

DIN EN IEC 62714-4



ICS 25.040.40; 35.060; 35.240.50

**Datenaustauschformat für Planungsdaten industrieller
Automatisierungssysteme –
Automation Markup Language (AML) –
Teil 4: Logik
(IEC 62714-4:2020 + COR1:2020);
Deutsche Fassung EN IEC 62714-4:2020 + AC:2020**

Engineering data exchange format for use in industrial automation systems engineering –
Automation markup language –
Part 4: Logic
(IEC 62714-4:2020 + COR1:2020);
German version EN IEC 62714-4:2020 + AC:2020

Format d'échange de données pour une utilisation dans l'ingénierie des systèmes
d'automatisation industrielle –
Automation markup language –
Partie 4: Logique
(IEC 62714-4:2020 + COR1:2020);
Version allemande EN IEC 62714-4:2020 + AC:2020

Gesamtumfang 119 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE

Anwendungsbeginn

Anwendungsbeginn für die von CENELEC am 2020-07-21 angenommene Europäische Norm als DIN-Norm ist 2021-06-01.

Diese Norm enthält die Berichtigung EN IEC 62714-4:2020/AC:2020, die das Corrigendum 1:2020 zu IEC 62714-4:2020 übernimmt.

Nationales Vorwort

Vorausgegangener Norm-Entwurf: E DIN EN IEC 62714-4:2019-05.

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.

Aktuelle Informationen zu diesem Dokument können über die Internetseiten von DKE (www.dke.de) und DIN (www.din.de) durch eine Suche nach der Dokumentennummer aufgerufen werden.

Die enthaltene IEC-Publikation wurde vom IEC/SC 65E „Devices and integration in enterprise systems“ erarbeitet und unterliegt dem Copyright der IEC .

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

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

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.

AC

Die Änderung AC wurde eingearbeitet und mit einer senkrechten Linie und AC am linken Textrand gekennzeichnet.

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 (AML) –
Teil 4: Logik
(IEC 62714-4:2020 + COR1:2020)

Engineering data exchange format for use in
industrial automation systems engineering –
Automation markup language –
Part 4: Logic
(IEC 62714-4:2020 + COR1:2020)

Format d'échange de données pour une
utilisation dans l'ingénierie des systèmes
d'automatisation industrielle –
Automation markup language –
Partie 4: Logique
(IEC 62714-4:2020 + COR1:2020)

Diese Europäische Norm wurde von CENELEC am 2020-07-21 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.

Die Berichtigung AC:2020 tritt am 04. Dezember 2020 zur Einarbeitung in die deutsche Fassung der EN in Kraft.

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, Estland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Kroatien, Lettland, Litauen, Luxemburg, Malta, den Niederlanden, Norwegen, Österreich, Polen, Portugal, der Republik Nordmazedonien, Rumänien, Schweden, der Schweiz, Serbien, der Slowakei, Slowenien, Spanien, der Tschechischen Republik, der Türkei, Ungarn, dem Vereinigten Königreich und Zypern.



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

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

Europäisches Vorwort

Der Text des Dokuments 65E/654/CDV, zukünftige 1. Ausgabe der IEC 62714-4, 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 IEC 62714-4:2020 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): 2021-04-21
- spätestes Datum, zu dem nationale Normen, die diesem Dokument entgegenstehen, zurückgezogen werden müssen (dow): 2023-07-21

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

Anerkennungsnotiz

Der Text der Internationalen Norm IEC 62714-4:2020 wurde von CENELEC ohne irgendeine Abänderung als Europäische Norm angenommen.

In der offiziellen Fassung ist unter „Literaturhinweise“ zu der aufgelisteten Norm die nachstehende Anmerkung einzutragen:

IEC 62881:2018 ANMERKUNG Harmonisiert als EN IEC 62881:2018 (nicht modifiziert).

Anerkennungsnotiz zur Änderung AC

Der Text der Berichtigung IEC 62714-4:2020/COR1:2020 wurde von CENELEC ohne irgendeine Abänderung als EN IEC 62714-4:2020/AC:2020-12 angenommen.

AC

Anhang ZA (normativ)

Normative Verweisungen auf internationale Publikationen mit ihren entsprechenden europäischen Publikationen

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

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

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

<u>Publikation</u>	<u>Jahr</u>	<u>Titel</u>	<u>EN/HD</u>	<u>Jahr</u>
IEC 61131-3	–	Programmable controllers – Part 3: Programming languages	EN 61131-3	–
IEC 61131-10	–	Programmable controllers – Part 10: PLC open XML exchange format	EN IEC 61131-10	–
IEC 62714-1	2014	Engineering data exchange format for use in industrial automation systems engineering – Automation markup language – Part 1: Architecture and general requirements	EN 62714-1	–
W3C	2004	Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation 04 February 2004	–	–
W3C	2003	Mathematical Markup Language (MathML) Version 2.0 (Second Edition), W3C Recommendation 21 October 2003	–	–
IETF RFC 5646	–	Tags for Identifying Languages	–	–
IETF RFC 4122	–	A Universally Unique IDentifier (UUID) URN Namespace	–	–
PLCopen	–	XML 2.01: XML formats for IEC 61131-3	–	–

Inhalt

	Seite
Europäisches Vorwort.....	2
Anhang ZA (normativ) Normative Verweisungen auf internationale Publikationen mit ihren entsprechenden europäischen Publikationen	3
Einleitung	12
1 Anwendungsbereich	14
2 Normative Verweisungen	14
3 Begriffe und Abkürzungen	15
3.1 Begriffe	15
3.2 Abkürzungen	16
4 Konformität	17
5 Überblick.....	17
5.1 Logikinformationen in der Planung von Produktionssystemen	17
5.2 Logikmodelle in der Planung von Produktionssystemen.....	18
5.3 Speicherung von Logikmodellen in der AML logic XML.....	19
5.4 Referenzierung von Logikinformationen.....	21
6 Logikmodelle	22
6.1 Allgemeines	22
6.2 Gantt-Diagramme	22
6.2.1 Allgemeines	22
6.2.2 Grafische Elemente	22
6.2.3 Diagrammstruktur	22
6.2.4 Logikinformationen	23
6.2.5 Logikinformationen in Gantt-Diagrammen	23
6.3 Vorgangsknotennetze.....	23
6.3.1 Allgemeines	23
6.3.2 Grafische Elemente	24
6.3.3 Knotenstruktur	24
6.3.4 Netzstruktur	24
6.3.5 Logikinformationen	24
6.3.6 Logikinformationen in Vorgangsknotennetzen	25
6.4 Zeitverlaufsdigramme	25
6.4.1 Allgemeines	25
6.4.2 Grafische Elemente	25
6.4.3 Diagrammstruktur	26
6.4.4 Logikinformationen	26
6.4.5 Logikinformationen in Zeitverlaufsdigrammen	27
6.5 Ablaufsprache.....	27
6.6 Funktionsblockdiagramme.....	28

	Seite
6.7 Mathematischer Ausdruck.....	28
7 Beschreibung des AML logic XML Schemas	28
7.1 Überblick über das Schema	28
7.1.1 Verwendung des Schemas aus IEC 61131-10	28
7.1.2 Schemaversionierung	29
7.2 Root-Element „AMLLogic“	30
7.2.1 Allgemeines.....	30
7.2.2 Attribute	30
7.2.3 Untergeordnetes Element „WriterHeader“	30
7.2.4 Untergeordnetes Element „Types“	30
7.2.5 Untergeordnetes Element „Documentation“	31
7.3 Komplexer Typ „FunctionBlock“	31
7.3.1 Allgemeines.....	31
7.3.2 Attribute	31
7.3.3 Untergeordnetes Element „Parameters“	31
7.3.4 Untergeordnetes Element „Vars“	31
7.3.5 Untergeordnetes Element „MainBody“	31
7.3.6 Komplexer Typ „ParameterSet“	32
7.3.7 Komplexer Typ „VariableDecl“	32
7.4 Komplexer Typ „IML“	33
7.4.1 Allgemeines.....	33
7.4.2 Attribute	34
7.4.3 Untergeordnetes Element „Resource“	34
7.4.4 Untergeordnetes Element „TimeInformation“	34
7.4.5 Auswahl des „IML ***“-Elements.....	35
7.4.6 Komplexer Typ „IMLStep“	35
7.4.7 Komplexer Typ „IMLTransition“	36
7.4.8 Komplexer Typ „IMLSimultaneousDivergence“	36
7.4.9 Komplexer Typ „IMLSimultaneousConvergence“	36
7.5 Komplexer Typ „MathematicalExpression“	37
7.5.1 Allgemeines.....	37
7.5.2 Attribute	37
7.5.3 Untergeordnetes Element „VariableMapping“	37
7.5.4 Untergeordnetes Element „MathML“	37
7.6 Einfacher Typ „LogicModelTypeEnum“	37
7.7 Einfacher Typ „TimeUnion“	38
7.8 Einfacher Typ „TimeFormatEnum“	38
7.9 Einfacher Typ „UuidString“	38
8 Speicherung von Logikmodellen	38

	Seite
8.1 Allgemeines	38
8.2 Speicherung von Gantt-Diagrammen in der AML logic XML	38
8.2.1 Allgemeine Regeln	38
8.2.2 Speicherung des Startpunkts eines Gantt-Diagramms	39
8.2.3 Speicherung von Balken	39
8.2.4 Speicherung von Pfeilen	40
8.2.5 Speicherung von Balken für nachfolgende Aktivitäten	41
8.2.6 Speicherung von Balken für vorhergehende Aktivitäten	42
8.3 Speicherung von Vorgangsknotennetzen in AML logic XML	43
8.3.1 Allgemeine Regeln	43
8.3.2 Speicherung des Startpunkts eines Vorgangsknotennetzes	44
8.3.3 Speicherung von Knoten	44
8.3.4 Speicherung von Pfeilen	45
8.3.5 Speicherung von Knoten für nachfolgende Aktivitäten	45
8.3.6 Speicherung von Knoten für vorhergehende Aktivitäten	46
8.4 Speicherung von Zeitverlaufsdigrammen in AML logic XML	48
8.4.1 Allgemeine Regeln	48
8.4.2 Speicherung der Zeitachse eines Zeitverlaufsdigramms	48
8.4.3 Speicherung von Ressourcen und Ressourcenzuständen	49
8.4.4 Speicherung von Lebenslinien	49
8.4.5 Speicherung von Zeit- und Ressourcensignalen	51
8.5 Speicherung von der Ablaufsprache in AML logic XML	53
8.5.1 Allgemeine Regeln	53
8.5.2 Speicherung von Variablen	53
8.6 Speicherung von Funktionsblockdiagrammen in AML logic XML	54
8.6.1 Allgemeine Regeln	54
8.6.2 Speicherung von Variablen	54
8.7 Speicherung von mathematischen Ausdrücken in AML logic XML	55
8.7.1 Allgemeine Regeln	55
8.7.2 Speicherung von Variablen	56
8.7.3 Speicherung von Variablenzuordnungen	57
8.7.4 Speicherung von mathematischen Ausdrücken	57
9 Metainformationen über die AML logic XML Exportwerkzeuge	57
10 Erweiterung der AML-Klassen für Logik	59
10.1 Allgemeines	59
10.2 AutomationMLLogicRoleClassLib	59
10.2.1 AllgemeinesS	59
10.2.2 Rollenklasse InterlockingTargetGroup	59
10.2.3 Rollenklasse InterlockingSourceGroup	60

	Seite
10.2.4 Rollenklasse LogicModelObject	61
10.3 AutomationMLLogicInterfaceClassLib	62
10.3.1 Allgemeines	62
10.3.2 Schnittstellenklasse LogicModelInterface	63
10.3.3 Schnittstellenklasse SequencingLogicModelInterface	63
10.3.4 Schnittstellenklasse BehaviourLogicModelInterface	63
10.3.5 Schnittstellenklasse InterlockingLogicModelInterface	64
10.3.6 Schnittstellenklasse LogicModelElementInterface	64
10.3.7 Schnittstellenklasse VariableInterface	65
10.3.8 Schnittstellenklasse InterlockingVariableInterface	65
10.4 AutomationMLPLCopenXMLInterfaceClassLib	66
10.4.1 Allgemeines	66
10.4.2 Schnittstellenklasse VariableInterface	66
10.5 AutomationMLInterfaceClassLib	67
10.5.1 Allgemeines	67
10.5.2 Schnittstellenklasse InterlockingConnector	67
10.5.3 Schnittstellenklasse PLCopenXMLInterface	67
11 Referenzierung von AML logic XML Dokumenten	68
11.1 Allgemeines	68
11.2 Referenzierung von Logikinformationen	68
12 Verknüpfung von AML-Objekten mit Verriegelungsinformationen	69
12.1 Allgemeines	69
12.2 Referenzierung von Verriegelungsinformationen	69
Anhang A (informativ) Beispiele für die Speicherung von Logikmodellen in der AML logic XML	71
A.1 Beispiel für die Speicherung von Gantt-Diagrammen	71
A.1.1 Allgemeines	71
A.1.2 Speicherung von Aktivitäten ohne Vorgänger- und Nachfolger-Beziehung	71
A.1.3 Speicherung eines Ablaufs von Aktivitäten	72
A.1.4 Speicherung eines Ablaufs von Aktivitäten mit Divergenzen	73
A.1.5 Speicherung eines Ablaufs von Aktivitäten mit Konvergenz	74
A.2 Beispiel für die Speicherung von Vorgangsknotennetzen	76
A.2.1 Allgemeines	76
A.2.2 Speicherung von Aktivitäten ohne Vorgänger- und Nachfolger-Beziehung	76
A.2.3 Speicherung eines Ablaufs von Aktivitäten	77
A.2.4 Speicherung eines Ablaufs von Aktivitäten mit Divergenz	78
A.2.5 Speicherung eines Ablaufs von Aktivitäten mit Konvergenz	80
A.3 Beispiele für die Speicherung von Zeitverlaufdiagrammen	81
A.3.1 Allgemeines	81
A.3.2 Beispiel für die Speicherung eines internen Signals	81

