim Referenzieren von Instanzen muss der „ID“-Wert verwendet werden.
- Die Referenzierung von CAEX-Schnittstellen muss mit Hilfe der UUID des Elternobjektes der betreffenden Schnittstelle, ergänzt um das Trennzeichen „.“ und dem Namen der Schnittstelleninstanz erfolgen.

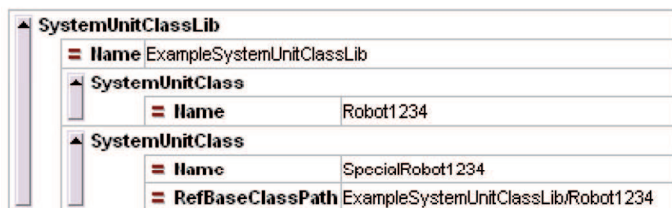
BEISPIEL 1 GUID:out“.

- Die Referenzierung von CAEX-Attributen muss mit Hilfe der UUID des Elternobjektes des betreffenden Attributes, ergänzt um das Trennzeichen „.“ und dem Namen des Attributes erfolgen. Sollte es sich bei dem Attribut um ein verschachteltes Attribut handeln, folgt dem Trennzeichen der Unterpfad des Attributes.

BEISPIEL 2 GUID.Colour“.

BEISPIEL 3 GUID.Colour.red“.

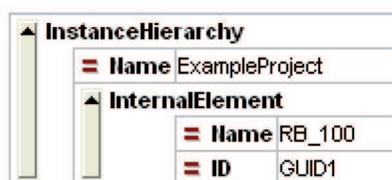
Bild 4 zeigt ein Beispiel für eine SystemUnitClassLib mit der SystemUnitClass „Robot1234“ sowie einer weiteren SystemUnitClass „SpecialRobot1234“, die von „Robot1234“ abgeleitet ist.



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

**Bild 4 – Beispiel für die Objekt-Identifizierung einer AML-Klasse**

Bild 5 zeigt ein Beispiel für eine InstanceHierarchy mit einem Objekt „RB\_100“, das einen eindeutigen Identifikator besitzt, der durch die Zeichenfolge „GUID1“ dargestellt wird.



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

**Bild 5 – Beispiel für die Objekt-Identifizierung einer AML-Objektinstanz**

## 5.6 Festlegungen für AML-Relationen

### 5.6.1 Allgemeines

Die Ausrichtung auf Objekte macht es notwendig, Mechanismen festzulegen, mit deren Hilfe sich Objekte in Beziehung zueinander setzen lassen. Der vorliegende Teil der IEC 62714 unterscheidet zwei Mechanismen für die Speicherung dieser Informationen: Referenzen und Relationen. Unterabschnitt 5.6 behandelt Relationen, während Referenzen in 5.7 behandelt werden. Ein informativer Überblick über Relationen und Referenzen findet sich in A.1.5.

