

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Engineering data exchange format for use in industrial automation systems
engineering – Automation markup language –
Part 4: Logic**

**Format d'échange de données pour une utilisation dans l'ingénierie des
systèmes d'automatisation industrielle – Automation markup language –
Partie 4: Logique**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2020 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

Electropedia - www.electropedia.org

The world's leading online dictionary on electrotechnology, containing more than 22 000 terminological entries in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

67 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC

Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Recherche de publications IEC -

webstore.iec.ch/advsearchform

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études,...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et une fois par mois par email.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: sales@iec.ch.

Electropedia - www.electropedia.org

Le premier dictionnaire d'électrotechnologie en ligne au monde, avec plus de 22 000 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 16 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

67 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Engineering data exchange format for use in industrial automation systems
engineering – Automation markup language –
Part 4: Logic**

**Format d'échange de données pour une utilisation dans l'ingénierie des
systèmes d'automatisation industrielle – Automation markup language –
Partie 4: Logique**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 25.040.40

ISBN 978-2-8322-8399-8

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

CONTENTS

FOREWORD	9
INTRODUCTION	11
1 Scope	13
2 Normative references	13
3 Terms, definitions and abbreviated terms	14
3.1 Terms and definitions	14
3.2 Abbreviated terms	15
4 Conformity	16
5 Overview	16
5.1 Logic information in production system engineering	16
5.2 Logic models in production system engineering	17
5.3 Storing logic models in AML logic XML	18
5.4 Referencing logic information	20
6 Logic models	21
6.1 General	21
6.2 Gantt charts	21
6.2.1 General	21
6.2.2 Graphical elements	21
6.2.3 Chart structure	21
6.2.4 Logic information	22
6.2.5 Logic information within Gantt chart	22
6.3 Activity-on-node networks	22
6.3.1 General	22
6.3.2 Graphical elements	22
6.3.3 Node structure	23
6.3.4 Network structure	23
6.3.5 Logic information	23
6.3.6 Logic information within activity-on-node networks	23
6.4 Timing diagrams	24
6.4.1 General	24
6.4.2 Graphical elements	24
6.4.3 Diagram structure	25
6.4.4 Logic information	25
6.4.5 Logic information within timing diagrams	26
6.5 Sequential function charts	26
6.6 Function block diagrams	27
6.7 Mathematical expression	27
7 AML logic XML schema description	27
7.1 Schema overview	27
7.1.1 Use of IEC 61131-10 schema	27
7.1.2 Schema versioning	28
7.2 Root element "AMLogic"	29
7.2.1 General	29
7.2.2 Attributes	29
7.2.3 Sub-element "WriterHeader"	29

7.2.4	Sub-element "Types"	29
7.2.5	Sub-element "Documentation"	30
7.3	Complex type "FunctionBlock"	30
7.3.1	General	30
7.3.2	Attributes	30
7.3.3	Sub-element "Parameters"	30
7.3.4	Sub-element "Vars"	30
7.3.5	Sub-element "MainBody"	30
7.3.6	Complex type "ParameterSet"	31
7.3.7	Complex type "VariableDecl"	31
7.4	Complex Type "IML"	32
7.4.1	General	32
7.4.2	Attributes	33
7.4.3	Sub-element "Resource"	33
7.4.4	Sub-element "TimeInformation"	33
7.4.5	Choice of "IML****" element	33
7.4.6	Complex type "IMLStep"	33
7.4.7	Complex type "IMLTransition"	34
7.4.8	Complex type "IMLSimultaneousDivergence"	35
7.4.9	Complex type "IMLSimultaneousConvergence"	35
7.5	Complex Type "MathematicalExpression"	35
7.5.1	General	35
7.5.2	Attributes	36
7.5.3	Sub-element "VariableMapping"	36
7.5.4	Sub-element "MathML"	36
7.6	Simple type "LogicModelTypeEnum"	36
7.7	Simple type "TimeUnion"	36
7.8	Simple type "TimeFormatEnum"	36
7.9	Simple type "UuidString"	37
8	Storing logic models	37
8.1	General	37
8.2	Storing Gantt charts in AML logic XML	37
8.2.1	Common rules	37
8.2.2	Storing the start of a Gantt chart	37
8.2.3	Storing bars	38
8.2.4	Storing arrows	38
8.2.5	Storing successor bars	39
8.2.6	Storing predecessor bars	41
8.3	Storing activity-on-node networks in AML logic XML	42
8.3.1	Common rules	42
8.3.2	Storing the start of an activity-on-node network	42
8.3.3	Storing nodes	43
8.3.4	Storing arrows	43
8.3.5	Storing successor nodes	44
8.3.6	Storing predecessor nodes	45
8.4	Storing timing diagrams in AML logic XML	46
8.4.1	Common rules	46
8.4.2	Storing the timeline of a timing diagram	47
8.4.3	Storing resources and resource states	47

8.4.4	Storing lifelines.....	48
8.4.5	Storing the time signal and the resource signal.....	50
8.5	Storing sequential function charts in AML logic XML	51
8.5.1	Common rules	51
8.5.2	Storing variables.....	51
8.6	Storing function block diagrams in AML logic XML	52
8.6.1	Common rules	52
8.6.2	Storing variables.....	52
8.7	Storing mathematical expressions in AML logic XML.....	53
8.7.1	Common rules	53
8.7.2	Storing variables.....	54
8.7.3	Storing variable mappings	54
8.7.4	Storing mathematical expressions	55
9	Meta information about AML logic XML writer tools.....	55
10	Extensions of AML classes for logic.....	56
10.1	General.....	56
10.2	AutomationMLLogicRoleClassLib	56
10.2.1	General	56
10.2.2	RoleClass InterlockingTargetGroup	57
10.2.3	RoleClass InterlockingSourceGroup	57
10.2.4	RoleClass LogicModelObject	58
10.3	AutomationMLLogicInterfaceClassLib	59
10.3.1	General	59
10.3.2	InterfaceClass LogicModelInterface	59
10.3.3	InterfaceClass SequencingLogicModelInterface	59
10.3.4	InterfaceClass BehaviourLogicModelInterface	60
10.3.5	InterfaceClass InterlockingLogicModelInterface	60
10.3.6	InterfaceClass LogicModelElementInterface	60
10.3.7	InterfaceClass VariableInterface	61
10.3.8	InterfaceClass InterlockingVariableInterface	61
10.4	AutomationMLPLCopenXMLInterfaceClassLib.....	62
10.4.1	General.....	62
10.4.2	InterfaceClass VariableInterface	62
10.5	AutomationMLInterfaceClassLib.....	63
10.5.1	General	63
10.5.2	InterfaceClass InterlockingConnector	63
10.5.3	InterfaceClass PLCopenXMLInterface	63
11	Referencing AML logic XML documents.....	63
11.1	General.....	63
11.2	Referencing logic information.....	63
12	Linking AML objects with interlocking information	64
12.1	General.....	64
12.2	Referencing interlocking information	64
Annex A (informative)	Examples for storing logic models in AML logic XML	66
A.1	Example for storing Gantt charts.....	66
A.1.1	General	66
A.1.2	Storing of activities without predecessor and successor relation	66
A.1.3	Storing of an activity sequence	67

A.1.4	Storing of an activity sequence with divergences	69
A.1.5	Storing of an activity sequence with convergences	70
A.2	Example for storing activity-on-node networks	71
A.2.1	General	71
A.2.2	Storing of activities without predecessor and successor relation	71
A.2.3	Storing of an activity sequence	72
A.2.4	Storing of an activity sequence with divergences	73
A.2.5	Storing of an activity sequence with convergences	75
A.3	Example for storing timing diagrams	76
A.3.1	General	76
A.3.2	Example of storing internal signal	77
A.3.3	Example of storing external signal	78
A.3.4	Example of storing signal between two resource states flows	79
A.4	Example for storing sequential function charts	81
A.5	Example for storing function block diagrams	83
A.6	Example for storing mathematical expressions	85
Annex B (informative)	Examples for referencing logic information	89
B.1	General	89
B.2	Referencing logic information expressed as logic models	89
B.2.1	General	89
B.2.2	Referencing logic information stored in one FunctionBlock	89
B.2.3	Referencing logic information, which is composed of several FunctionBlocks	90
B.2.4	Referencing logic information, which is composed of several AML logic XML documents	90
B.3	Referencing logic information as a part of logic models	92
B.3.1	General	92
B.3.2	Referencing a variable	92
B.3.3	Referencing a logic element	93
B.4	Referencing logic information as a part of already referenced logic models	93
Annex C (informative)	Examples for referencing interlocking information	96
C.1	General	96
C.2	Interlocking information	97
C.3	Referencing interlocking information without interlocking condition	97
C.4	Referencing interlocking information with interlocking condition	99
Annex D (normative)	XML representation of AML standard libraries	103
D.1	General	103
D.2	AutomationMLLogicRoleClassLib	103
D.3	AutomationMLLogicInterfaceClassLib	104
D.4	AutomationMLPLCOpenXMLInterfaceClassLib	105
Annex E (normative)	XML representation of AML logic XML schema	106
Bibliography		110
Figure 1	– Overview of the engineering data exchange format AML	11
Figure 2	– Example of system representation with roles of information in AML	17
Figure 3	– Logic models in AML	18
Figure 4	– Storing logic models in AML logic XML	19
Figure 5	– Modelling elements of the AML logic XML	20

Figure 6 – Model elements of Gantt charts.....	21
Figure 7 – Information provided by Gantt charts.....	22
Figure 8 – Model elements of activity-on-node networks	23
Figure 9 – Information provided by activity-on-node networks	24
Figure 10 – Model elements of timing diagrams	25
Figure 11 – Information provided by timing diagrams	26
Figure 12 – AML logic schema overview	28
Figure 13 – Root element "AMLogic".....	29
Figure 14 – Complex type "FunctionBlock".....	30
Figure 15 – Complex type "ParameterSet"	31
Figure 16 – Complex type "VariableDecl".....	31
Figure 17 – Complex type "IML".....	32
Figure 18 – Complex type "IMLStep".....	34
Figure 19 – Complex type "IMLTransition"	34
Figure 20 – Complex type "IMLSimultaneousDivergence"	35
Figure 21 – Complex type "IMLSimultaneousConvergence"	35
Figure 22 – Complex type "MathematicalExpression".....	35
Figure 23 – AutomationMLLogicRoleClassLib	57
Figure 24 – AutomationMLLogicInterfaceClassLib.....	59
Figure 25 – AutomationMLPLCopenXMLInterfaceClassLib.....	62
Figure A.1 – Flow rate of valves.....	86
Figure A.2 – Example for storing a mathematical expression	88
Figure B.1 – Referencing logic information (as SFC) stored in one FunctionBlock.....	89
Figure B.2 – XML text of the CAEX file for referencing logic information stored in one FunctionBlock	90
Figure B.3 – Referencing logic information, which is composed of several FunctionBlocks	90
Figure B.4 – Referencing logic information which is composed of several AML logic XML documents.....	91
Figure B.5 – XML text of the CAEX file for referencing logic information, which is composed of several AML logic XML documents.....	91
Figure B.6 – Referencing a variable	92
Figure B.7 – XML text of the CAEX file for referencing a variable.....	92
Figure B.8 – Referencing a logic element.....	93
Figure B.9 – XML text of the CAEX file for referencing a logic element	93
Figure B.10 – Referencing a variable of an already referenced logic model.....	94
Figure B.11 – XML text of the CAEX file for referencing a variable of an already referenced logic model.....	95
Figure C.1 – Example manufacturing system	96
Figure C.2 – Example interlocking source group and interlocking target group.....	97
Figure C.3 – Referencing interlocking information without interlocking condition	98
Figure C.4 – XML text of the CAEX file for referencing interlocking information without interlocking condition	99
Figure C.5 – Referencing interlocking information with interlocking condition	100

Figure C.6 – XML text of the CAEX file for referencing interlocking information with interlocking condition	101
Figure C.7 – Linking logical interface with physical interface (extension to Figure C.5)	102
Figure C.8 – XML text of the CAEX file for linking logical interface with physical interface (extension to Figure C.6)	102
Table 1 – Abbreviated terms	16
Table 2 – Mapping the start of a Gantt chart to IML element	38
Table 3 – Mapping of Gantt chart bars to IML elements	38
Table 4 – Mapping of Gantt chart arrow to IML elements	39
Table 5 – Mapping of Gantt chart bar with one or more successor bars to IML elements	40
Table 6 – Mapping of Gantt chart bar with one and more predecessor bars to IML elements	41
Table 7 – Mapping of the start of an activity-on-node network to IML elements	42
Table 8 – Mapping of activity-on-node network nodes to IML elements	43
Table 9 – Mapping of activity-on-node network arrows to IML elements	44
Table 10 – Mapping of activity-on-node network successor nodes to IML elements	44
Table 11 – Mapping of activity-on-node network predecessor nodes to IML elements	45
Table 12 – Mapping of the timeline of a timing diagram to IML elements	47
Table 13 – Mapping of timing diagram resources and resource states to IML elements	48
Table 14 – Mapping of timing diagram lifelines to IML elements	48
Table 15 – Mapping of the time signal and the resource signal to IML elements	50
Table 16 – Storing variable of a sequential function chart in AML logic XML	52
Table 17 – Storing variable of a function block diagram in AML logic XML	53
Table 18 – Storing variable of a mathematical expression in AML logic XML	54
Table 19 – Storing variable mappings of a mathematical expression in AML logic XML	55
Table 20 – Storing a mathematical expression in AML logic XML	55
Table 21 – Meta information about each AML logic XML writer tool	56
Table 22 – RoleClass InterlockingTargetGroup	57
Table 23 – RoleClass InterlockingSourceGroup	58
Table 24 – RoleClass LogicModelObject	58
Table 25 – InterfaceClass LogicModelInterface	59
Table 26 – InterfaceClass SequencingLogicModelInterface	60
Table 27 – InterfaceClass BehaviourLogicModelInterface	60
Table 28 – InterfaceClass InterlockingLogicModelInterface	60
Table 29 – InterfaceClass LogicModelElementInterface	61
Table 30 – InterfaceClass VariableInterface	61
Table 31 – InterfaceClass InterlockingVariableInterface	62
Table 32 – InterfaceClass VariableInterface	62
Table 33 – InterfaceClass InterlockingConnector	63
Table 34 – InterfaceClass PLCOpenXMLInterface	63
Table A.1 – Storing of the Gantt chart example "activities without predecessor and successor relations"	67
Table A.2 – Storing of the Gantt chart example "activity sequence"	68

Table A.3 – Storing of the Gantt chart example "activity sequence with divergence"	69
Table A.4 – Storing of the Gantt chart example "activity sequence with convergences"	70
Table A.5 – Storing of the activity-on-node network example "activities without predecessor and successor relations"	72
Table A.6 – Storing of the activity-on-node network example "activity sequence"	72
Table A.7 – Storing of the activity-on-node network example "activity sequence with divergence"	74
Table A.8 – Storing of the activity-on-node network example "activity sequence with convergences"	75
Table A.9 – Storing of the timing diagram example "transition from a state change to the subsequent state"	77
Table A.10 – Mapping of the timing diagram example "two external signals fired with delay of three seconds"	78
Table A.11 – Storing of the timing diagram example "signal fired by one resource state and consumed by another"	80
Table A.12 – Example for storing sequential function chart	81
Table A.13 – Example for storing a function block diagram	83

IECNORM.COM : Click to view the full PDF of IEC 62714-4:2020

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**ENGINEERING DATA EXCHANGE FORMAT FOR USE
IN INDUSTRIAL AUTOMATION SYSTEMS ENGINEERING –
AUTOMATION MARKUP LANGUAGE –****Part 4: Logic****FOREWORD**

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62714-4 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

The text of this International Standard is based on the following documents:

CDV	Report on voting
65E/654/CDV	65E/692/RVC

Full information on the voting for the approval of this International Standard can be found in the report on voting indicated in the above table.

This document has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts in the IEC 62714 series, published under the general title *Engineering data exchange format for use in industrial systems engineering – Automation Markup Language*, can be found on the IEC website.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under "<http://webstore.iec.ch>" in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

The contents of the corrigendum of November 2020 have been included in this copy.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

IECNORM.COM : Click to view the full PDF of IEC 62714-4:2020

INTRODUCTION

The data exchange format defined in IEC 62714 (Automation Markup Language (AML)) is an XML schema-based data format and has been developed in order to support the data exchange between engineering tools in a heterogeneous engineering tool landscape. IEC 62714-1 gives an overview about the format.

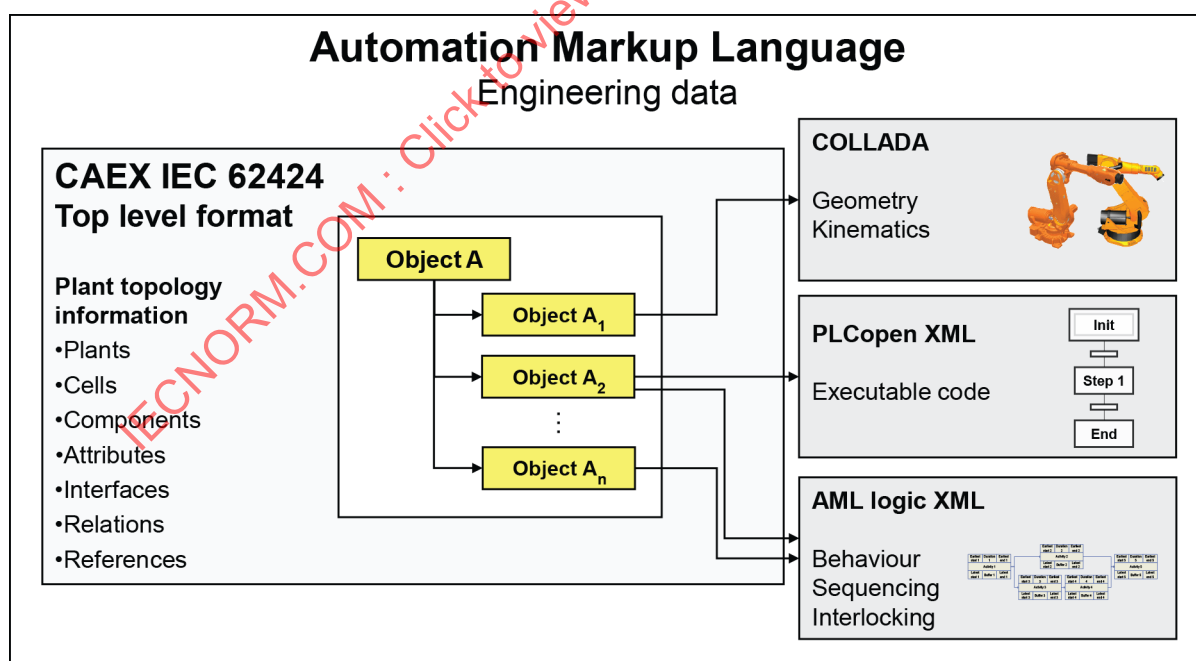
The goal of AML is to interconnect engineering tools from the existing heterogeneous tool landscape in their different disciplines, e.g. mechanical plant engineering, electrical design, process engineering, process control engineering, HMI development, PLC programming, robot programming, etc.

AML stores engineering information following the object-oriented paradigm and allows modelling of physical and logical plant components as data objects encapsulating different aspects. An object may consist of other sub-objects and may itself be part of a larger composition or aggregation. Typical objects in plant automation comprise information on topology, geometry, kinematics, and logic, whereas logic comprises sequencing, behaviour, and control.

AML combines existing industry data formats that are designed for the storage and exchange of different aspects of engineering information. These data formats are used on "as-is" basis within their own specifications and are not branched for AML needs.

The core of AML is the top-level data format CAEX that connects the different data formats. Therefore, AML has an inherent distributed document architecture.

Figure 1 illustrates the basic AML architecture and the distribution of topology, geometry, kinematic, and logic information.



IEC

Figure 1 – Overview of the engineering data exchange format AML

Owing to the different aspects of AML, IEC 62714 consists of different parts focussing on different aspects.

- IEC 62714-1: Architecture and general requirements

This part specifies the general AML architecture, the modelling of engineering data, classes, instances, relations, references, hierarchies, basic AML libraries and extended AML concepts.

- IEC 62714-2: Role class libraries

This part specifies additional AML libraries.

- IEC 62714-3: Geometry and kinematics

This part specifies the modelling of geometry and kinematics information.

- IEC 62714-4: Logic

This part specifies the modelling and referencing of logic information.

Further parts may be added in the future in order to interconnect further data standards to AML.

Clause 5 gives an informative overview of this part of the standard.

Clause 6 gives a normative description of the considered logic models.

Clause 7 gives a normative description of the AML logic XML schema, with which logic models can be stored.

Clause 8 specifies the normative provisions to store the logic models in AML logic XML.

Clause 9 defines how to store meta information about the source tool directly into the AML logic XML document.

Clause 10 defines a logic related role class library and interface class library.

Clause 11 gives a normative description regarding referencing logic information in AML logic XML documents.

Clause 12 gives a normative description regarding referencing interlocking information in AML logic XML documents.

Annex A provides examples for the storage of logic models in AML logic XML.

Annex B describes the referencing methods for logic information.

Annex C describes the referencing methods for interlocking information.

Annex D gives a normative XML representation of the libraries defined in this document.

Annex E gives a normative XML representation of the AML logic XML schema defined in this document.

ENGINEERING DATA EXCHANGE FORMAT FOR USE IN INDUSTRIAL AUTOMATION SYSTEMS ENGINEERING – Automation Markup Language –

Part 4: Logic

1 Scope

This part of IEC 62714 specifies the integration of logic information as part of an AML model for the data exchange in a heterogenous engineering tool landscape of production systems.

This document specifies three types of logic information: sequencing, behaviour, and interlocking information.

This document deals with the six following sequencing and behaviour logic models (covering the different phases of the engineering process of production systems) and how they are integrated in AML: Gantt chart, activity-on-node network, timing diagram, Sequential Function Chart (SFC), Function Block Diagram (FBD), and mathematical expression.

This document specifies how to model Gantt chart, activity-on-node network, and timing diagram and how they are stored in Intermediate Modelling Layer (IML).

NOTE 1 With this, it is possible to transform one logic model into another one. A forward transformation supports the information enrichment process and reduces or avoids a re-entry of information between the exchanging engineering tools.

NOTE 2 Mapping of other logic models, e.g. event-driven logic models like state charts, onto IML is possible.

This document specifies how interlocking information is modelled (as interlocking source and target groups) in AML. The interlocking logic model is stored in Function Block Diagram (FBD).

This document specifies the AML logic XML schema that stores the logic models by using IEC 61131-10.

This document specifies how to reference PLC programs stored in PLCopen XML documents.

This document does not define details of the data exchange procedure or implementation requirements for the import/export tools.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

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]. Available at <http://www.w3.org/TR/2004/REC-xml-20040204>

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]. Available at <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]. Available at <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]. Available at <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]. Available at https://www.plcopen.org/system/files/downloads/tc6_xml_v201_technical_doc.pdf

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62714-1 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at <http://www.electropedia.org/>
- ISO Online browsing platform: available at <http://www.iso.org/obp>

3.1.1

AML logic object

AML object containing logic information

3.1.2

AML logic XML document

external XML document based on AML logic XML schema, which can be referenced from AML logic object

3.1.3

AML logic XML schema

XML schema for storing logic models of AML

¹ First edition. This first edition has been replaced in 2018 by a second edition IEC 62714-1:2018, *Engineering data exchange format for use in industrial automation systems engineering – Automation markup language – Part 1: Architecture and general requirements*.

3.1.4**behaviour information**

information that describes the responses of the controlled system to the sequencing information and to other external interactions

3.1.5**behaviour logic model**

representation of behaviour information

3.1.6**interlocking information**

information that describes the instructions to the controlled system to avoid the system from causing harmful effects on humans and environment

3.1.7**interlocking logic model**

representation of interlocking information

3.1.8**interlocking source group**

group of AML objects that indicate when actions are necessary to provide functional safety

3.1.9**interlocking target group**

group of AML objects that execute actions that are necessary to ensure functional safety

3.1.10**logic model element**

element of a logic model

3.1.11**logic information**

sequencing information, behaviour information, or interlocking information

3.1.12**logic model**

representation of logic information

3.1.13**mathematical expression**

meaningful combination of numbers, variables, and mathematical symbols to represent syntactically correct dependencies among them

Note 1 to entry: Mathematical expression may range from simple terms to complex equation systems.

3.1.14**sequencing information**

information that describes the instructions to the controlled system

3.1.15**sequencing logic model**

representation of sequencing information

3.2 Abbreviated terms

For the purpose of this document, the abbreviated terms of IEC 62714-1 apply as well as those listed in Table 1.

Table 1 – Abbreviated terms

SFC	Sequential Function Chart
FBD	Function Block Diagram
IML	Intermediate Modelling Layer
XSLT	Extensible Stylesheet Language Transformations
MathML	Mathematical Markup Language

4 Conformity

To claim conformity to this part of IEC 62714 with respect to the support of AML, the requirements of Clause 6, 7, 8, 9, 10, 11 and 12 shall be fulfilled. In the scope of AML, an AML logic XML document shall conform to the AML logic XML schema defined in Clause 7 and Annex E.

A PLCopen XML document shall conform to the specification of PLCopen® XML 2.0 or 2.01, or to the specification of IEC 61131-10.

This document does not define provisions of the data exchange procedure or implementation requirements for the import/export tools.

5 Overview

5.1 Logic information in production system engineering

- This document focuses on the storage of logic information - an important aspect for electrical design, HMI development, PLC and robot control programming, for simulation purposes, and virtual commissioning.
- To support different phases in the iterative production system engineering process covering different levels of detail, AML needs to be able to cover logic information from different tools and disciplines. Thus, different roles of logic information belonging to a production system or to single components have to be stored. This variety of information can be classified into two main kinds: sequencing information and behaviour information. Those two roles of information serve different purposes, but those may overlap depending upon the point-of-view and utilization of elements in AML (see Figure 2).
 - Sequencing information is information that describes the instructions to the controlled system (or subsystem). In its most detailed form, it is often represented by a program executed in one or more controllers, e.g. robot controller or PLC. But the development of sequences starts earlier in the engineering process – for modelling on high level of abstraction – with models that represent the required operations of a larger scale unit (cell, line, plant etc.).
 - Behaviour information is information that describes the responses of the controlled system (or subsystem) to the sequencing information and to other external interactions. It is often represented by a given model which reacts on external input, e.g. behaviour of a gripper or valve. Those external inputs would trigger the behaviour or signalling of the gripper's states.
 - Besides sequencing information and behaviour information, interlocking information is a third complementary type of logic information (see Figure 2). It is an important part within the engineering of production systems to ensure their safe behaviour. It affects not only the safe interaction with humans and the environment, but it also helps to avoid unstable states of the systems possibly causing harmful effects on humans and environment or damages on machines and products.

A controlled system (or subsystem) may be represented by more than just one role of above information in AML - e.g. robot (see Figure 2).

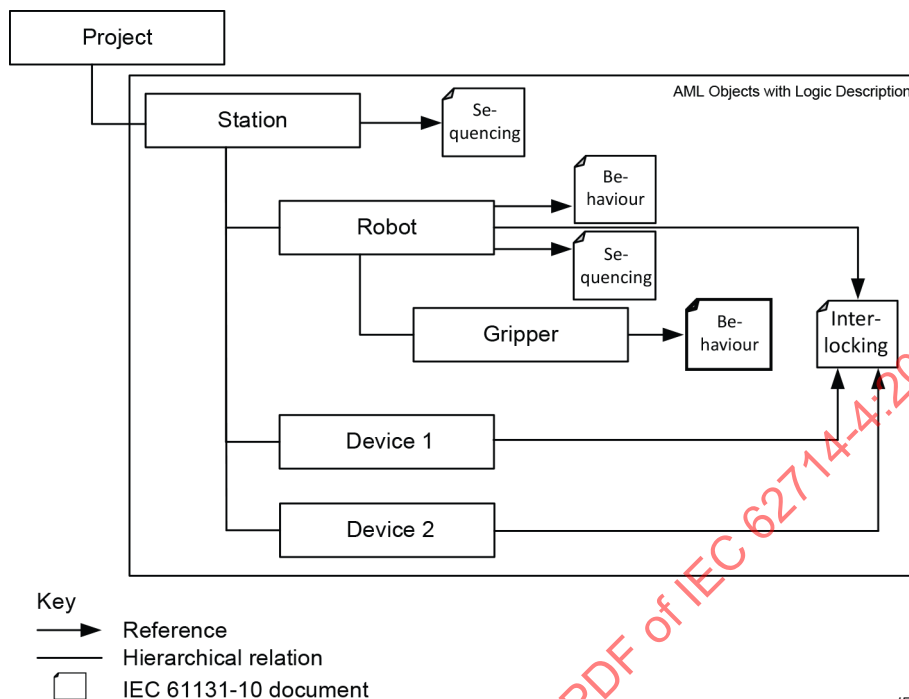


Figure 2 – Example of system representation with roles of information in AML

5.2 Logic models in production system engineering

To be applicable within the engineering process of production systems, AML needs to support typical logic models – storing the logic information – which cover each phase of the engineering process (see Figure 3). These logic models are the following:

- Gantt charts are used mainly within early phases of the engineering process to represent the timing and duration of manufacturing processes on a high level of abstraction (see 6.2).
- Activity-on-node networks are applied mainly within early phases of the engineering process to represent the timing and interdependence of manufacturing processes with the capability to store timing information (see 6.3).
- Timing diagrams are used mainly within the later phases of the engineering process to describe the devices with their states and timing relationships (see 6.4). Real signals are introduced, and it is possible to express sequencing information.
- Sequential Function Charts (SFC) are one of the programming languages of the IEC 61131-3 mainly applied within later phases of production system engineering (see 6.5). Modelling of sequencing information as well as behaviour information is possible.
- Function Block Diagrams (FBD) are one of the programming languages of the IEC 61131-3 mainly applied within later phases of production system engineering (see 6.6). Modelling of interlocking information as well as behaviour information is possible.
- Mathematical expressions are used mainly within the later phases of the engineering process to describe mathematical, physical, chemical, or logical relationships of a device (see 6.7).

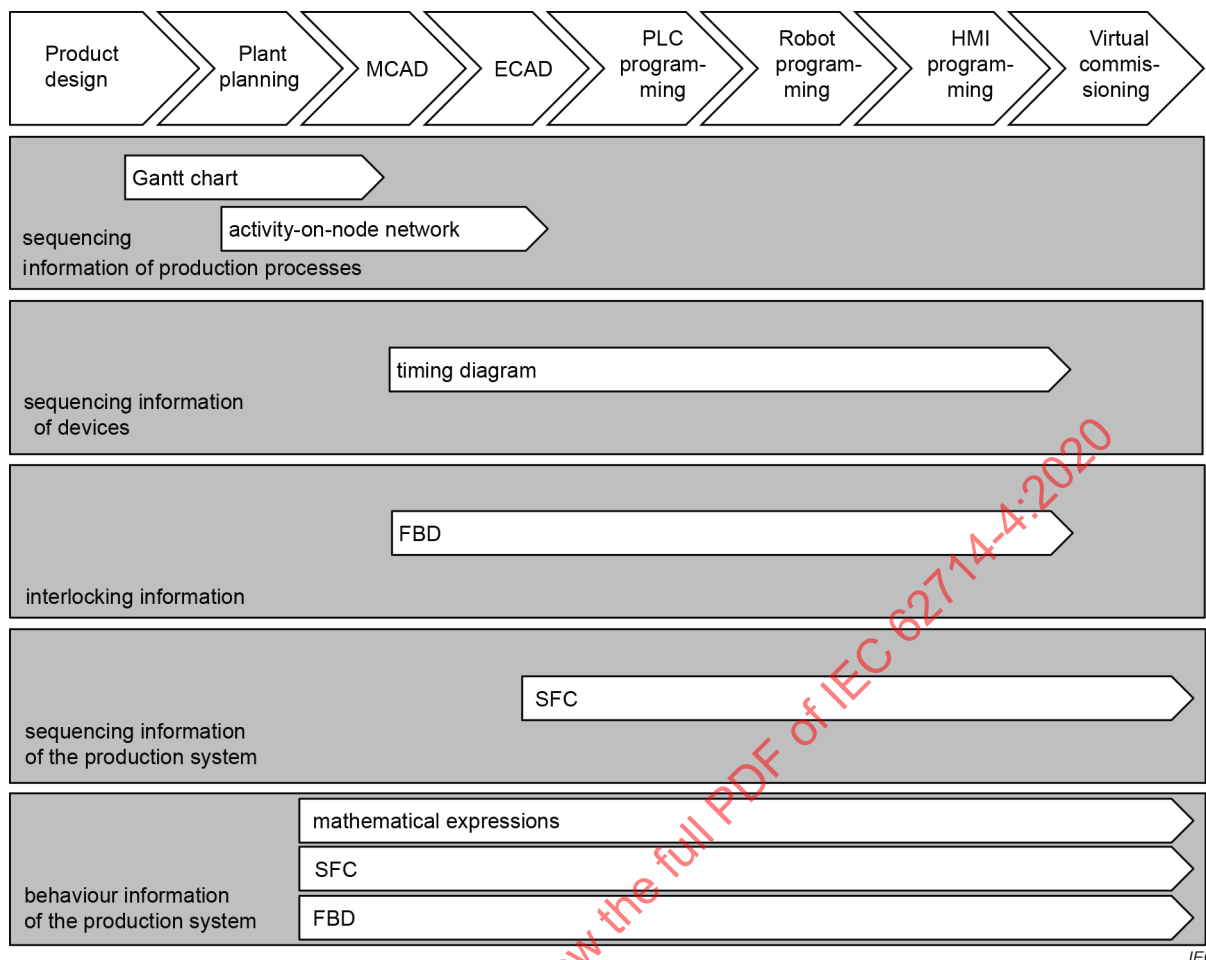


Figure 3 – Logic models in AML

5.3 Storing logic models in AML logic XML

It is intended to enable a transformation between the logic models. This is essential since the engineering process of production systems is a process of information enrichment. In the beginning, the sequencing information of a production system is only roughly described by using a Gantt chart and needs to be detailed in the next step. So, the Gantt chart is transformed into an activity-on-node network: the information from the Gantt chart is taken over and the complex timing conditions can be added now. Theoretically, this workflow can be carried out to the executable program, incrementing information in each step. A re-entry of information within the subsequent engineering tool is, therefore, not necessary. However, since the logic models have different modelling powers, forward and backward transformation between them, will cause information loss in particular cases, e.g. transforming an activity-on-node network to a Gantt chart.

An overview of the scope of this document is given in Figure 4.

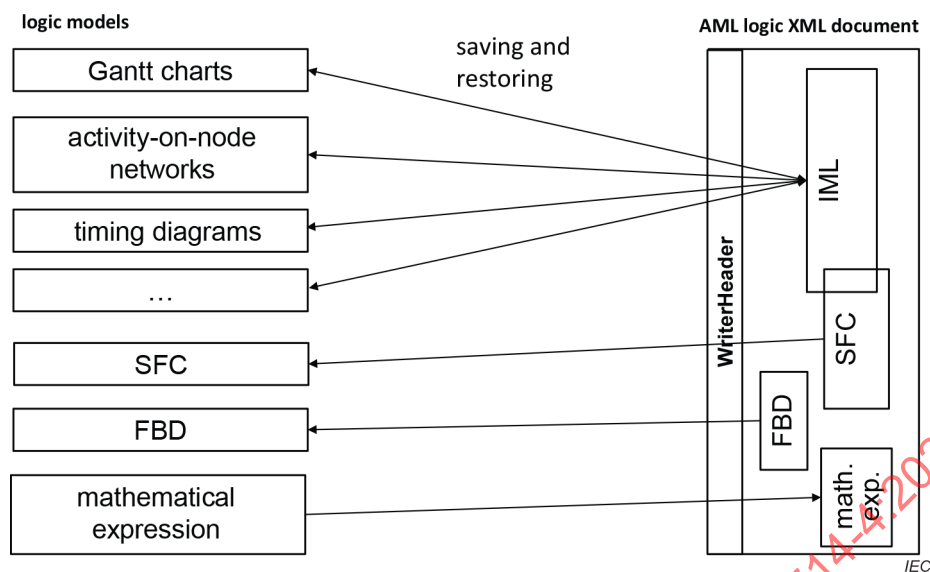


Figure 4 – Storing logic models in AML logic XML

NOTE To convert a SFC in the context of IML into a SFC in the context of IEC 61131-10, some XSLT files can be used that are available to public. When converting from IML SFC into an IEC 61131-10 SFC information might get lost.

AML uses IEC 61131-10 to store the logic models. To decouple IEC 61131-10 from the logic models, this part defines an Intermediate Modelling Layer (IML). Thus, IML forms the common model for logic models, which enhances the forward and backward transformation and the extensibility of AML for new logic models, like event-driven logic models like state charts.

Since this document is not simply about storing PLC programs, it adapts the usage of IEC 61131-10. An AML logic XML schema is used, specified in Clause 7, that allow the storage of the mentioned logic models. The modelling elements of AML logic XML to store the considered logic models are shown in Figure 5. To those modelling elements the logic models can be mapped. This is described in Clause 8. Annex A provides examples.

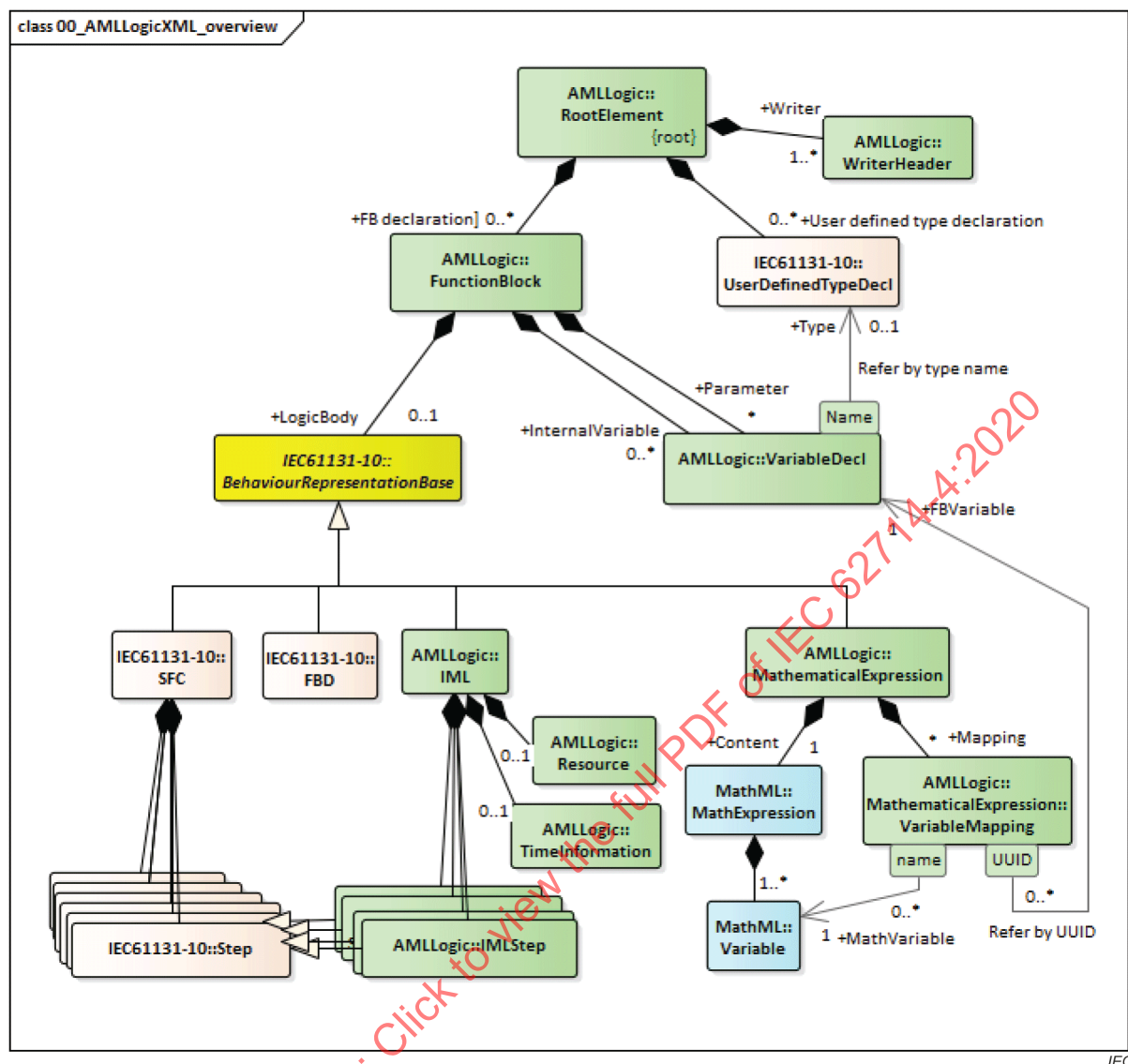


Figure 5 – Modelling elements of the AML logic XML

In addition, to enable a proper data exchange in the scope of AML, meta information on AML logic XML document level are necessary – called WriterHeader - to identify and characterize the document. Normative provisions are defined in Clause 9.

Since the AML logic XML is based on IEC 61131-10, it is also possible to use the other logic models provided by it. In this part of IEC 62714, only SFC (Sequential Function Charts) and FBD (Function Block Diagrams) are considered as relevant for the scope of this document.

Also, it is possible to store a logic model represented as a mathematical expression within the AML logic XML document. Either it is the logic model itself that contains the behaviour description of the system, or it can be used (or called) to provide a mathematical expression to one of the other logic models.

5.4 Referencing logic information

In the multi-document architecture of AML, logic information is stored in separate AML logic XML documents based on the IEC 61131-10 data format. Modelling logic information is, therefore, divided into two parts. First, an object, which has logic information, is modelled in CAEX, the top-level format. Second, an AML logic XML document has to be provided containing this logic information. Finally, a reference between the CAEX object and the AML logic XML

document or an object within needs to be created. Normative provisions for referencing all of the three roles of information are defined in Clause 10 and Clause 11. An informative overview is provided in Annex B.

For referencing interlocking information, additional provisions shall be considered. These are normatively defined in Clause 12. An informative overview is provided in Annex C.

6 Logic models

6.1 General

This clause defines the logic models that are considered in this part of IEC 62714. This comprises the graphical elements and their structuring rules, logic information they represent, and how this information is mapped to the model elements.

NOTE This document does not provide information about the execution logic, i.e. how the model elements are executed.

6.2 Gantt charts

6.2.1 General

Gantt charts are graphical representations of schedules typically used to model partially ordered sets of sequentially and concurrently running activities, which comprise of execution order and execution time of the activities. They do not provide means for modelling alternatives and cycles.

NOTE Gantt charts are also known as bar charts.

6.2.2 Graphical elements

In a Gantt chart, the following graphical elements shall be used (see Figure 6):

- one timeline (horizontal axis) realised as a global clock;
- one vertical axis listing all the names of the corresponding bars;
- zero or more horizontal bars;
- zero or more directed arrows.

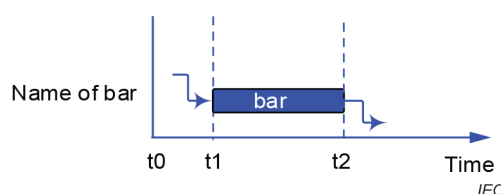


Figure 6 – Model elements of Gantt charts

6.2.3 Chart structure

The following structure for a Gantt chart shall be used:

- Each Gantt chart shall be represented by two orthogonal axes. The horizontal axis represents the time, having a direction, representing linear time. The vertical axis represents the list of bar names
- Each bar shall have its own line in the graphical representation with its corresponding name on the vertical axis.
- The timed start point, and timed end point of a bar shall correspond with the horizontal axis.
- Bars may be connected by arrows to other bars.
- An arrow shall be connected from the right side of one bar to the left side of another bar.

6.2.4 Logic information

Gantt charts shall comprise the following types of logic information:

- Activity information:
 - name;
 - start time;
 - finish time;
 - duration.
- Zero or more number of predecessor relation information between activities.
- Zero or more number of successor relation information between activities.

6.2.5 Logic information within Gantt chart

For the representation of the logic information within a Gantt chart, the following provisions apply (see Figure 7):

- An activity shall be represented by a bar.
 - Activity name shall be represented by the name of a bar listed on the vertical axis.
 - Activity start time shall be represented by the timed start point of a bar listed on the horizontal axis.
 - Activity finish time shall be represented by the timed end point of a bar listed on the horizontal axis.
 - Activity duration shall be represented by the length of a bar.
- A predecessor-successor-relation between activities shall be represented by an arrow.

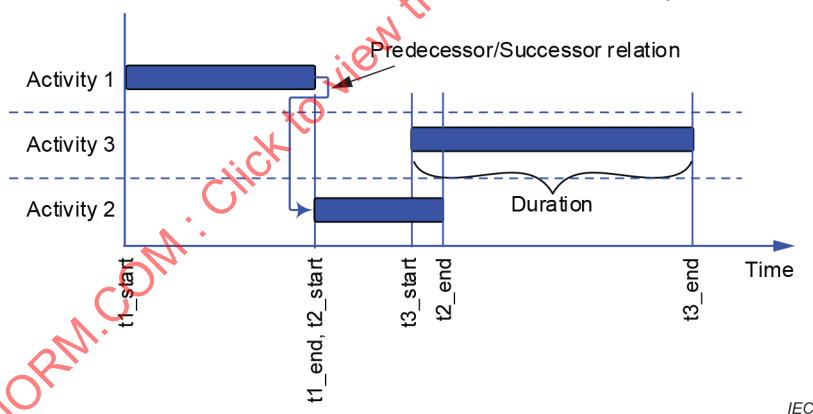


Figure 7 – Information provided by Gantt charts

6.3 Activity-on-node networks

6.3.1 General

Activity-on-node networks are used to describe and analyse temporal and causal relations of a set of interdependent activities. The end-start-relation between the nodes states that a node can only start after the previous node has ended. Activity-on-node networks do not provide means for modelling alternatives and cycles.

6.3.2 Graphical elements

In an activity-on-node network the following graphical elements shall be used (see Figure 8):

- zero or more nodes, represented by rectangles, each of them divided in seven boxes;
- zero or more directed arrows.

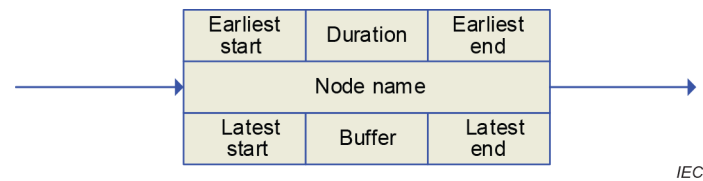


Figure 8 – Model elements of activity-on-node networks

6.3.3 Node structure

The following structure for a node shall be used:

- The earliest start (t_{es}) of a node should be located as a number in the box in the upper left corner of the node.
- The earliest end (t_{ee}) of a node should be located as a number in the box in the upper right corner of the node.
- The duration (t_d) of a node should be located as a number in the box in the upper centre part of the node and is calculated by the earliest start and earliest end of the node using the $t_d = t_{ee} - t_{es}$ formula
- The node name should be located in the box in the centre of the node.
- The latest start (t_{ls}) of a node should be located as a number in the box in the lower left corner of the node.
- The latest end (t_{le}) of a node should be located as a number in the box in the lower right corner of the node.
- The buffer value (t_b) of a node should be located as a number in the box in the lower centre part of the node and is calculated by the earliest end and the latest end of the node using the $t_b = t_{le} - t_{ee}$ formula.

6.3.4 Network structure

The following structure for an activity-on-node network shall be used:

- Nodes may be connected by arrows to other nodes.
- An arrow shall be connected from the right side of one node to the left side of another node.

6.3.5 Logic information

Activity-on-node networks shall comprise the following types of logic information:

- Activity information:
 - name;
 - earliest start time;
 - latest start time;
 - earliest end time;
 - latest end time;
 - duration;
 - buffer.
- Predecessor-successor relation information between activities.