	Seite
A.3.3 Beispiel für die Speicherung eines externen Signals	83
A.3.4 Beispiel für die Speicherung eines Signals zwischen zwei Zustandsflüssen einer Ressource	84
A.4 Beispiel für die Speicherung der Ablaufsprache	86
A.5 Beispiel für die Speicherung von Funktionsblockdiagrammen.....	88
A.6 Beispiel für die Speicherung von mathematischen Ausdrücken	91
Anhang B (informativ) Beispiele für Referenzierung von Logikinformationen	95
B.1 Allgemeines	95
B.2 Referenzierung von als Logikmodelle ausgedrückten Logikinformationen	95
B.2.1 Allgemeines	95
B.2.2 Referenzierung von Logikinformationen in einem Funktionsblock.....	95
B.2.3 Referenzierung von aus mehreren Funktionsblöcken bestehenden Logikinformationen	96
B.2.4 Referenzierung von aus mehreren AML logic XML Dokumenten bestehenden Logikinformationen	97
B.3 Referenzierung von Logikinformationen als Teil von Logikmodellen	99
B.3.1 Allgemeines	99
B.3.2 Referenzierung einer Variablen.....	99
B.3.3 Referenzierung eines Logikelements	100
B.4 Referenzierung von als Teile von Logikmodellen bereits referenzierten Logikinformationen.....	101
Anhang C (informativ) Beispiele für Referenzierung von Verriegelungsinformationen	103
C.1 Allgemeines	103
C.2 Verriegelungsinformationen	104
C.3 Referenzierung von Verriegelungsinformationen ohne Verriegelungsbedingung.....	104
C.4 Referenzierung von Verriegelungsinformationen mit Verriegelungsbedingung.....	106
Anhang D (normativ) XML Darstellung von AML-Standardbibliotheken	110
D.1 Allgemeines	110
D.2 AutomationMLLogicRoleClassLib	110
D.3 AutomationMLLogicInterfaceClassLib	111
D.4 AutomationMLPLCOpenXMLInterfaceClassLib	112
Anhang E (normativ) XML Darstellung des AML logic XML Schemas	113
Literaturhinweise	117
 Bilder	
Bild 1 – Überblick über das Planungsdatenaustauschformat AML	12
Bild 2 – Beispiel für eine Systemdarstellung mit verschiedenen Informationsarten in AML.....	18
Bild 3 – Logikmodelle in AML	19
Bild 4 – Speicherung von Logikmodellen in der AML logic XML	20
Bild 5 – Modellierungselemente in der AML logic XML	21
Bild 6 – Modellelemente von Gantt-Diagrammen.....	22
Bild 7 – Daten in Gantt-Diagrammen.....	23
Bild 8 – Modellelemente von Vorgangsknotennetzen	24

	Seite
Bild 9 – Daten in Vorgangsknotennetzen	25
Bild 10 – Modellelemente von Zeitverlaufsdiagrammen	26
Bild 11 – Informationen in Zeitverlaufsdiagrammen	27
Bild 12 – Überblick über das AML-Schema	29
Bild 13 – Root-Element „AMLLogic“	30
Bild 14 – Komplexer Typ „FunctionBlock“	31
Bild 15 – Komplexer Typ „ParameterSet“	32
Bild 16 – Komplexer Typ „VariableDecl“	33
Bild 17 – Komplexer Typ „IML“	34
Bild 18 – Komplexer Typ „IMLStep“	35
Bild 19 – Komplexer Typ „IMLTransition“	36
Bild 20 – Komplexer Typ „IMLSimultaneousDivergence“	36
Bild 21 – Komplexer Typ „IMLSimultaneousConvergence“	36
Bild 22 – Komplexer Typ „MathematicalExpression“	37
Bild 23 – AutomationMLLogicRoleClassLib	59
Bild 24 – AutomationMLLogicInterfaceClassLib	62
Bild 25 – AutomationMLPLCopenXMLInterfaceClassLib	66
Bild A.1 – Durchflussmengen von Ventilen	91
Bild A.2 – Beispiel für die Speicherung eines mathematischen Ausdrucks	94
Bild B.1 – Referenzierung von (als SFC ausgedrückten) in einem Funktionsblock gespeicherten Logikinformationen	96
Bild B.2 – XML Text der CAEX-Datei zur Referenzierung von in einem Funktionsblock gespeicherten Logikinformationen	96
Bild B.3 – Referenzierung von aus mehreren Funktionsblöcken bestehenden Logikinformationen	97
Bild B.4 – Referenzierung von aus mehreren AML logic XML Dokumenten bestehenden Logikinformationen	98
Bild B.5 – XML Text der CAEX-Datei zur Referenzierung von aus mehreren AML logic XML Dokumenten bestehenden Logikinformationen	98
Bild B.6 – Referenzierung einer Variablen	99
Bild B.7 – XML Text der CAEX-Datei zur Referenzierung einer Variablen	99
Bild B.8 – Referenzierung eines Logikelements	100
Bild B.9 – XML Text der CAEX-Datei zur Referenzierung eines Logikelements	100
Bild B.10 – Referenzierung einer Variablen eines bereits referenzierten Logikmodells	101
Bild B.11 – XML Text der CAEX-Datei zur Referenzierung einer Variablen eines bereits referenzierten Logikmodells	102
Bild C.1 – Beispiel für ein Fertigungssystem	103
Bild C.2 – Beispiel für eine Verriegelungsquellengruppe und eine Verriegelungszielgruppe	104
Bild C.3 – Referenzierung von Verriegelungsinformationen ohne Verriegelungsbedingung	105
Bild C.4 – XML Text der CAEX-Datei zur Referenzierung von Verriegelungsinformationen ohne Verriegelungsbedingung	106
Bild C.5 – Referenzierung von Verriegelungsinformationen mit Verriegelungsbedingung	107

	Seite
Bild C.6 – XML Text der CAEX-Datei zur Referenzierung von Verriegelungsinformationen mit Verriegelungsbedingung.....	108
Bild C.7 – Verknüpfung einer logischen Schnittstelle mit einer physikalischen Schnittstelle (Erweiterung von Bild C.5).....	109
Bild C.8 – XML Text der CAEX-Datei zur Verknüpfung einer logischen Schnittstelle mit einer physikalischen Schnittstelle (Erweiterung von Bild C.6)	109
Tabellen	
Tabelle 1 – Abkürzungen.....	16
Tabelle 2 – Zuordnung des Startpunkts eines Gantt-Diagramms zu IML-Elementen.....	39
Tabelle 3 – Zuordnung der Balken eines Gantt-Diagramms zu IML-Elementen.....	40
Tabelle 4 – Zuordnung der Pfeile eines Gantt-Diagramms zu IML-Elementen.....	40
Tabelle 5 – Zuordnung des Balkens eines Gantt-Diagramms mit mindestens einem nachfolgenden Balken zu IML-Elementen	41
Tabelle 6 – Zuordnung des Balkens eines Gantt-Diagramms mit mindestens einem vorhergehenden Balken zu IML-Elementen	42
Tabelle 7 – Zuordnung des Startpunkts eines Vorgangsknotennetzes zu IML-Elementen	44
Tabelle 8 – Zuordnung der Knoten eines Vorgangsknotennetzes zu IML-Elementen.....	44
Tabelle 9 – Zuordnung der Pfeile eines Vorgangsknotennetzes zu IML-Elementen	45
Tabelle 10 – Zuordnung der Knoten für nachfolgende Aktivitäten eines Vorgangsknotennetzes zu IML-Elementen	45
Tabelle 11 – Zuordnung der Knoten für vorhergehende Aktivitäten eines Vorgangsknotennetzes zu IML-Elementen	47
Tabelle 12 – Zuordnung der Zeitachse eines Zeitverlaufsdigramms zu IML-Elementen	48
Tabelle 13 – Zuordnung von Ressourcen und Ressourcenzuständen eines Zeitverlaufsdigramms zu IML-Elementen	49
Tabelle 14 – Zuordnung der Lebenslinien eines Zeitverlaufsdigramms zu IML-Elementen	49
Tabelle 15 – Zuordnung von Zeit- und Ressourcensignalen zu IML-Elementen	52
Tabelle 16 – Speicherung von der Ablaufsprache in AML logic XML	53
Tabelle 17 – Speicherung der Variablen eines Funktionsblockdiagramms in der AML logic XML	55
Tabelle 18 – Speicherung der Variablen eines mathematischen Ausdrucks in der AML logic XML.....	56
Tabelle 19 – Speicherung der Variablenzuordnungen eines mathematischen Ausdrucks in der AML logic XML	57
Tabelle 20 – Speicherung eines mathematischen Ausdrucks in der AML logic XML	57
Tabelle 21 – Metainformationen zu jedem AML logic XML Exportwerkzeug	58
Tabelle 22 – Rollenklasse InterlockingTargetGroup.....	60
Tabelle 23 – Rollenklasse InterlockingSourceGroup.....	61
Tabelle 24 – Rollenklasse LogicModelObject.....	62
Tabelle 25 – Schnittstellenklasse LogicModelInterface.....	63
Tabelle 26 – Schnittstellenklasse SequencingLogicModelInterface.....	63
Tabelle 27 – Schnittstellenklasse BehaviourLogicModelInterface	64
Tabelle 28 – Schnittstellenklasse InterlockingLogicModelInterface	64
Tabelle 29 – Schnittstellenklasse LogicModelElementInterface	65

	Seite
Tabelle 30 – Schnittstellenklasse VariableInterface.....	65
Tabelle 31 – Schnittstellenklasse InterlockingVariableInterface	66
Tabelle 32 – Schnittstellenklasse VariableInterface.....	67
Tabelle 33 – Schnittstellenklasse InterlockingConnector.....	67
Tabelle 34 – Schnittstellenklasse PLCopenXMLInterface	68
Tabelle A.1 – Speicherung des Beispiels für ein Gantt-Diagramm „Aktivitäten ohne Vorgänger- und Nachfolger-Beziehung“	71
Tabelle A.2 – Speicherung des Beispiels für ein Gantt-Diagramm „Ablauf von Aktivitäten“.....	72
Tabelle A.3 – Speicherung des Beispiels für ein Gantt-Diagramm „Sequenzierung von Aktivitäten mit Divergenz“	73
Tabelle A.4 – Speicherung des Beispiels für ein Gantt-Diagramm „Sequenzierung von Aktivitäten mit Konvergenz“	74
Tabelle A.5 – Speicherung des Beispiels für ein Vorgangsknotennetz „Aktivitäten ohne Vorgänger- und Nachfolger-Beziehung“	76
Tabelle A.6 – Speicherung des Beispiels für ein Vorgangsknotennetz „Ablauf von Aktivitäten“	77
Tabelle A.7 – Speicherung des Beispiels für ein Vorgangsknotennetz „Ablauf von Aktivitäten mit Divergenz“	78
Tabelle A.8 – Speicherung des Beispiels für ein Vorgangsknotennetz „Ablauf von Aktivitäten mit Konvergenz“	80
Tabelle A.9 – Beispiel für die Speicherung eines Zeitverlaufsdigramms „Umwandlung eines Zustandswechsels in einen nachfolgenden Zustand“	81
Tabelle A.10 – Beispiel für die Zuordnung eines Zeitverlaufsdigramms „Zwei mit einer Verzögerung von drei Sekunden ausgelöste externe Signale“.....	83
Tabelle A.11 – Beispiel für die Speicherung eines Zeitverlaufsdigramms „Vom Zustand einer Ressource ausgelöstes Signal, das von einem anderen Zustand verwendet wird“	84
Tabelle A.12 – Beispiel für die Speicherung von der Ablaufsprache	86
Tabelle A.13 – Beispiel für die Speicherung von Funktionsblockdiagrammen	89

Einleitung

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

Das Ziel von AML besteht darin, Planungswerkzeuge der verschiedenen Disziplinen der vorhandenen heterogenen Werkzeuglandschaft, 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 und ermöglicht die Modellierung physischer 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 ihrerseits Abläufe, Verhalten und Steuerung beinhaltet.

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 Anpassungen für AML 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 gibt einen Überblick über die Grundarchitektur von AML sowie über die Verteilung der Topologie, Geometrie, Kinematik und Logikinformationen.