### 5.6.2 Eltern-Kind-Beziehungen zwischen AML-Objekten

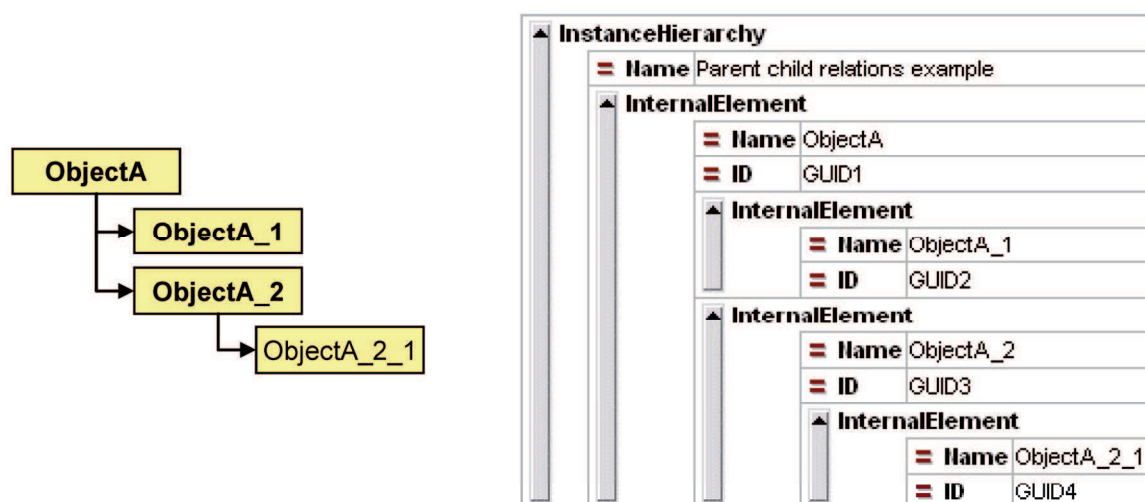
Eltern-Kind-Beziehungen zwischen Objektinstanzen werden zur Darstellung von hierarchischen Objektstrukturen verwendet und beschreiben eine „besteht-aus“-Beziehung.

Für Eltern-Kind-Beziehungen zwischen AML-Objekten gilt die folgende Festlegung:

- Die Speicherung von Hierarchien muss gemäß IEC 62424:2008, Anhang A, z. B. A.2.11, erfolgen.

ANMERKUNG Neben einfachen Hierarchien können gemäß IEC 62424:2008, A.2.14, auch überkreuzte Hierarchien (Objektnetzwerke) gespeichert werden.

Bild 6 zeigt ein Beispiel für eine Objekthierarchie und ihre Datenabbildung.



```
<InstanceHierarchy Name="Parent child relations example">
  <InternalElement Name="ObjectA" ID="GUID1">
    <InternalElement Name="ObjectA_1" ID="GUID2"/>
    <InternalElement Name="ObjectA_2" ID="GUID3">
      <InternalElement Name="ObjectA_2_1" ID="GUID4"/>
    </InternalElement>
  </InternalElement>
</InstanceHierarchy>
```

**Bild 6 – Beispiel einer Eltern-Kind-Beziehung zwischen AML-Objekten**

### 5.6.3 Eltern-Kind-Beziehungen zwischen AML-Klassen

Für Eltern-Kind-Beziehungen zwischen AML-Klassen gelten folgende Festlegungen:

- Eine Eltern-Kind-Beziehung zwischen AML-Klassen darf nur ihre hierarchische Nachbarschaft beschreiben. Dies erlaubt die Definition beliebiger nutzerdefinierter hierarchischer Strukturen.
- Diese Relation hat keine weitere Semantik.

**ANMERKUNG** Eine Eltern-Kind-Relation ist nicht als Vererbungsbeziehung zu interpretieren. Vererbungsbeziehungen werden explizit wie in 5.6.4 beschrieben modelliert.

Bild 7 zeigt ein Beispiel einer Eltern-Kind-Beziehung zwischen den Klassen „ParentClass“ und „ChildClass“. Die Klasse „ChildClass“ hat keine Vererbungsbeziehung zu seinem Elternobjekt.