6.3.6 Logic information within activity-on-node networks

For the representation of the logic information within an activity-on-node network, the following provisions apply (see Figure 9):

- An activity shall be represented by a node.
 - Activity name shall be represented by the name of a node.
 - Activity earliest start time shall be represented by the earliest start of a node.
 - Activity latest start time shall be represented by the latest start of a node.
 - Activity earliest end time shall be represented by the earliest end of a node.
 - Activity latest end time shall be represented by the latest end of a node.
 - Activity duration shall be represented by the duration of a node.
 - Activity delay shall be represented by the buffer of a node.
- A predecessor-successor-relation between activities shall be represented by an arrow.

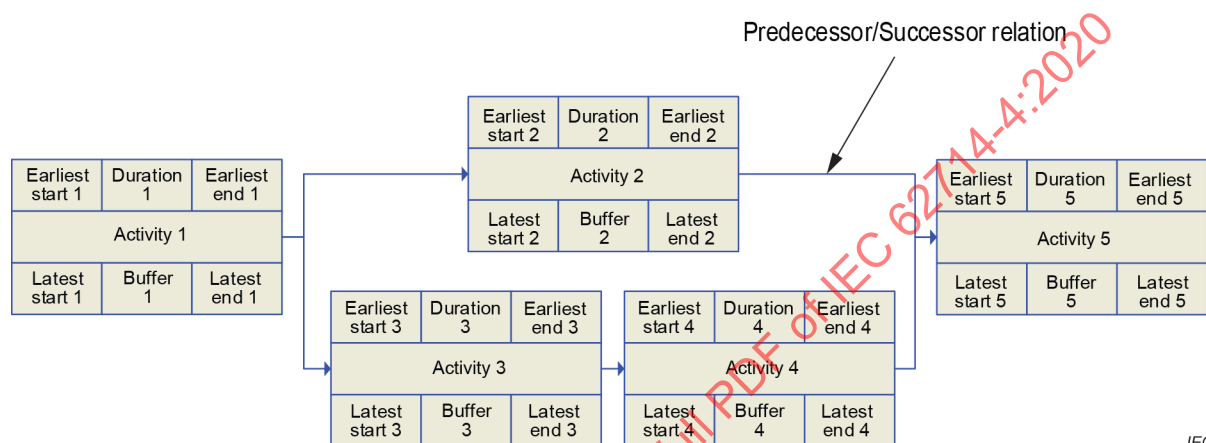


Figure 9 – Information provided by activity-on-node networks

6.4 Timing diagrams

6.4.1 General

Timing diagrams are used to describe the temporal and causal relations between signals and system states. Furthermore, they describe how changes of system states are affected by signals and how signals are generated by states changes. Timing diagrams enable the modelling of alternatives.

6.4.2 Graphical elements

In a timing diagram, the following graphical elements shall be used (see Figure 10):

- one or more lanes represented as columns;
- one or more levels represented as columns;
- zero or more directed and named arrows;
- one timeline (horizontal axis) realized as a global clock;
- one or more lifelines.

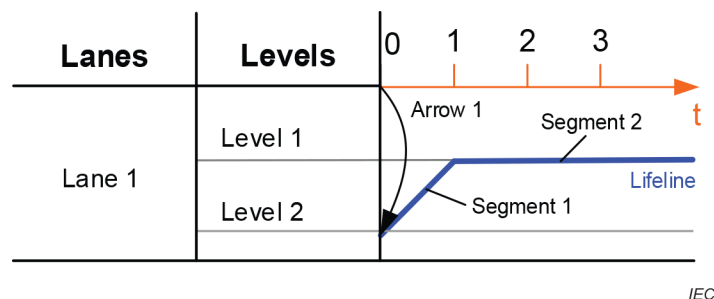


Figure 10 – Model elements of timing diagrams

6.4.3 Diagram structure

The following structure for a timing diagram shall be used:

- Each level shall have its own horizontal stripe. The names of the levels shall be put in the corresponding levels column.
- Levels that belong to one lane shall be grouped in the lanes column. The names of the lanes shall be put in the corresponding lanes column. One lane shall be represented by one or more levels.
- Each lane shall have one lifeline, representing the change of level over linear time.
- A lifeline shall be composed of one or more segments.
- A segment shall be represented by a straight line.
- An arrow shall connect:
 - The timeline with the starting point of one segment,
 - An end point of a segment with the starting point of the other segment in the same lifeline or other lifeline.
- The end point of a segment shall be the starting point of zero or more arrows.
- Zero or more arrows shall start at any point at the timeline.

6.4.4 Logic information

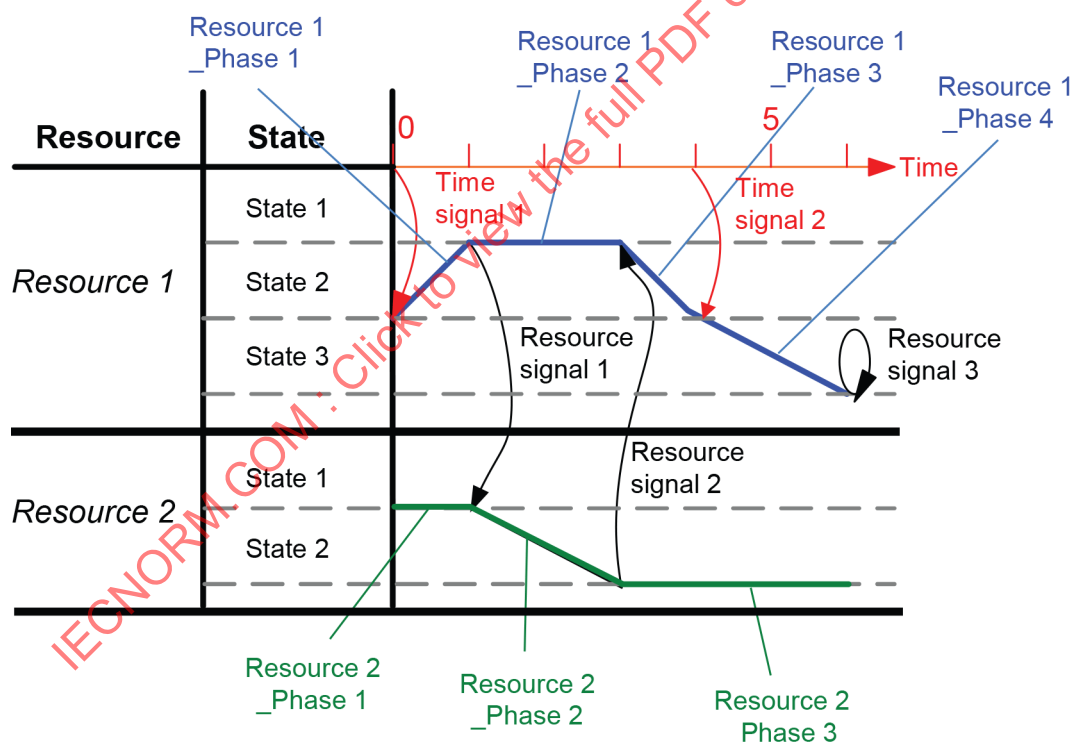
Timing diagrams shall comprise the following types of logic information:

- Resource information:
 - name.
- State information:
 - name;
 - phase (represents the duration);
 - state flow.
- Signal information:
 - name;
 - two different types: time and resource signal;
 - point in time when a signal occurs;
 - influence of a signal on the state of a resource.

6.4.5 Logic information within timing diagrams

For the representation of the logic information within a timing diagram, the following provisions apply (see Figure 11):

- A resource shall be represented by a lane.
 - Resource name shall be represented by the name of the lane.
- A state shall be represented by a level.
 - State name shall be represented by the name of the level.
 - Variation of state in time shall be presented by a lifeline.
 - A phase shall be represented by a segment.
- A signal shall be represented by an arrow.
 - Signal name shall be represented by the name of the arrow.
 - A time signal shall be represented by an arrow that starts at a point on the timeline. A resource signal shall be represented by an arrow that starts at the end of a segment of one lifeline.
 - Point in time when a signal occurs shall correspond with the x-axis.
 - Influence of a signal on the state of a resource shall be represented by a new beginning of a segment of the corresponding lifeline.



IEC

Figure 11 – Information provided by timing diagrams

6.5 Sequential function charts

Sequential Function Charts (SFC) describes causal and temporal dependencies of sequences of system steps and transitions. The modelling of complex sequences with loops and conditional execution is possible.

This document shall use SFCs as specified in IEC 61131-3.

6.6 Function block diagrams

Function Block Diagrams (FBD) describes logical networks of function blocks respectively how input and output variables are connected functionally.

This document shall use FBDs as specified in IEC 61131-3. The usage of function blocks specified in IEC 61131-3 shall be allowed.

6.7 Mathematical expression

Mathematical expressions are an arrangement of numbers, absolute terms, or letters to express mathematical, physical, chemical, or logical relationships. A mathematical expression consists of one formula or a system of equations.

For defining mathematical expressions, this document shall use the Mathematical Markup Language (MathML) version 2.0 content part.

7 AML logic XML schema description

7.1 Schema overview

7.1.1 Use of IEC 61131-10 schema

This clause describes "AML logic XML schema", which is formally defined in Annex E.

AML logic XML schema imports the IEC 61131-10 schema and partially reuses its type definitions to represent logic models using variables, function blocks, sequential function charts, etc.

The IEC 61131-10 schema supports two extension mechanisms allowing incorporation of information in the exchange format. The first mechanism is the "AddData" element, which can be nested below various other elements. Its usage is intended for the extension of other elements, by which they are enriched with information, but which is not crucial for the element's exchange between tools. The second mechanism is using an XML schema technique called "abstract type substitution". A new element type can be derived from the abstract type and used as a substitution for the element type where the abstract type is specified.

AML logic XML schema succeeds both extension mechanisms of the IEC 61131-10 schema and naming conventions for elements naturally. The schema structure overview is shown in Figure 12 in UML class diagram style. Classes whose names start with the prefix "IEC61131_10_Ed1_0" represent types imported from the IEC 61131-10 schema. Also, classes whose names are written in italic style (also drawn in bright yellow) represent abstract types that shall be substituted with one of its extended non-abstract types.

- User-defined data type declaration and its reference from variable declaration are fully reused by AML Logic XML schema as shown in Figure 12.
- Logic descriptions and their base type of IEC 61131-10 are also reused by AML Logic XML schema as shown in Figure 12. Thus, all logic descriptions of IEC 61131-10 are all available in AML Logic XML. In addition, AML logic XML schema adds its specific logic models such as IML and mathematical expression by "abstract type substitution", that is the second extension mechanism succeeded from IEC 61131-10.
- AML Logic XML schema extends fundamental SFC elements of IEC 61131-10 to represent Intermediate Modelling Layer (IML) described in 5.3, as shown in Figure 12.



7.1.2 Schema versioning

Additionally, the root element "AMLLogic" has a "schemaVersion" attribute. On creating an XML document based on the schema, its version number shall be set to the attribute.

7.2 Root element "AMLLLogic"

7.2.1 General

The root element "AMLLLogic" represents a kind of project or library, which holds logic information. Its content is shown in Figure 13.

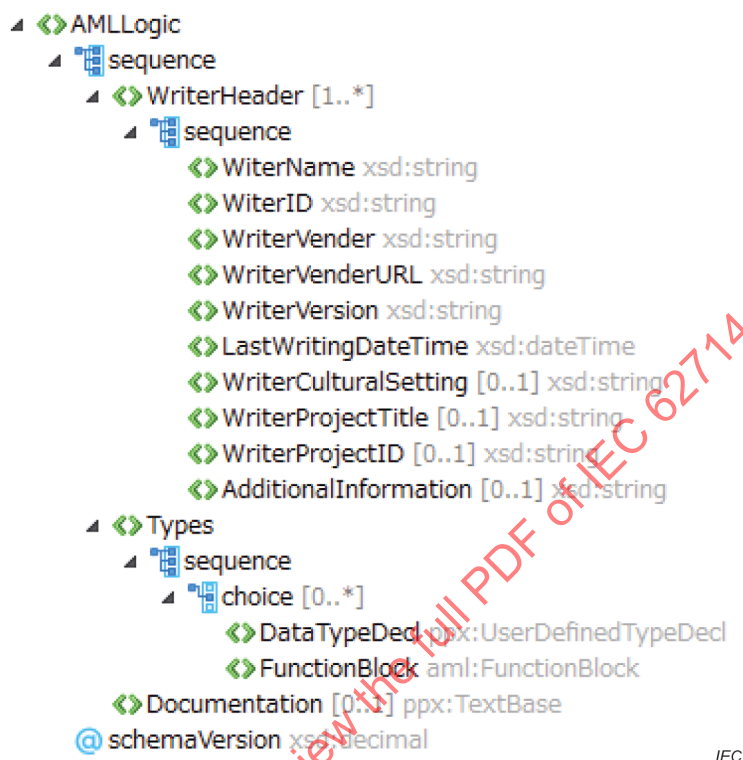


Figure 13 – Root element "AMLLLogic"

7.2.2 Attributes

"schemaVersion" attribute shall hold the version number of AML Logic XML schema that the XML document is compliant to.

7.2.3 Sub-element "WriterHeader"

"WriterHeader" element represents meta information about the tool, which writes the XML document, as explained in Clause 9.

7.2.4 Sub-element "Types"

"Types" element is place holder for user-defined data type declaration and / or function block declaration.

"DataTypeDecl" element of "ppx:UserDefinedTypeDecl" type represents user-defined data type declaration of IEC 61131-3 language specification. Thus, any kind of data type available in IEC 61131-3 language specification can be defined in AML Logic XML as well. IEC 61131-10 document should be referenced for the detailed specification of the type "ppx:UserDefinedTypeDecl".

"FunctionBlock" element of "aml:FunctionBlock" type represents a kind of function block specialized for AML logic description, where only necessary features of IEC 61131-3 function block are supported. Its content is explained in 7.3.

7.2.5 Sub-element "Documentation"

"Documentation" element represents general description for the root "AMLLogic" element. The type "ppx: TextBase" is an abstract type imported from IEC 61131-10, which is substituted with plain text of "ppx: SimpleText" type by default.

7.3 Complex type "FunctionBlock"

7.3.1 General

Complex type "FunctionBlock" (as shown in Figure 14) represents a kind of function block specialized for AML logic description, where only necessary features of IEC 61131-3 function block are supported.

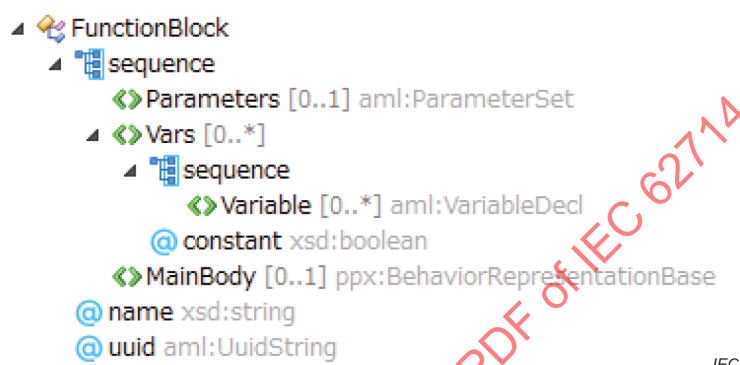


Figure 14 – Complex type "FunctionBlock"

7.3.2 Attributes

"name" attribute of "xsd:string" represents the type name of the function block.

"uuid" attribute of "aml:UuidString" represents universal unique identifier for the type definition of the function block. Simple type "UuidString" is explained in 7.9.

7.3.3 Sub-element "Parameters"

"Parameters" element of "aml:ParameterSet" type represents input / output / in-out variable declaration of the function block. The type "ParameterSet" is explained in 7.3.6 in detail.

7.3.4 Sub-element "Vars"

"Vars" element represents internal variable declarations of the function block.

"Variable" element of "aml:VariableDecl" type represents variable declaration. Complex type "aml:VariableDecl" is explained in 7.3.7.

"constant" attribute represents that variable declarations within the <Vars> element are variables with constant values.

7.3.5 Sub-element "MainBody"

"MainBody" element of "ppx:BehaviorRepresentationBase" type represents a logic model which shall be evaluated when the function block is executed. Any non-abstract complex type derived from "ppx:BehaviorRepresentationBase" can be substitute for the type of the element. Thus all logic models defined in IEC 61131-10 schema are also available in this function block of AML Logic XML, such as Sequential Function Chart (SFC), Function Block Diagram (FBD), Ladder Diagram (LD), and Structured Text (ST).

7.3.6 Complex type "ParameterSet"

Complex type "ParameterSet" (as shown in Figure 15) represents a declaration set of input variables, output variables, and in-out variables.

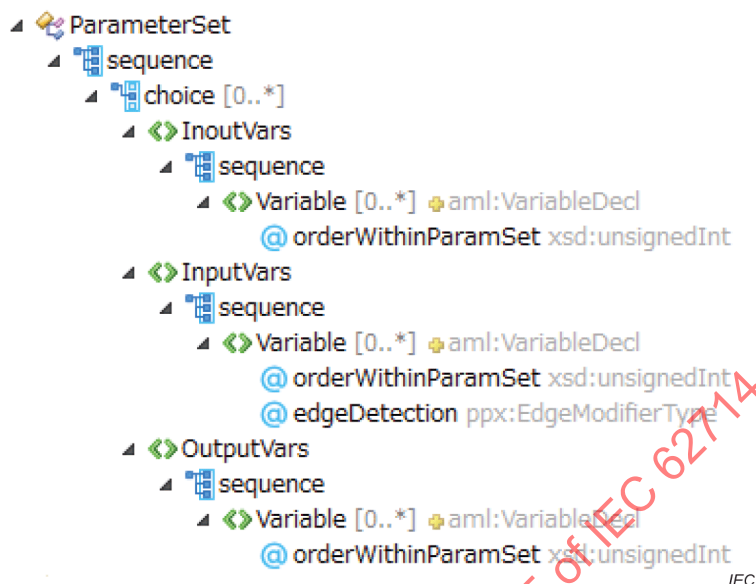


Figure 15 – Complex type "ParameterSet"

"InoutVars" element represents declaration section for In-Out variables. In terms of "Variable" element within the "InoutVars", its type is extended from "aml:VariableDecl" which is explained in 7.3.7, and added "orderWithinParamSet" attribute. The reason of this extension is because the order of parameters of a function block is significant but appearance order of elements of the same element type is not ensured to be significant in an XML document.

"InputVars" element represents declaration section for input variables of the function block. In terms of "Variable" element within the "InputVars", its type is extended from "aml:VariableDecl" explained in 7.3.7 as well, and added "edgeDetection" attribute in addition to the "orderWithinParamSet" attribute.

"OutputVars" element represents declaration section for output variables of the function block. In terms of "Variable" element within the "OutputVars", its type is extended from "aml:VariableDecl" explained in 7.3.7 as well, and added the "orderWithinParamSet" attribute as well.

7.3.7 Complex type "VariableDecl"

Complex type "VariableDecl" (as shown in Figure 16) represents variable declaration section.

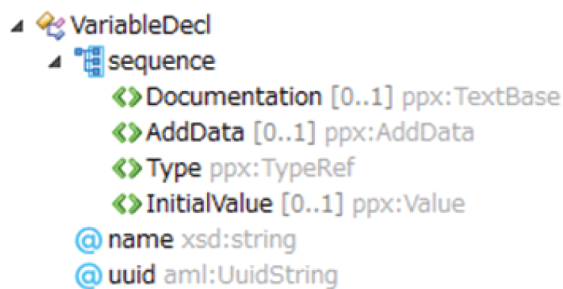


Figure 16 – Complex type "VariableDecl"

"Documentation" element of "ppx:TextBase" type represents the description of the variable.

"AddData" element of "ppx:AddData" type can be used to store any type of supplemental data.

"Type" element of "ppx:TypeRef" type represents reference to the type definition of the variable. It can be type name or instant declaration such as "ARRAY [0..8] OF INT". As for the details of the complex type "ppx:TypeRef", IEC 61131-10 should be referenced.

"InitialValue" element of "ppx:Value" type represents initial value for the variable. It can be single value, array value for array type, or structured value for structured data type depending on the data type of the variable. As for details of complex type "ppx:Value", IEC 61131-10 specification should be referenced.

"name" attribute of "xsd:string" type represents name of the variable.

"uuid" attribute of "aml:UuidString" represents universal unique identifier for the variable. Simple type "UuidString" is explained in 7.9.

7.4 Complex Type "IML"

7.4.1 General

Complex type "IML" (as shown in Figure 17) represents Intermediate Modelling Layer for certain multiple logic models, as described in 5.3. This complex type can be used as a substitute for the type of "MainBody" element of function block in AML Logic XML since it extends "ppx:BehaviourRepresentationBase", as described in Figure 12.



IEC

Figure 17 – Complex type "IML"

7.4.2 Attributes

"logicModelType" attribute of "aml:LogicModelTypeEnum" type represents what kind of logic model is represented with the IML. Simple type "LogicModelTypeEnum" is explained in 7.6.

7.4.3 Sub-element "Resource"

"Resource" element represents additional logic information necessary for timing diagram as described in 6.4.4.

- "uuid" attribute represents Universal Unique Identifier for the resource element. Simple type "UuidString" is explained in 7.9.
- "name" attribute represents the name of the resource.
- "order" attribute represents the order of the resource within the IML. Smaller numbers should be listed first.

"Status" elements represent possible status of the "Resource" element.

- "uuid" attribute represents Universal Unique Identifier for the status element. Simple type "UuidString" is explained in 7.9.
- "name" attribute represents the name of the status.
- "order" attribute represents the order of the status within the resource. Smaller numbers should be listed first.

7.4.4 Sub-element "TimeInformation"

"TimeInformation" represents logical time scale, especially for Gantt chart or timing diagram.

- "timeFormat" attribute of "aml:TimeFormatEnum" type represents format of the time expression within this IML. Simple type "aml:TimeFormatEnum" is explained in 7.8.
- "startOfScale" attribute of "aml:TimeUnion" type represents starting point of the time scale in the format specified by "timeFormat" attribute above. Simple type "aml:TimeUnion" is explained in 7.7.
- "endOfScale" attribute of "aml:TimeUnion" type represents ending point of the time scale in the format specified by "timeFormat" attribute above. Simple type "aml:TimeUnion" is explained in 7.7.
- "scaleInterval" attribute of "aml:TimeUnion" type represents scale interval of the time scale in the format specified by "timeFormat" attribute above. Simple type "aml:TimeUnion" is explained in 7.7.

7.4.5 Choice of "IML****" element

As the "choice" element, either of "IMLStep", "IMLTransition", "IMLSimultaneousDivergence", or "SimultaneousConvergence" can be selected. This schema structure is basically the same as "SFC" complex type of IEC 61131-10. However, alternatives for the choice are limited to what is necessary for IML representation. Also, the type of each alternative element is extended from original one of IEC 61131-10.

7.4.6 Complex type "IMLStep"

Complex type "IMLStep" (as shown in Figure 18) represents a phase or state of an activity. It extends "ppx:Step" of IEC 61131-10, which represents a STEP in the Sequential Function Chart.



Figure 18 – Complex type "IMLStep"

The following attributes are added as extension to the base type "ppx:Step" of IEC 61131-10.

- "uuid" attribute of "aml:UuidString" represents Universal Unique Identifier for the IMLStep element. Simple type "UuidString" is explained in 7.9.
- "startTime" attribute of "aml:TimeUnion" represents start time of the phase of the activity which corresponds to this IMLStep. Simple type "aml:TimeUnion" is explained in 7.7. The time shall be represented in the format specified by the "timeFormat" attribute of the "TimeInformation" element explained in 7.4.4.
- "endTime" attribute of "xsd:duration" represents end time of the phase of the activity which corresponds to this IMLStep. Simple type "aml:TimeUnion" is explained in 7.7. The time shall be represented in the format specified by the "timeFormat" attribute of the "TimeInformation" element explained in 7.4.4.
- "buffer" attribute of "xsd:duration" represents a margin for the time span between "startTime" and "endTime".
- "refStartStatusId" attribute of "aml:UuidString" represents a reference to the source status of the segment of the pseudo waveform, which the "IMLStep" element represents in the case that the IML represents timing diagram. It concretely represents the "uuid" attribute's value of certain "Status" element within certain "Resource" element within the "IML" element.
- "refEndStatusId" attribute of "aml:UuidString" represents a reference to the destination status of the segment of the pseudo waveform which the "IMLStep" elements represents in the case that the IML represents timing diagram. It concretely represents the "uuid" attribute's value of certain "Status" element within certain "Resource" element within the "IML" element.

7.4.7 Complex type "IMLTransition"

Complex type "IMLTransition" (shown in Figure 19) represents a kind of predecessor/successor relationship between "IMLStep" elements. It extends "ppx:Transition" of IEC 61131-10, which represents a TRANSITION in Sequential Function Chart.

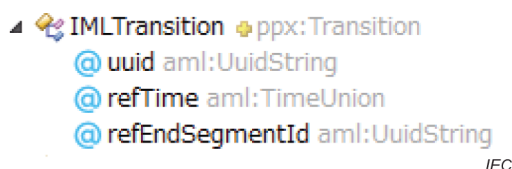


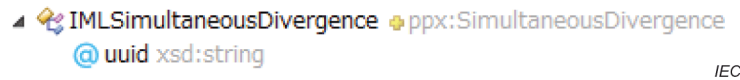
Figure 19 – Complex type "IMLTransition"

The following attributes are added as extension to the base type "ppx:Transition" of IEC 61131-10.

- "uuid" attribute of "aml:UuidString" represents Universal Unique Identifier for the IMLTransition element. Simple type "UuidString" is explained in 7.9.
- "refTime" attribute represents time when this transition should be triggered. Simple type "aml:TimeUnion" is explained in 7.7. The time shall be represented in the format specified by the "timeFormat" attribute of the "TimeInformation" element explained in 7.4.4.
- "refEndSegmentId" attribute of "aml:uuisString" type represents reference to the IMLStep element by UUID whose end should trigger this transition.

7.4.8 Complex type "IMLSimultaneousDivergence"

Complex type "IMLSimultaneousDivergence" extends "ppx:SimultaneousDivergence" of IEC 61131-10 (shown in Figure 20), which represents a starting point of parallel branch in Sequential Function Chart.



IEC

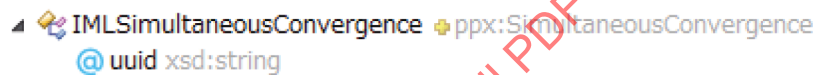
Figure 20 – Complex type "IMLSimultaneousDivergence"

The following attributes are added as extension to the base type "ppx:SimultaneousDivergence" of IEC 61131-10.

- "uuid" attribute of "aml:UuidString" represents Universal Unique Identifier for the IMLSimultaneousDivergence element. Simple type "UuidString" is explained in 7.9.

7.4.9 Complex type "IMLSimultaneousConvergence"

Complex type "IMLSimultaneousConvergence" extends "ppx:SimultaneousConvergence" of IEC 61131-10 (shown in Figure 21), which represents an ending point of parallel branch in Sequential Function Chart.



IEC

Figure 21 – Complex type "IMLSimultaneousConvergence"

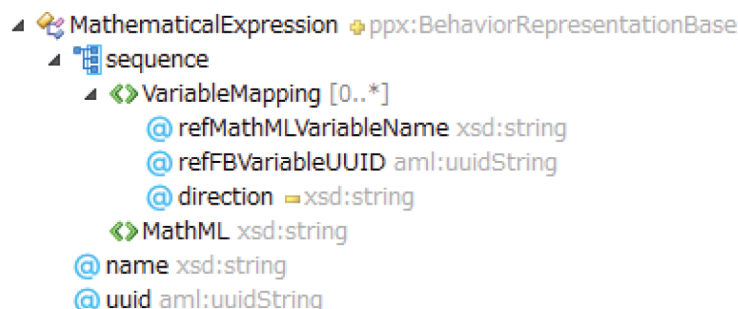
The following attributes are added as extension to the base type "ppx:SimultaneousConvergence" of IEC 61131-10.

- "uuid" attribute of "aml:UuidString" represents Universal Unique Identifier for the IMLSimultaneousConvergence element. Simple type "UuidString" is explained in 7.9.

7.5 Complex Type "MathematicalExpression"

7.5.1 General

Complex type "MathematicalExpression" (as shown in Figure 22) represents logic description in mathematical formula expression as explained in 6.7. It can be used as substitution for the "MainBody" element of function block in AML logic XML, since it extends the abstract complex type "ppx:BehaviorRepresentationBase", as described in Figure 12.



IEC

Figure 22 – Complex type "MathematicalExpression"

7.5.2 Attributes

"name" attribute of string type represents the name of the mathematical expression.

"uuid" attribute represents Universal Unique Identifier for the mathematical expression. Simple type "UuidString" is explained in 7.9.

7.5.3 Sub-element "VariableMapping"

"VariableMapping" element represents mapping between mathematical variable written in MathML and variable declared in the function block which this logic model belongs to.

"refMathMLVariableName" represents the name of the variable defined in MathML.

"refFBVariableUUID" represents UUID of the variable declared in the same function block. Simple type "UuidString" is explained in 7.9.

"direction" attribute represents mapping direction with enumerated string which only allows "In" or "Out".

7.5.4 Sub-element "MathML"

"MathML" element of string type represents the XML text data of MathML. It is stored just as plane text. Parsing with MathML schemata is out of scope of AML Logic XML schema.

7.6 Simple type "LogicModelTypeEnum"

Simple type "LogicModelTypeEnum" is restricted string type which only allows one of the enumerated strings, which are "GanttChart", "ActivityOnNodeNetwork", "TimingDiagram", or "Unknown". It is used to specify what kind of logic model type the IML represents.

7.7 Simple type "TimeUnion"

Simple type "TimeUnion" is a union whose members are composed of following time-related simple types built-in XML schema. XML schema specification shall be referenced for detailed value format specification of each built-in type.

- xsd:duration
 - The value format shall be "PnYnMnDTnHnMnS", for instance "P10DT15H" (10 days and 15 hours), "PT1M20S" (1 minute and 20 seconds), "PT95.4" (95,4 seconds), etc.
- xsd:time
 - The value format shall be "hh:mm:ss", for instance simply "15:45:30", or "15:45:30+02:00" to represent time zone offset from UTC.
- xsd:dateTime
 - The value format shall be "YYYY-MM-DDThh:mm:ss", for instance simply "2017-05-30T09:00:00", or "2017-05-30T09:00:00+02:00" to represent time zone offset from UTC.
- xsd:date
 - The value format shall be "YYYY-MM-DD", for instance "2017-09-24".

7.8 Simple type "TimeFormatEnum"

Simple type "TimeFormatEnum" is restricted string type which only allows one of the enumerated strings, which are "xsd:duration", "xsd:time", "xsd:dateTime", or "xsd:date". It is used to specify which time format shall be used for the "aml:TimeUnion" type explained in 7.7.

7.9 Simple type "UuidString"

Simple type "UuidString" is restricted string type to represent UUID in text form. No syntax restriction is actually defined.

It is recommended to implement the UUID as a GUID (as specified in RFC 4122).

8 Storing logic models

8.1 General

This clause defines rules for the storage of logic models to IML. A textual representation in the IML is presented for mapping of each logic model element. For examples, see Annex A.

8.2 Storing Gantt charts in AML logic XML

8.2.1 Common rules

For storing a Gantt chart in AML logic XML, the following provisions apply:

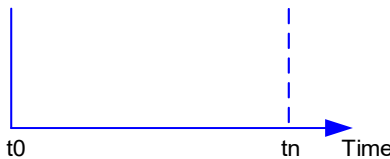
- One Gantt chart shall be mapped to one IML logic element of type "GanttChart" using the following provisions.
- The IML shall start with the "TimeInformation" element, defining the timeline of the Gantt chart, see 8.2.2.
- Each Gantt bar shall be represented by an IMLStep, see 8.2.3.
- Each Gantt arrow shall be represented by an IMLTransition, see 8.2.4.
- For predecessor and successor relations in a Gantt chart the following provisions apply:
 - If a Gantt bar has two or more successor bars, it shall be connected to them by an IMLTransition and IMLSimultaneousDivergence elements, see 8.2.5.
 - If a Gantt bar has two or more predecessors, they shall be connected to them by an IMLSimultaneousConvergence and IMLTransition elements, see 8.2.6.
 - If there are no explicit relations between Gantt bars, i.e. they are concurrent, the order of associated IMLStep shall be parallel and they shall have no ordering relation based on step-step transition sequences, see 8.2.5 and 8.2.6.

If predecessor/successor relations between Gantt bars are defined, the resulting structure shall be a sequence of IMLStep and IMLTransition elements, see 8.2.5 and 8.2.6.

8.2.2 Storing the start of a Gantt chart

Table 2 specifies the mapping of properties and relations of the start of a Gantt chart to IML elements.

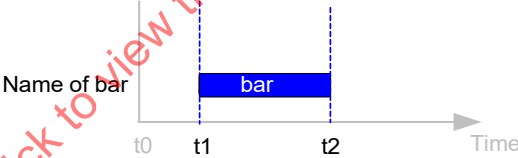
Table 2 – Mapping the start of a Gantt chart to IML element

Type	Representation
Gantt chart element	
IML element	<pre> <xsd:element name="TimeInformation" minOccurs="0"> <xsd:complexType> <xsd:attribute name="timeFormat" type="aml:TimeFormatEnum" use="required" /> <!-- This shall be value of type simple type "TimeFormatEnum" --> <xsd:attribute name="startOfScale" type="aml:TimeUnion" use="optional" default="PT0S"/> <!-- value of t0 in format defined by "timeFormat" attribute --> <xsd:attribute name="endOfScale" type="aml:TimeUnion" use="optional"/> <!-- value of tn in format defined by "timeFormat" attribute --> <xsd:attribute name="scaleInterval" type="aml:TimeUnion" use="optional"/> <!-- value of the time scale in the format defined by xsd:duration type--> </xsd:complexType> </xsd:element> </pre>

8.2.3 Storing bars

Table 3 specifies the mapping of Gantt chart bars to IML elements.

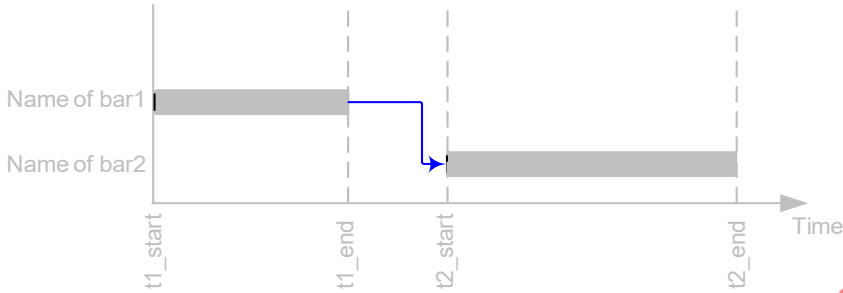
Table 3 – Mapping of Gantt chart bars to IML elements

Type	Representation
Gantt chart element	
IML element	<pre> <xsd:complexType name="IMLStep"> <xsd:complexContent> <xsd:extension base="ppx:Step"> <xsd:attribute name="uuid" type="aml:UuidString"/> <!-- This shall be some textual representation of UUID, which can be GUID. --> <xsd:attribute name="name" type="xsd:string" use="required"/> <!-- "Name of bar" designation in textual representation of type xsd:string --> <xsd:attribute name="startTime" type="aml:TimeUnion"/> <!-- value of t1 in format of type "aml:TimeUnion" --> <xsd:attribute name="endTime" type="aml:TimeUnion"/> <!-- value of t2 in format of type "aml:TimeUnion" --> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre>

8.2.4 Storing arrows

Table 4 specifies the mapping of Gantt chart bar with one or more successor activities to IML elements.

Table 4 – Mapping of Gantt chart arrow to IML elements

Type	Representation
Gantt chart element	
IML element	<pre> <xsd:complexType name="IMLTransition"> <xsd:complexContent> <xsd:extension base="ppx:Transition"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- This shall be some textual representation of UUID, which can be GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- Global ID of the outward connection point of the "Name of bar1" element --> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- Global ID of the outward connection point of the "Name of bar2" element --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> </xsd:complexType> </pre>

8.2.5 Storing successor bars

Table 5 specifies the mapping of Gantt chart bar with one or more successor bar to IML elements.

Table 5 – Mapping of Gantt chart bar with one or more successor bars to IML elements

Type	Representation
Gantt chart element	
IML element	<pre> <xsd:complexType name="IMLTransition"> <xsd:complexContent> <xsd:extension base="ppx:Transition"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- This shall be some textual representation of UUID, which can be GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- Global ID of the outward connection point of the "Name of bar1" element --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- Global ID of the outward connection point of "IMLTransition" element --> </xsd:complexType> </xsd:element> </xsd:complexType> <xsd:complexType name="IMLSimultaneousDivergence"> <xsd:complexContent> <xsd:extension base="ppx:IMLSimultaneousDivergence"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- This shall be some textual representation of UUID, which can be GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- Global ID of the outward connection point of "IMLTransition" element --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- Global ID of the inward connection point of "Name of bar2" element --> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- Global ID of the inward connection point of "Name of bar3" element --> </xsd:complexType> </xsd:element> </xsd:complexType> </pre>

8.2.6 Storing predecessor bars

Table 6 specifies the mapping of Gantt chart bar with one or more predecessor bars to IML elements.

Table 6 – Mapping of Gantt chart bar with one and more predecessor bars to IML elements

Type	Representation
Gantt chart element	
IML element	<pre> <xsd:complexType name="IMLTransition"> <xsd:complexContent> <xsd:extension base="ppx:Transition"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- This shall be some textual representation of UUID, which can be GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- Global ID of the inward connection point of the "IMLTransition" element --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- Global ID of the outward connection point of "Name of bar3" --> </xsd:complexType> </xsd:element> </xsd:complexType> <xsd:complexType name="IMLSimultaneousConvergence"> <xsd:complexContent> <xsd:extension base="ppx:SimultaneousConvergence"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- This shall be some textual representation of UUID, which can be GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- Global ID of the outward connection point of "Name of bar1" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- Global ID of the outward connection point of "Name of bar2" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> </xsd:complexType> </pre>

Type	Representation
	<pre> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- Global ID of the inward connection point of "IMLTransition" element --> </xsd:complexType> </xsd:element> </xsd:complexType> </pre>

8.3 Storing activity-on-node networks in AML logic XML

8.3.1 Common rules

For storing an activity-on-node network in AML logic XML, the following provisions apply:

- One activity-on-node network shall be mapped to one IML logic element of type "ActivityOnNodeNetwork" using the following provisions.
- The entire activity-on-node network shall be mapped to one IML using the following provisions.
- The IML shall start with the "TimeInformation" element, defining the time format of the activity-on-node network, see 8.3.2.
- Each activity-on-node network node shall be represented by an IMLStep, see 8.3.3.
- Each activity-on-node arrow shall be represented by an IMLTransition, see 8.3.4.
- For predecessor and successor relations in an activity-on-node network the following provisions apply:
 - If an activity-on-node network node has two or more successor nodes, it shall be connected to them by an IMLTransition and IMLSimultaneousDivergence elements, see 8.3.5.
 - If an activity-on-node network node has two or more predecessors, they shall be connected to them by an IMLSimultaneousConvergence and IMLTransition elements, see 8.3.6.
 - If there are no explicit relations between activity-on-node network nodes, i.e. they are concurrent, the order of associated IMLStep shall be parallel and they shall have no ordering relation based on step-step transition sequences, see 8.3.5 and 8.3.6.
 - If predecessor/successor relations between activity-on-node network nodes are defined, the resulting structure shall be a sequence of IMLStep and IMLTransition elements, see 8.3.5 and 8.3.6.

8.3.2 Storing the start of an activity-on-node network

Table 7 specifies the mapping of properties and relations of the start of an activity-on-node network to IML elements.

Table 7 – Mapping of the start of an activity-on-node network to IML elements

Type	Representation
Activity-on-node network element	No graphical representation available.
IML element	<pre> <xsd:element name="TimeInformation" minOccurs="0"> <xsd:complexType> <xsd:attribute name="timeFormat" type="aml:TimeFormatEnum" use="required" /> <!-- This shall be value of type simple type "TimeFormatEnum" --> </xsd:complexType> </xsd:element> </pre>

8.3.3 Storing nodes

Table 8 specifies the mapping of activity-on-node network nodes to IML elements.

Table 8 – Mapping of activity-on-node network nodes to IML elements

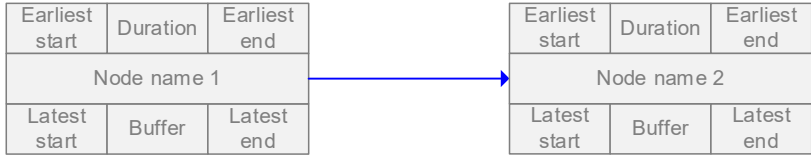
Type	Representation									
Activity-on-node network element	<table><tr><td>Earliest start</td><td>Duration</td><td>Earliest end</td></tr><tr><td colspan="3">Node name</td></tr><tr><td>Latest start</td><td>Buffer</td><td>Latest end</td></tr></table>	Earliest start	Duration	Earliest end	Node name			Latest start	Buffer	Latest end
Earliest start	Duration	Earliest end								
Node name										
Latest start	Buffer	Latest end								
IML element	<pre><xsd:complexType name="IMLStep"> <xsd:complexContent> <xsd:extension base="ppx:Step"> <xsd:attribute name="uuid" type="aml:UuidString"/> <!-- This shall be some textual representation of UUID, which can be GUID. --> <xsd:attribute name="name" type="xsd:string" use="required"/> <!-- "Name of bar" designation in textual representation of type xsd:string --> <xsd:attribute name="startTime" type="aml:TimeUnion"/> <!-- value of earliest start time in format of type "aml:TimeUnion" --> Having the earliest start time and the earliest end time, the duration can be calculated.--> <xsd:attribute name="endTime" type="aml:TimeUnion"/> <!-- value of earliest end time in format of type "aml:TimeUnion" --> Having the earliest start time and the earliest end time, the duration can be calculated.--> <xsd:attribute name="buffer" type="aml:duration" use="optional"/> <!-- value of buffer in format of type "aml:duration" --> Having the earliest start time, earliest end time, and the buffer, the latest start time and the latest end time can be calculated.--> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre>									

8.3.4 Storing arrows

Table 9 specifies the mapping of activity-on-node network arrows to IML elements.

NOTE Both nodes are not mapped to IML elements, only the arrow, which connects both.

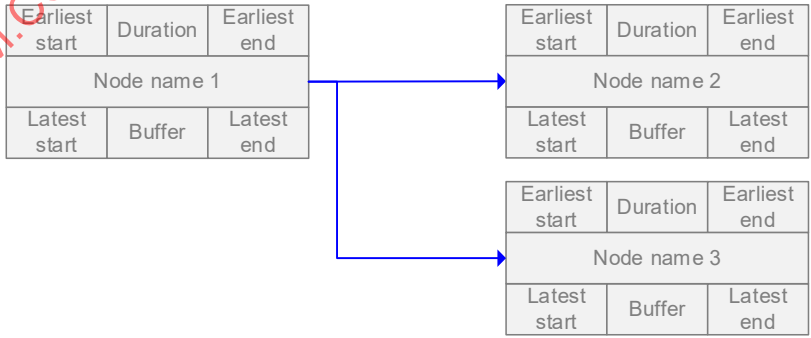
Table 9 – Mapping of activity-on-node network arrows to IML elements

Type	Representation
Activity-on-node network element	
IML element	<pre> <xsd:complexType name="IMLTransition"> <xsd:complexContent> <xsd:extension base="ppx:Transition"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- This shall be some textual representation of UUID, which can be GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- Global ID of the outward connection point of the "Name of bar1" element --> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- Global ID of the outward connection point of the "Name of bar2" element --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> </xsd:complexType> </pre>

8.3.5 Storing successor nodes

Table 10 specifies the mapping of activity-on-node network successor activities to IML elements.

Table 10 – Mapping of activity-on-node network successor nodes to IML elements

Type	Representation
Activity-on-node network element	
IML element	<pre> <xsd:complexType name="IMLTransition"> <xsd:complexContent> <xsd:extension base="ppx:Transition"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- This shall be some textual representation of UUID, which can be GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> </xsd:complexType> </pre>

Type	Representation
	<pre> <!--Global ID of the outward connection point of the "Name of bar1" element --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--Global ID of the outward connection point of "IMLTransition" element --> </xsd:complexType> </xsd:element> </xsd:complexType> <xsd:complexType name="IMLSimultaneousDivergence"> <xsd:complexContent> <xsd:extension base="ppx:IMLSimultaneousDivergence"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- This shall be some textual representation of UUID, which can be GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--Global ID of the outward connection point of "IMLTransition" element --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--Global ID of the inward connection point of "Name of bar2" element --> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--Global ID of the inward connection point of "Name of bar3" element --> </xsd:complexType> </xsd:element> </xsd:complexType> </pre>

8.3.6 Storing predecessor nodes

Table 11 specifies the mapping of activity-on-node network predecessor nodes to IML elements.

Table 11 – Mapping of activity-on-node network predecessor nodes to IML elements

Type	Representation
Activity-on-node network element	
IML element	<pre> <xsd:complexType name="IMLTransition"> <xsd:complexContent> <xsd:extension base="ppx:Transition"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre>

Type	Representation
	<pre> <!-- This shall be some textual representation of UUID, which can be GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--Global ID of the inward connection point of the "IMLTransition" element --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--Global ID of the inward connection point of "Name of bar3" --> </xsd:complexType> </xsd:element> </xsd:complexType> <xsd:complexType name="IMLSimultaneousConvergence"> <xsd:complexContent> <xsd:extension base="ppx:SimultaneousConvergence"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- This shall be some textual representation of UUID, which can be GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--Global ID of the outward connection point of "Name of bar1" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--Global ID of the outward connection point of "Name of bar2" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--Global ID of the inward connection point of "IMLTransition" element --> </xsd:complexType> </xsd:element> </xsd:complexType> </pre>

8.4 Storing timing diagrams in AML logic XML

8.4.1 Common rules

For storing a timing diagram in AML logic XML, the following provisions apply:

- One timing diagram shall be mapped to one IML logic element of type "TimingDiagram" using the following provisions.
- The IML shall start with the timeline definition, see 8.4.2.
- The Resources and their States within a timing diagram shall be represented through IML elements, see 8.4.3.

- The lifeline for each Resource shall be defined through its segments as IMLStep elements coupled with IMLTransition elements, see 8.4.4.
- The signals in a timing diagram shall be defined through IMLTransition elements, see 8.4.5.

NOTE Signals can be fired by the timeline at any time or by a resource after remaining within a resource state with a predefined duration.

8.4.2 Storing the timeline of a timing diagram

Table 12 specifies the mapping of the timeline of a timing diagram to IML elements.

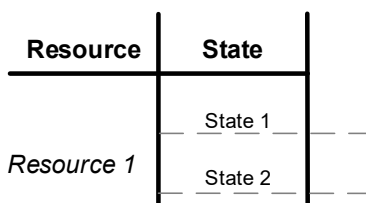
Table 12 – Mapping of the timeline of a timing diagram to IML elements

Type	Representation
Timing diagram element	
IML element	<pre> <xsd:element name="TimeInformation" minOccurs="0"> <xsd:complexType> <xsd:attribute name="timeFormat" type="aml:TimeFormatEnum" use="required" /> <!-- This shall be value of type simple type "TimeFormatEnum" --> <xsd:attribute name="startOfScale" type="aml:TimeUnion" use="optional" default="PT0S"/> <!-- value of t0 in format defined by "timeFormat" attribute --> <xsd:attribute name="endOfScale" type="aml:TimeUnion" use="optional"/> <!-- value of tn in format defined by "timeFormat" attribute --> <xsd:attribute name="scaleInterval" type="aml:TimeUnion" use="optional"/> <!-- value of the time scale in the format defined by xsd:duration type --> </xsd:complexType> </xsd:element> </pre>

8.4.3 Storing resources and resource states

Table 13 specifies the mapping of timing diagram resources and resource states to IML elements.