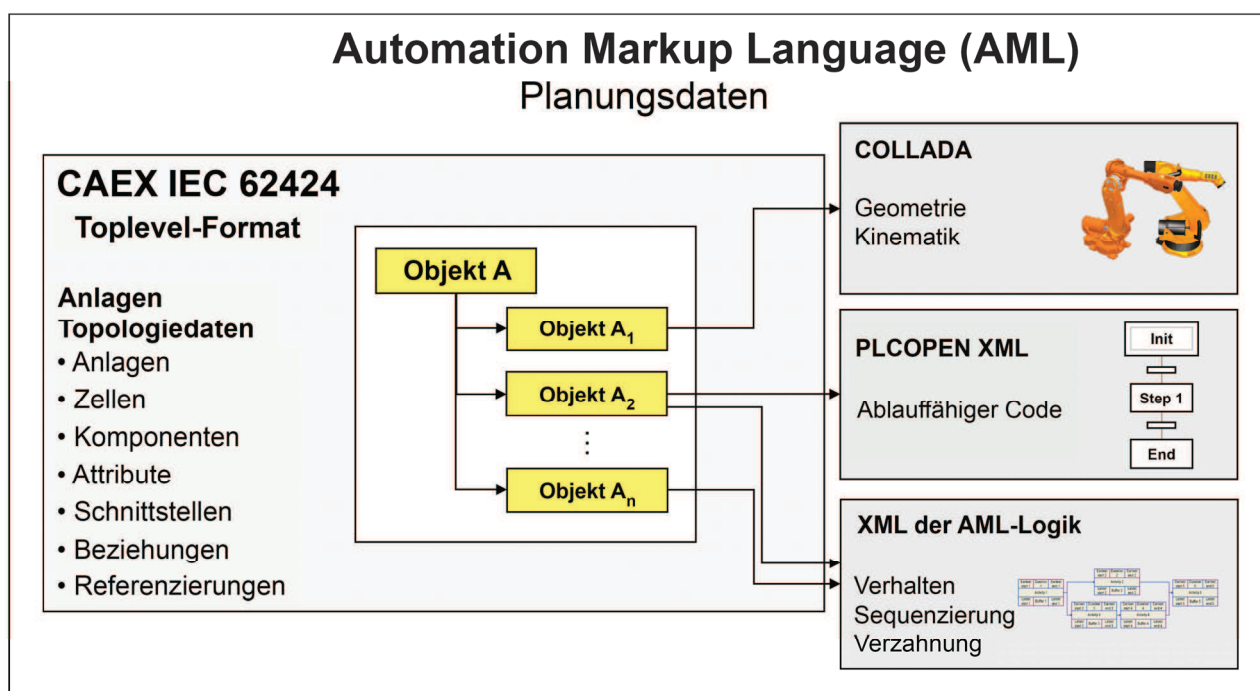


Bild 1 – Überblick über das Planungsdatenaustauschformat AML

Infolge 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 beschreibt die allgemeine Architektur von AML, die Modellierung von Planungsdaten, Klassen, Instanzen, Beziehungen, Referenzen, Hierarchien, grundlegende AML-Bibliotheken sowie erweiterte AML-Konzepte.
- IEC 62714-2: Rollenklassenbibliotheken
Dieser Teil legt zusätzliche AML-Bibliotheken fest.
- IEC 62714-3: Geometrie und Kinematik
Dieser Teil legt die Modellierung von Geometrie- und Kinematikinformationen fest.
- IEC 62714-4: Logik
Dieser Teil legt die Modellierung und Referenzierung von Logikinformationen fest.

Um die Verknüpfung weiterer Datenstandards mit AML zu ermöglichen, werden möglicherweise zukünftig weitere Teile ergänzt.

Abschnitt 5 enthält einen informativen Überblick über diesen Teil der Norm.

Abschnitt 6 enthält eine normative Festlegung der betrachteten Logikmodelle.

Abschnitt 7 enthält eine normative Festlegung zum AML logic XML Schema, mit dem Logikmodelle gespeichert werden können.

Abschnitt 8 enthält die normativen Festlegungen zur Speicherung der Logikmodelle in AML logic XML Dokumenten.

Abschnitt 9 legt fest, wie Metainformationen zum Quellwerkzeug direkt im AML logic XML Dokument gespeichert werden.

Abschnitt 10 legt die logikbezogenen Erweiterungen der Rollen- und Schnittstellen-Klassenbibliothek fest.

AC | Abschnitt 11 enthält eine normative Festlegung zur Referenzierung von Logikinformationen in AML logic XML Dokumenten.

Abschnitt 12 enthält eine normative Festlegung zur Referenzierung von Verriegelungsinformationen in AML logic XML Dokumenten.

Anhang A enthält Beispiele für die Speicherung von Logikmodellen in AML logic XML Dokumenten.

Anhang B beschreibt die Referenzierungsmethoden für Logikinformationen.

Anhang C beschreibt die Referenzierungsmethoden für Verriegelungsinformationen.

Anhang D enthält eine normative XML Darstellung der in diesem Dokument festgelegten Bibliotheken.

Anhang E enthält eine normative XML Darstellung des in diesem Dokument definierten AML logic XML Schemas.

1 Anwendungsbereich

Dieser Teil der IEC 62714 legt die Integration von Logikinformationen als Bestandteil eines AML-Modells für den Datenaustausch in einer verschiedenartigen Werkzeuglandschaft von Produktionssystemen fest.

Dieses Dokument legt drei Arten von Logikinformationen fest: Ablauf-, Verhaltens- und Verriegelungsinformationen.

Dieses Dokument befasst sich mit den folgenden sechs Ablauf- und Verhaltenslogikmodellen (deckt die verschiedenen Phasen des Planungsprozesses von Produktionssystemen ab) und wie sie in AML integriert sind: Gantt-Diagramm, Vorgangsknotennetz, Zeitverlaufdiagramm, Ablaufsprache (SFC), Funktionsblockdiagramm (FBD) und mathematischer Ausdruck.

Dieses Dokument legt die Modellierung von Gantt-Diagramm, Vorgangsknotennetz und Zeitverlaufdiagramm fest und wie sie in Intermediate Modelling Layer (IML) gespeichert werden.

ANMERKUNG 1 So ist die Transformation eines Logikmodells in ein anderes ermöglicht. Die vorwärts gerichtete Transformation unterstützt den Informationsanreicherungsprozess und verringert oder vermeidet die erneute Eingabe von Daten zwischen den austauschenden Planungswerkzeugen.

ANMERKUNG 2 IML ermöglicht auch die Abbildung anderer Logikmodelle, beispielsweise ereignisgesteuerte Logikmodelle, wie Zustandsdiagramme.

Dieses Dokument legt fest, wie Verriegelungsinformationen (wie Verriegelungsquelle und Zielgruppen) in AML modelliert sind. Das Verriegelungslogikmodell ist im Funktionsblockdiagramm (FBD) gespeichert.

Dieses Dokument legt das AML logic XML Schema fest, das die Logikmodelle durch Verwendung der IEC 61131-10 speichert.

Dieses Dokument legt fest, wie in PLCopen XML Dokumenten gespeicherte Steuerungsprogramme zu referenzieren sind.

Dieses Dokument legt keine Einzelheiten zum Datenaustauschverfahren oder Umsetzungsanforderungen für Import-/Export-Werkzeuge fest.

2 Normative Verweisungen

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

IEC 61131-3, *Programmable controllers – Part 3: Programming languages*

IEC 61131-10, *Programmable controllers – Part 10: PLC open XML exchange format*

IEC 62714-1:2014¹, *Engineering data exchange format for use in industrial automation systems engineering – Automation markup language – Part 1: Architecture and general requirements*

W3C. *Extensible Markup Language (XML) 1.0 (Third Edition)*, W3C Recommendation 04 February 2004 [online]. Edited by T. Bray et al., February 2004 [viewed on 2020-05-14]. Verfügbar unter <http://www.w3.org/TR/2004/REC-xml-20040204>

¹ Erste Auflage. Diese erste Auflage wurde 2018 durch eine zweite Auflage IEC 62714-1:2018, *Datenaustauschformat für Planungsdaten industrieller Automatisierungssysteme – Automation markup language – Teil 1: Architektur und allgemeine Anforderungen* ersetzt.

W3C. *Mathematical Markup Language (MathML) Version 2.0 (Second Edition)*, W3C Recommendation 21 October 2003 [online]. Edited by D. Carlisle et al., October 2003 [viewed 2020-05-14].
Verfügbar unter <<https://www.w3.org/TR/MathML2/>>

INTERNET ENGINEERING TASK FORCE (IETF). RFC 5646: *Tags for Identifying Languages* [online]. Edited by A. Phillips and M. Davis, September 2009 [viewed 2020-05-14].
Verfügbar unter <https://tools.ietf.org/html/rfc5646>

INTERNET ENGINEERING TASK FORCE (IETF). RFC 4122, *A Universally Unique IDentifier (UUID) URN Namespace* [online]. Edited by P. Leach et al., July 2005 [viewed 2020-05-14].
Verfügbar unter <https://tools.ietf.org/html/rfc4122>

PLCopen. *XML 2.01: XML formats for IEC 61131-3* [online]. Edited by K. Ketterle et al., May 2009 [viewed 2020-05-14].
Verfügbar unter https://www.plcopen.org/system/files/downloads/tc6_xml_v201_technical_doc.pdf

3 Begriffe und Abkürzungen

3.1 Begriffe

Für die Anwendung dieses Dokuments gelten die Begriffe nach IEC 62714-1 und die folgenden Begriffe.

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

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

3.1.1

AML-Logikobjekt

AML-Objekt, das Logikinformationen enthält

3.1.2

AML logic XML Dokument

externes XML Dokument auf der Grundlage des AML logic XML Schemas, das von einem AML-Logikobjekt referenziert werden kann

3.1.3

AML logic XML Schema

XML Schema zur Speicherung der Logikmodelle von AML

3.1.4

Verhaltensinformationen

Informationen, mit denen die Reaktion eines geregelten Systems auf Ablaufinformationen und andere externe Einflussfaktoren beschrieben werden

3.1.5

Verhaltenslogikmodell

Darstellung von Verhaltensinformationen

3.1.6

Verriegelungsinformationen

Informationen zur Beschreibung von Anweisungen für ein geregeltes Objekt, mit denen verhindert werden soll, dass sich aus der Nutzung des Objekts gefährdende Auswirkungen für Menschen und die Umwelt ergeben

3.1.7

Verriegelungslogikmodell

Darstellung von Verriegelungsinformationen

3.1.8

Verriegelungsquellengruppe

Gruppe von AML-Objekten, die anzeigen, wenn Aktionen erforderlich sind, um die Funktionssicherheit herzustellen

3.1.9

Verriegelungszielgruppe

Gruppe von AML-Objekten, die Aktionen ausführen, um die funktionale Sicherheit herzustellen

3.1.10

Logikmodellelement

Element eines Logikmodells

3.1.11

Logikinformationen

Ablaufinformationen, Verhaltensinformationen oder Verriegelungsinformationen

3.1.12

Logikmodell

Darstellung von Logikinformationen

3.1.13

mathematischer Ausdruck

sinnvolle Kombination aus Zahlen, Variablen und mathematischen Symbolen zur Darstellung syntaktisch korrekter gegenseitiger Abhängigkeiten

Anmerkung 1 zum Begriff: Ein mathematischer Ausdruck kann ebenso ein einfacher Ausdruck wie ein komplexes Gleichungssystem sein.

3.1.14

Ablaufinformationen

Informationen, mit denen die Anweisungen für ein geregeltes System beschrieben werden

3.1.15

Ablauflogikmodell

Darstellung von Ablaufinformationen

3.2 Abkürzungen

Für die Anwendung dieses Dokuments gelten die Abkürzungen der IEC 62714-1 und die in Tabelle 1 aufgelisteten Abkürzungen.

Tabelle 1 – Abkürzungen

SFC	Ablaufsprache (en: Sequential Function Chart)
FBD	Funktionsblockdiagramm (en: Function Block Diagram)
IML	Intermediate Modelling Layer
XSLT	Extensible Stylesheet Language Transformations
MathML	Mathematical Markup Language

4 Konformität

AC

Um Konformität mit diesem Teil von IEC 62714 in Bezug auf die Unterstützung von AML zu beanspruchen, müssen die Anforderungen der Abschnitte 6, 7, 8, 9, 10, 11 und 12 erfüllt sein. Im Anwendungsbereich von AML muss ein AML logic XML Dokument dem in Abschnitt 7 und Anhang E festgelegten AML logic XML Schema entsprechen.

Ein PLCopen-XML-Dokument muss der Spezifikation von PLCopen[®] XML 2.0 oder 2.01 oder der Spezifikation von IEC 61131-10 entsprechen.

Dieses Dokument legt keine Festlegungen zum Datenaustauschverfahren oder Umsetzungsanforderungen für Import-/Export-Werkzeuge fest.