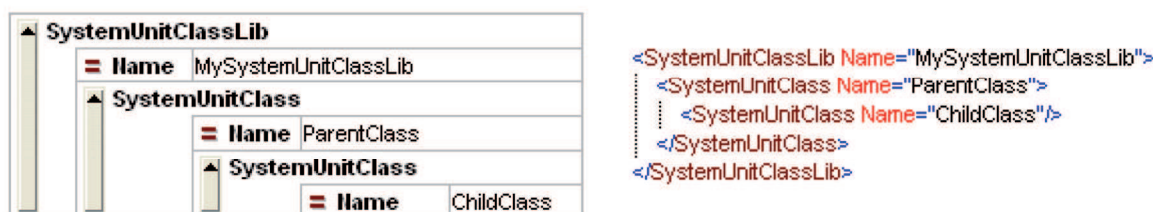


**Bild 7 – Beispiel einer Eltern-Kind-Beziehung zwischen Klassen**

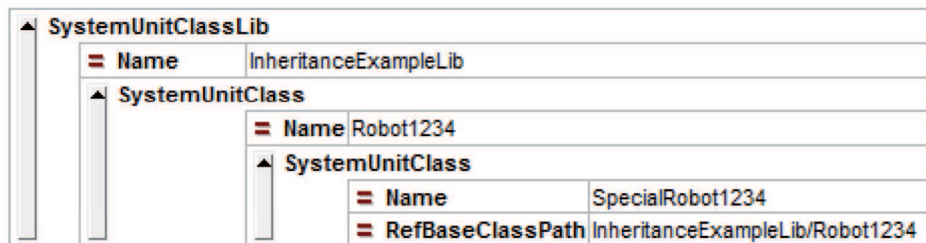
### 5.6.4 Vererbungsbeziehungen

Für Vererbungsbeziehungen gelten die folgenden Festlegungen:

- Vererbung zwischen Klassen muss gemäß IEC 62424:2008, A.2.7, definiert werden.

- Ist Vererbung erforderlich, muss die Elternklasse mit ihrem vollständige Pfad gemäß IEC 62424:2008 A.2.7, unter Verwendung des CAEX-Tag „RefBaseClassPath“ angegeben werden.
- Wenn sich die gewünschte Elternklasse eine Hierarchieebene über der betrachteten Kindklasse befindet, kann die Elternklasse durch Speicherung des Namens der Elternklasse im CAEX-Tag „RefBaseClassPath“ festgelegt werden, ohne den vollständigen Pfad anzugeben.

Bild 8 zeigt ein Beispiel für die Klasse „Robot1234“ sowie einer weiteren Klasse „SpecialRobot1234“, die von „Robot1234“ abgeleitet ist.



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

**Bild 8 – Beispiel für eine Vererbungsbeziehung zwischen zwei Klassen**

In Ergänzung zu diesem Beispiel kann der CAEX-Tag „RefBaseClassPath“ den Wert „InheritanceExampleLib/Robot1234“ oder alternativ „Robot1234“ enthalten, da sich die Elternklasse eine Hierarchieebene über der Klasse „SpecialRobot1234“ befindet.

### 5.6.5 Klassen-Instanz-Beziehungen

Instanzen sind durch einen eindeutigen Identifikator und einen Satz Parametern charakterisiert. Es gelten die folgenden Festlegungen:

- Ein AML-Objekt muss als CAEX InternalElement als Bestandteil der CAEX-InstanceHierarchy oder einer SystemUnitClass modelliert werden.
- Ein AML-Objekt kann ein Einzelobjekt (en: singleton) ohne Beziehung zu irgendeiner SystemUnitClass sein.

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

ANMERKUNG 2 Instanzen ohne eine Beziehung zur AutomationMLBaseRole sind zulässig, es handelt sich dann um benutzerdefinierte Objekte. Diese sind jedoch keine AML-Objekte.

- Hat ein AML-Objekt eine Klassen-Instanz-Beziehung zu einer SystemUnitClass, so ist es beim Erzeugen durch Kopieren dieser SystemUnitClass einschließlich ihrer internen Architektur und alle geerbten Informationen zu erstellen.

ANMERKUNG 3 Eine SystemUnitClass fungiert somit als Vorlage (en: template). Änderungen an der SystemUnitClass führen jedoch nicht automatisch zu Änderungen in den entsprechenden Automatisierungsobjekten. Darüber hinaus kann das Automatisierungsobjekt auch ohne Klasseninformationen übermittelt werden, es beinhaltet selbst alle zugehörigen Informationen.

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

ANMERKUNG 4 Die Quellklasse soll als geeigneter Startpunkt für die Modellierung der Instanz dienen.