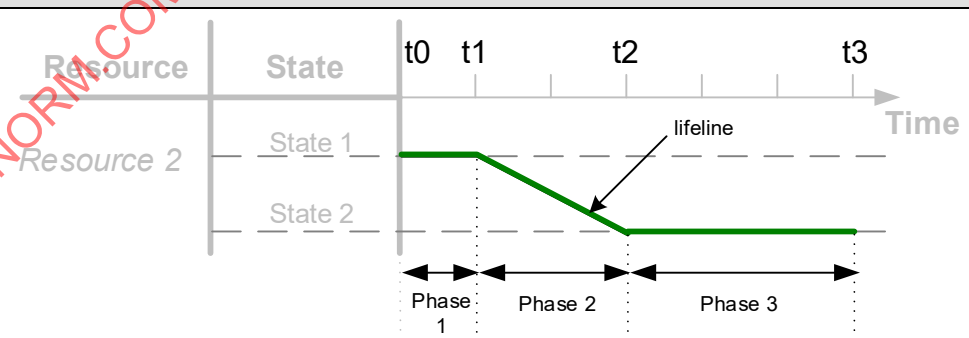
Table 13 – Mapping of timing diagram resources and resource states to IML elements

Type	Representation
Timing diagram element	
IML element	<pre> <xsd:element name="Resource" minOccurs="0" maxOccurs="unbounded"> <xsd:complexType> <xsd:sequence> <xsd:element name="Status" maxOccurs="unbounded"> <xsd:complexType> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- This shall be some textual representation of UUID, which can be GUID. --> <xsd:attribute name="name" type="xsd:string" use="required"/> <!-- Name of state in textual representation of type xsd:string --> <xsd:attribute name="order" type="xsd:integer" use="optional"/> <!-- Order of state in xsd:integer representation --> </xsd:complexType> </xsd:element> </xsd:sequence> </xsd:complexType> </xsd:element> </pre>

8.4.4 Storing lifelines

Table 14 specifies the mapping of timing diagram lifelines to IML elements.

Table 14 – Mapping of timing diagram lifelines to IML elements

Type	Representation
Timing diagram element	
IML element	<pre> <xsd:complexType name="IMLStep"> <xsd:complexContent> <xsd:extension base="ppx:Step"> <xsd:attribute name="uuid" type="aml:UuidString"/> <!-- This shall be some textual representation of UUID, which can be GUID. --> <xsd:attribute name="name" type="xsd:string" use="required"/> <!-- "Phase 1" segment in textual representation of type xsd:string --> <xsd:attribute name="startTime" type="aml:TimeUnion"/> <!-- value of t0 in format of type "aml:TimeUnion" --> <xsd:attribute name="endTime" type="aml:TimeUnion"/> <!-- value of t1 in format of type "aml:TimeUnion" --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"/> </xsd:complexType> </pre>

Type	Representation
	<pre> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- Global ID of the outward connection point of "Phase 1" --> </xsd:complexType> </xsd:element> </xsd:complexType> <xsd:complexType name="IMLStep"> <xsd:complexContent> <xsd:extension base="ppx:Step"> <xsd:attribute name="uuid" type="aml:UuidString"/> <!-- This shall be some textual representation of UUID, which can be GUID. --> <xsd:attribute name="name" type="xsd:string" use="required"/> <!-- "Phase 2" segment in textual representation of type xsd:string --> <xsd:attribute name="startTime" type="aml:TimeUnion"/> <!-- value of t1 in format of type "aml:TimeUnion" --> <xsd:attribute name="endTime" type="aml:TimeUnion"/> <!-- value of t2 in format of type "aml:TimeUnion" --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- Global ID of the inward connection point of "Phase 2" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- Global ID of the outward connection point of "Phase 2" --> </xsd:complexType> </xsd:element> </xsd:complexType> <xsd:complexType name="IMLStep"> <xsd:complexContent> <xsd:extension base="ppx:Step"> <xsd:attribute name="uuid" type="aml:UuidString"/> <!-- This shall be some textual representation of UUID, which can be GUID. --> <xsd:attribute name="name" type="xsd:string" use="required"/> <!-- "Phase 3" segment in textual representation of type xsd:string --> <xsd:attribute name="startTime" type="aml:TimeUnion"/> <!-- value of t2 in format of type "aml:TimeUnion" --> <xsd:attribute name="endTime" type="aml:TimeUnion"/> <!-- value of t3 in format of type "aml:TimeUnion" --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- Global ID of the inward connection point of "Phase 3" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> </xsd:complexType> <xsd:complexType name="IMLTransition"> <xsd:complexContent> <xsd:extension base="ppx:Transition"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- This shall be some textual representation of UUID, which can be GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- Global ID of the outward connection point of "Phase 1" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> </pre>

Type	Representation
	<pre> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- Global ID of the inward connection point of "Phase 2" --> </xsd:complexType> </xsd:element> </xsd:complexType> <xsd:complexType name="IMLTransition"> <xsd:complexContent> <xsd:extension base="ppx:Transition"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- This shall be some textual representation of UUID, which can be GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- Global ID of the outward connection point of "Phase 2" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- Global ID of the inward connection point of "Phase 3" --> </xsd:complexType> </xsd:element> </xsd:complexType> </pre>

8.4.5 Storing the time signal and the resource signal

Table 15 specifies the mapping of the time signal and the resource signal to IML elements.

Table 15 – Mapping of the time signal and the resource signal to IML elements

Type	Representation
Timing diagram element	The diagram illustrates the mapping of time and resource signals to IML elements. It shows two resources, Resource 1 and Resource 2, each with multiple states. The time axis is marked with t0, t1, t2, and t3. The diagram is divided into three phases: Phase 1, Phase 2, and Phase 3. A 'Time signal' is shown as a step function, and a 'Resource signal' is shown as a line graph. The time axis is marked with t0, t1, t2, and t3.

Type	Representation
IML element	<pre> <xsd:complexType name="IMLTransition"> <xsd:complexContent> <xsd:extension base="ppx:Transition"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- This shall be some textual representation of UUID, which can be GUID. --> <xsd:attribute name="refTime" type="aml:TimeUnion" use="optional"/> <!-- value of t2 in format of type "aml:TimeUnion " --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--Global ID of the outward connection point of "Phase 2 Resource 1" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--Global ID of the inward connection point of "Phase 3 Resource 1" --> </xsd:complexType> </xsd:element> </xsd:complexType> <xsd:complexType name="IMLTransition"> <xsd:complexContent> <xsd:extension base="ppx:Transition"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- This shall be some textual representation of UUID, which can be GUID. --> <xsd:attribute name="refEndSegmentId" type="aml:UuidString" use="optional"/> <!-- This shall be some textual representation of UUID, which can be GUID, of the IMLStep for "Phase 1 Resource 1" segment--> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--Global ID of the outward connection point of "Phase 1 Resource 2" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--Global ID of the inward connection point of "Phase 2 Resource 2" --> </xsd:complexType> </xsd:element> </xsd:complexType> </pre>

8.5 Storing sequential function charts in AML logic XML

8.5.1 Common rules

For storing a sequential function chart in AML logic XML, the following provisions apply:

- The entire sequential function chart shall be mapped to one "FunctionBlock" element of the type "SFC" within an "AMLLLogic" element.
- The elements of a sequential function chart shall be used in accordance with IEC 61131-10.

8.5.2 Storing variables

Table 16 specifies the storage of variables of a sequential function chart in AML logic XML.

Table 16 – Storing variable of a sequential function chart in AML logic XML

Type	Representation
Sequential function chart element	No graphical representation available.
AML logic XML element	<pre> <xsd:element name="Parameters" type="ppx:ParameterSet" minOccurs="0" maxOccurs="1"/> <xsd:complexType name="ParameterSet"> <xsd:sequence> <xsd:choice minOccurs="0" maxOccurs="unbounded"> <xsd:element name="InputVars"> <xsd:complexType> <xsd:sequence> <xsd:element name="Variable" minOccurs="0" maxOccurs="unbounded"> <xsd:complexType> <xsd:complexContent> <xsd:extension base="ppx:VariableDecl"> <xsd:attribute name="orderWithinParamSet" type="xsd:unsignedInt" use="required"/> <!-- This shall indicate the order of the parameters. --> <xsd:attribute name="edgeDetection" type="ppx:EdgeModifierType" use="optional" default="none"/> <!--This parameter can be characterized regarding edge detection. --> </xsd:extension> </xsd:complexContent> </xsd:complexType> </xsd:element> </xsd:sequence> </xsd:complexType> </xsd:element> </xsd:choice> </xsd:sequence> <xsd:element name="OutputVars"> <xsd:complexType> <xsd:sequence> <xsd:element name="Variable" minOccurs="0" maxOccurs="unbounded"> <xsd:complexType> <xsd:complexContent> <xsd:extension base="ppx:VariableDecl"> <xsd:attribute name="orderWithinParamSet" type="xsd:unsignedInt" use="required"/> <!-- This shall indicate the order of the parameters. --> </xsd:extension> </xsd:complexContent> </xsd:complexType> </xsd:element> </xsd:sequence> </xsd:complexType> </xsd:element> </xsd:choice> </xsd:sequence> </xsd:complexType> </pre>

8.6 Storing function block diagrams in AML logic XML

8.6.1 Common rules

For storing a function block diagram in AML logic XML, the following provisions apply:

- The entire function block diagram shall be mapped to one "FunctionBlock" element of the type "FBD" within an "AMLLogic" element.
- The elements of a function block diagram shall be used in accordance with IEC 61131-10.

8.6.2 Storing variables

Table 17 specifies the storage of variables of a function block diagram in AML logic XML.

Table 17 – Storing variable of a function block diagram in AML logic XML

Type	Representation
Function block diagram element	No graphical representation available.
AML logic XML element	<pre> <xsd:element name="Parameters" type="ppx:ParameterSet" minOccurs="0" maxOccurs="1"/> <xsd:complexType name="ParameterSet"> <xsd:sequence> <xsd:choice minOccurs="0" maxOccurs="unbounded"> <xsd:element name="InputVars"> <xsd:complexType> <xsd:sequence> <xsd:element name="Variable" minOccurs="0" maxOccurs="unbounded"> <xsd:complexType> <xsd:complexContent> <xsd:extension base="ppx:VariableDecl"> <xsd:attribute name="orderWithinParamSet" type="xsd:unsignedInt" use="required"/> <!-- This shall indicate the order of the parameters. --> <xsd:attribute name="edgeDetection" type="ppx:EdgeModifierType" use="optional" default="none"/> <!--This parameter can be characterized regarding edge detection. --> </xsd:extension> </xsd:complexContent> </xsd:complexType> </xsd:element> </xsd:sequence> </xsd:complexType> </xsd:element> <xsd:element name="OutputVars"> <xsd:complexType> <xsd:sequence> <xsd:element name="Variable" minOccurs="0" maxOccurs="unbounded"> <xsd:complexType> <xsd:complexContent> <xsd:extension base="ppx:VariableDecl"> <xsd:attribute name="orderWithinParamSet" type="xsd:unsignedInt" use="required"/> <!-- This shall indicate the order of the parameters. --> </xsd:extension> </xsd:complexContent> </xsd:complexType> </xsd:element> </xsd:sequence> </xsd:complexType> </xsd:element> </xsd:choice> </xsd:sequence> </xsd:complexType> </pre>

8.7 Storing mathematical expressions in AML logic XML

8.7.1 Common rules

For storing a mathematical expression AML logic XML, the following provisions apply:

- The entire mathematical expression shall be mapped to one "FunctionBlock" element of the type "MathematicalExpression" within an "AMLLogic" element.
- Variables of the "FunctionBlock" element shall be mapped to the elements "InputVars" or "OutputVars" within the "Parameters" element accordingly.
- The mathematical expression shall be stored in the "MathML" element in accordance with MathML 2.0 content part.
- How the variables of the "MathML" element correlate with the variables of the "FunctionBlock" element shall be modelled in the "VariableMapping" element.
 - The "VariableMapping" element shall have the following attributes:
 - The attribute "refMathMLVariableName" shall store the name of the variable in the "MathML" element.

- The attribute "refFBVariableUUID" shall store the UUID of the variable of the "FunctionBlock" element.
- The attribute "direction" shall indicate whether it is an input variable or an output variable for the mathematical expression stored in the "MathML" element. The value "In" shall indicate an input variable. The value "Out" shall indicate an output variable.

8.7.2 Storing variables

Table 18 specifies the storage of variables of a mathematical expression in AML logic XML.

Table 18 – Storing variable of a mathematical expression in AML logic XML

Type	Representation
Mathematical expression element	No graphical representation available.
AML logic XML element	<pre> <xsd:element name="Parameters" type="ppx:ParameterSet" minOccurs="0" maxOccurs="1"/> <xsd:complexType name="ParameterSet"> <xsd:sequence> <xsd:choice minOccurs="0" maxOccurs="unbounded"> <xsd:element name="InputVars"> <xsd:complexType> <xsd:sequence> <xsd:element name="Variable" minOccurs="0" maxOccurs="unbounded"> <xsd:complexType> <xsd:complexContent> <xsd:extension base="ppx:VariableDecl"> <xsd:attribute name="orderWithinParamSet" type="xsd:unsignedInt" use="required"/> <!-- This shall indicate the order of the parameters. --> <xsd:attribute name="edgeDetection" type="ppx:EdgeModifierType" use="optional" default="none"/> <!--This parameter can be characterized regarding edge detection. --> </xsd:extension> </xsd:complexContent> </xsd:complexType> </xsd:element> </xsd:sequence> </xsd:complexType> </xsd:element> </xsd:choice> <xsd:element name="OutputVars"> <xsd:complexType> <xsd:sequence> <xsd:element name="Variable" minOccurs="0" maxOccurs="unbounded"> <xsd:complexType> <xsd:complexContent> <xsd:extension base="ppx:VariableDecl"> <xsd:attribute name="orderWithinParamSet" type="xsd:unsignedInt" use="required"/&; <!-- This shall indicate the order of the parameters. --> </xsd:extension> </xsd:complexContent> </xsd:complexType> </xsd:element> </xsd:sequence> </xsd:complexType> </xsd:element> </xsd:sequence> </xsd:complexType> </pre>

8.7.3 Storing variable mappings

Table 19 specifies the storage of variable mappings of a mathematical expression in AML logic XML.

Table 19 – Storing variable mappings of a mathematical expression in AML logic XML

Type	Representation
Mathematical expression element	No graphical representation available.
AML logic XML element	<pre> <xsd:element name="VariableMapping" minOccurs="0" maxOccurs="unbounded"> <xsd:complexType> <xsd:attribute name="refMathMLVariableName" type="xsd:string" use="required"/> <!--It shall represent the name of the variable defined in MathML.--> <xsd:attribute name="refFBVariableUUID" type="aml:UuidString" use="required"/> <!--It shall represent UUID of the variable declared in the same function block.--> <xsd:attribute name="direction" use="required"> <xsd:simpleType> <xsd:restriction base="xsd:string"> <xsd:enumeration value="In"/> <xsd:enumeration value="Out"/> </xsd:restriction> </xsd:simpleType> </xsd:attribute> <!--It shall represent mapping direction with enumerated string which only allows "In" or "Out". --> </xsd:complexType> </xsd:element> </pre>

8.7.4 Storing mathematical expressions

Table 20 specifies the storage of a mathematical expression in AML logic XML.

Table 20 – Storing a mathematical expression in AML logic XML

Type	Representation
Mathematical expression element	Mathematical expression
AML logic XML element	<pre> <xsd:element name="MathML"> <xsd:complexType> <xsd:sequence> <xsd:any namespace="##any" processContents="lax"/> </xsd:sequence> </xsd:complexType> </xsd:element> </pre>

9 Meta information about AML logic XML writer tools

In order to simplify the data exchange between a source tool and a destination tool, it is useful to store information about the writer tools directly into the AML logic XML document. Hence, the following provisions apply:

NOTE In the engineering process it is useful to transfer the information of the project the data belongs to and the time of the export of the data. This supports the implementation process of interfaces.

- An AML logic XML document shall provide information about each writer tool that has written the AML logic XML document.
- In a data exchange tool chain, the last participating writer tools shall store this information in the AML logic XML document.
- Each AML logic XML document shall have one or more XML elements "WriterHeader".
- In a data exchange tool chain, all participating tools shall store this information in the AML logic XML document in the same way. Hence, the document may contain information about multiple tools of a data exchange tool chain. A tool may remove the writer information of other tools. This may hinder the iterative data exchange with the other tools: hence the removal of writer information of other tools is not recommended.

- This information shall be stored within XML elements of the XML element "WriterHeader".
- The meta information shall provide information about:
 - name of the writer tool;
 - unique ID of the writer tool (e.g. UUID, URL of the tool);
 - name of the writer tool vendor;
 - URL of the writer tool vendor;
 - complete version information of the writer tool (including major, minor and build revision); and
 - last writing date and time of the AML logic XML document.
- Additionally, the following optional information may be included:
 - cultural settings of the writer tool following RFC 5646;
 - name of the project associated with exported data;
 - unique ID of the project associated with exported data; and
 - additional information.
- The above information shall be stored by means of the XML elements shown in Table 21 for each writer tool.

Table 21 – Meta information about each AML logic XML writer tool

XML element name	Type	Cardinality	Example
WriterName	xs:string	1	"ToolX"
WriterID	xs:string	1	"4F1D5716-1C2A-4CBC-8C40-91A91F7E09A4"
WriterVendor	xs:string	1	"ToolX Vendor"
WriterVendorURL	xs:string	1	"http://www.ToolX-Vendor.org"
WriterVersion	xs:string	1	"0.2.123 prealpha"
LastWritingDateTime	xs:dateTime	1	"2011-05-25T09:30:47"
WriterCulturalSetting	xs:string	0..1	"en-US"
WriterProjectTitle	xs:string	0..1	"ProjectY"
WriterProjectID	xs:string	0..1	"6BA300EA-DF77-4E78-80F9-2B050E2C6799"
AdditionalInformation	xs:string	0..1	"This tool vendor was previously called ToolYVendor."

10 Extensions of AML classes for logic

10.1 General

This clause defines extensions of the AML role classes and AML interface classes. These classes are either a part of own libraries or are specializations of AML standard classes defined in IEC 62714-1. All described attributes and interfaces are part of these libraries and may be removed in the InstanceHierarchy if not needed.

10.2 AutomationMLLogicRoleClassLib

10.2.1 General

Figure 23 presents the normative AutomationMLLogicRoleClassLib as object tree. AutomationMLLogicRoleClassLib specifies all logic related role classes. Details to each role class are given in 10.2.2 to 10.2.4. The XML text of AutomationMLLogicRoleClassLib is given in Clause D.2.

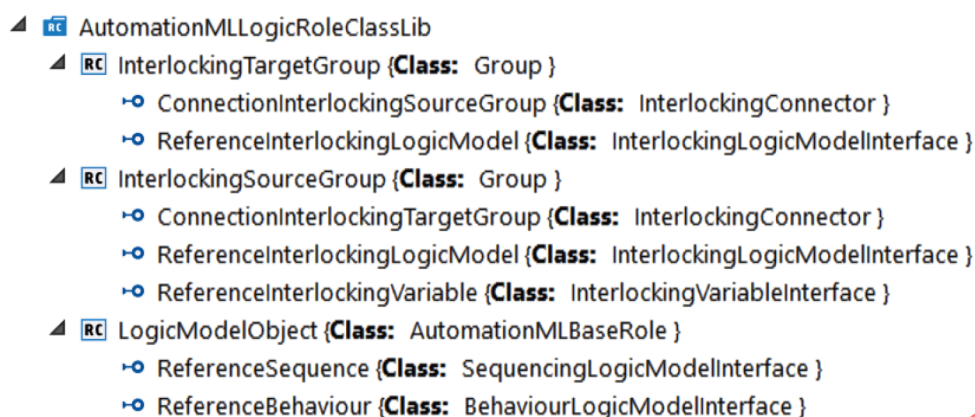


Figure 23 – AutomationMLLogicRoleClassLib

10.2.2 RoleClass InterlockingTargetGroup

The role class "InterlockingTargetGroup" shall be used as specified in Table 22.

Table 22 – RoleClass InterlockingTargetGroup

Class name	InterlockingTargetGroup	
Description	The role class "InterlockingTargetGroup" shall be used for interlocking target groups. Details are specified in Clause 12. For examples see Annex C.	
Parent class	AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Group	
Path for element reference	AutomationMLLogicRoleClassLib/InterlockingTargetGroup	
Attributes	none	
Interfaces	Interface of type="InterlockingConnector"	The interfaces shall be of type "InterlockingConnector" specified in 10.5.2. This interface shall be required for the interlocking concept specified in Clause 12. The name is user-defined. The use of the interface shall be mandatory.
	Interface of type="InterlockingLogicModelInterface"	The interfaces shall be of type "InterlockingLogicModelInterface" specified in 10.3.5. This interface shall be used for the interlocking concept specified in Clause 12 for referencing the interlocking logic model. The name is user-defined. The use of the interface shall be optional.

10.2.3 RoleClass InterlockingSourceGroup

The role class "InterlockingSourceGroup" shall be used as specified in Table 23.

Table 23 – RoleClass InterlockingSourceGroup

Class name	InterlockingSourceGroup	
Description	The role class "InterlockingSourceGroup" shall be used for interlocking source groups. Details are specified in Clause 12. For examples see Annex C.	
Parent class	AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Group	
Path for element reference	AutomationMLLogicRoleClassLib/InterlockingSourceGroup	
Attributes	none	
Interfaces	Interface of type="InterlockingConnector"	The interfaces shall be of type "InterlockingConnector" specified in 10.5.2. This interface shall be required for the interlocking concept specified in Clause 12. The name is user-defined. The use of the interface shall be mandatory.
	Interface of type="InterlockingLogicModelInterface"	The interfaces shall be of type "InterlockingLogicModelInterface" specified in 10.3.5. This interface shall be used for the interlocking concept specified in Clause 12 for referencing the interlocking logic model. The name is user-defined. The use of the interface shall be optional.
	Interface of type="InterlockingVariableInterface"	The interfaces shall be of type "InterlockingVariableInterface" specified in 10.3.8. This interface shall be used for the interlocking concept specified in Clause 12 for referencing the output variable in a logic model containing interlocking information. The name is user-defined. The use of the interface shall be optional.

10.2.4 RoleClass LogicModelObject

The role class "LogicModelObject" shall be used as specified in Table 24.

Table 24 – RoleClass LogicModelObject

Class name	LogicModelObject	
Description	The role class "LogicModelObject" shall denote an AML logic object, instantiating a logic model.	
Parent class	AutomationMLBaseRoleClassLib/AutomationMLBaseRole	
Path for element reference	AutomationMLLogicRoleClassLib/LogicModelObject	
Attributes	none	
Interfaces	Interface of type="SequencingLogicModelInterface"	The interfaces shall be of type "SequencingLogicModelInterface" specified in 10.3.3. This interface shall be used to specify the logic model semantics specified in Clause 11. The name is user-defined. The use of the interface shall be optional.
	Interface of type="BehaviourLogicModelInterface"	The interfaces shall be of type "BehaviourLogicModelInterface" specified in 10.3.4. This interface shall be used to specify the logic model semantics specified in Clause 11. The name is user-defined. The use of the interface shall be optional.

NOTE These objects can be enriched and further referenced in CAEX.

10.3 AutomationMLLogicInterfaceClassLib

10.3.1 General

Figure 24 presents the normative AutomationMLLogicInterfaceClassLib as object tree. AutomationMLLogicInterfaceClassLib specifies all logic related interface classes for referencing AML logic XML documents. Details to each interface class are given in 10.3.2 to 10.3.8. The XML text of AutomationMLInterfaceClassLib is given in Clause D.3.

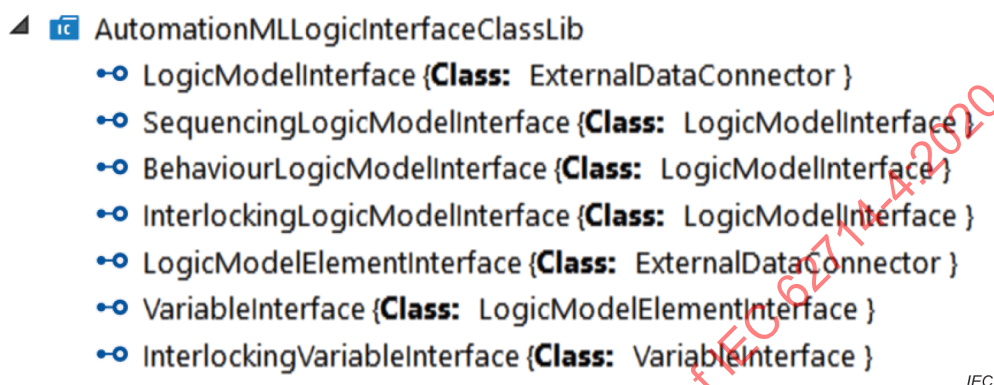


Figure 24 – AutomationMLLogicInterfaceClassLib

10.3.2 InterfaceClass LogicModelInterface

The interface class "LogicModelInterface" shall be used as specified in Table 25.

Table 25 – InterfaceClass LogicModelInterface

Class name	LogicModelInterface
Description	All derivations of the interface class "LogicModelInterface" shall be used in order to reference a logic model stored in a FunctionBlock within an external AML logic XML document. Details are specified in Clause 11. For examples, see Annex B.
Parent class	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector
Path for element reference	AutomationMLLogicInterfaceClassLib/LogicModelInterface
Attributes	none

10.3.3 InterfaceClass SequencingLogicModelInterface

The interface class "SequencingLogicModelInterface" shall be used as specified in Table 26.

Table 26 – InterfaceClass SequencingLogicModelInterface

Class name	SequencingLogicModelInterface
Description	The interface class "SequencingLogicModelInterface" shall be used in order to reference a logic model containing sequencing information stored in a FunctionBlock within an external AML logic XML document. Details are specified in Clause 11. For examples, see Annex B.
Parent class	AutomationMLLogicInterfaceClassLib/LogicModelInterface
Path for element reference	AutomationMLLogicInterfaceClassLib/SequencingLogicModelInterface
Attributes	none

10.3.4 InterfaceClass BehaviourLogicModelInterface

The interface class "BehaviourLogicModelInterface" shall be used as specified in Table 27.

Table 27 – InterfaceClass BehaviourLogicModelInterface

Class name	BehaviourLogicModelInterface
Description	The interface class "BehaviourLogicModelInterface" shall be used in order to reference a logic model containing behaviour information stored in a FunctionBlock within an external AML logic XML document. Details are specified in Clause 11. For examples, see Annex B.
Parent class	AutomationMLLogicInterfaceClassLib/LogicModelInterface
Path for element reference	AutomationMLLogicInterfaceClassLib/BehaviourLogicModelInterface
Attributes	none

10.3.5 InterfaceClass InterlockingLogicModelInterface

The interface class "InterlockingLogicModelInterface" shall be used as specified in Table 28.

Table 28 – InterfaceClass InterlockingLogicModelInterface

Class name	InterlockingLogicModelInterface
Description	The interface class "InterlockingLogicModelInterface" shall be used in order to reference a logic model containing interlocking information stored in a FunctionBlock within an external AML logic XML document. This interface class shall only be used by interlocking source groups and interlocking target groups. Details are specified in Clause 12. For examples, see Annex C.
Parent class	AutomationMLLogicInterfaceClassLib/LogicModelInterface
Path for element reference	AutomationMLLogicInterfaceClassLib/InterlockingLogicModelInterface
Attributes	none

10.3.6 InterfaceClass LogicModelElementInterface

The interface class "LogicModelElementInterface" shall be used as specified in Table 29.

Table 29 – InterfaceClass LogicModelElementInterface

Class name	LogicModelElementInterface
Description	The interface class "LogicModelElementInterface" shall be used to reference an element in a logic model stored in a FunctionBlock within an external AML logic XML document. Details are specified in Clause 11. For examples, see Annex B.
Parent class	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector
Path for element reference	AutomationMLLogicInterfaceClassLib/LogicModelElementInterface
Attributes	none

10.3.7 InterfaceClass VariableInterface

The interface class "VariableInterface" shall be used as specified in Table 30.

Table 30 – InterfaceClass VariableInterface

Class name	VariableInterface
Description	The interface class "VariableInterface" shall be used to reference a variable in a logic model stored in a FunctionBlock within an external AML logic XML document. Details are specified in Clause 11. For examples, see Annex B.
Parent class	AutomationMLLogicInterfaceClassLib/LogicModelElementInterface
Path for element reference	AutomationMLLogicInterfaceClassLib/VariableInterface
Attributes	Direction (type="xs:string") This attribute shall be used to describe a direction of the VariableInterface. Allowed values: "In", "Out", or "InOut". The attribute value shall be "In" if the variable is consumed by CAEX. The attribute value shall be "Out" if the information flow is from CAEX to the variable. The attribute value shall be "InOut" if both directions are possible. The use of the attribute shall be optional.

10.3.8 InterfaceClass InterlockingVariableInterface

The interface class "InterlockingVariableInterface" shall be used as specified in Table 31.

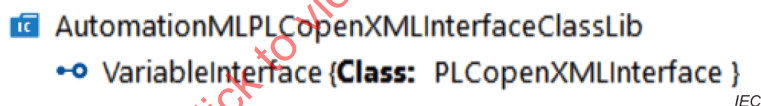
Table 31 – InterfaceClass InterlockingVariableInterface

Class name	InterlockingVariableInterface	
Description	The interface class "InterlockingVariableInterface" shall be used to reference the output variable in a logic model containing interlocking information stored in a FunctionBlock within an external AML logic XML document. This variable represents the result of a function describing the interlocking condition. This interface class shall only be used by interlocking source groups and interlocking target groups. Details are specified in Clause 12. For examples, see Annex C.	
Parent class	AutomationMLLogicInterfaceClassLib/VariableInterface	
Path for element reference	AutomationMLLogicInterfaceClassLib/InterlockingVariableInterface	
Attributes	SafeConditionEquals (type="xs:boolean")	The attribute "SafeConditionEquals" shall be used in order to specify the result of the function describing the interlocking condition. The allowed values shall be "true" or "false" and shall indicate which value of a unique Boolean variable represents the safe state of an interlocking source group. The use of the attribute shall be optional. The default value shall be "true".

10.4 AutomationMLPLCopenXMLInterfaceClassLib

10.4.1 General

Figure 25 presents the normative AutomationMLPLCopenXMLInterfaceClassLib as object tree. AutomationMLPLCopenXMLInterfaceClassLib specifies the logic related interface class for referencing IEC 61131-10 documents. Details to the interface class are given in 10.4.2. The XML text of AutomationMLInterfaceClassLib is given in Clause D.4.

**Figure 25 – AutomationMLPLCopenXMLInterfaceClassLib**

10.4.2 InterfaceClass VariableInterface

The interface class "VariableInterface" shall be used as specified in Table 32.

Table 32 – InterfaceClass VariableInterface

Class name	VariableInterface	
Description	The interface class "VariableInterface" shall be used in order to publish signals or variables that are defined inside of a PLCopen XML document of the PLCopenXML version 2.0, 2.01, or IEC 61131-10.	
Parent class	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector/PLCopenXMLInterface	
Path for element reference	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector/PLCopenXMLInterface/VariableInterface	
Attributes	none	

10.5 AutomationMLInterfaceClassLib

10.5.1 General

Subclause 10.5 defines specifications for AML standard interface classes. These classes are part of the AML standard library. All described attributes and interfaces are part of the standard library and may be removed in the InstanceHierarchy if not needed. Details to the interface classes are given in 10.5.2 and 10.5.3. The XML text of AutomationMLInterfaceClassLib is given in IEC 62714-1.

10.5.2 InterfaceClass InterlockingConnector

The interface class "InterlockingConnector" is a class of the AML standard interface class library, defined in IEC 62714-1, and shall be used as specified in Table 33.

Table 33 – InterfaceClass InterlockingConnector

Class name	InterlockingConnector
Description	The interface class "InterlockingConnector" shall be used in order to model relations between an interlocking source group and an interlocking target group. Details are specified in Clause 12. For examples see Annex C.
Parent class	AutomationMLInterfaceClassLib/AutomationMLBaseInterface
Path for element reference	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/InterlockingConnector
Attributes	none

10.5.3 InterfaceClass PLCopenXMLInterface

The interface class "PLCopenXMLInterface" is a class of the AML standard interface class library, defined in IEC 62714-1, and shall be used as specified in Table 34.

Table 34 – InterfaceClass PLCopenXMLInterface

Class name	PLCopenXMLInterface
Description	The interface class "PLCopenXMLInterface" shall be used in order to reference external PLCopen XML documents of the PLCopenXML version 2.0, 2.01, or IEC 61131-10.
Parent class	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector
Path for element reference	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector/PLCopenXMLInterface
Attributes	none

11 Referencing AML logic XML documents

11.1 General

According to the distributed document structure of AML, it is necessary to put the different information within the different documents in relation to each other. This clause focuses on the referencing of AML objects (within a CAEX document) with logic information stored in AML logic XML documents. An informative overview of the referencing methods is provided in Annex B.

11.2 Referencing logic information

Regarding referencing logic information, the following provisions apply:

- A reference from an AML object to logic information within an AML logic XML document shall be modelled by means of a CAEX ExternalInterface using the AML InterfaceClasses "LogicModelInterface", "LogicModelElementInterface" and their derivations, specified in 10.3.
 - Logic information shall be referenced by its URI within the attribute "refURI" of these CAEX ExternalInterfaces.
 - The value of the attribute "refURI" shall point to the logic information within the AML logic XML document and shall follow the URI syntax, using the AML logic XML attribute "uuid" as fragment.

NOTE 1 Example for the value of the refURI: file:///behaviour.xml#74e94dc8-b5a2-490c-bf0f-f0fdaa4515ef

NOTE 2 In this part of IEC 62714, UUIDs are presented in a short form such as "UUID1", "UUID100" etc. This serves the readability and acts as a real UUID.

NOTE 3 It is not intended to just reference the AML logic XML document itself, since the document might contain several logic models containing logic information.

- Every "LogicModelElementInterface" and its derivations shall refer to one corresponding derivation of "LogicModelInterface" via an InternalLink.

NOTE 4 A derivation of "LogicModelInterface" can be connected to more than one "LogicModelElementInterface" and its derivations via an InternalLink.

NOTE 5 The variables can, then, be assigned, e.g. to signals of other AML objects or to variables of other logic models, by using InternalLinks.

- Logic models shall be distinguishable, i.e. an AML logic object shall not have more than one CAEX ExternalInterface of the same AML InterfaceClass "SequencingLogicModelInterface", "BehaviourLogicModelInterface" or a derivation of them. Every interlocking source group or interlocking target group shall only have one CAEX ExternalInterface of the AML InterfaceClass "InterlockingLogicModelInterface".

NOTE 6 For example, different levels of detail of one logic model are not allowed at one AML logic object.

12 Linking AML objects with interlocking information

12.1 General

Interlocking information is an important part of the engineering of production systems to ensure their safe behaviour. This clause focuses on the interlocking concept which comprises two levels of detail to model and store interlocking information in AML. An informative overview of interlocking concept is provided in Annex C.

NOTE In this part of IEC 62714, interlocking addresses safety relevant dependencies between equipment, e.g. light barriers which prevent humans from getting harmed by a robot. In this part of IEC 62714, interlocking does not address the functional relevant dependencies between equipment, e.g. that one of two collaborating robots needs to wait until it had finished its operation.

12.2 Referencing interlocking information

On the first level of detail of the interlocking concept, interlocking source groups and interlocking target groups are linked to describe functional dependencies among AML object groups respectively AML objects. For this, the following provisions apply:

- An interlocking target group shall be created by using the AML Group concept according to IEC 62714-1.
 - For identifying the group as an interlocking target group the role class "InterlockingTargetGroup" shall be used, which is described in 10.2.2.

NOTE 1 The objects of an interlocking target group execute the actions which are necessary to ensure functional safety.

- An interlocking source group shall be created by using the AML Group concept according to IEC 62714-1.

- For identifying the group as an interlocking source group the role class "InterlockingSourceGroup" shall be used, which is described in 10.2.3.

NOTE 2 The objects of an interlocking source group indicate when actions are necessary to provide functional safety.

- Each interlocking target group and each interlocking source group shall have the CAEX ExternalInterface of the AML InterfaceClass "InterlockingConnector", which is described in 10.5.2.
 - The CAEX ExternalInterface of the interlocking source group shall be linked to the CAEX ExternalInterface of the corresponding interlocking target group via an InternalLink.
- An AML object shall be in zero or more interlocking groups.
- An AML object shall be allowed to be in an interlocking source group as well as in an interlocking target group at the same time.
- One or more interlocking source groups shall be allowed to be connected to one or more interlocking target groups (i.e. the assignment between both groups is not restricted to a 1:1 relation)
- The relations between the different groups shall be independent.

NOTE 3 A state of the interlocking source group affects the interlocking target group.

On the second level of detail of the interlocking concept, the simple grouping of AML objects into interlocking source groups and interlocking target groups (of the first level of detail) is extended by a function describing the interlocking condition of interlocking source groups. For this, the following provisions apply:

- The function shall be represented by a FunctionBlock of the type "FBD", which is described in Clause 7.
- This FunctionBlock shall be referenced by the CAEX ExternalInterface of the AML interface class "InterlockingLogicModelInterface", as specified in 10.3.5.
- The FunctionBlock shall result in a unique Boolean variable describing the output respectively evaluation result of this FunctionBlock.

NOTE 4 This indicates whether the safe state of the interlocking source group is represented by a "true" or "false" value.

- The unique Boolean variable shall be referenced by an interlocking source group modelling one CAEX ExternalInterface of the AML InterfaceClass "InterlockingVariableInterface", as specified in 10.3.8.
 - This CAEX ExternalInterface may have an attribute "SafeConditionEquals", which shall store the Boolean value indicating the safe state. The default value is "true".
- Following the mechanism described in Clause 11, the ExternalInterfaces of the input and output variables of the FunctionBlock shall be connected with the ExternalInterface of the interlocking logic model by InternalLinks.
 - The modelling of the reaction of the corresponding interlocking target groups shall follow the same provisions as defined for the interlocking source group.

Annex A (informative)

Examples for storing logic models in AML logic XML

A.1 Example for storing Gantt charts

A.1.1 General

The examples given in Table A.1 to Table A.4 show the usage of the mapping rules to transform Gantt charts to IML.

A.1.2 Storing of activities without predecessor and successor relation

The Gantt chart example in Table A.1 has no predecessor and successor relations among the bars.

IECNORM.COM : Click to view the full PDF of IEC 62714-4:2020

Table A.1 – Storing of the Gantt chart example "activities without predecessor and successor relations"

Type	Representation
Gantt Chart	The Gantt chart displays three sequential activities on a time axis from 0 to 9 seconds. The first activity, 'Handover to Conveyor', starts at 0s and ends at 4s. The second activity, 'Move to Lift Position', starts at 4s and ends at 7s. The third activity, 'Lift skid', starts at 7s and ends at 9s. The bars are blue and connected by vertical dashed lines at the transition points.
AML logic XML	<pre> <?xml version="1.0" encoding="utf-8"?> <aml:AMLLogic xmlns="http://www.iec.ch/public/TC65SC65BWG7TF10" xmlns:aml="http://www.automationml.org/IEC62714-4Ed1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.automationml.org/IEC62714-4Ed1 AMLLogicXML.xsd" schemaVersion="1.0"> <aml:WriterHeader> <aml:WiterName>ToolA</aml:WiterName> <aml:WiterID>010990003</aml:WiterID> <aml:WriterVender>Vendor1</aml:WriterVender> <aml:WriterVenderURL>https://www.automationml.org/</aml:WriterVenderURL> <aml:WriterVersion>1.00</aml:WriterVersion> <aml:LastWritingDateTime>2002-05-30T09:00:00</aml:LastWritingDateTime> <aml:WriterCulturalSetting>en-US</aml:WriterCulturalSetting> <aml:WriterProjectTitle>SampleXML</aml:WriterProjectTitle> <aml:WriterProjectID>829e1797-dd62-44ac-b29c-c6786d93ca93</aml:WriterProjectID> <aml:AdditionalInformation>This is a sample XML document.</aml:AdditionalInformation> </aml:WriterHeader> <aml:Types> <aml:FunctionBlock uuid="2edd1b2e-64e1-4d2f-9fa6-12022d2ea768" name="Gantt-Chart_Example"> <aml:MainBody xsi:type="aml:IML" logicModelType="GanttChart"> <aml:TimeInformation timeFormat="xsd:duration" startOfScale="PT0S" endOfScale="PT9S" scaleInterval="PT1S"> <aml:IMLStep name="Handover to Conveyor" uuid="f103c162-c087-48b8-a755-3293f63d8fbf" startTime="PT0S" endTime="PT4S"/> <aml:IMLStep name="Move to Lift Position" uuid="415d47ed-c13d-4727-8f13-bc8f65921ee6" startTime="PT4S" endTime="PT7S"/> <aml:IMLStep name="Lift skid" uuid="d60104de-ec05-46df-967b-5ab846e85888" startTime="PT7S" endTime="PT9S"/> </aml:MainBody> </aml:FunctionBlock> </aml:Types> </aml:AMLLogic> </pre>

A.1.3 Storing of an activity sequence

Table A.2 depicts a Gantt chart with a predecessor/successor sequence among the bars.

Table A.2 – Storing of the Gantt chart example "activity sequence"

Type	Representation
Gantt Chart	
AML logic XML	<pre> <?xml version="1.0" encoding="utf-8"?> <aml:AMLLogic xmlns="www.iec.ch/public/TC65SC65BWG7TF10" xmlns:aml="http://www.automationml.org/IEC62714-4Ed1" xmlns:xsi="http://www.w3.org/2001/XMLSchema- instance" xsi:schemaLocation="http://www.automationml.org/IEC62714-4Ed1 AMLLogicXML.xsd" schemaVersion="1.0"> <aml:WriterHeader> <aml:WiterName>ToolA</aml:WiterName> <aml:WiterID>010990003</aml:WiterID> <aml:WriterVender>Vendor1</aml:WriterVender> <aml:WriterVenderURL>https://www.automationml.org/</aml:WriterVenderURL> <aml:WriterVersion>1.00</aml:WriterVersion> <aml:LastWritingDateTime>2002-05-30T09:00:00</aml:LastWritingDateTime> <aml:WriterCulturalSetting>en-US</aml:WriterCulturalSetting> <aml:WriterProjectTitle>SampleXML</aml:WriterProjectTitle> <aml:WriterProjectID>829e1797-dd62-44ac-b29c-c6786d93ca93</aml:WriterProjectID> <aml:AdditionalInformation>This is a sample XML document.</aml:AdditionalInformation> </aml:WriterHeader> <aml:Types> <aml:FunctionBlock uuid="2edd1b2e-64e1-4d2f-9fa6-12022d2ea768" name="Gantt-Chart_Example"> <aml:MainBody xsi:type="aml:IML" logicModelType="GanttChart"> <aml:TimeInformation timeFormat="xsd:duration" startOfScale="PT0S" endOfScale="PT9S" scaleInterval="PT1S"> </aml:TimeInformation> <aml:IMLStep name="Handover to Conveyor" uuid="f103c162-c087-48b8-a755-3293f63d8bf" startTime="PT0S" endTime="PT4S"> <ConnectionPointOut connectionPointOutId="1"/> </aml:IMLStep> <aml:IMLStep name="Move to Lift Position" uuid="415d47ed-c13d-4727-8f13-bc8f65921ee6" startTime="PT4S" endTime="PT7S"> <ConnectionPointOut connectionPointOutId="2"/> <ConnectionPointIn> <Connection refConnectionPointOutId="3"/> </ConnectionPointIn> </aml:IMLStep> <aml:IMLStep name="Lift skid" uuid="d60104de-ec05-46df-967b-5ab846e85888" startTime="PT7S" endTime="PT9S"> <ConnectionPointIn> <Connection refConnectionPointOutId="4"/> </ConnectionPointIn> </aml:IMLStep> <aml:IMLTransition uuid="78cb4068-b9ea-4492-ab64-587839949a8e"> <ConnectionPointIn> <Connection refConnectionPointOutId="1"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="3"/> </aml:IMLTransition> <aml:IMLTransition uuid="909b2b5b-8575-41ea-986e-b20d81d302d7"> <ConnectionPointIn> <Connection refConnectionPointOutId="2"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="4"/> </aml:IMLTransition> </aml:MainBody> </aml:FunctionBlock> </aml:Types> </aml:AMLLogic> </pre>

A.1.4 Storing of an activity sequence with divergences

In Table A.3, a simultaneous divergence is introduced to describe multiple successors of one bar in IML.

Table A.3 – Storing of the Gantt chart example "activity sequence with divergence"

Type	Representation
Gantt Chart	The Gantt Chart displays three activities over a timeline from 0 to 18 seconds. 'Initialize Robot 1' runs from 0 to 6 seconds. At 6 seconds, the sequence diverges into two parallel activities: 'Execute Manufacturing Robot 1' (6 to 18 seconds) and 'Initialize Robot 2' (6 to 10 seconds). The timeline ends at 18 seconds.
AML logic XML	<pre> <?xml version="1.0" encoding="utf-8"?> <aml:AMLLogic xmlns="http://www.iec.ch/public/TC65SC65BWG7TF10" xmlns:aml="http://www.automationml.org/IEC62714-4Ed1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.automationml.org/IEC62714-4Ed1 AMLLogicXML.xsd" schemaVersion="1.0"> <aml:WriterHeader> <aml:WriterName>ToolA</aml:WriterName> <aml:WriterID>010990003</aml:WriterID> <aml:WriterVendor>Vendor1</aml:WriterVendor> <aml:WriterVendorURL>https://www.automationml.org/</aml:WriterVendorURL> <aml:WriterVersion>1.00</aml:WriterVersion> <aml:LastWritingDateTime>2002-05-30T09:00:00</aml:LastWritingDateTime> <aml:WriterCulturalSetting>en-US</aml:WriterCulturalSetting> <aml:WriterProjectTitle>SampleXML</aml:WriterProjectTitle> <aml:WriterProjectID>829e1797-dd62-44ac-b29c-c6786d93ca93</aml:WriterProjectID> <aml:AdditionalInformation>This is a sample XML document.</aml:AdditionalInformation> </aml:WriterHeader> <aml:Types> <aml:FunctionBlock uuid="2eddb1b2e-64e1-4d2f-9fa6-12022d2ea768" name="Gantt-Chart_Example"> <aml:MainBody xsi:type="aml:IML" logicModelType="GanttChart"> <aml:TimeInformation timeFormat="xsd:duration" startOfScale="PT0S" endOfScale="PT18S" scaleInterval="PT1S"> </aml:TimeInformation> <aml:IMLStep name="Initialize Robot 1" uuid="f103c162-c087-48b8-a755-3293f63d8fbf" startTime="PT0S" endTime="PT6S"> <ConnectionPointOut connectionPointOutId="1"/> </aml:IMLStep> <aml:IMLStep name="Execute Manufacturing Robot 1" uuid="415d47ed-c13d-4727-8f13-bc8f65921ee6" startTime="PT9S" endTime="PT18S"> <ConnectionPointIn> <Connection refConnectionPointOutId="3"/> </ConnectionPointIn> </aml:IMLStep> <aml:IMLStep name="Initialize Robot 2" uuid="d60104de-ec05-46df-967b-5ab846e85888" startTime="PT6S" endTime="PT10S"> <ConnectionPointIn> <Connection refConnectionPointOutId="4"/> </ConnectionPointIn> </aml:IMLStep> <aml:IMLTransition uuid="78cb4068-b9ea-4492-ab64-587839949a8e"> <ConnectionPointIn> <Connection refConnectionPointOutId="1"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="2"/> </aml:IMLTransition> <aml:IMLSimultaneousDivergence uuid="ffc63a33-5f4f-48fe-98d3-6a49deaaec1"> <ConnectionPointIn> <Connection refConnectionPointOutId="2"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="3"/> </aml:IMLSimultaneousDivergence> </aml:MainBody> </aml:FunctionBlock> </aml:Types> </aml:AMLLogic> </pre>

Type	Representation
	<pre> <ConnectionPointOut connectionPointOutId="4"/> </aml:IMLSimultaneousDivergence> </aml:MainBody> </aml:FunctionBlock> </aml:Types> </aml:AMLLogic> </pre>

A.1.5 Storing of an activity sequence with convergences

The Gantt chart example in Table A.4 includes a convergence with the predecessor sequence among the bars of the Gantt chart. This results in a simultaneous convergence in the IML.