5 Überblick

5.1 Logikinformationen in der Planung von Produktionssystemen

- Dieses Dokument befasst sich mit der Speicherung von Logikinformationen, einem wesentlichen Punkt des elektrischen Designs, der HMI-Entwicklung, der Steuerungs- und Robotersteuerungsprogrammierung, der Simulation und der virtuellen Inbetriebnahme.
- Zur Unterstützung verschiedener Phasen im iterativen Planungsprozess von Produktionssystemen und der unterschiedlichen Detailstufen muss AML in der Lage sein, die Logikinformationen verschiedener Werkzeuge und Bereiche abzudecken. So müssen verschiedene Arten Logikinformationen gespeichert werden, die zu einem ganzen Produktionssystem oder zu betreffenden Einzelkomponenten gehören. Diese Vielfalt an Daten kann in zwei Gruppen unterteilt werden: Ablaufinformationen und Verhaltensinformationen. Beide Informationsarten dienen unterschiedlichen Zwecken, die sich allerdings je nach Sichtweise und Nutzung der Elemente in AML überschneiden können (siehe Bild 2).
 - Ablaufinformationen sind Informationen, mit denen die Anweisungen für ein geregeltes System (oder Subsystem) beschrieben werden. Auf der höchsten Detailstufe werden sie häufig durch ein Programm abgebildet, das in einer oder mehreren Regeleinheiten ausgeführt wird, beispielsweise in einer Roboter- oder Programmlogiksteuerung. Aber die Entwicklung von Abläufen beginnt – bei Modellen mit einer hohen Abstraktionsstufe – früher im Planungsprozess, und zwar bei Modellen, welche die erforderlichen Vorgänge einer größeren Maßstabseinheit darstellen (Zelle, Linie, Anlage usw.).
 - Verhaltensinformationen sind Informationen, mit denen die Reaktion eines geregelten Systems (oder Subsystems) auf Ablaufinformationen und andere externe Wechselwirkungen beschrieben werden. Sie werden häufig anhand eines bestimmten Modells dargestellt, das auf externe Eingänge reagiert, beispielsweise das Verhalten eines Greifers oder Ventils. Diese externen Eingänge würden das Verhalten eines Greifers oder die Anzeige seiner Zustände auslösen.
 - Neben den Ablauf- und Verhaltensinformationen gibt es mit den Verriegelungsinformationen einen dritten ergänzenden Typ von Logikinformationen (siehe Bild 2). Sie sind ein bedeutender Teil der Planung von Produktionssystemen, mit dem das sichere Verhalten umgesetzt werden soll. Sie beeinflussen nicht nur die sichere Interaktion mit Menschen und der Umgebung, sondern tragen auch zur Vermeidung instabiler Systemzustände bei, die unter Umständen zu schädigenden Einflüssen für Menschen und der Umgebung oder zu Schäden an Maschinen und Produkten führen.

Eine Regelstrecke (oder ein Subsystem) kann durch mehr als nur eine Art der oben genannten Informationen in AML dargestellt werden, beispielsweise ein Roboter (siehe Bild 2).

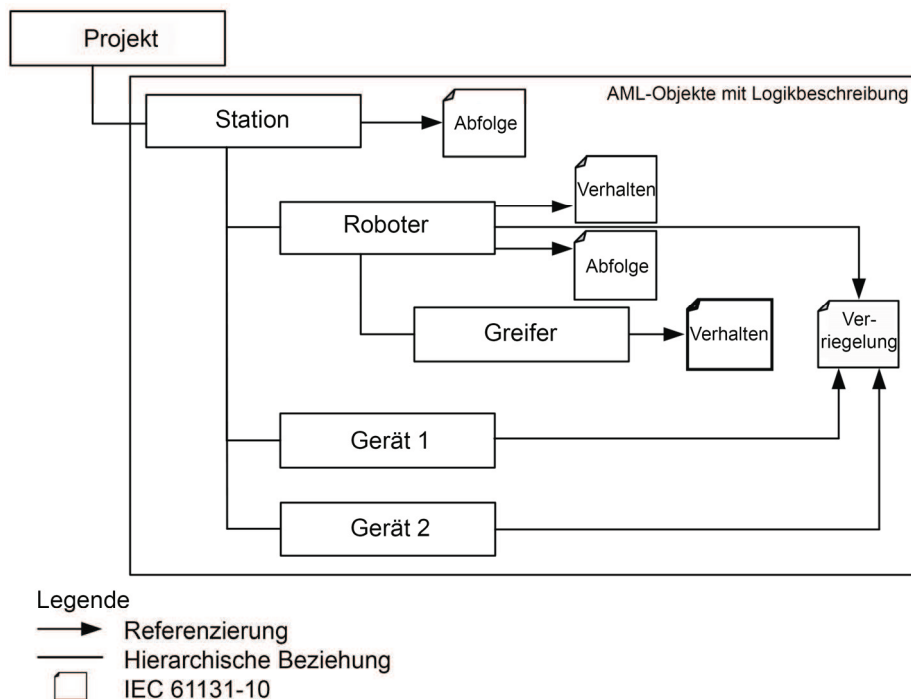


Bild 2 – Beispiel für eine Systemdarstellung mit verschiedenen Informationsarten in AML

5.2 Logikmodelle in der Planung von Produktionssystemen

Damit AML im Rahmen des Planungsprozesses von Produktionssystemen anwendbar ist, muss diese Auszeichnungssprache übliche Logikmodelle unterstützen und deren Logikinformationen speichern, welche die einzelnen Phasen des Planungsprozesses abdecken (siehe Bild 3). Bei diesen Logikmodellen handelt es sich um die folgenden:

- In den frühen Stadien des Planungsprozesses werden hauptsächlich Gantt-Diagramme verwendet, um die Zeitplanung und Dauer der Herstellungsprozesse auf einer hohen Abstraktionsebene darzustellen (siehe 6.2).
- Vorgangsknotennetze dienen in den frühen Stadien des Planungsprozesses in erster Linie zur Darstellung der Zeitplanung und der wechselseitigen Beziehungen von Herstellungsprozessen. Dabei können auch Zeitplanungsinformationen gespeichert werden (siehe 6.3).
- Zeitverlaufdiagramme werden hauptsächlich in späteren Phasen des Planungsprozesses verwendet, um die Geräte mit ihren Zuständen und Zeitbeziehungen zu beschreiben (siehe 6.4). Dabei werden echte Signale eingeführt und die Möglichkeit, Ablaufinformationen auszudrücken.
- Die Ablaufsprache (SFC) zählt zu den Programmiersprachen aus IEC 61131-3 und wird im Wesentlichen im späteren Verlauf der Planung von Produktionssystemen verwendet (siehe 6.5). Sowohl Ablauf- als auch Verhaltensinformationen können modelliert werden.
- Funktionsblockdiagramme (FBD) zählen ebenfalls zu den Programmiersprachen aus IEC 61131-3 und werden im Wesentlichen gleichfalls im späteren Verlauf der Planung von Produktionssystemen verwendet (siehe 6.6). Sowohl Verriegelungs- als auch Verhaltensinformationen können modelliert werden.
- Mathematische Ausdrücke dienen hauptsächlich im späteren Verlauf der Planung zur Beschreibung von mathematischen, physikalischen, chemischen oder logischen Beziehungen eines Geräts (siehe 6.7).

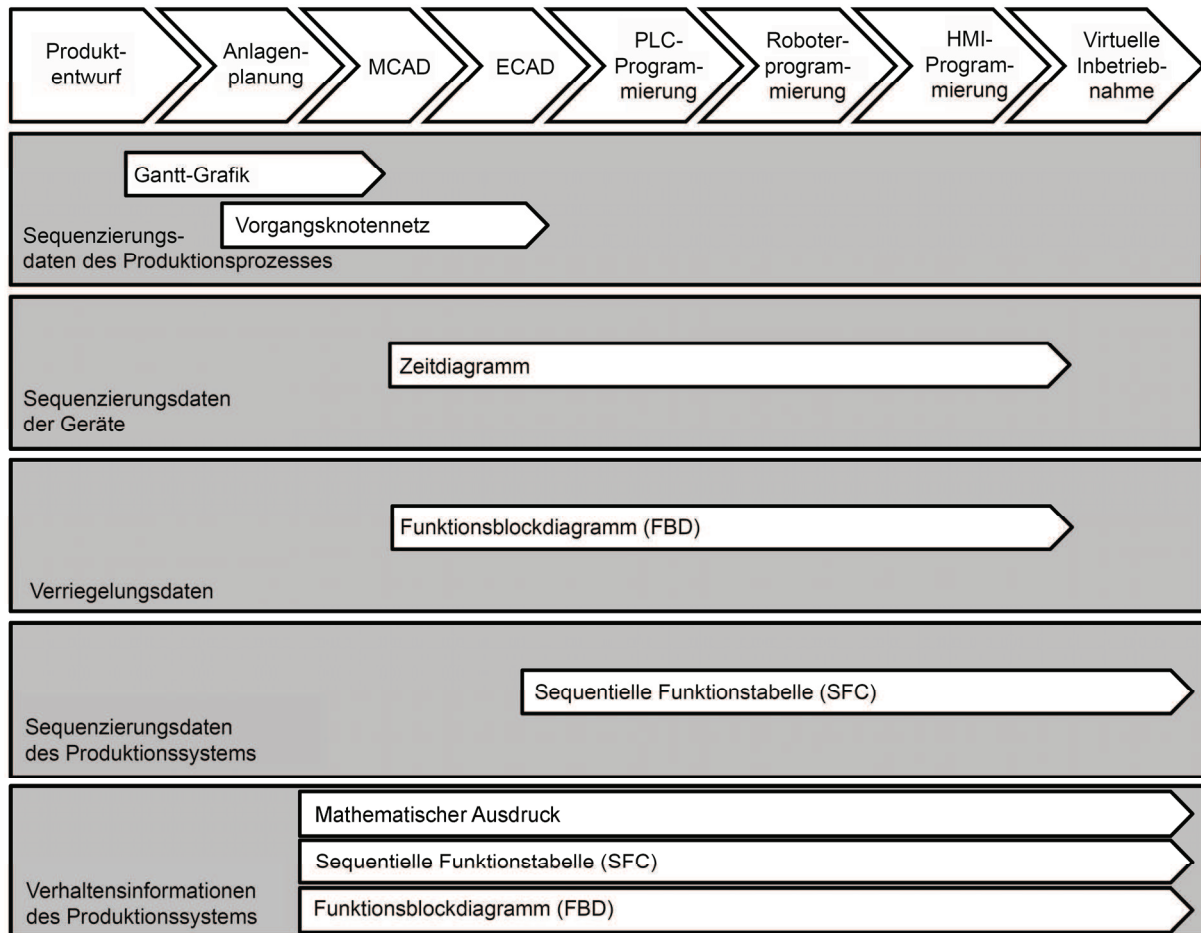


Bild 3 – Logikmodelle in AML

5.3 Speicherung von Logikmodellen in der AML logic XML

Es ist beabsichtigt, die Umwandlung von einem Logikmodell in ein anderes zu ermöglichen. Dies ist von entscheidender Bedeutung, da es sich bei der Planung von Produktionssystemen um einen Datenanreicherungsprozess handelt. Dabei werden die Ablaufinformationen für ein Produktionssystem zunächst nur grob mithilfe eines Gantt-Diagramms beschrieben und müssen im nächsten Schritt detaillierter dargestellt werden. Damit wird aus dem Gantt-Diagramm ein Vorgangsknotennetz: Die Informationen aus dem Gantt-Diagramm werden übernommen, und die komplexen Zeitablaufbedingungen können hinzugefügt werden. Theoretisch kann dieser Workflow bis hin zu einem ablauffähigen Programm fortgesetzt werden, wobei sich die Informationsdichte bei jedem Schritt erhöht. Dies führt dazu, dass eine erneute Eingabe von Daten innerhalb des anschließend verwendeten Planungswerkzeugs nicht erforderlich ist. Da die Logikmodelle aber unterschiedliche Modellierungsmöglichkeiten aufweisen, ergeben sich – bei vorwärts und rückwärtsgerichteter Transformation – in bestimmten Fällen Informationsverluste, beispielsweise bei der Transformation eines Vorgangsknotennetzes in ein Gantt-Diagramm.

Ein Überblick über den Anwendungsbereich dieses Dokuments ist in Bild 4 abgebildet.

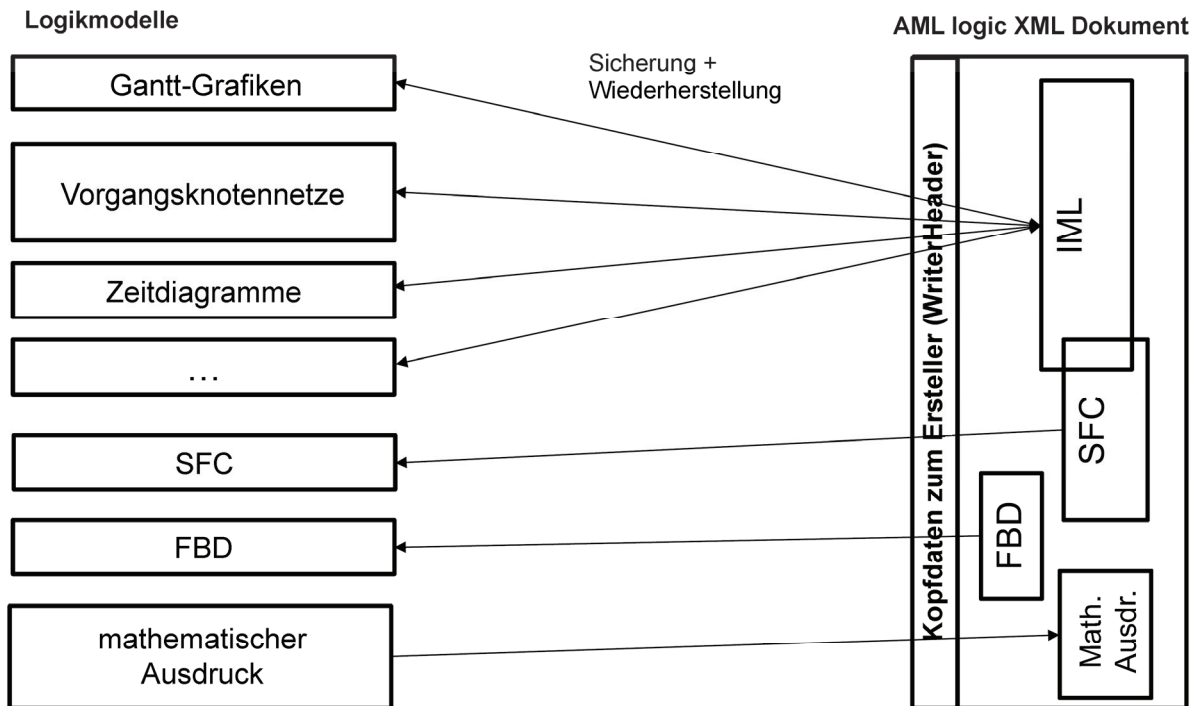


Bild 4 – Speicherung von Logikmodellen in der AML logic XML

ANMERKUNG Zur Umwandlung einer Ablaufsprache im IML-Kontext in eine Ablaufsprache im Kontext von IEC 61131-10 können einige allgemein erhältliche XSLT-Dateien verwendet werden. Bei der Konvertierung einer Ablaufsprache des IML-Kontexts (IML SFC) in eine Ablaufsprache nach IEC 61131-10 könnten Informationen verloren gehen.