- Die kopierte Quellklasse muss zur späteren Verwendung im CAEX-Tag „RefBaseSystemUnitPath“ der Instanz vermerkt werden. Dieser Vermerk muss den vollständigen Pfad und den Namen der Quellklasse umfassen.

ANMERKUNG 5 Wenn sich die Quellklasse einer Instanz ändert, hat dies keine Änderung der Instanz zur Folge. Die Aktualisierung von Instanzen ist eine mögliche Funktionalität eines Werkzeugs, was jedoch außerhalb des Anwendungsbereichs der IEC 62714-1 liegt.

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

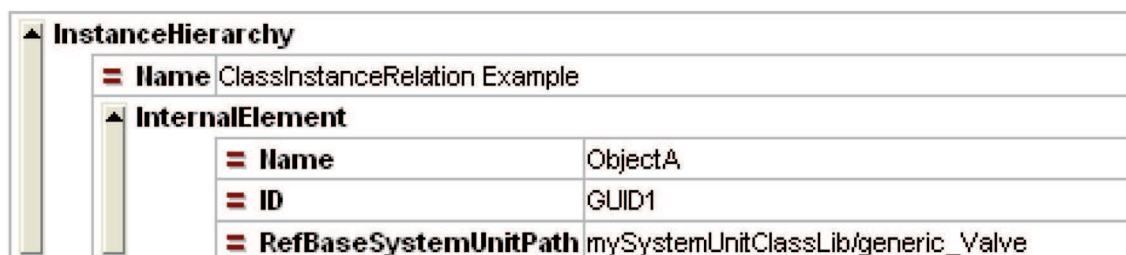
ANMERKUNG 6 Diese Festlegung unterstützt die Verfolgung von Änderungen über unterschiedliche Versionen einer Klasse.

- Vererbung zwischen SystemUnitClasses und Objektinstanzen ist nicht erlaubt.

ANMERKUNG 7 Eine Instanz kann nur eine Kopie ihrer Klasse sein. Dies ist eine Einschränkung gegenüber der IEC 62424:2008, A.2.7. Vererbung zwischen Klassen und Instanzen kann Bestandteil einer zukünftigen Erweiterung sein.

- Die Beziehung zwischen einer Instanz und einer Rollenklasse ist gemäß IEC 62424:2008, A.2.7, im Attribut „RefBaseRoleClassPath“ des zugehörigen RoleRequirement zu speichern. Im Gegensatz zu IEC 62424:2008, A.2.7, ist keine Vererbung zulässig, es müssen vielmehr alle Rollenklassenspezifikationen in das entsprechende AML-Objekt kopiert werden.
- Die Beziehung zwischen einem CAEX ExternalInterface und einer CAEX InterfaceClass muss gemäß IEC 62424:2008, A.2.7 modelliert werden. Im Gegensatz zu IEC 62424:2008 ist jedoch keine Vererbung zulässig, es müssen vielmehr alle Schnittstellenklassenspezifikationen in das entsprechende AML-Objekt kopiert werden.

Bild 9 zeigt ein Beispiel einer Klassen-Instanz-Beziehung zwischen dem Objekt „ObjectA“ und einer benutzerdefinierten SystemUnitClass „generic\_Valve“.



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

**Bild 9 – Beispiel einer Klassen-Instanz-Beziehung**

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

- Der Tag „RefBaseSystemUnitPath“ kann nach dem Spiegel-Konzept von IEC 62424:2008, A.2.14, anstelle einer SystemUnitClass auf eine andere Objektinstanz verweisen.

### 5.6.6 Instanz-Instanz-Beziehungen

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

Für Instanz-Instanz-Beziehungen gelten die folgenden Festlegungen:

- Instanz-Instanz-Beziehungen sind gemäß IEC 62424:2008, A.2.5.3 sowie A.2.14, unter Verwendung von CAEX InternalLinks zu speichern.
- CAEX InternalLinks sollten an demjenigen CAEX InternalElement gespeichert werden, das das nächste gemeinsame Elternteil der miteinander zu verbindenden CAEX-Objekte ist.