Table A.4 – Storing of the Gantt chart example "activity sequence with convergences"

Type	Representation
Gantt Chart	
AML logic XML	<pre> <?xml version="1.0" encoding="utf-8"?> <aml:AMLLogic xmlns="http://www.iec.ch/public/TC65SC65BWG7TF10" xmlns:aml="http://www.automationml.org/IEC62714-4Ed1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.automationml.org/IEC62714-4Ed1 AMLLogicXML.xsd" schemaVersion="1.0"> <aml:WriterHeader> <aml:WiterName>ToolA</aml:WiterName> <aml:WiterID>010990003</aml:WiterID> <aml:WriterVender>Vendor1</aml:WriterVender> <aml:WriterVenderURL>https://www.automationml.org/</aml:WriterVenderURL> <aml:WriterVersion>1.00</aml:WriterVersion> <aml:LastWritingDateTime>2002-05-30T09:00:00</aml:LastWritingDateTime> <aml:WriterCulturalSetting>en-US</aml:WriterCulturalSetting> <aml:WriterProjectTitle>SampleXML</aml:WriterProjectTitle> <aml:WriterProjectID>829e1797-dd62-44ac-b29c-c6786d93ca93</aml:WriterProjectID> <aml:AdditionalInformation>This is a sample XML document.</aml:AdditionalInformation> </aml:WriterHeader> <aml:Types> <aml:FunctionBlock uuid="2edd1b2e-64e1-4d2f-9fa6-12022d2ea768" name="Gantt-Chart_Example"> <aml:MainBody xsi:type="aml:IML" logicModelType="GanttChart"> <aml:TimeInformation timeFormat="xsd:duration" startOfScale="PT0S" endOfScale="PT18S" scaleInterval="PT1S"> </aml:TimeInformation> <aml:IMLStep name="Initialize Robot 1" uuid="f103c162-c087-48b8-a755-3293f63d8fbf" startTime="PT0S" endTime="PT6S"> <ConnectionPointOut connectionPointOutId="1"/> </aml:IMLStep> <aml:IMLStep name="Lift skid" uuid="415d47ed-c13d-4727-8f13-bc8f65921ee6" startTime="PT7S" endTime="PT9S"> <ConnectionPointOut connectionPointOutId="2"/> </aml:IMLStep> <aml:IMLStep name="Execute Manufacturing Robot 1" uuid="d60104de-ec05-46df-967b-5ab846e85888" startTime="PT9S" endTime="PT18S"> <ConnectionPointIn> <Connection refConnectionPointOutId="4"/> </ConnectionPointIn> </aml:IMLStep> <aml:IMLSimultaneousConvergence uuid="ffc63a33-5f4f-48fe-98d3-6a49deaaec1"> <ConnectionPointIn> <Connection refConnectionPointOutId="1"/> </ConnectionPointIn> </aml:IMLSimultaneousConvergence> </aml:MainBody> </aml:FunctionBlock> </aml:Types> </aml:AMLLogic> </pre>

Type	Representation
	<pre> </ConnectionPointIn> <ConnectionPointIn> <Connection refConnectionPointOutId="2"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="3"/> </aml:IMLSimultaneousConvergence> <aml:IMLTransition uuid="78cb4068-b9ea-4492-ab64-587839949a8e"> <ConnectionPointIn> <Connection refConnectionPointOutId="3"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="4"/> </aml:IMLTransition> </aml:MainBody> </aml:FunctionBlock> </aml:Types> </aml:AMLLogic> </pre>

A.2 Example for storing activity-on-node networks

A.2.1 General

The examples in A.2.2 to A.2.5 show the usage of the mapping rules to transform activity-on-node networks to IML

A.2.2 Storing of activities without predecessor and successor relation

The activity-on-node network example shown in Table A.5 has no predecessor and successor relations among the nodes.

Table A.5 – Storing of the activity-on-node network example "activities without predecessor and successor relations"

Type	Representation										
Activity-on-node network	0	4	4		4	3	7		7	2	9
	Handover to Conveyor				Move to Lift Position				Lift Skid		
	0	0	4		6	2	9		7	0	9
AML logic XML	<pre><?xml version="1.0" encoding="utf-8"?> <aml:AMLLLogic xmlns="www.iec.ch/public/TC65SC65BWG7TF10" xmlns:aml="http://www.automationml.org/IEC62714-4Ed1" xmlns:xsi="http://www.w3.org/2001/XMLSchema- instance" xsi:schemaLocation="http://www.automationml.org/IEC62714-4Ed1 AMLLogicXML.xsd" schemaVersion="1.0"> <aml:WriterHeader> <aml:WiterName>ToolA</aml:WiterName> <aml:WiterID>010990003</aml:WiterID> <aml:WriterVender>Vendor1</aml:WriterVender> <aml:WriterVenderURL>https://www.automationml.org/</aml:WriterVenderURL> <aml:WriterVersion>1.00</aml:WriterVersion> <aml:LastWritingDateTime>2002-05-30T09:00:00</aml:LastWritingDateTime> <aml:WriterCulturalSetting>en-US</aml:WriterCulturalSetting> <aml:WriterProjectTitle>SampleXML</aml:WriterProjectTitle> <aml:WriterProjectID>829e1797-dd62-44ac-b29c-c6786d93ca93</aml:WriterProjectID> <aml:AdditionalInformation>This is a sample XML document.</aml:AdditionalInformation> </aml:WriterHeader> <aml:Types> <aml:FunctionBlock uuid="2edd1b2e-64e1-4d2f-9fa6-12022d2ea768" name="Activity-on- node_network_Example"> <aml:MainBody xsi:type="aml:IML" logicModelType="ActivityOnNodeNetwork"> <aml:TimeInformation timeFormat="xsd:duration"/> <aml:IMLStep name="Handover to Conveyor" uuid="f103c162-c087-48b8-a755-3293f63d8bf" startTime="PT0S" endTime="PT4S"/> <aml:IMLStep name="Move to Lift Position" uuid="415d47ed-c13d-4727-8f13-bc8f65921ee6" startTime="PT4S" endTime="PT7S"/> <aml:IMLStep name="Lift skid" uuid="d60104de-ec05-46df-967b-5ab846e85888" startTime="PT7S" endTime="PT9S"/> </aml:MainBody> </aml:FunctionBlock> </aml:Types> </aml:AMLLLogic></pre>										

A.2.3 Storing of an activity sequence

The example given in Table A.6 depicts an activity-on-node network with a predecessor/successor sequence among the nodes.

Table A.6 – Storing of the activity-on-node network example "activity sequence"

Type	Representation										
Activity-on-node network	044			→	437			→	729		
	Handover to Conveyor				Move to Lift Position				Lift Skid		
	004				629				709		
AML logic XML	<?xml version="1.0" encoding="utf-8"?> <aml:AMLLLogic xmlns="www.iec.ch/public/TC65SC65BWG7TF10" xmlns:aml="http://www.automationml.org/IEC62714-4Ed1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-										

Type	Representation
	<pre> instance" xsi:schemaLocation="http://www.automationml.org/IEC62714-4Ed1 AMLLogicXML.xsd" schemaVersion="1.0"> <aml:WriterHeader> <aml:WiterName>ToolA</aml:WiterName> <aml:WiterID>010990003</aml:WiterID> <aml:WriterVender>Vendor1</aml:WriterVender> <aml:WriterVenderURL>https://www.automationml.org/</aml:WriterVenderURL> <aml:WriterVersion>1.00</aml:WriterVersion> <aml:LastWritingDateTime>2002-05-30T09:00:00</aml:LastWritingDateTime> <aml:WriterCulturalSetting>en-US</aml:WriterCulturalSetting> <aml:WriterProjectTitle>SampleXML</aml:WriterProjectTitle> <aml:WriterProjectID>829e1797-dd62-44ac-b29c-c6786d93ca93</aml:WriterProjectID> <aml:AdditionalInformation>This is a sample XML document.</aml:AdditionalInformation> </aml:WriterHeader> <aml:Types> <aml:FunctionBlock uuid="2edd1b2e-64e1-4d2f-9fa6-12022d2ea768" name="Activity-on- node_network_Example"> <aml:MainBody xsi:type="aml:IML" logicModelType="ActivityOnNodeNetwork "> <aml:TimeInformation timeFormat="xsd:duration"> </aml:TimeInformation> <aml:IMLStep name="Handover to Conveyor" uuid="f103c162-c087-48b8-a755-3293f63d8fbf" startTime="PT0S" endTime="PT4S"> <ConnectionPointOut connectionPointOutId="1"/> </aml:IMLStep> <aml:IMLStep name="Move to Lift Position" uuid="415d47ed-c13d-4727-8f13-bc8f65921ee6" startTime="PT4S" endTime="PT7S"> buffer="PT2H"> <ConnectionPointIn> <Connection refConnectionPointOutId="3"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="2"/> </aml:IMLStep> <aml:IMLStep name="Lift skid" uuid="d60104de-ec05-46df-967b-5ab846e85888" startTime="PT7S" endTime="PT9S"> <ConnectionPointIn> <Connection refConnectionPointOutId="4"/> </ConnectionPointIn> </aml:IMLStep> <aml:IMLTransition uuid="78cb4068-b9ea-4492-ab64-587839949a8e"> <ConnectionPointIn> <Connection refConnectionPointOutId="1"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="3"/> </aml:IMLTransition> <aml:IMLTransition uuid="909b2b5b-8575-41ea-986e-b20d81d302d7"> <ConnectionPointIn> <Connection refConnectionPointOutId="2"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="4"/> </aml:IMLTransition> </aml:MainBody> </aml:FunctionBlock> </aml:Types> </aml:AMLLogic> </pre>

A.2.4 Storing of an activity sequence with divergences

In example given in Table A.7 a simultaneous divergence is introduced to describe multiple successors of one node in IML.

Table A.7 – Storing of the activity-on-node network example "activity sequence with divergence"

Type	Representation
Activity-on-node network	The diagram illustrates an activity-on-node network with two parallel paths diverging from a single start node. The first path consists of 'Initialize Robot 1' (0 to 6) followed by 'Execute Manufacturing Robot 1' (6 to 10). The second path consists of 'Initialize Robot 2' (0 to 6) followed by 'Execute Manufacturing Robot 2' (6 to 10).
AML logic XML	<pre> <?xml version="1.0" encoding="utf-8"?> <aml:AMLLogic xmlns="http://www.iec.ch/public/TC65SC65BWG7TF10" xmlns:aml="http://www.automationml.org/IEC62714-4Ed1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.automationml.org/IEC62714-4Ed1 AMLLogicXML.xsd" schemaVersion="1.0"> <aml:WriterHeader> <aml:WriterName>ToolA</aml:WriterName> <aml:WriterID>010990003</aml:WriterID> <aml:WriterVender>Vendor1</aml:WriterVender> <aml:WriterVenderURL>https://www.automationml.org/</aml:WriterVenderURL> <aml:WriterVersion>1.00</aml:WriterVersion> <aml:LastWritingDateTime>2002-05-30T09:00:00</aml:LastWritingDateTime> <aml:WriterCulturalSetting>en-US</aml:WriterCulturalSetting> <aml:WriterProjectTitle>SampleXML</aml:WriterProjectTitle> <aml:WriterProjectID>829e1797-dd62-44ac-b29c-c6786d93ca93</aml:WriterProjectID> <aml:AdditionalInformation>This is a sample XML document.</aml:AdditionalInformation> </aml:WriterHeader> <aml:Types> <aml:FunctionBlock uuid="2edd1b2e-64e1-4d2f-9fa6-12022d2ea768" name="Activity-on-node_network_Example"> <aml:MainBody xsi:type="aml:IML" logicModelType="ActivityOnNodeNetwork"> <aml:TimeInformation timeFormat="xsd:duration"/> <aml:IMLStep name="Initialize Robot 1" uuid="f103c162-c087-48b8-a755-3293f63d8fbf" startTime="PT0S" endTime="PT6S"> <ConnectionPointOut connectionPointOutId="1"/> </aml:IMLStep> <aml:IMLStep name="Execute Manufacturing Robot 1" uuid="415d47ed-c13d-4727-8f13-bc8f65921ee6" startTime="PT9S" endTime="PT18S"> <ConnectionPointIn> <Connection refConnectionPointOutId="3"/> </ConnectionPointIn> </aml:IMLStep> <aml:IMLStep name="Initialize Robot 2" uuid="d60104de-ec05-46df-967b-5ab846e85888" startTime="PT6S" endTime="PT10S"> <ConnectionPointIn> <Connection refConnectionPointOutId="4"/> </ConnectionPointIn> </aml:IMLStep> <aml:IMLTransition uuid="78cb4068-b9ea-4492-ab64-587839949a8e"> <ConnectionPointIn> <Connection refConnectionPointOutId="1"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="2"/> </aml:IMLTransition> <aml:IMLSimultaneousDivergence uuid="ffc63a33-5f4f-48fe-98d3-6a49deaaec1"> <ConnectionPointIn> <Connection refConnectionPointOutId="2"/> </ConnectionPointIn> </aml:IMLSimultaneousDivergence> </aml:MainBody> </aml:FunctionBlock> </aml:Types> </aml:AMLLogic> </pre>

Type	Representation
	<pre> <ConnectionPointOut connectionPointOutId="3"/> <ConnectionPointOut connectionPointOutId="4"/> </aml:IMLSimultaneousDivergence> </aml:MainBody> </aml:FunctionBlock> </aml:Types> </aml:AMLLogic> </pre>

A.2.5 Storing of an activity sequence with convergences

The activity-on-node network example given in Table A.8 includes a convergence with the predecessor sequence among the nodes of the activity-on-node network. This results in a simultaneous convergence in the IML.

Table A.8 – Storing of the activity-on-node network example "activity sequence with convergences"

Type	Representation
Activity-on-node network	

Type	Representation
AML logic XML	<pre> <?xml version="1.0" encoding="utf-8"?> <aml:AMLLogic xmlns="www.iec.ch/public/TC65SC65BWG7TF10" xmlns:aml="http://www.automationml.org/IEC62714-4Ed1" xmlns:xsi="http://www.w3.org/2001/XMLSchema- instance" xsi:schemaLocation="http://www.automationml.org/IEC62714-4Ed1 AMLLogicXML.xsd" schemaVersion="1.0"> <aml:WriterHeader> <aml:WiterName>ToolA</aml:WiterName> <aml:WiterID>010990003</aml:WiterID> <aml:WriterVender>Vendor1</aml:WriterVender> <aml:WriterVenderURL>https://www.automationml.org/</aml:WriterVenderURL> <aml:WriterVersion>1.00</aml:WriterVersion> <aml:LastWritingDateTime>2002-05-30T09:00:00</aml:LastWritingDateTime> <aml:WriterCulturalSetting>en-US</aml:WriterCulturalSetting> <aml:WriterProjectTitle>SampleXML</aml:WriterProjectTitle> <aml:WriterProjectID>829e1797-dd62-44ac-b29c-c6786d93ca93</aml:WriterProjectID> <aml:AdditionalInformation>This is a sample XML document.</aml:AdditionalInformation> </aml:WriterHeader> <aml:Types> <aml:FunctionBlock uuid="2edd1b2e-64e1-4d2f-9fa6-12022d2ea768" name="Activity on node_network_Example"> <aml:MainBody xsi:type="aml:IML" logicModelType="ActivityOnNodeNetwork"> <aml:TimeInformation timeFormat="xsd:duration"> </aml:TimeInformation> <aml:IMLStep name="Initialize Robot 1" uuid="f103c162-c087-48b8-a755-3293f63d8bfb" startTime="PT0S" endTime="PT6S"> buffer="PT6H"> <ConnectionPointOut connectionPointOutId="1"/> </aml:IMLStep> <aml:IMLStep name="Lift skid" uuid="415d47ed-c13d-4727-8f13-bc8f65921ee6" startTime="PT7S" endTime="PT9S"> buffer="PT2H"> <ConnectionPointOut connectionPointOutId="2"/> </aml:IMLStep> <aml:IMLStep name="Execute Manufacturing Robot 1" uuid="d60104de-ec05-46df-967b-5ab846e85888" startTime="PT9S" endTime="PT18S"> <ConnectionPointIn> <Connection refConnectionPointOutId="4"/> </ConnectionPointIn> </aml:IMLStep> <aml:IMLSimultaneousConvergence uuid="ffc63a33-5f4f-48fe-98d3-6a49deaaec1"> <ConnectionPointIn> <Connection refConnectionPointOutId="1"/> </ConnectionPointIn> <ConnectionPointIn> <Connection refConnectionPointOutId="2"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="3"/> </aml:IMLSimultaneousConvergence> <aml:IMLTransition uuid="78cb4068-b9ea-4492-ab64-587839949a8e"> <ConnectionPointIn> <Connection refConnectionPointOutId="3"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="4"/> </aml:IMLTransition> </aml:MainBody> </aml:FunctionBlock> </aml:Types> </aml:AMLLogic> </pre>

A.3 Example for storing timing diagrams

A.3.1 General

The examples given in Table A.9 to Table A.11 show the use of the mapping rules to transform timing diagrams to IML.

A.3.2 Example of storing internal signal

The example in Table A.9 handles the transition from a state change to the subsequent state.

Table A.9 – Storing of the timing diagram example "transition from a state change to the subsequent state"

Type	Representation
Timing diagram	The diagram shows a timing diagram for a resource named 'Motor1'. The vertical axis represents the state, with 'Fast_run' and 'Slow_run' states. The horizontal axis represents time in seconds (t(s)), with markers at 0, 1, 2, and 3. A 'Time signal' is shown as a blue line that starts at 'Slow_run' at t=0, transitions to 'Fast_run' at t=1, and then transitions back to 'Slow_run' at t=2. A 'Motor signal' is shown as a blue line that starts at 'Slow_run' at t=0, transitions to 'Fast_run' at t=1, and then transitions back to 'Slow_run' at t=2. The diagram is divided into three segments: 'Motor1_1 segment' (from t=0 to t=1), 'Motor1_2 segment' (from t=1 to t=2), and 'Motor1_3 segment' (from t=2 to t=3).
AML logic XML	<pre> <?xml version="1.0" encoding="utf-8"?> <aml:AMLLLogic xmlns="http://www.iec.ch/public/TC65SC65BWG7TF10" xmlns:aml="http://www.automationml.org/IEC62714-4Ed1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.automationml.org/IEC62714-4Ed1 AMLLogicXML.xsd" schemaVersion="1.0"> <aml:WriterHeader> <aml:WriterName>ToolA</aml:WriterName> <aml:WriterID>010990003</aml:WriterID> <aml:WriterVendor>Vendor1</aml:WriterVendor> <aml:WriterVendorURL>https://www.automationml.org/</aml:WriterVendorURL> <aml:WriterVersion>1.00</aml:WriterVersion> <aml:LastWritingDateTime>2002-05-30T09:00:00</aml:LastWritingDateTime> <aml:WriterCulturalSetting>en-US</aml:WriterCulturalSetting> <aml:WriterProjectTitle>SampleXML</aml:WriterProjectTitle> <aml:WriterProjectID>829e1797-dd62-44ac-b29c-c6786d93ca93</aml:WriterProjectID> <aml:AdditionalInformation>This is a sample XML document.</aml:AdditionalInformation> </aml:WriterHeader> <aml:Types> <aml:FunctionBlock uuid="7040cf3a-745e-413e-8220-557545d25497" name="Timing-Diagram_Example"> <aml:MainBody xsi:type="aml:IML" logicModelType="TimingDiagram"> <aml:Resource name="Motor1" uuid="9be204e8-f3b5-4551-b8f7-a6f8894899fd"> <aml:Status name="Fast_run" uuid="630f2307-2aa5-4d57-a436-fc30aeb10faf"/> <aml:Status name="Slow_run" uuid="2a956645-faea-4c08-91e1-0ea8fd24f75d"/> </aml:Resource> <aml:TimeInformation timeFormat="xsd:duration" startOfScale="PT0S" endOfScale="PT4S" scaleInterval="PT1S"/> <aml:IMLStep name="Motor1_1 segment" uuid="cc8c1223-3c82-4e4b-a1da-341db56a4ec3" startTime="PT0S" endTime="PT1S"> <ConnectionPointOut connectionPointOutId="1"/> </aml:IMLStep> <aml:IMLTransition uuid="bf3b9413-9312-4527-a9b2-944ef91c5ec4" refTime="PT1S"> <ConnectionPointIn> <Connection refConnectionPointOutId="1"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="2"/> </aml:IMLTransition> <aml:IMLStep name="Motor1_2 segment" uuid="ac84ba1b-ff34-4f85-ba40-a3ebe4c3829c" startTime="PT1S" endTime="PT2S"> <ConnectionPointIn> <Connection refConnectionPointOutId="2"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="3"/> </aml:IMLStep> <aml:IMLTransition uuid="bf3b9413-9312-4527-a9b2-944ef91c5ec4" refEndSegmentId="ac84ba1b-ff34-4f85-ba40-a3ebe4c3829c"> <ConnectionPointIn> </pre>

Type	Representation
	<pre> <Connection refConnectionPointOutId="3"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="4"/> </aml:IMLTransition> <aml:IMLStep name="Motor1_3 segment" uuid="ac84ba1b-ff34-4f85-ba40-a3ebe4c3829c" startTime="PT2S" endTime="PT3S"> <ConnectionPointIn> <Connection refConnectionPointOutId="4"/> </ConnectionPointIn> </aml:IMLStep> </aml:MainBody> </aml:FunctionBlock> </aml:Types> </aml:AMLLogic> </pre>

A.3.3 Example of storing external signal

The example in Table A.10 handles two external signals, which are fired with a delay of three seconds.

Table A.10 – Mapping of the timing diagram example "two external signals fired with delay of three seconds"

Type	Representation
Timing diagram	The timing diagram shows the states of Motor1 over time t(s). The time axis is marked from 0 to 4 seconds. The Motor1 resource has two states: Fast_run and Slow_run. The diagram is divided into three segments: Motor1_1 (0 to 1s), Motor1_2 (1 to 3s), and Motor1_3 (3 to 4s). Time signal 1 is fired at t=3s and Time signal 2 is fired at t=4s.
AML logic XML	<pre> <?xml version="1.0" encoding="utf-8"?> <aml:AMLLogic xmlns="http://www.iec.ch/public/TC65SC65BWG7TF10" xmlns:aml="http://www.automationml.org/IEC62714-4Ed1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.automationml.org/IEC62714-4Ed1 AMLLogicXML.xsd" schemaVersion="1.0"> <aml:WriterHeader> <aml:WriterName>ToolA</aml:WriterName> <aml:WriterID>010990003</aml:WriterID> <aml:WriterVendor>Vendor1</aml:WriterVendor> <aml:WriterVendorURL>https://www.automationml.org/</aml:WriterVendorURL> <aml:WriterVersion>1.00</aml:WriterVersion> <aml:LastWritingDateTime>2002-05-30T09:00:00</aml:LastWritingDateTime> <aml:WriterCulturalSetting>en-US</aml:WriterCulturalSetting> <aml:WriterProjectTitle>SampleXML</aml:WriterProjectTitle> <aml:WriterProjectID>829e1797-dd62-44ac-b29c-c6786d93ca93</aml:WriterProjectID> <aml:AdditionalInformation>This is a sample XML document.</aml:AdditionalInformation> </aml:WriterHeader> <aml:Types> <aml:FunctionBlock uuid="7040cf3a-745e-413e-8220-557545d25497" name="Timing-Diagram_Example"> <aml:MainBody xsi:type="aml:IML" logicModelType="TimingDiagram"> <aml:Resource name="Motor1" uuid="9be204e8-f3b5-4551-b8f7-a6f8894899fd"> <aml:Status name="Fast_run" uuid="630f2307-2aa5-4d57-a436-fc30aeb10faf"/> <aml:Status name="Slow_run" uuid="2a956645-faea-4c08-91e1-0ea8fd24f75d"/> </aml:Resource> <aml:TimeInformation timeFormat="xsd:duration" startOfScale="PT0S" endOfScale="PT3S" scaleInterval="PT1S"/> <aml:IMLTransition uuid="5b6fa623-6bd1-4d06-99cd-6fcfec0e2960" refTime="PT0S"> <ConnectionPointOut connectionPointOutId="1"/> </pre>

Type	Representation
	<pre> </aml:IMLTransition> <aml:IMLStep name="Motor1_1 segment" uuid="cc8c1223-3c82-4e4b-a1da-341db56a4ec3" startTime="PT0S" endTime="PT1S"> <ConnectionPointIn> <Connection refConnectionPointOutId="1"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="2"/> </aml:IMLStep> <aml:IMLTransition uuid="bf3b9413-9312-4527-a9b2-944ef91c5ec4"> <ConnectionPointIn> <Connection refConnectionPointOutId="2"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="3"/> </aml:IMLTransition> <aml:IMLStep name="Motor1_2 segment" uuid="ac84ba1b-ff34-4f85-ba40-a3ebe4c3829c" startTime="PT1S" endTime="PT3S"> <ConnectionPointIn> <Connection refConnectionPointOutId="3"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="4"/> </aml:IMLStep> <aml:IMLTransition uuid="e9e38b5d-efcc-4b0d-be04-ba281756f511" refTime="PT3S"> <ConnectionPointIn> <Connection refConnectionPointOutId="4"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="5"/> </aml:IMLTransition> <aml:IMLStep name="Motor1_3 segment" uuid="053d73af-242d-4049-b7af-9af0ecef150c" startTime="PT3S" endTime="PT4S"> <ConnectionPointIn> <Connection refConnectionPointOutId="5"/> </ConnectionPointIn> </aml:IMLStep> </aml:MainBody> </aml:FunctionBlock> </aml:Types> </aml:AMLLLogic> </pre>

A.3.4 Example of storing signal between two resource states flows

The example in Table A.11 handles the transformation of a signal fired by one resource state and consumed by another.

Table A.11 – Storing of the timing diagram example "signal fired by one resource state and consumed by another"

Type	Representation
Timing diagram	
AML logic XML	<pre> <?xml version="1.0" encoding="utf-8"?> <aml:AMLLogic xmlns="www.iec.ch/public/TC65SC65BWG7TF10" xmlns:aml="http://www.automationml.org/IEC62714-4Ed1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.automationml.org/IEC62714-4Ed1 AMLLogicXML.xsd" schemaVersion="1.0"> <aml:WriterHeader> <aml:WriterName>ToolA</aml:WriterName> <aml:WriterID>010990003</aml:WriterID> <aml:WriterVender>Vendor1</aml:WriterVender> <aml:WriterVenderURL>https://www.automationml.org/</aml:WriterVenderURL> <aml:WriterVersion>1.00</aml:WriterVersion> <aml:LastWritingDateTime>2002-05-30T09:00:00</aml:LastWritingDateTime> <aml:WriterCulturalSetting>en-US</aml:WriterCulturalSetting> <aml:WriterProjectTitle>SampleXML</aml:WriterProjectTitle> <aml:WriterProjectID>829e1797-dd62-44ac-b29c-c6786d93ca93</aml:WriterProjectID> <aml:AdditionalInformation>This is a sample XML document.</aml:AdditionalInformation> </aml:WriterHeader> <aml:Types> <aml:FunctionBlock uuid="7040cf3a-745e-413e-8220-557545d25497" name="Timing-Diagram_Example"> <aml:MainBody xsi:type="aml:IML" logicModelType="TimingDiagram"> <aml:Resource name="Motor1" uuid="9be204e8-f3b5-4551-b8f7-a6f8894899fd"> <aml:Status name="Fast_run" uuid="630f2307-2aa5-4d57-a436-fc30aeb10faf"/> <aml:Status name="Slow_run" uuid="2a956645-faea-4c08-91e1-0ea8fd24f75d"/> </aml:Resource> <aml:Resource name="Gripper1" uuid="9238a217-0f72-48ae-b68b-adc79089c89d"> <aml:Status name="Open" uuid="558412d4-844f-4ab1-ab6f-5e8de37cb871"/> <aml:Status name="Close" uuid="480bb954-151b-49a0-acc4-c06410c70dd0"/> </aml:Resource> <aml:TimeInformation timeFormat="xsd:duration" startOfScale="PT0S" endOfScale="PT3S" scaleInterval="PT1S"/> <aml:IMLStep name="Motor1_1 segment" uuid="cc8c1223-3c82-4e4b-a1da-341db56a4ec3" startTime="PT0S" endTime="PT1S"> <ConnectionPointOut connectionPointOutId="1"/> </aml:IMLStep> <aml:IMLTransition uuid="bf3b9413-9312-4527-a9b2-944ef91c5ec4"> <ConnectionPointIn> <Connection refConnectionPointOutId="1"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="2"/> </aml:IMLTransition> <aml:IMLStep name="Motor1_2 segment" uuid="ac84ba1b-ff34-4f85-ba40-a3ebe4c3829c" startTime="PT1S" endTime="PT3S"> <ConnectionPointIn> <Connection refConnectionPointOutId="2"/> </ConnectionPointIn> </aml:IMLStep> </aml:MainBody> </aml:FunctionBlock> </aml:Types> </aml:AMLLogic> </pre>

Type	Representation
	<pre> <aml:IMLStep name="Gripper1_1 segment" uuid="db04c65d-907d-4f7d-9477-966f09c2c880" startTime="PT0S" endTime="PT1S"> <ConnectionPointOut connectionPointOutId="3"/> </aml:IMLStep> <aml:IMLTransition uuid="635abe59-5e21-4107-88be-e70fdb05a764" refEndSegmentId="cc8c1223-3c82-4e4b-a1da-341db56a4ec3"> <ConnectionPointIn> <Connection refConnectionPointOutId="3"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="4"/> </aml:IMLTransition> <aml:IMLStep name="Gripper1_2 segment" uuid="d33b1e7c-68f1-4841-bc1e-54f6c77f7998" startTime="PT1S" endTime="PT3S"> <ConnectionPointIn> <Connection refConnectionPointOutId="4"/> </ConnectionPointIn> </aml:IMLStep> </aml:MainBody> </aml:FunctionBlock> </aml:Types> </aml:AMLLogic> </pre>

A.4 Example for storing sequential function charts

The example given in Table A.12 shows the use of the mapping rules to store sequential function charts in AML logic XML.

Table A.12 – Example for storing sequential function chart

Type	Representation
Sequential function chart	<pre> graph TD S0[S0] -- TRUE --> S1[S1] S1 -- "S1.T > T#1s" --> S2[S2] S2 -- "S2.T > T#1s" --> S1 S1 --> N[N OUTPUT] </pre>
AML logic XML	<pre> <?xml version="1.0" encoding="utf-8"?> <aml:AMLLogic xmlns="www.iec.ch/public/TC65SC65BWG7TF10" xmlns:aml="http://www.automationml.org/IEC62714-4Ed1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.automationml.org/IEC62714-4Ed1 AMLLogicXML.xsd" schemaVersion="1.0"> </pre>

Type	Representation
	<pre> <aml:WriterHeader> <aml:WiterName>ToolA</aml:WiterName> <aml:WiterID>010990003</aml:WiterID> <aml:WriterVender>Vendor1</aml:WriterVender> <aml:WriterVenderURL>https://www.automationml.org/</aml:WriterVenderURL> <aml:WriterVersion>1.00</aml:WriterVersion> <aml:LastWritingDateTime>2002-05-30T09:00:00</aml:LastWritingDateTime> <aml:WriterCulturalSetting>en-US</aml:WriterCulturalSetting> <aml:WriterProjectTitle>SampleXML</aml:WriterProjectTitle> <aml:WriterProjectID>829e1797-dd62-44ac-b29c-c6786d93ca93</aml:WriterProjectID> <aml:AdditionalInformation>This is a sample XML document.</aml:AdditionalInformation> </aml:WriterHeader> <aml:Types> <aml:FunctionBlock name="IEC61131-3_SFC" uuid="e7a28279-952c-465d-850e-cf234a3d27da"> <aml:Parameters> <aml:OutputVars> <aml:Variable name="Output" orderWithinParamSet="1" uuid="69200d1f-1752-4d5c-92ef- cd6fdb697e9f"> <aml:Type> <TypeName>BOOL</TypeName> </aml:Type> </aml:Variable> </aml:OutputVars> </aml:Parameters> <aml:MainBody xsi:type="SFC"> <!--Sub-elements within this element are completely same as IEC 61131-10 SFC.--> <SfcObject xsi:type="Step" name="S0" initialStep="true"> <ConnectionPointOut connectionPointOutId="1"/> </SfcObject> <SfcObject xsi:type="Transition"> <ConnectionPointIn> <Connection refConnectionPointOutId="1"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="2"/> <Condition> <TextualPredicate> <PredicateContent xsi:type="ST"> <ST>TRUE</ST> </PredicateContent> </TextualPredicate> </Condition> </SfcObject> <SfcObject xsi:type="SelectionConvergence"> <ConnectionPointIn> <Connection refConnectionPointOutId="14"/> </ConnectionPointIn> <ConnectionPointIn> <Connection refConnectionPointOutId="2"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="3"/> </SfcObject> <SfcObject xsi:type="Step" name="S1"> <ConnectionPointIn> <Connection refConnectionPointOutId="3"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="5"/> <ConnectionPointOutAction connectionPointOutId="4"/> </SfcObject> <CommonObject xsi:type="ActionBlocks"> <ConnectionPointIn> <Connection refConnectionPointOutId="4"/> </ConnectionPointIn> <ActionBlock> <ActionQualifier qualifier="N"/> <ComplexOperand>Output</ComplexOperand> </ActionBlock> </CommonObject> <SfcObject xsi:type="Transition"> <ConnectionPointIn> <Connection refConnectionPointOutId="5"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="6"/> <Condition> <TextualPredicate> <PredicateContent xsi:type="ST"> </pre>

Type	Representation
	<pre> <ST> <![CDATA[S1.t > T#1s]]> </ST> </PredicateContent> </TextualPredicate> </Condition> </SfcObject> <SfcObject xsi:type="Step" name="S2"> <ConnectionPointIn> <Connection refConnectionPointOutId="6"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="10"/> </SfcObject> <SfcObject xsi:type="Transition"> <ConnectionPointIn> <Connection refConnectionPointOutId="10"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="14"/> </SfcObject> <Condition> <TextualPredicate> <PredicateContent xsi:type="ST"> <ST> <![CDATA[S2.t > T#1s]]> </ST> </PredicateContent> </TextualPredicate> </Condition> </SfcObject> </aml:MainBody> </aml:FunctionBlock> </aml:Types> </aml:AMLLogic> </pre>

A.5 Example for storing function block diagrams

The example given in Table A.13 shows the use of the mapping rules to store function block diagrams in AML logic XML.

Table A.13 – Example for storing a function block diagram

Type	Representation
Function block diagram	The diagram shows a function block diagram on a grid. At the top left, there is a label 'S1 -> S2'. Below it is a TON (Timer On Delay) block labeled 'S1TON'. The TON block has two inputs: 'IN' (labeled 'S1') and 'PT' (labeled 'T#1s'). It has two outputs: 'Q' and 'ET'. The 'Q' output is connected to the input of an 'AND' block. The 'AND' block has one input (labeled 'S1') and one output (labeled 'S1'). The 'ET' output of the TON block is connected to the input of an 'OR' block. The 'OR' block has one input (labeled 'S2') and one output (labeled 'S2').
AML logic XML	<pre> <?xml version="1.0" encoding="utf-8"?> <aml:AMLLogic xmlns="http://www.iec.ch/public/TC65SC65BWG7TF10" xmlns:aml="http://www.automationml.org/IEC62714-4Ed1" xmlns:xsi="http://www.w3.org/2001/XMLSchema- </pre>

Type	Representation
	<pre> instance" xsi:schemaLocation="http://www.automationml.org/IEC62714-4Ed1 AMLLogicXML.xsd" schemaVersion="1.0"> <aml:WriterHeader> <aml:WiterName>ToolA</aml:WiterName> <aml:WiterID>010990003</aml:WiterID> <aml:WriterVender>Vendor1</aml:WriterVender> <aml:WriterVenderURL>https://www.automationml.org/</aml:WriterVenderURL> <aml:WriterVersion>1.00</aml:WriterVersion> <aml:LastWritingDateTime>2002-05-30T09:00:00</aml:LastWritingDateTime> <aml:WriterCulturalSetting>en-US</aml:WriterCulturalSetting> <aml:WriterProjectTitle>SampleXML</aml:WriterProjectTitle> <aml:WriterProjectID>829e1797-dd62-44ac-b29c-c6786d93ca93</aml:WriterProjectID> <aml:AdditionalInformation>This is a sample XML document.</aml:AdditionalInformation> </aml:WriterHeader> <aml:Types> <aml:FunctionBlock uuid="8d5952dc-05fd-4fb4-bbf8-64b6815c1d58" name="IEC61131-3_FBD"> <aml:Vars> <aml:Variable name="S1" uuid="746f9a22-236e-4933-9fd0-dfe66166f39c"> <aml:Type> <TypeName>BOOL</TypeName> </aml:Type> </aml:Variable> <aml:Variable name="S1TON" uuid="da6c6c9b-efb7-4841-b876-c8c88021876e"> <aml:Type> <TypeName>TON</TypeName> </aml:Type> </aml:Variable> <aml:Variable name="S2" uuid="46d7b82a-b7cf-40b0-bb53-92d1477dbce4"> <aml:Type> <TypeName>BOOL</TypeName> </aml:Type> </aml:Variable> </aml:Vars> <aml:MainBody xsi:type="FBD"> <Network xsi:type="FbdNetwork" evaluationOrder="1"> <Documentation xsi:type="SimpleText">S1 -> S2</Documentation> <FbdObject xsi:type="DataSource" identifier="S1"> <ConnectionPointOut connectionPointOutId="1"/> </FbdObject> <FbdObject xsi:type="DataSource" identifier="T#1s"> <ConnectionPointOut connectionPointOutId="2"/> </FbdObject> <FbdObject xsi:type="Block" typeName="TON" instanceName="S1TON"> <InputVariables> <InputVariable parameterName="IN"> <ConnectionPointIn> <Connection refConnectionPointOutId="1"/> </ConnectionPointIn> </InputVariable> <InputVariable parameterName="PT"> <ConnectionPointIn> <Connection refConnectionPointOutId="2"/> </ConnectionPointIn> </InputVariable> </InputVariables> <OutputVariables> <OutputVariable parameterName="Q"> <ConnectionPointOut connectionPointOutId="3"/> </OutputVariable> <OutputVariable parameterName="ET"> <ConnectionPointOut connectionPointOutId="4"/> </OutputVariable> </OutputVariables> </FbdObject> <FbdObject xsi:type="DataSource" identifier="S1"> <ConnectionPointOut connectionPointOutId="5"/> </FbdObject> <FbdObject xsi:type="Block" typeName="AND"> <InputVariables> <InputVariable parameterName="IN1" suppressName="true" negated="true"> <ConnectionPointIn> <Connection refConnectionPointOutId="3"/> </ConnectionPointIn> </InputVariable> <InputVariable parameterName="IN2" suppressName="true"> </pre>

Type	Representation
	<pre> <ConnectionPointIn> <Connection refConnectionPointOutId="5"/> </ConnectionPointIn> </InputVariable> </InputVariables> <OutputVariables> <OutputVariable parameterName="OUT" suppressName="true"> <ConnectionPointOut connectionPointOutId="6"/> </OutputVariable> </OutputVariables> </FbdObject> <FbdObject xsi:type="DataSink" identifier="S1"> <ConnectionPointIn> <Connection refConnectionPointOutId="6"/> </ConnectionPointIn> </FbdObject> <FbdObject xsi:type="DataSource" identifier="S2"> <ConnectionPointOut connectionPointOutId="7"/> </FbdObject> <FbdObject xsi:type="Block" typeName="OR"> <InputVariables> <InputVariable parameterName="IN1" suppressName="true"> <ConnectionPointIn> <Connection refConnectionPointOutId="3"/> </ConnectionPointIn> </InputVariable> <InputVariable parameterName="IN2" suppressName="true"> <ConnectionPointIn> <Connection refConnectionPointOutId="7"/> </ConnectionPointIn> </InputVariable> </InputVariables> <OutputVariables> <OutputVariable parameterName="OUT" suppressName="true"> <ConnectionPointOut connectionPointOutId="8"/> </OutputVariable> </OutputVariables> </FbdObject> <FbdObject xsi:type="DataSink" identifier="S2"> <ConnectionPointIn> <Connection refConnectionPointOutId="8"/> </ConnectionPointIn> </FbdObject> </Network> </aml:MainBody> </aml:FunctionBlock> </aml:Types> </aml:AMLLogic> </pre>

A.6 Example for storing mathematical expressions

The following example shows the use of the mapping rules to store mathematical expressions in AML logic XML.

The example shows the flow rate of the valve. The flow rate depends on the pressure at the input and output side and the adjustment of the adjustment screw. In the supercritical range the flow rate remains constant with a changing input/output pressure ratio. In the subcritical range, the flow rate decreases if the input/output pressure decreases; see Figure A.1.

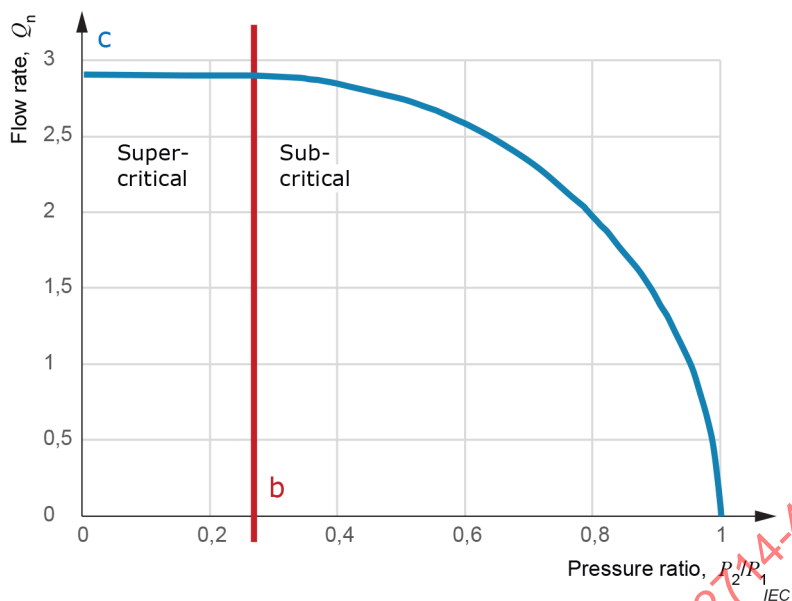


Figure A.1 – Flow rate of valves

The flow rate can be calculated with the following formula:

$$Q = \begin{cases} P_1 * n * C * \sqrt{1 - \left(\frac{\frac{P_2}{P_1} - b}{1 - b} \right)^2}, & \frac{P_2}{P_1} \geq b \text{ (subcritical)} \\ P_1 * n * C, & \frac{P_2}{P_1} < b \text{ (supercritical)} \end{cases} \quad (1)$$

Q = flow rate

P_1 = input absolute static pressure

P_2 = output absolute static pressure

C = sonic conductance

b = critical pressure ratio

n = setting of adjustment screw (0..1)

According to the provisions of Clause 8, mathematical expression can be stored in AML logic XML by expressing them with the content part of MathML version 2.0. The mapping is shown in Figure A.2.

```

<?xml version="1.0" encoding="utf-8"?>
<aml:AMLLogic xmlns="www.iec.ch/public/TC65SC65BWG7TF10" xmlns:aml="http://www.automationml.org/IEC62714-4Ed1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.automationml.org/IEC62714-4Ed1
AMLLogicXML.xsd" schemaVersion="1.0">
  <aml:WriterHeader>
    <aml:WriterName>ToolA</aml:WriterName>
    <aml:WriterID>010990003</aml:WriterID>
    <aml:WriterVender>Vendor1</aml:WriterVender>
    <aml:WriterVenderURL>https://www.automationml.org/</aml:WriterVenderURL>
    <aml:WriterVersion>1.00</aml:WriterVersion>
    <aml:LastWritingDateTime>2002-05-30T09:00:00</aml:LastWritingDateTime>
    <aml:WriterCulturalSetting>en-US</aml:WriterCulturalSetting>
    <aml:WriterProjectTitle>SampleXML</aml:WriterProjectTitle>
    <aml:WriterProjectID>829e1797-dd62-44ac-b29c-c6786d93ca93</aml:WriterProjectID>
    <aml:AdditionalInformation>This is a sample XML document.</aml:AdditionalInformation>
  </aml:WriterHeader>
  <aml:Types>
    <aml:FunctionBlock name="flow rate of a control valve" uuid="74e94dc8-b5a2-490c-bf0f-f0fdaa4515ef">
      <aml:Parameters>
        <aml:InputVars>
          <aml:Variable name="Var2" orderWithinParamSet="2" uuid="028a1a10-5c9f-11e7-9598-0800200c9a66">
            <aml:Type>
              <TypeName>REAL</TypeName>
            </aml:Type>
          </aml:Variable>
          <aml:Variable name="Var3" orderWithinParamSet="3" uuid="fac76284-c20f-4bd1-9c77-a65dff4fcacd">
            <aml:Type>
              <TypeName>REAL</TypeName>
            </aml:Type>
          </aml:Variable>
          <aml:Variable name="Var4" orderWithinParamSet="4" uuid="3d1c1a64-13cf-4fc0-a2bc-e70c2c7dd740">
            <aml:Type>
              <TypeName>REAL</TypeName>
            </aml:Type>
          </aml:Variable>
          <aml:Variable name="Var5" orderWithinParamSet="5" uuid="970b7f93-b3c8-4e4b-b3a6-86acfe533c94">
            <aml:Type>
              <TypeName>REAL</TypeName>
            </aml:Type>
          </aml:Variable>
          <aml:Variable name="Var6" orderWithinParamSet="6" uuid="970b7f93-b3c8-4e4b-b3a6-86acfe533c94">
            <aml:Type>
              <TypeName>REAL</TypeName>
            </aml:Type>
          </aml:Variable>
        </aml:InputVars>
        <aml:OutputVars>
          <aml:Variable name="Var1" orderWithinParamSet="1" uuid="c3c8d294-3e18-4aa5-9a54-b72be85c9eb7">
            <aml:Type>
              <TypeName>REAL</TypeName>
            </aml:Type>
          </aml:Variable>
        </aml:OutputVars>
      </aml:Parameters>
      <aml:MainBody xsi:type="aml:MathematicalExpression" name="flow rate of a control valve" uuid="f59f2dba-442e-4f30-aa52-
6bb83ec6180c">
        <aml:VariableMapping refMathMLVariableName="Q" refFBVariableUUID="c3c8d294-3e18-4aa5-9a54-b72be85c9eb7"
direction="Out"/>
        <aml:VariableMapping refMathMLVariableName="P1" refFBVariableUUID="028a1a10-5c9f-11e7-9598-0800200c9a66"
direction="In"/>
        <aml:VariableMapping refMathMLVariableName="P2" refFBVariableUUID="fac76284-c20f-4bd1-9c77-a65dff4fcacd"
direction="In"/>
        <aml:VariableMapping refMathMLVariableName="C" refFBVariableUUID="3d1c1a64-13cf-4fc0-a2bc-e70c2c7dd740"
direction="In"/>
        <aml:VariableMapping refMathMLVariableName="b" refFBVariableUUID="970b7f93-b3c8-4e4b-b3a6-86acfe533c94"
direction="In"/>
        <aml:VariableMapping refMathMLVariableName="n" refFBVariableUUID="970b7f93-b3c8-4e4b-b3a6-86acfe533c94"
direction="In"/>
        <aml:MathML>
          <math xmlns="http://www.w3.org/1998/Math/MathML" display="block">
            <apply>
              <eq>
                <ci>Q</ci>
                <piecewise>
                  <piece>
                    <apply>
                      <times>
                        <ci>P1</ci>
                    </apply>
                    <times>
                      <ci>n</ci>
                    </times>
                  </piece>
                </piecewise>
              </eq>
            </apply>
          </math>
        </aml:MathML>
      </aml:MainBody>
    </aml:FunctionBlock>
  </aml:Types>
</aml:AMLLogic>

```

</apply>
 <ci>b</ci>
 </apply>
 <apply>
 <minus/>
 <cn>1</cn>
 <ci>b</ci>
 </apply>
 </apply>
 <cn>2</cn>
 </apply>
 </apply>
 </apply>
 </apply>
 </apply>
 </apply>
 </apply>
 </apply>
 <geq/>
 <apply>
 <divide/>
 <ci>P2</ci>
 <ci>P1</ci>
 </apply>
 <ci>b</ci>
 </apply>
 </apply>
 </piece>
 </piece>
 <apply>
 <times/>
 <ci>P1</ci>
 <apply>
 <times/>
 <ci>n</ci>
 <ci>C</ci>
 </apply>
 </apply>
 <apply>
 </it>
 <apply>
 <divide/>
 <ci>P2</ci>
 <ci>P1</ci>
 </apply>
 </apply>
 <ci>b</ci>

Figure A.2 – Example for storing a mathematical expression

Annex B (informative)

Examples for referencing logic information

B.1 General

This annex describes the referencing methods for logic information, which are stored in AML logic XML documents. It comprises the referencing of logic information that is stored within one FunctionBlock, which can be composed of several FunctionBlocks, and the referencing of interlocking information.