AML verwendet IEC 61131-10, um die Logikmodelle zu speichern. Um IEC 61131-10 von den Logikmodellen zu entkoppeln, wird in diesem Teil die Intermediate Modelling Layer (IML) festgelegt. Damit bildet IML ein allgemeines Modell für Logikmodelle, mit dem die Möglichkeiten der Vorwärts- und Rückwärtstransformation und der Erweiterbarkeit von AML für neue Logikmodelle verbessert werden. Beispiele hierfür sind ereignisgesteuerte Logikmodelle, wie Zustandsdiagramme.

Da es in diesem Dokument nicht nur einfach um die Speicherung von Steuerungsprogrammen geht, wird die Verwendung von IEC 61131-10 adaptiert. Hierzu wird ein AML logic XML Schema verwendet (siehe Abschnitt 7), das die Speicherung der genannten Logikmodelle erlaubt. Die Modellierungselemente des AML logic XML, mit denen die betreffenden Logikmodelle gespeichert werden können, sind in Bild 5 dargestellt. Auf diese Modellierungselemente können die Logikmodelle abgebildet werden. Dies wird in Abschnitt 8 beschrieben. Anhang A enthält Beispiele.

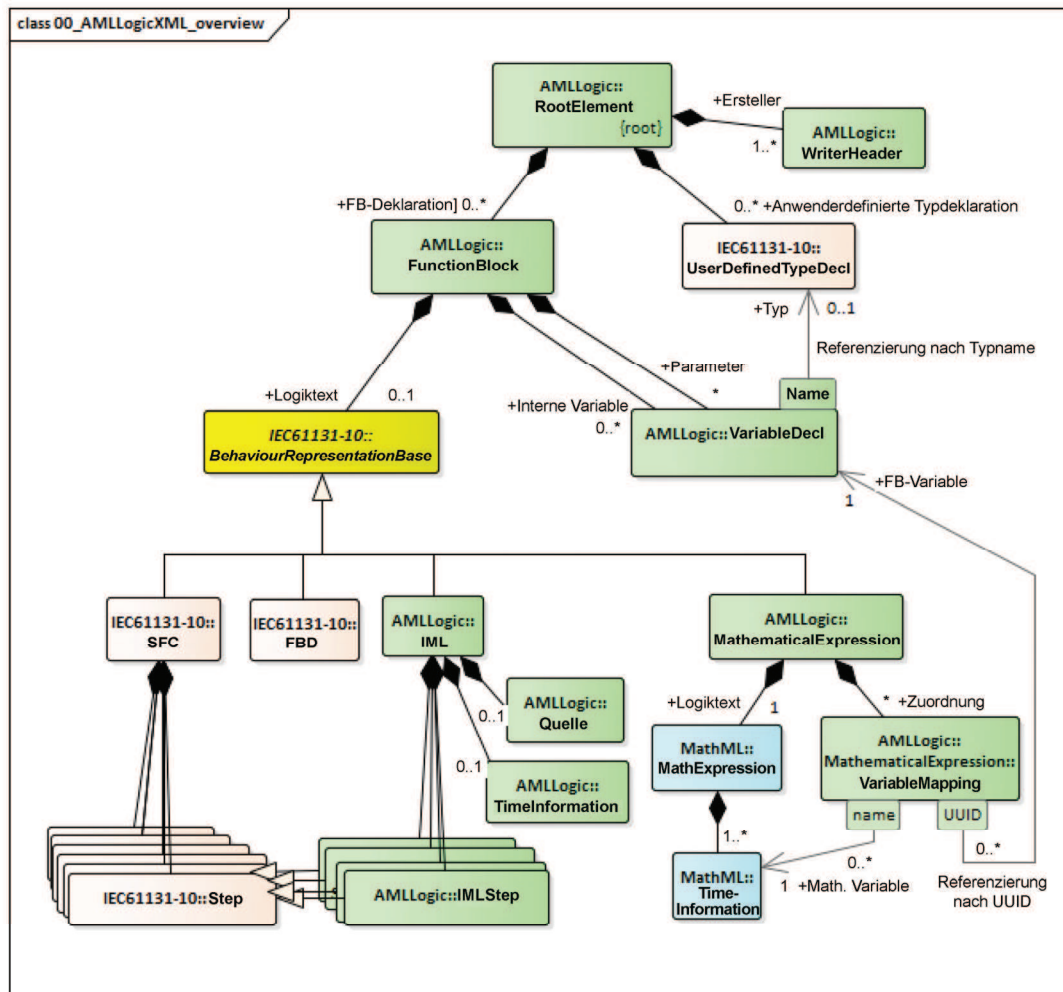


Bild 5 – Modellierungselemente in der AML logic XML

Darüber hinaus sind zur Ermöglichung eines ordnungsgemäßen Datenaustauschs im Anwendungsbereich von AML als „WriterHeader“ bezeichnete dokumentenbezogene Metadaten zum AML logic XML Dokument erforderlich, um das Dokument zu identifizieren und zu charakterisieren. Normative Festlegungen sind in Abschnitt 9 enthalten.

Da das AML logic XML auf IEC 61131-10 basiert, können auch die anderen darauf beruhenden Logikmodelle genutzt werden. In diesem Teil von IEC 62714 werden nur die Ablaufsprache (SFC) und die Funktionsblockdiagramme (FBD) als für den Anwendungsbereich dieses Dokuments relevant angesehen.

Auch ein als mathematischer Ausdruck dargestelltes Logikmodell kann im AML logic XML Dokument gespeichert werden. Dabei enthält entweder das Logikmodell selbst die Verhaltensinformationen des Systems, oder es kann verwendet werden, um einen mathematischen Ausdruck bereitzustellen, der auf die anderen Logikmodelle verweist.

5.4 Referenzierung von Logikinformationen

In der Mehrdokumentenarchitektur von AML werden die Logikinformationen in getrennten AML logic XML Dokumenten gespeichert, die auf dem Datenformat nach IEC 61131-10 basieren. Aus diesem Grund werden die Modellierungs-Logikinformationen in zwei Schritte unterteilt. Zunächst wird ein die Logikinformationen enthaltendes Objekt in dem mit CAEX bezeichneten Toplevel-Format von AML modelliert. Im zweiten Schritt muss ein AML logic XML Dokument bereitgestellt werden, das die betreffenden Logikinformationen enthält. Schließlich wird eine Beziehung zwischen dem CAEX-Objekt und dem AML logic XML Dokument oder einem darin enthaltenen Objekt hergestellt. Normative Festlegungen für die Referenzierung aller drei Informations-

AC | arten werden in Abschnitt 10 und in Abschnitt 11 definiert. Ein informativer Überblick ist in Anhang B enthalten.

Für die Referenzierung von Verriegelungsinformationen müssen weitere Festlegungen berücksichtigt werden. Diese werden normativ im Abschnitt 12 definiert. Ein informativer Überblick ist in Anhang C enthalten.

6 Logikmodelle

6.1 Allgemeines

In diesem Abschnitt werden die Logikmodelle festgelegt, die in diesem Teil von IEC 62714 berücksichtigt werden. Hierzu zählen die grafischen Elemente und ihre strukturellen Regeln, die Logikinformationen, die sie darstellen und wie diese Informationen den Modellelementen zugeordnet werden.

ANMERKUNG Dieses Dokument stellt keine Informationen zur Ausführungslogik bereit, d. h. zur Art und Weise, in der die Modellelemente ausgeführt werden.

6.2 Gantt-Diagramme

6.2.1 Allgemeines

Gantt-Diagramme sind grafische Darstellungen von Zeitplänen, die üblicherweise zur Modellierung von teilweise angeordneten Gruppierungen nacheinander und gleichzeitig ablaufender Aktivitäten verwendet werden. Dazu zählen auch die Ausführungsreihenfolge und -dauer dieser Aktivitäten. Sie enthalten keine Angaben zur Modellierung von alternierenden Pfaden und Zyklen.

ANMERKUNG Gantt-Diagramme werden auch als Balkengrafiken bezeichnet.

6.2.2 Grafische Elemente

In einem Gantt-Diagramm müssen die folgenden grafischen Elemente verwendet werden (siehe Bild 6):

- Eine als allgemeine Zeitachse ausgeführte Zeitleiste (horizontale Achse);
- eine vertikale Achse mit allen Namen der betreffenden Balken;
- keine oder beliebig viele horizontale Balken;
- keine oder beliebig viele gerichtete Pfeile.

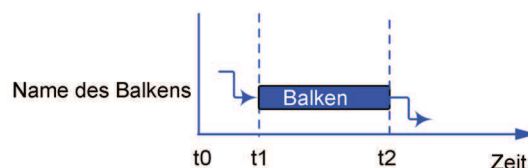


Bild 6 – Modellelemente von Gantt-Diagrammen

6.2.3 Diagrammstruktur

Für ein Gantt-Diagramm muss die folgende Struktur verwendet werden:

- Jedes Gantt-Diagramm muss durch zwei orthogonale Achsen dargestellt werden. Die horizontale Achse stellt die Zeit dar, die zur Abbildung der linearen Zeit mit einer Laufrichtung versehen ist. Die vertikale Achse enthält eine Liste mit den Balkennamen.
- Jeder Balken muss in der grafischen Darstellung auf einer eigenen Zeile stehen und auf der vertikalen Achse einen entsprechenden Namen aufweisen.
- Die zeitbezogenen Start- und Endpunkte des Balkens müssen der horizontalen Achse entsprechen.

- Die Balken können mithilfe von Pfeilen mit anderen Balken verbunden werden.
- Ein Pfeil muss an der rechten Seite eines Balkens beginnen und an der linken Seite eines anderen Balkens enden.

6.2.4 Logikinformationen

Gantt-Diagramme müssen die folgenden Arten von Logikinformationen umfassen:

- Angaben zur Aktivität:
 - Bezeichnung;
 - Startzeitpunkt;
 - Endzeitpunkt;
 - Dauer.
- Keine oder beliebig viele Beziehungsdaten zur vorangegangenen Aktivität.
- Keine oder beliebig viele Beziehungsdaten zur nachfolgenden Aktivität.

6.2.5 Logikinformationen in Gantt-Diagrammen

Im Hinblick auf die Darstellung der Logikinformationen in einem Gantt-Diagramm gelten die folgenden Festlegungen (siehe Bild 7):

- Eine Aktivität muss durch genau einen Balken dargestellt werden.
 - Der Name der Aktivität muss durch den Namen des Balkens auf der vertikalen Achse dargestellt werden.
 - Der Startzeitpunkt der Aktivität muss durch einen zeitlich definierten Startpunkt des Balkens auf der horizontalen Achse dargestellt werden.
 - Der Endzeitpunkt der Aktivität muss durch einen zeitlich definierten Endpunkt des Balkens auf der horizontalen Achse dargestellt werden.
 - Die Dauer der Aktivität muss durch die Länge des Balkens angegeben werden.
- Angaben zur Vorgänger-Nachfolger-Beziehung zwischen Aktivitäten müssen durch einen Pfeil angegeben werden.