- Instanz-Instanz-Beziehungen dürfen nur zwischen CAEX ExternalInterfaces definiert werden, die zu den zu verbindenden AML-Objekten gehören. Dies entspricht IEC 62424:2008, A.2.3.1.
- Die externen Schnittstellen sollten direkt oder indirekt von einer der AML-Standard-Schnittstellenklassen abgeleitet werden.

ANMERKUNG 1 Die AML-Standard-Schnittstellenklassenbibliothek ist in 6.3 festgelegt. Die Schnittstellenklassen legen die Semantik der Schnittstelle und damit die Semantik der Verknüpfung fest. Eine Verknüpfung zwischen Schnittstellen ohne Verweis auf eine Schnittstellenklasse hat keine Semantik.

- COLLADA-Dokumente können miteinander verknüpft werden. Die entsprechenden COLLADA-Schnittstellen können beliebige Elemente sein, die über einen gültigen URI (Uniform Resource Identifier, d.h. einen einheitlichen Bezeichner für Ressourcen) verfügen. Falls diese Knoten mit Hilfe von CAEX miteinander verknüpft werden sollen, müssen sie in CAEX durch Hinzufügung eines CAEX ExternalInterfaces zu dem betreffenden Objekt publiziert werden. Dieses ExternalInterface muss von der AML-Standard-InterfaceClass „COLLADAInterface“ oder einer ihrer Ableitungen abgeleitet werden.

ANMERKUNG 2 Die Standard-Schnittstellenklasse „COLLADAInterface“ ist in 6.3.7 festgelegt. Einzelheiten hierzu sind in Teil 3 der IEC 62714 vorgesehen.

- PLCopen-XML-Dokumente können durch Verwendung entsprechender PLCopen-XML-Schnittstellen miteinander verknüpft werden. Falls PLCopen-XML-Elemente mit Hilfe von CAEX miteinander verknüpft werden sollen, müssen sie durch Hinzufügung eines CAEX ExternalInterface zu dem betreffenden Objekt publiziert werden. Diese ExternalInterface muss von der AML-Standard-Schnittstellenklasse „PLCopenXMLInterface“ oder einer ihrer Ableitungen abgeleitet werden.

ANMERKUNG 3 Die Standard-Schnittstellenklasse „PLCopenXMLInterface“ ist in 6.3.8 festgelegt. Einzelheiten hierzu sind in Teil 4 der IEC 62714 vorgesehen.

Bild 10 a) zeigt dies am Beispiel eines Roboters „Rob1“ sowie einer SPS „PLC1“, die jeweils mit einer Signalschnittstelle ausgestattet sind und die miteinander verbunden sind. Bild 10 b) stellt dieses Beispiel als Objekthierarchie dar.