Logic information is expressed as logic models. To reference not only the logic model itself (as specified in B.2) but also certain parts of that logic model, e.g. a variable, additional referencing methods are specified in B.3.

NOTE 1 In the CAEX file (see Figure B.2, Figure B.5, Figure B.7, Figure B.9, and Figure B.11), GUIDs are presented in a short form such as "GUID1", "GUID100" etc. This serves the readability and acts as a real GUID.

NOTE 2 The role class "LogicModelObject" is not used in the following examples.

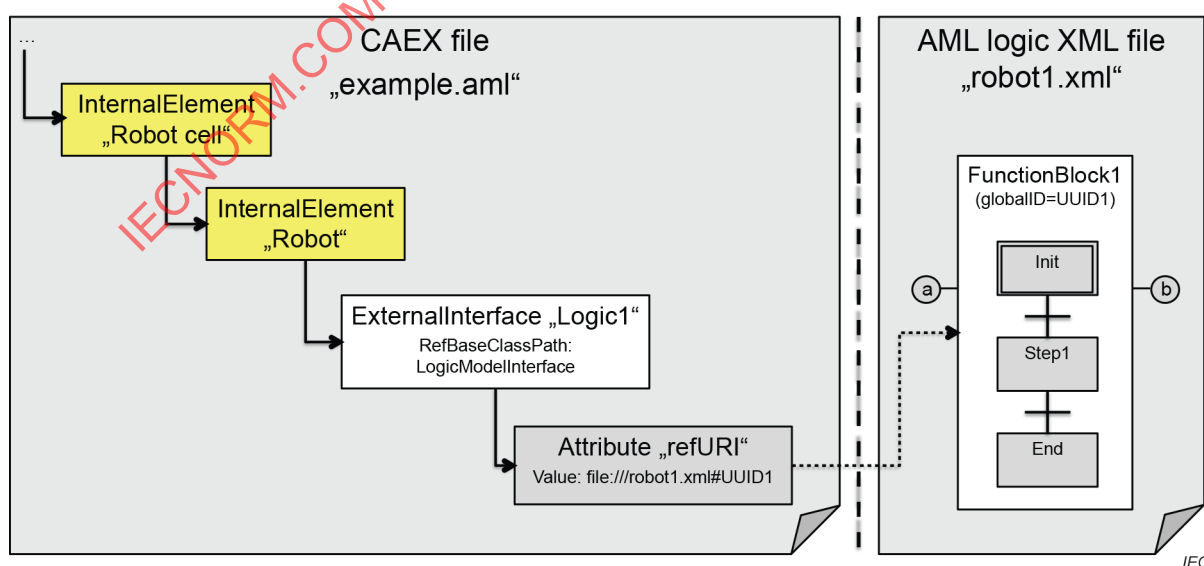
B.2 Referencing logic information expressed as logic models

B.2.1 General

This subclause describes the referencing of logic information expressed as logic models of an AML object as specified in Clause 11.

B.2.2 Referencing logic information stored in one FunctionBlock

Sequencing, behaviour, or interlocking information, stored as a logic model within one FunctionBlock, is referenced by modelling a CAEX ExternalInterface with an AML InterfaceClass "LogicModelInterface" or a derivation of it, associated to it. This is depicted in Figure B.1 and Figure B.2, in which logic information expressed as SFC is referenced.



IEC

Figure B.1 – Referencing logic information (as SFC) stored in one FunctionBlock

```
<InternalElement Name="Robot cell" ID="GUID100">
  <InternalElement Name="Robot" ID="GUID101">
    <ExternalInterface Name="Logic1" RefBaseClassPath="AutomationMLLogicInterfaceClassLib/LogicModelInterface">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>file:///robot1.xml#UUID1</Value>
      </Attribute>
    </ExternalInterface>
  </InternalElement>
</InternalElement>
```

Figure B.2 – XML text of the CAEX file for referencing logic information stored in one FunctionBlock

Logic information can also be expressed as FBD as defined in 5.3. The referencing method remains unaffected.

B.2.3 Referencing logic information, which is composed of several FunctionBlocks

Sequencing, behaviour, or interlocking information, stored as a logic model, which is composed of several FunctionBlocks, is referenced by modelling a CAEX ExternalInterface with an AML InterfaceClass "LogicModelInterface" or a derivation of it.

This is depicted in Figure B.3. Here, a FunctionBlock, which contains a SFC, calls other FunctionBlocks containing SFC or FBD by IEC 61131-10 means. Also a FunctionBlock, which contains a FBD, can call other FunctionBlocks. The referencing method remains unaffected. The XML text of the CAEX file for referencing logic information, which is composed of several FunctionBlocks is the same as depicted in Figure B.2.

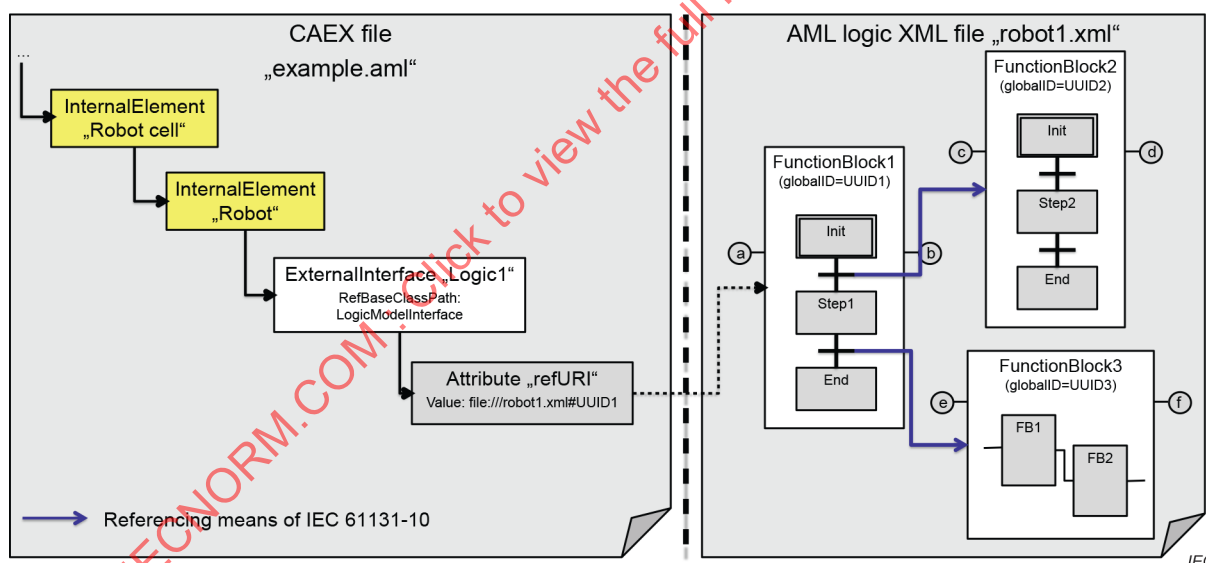


Figure B.3 – Referencing logic information, which is composed of several FunctionBlocks

It is also possible that an AML logic XML document contains several FunctionBlocks which are not interlinked with each other. In that case, the referencing method as specified in B.2.2 is applied.

B.2.4 Referencing logic information, which is composed of several AML logic XML documents

Logic information, stored as logic models, which is composed of several AML logic XML documents, is modelled by a CAEX ExternalInterface for each AML logic XML document resp. FunctionBlock. These CAEX ExternalInterfaces are associated with an AML InterfaceClass "LogicModelInterface", "SequencingLogicModelInterface", "BehaviourLogicModelInterface" or

a derivation of them. This is depicted in Figure B.4 and Figure B.5. Here, the logic information is composed of two AML logic XML documents.

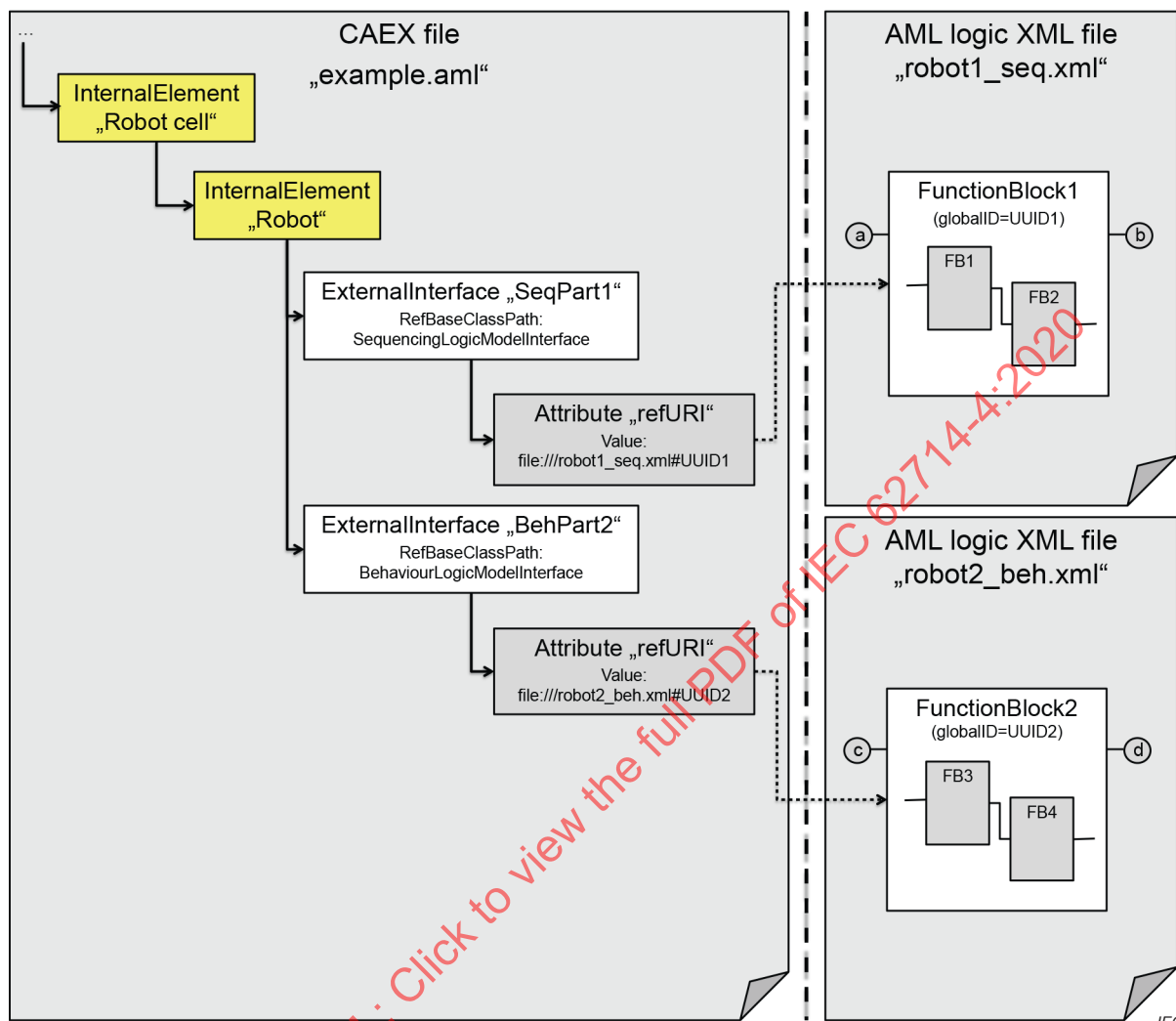


Figure B.4 – Referencing logic information which is composed of several AML logic XML documents

```
<InternalElement Name="Robot cell" ID="GUID100">
  <InternalElement Name="Robot" ID="GUID101">
    <ExternalInterface Name="SeqPart1"
      RefBaseClassPath="AutomationMLLogicInterfaceClassLib/SequencingLogicModelInterface">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>file:///robot1_seq.xml#UUID1</Value>
      </Attribute>
    </ExternalInterface>
    <ExternalInterface Name="BehPart2"
      RefBaseClassPath="AutomationMLLogicInterfaceClassLib/BehaviourLogicModelInterface">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>file:///robot2_beh.xml#UUID2</Value>
      </Attribute>
    </ExternalInterface>
  </InternalElement>
</InternalElement>
```

Figure B.5 – XML text of the CAEX file for referencing logic information, which is composed of several AML logic XML documents

A logic model is always entirely stored in one AML logic XML document.

B.3 Referencing logic information as a part of logic models

B.3.1 General

This subclause describes the referencing of logic information as a part of logic models of an AML object as specified in Clause 11.

B.3.2 Referencing a variable

A variable in a logic model, containing sequencing or behaviour information, is referenced by modelling a CAEX ExternalInterface of the AML InterfaceClass "VariableInterface". This is depicted in Figure B.6 and Figure B.7, in which a variable "b" is referenced. But "FunctionBlock1", which defines the variable "b", needs to be referenced as described in B.2 using an InternalLink.

NOTE Variables can be signals, parameters, internal variables etc.

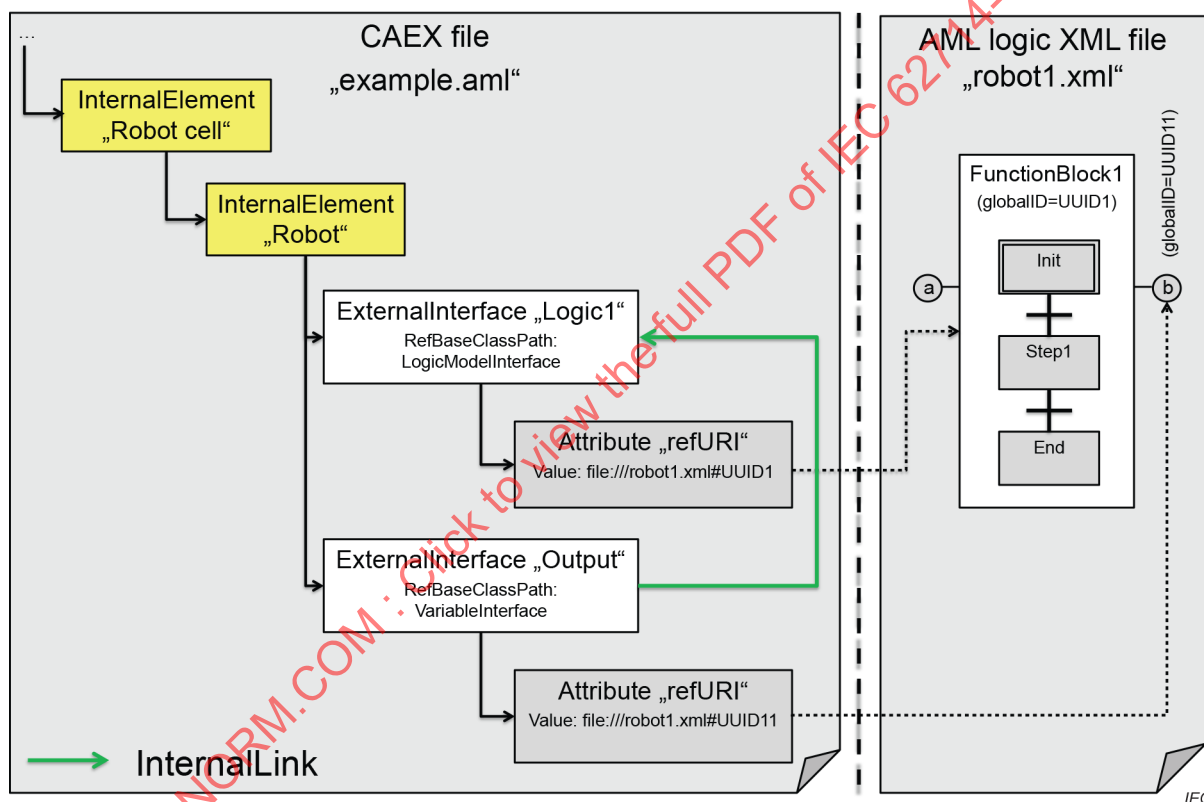


Figure B.6 – Referencing a variable

```
<InternalElement Name="Robot cell" ID="GUID100">
  <InternalElement Name="Robot" ID="GUID101">
    <ExternalInterface Name="Logic1" RefBaseClassPath="AutomationMLLogicInterfaceClassLib/LogicModelInterface">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>file:///robot1.xml#UUID1</Value>
      </Attribute>
    </ExternalInterface>
    <ExternalInterface Name="Output" RefBaseClassPath="AutomationMLLogicInterfaceClassLib/VariableInterface">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>file:///robot1.xml#UUID11</Value>
      </Attribute>
      <InternalLink Name="Link_VariableToLogic" RefPartnerSideA="GUID101:Output" RefPartnerSideB="GUID101:Logic1" />
    </ExternalInterface>
  </InternalElement>
</InternalElement>
```

Figure B.7 – XML text of the CAEX file for referencing a variable

B.3.3 Referencing a logic element

A logic element in a logic model, containing sequencing or behaviour information, is referenced by modelling a CAEX ExternalInterfaces of the AML InterfaceClass "LogicModelElementInterface" or a derivation of it. This is depicted in Figure B.8 and Figure B.9, where the step "Step1" is referenced. But the logic model (stored in "FunctionBlock1") of which the logic element is a part of needs to be referenced as described in Clause B.2 using an InternalLink.

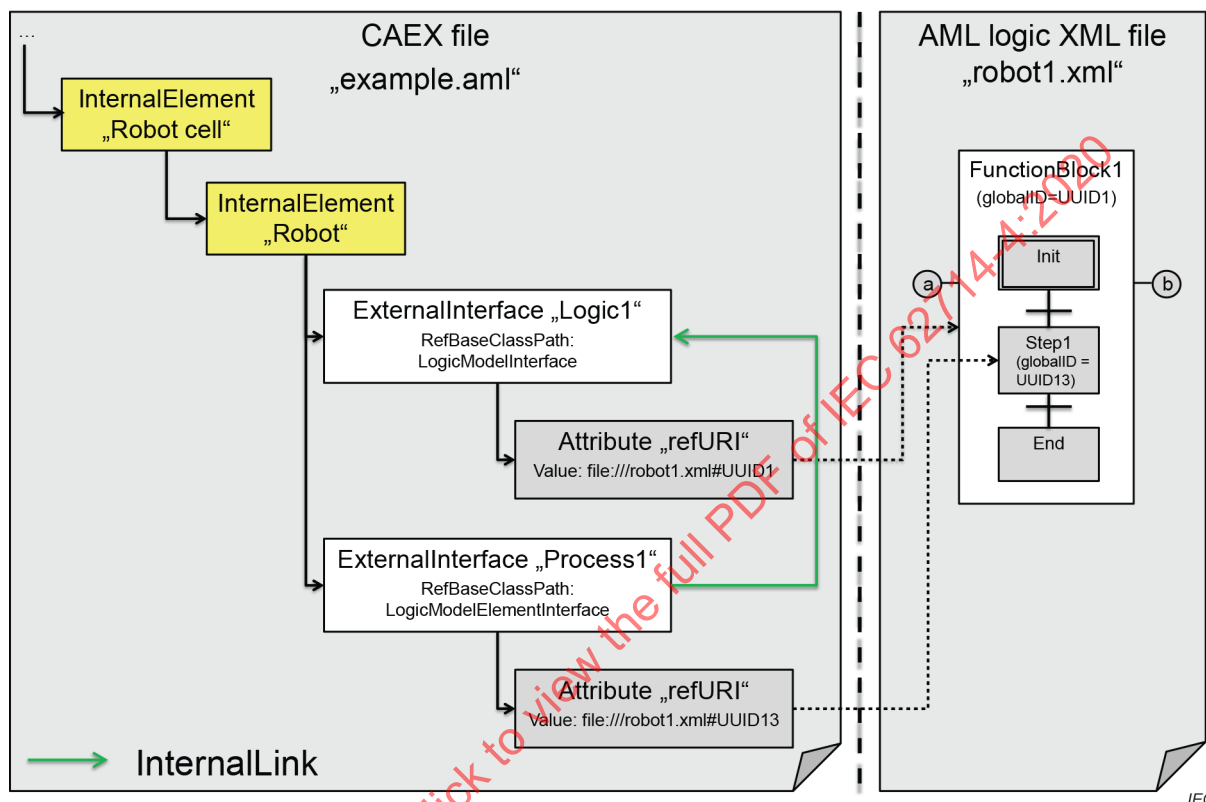


Figure B.8 – Referencing a logic element

```
<InternalElement Name="Robot cell" ID="GUID100">
  <InternalElement Name="Robot" ID="GUID101">
    <ExternalInterface Name="Logic1" RefBaseClassPath="AutomationMLLogicInterfaceClassLib/LogicModelInterface">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>file:///robot1.xml#UUID1</Value>
      </Attribute>
    </ExternalInterface>
    <ExternalInterface Name="Process1"
      RefBaseClassPath="AutomationMLLogicInterfaceClassLib/LogicModelElementInterface">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>file:///robot1.xml#UUID13</Value>
      </Attribute>
      <InternalLink Name="Link_ProcessToLogic" RefPartnerSideA="GUID101:Process1" RefPartnerSideB="GUID101:Logic1"
    </InternalLink>
    </ExternalInterface>
  </InternalElement>
</InternalElement>
```

Figure B.9 – XML text of the CAEX file for referencing a logic element

B.4 Referencing logic information as a part of already referenced logic models

Parts of a logic model, containing sequencing, behaviour, or interlocking information, are referenced from AML objects as described in Clause B.3. But if this logic model is already referenced by another AML objects, it is not necessary to reference this logic model again. Instead a prompt, referencing of the parts of this logic model is possible. This is depicted in

Figure B.10 and Figure B.11, in which a variable "b" of an already referenced logic model is referenced.

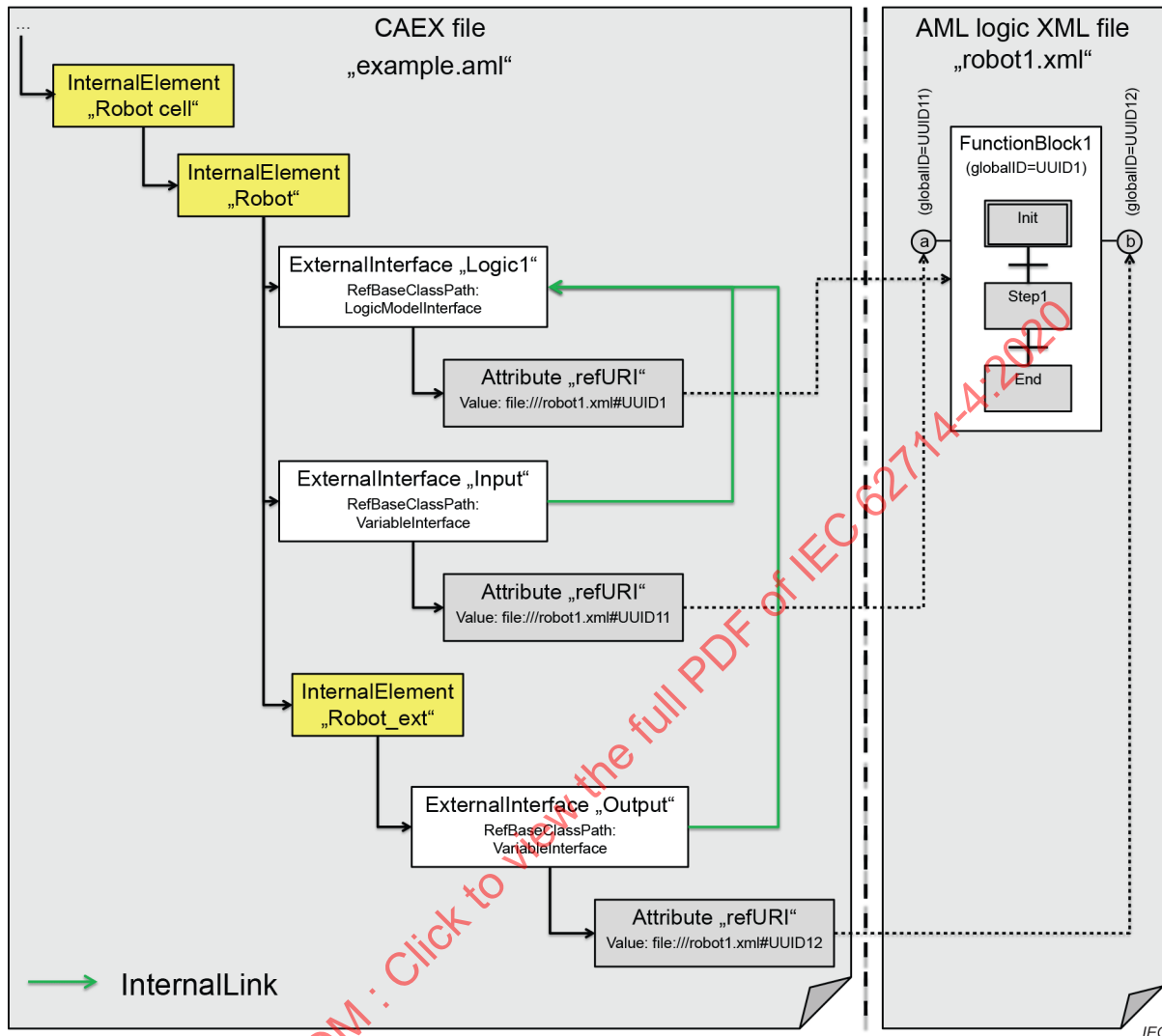


Figure B.10 – Referencing a variable of an already referenced logic model

```
<InternalElement Name="Robot cell" ID="GUID100">
  <InternalElement Name="Robot" ID="GUID101">
    <ExternalInterface Name="Logic1" RefBaseClassPath="AutomationMLLogicInterfaceClassLib/LogicModelInterface">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>file:///robot1.xml#UUID1</Value>
      </Attribute>
    </ExternalInterface>
    <ExternalInterface Name="Input" RefBaseClassPath="AutomationMLLogicInterfaceClassLib/VariableInterface">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>file:///robot1.xml#UUID11</Value>
      </Attribute>
    </ExternalInterface>
    <InternalElement Name="Robot_ext" ID="GUID102">
      <ExternalInterface Name="Output" RefBaseClassPath="AutomationMLLogicInterfaceClassLib/VariableInterface">
        <Attribute Name="refURI" AttributeDataType="xs:anyURI">
          <Value>file:///robot1.xml#UUID12</Value>
        </Attribute>
      </ExternalInterface>
      <InternalLink Name="Link_OutputToLogic1" RefPartnerSideA="GUID102:Output" RefPartnerSideB="GUID101:Logic1" />
      <InternalLink Name="Link_InputToLogic1" RefPartnerSideA="GUID101:Input" RefPartnerSideB="GUID101:Logic1" />
    </InternalElement>
  </InternalElement>
</InternalElement>
```

Figure B.11 – XML text of the CAEX file for referencing a variable of an already referenced logic model

Annex C (informative)

Examples for referencing interlocking information

C.1 General

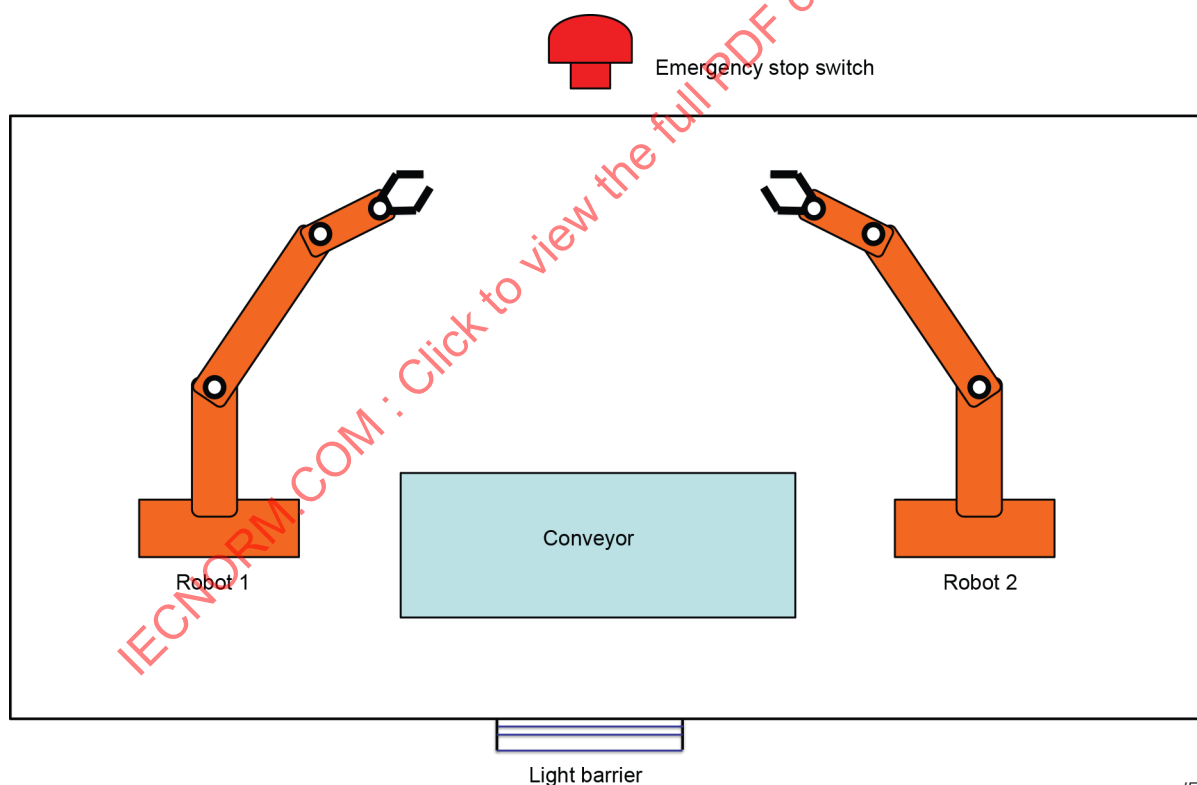
This annex describes the two mechanisms for referencing interlocking information, which is stored in AML logic XML documents. This comprises the modelling and storage of interlocking information without and with an explicitly modelled interlocking condition. Both are explained consecutively using one and the same example.

Normative provisions are given in Clause 12.

NOTE 1 In the CAEX file (see Figure C.4, Figure C.6, and Figure C.8) GUIDs are presented in a short form such as "GUID1", "GUID100" etc. This serves the readability and acts as a real GUID.

NOTE 2 The role class "LogicModelObject" is not used in the following examples.

The example depicts a manufacturing system containing different manufacturing equipment and safety devices (see Figure C.1).



IEC

Figure C.1 – Example manufacturing system

The robot cell contains the objects respectively manufacturing resources "Robot 1", "Robot 2" and "Conveyor". It also contains the safety devices "Light barrier" and "Emergency stop switch". Figure C.2 shows the exemplary aggregation of the manufacturing resources into an interlocking source group and an interlocking target group. It is allowed to have more than one interlocking source group and interlocking target group. In this example, there is one of each.

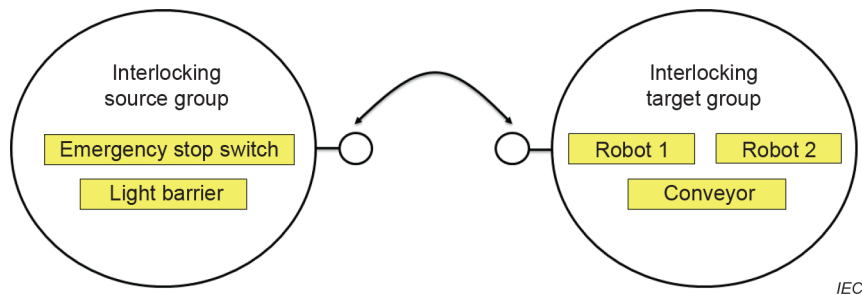


Figure C.2 – Example interlocking source group and interlocking target group

C.2 Interlocking information

The third, complementary concept of logic information covered by AML is the interlocking information expressed on two different levels of detail belonging to an industrial production system or to single components. It reflects the two main concepts:

- The causes of an interlocking condition, and
- The resulting effects of an interlocking condition.

Hereby, the cause represents the situation resulting in the need to interlock something. This case is covered by the definition and description of interlocking source groups. The effect to other groups of objects is expressed by the definition, description of and assignment to interlocking target groups.

These relations between interlocking source group and interlocking target group express the relation between cause and effect within interlocking information. In AML different levels of detail are used to exchange these interlocking conditions. Therefore, AML provides two mechanisms to reference interlocking information.

To describe interlocking source groups and interlocking target groups the basic AML group concept is used. This concept exploits fundamental CAEX capabilities and enables an integration of the necessary information within the AML top-level documents (see Clause C.3).

To express the relation between interlocking source group and interlocking target group and to express the internal logical relation within the groups, Function Block Diagrams (FBD) of the IEC 61131-3 are used (see Clause C.4). They are stored in AML logic XML documents similar to sequencing and behaviour information.

NOTE A detailed specification of cause and effect tables can be found in IEC 62881. This can be a possible implementation for an interlocking logic model.

C.3 Referencing interlocking information without interlocking condition

On the first level of detail, functional dependencies among groups of objects are modelled – without explicitly modelling an interlocking condition that would further specify that functional dependency. Objects that indicate unsafe states within the manufacturing system are aggregated to the interlocking source group. These are the "Light barrier" and "Emergency stop switch". Objects that are influenced in their behaviour by the state of the interlocking source group are aggregated to the interlocking target group. These are the "Robot 1", "Robot 2", and "Conveyor". They then need to execute specific activities to re-establish safe system behaviour, e.g. stop any movement of the objects of the interlocking target group.

For each group, an additional CAEX InternalElement, with either an AML RoleClass "InterlockingSourceGroup" or "InterlockingTargetGroup", is modelled to separate structure information from instance information. Each group contains a CAEX ExternalInterface of the AML InterfaceClass "InterlockingConnector" to relate the interlocking source group and interlocking target group to each other. CAEX Objects which belong to one of those groups are modelled as mirror objects (see Figure C.3 and Figure C.4).

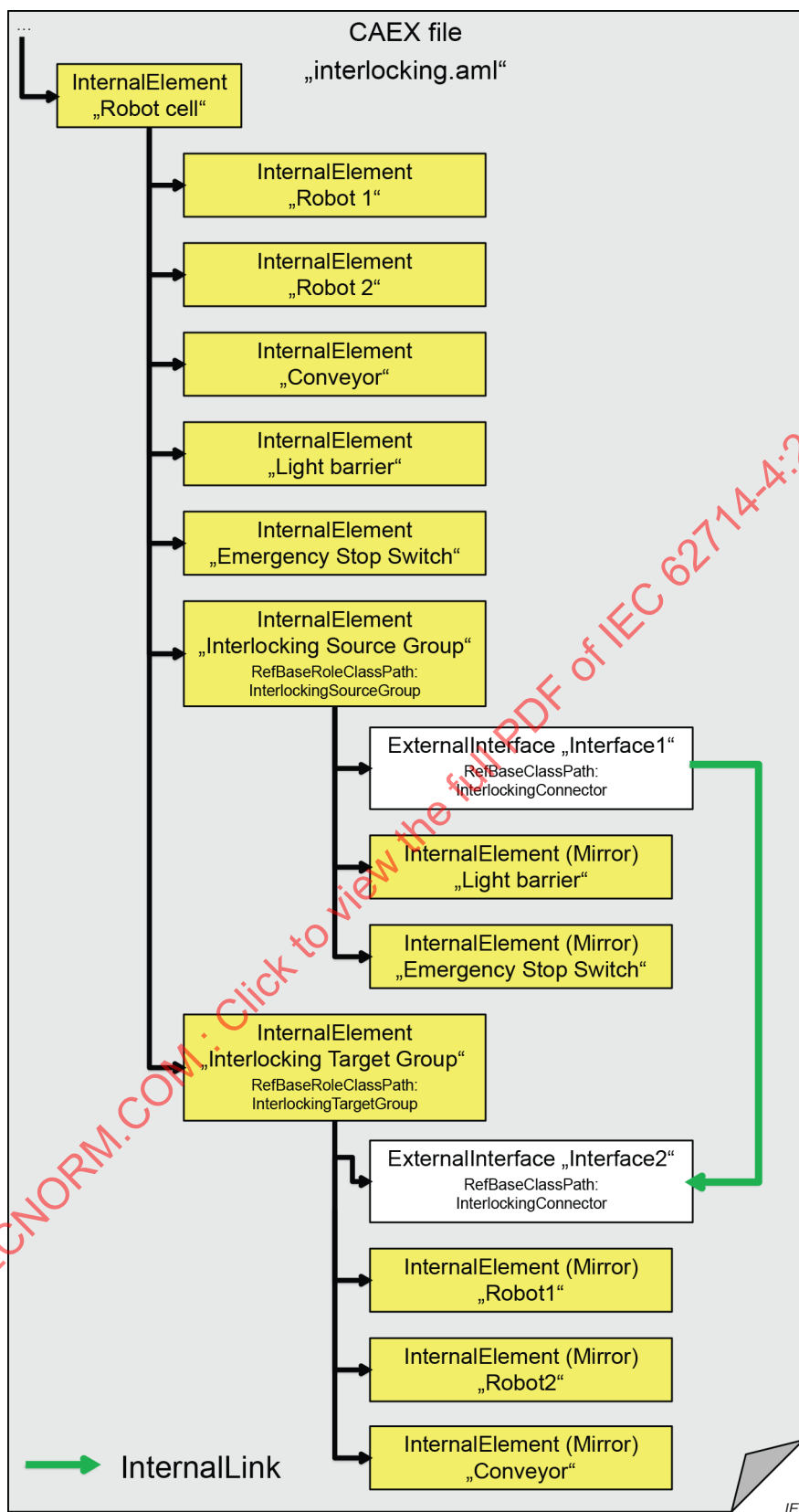


Figure C.3 – Referencing interlocking information without interlocking condition

```

<InternalElement Name="Robot cell" ID="GUID200">
  <InternalElement Name="Robot 1" ID="GUID201" />
  <InternalElement Name="Robot 2" ID="GUID202" />
  <InternalElement Name="Conveyor" ID="GUID203" />
  <InternalElement Name="Light barrier" ID="GUID204" />
  <InternalElement Name="Emergency Stop Switch" ID="GUID205" />
  <InternalElement Name="Interlocking Source Group" ID="GUID206">
    <ExternalInterface Name="Interface1"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/InterlockingConnector" />
    <InternalElement Name="Light barrier" ID="GUID207" RefBaseSystemUnitPath="GUID204" />
    <InternalElement Name="Emergency Stop Switch" ID="GUID208" RefBaseSystemUnitPath="GUID205" />
    <RoleRequirements RefBaseRoleClassPath="AutomationMLLogicRoleClassLib/InterlockingSourceGroup" />
  </InternalElement>
  <InternalElement Name="Interlocking Target Group" ID="GUID209">
    <ExternalInterface Name="Interface2"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/InterlockingConnector" />
    <InternalElement Name="Robot 1" ID="GUID210" RefBaseSystemUnitPath="GUID201" />
    <InternalElement Name="Robot 2" ID="GUID211" RefBaseSystemUnitPath="GUID202" />
    <InternalElement Name="Conveyor" ID="GUID212" RefBaseSystemUnitPath="GUID203" />
    <RoleRequirements RefBaseRoleClassPath="AutomationMLLogicRoleClassLib/InterlockingTargetGroup" />
  </InternalElement>
  <InternalLink Name="Link_InterlockingGroups" RefPartnerSideA="GUID206:Interface1"
RefPartnerSideB="GUID209:Interface2" />
</InternalElement>

```

Figure C.4 – XML text of the CAEX file for referencing interlocking information without interlocking condition

C.4 Referencing interlocking information with interlocking condition

On the second level of detail, a function describing the interlocking condition (modelled as FBD) is associated to the interlocking source group. This level extends the first level of detail described in Clause C.3.

Within the interlocking source group, two additional CAEX ExternalInterfaces are modelled: one with an AML InterfaceClass "InterlockingLogicModelInterface" following the rules of Clause 11 and one CAEX ExternalInterface with an AML InterfaceClass "InterlockingVariableInterface". This references the unique Boolean variable that describes the output or the evaluation result of the interlocking information (see variable "z" in Figure C.5). And this result represents, whether the manufacturing system is in a safe state or not, indicated by the value of the attribute "SafeConditionEquals".

The interlocking information also needs input variables of the objects of the interlocking source group respectively the safety devices (see variable "a" and "b" in Figure C.5 and Figure C.6). For this, "Light barrier" and "Emergency stop switch" have a CAEX ExternalInterface of the AML InterfacesClass "VariableInterface".

To link the logical interface "VariableInterface" to the physical interface of the safety device, a CAEX InternalLink is used. Figure C.7 and Figure C.8 depict this exemplarily for the light barrier.

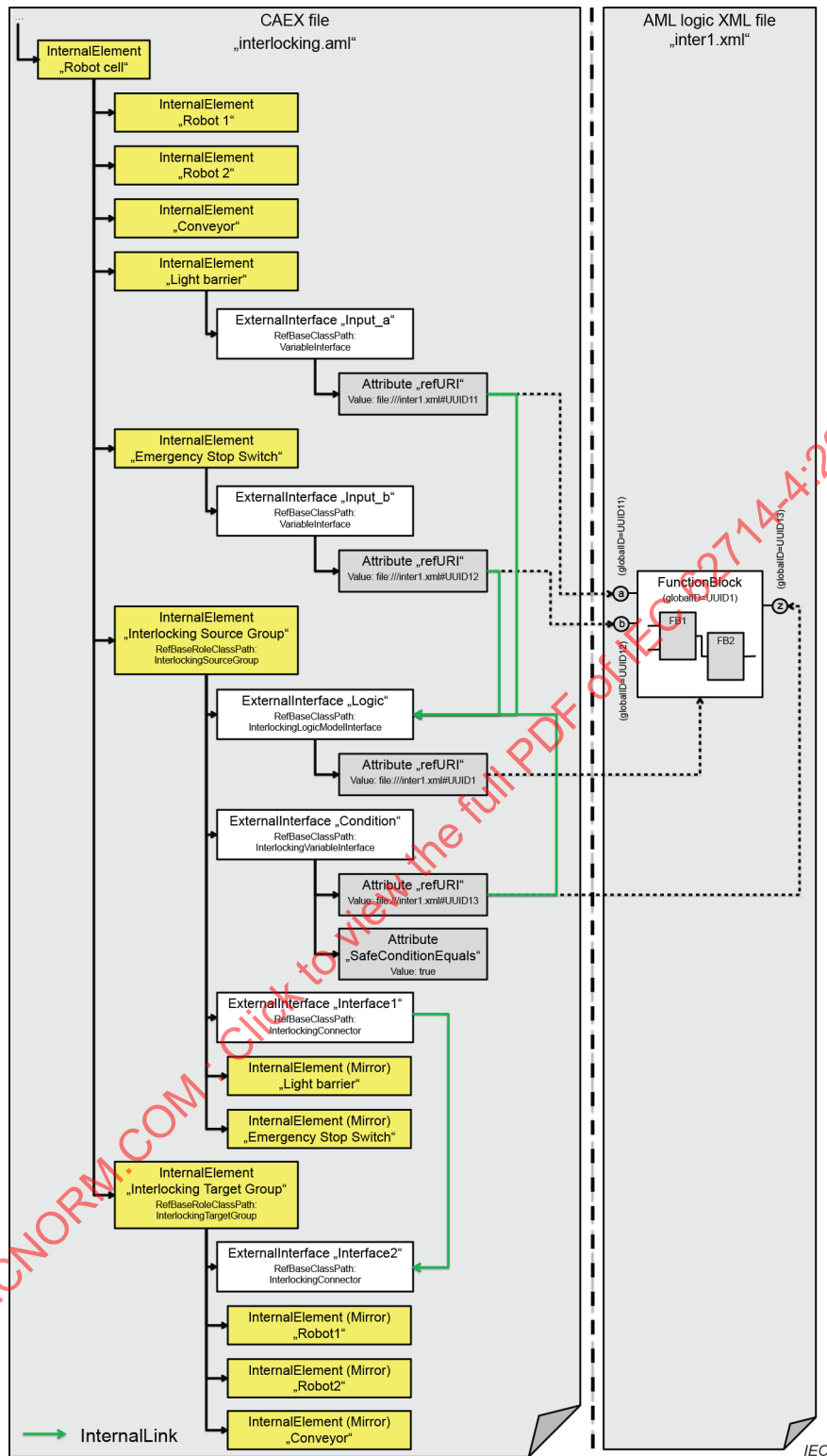


Figure C.5 – Referencing interlocking information with interlocking condition

```

<InternalElement Name="Robot cell" ID="GUID200">
  <InternalElement Name="Robot 1" ID="GUID201" />
  <InternalElement Name="Robot 2" ID="GUID202" />
  <InternalElement Name="Conveyor" ID="GUID203" />
  <InternalElement Name="Light barrier" ID="GUID204">
    <ExternalInterface Name="Input_a" RefBaseClassPath="AutomationMLLogicInterfaceClassLib/VariableInterface">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>file:///inter1.xml#UUID11</Value>
      </Attribute>
    </ExternalInterface>
  </InternalElement>
  <InternalElement Name="Emergency Stop Switch" ID="GUID205">
    <ExternalInterface Name="Input_b" RefBaseClassPath="AutomationMLLogicInterfaceClassLib/VariableInterface">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>file:///inter1.xml#UUID12</Value>
      </Attribute>
    </ExternalInterface>
  </InternalElement>
  <InternalElement Name="Interlocking Source Group" ID="GUID206">
    <ExternalInterface Name="Logic"
RefBaseClassPath="AutomationMLLogicInterfaceClassLib/InterlockingLogicModelInterface">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>file:///inter1.xml#UUID1</Value>
      </Attribute>
    </ExternalInterface>
    <ExternalInterface Name="Condition"
RefBaseClassPath="AutomationMLLogicInterfaceClassLib/InterlockingVariableInterface">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>file:///inter1.xml#UUID13</Value>
      </Attribute>
      <Attribute Name="SafeConditionEquals" AttributeDataType="xs:boolean">
        <DefaultValue>true</DefaultValue>
        <Value>true</Value>
      </Attribute>
    </ExternalInterface>
    <ExternalInterface Name="Interface1"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/InterlockingConnector" />
    <InternalElement Name="Light barrier" ID="GUID207" RefBaseSystemUnitPath="GUID204" />
    <InternalElement Name="Emergency Stop Switch" ID="GUID208" RefBaseSystemUnitPath="GUID205" />
    <InternalLink Name="Link_FunctionBlockOutputToFunctionBlock" RefPartnerSideA="GUID206:Condition"
RefPartnerSideB="GUID206:Logic" />
    <RoleRequirements RefBaseRoleClassPath="AutomationMLLogicRoleClassLib/InterlockingSourceGroup" />
  </InternalElement>
  <InternalElement Name="Interlocking Target Group" ID="GUID209">
    <ExternalInterface Name="Interface2"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/InterlockingConnector" />
    <InternalElement Name="Robot 1" ID="GUID210" RefBaseSystemUnitPath="GUID201" />
    <InternalElement Name="Robot 2" ID="GUID211" RefBaseSystemUnitPath="GUID202" />
    <InternalElement Name="Conveyor" ID="GUID212" RefBaseSystemUnitPath="GUID203" />
    <RoleRequirements RefBaseRoleClassPath="AutomationMLLogicRoleClassLib/InterlockingTargetGroup" />
  </InternalElement>
  <InternalLink Name="Link_InterlockingGroups" RefPartnerSideA="GUID206:Interface1"
RefPartnerSideB="GUID209:Interface2" />
  <InternalLink Name="Link_LightBarrierToFunctionBlock" RefPartnerSideA="GUID204:Input_a"
RefPartnerSideB="GUID206:Logic" />
  <InternalLink Name="Link_EmergencyStopToFunctionBlock" RefPartnerSideA="GUID205:Input_b"
RefPartnerSideB="GUID206:Logic" />
</InternalElement>

```

Figure C.6 – XML text of the CAEX file for referencing interlocking information with interlocking condition

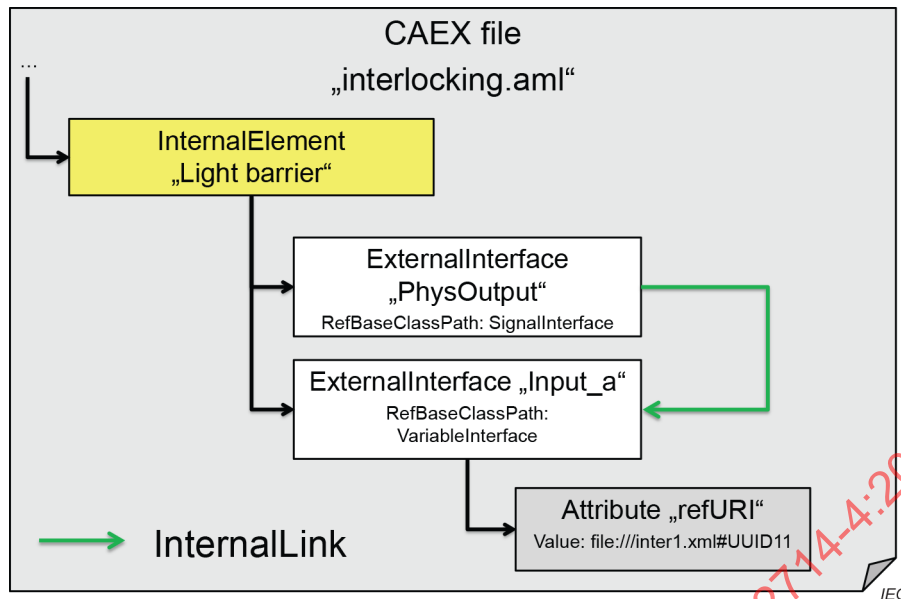


Figure C.7 – Linking logical interface with physical interface (extension to Figure C.5)

```
<InternalElement Name="Light barrier" ID="GUID204">
  <ExternalInterface Name="PhysOutput"
    RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Communication/SignalInterface" />
  <ExternalInterface Name="Input_a" RefBaseClassPath="AutomationMLLogicInterfaceClassLib/VariableInterface">
    <Attribute Name="refURI" AttributeDataType="xs:anyURI">
      <Value>file:///inter1.xml#UUID11</Value>
    </Attribute>
  </ExternalInterface>
  <InternalLink Name="Link1-LinkLogicPhysicalInterface" RefPartnerSideA="GUID204:PhysOutput"
    RefPartnerSideB="GUID204:Input_a" />
</InternalElement>
```

Figure C.8 – XML text of the CAEX file for linking logical interface with physical interface (extension to Figure C.6)

Annex D (normative)

XML representation of AML standard libraries

D.1 General

Annex D gives a normative XML representation of the libraries defined in this document.