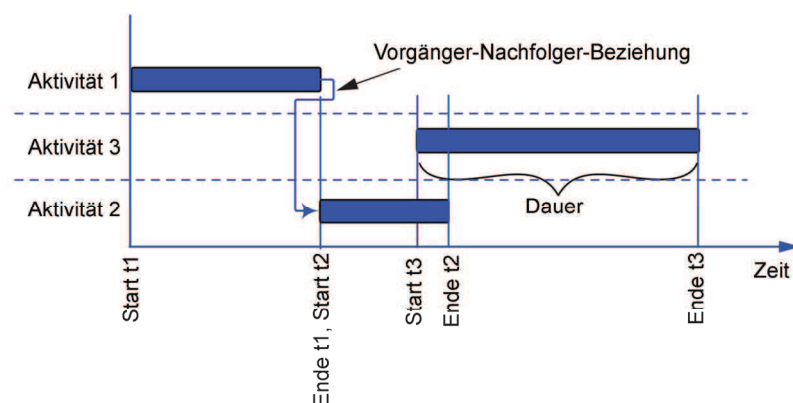


Bild 7 – Daten in Gantt-Diagrammen

6.3 Vorgangsknotennetze

6.3.1 Allgemeines

Mithilfe von Vorgangsknotennetzen werden zeitliche und kausale Zusammenhänge von gegenseitig abhängigen Aktivitäten beschrieben und analysiert. Die Ende-Start-Beziehung zwischen zwei Knoten gibt an,

dass ein Knoten nur dann starten kann, wenn der vorherige Knoten beendet ist. Vorgangsknotennetze enthalten keine Angaben zur Modellierung von alternativen Pfaden und Zyklen.

6.3.2 Grafische Elemente

In einem Vorgangsknotennetz müssen die folgenden grafischen Elemente verwendet werden (siehe Bild 8):

- Keine oder beliebig viele durch Rechtecke dargestellte Knoten, die jeweils in sieben Felder unterteilt sind;
- keine oder beliebig viele gerichtete Pfeile.



Bild 8 – Modellelemente von Vorgangsknotennetzen

6.3.3 Knotenstruktur

Für einen Knoten muss die folgende Struktur verwendet werden:

- Der früheste Start eines Knotens (t_{es}) sollte als Zahl im Feld in der linken oberen Ecke des Knotens eingetragen werden.
- Das früheste Ende eines Knotens (t_{ee}) sollte als Zahl im Feld in der rechten oberen Ecke des Knotens eingetragen werden.
- Die Dauer eines Knotens (t_d) sollte als Zahl im Feld in der Mitte oben des Knotens eingetragen werden. Sie errechnet sich nach der Gleichung $t_d = t_{ee} - t_{es}$ aus dem frühesten Start und dem frühesten Ende des Knotens.
- Der Name des Knotens sollte im Feld in der Mitte des Knotens eingetragen werden.
- Der späteste Start eines Knotens (t_{ls}) sollte als Zahl im Feld in der linken unteren Ecke des Knotens eingetragen werden.
- Das späteste Ende eines Knotens (t_{le}) sollte als Zahl im Feld in der rechten unteren Ecke des Knotens eingetragen werden.
- Der Pufferwert eines Knotens (t_b) sollte als Zahl im Feld in der Mitte unten des Knotens eingetragen werden. Sie errechnet sich nach der Gleichung $t_b = t_{le} - t_{ee}$ aus dem frühesten Ende und dem spätesten Ende des Knotens.

6.3.4 Netzstruktur

Für ein Vorgangsknotennetz muss die folgende Struktur verwendet werden:

- Die Knoten können mithilfe von Pfeilen mit anderen Knoten verbunden werden.
- Ein Pfeil muss an der rechten Seite eines Knotens beginnen und an der linken Seite eines anderen Knotens enden.

6.3.5 Logikinformationen

Vorgangsknotennetze müssen die folgenden Arten von Logikinformationen umfassen:

- Angaben zur Aktivität:
 - Bezeichnung;
 - frühester Start;

- spätestester Start;
- frühestes Ende;
- spätestes Ende;
- Dauer;
- Puffer.
- Angaben zur Vorgänger-Nachfolger-Beziehung zwischen Aktivitäten.

6.3.6 Logikinformationen in Vorgangsknotennetzen

Im Hinblick auf die Darstellung der Logikinformationen in einem Vorgangsknotennetz gelten die folgenden Festlegungen (siehe Bild 9):

- Eine Aktivität muss durch genau einen Knoten dargestellt werden.
 - Die Bezeichnung der Aktivität muss durch den Namen des Knotens angegeben werden.
 - Der früheste Start der Aktivität muss durch den frühesten Start des Knotens angegeben werden.
 - Der späteste Start der Aktivität muss durch den spätesten Start des Knotens angegeben werden.
 - Das früheste Ende der Aktivität muss durch das früheste Ende des Knotens angegeben werden.
 - Das späteste Ende der Aktivität muss durch das späteste Ende des Knotens angegeben werden.
 - Die Dauer der Aktivität muss durch die Dauer des Knotens angegeben werden.
 - Die Verzögerung der Aktivität muss durch den Puffer des Knotens angegeben werden.
- Angaben zur Vorgänger-Nachfolger-Beziehung zwischen Aktivitäten müssen durch einen Pfeil angegeben werden.

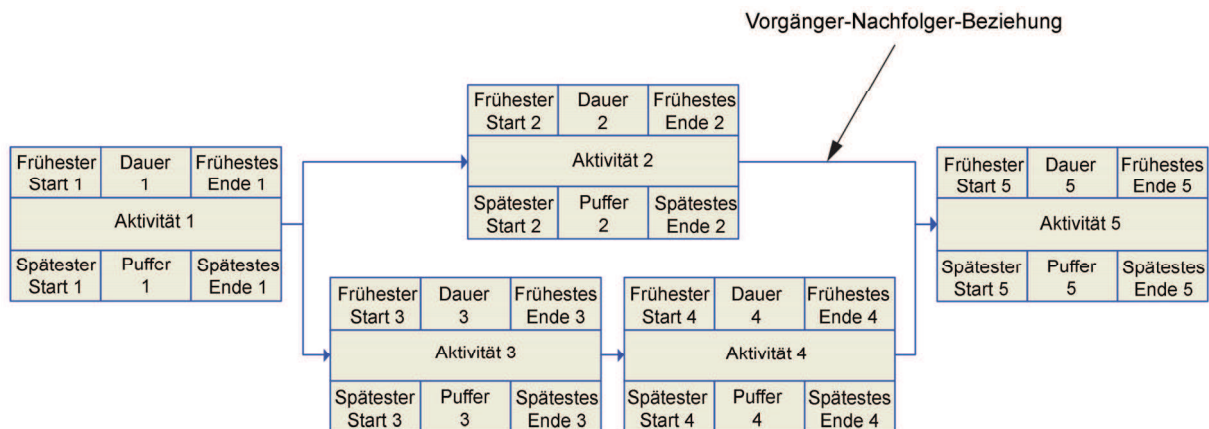


Bild 9 – Daten in Vorgangsknotennetzen

6.4 Zeitverlaufdiagramme

6.4.1 Allgemeines

Mithilfe von Zeitverlaufdiagrammen werden zeitliche und kausale Beziehungen zwischen Signalen und Systemzuständen beschrieben. Außerdem wird dabei beschrieben, wie Änderungen der Systemzustände durch Signale beeinflusst werden und wie Zustandsänderungen die Signale generieren. Zeitverlaufdiagramme ermöglichen die Modellierung von Alternativen.

6.4.2 Grafische Elemente

In einem Zeitverlaufdiagramm müssen die folgenden grafischen Elemente verwendet werden (siehe Bild 10):

- mindestens eine in einer Spalte dargestellte Spur;
- mindestens eine in einer Spalte dargestellte Ebene;
- keine oder beliebig viele gerichtete und bezeichnete Pfeile;
- eine als allgemeine Zeitachse ausgeführte Zeitleiste (horizontale Achse);
- mindestens eine Lebenslinie.

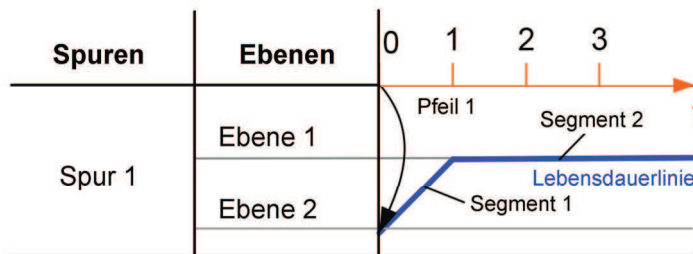


Bild 10 – Modellelemente von Zeitverlaufsdigrammen

6.4.3 Diagrammstruktur

Für ein Zeitverlaufsdigramm muss die folgende Struktur verwendet werden:

- Jede Ebene muss in einem eigenen horizontalen Band dargestellt werden. Die Namen der Ebenen müssen in die jeweiligen Felder in der Ebenenspalte eingetragen werden.
- Zu einer Spur gehörende Ebenen müssen in den jeweiligen Feldern der Spalte „Spuren“ zusammengefasst werden. Die Namen der einzelnen Spuren müssen in die jeweiligen Felder in der Spalte „Spuren“ eingetragen werden. Eine Spur muss durch mindestens eine Ebene dargestellt werden.
- Jede Spur muss eine Lebenslinie aufweisen, die entlang der Zeitachse den Ebenenwechsel darstellt.
- Eine Lebenslinie muss aus mindestens einem Segment bestehen.
- Ein Segment muss durch eine Gerade dargestellt werden.
- Ein Pfeil muss folgende Elemente verbinden:
 - die Zeitachse mit dem Startpunkt eines Segments;
 - den Endpunkt eines Segments mit dem Startpunkt eines anderen Segments auf der gleichen oder einer anderen Lebenslinie.
- Ggf. vorhandene Pfeile müssen am Endpunkt eines Segments beginnen.
- Ggf. vorhandene Pfeile müssen an einem Punkt auf der Zeitachse beginnen.

6.4.4 Logikinformationen

Zeitverlaufsdigramme müssen die folgenden Arten von Logikinformationen umfassen:

- Angaben zur Quelle:
 - Name.
- Angaben zum Zustand:
 - Name;
 - Phase (stellt die Dauer dar);
 - Zustandsfluss.
- Angaben zum Signal:
 - Name;
 - zwei unterschiedliche Typen: Zeit- und Ressourcensignal;
 - Zeitpunkt, zu dem das Signal ausgelöst wird;

- Einfluss eines Signals auf den Zustand einer Quelle.

6.4.5 Logikinformationen in Zeitverlaufdiagrammen

Im Hinblick auf die Darstellung der Logikinformationen in einem Zeitverlaufdiagramm gelten die folgenden Festlegungen (siehe Bild 11):

- Eine Ressource muss durch eine Spur dargestellt werden.
 - Die Bezeichnung der Ressource muss anhand des Namens der Spur angegeben werden.
- Ein Zustand muss durch eine Ebene dargestellt werden.
 - Die Bezeichnung des Zustands muss durch den Namen der Ebene angegeben werden.
 - Zustandsänderungen im zeitlichen Ablauf müssen durch eine Lebenslinie dargestellt werden.
 - Eine Phase muss durch ein Segment dargestellt werden.
- Ein Signal muss durch einen Pfeil dargestellt werden.
 - Die Bezeichnung des Signals muss durch den Namen des Pfeils angegeben werden.
 - Ein Zeitsignal muss durch einen Pfeil angegeben werden, der bei einem Punkt auf der Zeitachse beginnt. Ein Ressourcensignal muss durch einen Pfeil angegeben werden, der am Endpunkt eines Segments auf der Zeitachse beginnt.
 - Der Zeitpunkt, zu dem ein Signal ausgelöst wird, muss einem Punkt auf der x-Achse entsprechen.
 - Der Einfluss eines Signals auf den Zustand einer Ressource muss durch einen neuen Startpunkt eines Segments auf der betreffenden Lebenslinie dargestellt werden.

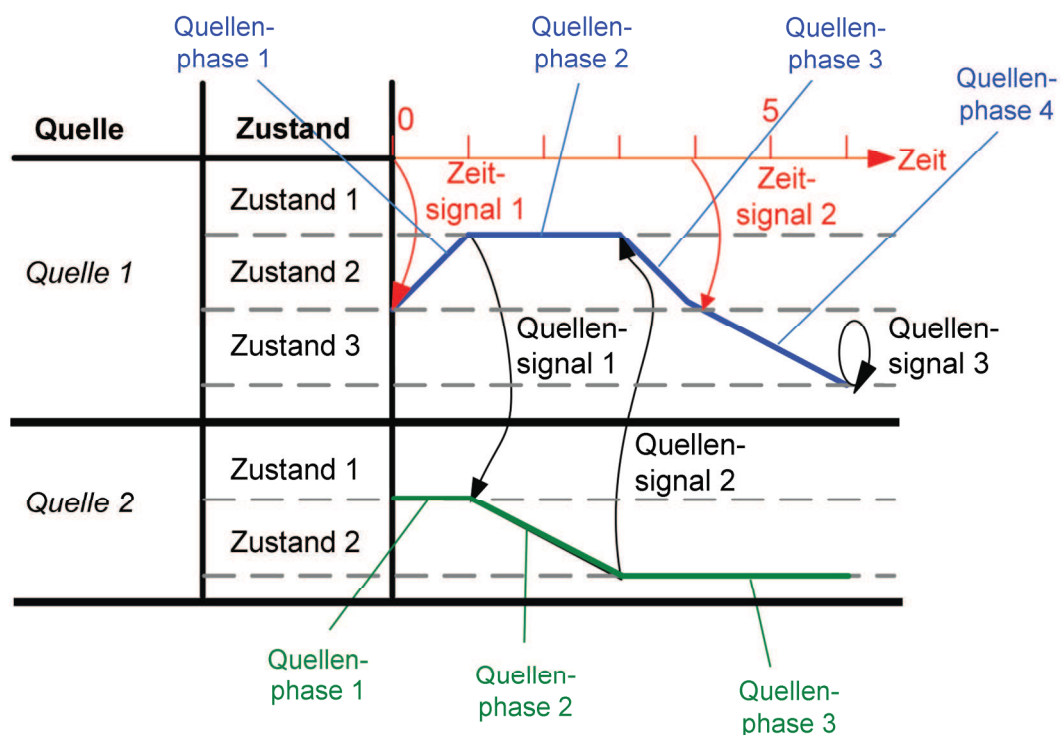


Bild 11 – Informationen in Zeitverlaufdiagrammen

6.5 Ablaufsprache

Eine Ablaufsprache (SFC) beschreibt die kausalen und zeitlichen Abhängigkeiten eines Ablaufs von Systemschritten und -übergängen. Komplexe Abläufe mit Schleifen und bedingter Ausführung können modelliert werden.