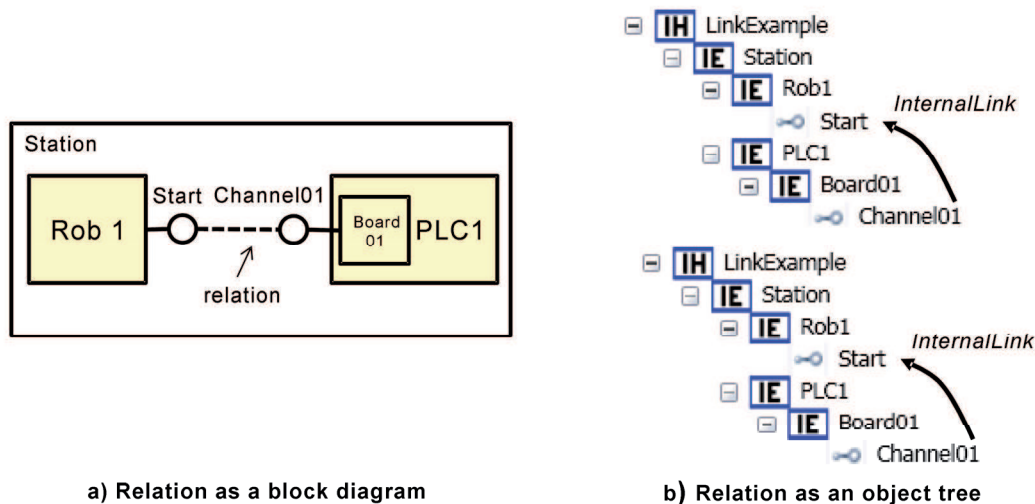
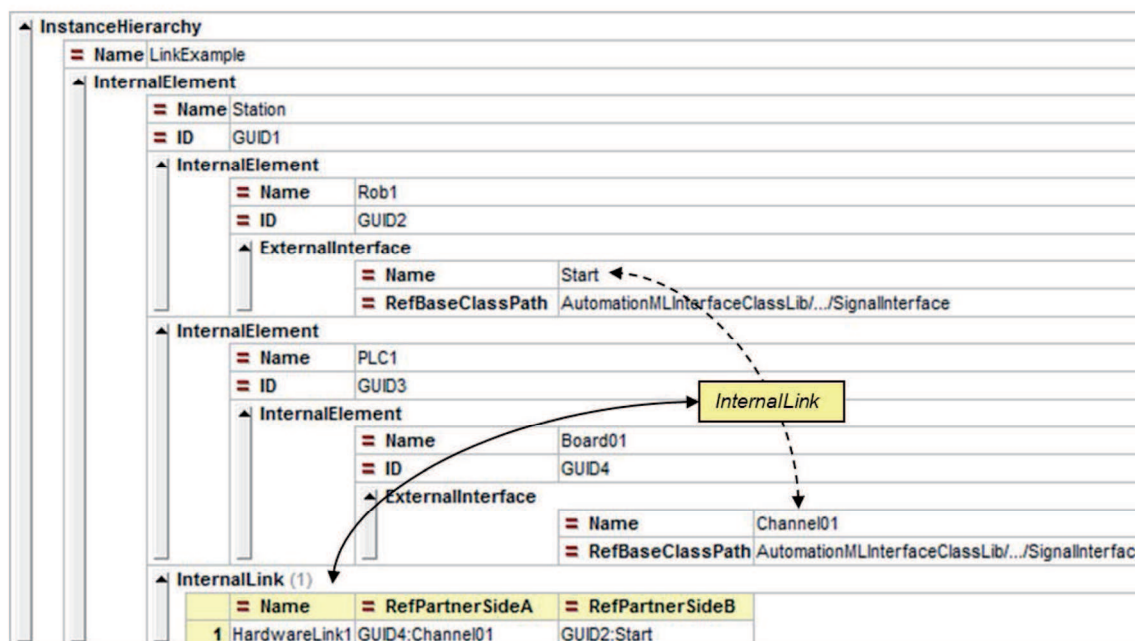


**Bild 10 – Beispiel einer Beziehung als a) Blockdiagramm und b) als Objektbaum**

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

ANMERKUNG 4 Die in diesem Beispiel enthaltenen Pfadzeichenfolgen wurden im Sinne der besseren Lesbarkeit unter Verwendung von „/.../“ verkürzt.