D.2 AutomationMLLogicRoleClassLib

```
<?xml version="1.0" encoding="utf-8"?>
<CAEXFile FileName="AutomationMLLogicRoleClassLib.aml" SchemaVersion="2.15"
xsi:noNamespaceSchemaLocation="CAEX_ClassModel_V2.15.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <AdditionalInformation>
    <WriterHeader>
      <WriterName>AutomationML Editor</WriterName>
      <WriterID>916578CA-FE0D-474E-A4FC-9E1719892369</WriterID>
      <WriterVendor>AutomationML e.V.</WriterVendor>
      <WriterVendorURL>www.AutomationML.org</WriterVendorURL>
      <WriterVersion>4.7.0.0</WriterVersion>
      <WriterRelease>4.7.0.0</WriterRelease>
      <LastWritingDateTime>2018-06-03T20:57:19.7275022+02:00</LastWritingDateTime>
      <WriterProjectTitle>unspecified</WriterProjectTitle>
      <WriterProjectID>unspecified</WriterProjectID>
    </WriterHeader>
  </AdditionalInformation>
  <AdditionalInformation AutomationMLVersion="2.0"/>
  <RoleClassLib Name="AutomationMLLogicRoleClassLib">
    <Description>AutomationMLLogicRoleClassLib specifies all logic related role classes.</Description>
    <Version>1.0.0</Version>
    <RoleClass Name="InterlockingTargetGroup"
RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Group">
      <ExternalInterface Name="ConnectionInterlockingSourceGroup" ID="1d3bfa87-5538-40e2-b36a-6b31c76208db"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/InterlockingConnector"/>
      <ExternalInterface Name="ReferenceInterlockingLogicModel" ID="b1fd5920-a553-4e68-b571-c672bf00592a"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector/AMLLogicXMLInter
face/LogicModelInterface/InterlockingLogicModelInterface">
        <Attribute Name="refURI" AttributeDataType="xs:anyURI"/>
      </ExternalInterface>
    </RoleClass>
    <RoleClass Name="InterlockingSourceGroup"
RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Group">
      <ExternalInterface Name="ConnectionInterlockingTargetGroup" ID="3cc169e6-1e34-4d50-907f-df361c9a6d03"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/InterlockingConnector"/>
      <ExternalInterface Name="ReferenceInterlockingLogicModel" ID="59fcfe8e-7531-4f35-9b13-bd1eb9fecc88"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector/AMLLogicXMLInter
face/LogicModelInterface/InterlockingLogicModelInterface">
        <Attribute Name="refURI" AttributeDataType="xs:anyURI"/>
      </ExternalInterface>
      <ExternalInterface Name="ReferenceInterlockingVariable" ID="09546c38-cca7-4eec-85b1-a5f33f11f8bd"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector/AMLLogicXMLInter
face/LogicModelElementInterface/VariableInterface/InterlockingVariableInterface">
        <Attribute Name="refURI" AttributeDataType="xs:anyURI"/>
        <Attribute Name="Direction" AttributeDataType="xs:string"/>
        <Attribute Name="SafeConditionEquals" AttributeDataType="xs:boolean"/>
      </ExternalInterface>
    </RoleClass>
    <RoleClass Name="LogicModelObject"
RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Group">
      <ExternalInterface Name="ReferenceSequence" ID="dbbb8615-3003-43ea-81e5-5b78e0654a79"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector/AMLLogicXMLInter
face/LogicModelInterface/SequencingLogicModelInterface">
        <Attribute Name="refURI" AttributeDataType="xs:anyURI"/>
      </ExternalInterface>
      <ExternalInterface Name="ReferenceBehaviour" ID="b34a2b98-dd40-4d94-b5ad-110d46729650"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector/AMLLogicXMLInter
face/LogicModelInterface/BehaviourLogicModelInterface">
        <Attribute Name="refURI" AttributeDataType="xs:anyURI"/>
      </ExternalInterface>
    </RoleClass>
  </RoleClassLib>
</CAEXFile>
```

```

</RoleClass>
</RoleClassLib>
</CAEXFile>

```

D.3 AutomationMLLogicInterfaceClassLib

```

<?xml version="1.0" encoding="utf-8"?>
<CAEXFile FileName="AutomationMLLogicInterfaceClassLib.xml" SchemaVersion="2.15"
xmlns:noNamespaceSchemaLocation="CAEX_ClassModel_V2.15.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <AdditionalInformation>
    <WriterHeader>
      <WriterName>AutomationML Editor</WriterName>
      <WriterID>916578CA-FE0D-474E-A4FC-9E1719892369</WriterID>
      <WriterVendor>AutomationML e.V.</WriterVendor>
      <WriterVendorURL>www.AutomationML.org</WriterVendorURL>
      <WriterVersion>4.7.0.0</WriterVersion>
      <WriterRelease>4.7.0.0</WriterRelease>
      <LastWritingDateTime>2018-06-03T20:57:19.7275022+02:00</LastWritingDateTime>
      <WriterProjectTitle>unspecified</WriterProjectTitle>
      <WriterProjectID>unspecified</WriterProjectID>
    </WriterHeader>
  </AdditionalInformation>
  <AdditionalInformation AutomationMLVersion="2.0"/>
  <InterfaceClassLib Name="AutomationMLLogicInterfaceClassLib">
    <Description>AutomationMLLogicInterfaceClassLib specifies all logic related interface classes for referencing AML logic
XML documents.</Description>
    <Version>1.0.0</Version>
    <InterfaceClass Name="LogicModelInterface"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector">
    </InterfaceClass>
    <InterfaceClass Name="SequencingLogicModelInterface"
RefBaseClassPath="AutomationMLLogicInterfaceClassLib/LogicModelInterface"/>
    <InterfaceClass Name="BehaviourLogicModelInterface"
RefBaseClassPath="AutomationMLLogicInterfaceClassLib/LogicModelInterface"/>
    <InterfaceClass Name="InterlockingLogicModelInterface"
RefBaseClassPath="AutomationMLLogicInterfaceClassLib/LogicModelInterface"/>
    <InterfaceClass Name="LogicModelElementInterface"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector">
    </InterfaceClass>
    <InterfaceClass Name="VariableInterface"
RefBaseClassPath="AutomationMLLogicInterfaceClassLib/LogicModelElementInterface">
      <Attribute Name="Direction" AttributeDataType="xs:string"/>
    </InterfaceClass>
    <InterfaceClass Name="InterlockingVariableInterface"
RefBaseClassPath="AutomationMLLogicInterfaceClassLib/VariableInterface">
      <Attribute Name="SafeConditionEquals" AttributeDataType="xs:boolean"/>
    </InterfaceClass>
  </InterfaceClassLib>
</CAEXFile>

```

D.4 AutomationMLPLCopenXMLInterfaceClassLib

```

<?xml version="1.0" encoding="utf-8"?>
<CAEXFile FileName="AutomationMLPLCopenXMLInterfaceClassLib.xml" SchemaVersion="2.15"
xsi:noNamespaceSchemaLocation="CAEX_ClassModel_V2.15.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <AdditionalInformation>
    <WriterHeader>
      <WriterName>AutomationML Editor</WriterName>
      <WriterID>916578CA-FE0D-474E-A4FC-9E1719892369</WriterID>
      <WriterVendor>AutomationML e.V.</WriterVendor>
      <WriterVendorURL>www.AutomationML.org</WriterVendorURL>
      <WriterVersion>4.7.0.0</WriterVersion>
      <WriterRelease>4.7.0.0</WriterRelease>
      <LastWritingDateTime>2018-06-03T20:57:19.7275022+02:00</LastWritingDateTime>
      <WriterProjectTitle>unspecified</WriterProjectTitle>
      <WriterProjectID>unspecified</WriterProjectID>
    </WriterHeader>
  </AdditionalInformation>
  <AdditionalInformation AutomationMLVersion="2.0"/>
  <InterfaceClassLib Name="AutomationMLPLCopenXMLInterfaceClassLib">
    <Description>AutomationMLPLCopenXMLInterfaceClassLib specifies the logic related interface class for referencing
    IEC 61131-10 documents</Description>
    <Version>1.0.0</Version>
    <InterfaceClass Name="VariableInterface"
    RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector/PLCopenXMLInterf
    ace" />
  </InterfaceClassLib>
</CAEXFile>

```

Annex E (normative)

XML representation of AML logic XML schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:aml="http://www.automationml.org/IEC62714-4Ed1" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xhtml="http://www.w3.org/1999/xhtml" xmlns:ppx="http://www.iec.ch/public/TC65SC65BWG7TF10"
  targetNamespace="http://www.automationml.org/IEC62714-4Ed1" elementFormDefault="qualified"
  attributeFormDefault="unqualified" version="1.0">
  <xsd:import namespace="http://www.iec.ch/public/TC65SC65BWG7TF10" schemaLocation="IEC61131_10_Ed1_0.xsd"/>
  <xsd:annotation>
    <xsd:documentation>This is an example for AutomationML IML to extend IEC 61131-10 to represent vendor-specific
    extension.</xsd:documentation>
  </xsd:annotation>
  <xsd:element name="AMLLogic">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="WriterHeader" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="WiterName" type="xsd:string"/>
              <xsd:element name="WiterID" type="xsd:string"/>
              <xsd:element name="WriterVender" type="xsd:string"/>
              <xsd:element name="WriterVenderURL" type="xsd:string"/>
              <xsd:element name="WriterVersion" type="xsd:string"/>
              <xsd:element name="LastWritingDateTime" type="xsd:dateTime"/>
              <xsd:element name="WriterCulturalSetting" type="xsd:string" minOccurs="0"/>
              <xsd:element name="WriterProjectTitle" type="xsd:string" minOccurs="0"/>
              <xsd:element name="WriterProjectID" type="xsd:string" minOccurs="0"/>
              <xsd:element name="AdditionalInformation" type="xsd:string" minOccurs="0"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="Types">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element name="DataTypeDecl" type="ppx:UserDefinedTypeDecl"/>
                <xsd:element name="FunctionBlock" type="aml:FunctionBlock"/>
              </xsd:choice>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="Documentation" type="ppx:TextBase" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="schemaVersion" type="xsd:decimal" use="required"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="FunctionBlock">
    <xsd:annotation>
      <xsd:documentation>Conceptual POU where unnecessary elements for AML are removed from IEC 61131-10
      FunctionBlock.</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="Parameters" type="aml:ParameterSet" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="Vars" minOccurs="0" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="Variable" type="aml:VariableDecl" minOccurs="0" maxOccurs="unbounded"/>
          </xsd:sequence>
          <xsd:attribute name="constant" type="xsd:boolean" use="optional" default="false"/>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="MainBody" type="ppx:BehaviorRepresentationBase" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" use="required"/>
    <xsd:attribute name="uuid" type="aml:UuidString" use="required"/>
  </xsd:complexType>
  <!-- Extended Variable with required uuid -->
  <xsd:complexType name="ParameterSet">
    <xsd:sequence>
      <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="InoutVars">
          <xsd:complexType>
```

```

<xsd:sequence>
  <xsd:element name="Variable" minOccurs="0" maxOccurs="unbounded">
    <xsd:complexType>
      <xsd:complexContent>
        <xsd:extension base="aml:VariableDecl">
          <xsd:attribute name="orderWithinParamSet" type="xsd:unsignedInt" use="required"/>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>
  </xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="InputVars">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Variable" minOccurs="0" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:complexContent>
            <xsd:extension base="aml:VariableDecl">
              <xsd:attribute name="orderWithinParamSet" type="xsd:unsignedInt" use="required"/>
              <xsd:attribute name="edgeDetection" type="ppx:EdgeModifierType" use="optional" default="none"/>
            </xsd:extension>
          </xsd:complexContent>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="OutputVars">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Variable" minOccurs="0" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:complexContent>
            <xsd:extension base="aml:VariableDecl">
              <xsd:attribute name="orderWithinParamSet" type="xsd:unsignedInt" use="required"/>
            </xsd:extension>
          </xsd:complexContent>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
</xsd:choice>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="VariableDecl">
  <xsd:sequence>
    <xsd:element name="Documentation" type="ppx:TextBase" minOccurs="0"/>
    <xsd:element name="AddData" type="ppx:AddData" minOccurs="0"/>
    <xsd:element name="Type" type="ppx:TypeRef"/>
    <xsd:element name="InitialValue" type="ppx:Value" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="name" type="xsd:string" use="required"/>
  <xsd:attribute name="uuid" type="aml:UuidString" use="required"/>
</xsd:complexType>
<xsd:complexType name="StructTypeSpec">
  <xsd:complexContent>
    <xsd:extension base="ppx:TypeSpecBase">
      <xsd:sequence>
        <xsd:element name="Member" type="aml:VariableDecl" minOccurs="1" maxOccurs="unbounded"/>
        <xsd:element name="AddData" type="ppx:AddData" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="overlap" type="xsd:boolean" use="optional" default="false"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<!-- Intermediate Modelling Layer -->
<xsd:complexType name="IML">
  <xsd:complexContent>
    <xsd:extension base="ppx:BehaviorRepresentationBase">
      <xsd:sequence>
        <xsd:element name="Resource" minOccurs="0" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="Status" maxOccurs="unbounded">

```

```

<xsd:complexType>
  <xsd:attribute name="uuid" type="aml:UuidString" use="required"/>
  <xsd:attribute name="name" type="xsd:string" use="required"/>
  <xsd:attribute name="order" type="xsd:integer" use="optional"/>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
<xsd:attribute name="uuid" type="aml:UuidString" use="required"/>
<xsd:attribute name="name" type="xsd:string" use="required"/>
<xsd:attribute name="order" type="xsd:integer" use="optional"/>
</xsd:complexType>
</xsd:element>
<xsd:element name="TimeInformation" minOccurs="0">
  <xsd:complexType>
    <xsd:attribute name="timeFormat" type="aml:TimeFormatEnum" use="required"/>
    <xsd:attribute name="startOfScale" type="aml:TimeUnion" use="optional" default="PT0S"/>
    <xsd:attribute name="endOfScale" type="aml:TimeUnion" use="optional"/>
    <xsd:attribute name="scaleInterval" type="aml:TimeUnion" use="optional"/>
  </xsd:complexType>
</xsd:element>
<xsd:choice minOccurs="0" maxOccurs="unbounded">
  <xsd:element name="IMLStep" type="aml:IMLStep"/>
  <xsd:element name="IMLTransition" type="aml:IMLTransition"/>
  <xsd:element name="IMLSimultaneousDivergence" type="aml:IMLSimultaneousDivergence"/>
  <xsd:element name="IMLSimultaneousConvergence" type="aml:IMLSimultaneousConvergence"/>
</xsd:choice>
</xsd:sequence>
<xsd:attribute name="logicModelType" type="aml:LogicModelTypeEnum" use="required"/>
</xsd:extension>
</xsd:complexType>
</xsd:complexType name="IMLStep">
  <xsd:complexContent>
    <xsd:extension base="ppx:Step">
      <xsd:attribute name="uuid" type="aml:UuidString" use="required"/>
      <xsd:attribute name="startTime" type="aml:TimeUnion" use="required"/>
      <xsd:attribute name="endTime" type="aml:TimeUnion" use="required"/>
      <xsd:attribute name="buffer" type="xsd:duration" use="optional"/>
      <xsd:attribute name="refStartStatusId" type="aml:UuidString" use="optional"/>
      <xsd:attribute name="refEndStatuId" type="aml:UuidString" use="optional"/>
    </xsd:extension>
    <!-- For Activity on Node Network -->
    <!-- For Timing Diagram -->
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="IMLTransition">
  <xsd:complexContent>
    <xsd:extension base="ppx:Transition">
      <xsd:attribute name="uuid" type="aml:UuidString" use="required"/>
      <xsd:attribute name="refTime" type="aml:TimeUnion" use="optional"/>
      <xsd:attribute name="refEndSegmentId" type="aml:UuidString" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="IMLSimultaneousDivergence">
  <xsd:complexContent>
    <xsd:extension base="ppx:SimultaneousDivergence">
      <xsd:attribute name="uuid" type="aml:UuidString" use="required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="IMLSimultaneousConvergence">
  <xsd:complexContent>
    <xsd:extension base="ppx:SimultaneousConvergence">
      <xsd:attribute name="uuid" type="aml:UuidString" use="required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<!-- MathML Integration -->
<xsd:complexType name="MathematicalExpression">
  <xsd:complexContent>
    <xsd:extension base="ppx:BehaviorRepresentationBase">
      <xsd:sequence>
        <xsd:element name="VariableMapping" minOccurs="0" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:attribute name="refMathMLVariableName" type="xsd:string" use="required"/>
            <xsd:attribute name="refFBVariableUUID" type="aml:UuidString" use="required"/>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```