In diesem Dokument muss die Ablaufsprache nach den Angaben in IEC 61131-3 verwendet werden.

6.6 Funktionsblockdiagramme

Funktionsblockdiagramme (FBD) beschreiben logische Netze von Funktionsblöcken bzw. wie Eingangs- und Ausgangsvariablen funktional verbunden werden.

In diesem Dokument müssen FBDs nach den Angaben in IEC 61131-3 verwendet werden. Die Nutzung von in IEC 61131-3 festgelegten Funktionsblöcken muss zulässig sein.

6.7 Mathematischer Ausdruck

Mathematische Ausdrücke sind eine Zusammenfassung von Zahlen, absoluten Ausdrücken oder Buchstaben, um mathematische, physikalische, chemische oder logische Beziehungen auszudrücken. Ein mathematischer Ausdruck besteht aus einer Gleichung oder einem Gleichungssystem.

Zur Festlegung von mathematischen Ausdrücken muss dieses Dokument den Inhaltsteil der „Mathematical Markup Language (MathML)“, Version 2.0, nutzen.

7 Beschreibung des AML logic XML Schemas

7.1 Überblick über das Schema

7.1.1 Verwendung des Schemas aus IEC 61131-10

In diesem Abschnitt wird das „AML logic XML Schema“ beschrieben, das formal in Anhang E festgelegt wird.

Das AML logic XML Schema importiert das Schema nach IEC 61131-10 und verwendet teilweise dessen Typdefinitionen, um Variablen, Funktionsblöcke, Abläufe usw. verwendende Logikmodelle darzustellen.

Das Schema nach IEC 61131-10 unterstützt zwei Erweiterungsmechanismen, mit denen die Einbindung von Informationen in einem Austauschformat ermöglicht wird. Der erste Mechanismus ist das Element „AddData“, das unter verschiedene andere Elemente verschachtelt werden kann. Seine Nutzung ist für die Erweiterung anderer Elemente vorgesehen, um diese mit Informationen anzureichern, ohne dass er allerdings für den Austausch des Elements zwischen verschiedenen Werkzeugen erforderlich wäre. Der zweite Mechanismus besteht in der Nutzung eines XML Schemaverfahrens mit der Bezeichnung „abstrakte Typsubstitution“. Aus dem abstrakten Typ kann ein neues Element abgeleitet und dort als Substitution für den Elementtyp verwendet werden, wo der abstrakte Typ angegeben wird.

Das AML logic XML Schema folgt beiden Erweiterungsmechanismen von IEC 61131-10 und den Namenskonventionen für Elemente. Der Überblick über die Schemastruktur ist in Bild 12 im UML-Klassendiagramm dargestellt. Klassen, deren Bezeichnungen mit „IEC61131_10_Ed1_0“ beginnen, stellen aus dem Schema nach IEC 61131-10 importierte Typen dar. Darüber hinaus stellen Klassen, deren Bezeichnungen kursiv (in den gelb unterlegten Feldern) gedruckt sind, abstrakte Typen dar, die durch einen ihrer erweiterten nicht abstrakten Typen ersetzt werden müssen.

- Wie aus Bild 12 hervorgeht, werden die anwenderdefinierten Datentypdeklarationen und ihre Referenzierungen aus der Variablendeklaration vom AML logic XML Schema vollständig wiederverwendet.
- Die Logikbeschreibungen und ihre Basis-Typen aus IEC 61131-10 werden ebenfalls vom AML logic XML Schema wiederverwendet. Siehe hierzu Bild 12. Damit stehen alle Logikbeschreibungen aus IEC 61131-10 in der AML logic XML zur Verfügung. Außerdem fügt das AML logic XML Schema durch die „Substitution abstrakter Typen“, also durch den zweiten aus IEC 61131-10 folgenden Erweiterungsmechanismus, spezifische Logikmodelle wie IML und mathematische Ausdrücke hinzu.
- Das AML logic XML Schema erweitert nach der Darstellung in Bild 12 die grundlegenden SFC-Elemente von IEC 61131-10, um den in 5.3 bezeichneten Intermediate Modelling Layer (IML) abzubilden.

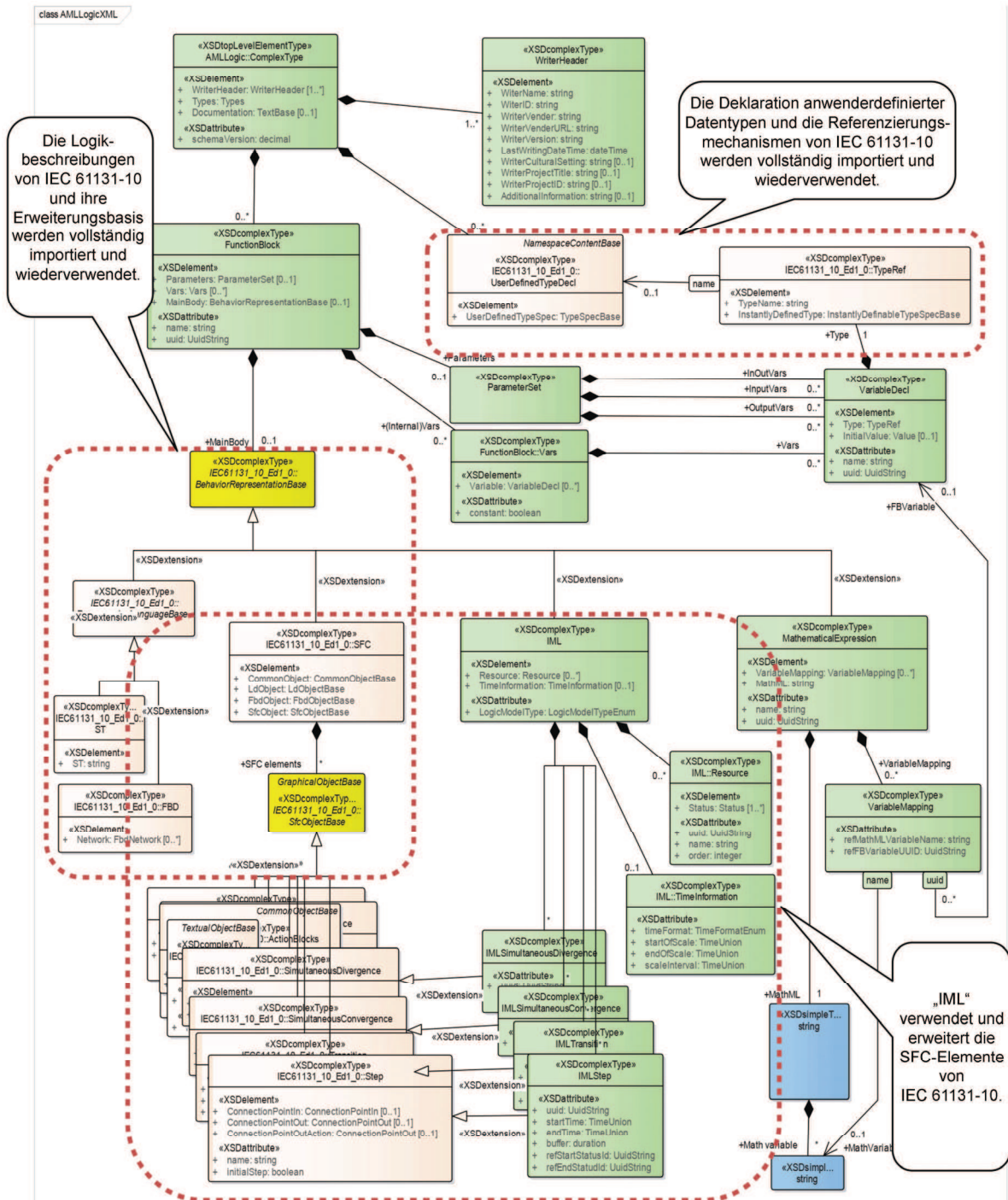


Bild 12 – Überblick über das AML-Schema

7.1.2 Schemaversionierung

Die AML logic XML Schemadefinition weist ein Versionsattribut auf. Die erste Zahl bezeichnet dabei die Ausgabennummer von IEC 62714-4. Die zweite Zahl ist für kleinere Änderungen innerhalb einer Ausgabe reserviert. Der Dateiname des Schemas weist ebenfalls die Ausgabennummer aus.

Zusätzlich hat auch das Root-Element „AMLLogic“ ein Attribut „schemaVersion“. Bei der Erstellung eines XML Dokuments auf der Grundlage dieses Schemas muss seine Versionsnummer auf den Wert dieses Attributs gesetzt werden.

7.2 Root-Element „AMLLogic“

7.2.1 Allgemeines

Das Root-Element „AMLLogic“ stellt eine Art Projekt oder Bibliothek dar, in der/dem Logikinformationen gespeichert sind. Sein Inhalt ist in Bild 13 abgebildet.

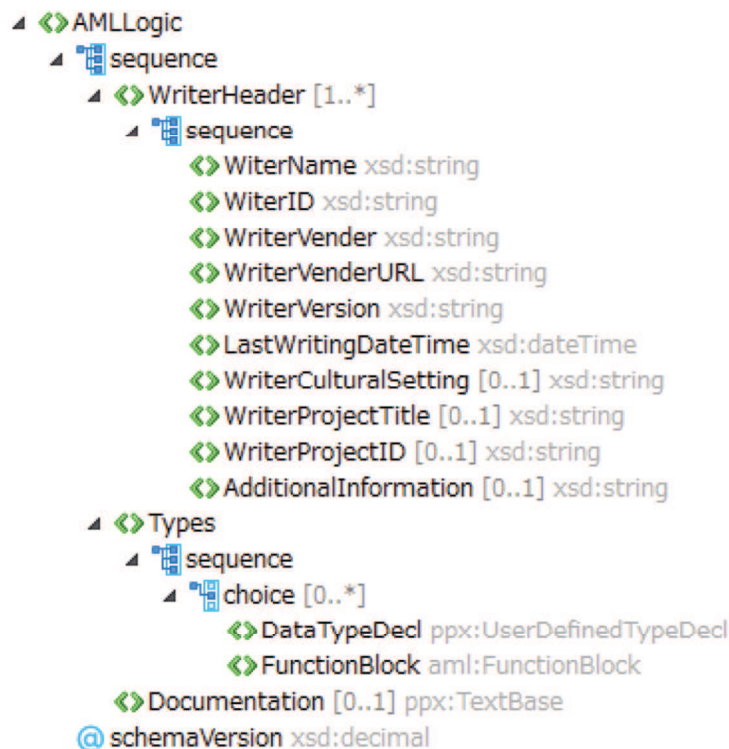


Bild 13 – Root-Element „AMLLogic“

7.2.2 Attribute

Das Attribut „schemaVersion“ muss die Versionsnummer des AML logic XML Schemas enthalten, mit der auch die Versionsnummer des XML Dokuments übereinstimmt.

7.2.3 Untergeordnetes Element „WriterHeader“

Wie im Abschnitt 9 erläutert, enthält das Element „WriterHeader“ Metadaten zu dem Werkzeug, mit dem das XML Dokument erstellt wird.

7.2.4 Untergeordnetes Element „Types“

Das Element „Types“ ist ein Platzhalter für die Deklaration des anwenderdefinierten Datentyps und/oder des Funktionsblocks.

Das Element „DataTypeDecl“ des Typs „ppx:UserDefinedTypeDecl“ stellt die Deklaration des anwenderdefinierten Datentyps in der Sprachspezifikation nach IEC 61131-3 dar. Damit kann auch jede Art von Datentyp, die in der Sprachspezifikation von IEC 61131-3 enthalten ist, mithilfe des AML logic XML festgelegt werden. IEC 61131-10 sollte für detaillierte Spezifikationen des Typs „ppx:UserDefinedTypeDecl“ referenziert werden.

Das Element „FunctionBlock“ vom Typ „aml:FunctionBlock“ gibt eine Art von Funktionsblock für die AML-Logikbeschreibung an, in der nur die erforderlichen Merkmale des Funktionsblocks nach IEC 61131-3 unterstützt werden. Sein Inhalt wird in 7.3 erläutert.

7.2.5 Untergeordnetes Element „Documentation“

Das Element „Documentation“ enthält eine allgemeine Beschreibung des Elements „AMLLogic“. Der Typ „ppx: TextBase“ ist ein abstrakter Typ, der aus IEC 61131-10 importiert wurde und durch den Klartext von „ppx: SimpleText“ ersetzt wird.