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

<InstanceHierarchy Name="LinkExample">
  <InternalElement Name="Station" ID="GUID1">
    <InternalElement Name="Rob1" ID="GUID2">
      <ExternalInterface Name="Start" RefBaseClassPath="AutomationMLInterfaceClassLib/.../SignalInterface"/>
    </InternalElement>
    <InternalElement Name="PLC1" ID="GUID3">
      <InternalElement Name="Board01" ID="GUID4">
        <ExternalInterface Name="Channel01" RefBaseClassPath="AutomationMLInterfaceClassLib/.../SignalInterface"/>
      </InternalElement>
    </InternalElement>
    <InternalLink Name="HardwareLink1" RefPartnerSideA="GUID4:Channel01" RefPartnerSideB="GUID2:Start"/>
  </InternalElement>
</InstanceHierarchy>
```

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

## 5.7 Festlegungen für AML-Dokumentreferenzen

### 5.7.1 Allgemeines

Eine Dokumentreferenz dient zur Verknüpfung eines AML-Objekts mit einem externen Dokument, das beispielsweise Geometrie-, Kinematik- oder Abfolgeinformationen enthalten kann. Der Referenzmechanismus basiert auf der AML-Standardschnittstelle „ExternalDataConnector“ (externer Datenverbinder) oder einer ihrer Ableitungen.

### 5.7.2 Referenzierung von COLLADA-Dokumenten

Die Referenzierung von COLLADA-Dokumenten muss unter Verwendung der AML-Standard-Schnittstellenklasse „COLLADAInterface“ oder einer ihrer Ableitungen erfolgen. Diese Klasse ist in 6.3.7 festgelegt. Einzelheiten hierzu sind in IEC 62714-3 vorgesehen.

### 5.7.3 Referenzierung von PLCopen-XML-Dokumenten

Die Referenzierung von PLCopen XML muss unter Verwendung der AML-Standard-Schnittstelle „PLCopenXMLInterface“ oder einer ihrer Ableitungen erfolgen. Diese Klasse ist in 6.3.8 festgelegt. Einzelheiten hierzu sind in IEC 62714-4 vorgesehen.

### 5.7.4 Referenzierung weiterer Dokumente

In zukünftigen Erweiterungen der IEC 62714 können weitere Schnittstellentypen für die Referenzierung weiterer Arten von Dokumenten hinzugefügt werden. Diese sind in gesonderten Teilen der IEC 62714 festgelegt und fallen nicht in den Anwendungsbereich des vorliegenden Teils. Für diese Erweiterungen gelten die folgenden Festlegungen:

- Falls weitere Arten von Dokumenten in der IEC 62714 aufgenommen werden sollen, müssen sie mit Hilfe zusätzlicher Schnittstellenklassen modelliert werden.
- Diese zusätzlichen Schnittstellen müssen als Erweiterung der AML-Schnittstellenklassenbibliothek modelliert werden und müssen direkt oder indirekt aus der Standard-Schnittstellenklasse „ExternalDataConnector“ abgeleitet werden.
- Die Speicherung von Referenzen sollte unter Verwendung derselben Standardattribute erfolgen, die von den Standard-Schnittstellenklassen zur Verfügung gestellt werden.
- Die Standard-Schnittstellenklasse „ExternalDataConnector“ darf nur für die in der IEC 62714 umfassten Dokumentenarten verwendet werden.

## 6 AML-Basis-Bibliotheken

### 6.1 Allgemeines

Abschnitt 6 definiert grundlegende AML-Basisbibliotheken mit AML-Basisklassen, die zur Modellierung von AML-Kernkonzepten benötigt werden. Alle beschriebenen Attribute sind Teil der AML-Standardbibliothek, dürfen jedoch, falls sie nicht benötigt werden, aus der Instanzhierarchie entfernt werden.

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

### 6.2 Allgemeine Festlegungen

In Bezug auf AML-Basisbibliotheken treffen folgende Festlegungen zu:

- Alle AML-Objekte müssen direkt oder indirekt von der Rollenklasse „AutomationMLBaseRole“ abgeleitet werden.
- Alle Schnittstellen müssen direkt oder indirekt von einer AML-Schnittstellenklasse abgeleitet werden.
- Bei Bedarf müssen AML-Attribute verwendet werden, wenn sie hingegen nicht benötigt werden, dürfen sie aus den AML-Objekten entfernt werden.