<xsd:attribute name="direction" use="required">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="In"/>
      <xsd:enumeration value="Out"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
</xsd:complexType>
</xsd:element>
<xsd:element name="MathML">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:any namespace="##any" processContents="lax"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
</xsd:sequence>
<xsd:attribute name="name" type="xsd:string" use="optional"/>
<xsd:attribute name="uuid" type="aml:UuidString" use="required"/>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<!-- String representation for Universal Unique Identifier -->
<xsd:simpleType name="LogicModelTypeEnum">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="GanttChart"/>
    <xsd:enumeration value="ActivityOnNodeNetwork"/>
    <xsd:enumeration value="TimingDiagram"/>
    <xsd:enumeration value="Unknown"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="TimeFormatEnum">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="xsd:duration"/>
    <xsd:enumeration value="xsd:time"/>
    <xsd:enumeration value="xsd:dateTime"/>
    <xsd:enumeration value="xsd:date"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="TimeUnion">
  <xsd:union memberTypes="xsd:duration xsd:time xsd:dateTime xsd:date"/>
</xsd:simpleType>
<xsd:simpleType name="UuidString">
  <xsd:restriction base="xsd:string">
    <xsd:annotation>
      <xsd:documentation>This shall be some textual representation of UUID, which can be GUID.</xsd:documentation>
    </xsd:annotation>
  </xsd:restriction>
</xsd:simpleType>
</xsd:schema>

```

Bibliography

IEC 62424:2008², *Representation of process control engineering – Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools*

IEC 62881:2018, *Cause and effect matrix*

IECNORM.COM : Click to view the full PDF of IEC 62714-4:2020

² First edition. This first edition has been replaced in 2016 by a second edition IEC 62424:2016, *Representation of process control engineering – Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools*.

IECNORM.COM : Click to view the full PDF of IEC 62714-4:2020

SOMMAIRE

AVANT-PROPOS	119
INTRODUCTION.....	121
1 Domaine d'application	123
2 Références normatives	123
3 Termes, définitions et abréviations	124
3.1 Termes et définitions	124
3.2 Abréviations	126
4 Conformité.....	126
5 Vue d'ensemble	126
5.1 Informations concernant la logique en ingénierie de système de production.....	126
5.2 Modèles logiques en ingénierie de système de production	127
5.3 Archivage des modèles logiques dans un document XML de logique AML	128
5.4 Référencement des informations concernant la logique.....	130
6 Modèles logiques.....	131
6.1 Généralités	131
6.2 Diagrammes de Gantt	131
6.2.1 Généralités	131
6.2.2 Éléments graphiques	131
6.2.3 Structure du diagramme	132
6.2.4 Informations concernant la logique.....	132
6.2.5 Informations concernant la logique dans le diagramme de Gantt	132
6.3 Réseaux de représentation nodale.....	133
6.3.1 Généralités	133
6.3.2 Éléments graphiques.....	133
6.3.3 Structure nodale	133
6.3.4 Structure de réseau	134
6.3.5 Informations concernant la logique	134
6.3.6 Informations concernant la logique dans les réseaux de représentation nodale.....	134
6.4 Chronogrammes	135
6.4.1 Généralités	135
6.4.2 Éléments graphiques	135
6.4.3 Structure du diagramme	135
6.4.4 Informations concernant la logique	136
6.4.5 Informations concernant la logique dans les chronogrammes	136
6.5 Diagrammes fonctionnels en séquence	137
6.6 Diagrammes de bloc fonctionnel	137
6.7 Expression mathématique	137
7 Description du schéma XML de logique AML	138
7.1 Vue d'ensemble du schéma	138
7.1.1 Utilisation du schéma IEC 61131-10	138
7.1.2 Versionnage du schéma	139
7.2 Élément racine "AMLLogic"	140
7.2.1 Généralités	140
7.2.2 Attributs.....	140
7.2.3 Sous-élément "WriterHeader"	140

7.2.4	Sous-élément "Types"	140
7.2.5	Sous-élément "Documentation"	141
7.3	Type complexe "FunctionBlock"	141
7.3.1	Généralités	141
7.3.2	Attributs	141
7.3.3	Sous-élément "Parameters"	141
7.3.4	Sous-élément "Vars"	141
7.3.5	Sous-élément "MainBody"	141
7.3.6	Type complexe "ParameterSet"	142
7.3.7	Type complexe "VariableDecl"	142
7.4	Type complexe "IML"	143
7.4.1	Généralités	143
7.4.2	Attributs	144
7.4.3	Sous-élément "Resource"	144
7.4.4	Sous-élément "TimeInformation"	144
7.4.5	Choix de l'élément "IML***"	145
7.4.6	Type complexe "IMLStep"	145
7.4.7	Type complexe "IMLTransition"	146
7.4.8	Type complexe "IMLSimultaneousDivergence"	146
7.4.9	Type complexe "IMLSimultaneousConvergence"	146
7.5	Type complexe "MathematicalExpression"	147
7.5.1	Généralités	147
7.5.2	Attributs	147
7.5.3	Sous-élément "VariableMapping"	147
7.5.4	Sous-élément "MathML"	148
7.6	Type simple "LogicModelTypeEnum"	148
7.7	Type simple "TimeUnion"	148
7.8	Type simple "TimeFormatEnum"	148
7.9	Type simple "UuidString"	148
8	Archivage de modèles logiques	148
8.1	Généralités	148
8.2	Archivage des diagrammes de Gantt dans un document XML de logique AML	149
8.2.1	Règles communes	149
8.2.2	Archivage du début d'un diagramme de Gantt	149
8.2.3	Archivage des barres	150
8.2.4	Archivage des flèches	150
8.2.5	Archivage de barres de successeur	151
8.2.6	Archivage de barres de prédécesseur	152
8.3	Archivage des réseaux de représentation nodale dans le schéma XML de logique AML	153
8.3.1	Règles communes	153
8.3.2	Archivage du début d'un réseau de représentation nodale	154
8.3.3	Archivage de nœuds	154
8.3.4	Archivage des flèches	155
8.3.5	Archivage de nœuds de successeur	156
8.3.6	Archivage de nœuds de prédécesseur	157
8.4	Archivage des chronogrammes dans un document XML de logique AML	158
8.4.1	Règles communes	158

8.4.2	Archivage de la ligne chronologique d'un chronogramme.....	159
8.4.3	Archivage des ressources et des états de ressource	159
8.4.4	Archivage des lignes de vie	160
8.4.5	Archivage du signal de temps et du signal de ressource	162
8.5	Archivage des diagrammes fonctionnels en séquence dans un document XML de logique AML	164
8.5.1	Règles communes	164
8.5.2	Archivage des variables.....	164
8.6	Archivage des diagrammes de bloc fonctionnel dans un document XML de logique AML.....	165
8.6.1	Règles communes	165
8.6.2	Archivage des variables.....	166
8.7	Archivage des expressions mathématiques dans un document XML de logique AML.....	166
8.7.1	Règles communes	166
8.7.2	Archivage des variables.....	167
8.7.3	Archivage des mappings de variable.....	168
8.7.4	Archivage des expressions mathématiques.....	168
9	Méta-informations relatives aux outils d'écriture XML de logique AML	168
10	Extensions des classes AML pour la logique	170
10.1	Généralités	170
10.2	AutomationMLLogicRoleClassLib	170
10.2.1	Généralités	170
10.2.2	RoleClass InterlockingTargetGroup	170
10.2.3	RoleClass InterlockingSourceGroup	171
10.2.4	RoleClass LogicModelObject.....	172
10.3	AutomationMLLogicInterfaceClassLib	173
10.3.1	Généralités	173
10.3.2	InterfaceClass LogicModelInterface	173
10.3.3	InterfaceClass SequencingLogicModelInterface	174
10.3.4	InterfaceClass BehaviourLogicModelInterface	174
10.3.5	InterfaceClass InterlockingLogicModelInterface	174
10.3.6	InterfaceClass LogicModelElementInterface	175
10.3.7	InterfaceClass VariableInterface	175
10.3.8	InterfaceClass InterlockingVariableInterface	176
10.4	AutomationMLPLCopenXMLInterfaceClassLib.....	176
10.4.1	Généralités	176
10.4.2	InterfaceClass VariableInterface	176
10.5	AutomationMLInterfaceClassLib.....	177
10.5.1	Généralités	177
10.5.2	InterfaceClass InterlockingConnector	177
10.5.3	InterfaceClass PLCopenXMLInterface	177
11	Référencement de documents XML de logique AML	178
11.1	Généralités	178
11.2	Référencement des informations concernant la logique.....	178
12	Liaison d'objets AML à des informations de verrouillage	179
12.1	Généralités	179
12.2	Référencement des informations concernant le verrouillage.....	179

Annexe A (Informative) Exemples d'archivage des modèles logiques dans un document XML de logique AML.....	181
A.1 Exemple de diagrammes de Gantt	181
A.1.1 Généralités	181
A.1.2 Archivage des activités sans relation prédécesseur/successeur.....	181
A.1.3 Archivage d'une séquence d'activités	182
A.1.4 Archivage d'une séquence d'activités avec divergences	184
A.1.5 Archivage d'une séquence d'activités avec convergences.....	185
A.2 Exemple d'archivage de réseau de représentation nodale	186
A.2.1 Généralités	186
A.2.2 Archivage des activités sans relation prédécesseur/successeur.....	186
A.2.3 Archivage d'une séquence d'activités	187
A.2.4 Archivage d'une séquence d'activités avec divergences	189
A.2.5 Archivage d'une séquence d'activités avec convergences.....	190
A.3 Exemple d'archivage de chronogrammes	191
A.3.1 Généralités	191
A.3.2 Exemple d'archivage de signal interne	191
A.3.3 Exemple d'archivage de signal externe	193
A.3.4 Exemple d'archivage de signal entre deux flux d'états de ressource	194
A.4 Exemple d'archivage de diagrammes fonctionnels en séquence.....	196
A.5 Exemple d'archivage de diagrammes de bloc fonctionnel	199
A.6 Exemple d'archivage d'expressions mathématiques	201
Annexe B (Informative) Exemples de référencement des informations concernant la logique.....	205
B.1 Généralités	205
B.2 Référencement des informations concernant la logique exprimées sous la forme de modèles logiques.....	205
B.2.1 Généralités	205
B.2.2 Référencement des informations concernant la logique archivées dans un FunctionBlock	205
B.2.3 Référencement des informations concernant la logique, qui sont composées de plusieurs FunctionBlocks.....	206
B.2.4 Référencement des informations concernant la logique, qui sont composées de plusieurs documents XML de logique AML	207
B.3 Référencement des informations concernant la logique en tant que partie de modèles logiques	209
B.3.1 Généralités	209
B.3.2 Référencement d'une variable	209
B.3.3 Référencement d'un élément logique	210
B.4 Référencement des informations concernant la logique en tant que partie de modèles logiques déjà référencés.....	211
Annexe C (Informative) Exemples de référencement des informations concernant le verrouillage.....	214
C.1 Généralités	214
C.2 Informations concernant le verrouillage.....	215
C.3 Référencements des informations concernant le verrouillage sans condition de verrouillage	215
C.4 Référencements des informations concernant le verrouillage avec condition de verrouillage	218
Annexe D (Normative) Représentation XML des bibliothèques normalisées AML	222
D.1 Généralités	222

D.2	AutomationMLLogicRoleClassLib	222
D.3	AutomationMLLogicInterfaceClassLib	223
D.4	AutomationMLPLCopenXMLInterfaceClassLib	224
Annexe E (Normative) Représentation XML du schéma XML de logique AML		225
Bibliographie		229
Figure 1 – Vue d'ensemble du format d'échange de données techniques (AML)		121
Figure 2 – Exemple de représentation du système avec les rôles des informations dans AML		127
Figure 3 – Modèles logiques dans AML		128
Figure 4 – Archivage des modèles logiques dans un document XML de logique AML		129
Figure 5 – Éléments de modélisation du schéma XML de logique AML		130
Figure 6 – Éléments de modèle des diagrammes de Gantt		131
Figure 7 – Informations données par des diagrammes de Gantt		133
Figure 8 – Éléments de modèle de réseaux de représentation nodale		133
Figure 9 – Informations fournies par des réseaux de représentation nodale d'activité		135
Figure 10 – Éléments de modèle des chronogrammes		135
Figure 11 – Informations données par des chronogrammes		137
Figure 12 – Vue d'ensemble du schéma de logique AML		139
Figure 13 – Élément racine "AMLLogic"		140
Figure 14 – Type complexe "FunctionBlock"		141
Figure 15 – Type complexe "ParameterSet"		142
Figure 16 – Type complexe "VariableDecl"		143
Figure 17 – Type complexe "IML"		144
Figure 18 – Type complexe "IMLStep"		145
Figure 19 – Type complexe "IMLTransition"		146
Figure 20 – Type complexe "IMLSimultaneousDivergence"		146
Figure 21 – Type complexe "IMLSimultaneousConvergence"		147
Figure 22 – Type complexe "MathematicalExpression"		147
Figure 23 – AutomationMLLogicRoleClassLib		170
Figure 24 – AutomationMLLogicInterfaceClassLib		173
Figure 25 – AutomationMLPLCopenXMLInterfaceClassLib		176
Figure A.1 – Débit des vannes		201
Figure A.2 – Exemple d'archivage d'expression mathématique		204
Figure B.1 – Référencement des informations concernant la logique (en tant que SFC) archivées dans un FunctionBlock		206
Figure B.2 – Texte XML du fichier CAEX pour le référencement des informations concernant la logique archivées dans un FunctionBlock		206
Figure B.3 – Référencement des informations concernant la logique, qui sont composées de plusieurs FunctionBlocks		207
Figure B.4 – Référencement des informations concernant la logique qui sont composées de plusieurs documents XML de logique AML		208
Figure B.5 – Texte XML du fichier CAEX pour le référencement des informations concernant la logique composées de plusieurs documents XML de logique AML		208
Figure B.6 – Référencement d'une variable		209

Figure B.7 – Texte XML du fichier CAEX pour le référencement d'une variable	210
Figure B.8 – Référencement d'un élément logique	210
Figure B.9 – Texte XML du fichier CAEX pour le référencement d'un élément logique	211
Figure B.10 – Référencement d'une variable d'un modèle logique déjà référencé	212
Figure B.11 – Texte XML d'un fichier CAEX pour le référencement d'une variable d'un modèle logique déjà référencé	213
Figure C.1 – Exemple de système de fabrication.....	214
Figure C.2 – Exemple de groupe source de verrouillage et de groupe cible de verrouillage.....	215
Figure C.3 – Référencements des informations concernant le verrouillage sans condition de verrouillage	217
Figure C.4 – Texte XML du fichier CAEX pour le référencement des informations concernant le verrouillage sans condition de verrouillage.....	218
Figure C.5 – Référencements des informations concernant le verrouillage avec condition de verrouillage	219
Figure C.6 – Texte XML du fichier CAEX pour le référencement des informations concernant le verrouillage avec condition de verrouillage.....	220
Figure C.7 – Association de l'interface logique à l'interface physique (extension vers la Figure C.5)	221
Figure C.8 – Texte XML du fichier CAEX pour l'association de l'interface logique à l'interface physique (extension vers la Figure C.6)	221
Tableau 1 – Abréviations	126
Tableau 2 – Mapping du début d'un diagramme de Gantt à un élément IML	149
Tableau 3 – Mapping des barres du diagramme de Gantt à des éléments IML	150
Tableau 4 – Mapping d'une flèche du diagramme de Gantt à des éléments IML	150
Tableau 5 – Mapping d'une barre de diagramme de Gantt comportant au moins une barre de successeur à des éléments IML	151
Tableau 6 – Mapping d'une barre de diagramme de Gantt comportant au moins une barre de prédécesseur à des éléments IML.....	152
Tableau 7 – Mapping du début d'un réseau de représentation nodale à des éléments IML	154
Tableau 8 – Mapping de nœuds de réseau de représentation nodale à des éléments IML.....	154
Tableau 9 – Mapping de flèches de réseau de représentation nodale à des éléments IML	155
Tableau 10 – Mapping de nœuds de successeur de réseau de représentation nodale à des éléments IML	156
Tableau 11 – Mapping de nœuds de prédécesseur de réseau de représentation nodale à des éléments IML.....	157
Tableau 12 – Mapping de la ligne chronologique d'un chronogramme à des éléments IML	159
Tableau 13 – Mapping des ressources et états de ressource du chronogramme à des éléments IML	159
Tableau 14 – Mapping des lignes de vie du chronogramme à des éléments IML	160
Tableau 15 – Mapping du signal de temps et du signal de ressource à des éléments IML	163
Tableau 16 – Archivage des variables d'un diagramme fonctionnel en séquence dans un document XML de logique AML	165

Tableau 17 – Archivage des variables d'un diagramme de bloc fonctionnel dans un document XML de logique AML.....	166
Tableau 18 – Archivage des variables d'une expression mathématique dans un document XML de logique AML.....	167
Tableau 19 – Archivage des mappings de variable d'une expression mathématique dans un document XML de logique AML	168
Tableau 20 – Archivage d'une expression mathématique dans un document XML de logique AML.....	168
Tableau 21 – Méta-informations relatives à chaque outil d'écriture XML de logique AML	169
Tableau 22 – RoleClass InterlockingTargetGroup	171
Tableau 23 – RoleClass InterlockingSourceGroup	172
Tableau 24 – RoleClass LogicModelObject	173
Tableau 25 – InterfaceClass LogicModelInterface	174
Tableau 26 – InterfaceClass SequencingLogicModelInterface	174
Tableau 27 – InterfaceClass BehaviourLogicModelInterface	174
Tableau 28 – InterfaceClass InterlockingLogicModelInterface	175
Tableau 29 – InterfaceClass LogicModelElementInterface	175
Tableau 30 – InterfaceClass VariableInterface	175
Tableau 31 – InterfaceClass InterlockingVariableInterface.....	176
Tableau 32 – InterfaceClass VariableInterface	177
Tableau 33 – InterfaceClass InterlockingConnector.....	177
Tableau 34 – InterfaceClass PLCOpenXMLInterface	178
Tableau A.1 – Exemple d'archivage de diagramme de Gantt "activités sans relation prédécesseur/successeur"	182
Tableau A.2 – Exemple d'archivage de diagramme de Gantt "séquence d'activités"	183
Tableau A.3 – Exemple d'archivage de diagramme de Gantt "séquence d'activités avec divergence".....	184
Tableau A.4 – Exemple d'archivage de diagramme de Gantt "séquence d'activités avec convergences"	185
Tableau A.5 – Exemple d'archivage de réseau de représentation nodale "activités sans relation prédécesseur/successeur".....	187
Tableau A.6 – Exemple d'archivage de réseau de représentation nodale "séquence d'activités"	188
Tableau A.7 – Exemple d'archivage du réseau de représentation nodale "séquence d'activités avec divergence"	189
Tableau A.8 – Exemple d'archivage du réseau de représentation nodale "séquence d'activités avec convergences"	190
Tableau A.9 – Exemple d'archivage du chronogramme "passage d'un état à l'état suivant".....	192
Tableau A.10 – Exemple de mapping du chronogramme "deux signaux externes retirés dans un délai de trois secondes"	193
Tableau A.11 – Exemple d'archivage du chronogramme "signal retiré par un état de ressource et utilisé par un autre"	195
Tableau A.12 – Exemple d'archivage de diagrammes fonctionnels en séquence	197
Tableau A.13 – Exemple d'archivage d'un diagramme de bloc fonctionnel	199

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

**FORMAT D'ÉCHANGE DE DONNÉES POUR UNE UTILISATION DANS
L'INGÉNIERIE DES SYSTÈMES D'AUTOMATISATION INDUSTRIELLE –
AUTOMATION MARKUP LANGUAGE –****Partie 4: Logique****AVANT-PROPOS**

- 1) La Commission Électrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. À cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés « Publication(s) de l'IEC »). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

La Norme internationale IEC 62714-4 a été établie par le sous-comité 65E: Les dispositifs et leur intégration dans les systèmes de l'entreprise, du comité d'études 65 de l'IEC: Commande et automation dans les processus industriels.

Le texte de cette Norme internationale est issu des documents suivants:

CDV	Rapport de vote
65E/654/CDV	65E/692/RVC

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette Norme internationale.

Ce document a été rédigé selon les Directives ISO/IEC, Partie 2.

Une liste de toutes les parties de la série IEC 62714, publiées sous le titre général *Format d'échange de données techniques pour une utilisation dans l'ingénierie des systèmes d'automatisation industrielle - Automation Markup Language*, peut être consultée sur le site web de l'IEC.

Le comité a décidé que le contenu de ce document ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous "<http://webstore.iec.ch>" dans les données relatives au document recherché. À cette date, le document sera

- reconduit,
- supprimé,
- remplacé par une édition révisée, ou
- amendé.

Le contenu du corrigendum de novembre 2020 a été pris en considération dans cet exemplaire.

IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer ce document en utilisant une imprimante couleur.

INTRODUCTION

Le format d'échange de données défini dans l'IEC 62714 (Automation Markup Language (AML)) repose sur un schéma XML et a été développé afin de prendre en charge l'échange de données entre les outils techniques évoluant dans un environnement d'outils techniques hétérogène. L'IEC 62714-1 donne un aperçu de ce format.

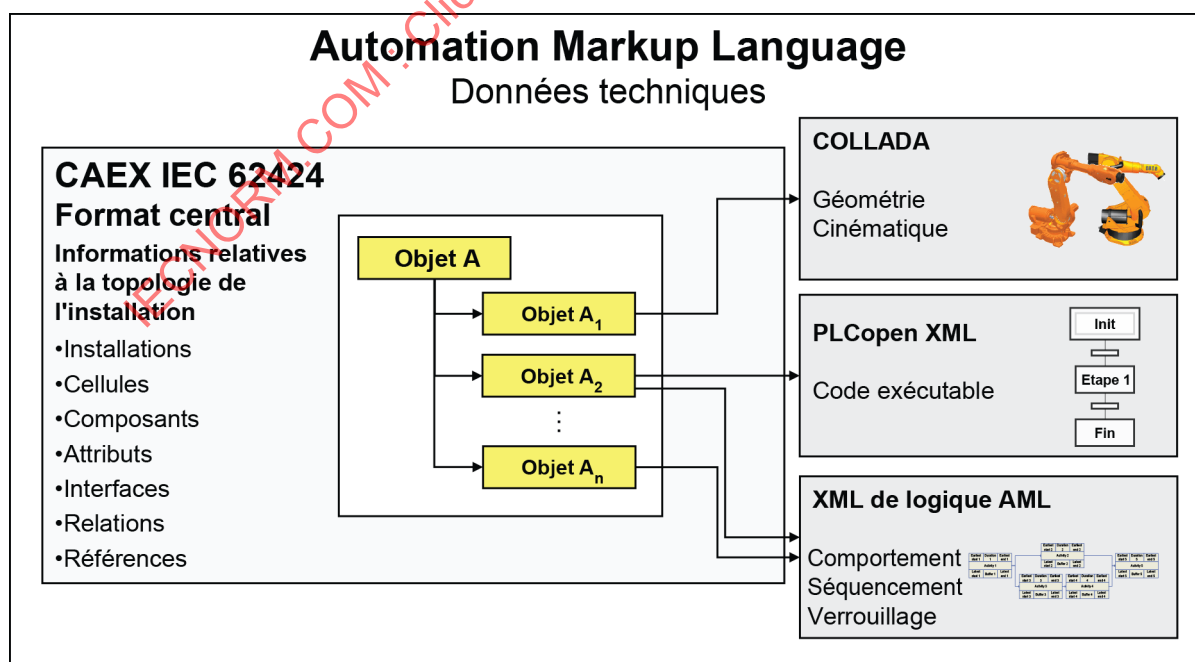
L'objectif du langage AML est l'interconnexion des outils techniques, issus de l'environnement d'outils hétérogène existant, dans leurs différentes disciplines, par exemple, ingénierie des installations mécaniques, études d'électricité, ingénierie de procédés, ingénierie de commande de processus, développement des IHM, programmation PLC, programmation de robots, etc.

Le langage AML archive les informations techniques en respectant le paradigme orienté objet, et permet la modélisation des composants d'installations physiques et logiques sous forme d'objets de données qui englobent différents aspects. Un objet peut comporter d'autres sous-objets, et peut lui-même faire partie intégrante d'une composition ou d'une agrégation plus importante. Les objets typiques existant dans l'automatisation d'installations comprennent les informations concernant la topologie, la géométrie, la cinématique et la logique, tandis que la logique comprend pour sa part le séquençement, le comportement et la commande.

AML combine les formats de données industrielles existants, conçus pour l'archivage et l'échange de différents aspects des informations techniques. Ces formats de données sont utilisés "en l'état" dans le cadre de leurs propres spécifications et ne sont pas associés aux besoins du langage AML.

La caractéristique centrale de l'AML est le format de données central CAEX, qui permet d'interconnecter les différents formats de données. Le langage AML a par conséquent une architecture de document répartie intrinsèque.

La Figure 1 représente l'architecture AML de base et la répartition des informations concernant la topologie, la géométrie, la cinématique et la logique.



IEC

Figure 1 – Vue d'ensemble du format d'échange de données techniques (AML)

Du fait des différents aspects d'AML, la série IEC 62714 comporte différentes parties concentrées sur différents aspects.

- IEC 62714-1: Architecture et exigences générales
Cette partie spécifie l'architecture AML générale, et la modélisation des données techniques, classes, instances, relations, références, hiérarchies, bibliothèques AML de base et concepts AML étendus.
- IEC 62714-2: Bibliothèques de classe de rôles
Cette partie spécifie d'autres bibliothèques AML.
- IEC 62714-3: Géométrie et cinématique
Cette partie spécifie la modélisation des informations concernant la géométrie et la cinématique.
- IEC 62714-4: Logique
Cette partie spécifie la modélisation et le référencement des informations relatives à la logique.

D'autres parties peuvent être ajoutées à l'avenir afin d'interconnecter d'autres normes de données avec l'AML.

L'Article 5 donne un aperçu informatif de la présente partie de la norme.

L'Article 6 donne une description normative des modèles logiques concernés.

L'Article 7 donne une description normative du schéma XML logique AML, avec lequel les modèles logiques peuvent être archivés.

L'Article 8 spécifie les dispositions normatives d'archivage des modèles logiques en langage XML de logique AML.

L'Article 9 définit la manière d'archiver les méta-informations relatives à l'outil source directement dans le document XML de logique AML.

L'Article 10 définit une bibliothèque de classe de rôles et une bibliothèque de classe d'interfaces liées à la logique.

L'Article 11 donne une description normative du référencement des informations concernant la logique dans les documents XML de logique AML.

L'Article 12 donne une description normative du référencement des informations relatives au verrouillage dans les documents XML de logique AML.

L'Annexe A donne des exemples d'archivage de modèles logiques en langage XML de logique AML.

L'Annexe B décrit les méthodes de référencement des informations concernant la logique.

L'Annexe C décrit les méthodes de référencement des informations concernant le verrouillage.

L'Annexe D donne une représentation XML normative des bibliothèques définies dans le présent document.

L'Annexe E donne une représentation XML normative du schéma XML de logique AML défini dans le présent document.

FORMAT D'ÉCHANGE DE DONNÉES POUR UNE UTILISATION DANS L'INGÉNIERIE DES SYSTÈMES D'AUTOMATISATION INDUSTRIELLE – Automation Markup Language –

Partie 4: Logique

1 Domaine d'application

La présente partie de l'IEC 62714 spécifie l'intégration des informations concernant la logique dans le cadre d'un modèle AML pour l'échange de données dans un environnement d'outils techniques hétérogène des systèmes de production.

Le présent document spécifie trois types d'informations concernant la logique: les informations relatives au séquençement, au comportement et au verrouillage.

Le présent document porte sur les six modèles logiques de séquençement et de comportement suivants (couvrant les différentes phases du processus d'ingénierie des systèmes de production) et sur la manière de les intégrer dans AML: diagramme de Gantt, réseau de représentation nodale, chronogramme, diagramme fonctionnel en séquence (SFC - *sequential function chart*), diagramme de bloc fonctionnel (FBD - *function block diagram*) et expression mathématique.

Le présent document spécifie la manière de modéliser le diagramme de Gantt, le réseau de représentation nodale et le chronogramme et de les archiver dans la couche de modélisation intermédiaire (IML - *intermediate modelling layer*).

NOTE 1 Ainsi, il est possible de transformer un modèle logique en un autre. Une transformation directe prend en charge le processus d'enrichissement des informations et réduit ou évite de saisir à nouveau les informations entre les outils techniques d'échange.

NOTE 2 D'autres modèles logiques (modèles logiques événementiels comme les diagrammes d'états, par exemple) peuvent être mappés avec IML.

Le présent document spécifie la manière de modéliser les informations concernant le verrouillage (source de verrouillage et groupes cibles, par exemple) dans AML. Le modèle logique de verrouillage est archivé dans le diagramme de bloc fonctionnel (FBD).

Le présent document spécifie le schéma XML de logique AML qui archive les modèles logiques selon l'IEC 61131-10.

Le présent document spécifie la manière de référencer les programmes PLC archivés dans les documents XML PLCopen.

Le présent document ne précise pas la procédure d'échange de données ni les exigences de mise en œuvre pour les outils d'importation/exportation.

2 Références normatives

Les documents suivants sont cités dans le texte de sorte qu'ils constituent, pour tout ou partie de leur contenu, des exigences du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC 61131-3, *Automates programmables – Partie 3: Langages de programmation*

IEC 61131-10, *Automates programmables – Partie 10: Format d'échange XML ouvert PLC*

IEC 62714-1:2014¹, *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*

W3C. *Extensible Markup Language (XML) 1.0 (Third Edition)*, W3C Recommendation 04 February 2004 [en ligne]. Edité par T. Bray et al., février 2004 [consulté le 2020-05-14]. Disponible à l'adresse <http://www.w3.org/TR/2004/REC-xml-20040204>

W3C. *Mathematical Markup Language (MathML) Version 2.0 (Second Edition)*, W3C Recommendation 21 October 2003 [en ligne]. Edité par D. Carlisle et al., octobre 2003 [consulté le 2020-05-14]. Disponible à l'adresse <https://www.w3.org/TR/MathML2/>

INTERNET ENGINEERING TASK FORCE (IETF). RFC 5646: *Tags for Identifying Languages* [en ligne]. Edité par A. Phillips et M. Davis, septembre 2009 [consulté le 2020-05-14]. Disponible à l'adresse <https://tools.ietf.org/html/rfc5646>

INTERNET ENGINEERING TASK FORCE (IETF). RFC 4122, *A Universally Unique Identifier (UUID) URN Namespace* [en ligne]. Edité par P. Leach et al., juillet 2005 [consulté le 2020-05-14]. Disponible à l'adresse <https://tools.ietf.org/html/rfc4122>

PLCopen. *XML 2.01: XML formats for IEC 61131-3* [en ligne]. Edité par K. Ketterle et al., mai 2009 [consulté le 2020-05-14]. Disponible à l'adresse suivante: https://www.plcopen.org/system/files/downloads/tc6_xml_v201_technical_doc.pdf

3 Termes, définitions et termes abrégés

3.1 Termes et définitions

Pour les besoins du présent document, les termes et les définitions de l'IEC 62714-1, ainsi que les suivants, s'appliquent:

L'ISO et l'IEC tiennent à jour des bases de données terminologiques destinées à être utilisées en normalisation, consultables aux adresses suivantes:

- IEC Electropedia: disponible à l'adresse <http://www.electropedia.org/>
- ISO Online browsing platform: disponible à l'adresse <http://www.iso.org/obp>

3.1.1

objet logique AML

objet AML contenant les informations concernant la logique

3.1.2

document XML de logique AML

document XML externe reposant sur le schéma XML de logique AML, qui peut être référencé à partir de l'objet logique AML

¹ Première édition (2014). Cette première édition a été remplacée en 2018 par une deuxième édition IEC 62714-1:2018, *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*.

3.1.3**schéma XML de logique AML**

schéma XML d'archivage des modèles logiques d'AML

3.1.4**informations concernant le comportement**

informations qui décrivent les réponses du système commandé aux informations de séquençement et aux interactions externes

3.1.5**modèle logique de comportement**

représentation des informations concernant le comportement

3.1.6**informations concernant le verrouillage**

informations qui décrivent les instructions données au système commandé pour éviter qu'il ait un impact dangereux sur les hommes et l'environnement

3.1.7**modèle logique de verrouillage**

représentation des informations concernant le verrouillage

3.1.8**groupe source de verrouillage**

groupe d'objets AML qui indique à quel moment des actions sont nécessaires pour assurer la sécurité fonctionnelle

3.1.9**groupe cible de verrouillage**

groupe d'objets AML qui exécute les actions nécessaires pour assurer la sécurité fonctionnelle

3.1.10**élément de modèle logique**

élément d'un modèle logique

3.1.11**informations concernant la logique**

informations de séquençement ou informations de verrouillage

3.1.12**modèle logique**

représentation des informations concernant la logique

3.1.13**expression mathématique**

combinaison significative de nombres, de variables et de symboles mathématiques permettant de représenter de manière syntaxique les dépendances correctes qu'ils entretiennent

Note 1 à l'article: Une expression mathématique peut aller de termes simples jusqu'à des systèmes d'équations complexes.

3.1.14**informations concernant le séquençement**

informations qui décrivent les instructions données au système commandé

3.1.15**modèle logique de séquençement**

représentation des informations concernant le séquençement

3.2 Termes abrégés

Pour les besoins du présent document, les termes abrégés de l'IEC 62714-1 et celles du Tableau 1 s'appliquent.

Tableau 1 – Termes abrégés

SFC	Sequential Function Chart (diagramme fonctionnel en séquence)
FBD	Function Block Diagram (diagramme de bloc fonctionnel)
IML	Intermediate Modelling Layer (couche de modélisation intermédiaire)
XSLT	Extensible Stylesheet Language Transformations
MathML	Mathematical Markup Language

4 Conformité

Pour revendiquer la conformité à la présente partie de l'IEC 62714 en ce qui concerne la prise en charge d'AML, les exigences des Articles 6, 7, 8, 9, 10, 11 et 12 doivent être satisfaites. Dans le domaine d'application d'AML, un document XML de logique XML doit être conforme au schéma XML de logique AML défini à l'Article 7 et à l'Annexe E.

Un document XML PLCopen doit être conforme à la spécification de PLCopen® XML 2.0 ou 2.01 ou à la spécification de l'IEC 61131-10.

Le présent document ne définit pas les dispositions en matière d'échange de données ni les exigences de mise en œuvre pour les outils d'importation/exportation.

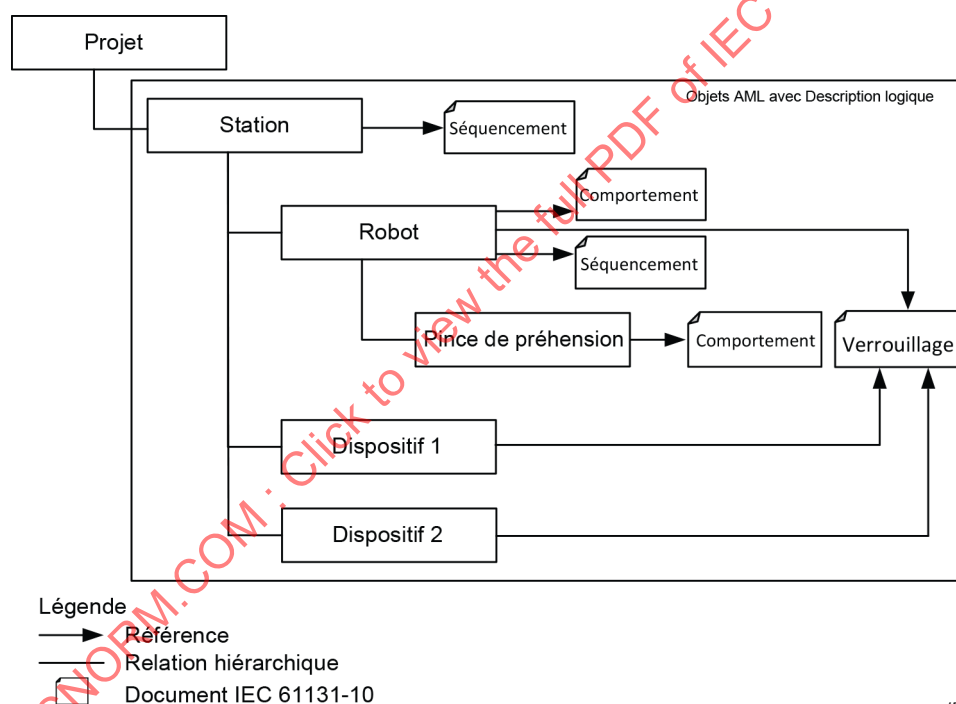
5 Vue d'ensemble**5.1 Informations concernant la logique en ingénierie de système de production**

- Le présent document porte sur l'archivage des informations concernant la logique (qui est un aspect important de la conception électrique, du développement IHM et de la programmation de commande de robot) pour les besoins de la simulation et de la mise en service virtuelle.
- Pour prendre en charge les différentes phases du processus d'ingénierie des systèmes de production itératifs couvrant les différents niveaux de détails, il est nécessaire qu'AML soit en mesure de couvrir les informations concernant la logique à partir de différents outils et différentes disciplines. Par conséquent, les différents rôles d'un système de production ou de simples composants doivent être archivés. Cette variété d'informations peut être classée dans deux types principaux: les informations concernant le séquençement et les informations concernant le comportement. Ces deux rôles d'informations visent des objectifs différents, mais ils peuvent présenter des points communs selon le point de vue et l'utilisation des éléments dans AML (voir la Figure 2).
 - Les informations concernant le séquençement décrivent les instructions données au système (ou sous-système) commandé. Dans leur forme la plus détaillée, elles sont souvent représentées par un programme exécuté dans un ou plusieurs automates (unité de commande de robot ou PLC, exemple). Toutefois, le développement des

séquences commence en amont dans le processus d'ingénierie (pour la modélisation à un niveau d'abstraction plus élevé) avec des modèles qui représentent les opérations exigées d'une unité d'échelle plus importante (cellule, ligne, usine, etc.).

- Les informations concernant le comportement sont des informations qui décrivent les réponses du système (ou sous-système) commandé aux informations de séquençement et aux interactions externes. Elles sont souvent représentées par un modèle donné qui réagit à une entrée externe (comportement d'une pince de préhension ou d'une vanne, par exemple). Ces entrées externes déclenchent le comportement ou la signalisation des états de l'outil de préhension.
- Outre les informations concernant le séquençement et le comportement, les informations concernant le verrouillage sont un troisième type complémentaire d'informations concernant la logique (voir la Figure 2). En ingénierie des systèmes de production, il est essentiel d'assurer un comportement sûr. Cela concerne non seulement l'interaction sûre avec les hommes et l'environnement, mais aide également à éviter les états instables des systèmes susceptibles d'avoir un impact néfaste sur les hommes et l'environnement ou d'endommager les machines et les produits.

Un système (ou sous-système) commandé peut être représenté par plusieurs rôles des informations ci-dessus dans AML - un robot, par exemple (voir la Figure 2).



IEC

Figure 2 – Exemple de représentation du système avec les rôles des informations dans AML

5.2 Modèles logiques en ingénierie de système de production

Pour être applicable dans le processus d'ingénierie des systèmes de production, il est nécessaire qu'AML prenne en charge les modèles logiques classiques (archivage des informations concernant la logique) qui couvrent chaque phase du processus d'ingénierie (voir la Figure 3). Ces modèles logiques sont les suivants:

- Les diagrammes de Gantt sont essentiellement utilisés lors des premières phases du processus d'ingénierie pour représenter le calendrier et la durée des processus de fabrication à un haut niveau d'abstraction (voir 6.2).
- Les réseaux de représentation nodale sont essentiellement appliqués lors des premières phases du processus d'ingénierie pour représenter le calendrier et l'interdépendance des processus de fabrication avec la capacité d'archivage des informations de temps (voir 6.3).

- Les chronogrammes sont essentiellement utilisés dans les dernières phases du processus d'ingénierie pour décrire les dispositifs avec leurs états et leurs relations temporelles (voir 6.4). Des signaux réels sont introduits et les informations concernant le séquençement peuvent être exprimées.
- Les diagrammes fonctionnels en séquence (SFC) sont l'un des langages de programmation de l'IEC 61131-3 essentiellement appliqués dans les dernières phases de l'ingénierie de système de production (voir 6.5). Les informations concernant le séquençement et les informations concernant le comportement peuvent être modélisées.
- Les diagrammes de bloc fonctionnel (FBD) sont l'un des langages de programmation de l'IEC 61131-3 essentiellement appliqués dans les dernières phases de l'ingénierie de système de production (voir 6.6). Les informations concernant le verrouillage et les informations concernant le comportement peuvent être modélisées.
- Les expressions mathématiques sont essentiellement utilisées dans les dernières phases du processus d'ingénierie pour décrire les relations mathématiques, physiques, chimiques ou logiques d'un dispositif (voir 6.7).

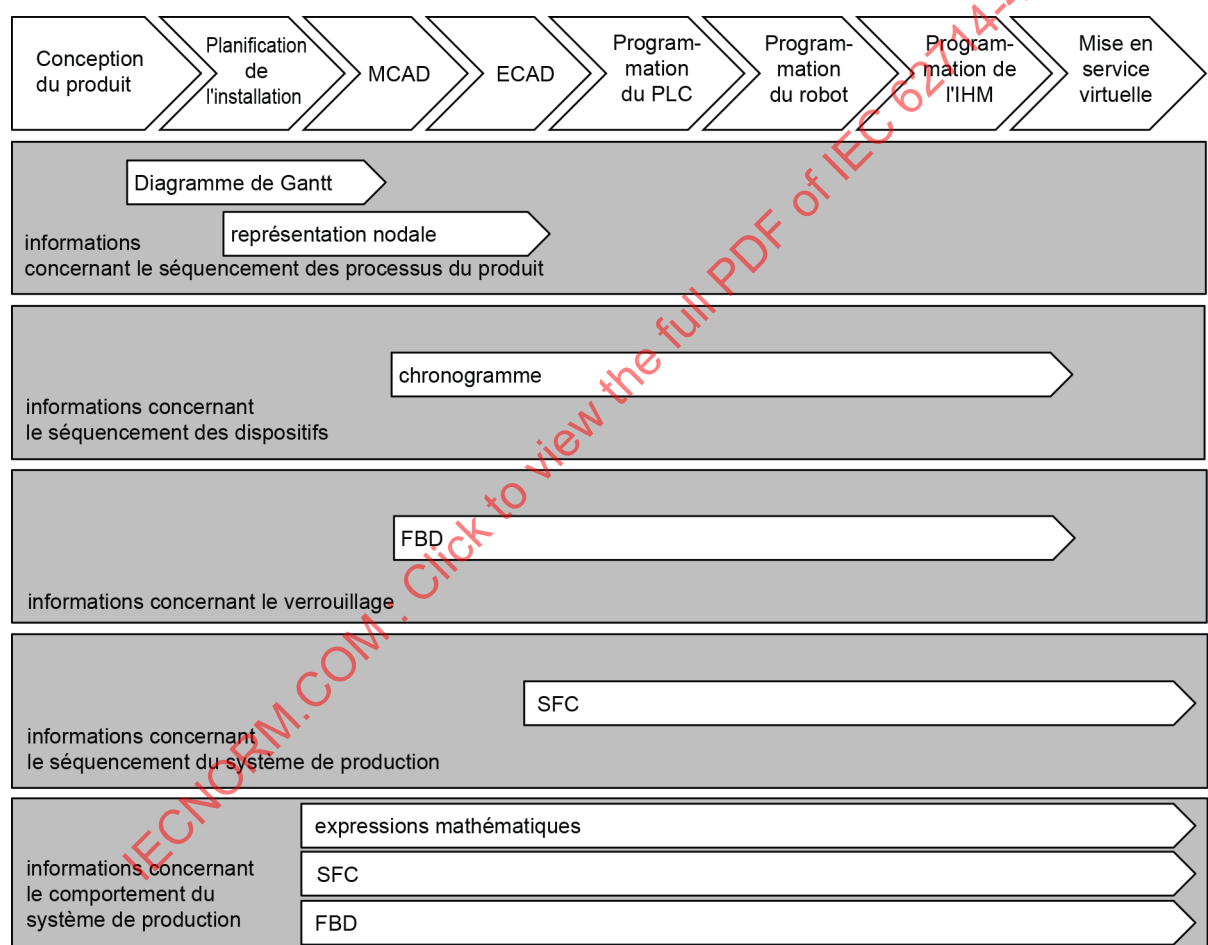


Figure 3 – Modèles logiques dans AML

5.3 Archivage des modèles logiques dans un document XML de logique AML

Il s'agit de permettre une transformation entre les modèles logiques. Cela est essentiel étant donné que le processus d'ingénierie des systèmes de production est un processus d'enrichissement des informations. Au début, les informations concernant le séquençement d'un système de production sont grossièrement décrites par un diagramme de Gantt, puis il est nécessaire de les détailler à l'étape suivante. Par conséquent, le diagramme de Gantt se transforme en réseau de représentation nodale: les informations issues du diagramme de Gantt sont reprises et les conditions de durée peuvent dès lors être ajoutées. En théorie, ce flux de travaux peut être réalisé selon le programme exécutable, en incrémentant les

informations à chaque étape. Il n'est donc pas utile d'entrer de nouveau les informations dans l'outil technique correspondant. Toutefois, étant donné que les modèles logiques ont différents pouvoirs de modélisation, les transformations directes et inverses entre eux provoquent une perte d'informations dans des cas particuliers (transformation d'un réseau de représentation nodale en diagramme de Gantt, par exemple).

La Figure 4 donne une vue d'ensemble du domaine d'application du présent document.

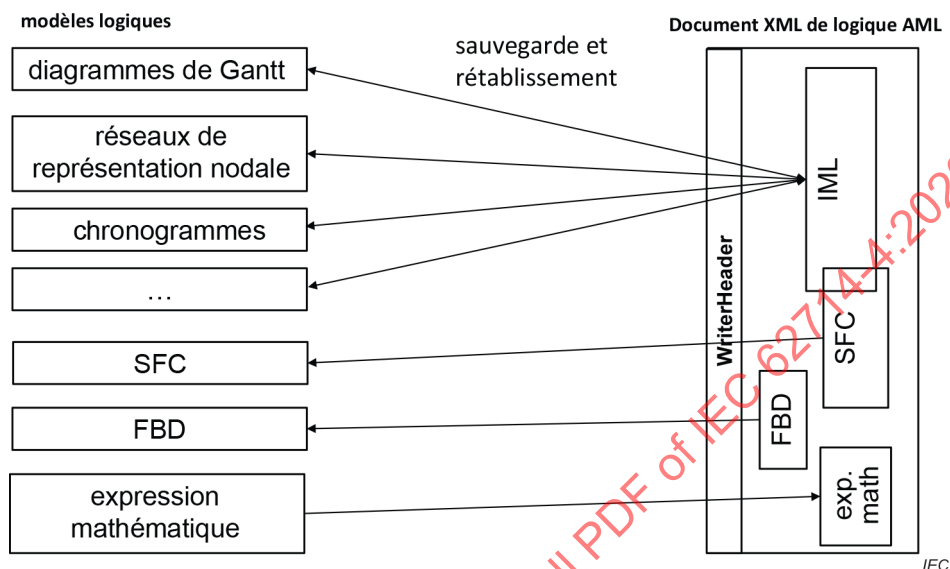


Figure 4 – Archivage des modèles logiques dans un document XML de logique AML

NOTE Pour convertir un diagramme fonctionnel en séquence (SFC) dans le contexte d'IML en diagramme fonctionnel en séquence dans le contexte de l'IEC 61131-10, certains fichiers XSLT accessibles au public peuvent être utilisés. Lors de la conversion d'un diagramme fonctionnel en séquence IML en diagramme fonctionnel en séquence IEC 61131-10, des informations sont susceptibles d'être perdues.

AML utilise IEC 61131-10 pour archiver les modèles logiques. Pour dissocier l'IEC 61131-10 des modèles logiques, la présente partie définit une couche de modélisation intermédiaire (IML). Par conséquent, IML constitue le modèle commun des modèles logiques, qui améliore les transformations directe et inverse, ainsi que l'extensibilité d'AML pour les nouveaux modèles logiques (les modèles logiques événementiels comme les diagrammes d'états, par exemple).

Étant donné que le présent document ne concerne pas simplement l'archivage des programmes PLC, il adapte l'utilisation de l'IEC 61131-10. Un schéma XML de logique AML, spécifié à l'Article 7, est utilisé et permet d'archiver les modèles logiques mentionnés. Les éléments de modélisation du schéma XML de logique AML pour archiver les modèles logiques à l'étude sont représentés à la Figure 5. Les modèles logiques peuvent être mappés à ces éléments de modélisation. Cela est décrit à l'Article 8. L'Annexe A donne des exemples.

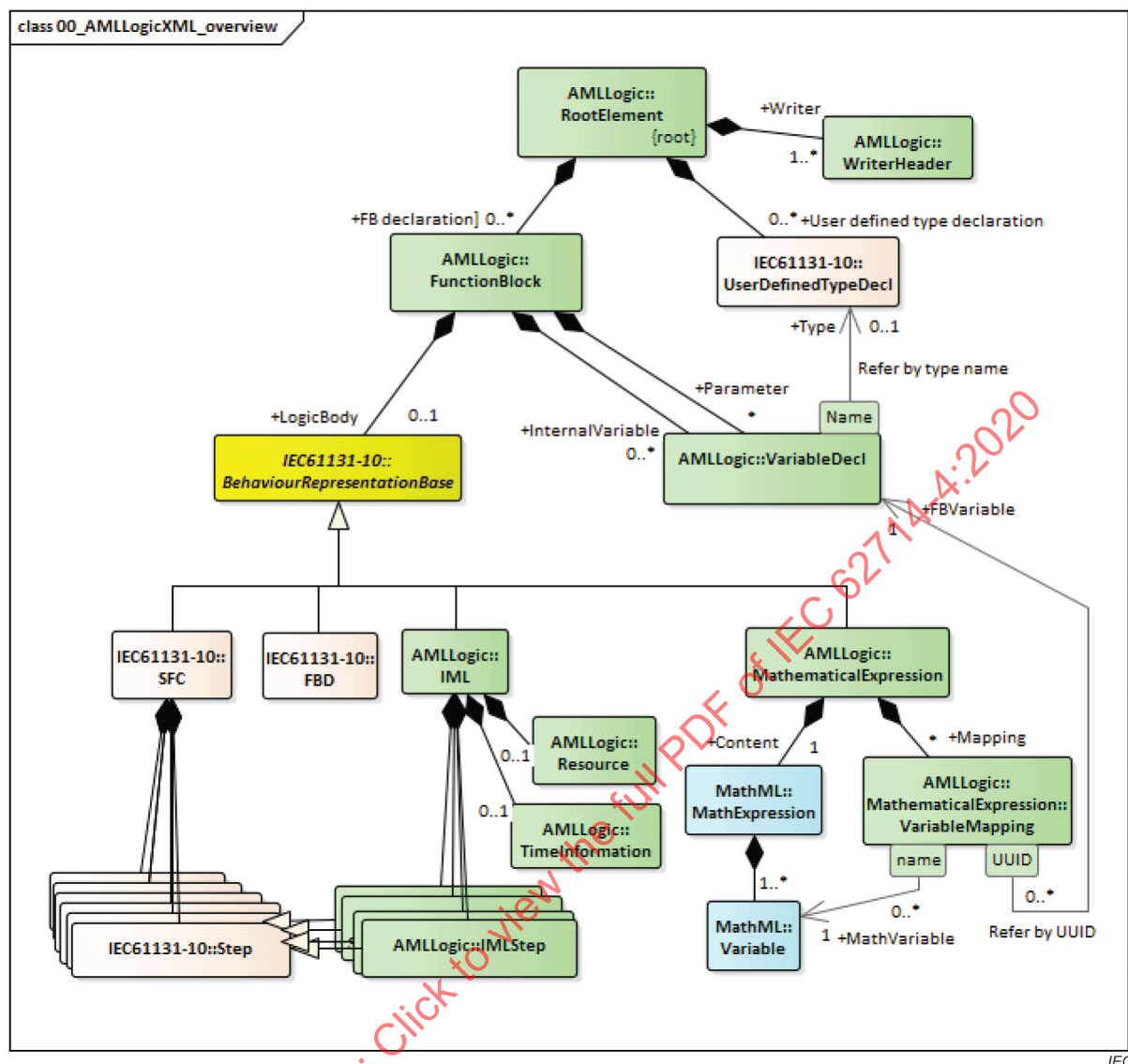


Figure 5 – Éléments de modélisation du schéma XML de logique AML

De plus, pour assurer un échange de données correct dans le domaine d'application d'AML, des méta-informations relatives au niveau du document XML de logique AML sont nécessaires à l'identification et la caractérisation du document. Des dispositions normatives sont définies à l'Article 9.

Étant donné que le schéma XML de logique AML repose sur l'IEC 61131-10, les autres modèles logiques qu'il fournit peuvent être utilisés. Dans la présente partie de l'IEC 62714, seuls les diagrammes fonctionnels en séquence (SFC) et les diagrammes de bloc fonctionnel (FBD) sont considérés comme étant pertinents pour le domaine d'application du présent document.

De même, un modèle logique représenté sous la forme d'une expression mathématique peut être archivé dans le document XML de logique AML. Soit le modèle logique lui-même contient la description de comportement du système, soit il peut être utilisé (ou appelé) pour fournir une expression mathématique à l'un des modèles logiques.

5.4 Référencement des informations concernant la logique

Dans l'architecture multidocument d'AML, les informations concernant la logique sont archivées dans des documents XML de logique AML séparés reposant sur le format de

données IEC 61131-10. La modélisation des informations concernant la logique est donc divisée en deux parties. En premier lieu, un objet, qui contient des informations concernant la logique, est modélisé en CAEX (format central). En second lieu, un document XML de logique AML contenant ces informations concernant la logique doit être fourni. Enfin, il est nécessaire de créer une référence entre l'objet CAEX et le document XML de logique AML ou un objet qu'il contient. Des dispositions normatives pour le référencement de ces trois rôles d'informations sont définies à l'Article 10 et l'Article 11. Une vue d'ensemble informative est donnée à l'Annexe B.

Pour le référencement des informations concernant le verrouillage, les dispositions supplémentaires doivent être prises en considération. Ces dispositions sont définies de manière normative à l'Article 12. Une vue d'ensemble informative est donnée à l'Annexe C.

6 Modèles logiques

6.1 Généralités

Le présent article définit les modèles logiques pris en considération dans la présente partie de l'IEC 62714. Il s'agit des éléments graphiques et de leurs règles de structure, des informations concernant la logique qu'ils représentent et de la manière dont ces informations sont mappées avec les éléments du modèle.

NOTE Le présent document ne donne aucune information relative à l'exécution de la logique, c'est-à-dire à la manière dont les éléments du modèle sont exécutés.

6.2 Diagrammes de Gantt

6.2.1 Généralités

Les diagrammes de Gantt sont des représentations graphiques des planifications généralement utilisées pour modéliser des ensembles partiellement ordonnés d'activités exécutées séquentiellement et simultanément, composés de l'ordre et de la durée d'exécution des activités. Ils ne donnent aucun moyen de modéliser les alternatives et les cycles.

NOTE Les diagrammes de Gantt sont également appelés graphiques à barres.

6.2.2 Éléments graphiques

Dans un diagramme de Gantt, les éléments graphiques suivants doivent être utilisés (voir la Figure 6):

- une ligne chronologique (axe horizontal) présentée sous la forme d'une horloge globale;
- un axe vertical sur lequel figurent tous les noms des barres correspondantes;
- aucune ou plusieurs barres horizontales;
- aucune ou plusieurs flèches orientées.

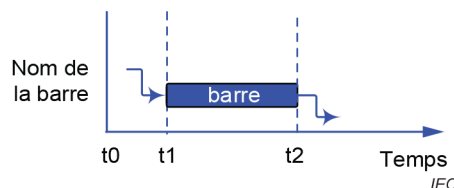


Figure 6 – Éléments de modèle des diagrammes de Gantt

6.2.3 Structure du diagramme

La structure suivante d'un diagramme de Gantt doit être utilisée:

- Chaque diagramme de Gantt doit être représenté par deux axes orthogonaux. L'axe horizontal représente le temps, ayant une direction, indiquant la durée linéaire. L'axe vertical représente la liste des noms de barre.
- Chaque barre doit avoir sa propre ligne dans la représentation graphique, avec son nom correspondant sur l'axe vertical.
- Le point de départ temporisé et le point final temporisé d'une barre doivent correspondre à l'axe horizontal.
- Les barres peuvent être reliées entre elles par des flèches.
- Une flèche doit être reliée entre le côté droit d'une barre et le côté gauche d'une autre barre.

6.2.4 Informations concernant la logique

Les diagrammes de Gantt doivent contenir les types suivants d'informations concernant la logique:

- Informations relatives à l'activité:
 - nom;
 - heure de début;
 - heure de fin;
 - durée.
- Aucune ou plusieurs informations relatives à la relation du prédécesseur entre les activités.
- Aucune ou plusieurs informations relatives la relation du successeur entre les activités.

6.2.5 Informations concernant la logique dans le diagramme de Gantt

Pour la représentation des informations concernant la logique dans un diagramme de Gantt, les dispositions suivantes s'appliquent (voir la Figure 7):

- Une activité doit être représentée par une barre.
 - Un nom d'activité doit être représenté par le nom d'une barre figurant sur l'axe vertical.
 - L'heure de début d'activité doit être représentée par le point de départ temporisé d'une barre figurant sur l'axe horizontal.
 - L'heure de fin d'activité doit être représentée par le point final temporisé d'une barre figurant sur l'axe horizontal.