7.3 Komplexer Typ „FunctionBlock“

7.3.1 Allgemeines

Der komplexe Typ „FunctionBlock“ stellt (wie aus Bild 14 hervorgeht) eine Art von Funktionsblock für die AML-Logikbeschreibung dar, bei dem nur die erforderlichen Merkmale des Funktionsblocks aus IEC 61131-3 unterstützt werden.

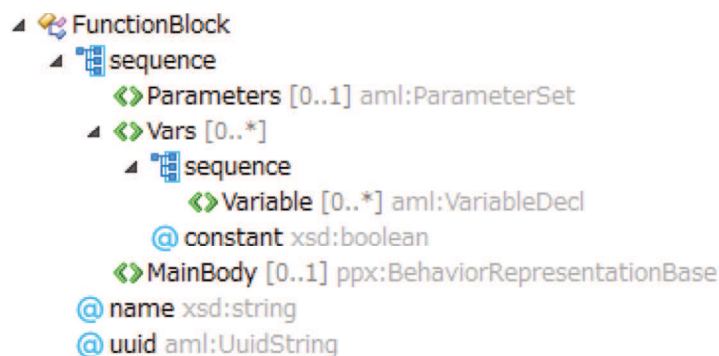


Bild 14 – Komplexer Typ „FunctionBlock“

7.3.2 Attribute

Das Attribut „name“ von „xsd:string“ gibt den Typnamen des Funktionsblocks an.

Das Attribut „uuid“ des Typs „aml:UuidString“ gibt den universellen eindeutigen Bezeichner der Typdefinition des Funktionsblocks an. Der einfache Typ „UuidString“ wird in 7.9 erläutert.

7.3.3 Untergeordnetes Element „Parameters“

Das Element „Parameters“ vom Typ „aml:ParameterSet“ gibt die Deklaration der Eingangs-, Ausgangs- oder Eingangs-/Ausgangsvariable des Funktionsblocks an. Der Typ „ParameterSet“ wird detailliert in 7.3.6 erläutert.

7.3.4 Untergeordnetes Element „Vars“

Das Element „Vars“ deklariert die internen Variablen des Funktionsblocks.

Das Element „Variable“ des Typs „aml:VariableDecl“ enthält die Variablendeklaration. Der komplexe Typ „aml:VariableDecl“ wird in 7.3.7 erläutert.

Das Attribut „constant“ legt fest, dass die Variablendeklaration im Element „Vars“ Variablen mit konstanten Werten sind.

7.3.5 Untergeordnetes Element „MainBody“

Das Element „MainBody“ des Typs „ppx:BehaviorRepresentationBase“ stellt ein Logikmodell dar, das bei Ausführung des Funktionsblocks evaluiert werden muss. Jeder von „ppx:BehaviorRepresentationBase“ abgeleitete komplexe, nicht abstrakte Typ kann gegen den Typ des Elements ausgetauscht werden. Damit stehen auch alle im Schema nach IEC 61131-10 definierten Logikmodelle in diesem Funktionsblock des AML

logic XML zur Verfügung, wie beispielsweise die Ablaufsprache (SFC), das Funktionsblockdiagramm (FBD), der Kontaktplan (LD) und der strukturierte Text (ST).

7.3.6 Komplexer Typ „ParameterSet“

Der komplexe Typ „ParameterSet“ (siehe Bild 15) stellt einen Deklarationsatz bestehend aus Eingangsvariablen, Ausgangsvariablen und Eingangs-/Ausgangsvariablen dar.

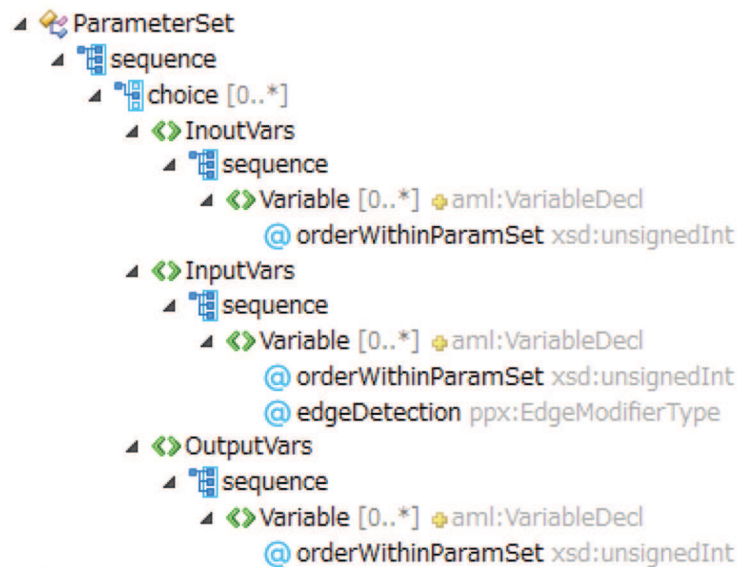


Bild 15 – Komplexer Typ „ParameterSet“

Das Element „InoutVars“ bezeichnet den Deklarationsabschnitt für Eingangs-/Ausgangsvariablen. Auf das Element „Variable“ in „InoutVars“ bezogen wird sein Typ ausgehend vom Typ „aml:VariableDecl“ erweitert, der in 7.3.7 erläutert wird. Das Attribut „orderWithinParamSet“ wird dabei hinzugefügt. Der Grund für diese Erweiterung besteht darin, dass die Reihenfolge der Parameter eines Funktionsblocks von Bedeutung ist, die Reihenfolge der Elemente eines Elementtyps in einem XML Dokument dagegen nicht als signifikant sichergestellt ist.

Das Element „InputVars“ enthält einen Deklarationsabschnitt für die Eingangsvariablen des Funktionsblocks. Auf das Element „Variable“ in „InputVars“ bezogen wird sein Typ ausgehend vom Typ „aml:VariableDecl“ erweitert, der ebenfalls in 7.3.7 erläutert wird. Das Attribut „edgeDetection“ wird dabei zusätzlich zum Attribut „orderWithinParamSet“ hinzugefügt.

Das Element „OutputVars“ enthält einen Deklarationsabschnitt für die Ausgangsvariablen des Funktionsblocks. Auf das Element „Variable“ in „OutputVars“ bezogen wird sein Typ ausgehend vom Typ „aml:VariableDecl“ erweitert, der ebenfalls in 7.3.7 erläutert wird. Das Attribut „orderWithinParamSet“ wird dabei ebenso hinzugefügt.

7.3.7 Komplexer Typ „VariableDecl“

Der komplexe Typ „VariableDecl“ (siehe Bild 16) stellt einen Deklarationsabschnitt für Variablen dar.

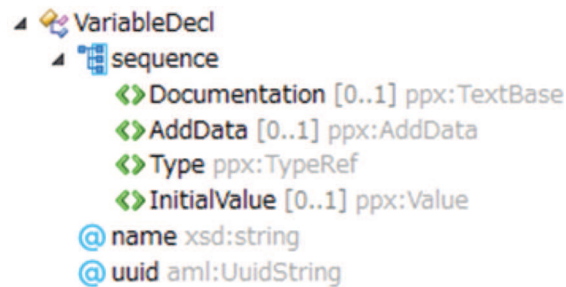


Bild 16 – Komplexer Typ „VariableDecl“

Das Element „Documentation“ des Typs „ppx:TextBase“ enthält die Beschreibung der Variable.

Das Element „AddData“ des Typs „ppx:AddData“ kann verwendet werden, um jeden Typ von zusätzlichen Daten zu speichern.

Das Element „Type“ des Typs „ppx:TypeRef“ enthält die Referenzierung der Typdefinition der Variable. Dabei kann es sich um eine Typbezeichnung oder eine Instanzendeklaration handeln, wie „ARRAY [0..8] OF INT“. Für Einzelheiten des komplexen Typs „ppx:TypeRef“ sollte auf IEC 61131-10 verwiesen werden.

Das Element „InitialValue“ des Typs „ppx:Value“ enthält den Ausgangswert der Variable. Dabei kann es sich, je nach Datentyp der Variable, um einen einzelnen Wert, einen Feldwert für einen Feldtyp oder, bei einem strukturierten Datentyp, um einen strukturierten Wert handeln. Für Einzelheiten des komplexen Typs „ppx:Value“ sollte auf die Spezifikation von IEC 61131-10 verwiesen werden.

Das Attribut „name“ des Typs „xsd:string“ gibt den Namen der Variable an.

Das Attribut „uuid“ des Typs „aml:UuidString“ gibt den universellen eindeutigen Bezeichner der Variable an. Der einfache Typ „UuidString“ wird in 7.9 erläutert.

7.4 Komplexer Typ „IML“

7.4.1 Allgemeines

Der komplexe Typ „IML“ (siehe Bild 17) stellt den Intermediate Modelling Layer mehrerer bestimmter Logikmodelle nach der Beschreibung in 5.3 dar. Dieser komplexe Typ kann als Ersatz für den Typ des Elements „MainBody“ im Funktionsblock der AML logic XML verwendet werden, da er nach der Beschreibung in Bild 12 „ppx:BehaviorRepresentationBase“ erweitert.

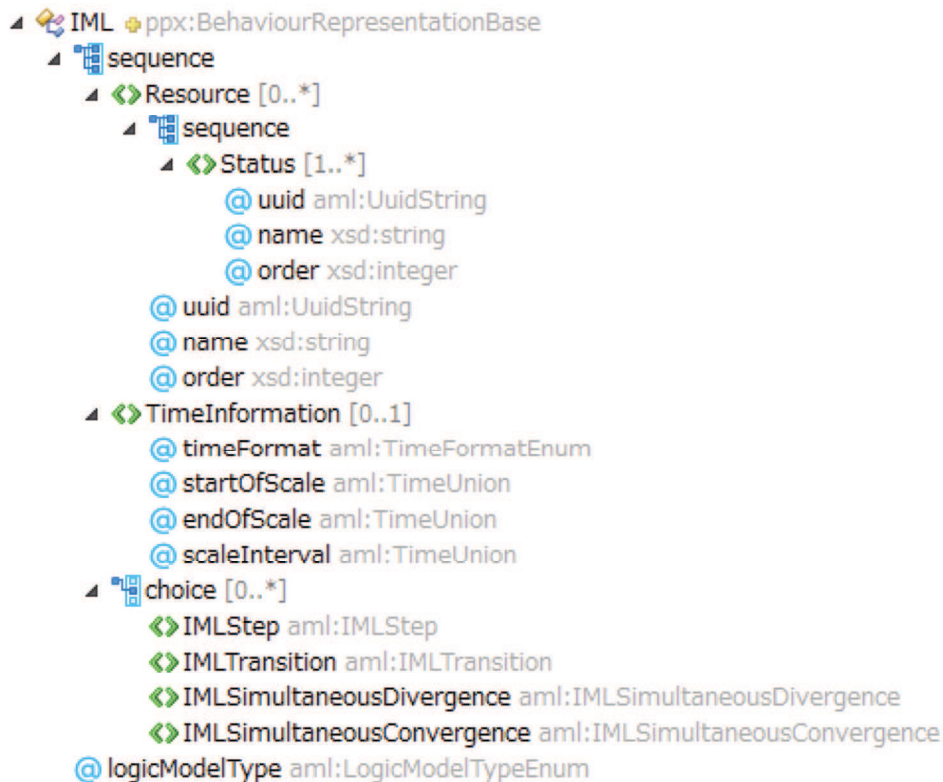


Bild 17 – Komplexer Typ „IML“

7.4.2 Attribute

Das Attribut „logicModelType“ des Typs „aml:LogicModelTypeEnum“ legt fest, welche Art von Logikmodell mit der IML dargestellt wird. Der einfache Typ „LogicModelTypeEnum“ wird in 7.6 erläutert.

7.4.3 Untergeordnetes Element „Resource“

Das Element „Resource“ gibt zusätzliche Logikinformationen an, die nach der Beschreibung in 6.4.4 für das Zeitverlaufsdiagramm erforderlich sind.

- Das Attribut „uuid“ gibt den universellen eindeutigen Bezeichner des „Resource“-Elements an. Der einfache Typ „UuidString“ wird in 7.9 erläutert.
- Das Attribut „name“ gibt den Namen des „Resource“-Elements an.
- Das Attribut „order“ gibt die Reihenfolge des „Resource“-Elements in der IML an. Kleinere Zahlen sollten an erster Stelle aufgeführt werden.

„Status“-Elemente geben mögliche Zustände des „Resource“-Elements an.

- Das Attribut „uuid“ gibt den universellen eindeutigen Bezeichner des „Status“-Elements an. Der einfache Typ „UuidString“ wird in 7.9 erläutert.
- Das Attribut „name“ gibt den Namen des „Status“-Elements an.
- Das Attribut „order“ gibt die Reihenfolge des „Status“-Elements im „Resource“-Element an. Kleinere Zahlen sollten an erster Stelle aufgeführt werden.

7.4.4 Untergeordnetes Element „TimeInformation“

„TimeInformation“ gibt die logische Zeitskala an, die insbesondere für Gantt- oder Zeitverlaufsdiagramme verwendet wird.

- Das Attribut „timeFormat“ des Typs „aml:TimeFormatEnum“ stellt das Format des Zeitausdrucks in dieser IML dar. Der einfache Typ „aml:TimeFormatEnum“ wird in 7.8 erläutert.