 - La durée d'activité doit être représentée par la longueur d'une barre.
- Une relation prédécesseur/successeur entre des activités doit être représentée par une flèche.

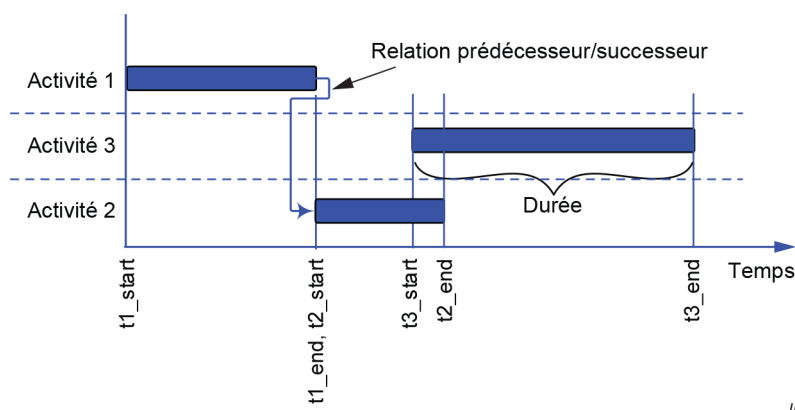


Figure 7 – Informations données par des diagrammes de Gantt

6.3 Réseaux de représentation nodale

6.3.1 Généralités

Les réseaux de représentation nodale sont utilisés pour décrire et analyser les relations temporelles et causales d'un ensemble d'activités interdépendantes. La relation fin/début entre les nœuds indique qu'un nœud peut uniquement commencer après la fin du nœud précédent. Les réseaux de représentation nodale ne donnent aucun moyen de modéliser les alternatives et les cycles.

6.3.2 Éléments graphiques

Dans un réseau de représentation nodale, les éléments graphiques suivants doivent être utilisés (voir la Figure 8):

- aucun ou plusieurs nœuds, représentés par des rectangles, chacun étant divisé en sept secteurs;
- aucune ou plusieurs flèches orientées.

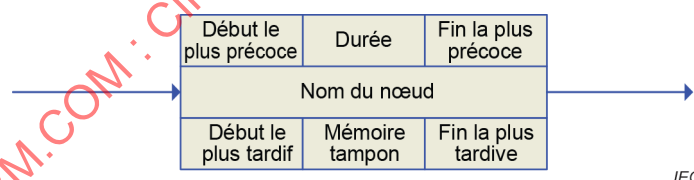


Figure 8 – Éléments de modèle de réseaux de représentation nodale

6.3.3 Structure nodale

La structure suivante d'un nœud doit être utilisée:

- Il convient que le début le plus précoce (t_{es}) d'un nœud soit indiqué par un nombre dans le cadre situé dans le coin supérieur gauche du nœud.
- Il convient que la fin la plus précoce (t_{ee}) d'un nœud soit indiquée par un nombre dans le cadre situé dans le coin supérieur droit du nœud.
- Il convient que la durée (t_d) d'un nœud soit indiquée par un nombre dans le cadre situé dans la partie centrale supérieure du nœud et qu'elle soit calculée par le début et la fin les plus précoces du nœud à l'aide de la formule $t_d = t_{ee} - t_{es}$
- Il convient que le nom du nœud soit placé dans le cadre situé au centre du nœud.
- Il convient que le début le plus tardif (t_{ls}) d'un nœud soit indiqué par un nombre dans le cadre situé dans le coin inférieur gauche du nœud.
- Il convient que la fin la plus tardive (t_{le}) d'un nœud soit indiquée par un nombre dans le cadre situé dans le coin inférieur droit du nœud.

- Il convient que la valeur de mémoire tampon (t_b) d'un nœud soit indiquée par un nombre dans le cadre situé dans la partie centrale inférieure du nœud et qu'elle soit calculée par la fin la plus précoce et la fin la plus tardive du nœud à l'aide de la formule $t_b = t_{le} - t_{ee}$.

6.3.4 Structure de réseau

La structure suivante d'un réseau de représentation nodale doit être utilisée:

- Les nœuds peuvent être reliés entre eux par des flèches.
- Une flèche doit être reliée entre le côté droit d'un nœud et le côté gauche d'un autre nœud.

6.3.5 Informations concernant la logique

Les réseaux de représentation nodale doivent contenir les types suivants d'informations concernant la logique:

- Informations relatives à l'activité:
 - nom;
 - heure de début la plus précoce;
 - heure de début la plus tardive;
 - heure de fin la plus précoce;
 - heure de fin la plus tardive;
 - durée;
 - mémoire tampon.
- Relation prédécesseur/successeur entre les activités.

6.3.6 Informations concernant la logique dans les réseaux de représentation nodale

Pour la représentation des informations concernant la logique dans un réseau de représentation nodale, les dispositions suivantes s'appliquent (voir la Figure 9):

- Une activité doit être représentée par un nœud.
 - Un nom d'activité doit être représenté par le nom du nœud.
 - L'heure de début la plus précoce de l'activité doit être représentée par le début le plus précoce d'un nœud.
 - L'heure de début la plus tardive de l'activité doit être représentée par le début le plus tardif d'un nœud.
 - L'heure de fin la plus précoce de l'activité doit être représentée par la fin la plus précoce d'un nœud.
 - L'heure de fin la plus tardive de l'activité doit être représentée par la fin la plus tardive d'un nœud.
 - La durée de l'activité doit être représentée par la durée d'un nœud.
 - Le délai d'activité doit être représenté par la mémoire tampon d'un nœud.
- Une relation prédécesseur/successeur entre des activités doit être représentée par une flèche.

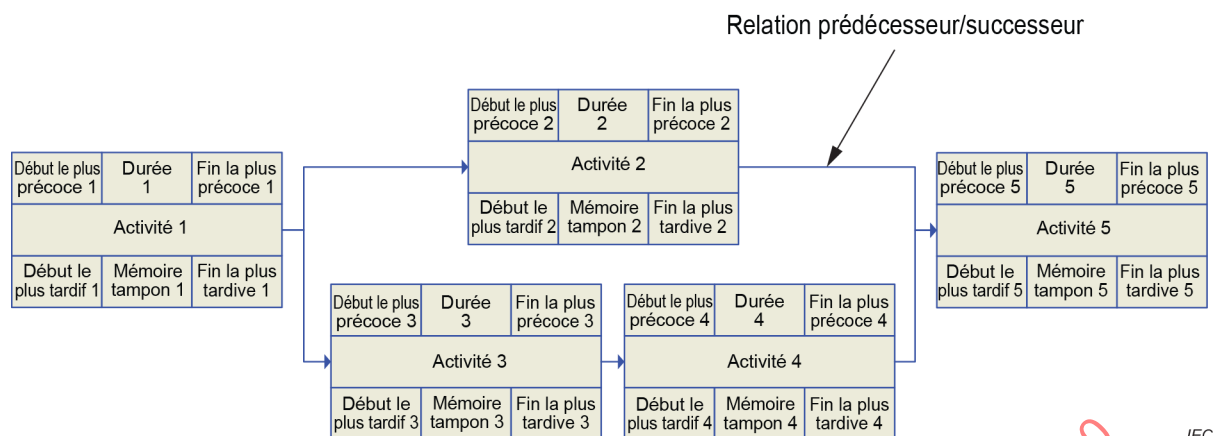


Figure 9 – Informations fournies par des réseaux de représentation nodale d'activité

6.4 Chronogrammes

6.4.1 Généralités

Les chronogrammes sont utilisés pour décrire les relations causales et temporelles entre les signaux et les états du système. De plus, ils décrivent dans quelle mesure les signaux ont un impact sur les changements d'état du système et la manière dont ils sont générés par les changements d'état. Les chronogrammes permettent de modéliser des alternatives.

6.4.2 Éléments graphiques

Dans un chronogramme, les éléments graphiques suivants doivent être utilisés (voir la Figure 10):

- un ou plusieurs couloirs représentés par des colonnes;
- un ou plusieurs niveaux représentés par des colonnes;
- aucune ou plusieurs flèches orientées et nommées;
- une ligne chronologique (axe horizontal) présentée sous la forme d'une horloge globale;
- une ou plusieurs lignes de vie.

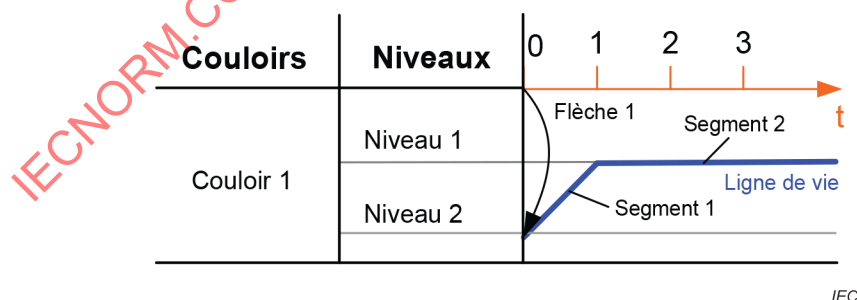


Figure 10 – Éléments de modèle des chronogrammes

6.4.3 Structure du diagramme

La structure suivante d'un chronogramme doit être utilisée:

- Chaque niveau doit avoir sa propre bande horizontale. Les noms des niveaux doivent être placés dans la colonne des niveaux correspondante.
- Les niveaux qui appartiennent à un couloir doivent être regroupés dans la colonne des couloirs. Les noms des couloirs doivent être placés dans la colonne des couloirs correspondante. Un couloir doit être représenté par au moins un niveau.

- Chaque couloir doit avoir une ligne de vie, représentant le changement de niveau sur une échelle de temps linéaire.
- Une ligne de vie doit être composée d'au moins un segment.
- Un segment doit être représenté par une ligne droite.
- Une flèche doit relier:
 - La ligne chronologique avec le point de départ d'un segment,
 - Un point final d'un segment avec le point de départ de l'autre segment dans la même ligne de vie ou une autre ligne de vie.
- Le point final d'un segment doit être le point de départ d'aucune ou de plusieurs flèche(s).
- Aucune ou plusieurs flèches doivent commencer en un point de la ligne chronologique.

6.4.4 Informations concernant la logique

Les chronogrammes doivent contenir les types suivants d'informations concernant la logique:

- Informations relatives aux ressources:
 - nom.
- Informations relatives à l'état:
 - nom;
 - phase (représente la durée);
 - flux d'état.
- Informations relatives au signal:
 - nom;
 - deux types différents: heure et signal de ressource;
 - point dans le temps où un signal se produit;
 - influence d'un signal sur l'état d'une ressource.

6.4.5 Informations concernant la logique dans les chronogrammes

Pour la représentation des informations concernant la logique dans un chronogramme, les dispositions suivantes s'appliquent (voir la Figure 11):

- Une ressource doit être représentée par un couloir.
 - Un nom de ressource doit être représenté par le nom du couloir.
- Un état doit être représenté par un niveau.
 - Un nom d'état doit être représenté par le nom du niveau.
 - La variation de l'état dans le temps doit être présentée sous la forme d'une ligne de vie.
 - Une phase doit être représentée par un segment.
- Un signal doit être représenté par une flèche.
 - Un nom de signal doit être représenté par le nom de la flèche.
 - Un signal de temps doit être représenté par une flèche qui part d'un point sur la ligne chronologique. Un signal de ressource doit être représenté par une flèche qui part de la fin d'un segment d'une ligne de vie.
 - Un point dans le temps où un signal se produit doit correspondre à l'axe x .
 - L'influence d'un signal sur l'état d'une ressource doit être représentée par un nouveau départ de segment de la ligne de vie correspondante.

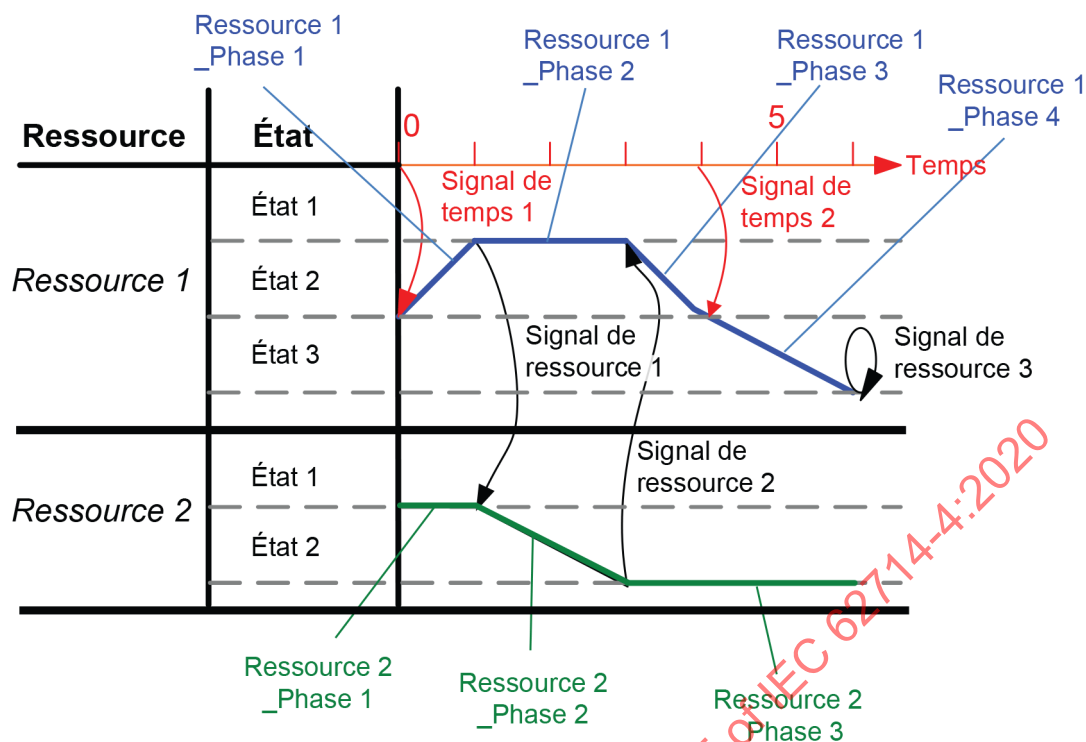


Figure 11 – Informations données par des chronogrammes

6.5 Diagrammes fonctionnels en séquence

Les diagrammes fonctionnels en séquence (SFC) décrivent les dépendances causales et temporelles de séquences des étapes et transitions du système. Les séquences complexes peuvent être modélisées avec des boucles et une exécution conditionnelle.

Le présent document doit utiliser les diagrammes fonctionnels en séquence comme cela est spécifié dans l'IEC 61131-3.

6.6 Diagrammes de bloc fonctionnel

Les diagrammes de bloc fonctionnel (FBD) décrivent les réseaux logiques de blocs fonctionnels, respectivement la relation fonctionnelle des variables d'entrée et de sortie.

Le présent document doit utiliser les diagrammes de bloc fonctionnel comme cela est spécifié dans l'IEC 61131-3. L'utilisation des blocs fonctionnels spécifiés dans l'IEC 61131-3 doit être admise.

6.7 Expression mathématique

Les expressions mathématiques sont une disposition de nombres, de valeurs absolues ou de lettres pour exprimer des relations mathématiques, physiques, chimiques ou logiques. Une expression mathématique est composée d'une formule ou d'un système d'équations.

Pour définir des expressions mathématiques, le présent document doit utiliser la partie de contenu du langage MathML (*Mathematical Markup Language*) version 2.0.

7 Description du schéma XML de logique AML

7.1 Vue d'ensemble du schéma

7.1.1 Utilisation du schéma IEC 61131-10

Le présent paragraphe décrit le "schéma XML de logique AML", qui est formellement défini à l'Annexe E.

Le schéma XML de logique AML importe le schéma IEC 61131-10 et réutilise en partie ses définitions de type pour représenter des modèles logiques à l'aide de variables, de blocs fonctionnels, de diagrammes fonctionnels en séquence, etc.

Le schéma IEC 61131-10 prend en charge deux mécanismes d'extension permettant d'intégrer des informations dans le format d'échange. Le premier mécanisme est l'élément "AddData", qui peut être imbriqué sous différents autres éléments. Son usage est destiné à l'extension d'autres éléments, par lequel ils sont enrichis avec des informations, mais qui n'est pas essentiel à l'échange de l'élément entre les outils. Le second mécanisme utilise une technique de schéma XML appelée "substitution de type abstrait". Un nouveau type d'élément peut être déduit du type abstrait et utilisé en remplacement du type d'élément dont le type abstrait est spécifié.

Un schéma XML de logique AML répond naturellement tant aux mécanismes d'extension du schéma IEC 61131-10 qu'aux conventions de dénomination. La vue d'ensemble de la structure du schéma est représentée à la

Figure 12 sous la forme d'un diagramme de classe UML. Les classes dont le nom commence par le préfixe "IEC61131_10_Ed1_0" représentent les types importés du schéma IEC 61131-10. De même, les classes dont le nom est en italique (également tracé en jaune clair) représentent les types abstraits qui doivent être remplacés par l'un de ses types non abstraits étendus.

- L'intégralité de la déclaration de type de données définies par l'utilisateur et de sa référence à la déclaration de variable est réutilisée par le schéma XML de logique AML, comme le représente la Figure 12.
- Les descriptions logiques et leur type de base de l'IEC 61131-10 sont également réutilisés par le schéma XML de logique AML, comme le représente la Figure 12. Par conséquent, toutes les descriptions logiques de l'IEC 61131-10 sont disponibles dans le schéma XML de logique AML. De plus, le schéma XML de logique AML ajoute ses modèles logiques spécifiques (IML et les expressions mathématiques, par exemple) par "substitution de type abstrait", qui est le second mécanisme d'extension issu de l'IEC 61131-10.
- Un schéma XML de logique AML étend les éléments SFC essentiels de l'IEC 61131-10 pour représenter la couche de modélisation intermédiaire (IML) décrite en 5.3, comme le représente la Figure 12.

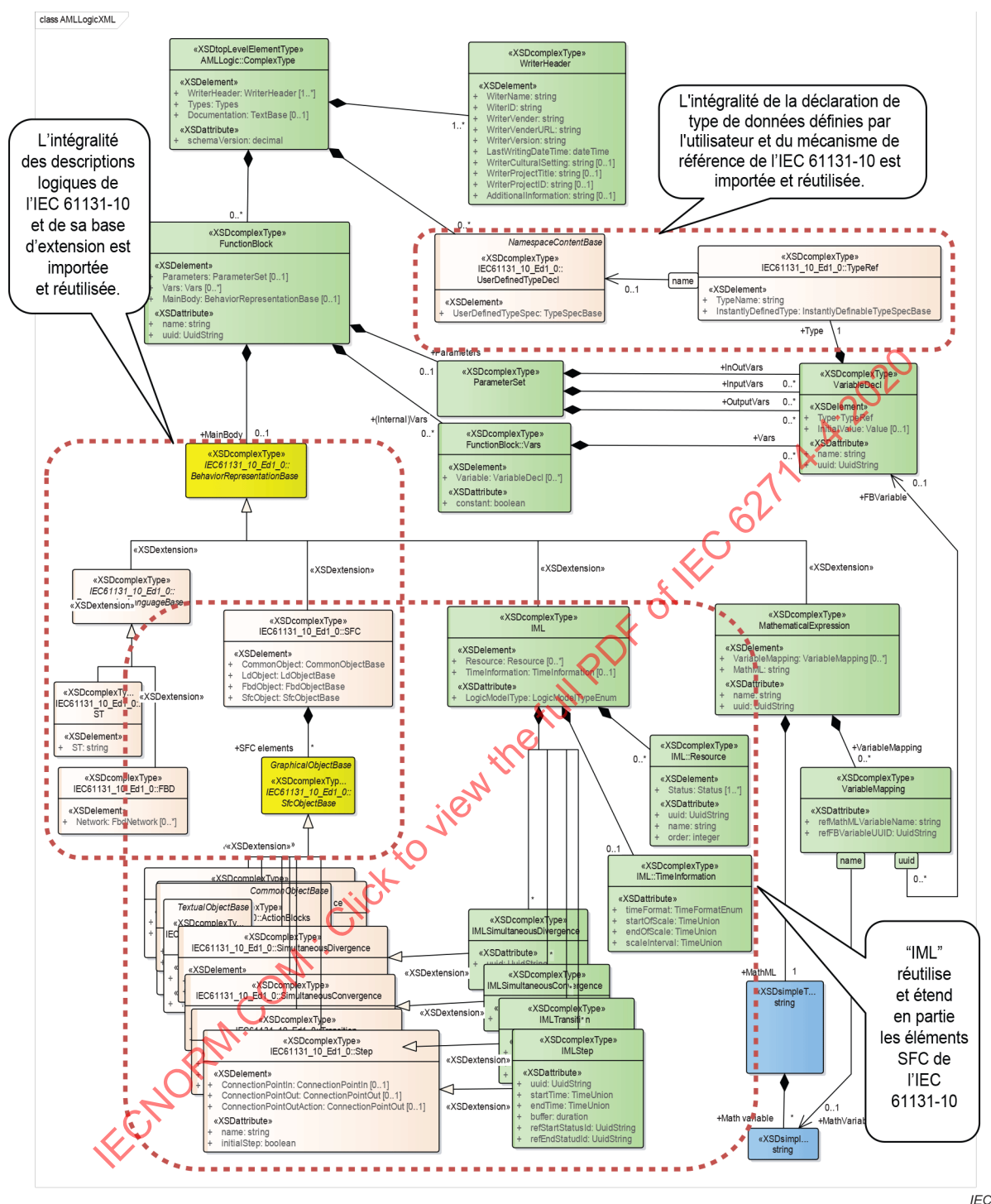


Figure 12 – Vue d'ensemble du schéma de logique AML

7.1.2 Versionnage du schéma

La définition du schéma XML de logique AML comporte un attribut de version. Son premier numéro est le numéro d'édition de l'IEC 62174-4. Le second numéro est réservé en cas de modifications mineures dans la même édition. Le nom de fichier du schéma reflète également le numéro d'édition.

De plus, l'élément racine "AMLLogic" a un attribut "schemaVersion". Lors de la création d'un document XML reposant sur le schéma, son numéro de version doit être défini sur l'attribut.

7.2 Élément racine "AMLLogic"

7.2.1 Généralités

L'élément racine "AMLLogic" représente un type de projet ou de bibliothèque contenant des informations logiques. Son contenu est représenté à la Figure 13.

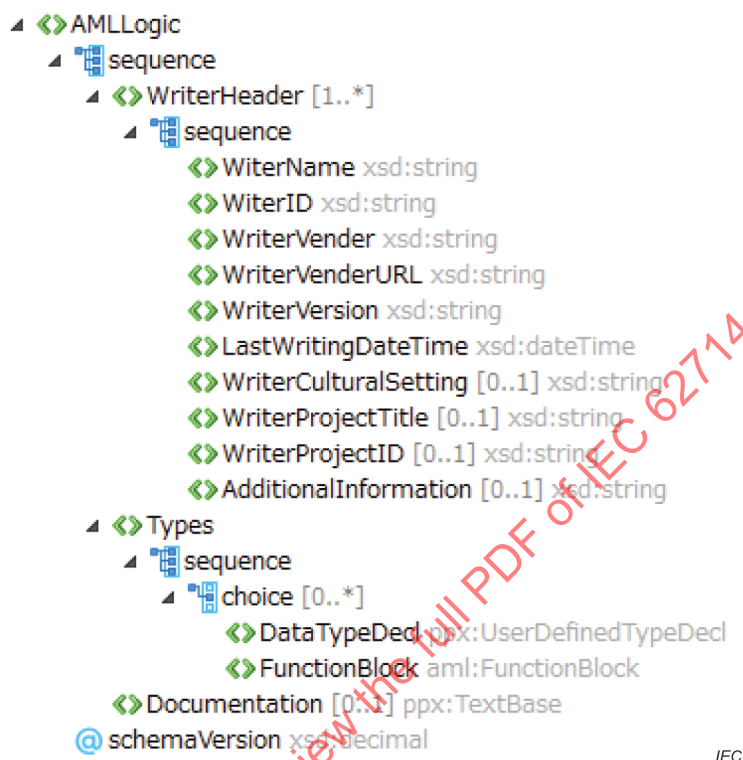


Figure 13 – Élément racine "AMLLogic"

7.2.2 Attributs

L'attribut "schemaVersion" doit contenir le numéro de version du schéma XML de logique AML auquel le document XML est conforme.

7.2.3 Sous-élément "WriterHeader"

L'élément "WriterHeader" représente les méta-informations relatives à l'outil qui écrit le document XML, comme l'explique l'Article 9).

7.2.4 Sous-élément "Types"

L'élément "Types" est un paramètre fictif pour la déclaration de type de données définies par l'utilisateur et/ou la déclaration de bloc fonctionnel.

L'élément "DataTypeDecl" du type "pplx:UserDefinedTypeDecl" représente la déclaration de type de données définies par l'utilisateur de la spécification de langage IEC 61131-3. Par conséquent, tous les types de données disponibles dans la spécification de langage IEC 61131-3 peuvent également être définis dans le schéma XML de logique AML. Il convient de citer l'IEC 61131-10 en référence pour la spécification particulière du type "pplx:UserDefinedTypeDecl".

L'élément "FunctionBlock" du type "aml:FunctionBlock" représente un type de bloc fonctionnel spécialisé pour la description de la logique AML, dont seules les caractéristiques nécessaires du bloc fonctionnel IEC 61131-3 sont prises en charge. Son contenu est expliqué en 7.3.

7.2.5 Sous-élément "Documentation"

L'élément "Documentation" représente une description générale de l'élément racine "AMLLogic". Le type "ppx: TextBase" est un type abstrait importé de l'IEC 61131-10, et qui est remplacé par le texte en clair du type "ppx: SimpleText" par défaut.

7.3 Type complexe "FunctionBlock"

7.3.1 Généralités

Le type complexe "FunctionBlock" (présenté à la Figure 14) représente un type de bloc fonctionnel spécialisé pour la description de la logique AML, dont seules les caractéristiques nécessaires du bloc fonctionnel IEC 61131-3 sont prises en charge.

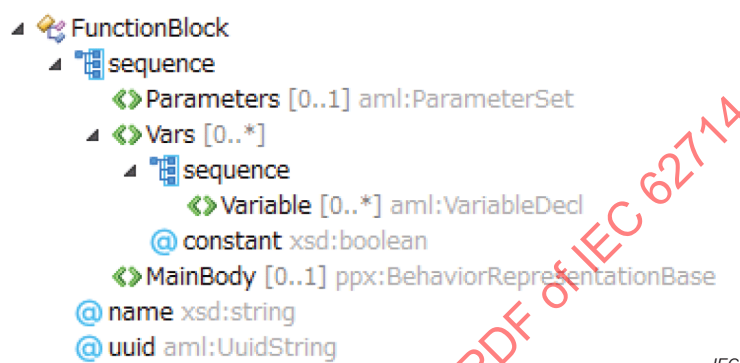


Figure 14 – Type complexe "FunctionBlock"

7.3.2 Attributs

L'attribut "name" de "xsd:string" représente le nom de type du bloc fonctionnel.

L'attribut "uuid" de "aml:UuidString" représente l'identifiant unique universel pour la définition de type du bloc fonctionnel. Le type simple "UuidString" est expliqué en 7.9.

7.3.3 Sous-élément "Parameters"

L'élément "Parameters" du type "aml:ParameterSet" représente la déclaration d'entrée/de sortie/d'entrée-sortie du bloc fonctionnel. Le type "ParameterSet" est expliqué en détail en 7.3.6.

7.3.4 Sous-élément "Vars"

L'élément "Vars" représente les déclarations de variable interne du bloc fonctionnel.

L'élément "Variable" du type "aml:VariableDecl" représente la déclaration de variable. Le type complexe "aml:VariableDecl" est expliqué en 7.3.7.

L'attribut "constant" indique que les déclarations de variable dans l'élément <Vars> sont des variables à valeurs constantes.

7.3.5 Sous-élément "MainBody"

L'élément "MainBody" du type "ppx:BehaviorRepresentationBase" représente un modèle logique qui doit être évalué lors de l'exécution du bloc fonctionnel. Un type complexe non abstrait déduit de "ppx:BehaviorRepresentationBase" peut être remplacé par le type de l'élément. Par conséquent, tous les modèles logiques du schéma IEC 61131-10 sont également disponibles dans ce bloc fonctionnel du schéma XML de logique AML, tels que le

diagramme fonctionnel en séquence (SFC), le diagramme de bloc fonctionnel (FBD), le diagramme à contacts (LD - *ladder diagram*) et le texte structuré (ST - *structured text*).

7.3.6 Type complexe "ParameterSet"

Le type complexe "ParameterSet" (comme le représente la Figure 15) représente un ensemble de déclarations de variables d'entrée, de sortie et d'entrée/sortie.

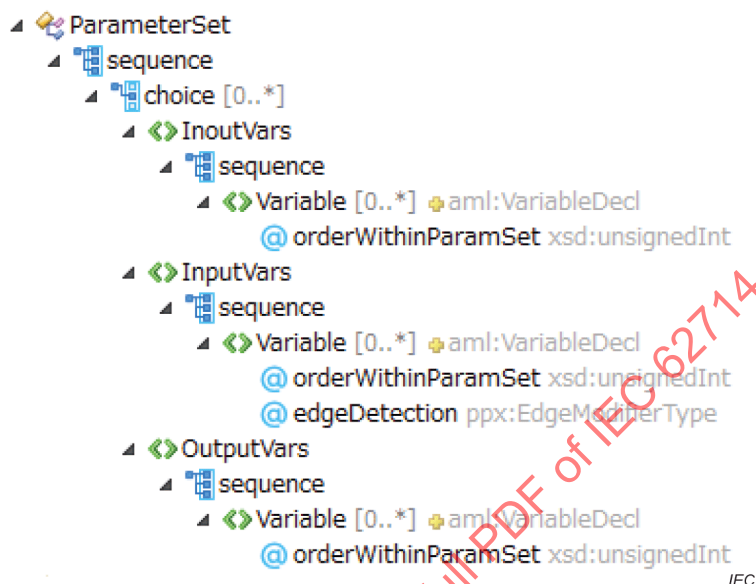


Figure 15 – Type complexe "ParameterSet"

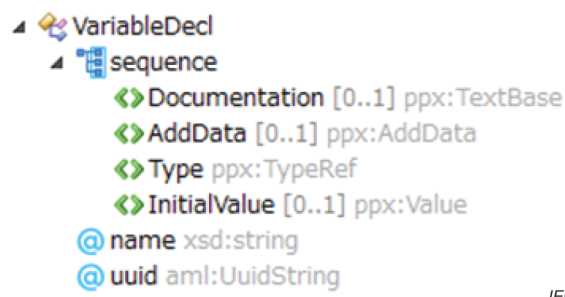
L'élément "InoutVars" représente la section de déclaration des variables d'entrée/sortie. En ce qui concerne l'élément "Variable" dans "InoutVars", son type est étendu à partir de "aml:VariableDecl" qui est expliqué en 7.3.7, et l'attribut "orderWithinParamSet" ajouté. Cette extension se justifie par le fait que l'ordre des paramètres d'un bloc fonctionnel est significatif, alors que l'ordre d'apparition des éléments du même type d'élément ne l'est pas dans un document XML.

L'élément "InputVars" représente la section de déclaration des variables d'entrée du bloc fonctionnel. En ce qui concerne l'élément "Variable" dans "InputVars", son type est étendu à partir de "aml:VariableDecl" également expliqué en 7.3.7, et l'attribut "edgeDetection" est ajouté en plus de l'attribut "orderWithinParamSet".

L'élément "OutputVars" représente la section de déclaration des variables de sortie du bloc fonctionnel. En ce qui concerne l'élément "Variable" dans "OutputVars", son type est étendu à partir de "aml:VariableDecl" également expliqué en 7.3.7, et l'attribut "orderWithinParamSet" est également ajouté.

7.3.7 Type complexe "VariableDecl"

Le type complexe "VariableDecl" (comme le représente la Figure 16) représente la section de déclaration de variable.



IEC

Figure 16 – Type complexe "VariableDecl"

L'élément "Documentation" du type "ppx:TextBase" représente la description de la variable.

L'élément "AddData" du type "ppx:AddData" peut être utilisé pour archiver tous les types de données supplémentaires.

L'élément "Type" du type "ppx:TypeRef" représente la référence à la définition de type de la variable. Il peut s'agir du nom de type ou de la déclaration d'instant telle que "ARRAY [0..8] OF INT". Comme pour les détails du type complexe "ppx:TypeRef", il convient de citer l'IEC 61131-10 en référence.

L'élément "InitialValue" du type "ppx:Value" représente la valeur initiale de la variable. Il peut s'agir d'une valeur simple, d'une valeur de matrice pour le type "array" ou d'une valeur structurée pour le type de données structurées en fonction du type de données de la variable. Comme pour les détails du type complexe "ppx:Value", il convient de citer la spécification IEC 61131-10 en référence.

L'attribut "name" de "xsd:string" représente le nom de la variable.

L'attribut "uuid" de "aml:UuidString" représente l'identifiant unique universel pour la variable. Le type simple "UuidString" est expliqué en 7.9.

7.4 Type complexe "IML"

7.4.1 Généralités

Le type complexe "IML" (comme le représente la Figure 17) représente la couche de modélisation intermédiaire pour plusieurs modèles logiques particuliers, comme décrit en 5.3. Ce type complexe peut être utilisé pour remplacer le type de l'élément "MainBody" du bloc fonctionnel dans le schéma XML de logique AML, étant donné qu'il étend "ppx:BehaviorRepresentationBase", comme le décrit la Figure 12

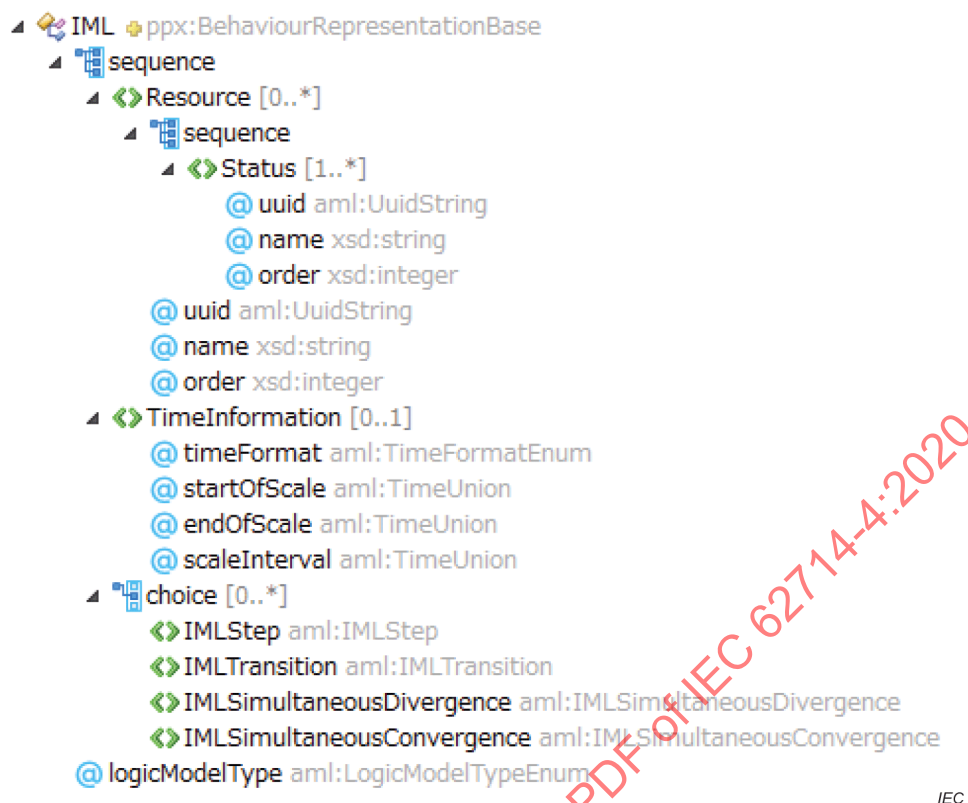


Figure 17 – Type complexe "IML"

7.4.2 Attributs

L'attribut "logicModelType" du type "aml:LogicModelTypeEnum" représente le type de modèle logique représenté avec l'IML. Le type simple "LogicModelTypeEnum" est expliqué en 7.6.

7.4.3 Sous-élément "Resource"

L'élément "Resource" représente les informations supplémentaires concernant la logique nécessaires pour le chronogramme, comme cela est décrit en 6.4.4.

- L'attribut "uuid" représente l'identifiant unique universel pour l'élément "resource". Le type simple "UuidString" est expliqué en 7.9.
- L'attribut "name" représente le nom de la ressource.
- L'attribut "order" représente l'ordre de la ressource dans l'IML. Il convient de répertorier les petits numéros en premier.

Les éléments "Status" représentent les états possibles de l'élément "Resource".

- L'attribut "uuid" représente l'identifiant unique universel pour l'élément "statuts". Le type simple "UuidString" est expliqué en 7.9.
- L'attribut "name" représente le nom de l'état.
- L'attribut "order" représente l'ordre de l'état dans la ressource. Il convient de répertorier les petits numéros en premier.

7.4.4 Sous-élément "TimeInformation"

"TimeInformation" représente l'échelle de temps logique, en particulier pour le diagramme de Gantt ou le chronogramme.

- L'attribut "timeFormat" du type "aml:TimeFormatEnum" représente le format de l'expression de temps dans cette IML. Le type simple "aml:TimeFormatEnum" est expliqué en 7.8.
- L'attribut "startOfScale" du type "aml:TimeUnion" représente le point de départ de l'échelle de temps au format spécifié par l'attribut "timeFormat" ci-dessus. Le type simple "aml:TimeUnion" est expliqué en 7.7.
- L'attribut "endOfScale" du type "aml:TimeUnion" représente le point d'arrivée de l'échelle de temps au format spécifié par l'attribut "timeFormat" ci-dessus. Le type simple "aml:TimeUnion" est expliqué en 7.7.
- L'attribut "scaleInterval" du type "aml:TimeUnion" représente l'échelon de l'échelle de temps au format spécifié par l'attribut "timeFormat" ci-dessus. Le type simple "aml:TimeUnion" est expliqué en 7.7.

7.4.5 Choix de l'élément "IML****"

"IMLStep", "IMLTransition", "IMLSimultaneousDivergence" ou "SimultaneousConvergence" peut être choisi comme élément "choice". Cette structure de schéma est fondamentalement la même que le type complexe "SFC" de l'IEC 61131-10. Toutefois, les alternatives à ce choix sont limitées à ce qui est nécessaire pour la représentation IML. De même, le type de chaque élément alternatif est étendu depuis l'élément original de l'IEC 61131-10.

7.4.6 Type complexe "IMLStep"

Le type complexe "IMLStep" (comme le représente la Figure 18) représente une phase ou un état d'une activité. Il étend "ppx:Step" de l'IEC 61131-10, qui représente une ÉTAPE dans le diagramme fonctionnel en séquence.

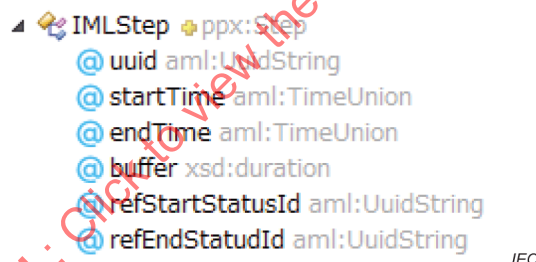


Figure 18 – Type complexe "IMLStep"

Les attributs suivants sont ajoutés comme extension au type de base "ppx:Step" de l'IEC 61131-10.

- L'attribut "uuid" de "aml:UuidString" représente l'identifiant unique universel pour l'élément IMLStep. Le type simple "UuidString" est expliqué en 7.9.
- L'attribut "startTime" de "aml:TimeUnion" représente l'heure de début de la phase de l'activité qui correspond à ce type simple IMLStep. "aml:TimeUnion" est expliqué en 7.7. L'heure doit être représentée au format spécifié par l'attribut "timeFormat" de l'élément "TimeInformation" expliqué en 7.4.4.
- L'attribut "endTime" de "xsd:duration" représente l'heure de fin de la phase de l'activité qui correspond à cet IMLStep. Le type simple "aml:TimeUnion" est expliqué en 7.7. L'heure doit être représentée au format spécifié par l'attribut "timeFormat" de l'élément "TimeInformation" expliqué en 7.4.4.
- L'attribut "buffer" de "xsd:duration" représente une marge pour le laps de temps entre "startTime" et "endTime".
- L'attribut "refStartStatusId" de "aml:UuidString" représente une référence à l'état source du segment de la pseudo forme d'onde, que l'élément "IMLStep" représente si l'IML représente le chronogramme. Concrètement, il représente la valeur de l'attribut "uuid" d'un certain élément "Status" avec un certain élément "Resource" dans l'élément "IML".

- L'attribut "refEndStatusId" de "aml:UuidString" représente une référence à l'état de destination du segment de la pseudo forme d'onde que les éléments "IMLStep" représentent si l'IML représente le chronogramme. Concrètement, il représente la valeur de l'attribut "uuid" d'un certain élément "Status" avec un certain élément "Resource" dans l'élément "IML".

7.4.7 Type complexe "IMLTransition"

Le type complexe "IMLTransition" (comme le représente la Figure 19) représente un type de relation prédécesseur/successeur entre des éléments "IMLStep". Il étend "ppx:Transition" de l'IEC 61131-10, qui représente une TRANSITION dans le diagramme fonctionnel en séquence.



Figure 19 – Type complexe "IMLTransition"

Les attributs suivants sont ajoutés comme extension au type de base "ppx:Transition" de l'IEC 61131-10.

- L'attribut "uuid" de "aml:UuidString" représente l'identifiant unique universel pour l'élément IMLTransition. Le type simple "UuidString" est expliqué en 7.9.
- L'attribut "refTime" représente l'heure à laquelle il convient de déclencher cette transition. Le type simple "aml:TimeUnion" est expliqué en 7.7. L'heure doit être représentée au format spécifié par l'attribut "timeFormat" de l'élément "TimeInformation" expliqué en 7.4.4.
- L'attribut "refEndSegmentId" du type "aml:uuisString" représente une référence à l'élément IMLStep par l'UUID dont il convient que la fin déclenche cette transition.

7.4.8 Type complexe "IMLSimultaneousDivergence"

Le type complexe "IMLSimultaneousDivergence" étend "ppx:SimultaneousDivergence" de l'IEC 61131-10 (comme le représente la Figure 20), qui représente un point de départ de la branche parallèle dans le diagramme fonctionnel en séquence.

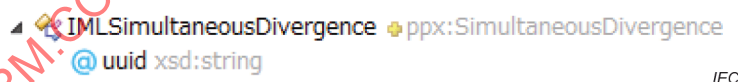


Figure 20 – Type complexe "IMLSimultaneousDivergence"

Les attributs suivants sont ajoutés comme extension au type de base "ppx:SimultaneousDivergence" de l'IEC 61131-10.

- L'attribut "uuid" de "aml:UuidString" représente l'identifiant unique universel pour l'élément IMLSimultaneousDivergence. Le type simple "UuidString" est expliqué en 7.9.

7.4.9 Type complexe "IMLSimultaneousConvergence"

Le type complexe "IMLSimultaneousConvergence" étend "ppx:SimultaneousConvergence" de l'IEC 61131-10 (comme le représente la Figure 21), qui représente un point final de la branche parallèle dans le diagramme fonctionnel en séquence.



IEC

Figure 21 – Type complexe "IMLSimultaneousConvergence"

Les attributs suivants sont ajoutés comme extension au type de base "pplx:SimultaneousConvergence" de l'IEC 61131-10.

- L'attribut "uuid" de "aml:UuidString" représente l'identifiant unique universel pour l'élément IMLSimultaneousConvergence. Le type simple "UuidString" est expliqué en 7.9.

7.5 Type complexe "MathematicalExpression"

7.5.1 Généralités

Le type complexe "MathematicalExpression" (comme le représente la Figure 22) représente la description logique dans une formule mathématique, comme décrit en 6.7. Il peut être utilisé pour remplacer l'élément "MainBody" du bloc fonctionnel dans le schéma XML de logique AML, étant donné qu'il étend le type complexe abstrait "pplx:BehaviorRepresentationBase", comme le décrit la Figure 12.



IEC

Figure 22 – Type complexe "MathematicalExpression"

7.5.2 Attributs

L'attribut "name" du type "string" représente le nom de l'expression mathématique.

L'attribut "uuid" représente l'identifiant unique universel pour l'expression mathématique. Le type simple "UuidString" est expliqué en 7.9.

7.5.3 Sous-élément "VariableMapping"

L'élément "VariableMapping" représente le mapping entre la variable mathématique écrite en langage MathML et la variable déclarée dans le bloc fonctionnel auquel appartient ce modèle logique.

"refMathMLVariableName" représente le nom de la variable définie en langage MathML.

"refFBVariableUUID" représente l'UUID de la variable déclarée dans le même bloc fonctionnel. Le type simple "UuidString" est expliqué en 7.9.

L'attribut "direction" représente la direction du mapping avec la chaîne émulée qui ne permet que "In" ou "Out".

7.5.4 Sous-élément "MathML"

L'élément "MathML" du type "string" représente les données de texte XML de MathML. Il est archivé en texte clair. L'analyse syntaxique avec le schéma MathML ne relève pas du domaine d'application du schéma XML de logique AML.

7.6 Type simple "LogicModelTypeEnum"

Le type simple "LogicModelTypeEnum" est le type "string" restreint qui admet uniquement l'une des chaînes énumérées, qui sont "GanttChart", "ActivityOnNodeNetwork", "TimingDiagram" ou "Unkown". Il est utilisé pour spécifier le type de modèle logique que représente l'IML.

7.7 Type simple "TimeUnion"

Le type simple "TimeUnion" est une union dont les membres sont composés des types simples temporels intégrés dans le schéma XML. La spécification de schéma XML doit être référencée pour la spécification détaillée du format de valeur de chaque type intégré.

- xsd:duration
- Le format de valeur doit être "PnYnMnDTnHnMnS", par exemple "P10DT15H" (10 jours et 15 heures), "PT1M20S" (1 min et 20 s), "PT95.4" (95,4 s), etc.
- xsd:time
- Le format de valeur doit être "hh:mm:ss", par exemple simplement "15:45:30" ou "15:45:30+02:00" pour représenter le décalage horaire par rapport à UTC.
- xsd:dateTime
- Le format de valeur doit être "YYYY-MM-DDThh:mm:ss", par exemple simplement "2017-05-30T09:00:00" ou "2017-05-30T09:00:00+02:00" pour représenter le décalage horaire par rapport à UTC.
- xsd:date
- Le format de valeur doit être "YYYY-MM-DD", par exemple "2017-09-24".

7.8 Type simple "TimeFormatEnum"

Le type simple "TimeFormatEnum" est le type "string" restreint qui admet uniquement l'une des chaînes énumérées, qui sont "xsd:duration", "xsd:time", "xsd:dateTime" ou "xsd:date". Il est utilisé pour spécifier le format horaire qui doit être utilisé pour le type "aml:TimeUnion" expliqué en 7.7.

7.9 Type simple "UuidString"

Le type simple "UuidString" et le type "string" restreint pour représenter UUID sous la forme d'un texte. Aucune restriction syntaxique n'est actuellement définie.

Il est recommandé de mettre en œuvre l'UUID en tant que GUID (comme cela est spécifié dans le RFC 4122).

8 Archivage de modèles logiques

8.1 Généralités

Le présent article définit les règles d'archivage des modèles logiques dans l'IML. Une représentation textuelle dans l'IML est présentée pour le mapping de chaque élément de modèle logique. Pour des exemples, voir l'Annexe A.

8.2 Archivage des diagrammes de Gantt dans un document XML de logique AML

8.2.1 Règles communes

Pour archiver un diagramme de Gantt dans un document XML de logique AML, les dispositions suivantes s'appliquent:

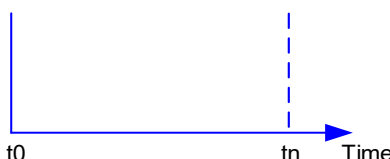
- Un diagramme de Gantt doit être mappé à un élément logique IML de type "GanttChart" selon les dispositions suivantes.
- L'IML doit commencer par l'élément "TimeInformation", qui définit la ligne chronologique du diagramme de Gantt (voir 8.2.2).
- Chaque barre de Gantt doit être représentée par un IMLStep (voir 8.2.3).
- Chaque flèche de Gantt doit être représentée par un IMLTransition (voir 8.2.4).
- Pour les relations prédécesseur/successeur dans un diagramme de Gantt, les dispositions suivantes s'appliquent:
 - Si une barre de Gantt a au moins deux barres de successeur, elle doit y être reliée par des éléments IMLTransition et IMLSimultaneousDivergence (voir 8.2.5).
 - Si une barre de Gantt a au moins deux prédécesseurs, elle doit y être reliée par des éléments IMLSimultaneousConvergence et IMLTransition (voir 8.2.6).
 - En l'absence de relation explicite entre les barres de Gantt (c'est-à-dire qu'elles sont concurrentes), l'ordre de l'IMLStep associé doit être parallèle et elles ne doivent présenter aucune relation ordonnée reposant sur les séquences de transition pas à pas (voir 8.2.5 et 8.2.6).

Si des relations prédécesseur/successeur entre des barres Gantt sont définies, la structure obtenue doit être une séquence d'éléments IMLStep et IMLTransition (voir 8.2.5 et 8.2.6).

8.2.2 Archivage du début d'un diagramme de Gantt

Le Tableau 2 spécifie le mapping des propriétés et des relations du début d'un diagramme de Gantt à des éléments IML.

Tableau 2 – Mapping du début d'un diagramme de Gantt à un élément IML

Type	Représentation				
Élément de diagramme de Gantt	 <table border="1"> <thead> <tr> <th>Anglais</th><th>Français</th></tr> </thead> <tbody> <tr> <td>Time</td><td>Temps</td></tr> </tbody> </table>	Anglais	Français	Time	Temps
Anglais	Français				
Time	Temps				
Élément IML	<pre> <xsd:element name="TimeInformation" minOccurs="0"> <xsd:complexType> <xsd:attribute name="timeFormat" type="aml:TimeFormatEnum" use="required" /> <!-- Il doit s'agir de la valeur du type simple "TimeFormatEnum" --> <xsd:attribute name="startOfScale" type="aml:TimeUnion" use="optional" default="PT0S"/> <!-- valeur de t0 au format défini par l'attribut "timeFormat" --> <xsd:attribute name="endOfScale" type="aml:TimeUnion" use="optional"/> <!-- valeur de tn au format défini par l'attribut "timeFormat" --> <xsd:attribute name="scaleInterval" type="aml:TimeUnion" use="optional"/> <!-- value of the time scale in the format defined by xsd:duration type--> </xsd:complexType> </xsd:element> </pre>				

8.2.3 Archivage des barres

Le Tableau 3 spécifie le mapping des barres du diagramme de Gantt à des éléments IML.

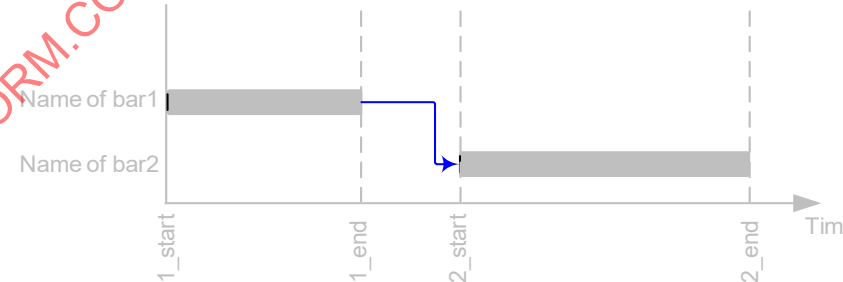
Tableau 3 – Mapping des barres du diagramme de Gantt à des éléments IML

Type	Représentation								
Élément de diagramme de Gantt	 <table border="1"> <thead> <tr> <th>Anglais</th><th>Français</th></tr> </thead> <tbody> <tr> <td>Name of bar</td><td>Nom de la barre</td></tr> <tr> <td>Time</td><td>Temps</td></tr> <tr> <td>bar</td><td>barre</td></tr> </tbody> </table>	Anglais	Français	Name of bar	Nom de la barre	Time	Temps	bar	barre
Anglais	Français								
Name of bar	Nom de la barre								
Time	Temps								
bar	barre								
Élément IML	<pre> <xsd:complexType name="IMLStep"> <xsd:complexContent> <xsd:extension base="ppx:Step"> <xsd:attribute name="uuid" type="aml:UuidString"/> <!-- Il doit s'agir d'une représentation textuelle de l'UUID, qui peut être GUID. --> <xsd:attribute name="name" type="xsd:string" use="required"/> <!-- Désignation "Nom de la barre" dans la représentation textuelle du type xsd:string --> <xsd:attribute name="startTime" type="aml:TimeUnion"/> <!-- valeur de t1 au format de type "aml:TimeUnion " --> <xsd:attribute name="endTime" type="aml:TimeUnion"/> <!-- valeur de t2 au format de type "aml:TimeUnion " --> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre>								

8.2.4 Archivage des flèches

Le Tableau 4 spécifie le mapping d'une barre de diagramme de Gantt comportant au moins une activité de successeur à des éléments IML.

Tableau 4 – Mapping d'une flèche du diagramme de Gantt à des éléments IML

Type	Représentation						
Élément de diagramme de Gantt	 <table border="1"> <thead> <tr> <th>Anglais</th><th>Français</th></tr> </thead> <tbody> <tr> <td>Name of bar</td><td>Nom de la barre</td></tr> <tr> <td>Time</td><td>Temps</td></tr> </tbody> </table>	Anglais	Français	Name of bar	Nom de la barre	Time	Temps
Anglais	Français						
Name of bar	Nom de la barre						
Time	Temps						
Élément IML	<pre> <xsd:complexType name="IMLTransition"> <xsd:complexContent> <xsd:extension base="ppx:Transition"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- Il doit s'agir d'une représentation textuelle de l'UUID, qui peut être GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- ID global du point de connexion extérieur de l'élément "Nom de bar1" --> </xsd:complexType> </xsd:element> </xsd:complexType> </pre>						

Type	Représentation
	<pre> </xsd:element> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--ID global du point de connexion extérieur de l'élément "Nom de bar2" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> </xsd:complexType> </pre>

8.2.5 Archivage de barres de successeur

Le Tableau 5 spécifie le mapping d'une barre de diagramme de Gantt comportant au moins une barre de successeur à des éléments IML.

Tableau 5 – Mapping d'une barre de diagramme de Gantt comportant au moins une barre de successeur à des éléments IML

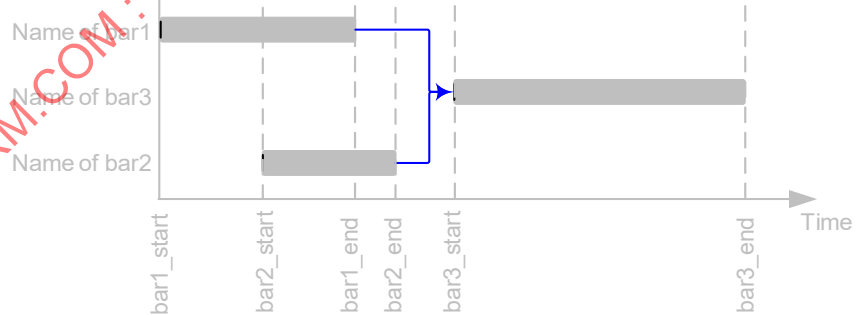
Type	Représentation						
Élément de diagramme de Gantt	<table border="1"> <thead> <tr> <th>Anglais</th><th>Français</th></tr> </thead> <tbody> <tr> <td>Name of bar</td><td>Nom de la barre</td></tr> <tr> <td>Time</td><td>Temps</td></tr> </tbody> </table>	Anglais	Français	Name of bar	Nom de la barre	Time	Temps
Anglais	Français						
Name of bar	Nom de la barre						
Time	Temps						
Élément IML	<pre> <xsd:complexType name="IMLTransition"> <xsd:complexContent> <xsd:extension base="ppx:Transition"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- Il doit s'agir d'une représentation textuelle de l'UUID, qui peut être GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--ID global du point de connexion extérieur de l'élément "Nom de bar1" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--ID global du point de connexion extérieur de l'élément "IMLTransition" --> </xsd:complexType> </xsd:element> </xsd:complexType> <xsd:complexType name="IMLSimultaneousDivergence"> <xsd:complexContent> <xsd:extension base="ppx:IMLSimultaneousDivergence"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> </pre>						

Type	Représentation
	<pre> <!-- Il doit s'agir d'une représentation textuelle de l'UUID, qui peut être GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- ID global du point de connexion extérieur de l'élément "IMLTransition" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- ID global du point de connexion intérieur de l'élément "Nom de bar2" --> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- ID global du point de connexion intérieur de l'élément "Nom de bar3" --> </xsd:complexType> </xsd:element> </xsd:complexType> </pre>

8.2.6 Archivage de barres de prédécesseur

Le Tableau 6 spécifie le mapping d'une barre de diagramme de Gantt comportant au moins une barre de prédécesseur à des éléments IML.

Tableau 6 – Mapping d'une barre de diagramme de Gantt comportant au moins une barre de prédécesseur à des éléments IML

Type	Représentation						
Élément de diagramme de Gantt	 <table border="1" data-bbox="347 1662 1260 1742"> <thead> <tr> <th>Anglais</th><th>Français</th></tr> </thead> <tbody> <tr> <td>Name of bar</td><td>Nom de la barre</td></tr> <tr> <td>Time</td><td>Temps</td></tr> </tbody> </table>	Anglais	Français	Name of bar	Nom de la barre	Time	Temps
Anglais	Français						
Name of bar	Nom de la barre						
Time	Temps						
Élément IML	<pre> <xsd:complexType name="IMLTransition"> <xsd:complexContent> <xsd:extension base="ppx:Transition"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- Il doit s'agir d'une représentation textuelle de l'UUID, qui peut être GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- ID global du point de connexion intérieur de l'élément "IMLTransition" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> </pre>						

Type	Représentation
	<pre> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--ID global du point de connexion intérieur de "Nom de bar3" --> </xsd:complexType> </xsd:element> </xsd:complexType> <xsd:complexType name="IMLSimultaneousConvergence"> <xsd:complexContent> <xsd:extension base="ppx:SimultaneousConvergence"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- Il doit s'agir d'une représentation textuelle de l'UUID, qui peut être GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--ID global du point de connexion intérieur de l'élément "Nom de bar1" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--ID global du point de connexion extérieur de "Nom de bar2" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--ID global du point de connexion intérieur de l'élément "IMLTransition" --> </xsd:complexType> </xsd:element> </xsd:complexType> </pre>

8.3 Archivage des réseaux de représentation nodale dans le schéma XML de logique AML

8.3.1 Règles communes

Pour archiver un réseau de représentation nodale dans un document XML de logique AML, les dispositions suivantes s'appliquent:

- Un réseau de représentation nodale doit être mappé à un élément logique IML de type "ActivityOnNodeNetwork" selon les dispositions suivantes.
- L'ensemble du réseau de représentation nodale doit être mappé à une IML selon les dispositions suivantes.
- L'IML doit commencer par l'élément "TimeInformation", qui définit le format horaire du réseau de représentation nodale (voir 8.3.2).
- Chaque nœud de réseau de représentation nodale doit être représenté par un IMLStep (voir 8.3.3).

- Chaque flèche de réseau de représentation nodale doit être représentée par un IMLTransition (voir 8.3.4).
- Pour les relations prédécesseur/successeur dans un réseau de représentation nodale, les dispositions suivantes s'appliquent:
 - Si un nœud de réseau de représentation nodale a au moins deux nœuds de successeur, il doit y être relié par des éléments IMLTransition et IMLSimultaneousDivergence (voir 8.3.5).
 - Si un nœud de réseau de représentation nodale a au moins deux prédécesseurs, il doit y être relié par des éléments IMLSimultaneousConvergence et IMLTransition (voir 8.3.6).
 - En l'absence de relation explicite entre des nœuds de réseau de représentation nodale (c'est-à-dire qu'ils sont concurrents), l'ordre de l'IMLStep associé doit être parallèle et ils ne doivent présenter aucune relation ordonnée reposant sur les séquences de transition pas à pas (voir 8.3.5 et 8.3.6).
 - Si des relations prédécesseur/successeur entre des nœuds de réseau de représentation nodale sont définies, la structure obtenue doit être une séquence d'éléments IMLStep et IMLTransition (voir 8.3.5 et 8.3.6).

8.3.2 Archivage du début d'un réseau de représentation nodale

Le Tableau 7 spécifie le mapping des propriétés et des relations du début d'un réseau de représentation nodale à des éléments IML.

Tableau 7 – Mapping du début d'un réseau de représentation nodale à des éléments IML

Type	Représentation
Élément du réseau de représentation nodale	Aucune représentation graphique disponible.
Élément IML	<pre><xsd:element name="TimeInformation" minOccurs="0"> <xsd:complexType> <xsd:attribute name="timeFormat" type="aml:TimeFormatEnum" use="required" /> <!-- Il doit s'agir de la valeur du type simple "TimeFormatEnum" --> </xsd:complexType> </xsd:element></pre>

8.3.3 Archivage de nœuds

Le Tableau 8 spécifie le mapping des nœuds de réseau de représentation nodale à des éléments IML.

Tableau 8 – Mapping de nœuds de réseau de représentation nodale à des éléments IML

Type	Représentation																
Élément du réseau de représentation nodale	<table><tr><td>Earliest start</td><td>Duration</td><td>Earliest end</td></tr><tr><td colspan="3">Node name</td></tr><tr><td>Latest start</td><td>Buffer</td><td>Latest end</td></tr></table>	Earliest start	Duration	Earliest end	Node name			Latest start	Buffer	Latest end							
	Earliest start	Duration	Earliest end														
	Node name																
	Latest start	Buffer	Latest end														
	<table><tr><th>Anglais</th><th>Français</th></tr><tr><td>Earliest start</td><td>Début le plus précoce</td></tr><tr><td>Duration</td><td>Durée</td></tr><tr><td>Earliest end</td><td>Fin la plus précoce</td></tr><tr><td>Node name</td><td>Nom du nœud</td></tr><tr><td>Latest start</td><td>Début le plus tardif</td></tr><tr><td>Buffer</td><td>Mémoire tampon</td></tr><tr><td>Latest end</td><td>Fin la plus tardive</td></tr></table>	Anglais	Français	Earliest start	Début le plus précoce	Duration	Durée	Earliest end	Fin la plus précoce	Node name	Nom du nœud	Latest start	Début le plus tardif	Buffer	Mémoire tampon	Latest end	Fin la plus tardive
	Anglais	Français															
	Earliest start	Début le plus précoce															
	Duration	Durée															
	Earliest end	Fin la plus précoce															
	Node name	Nom du nœud															
Latest start	Début le plus tardif																
Buffer	Mémoire tampon																
Latest end	Fin la plus tardive																
Élément IML	<xsd:complexType name="IMLStep">																

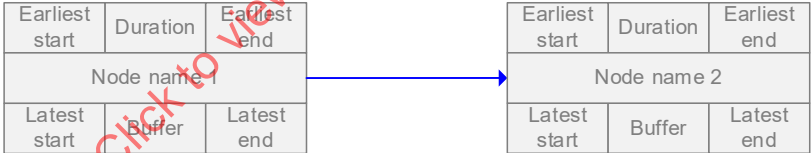
Type	Représentation
	<pre> <xsd:complexContent> <xsd:extension base="ppx:Step"> <xsd:attribute name="uuid" type="aml:UuidString"/> <!-- Il doit s'agir d'une représentation textuelle de l'UUID, qui peut être GUID. --> <xsd:attribute name="name" type="xsd:string" use="required"/> <!-- Désignation "Nom de la barre" dans la représentation textuelle du type xsd:string --> <xsd:attribute name="startTime" type="aml:TimeUnion"/> <!-- Valeur de l'heure de début la plus précoce au format de type "aml:TimeUnion " --> <!-- Avec l'heure de début la plus précoce et l'heure de fin la plus précoce, la durée peut être calculée.--> <xsd:attribute name="endTime" type="aml:TimeUnion"/> <!-- valeur l'heure de fin la plus précoce au format de type "aml:TimeUnion " --> <!-- Avec l'heure de début la plus précoce et l'heure de fin la plus précoce, la durée peut être calculée.--> <xsd:attribute name="buffer" type="aml:duration" use="optional"/> <!-- valeur de la mémoire tampon au format de type "aml:duration " --> <!-- Avec l'heure de début la plus précoce, l'heure de fin la plus précoce et la mémoire tampon, l'heure de début la plus tardive et l'heure de fin la plus tardive peuvent être calculées.--> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre>

8.3.4 Archivage des flèches

Le Tableau 9 spécifie le mapping de flèches de réseau de représentation nodale à des éléments IML.

NOTE Les deux nœuds ne sont pas mappés aux éléments IML, seule l'est la flèche qui les relie.

Tableau 9 – Mapping de flèches de réseau de représentation nodale à des éléments IML

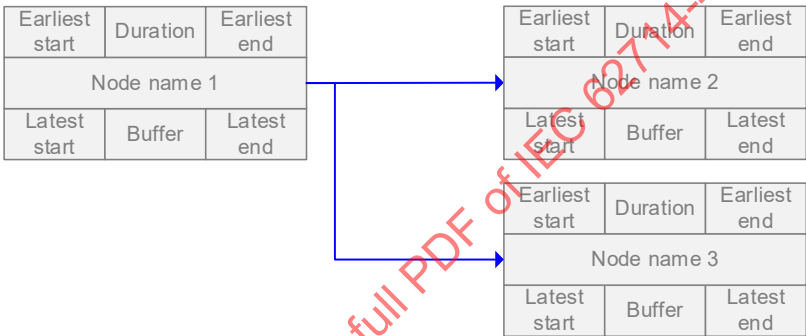
Type	Représentation																
Élément du réseau de représentation nodale	 <table border="1"> <thead> <tr> <th>Anglais</th><th>Français</th></tr> </thead> <tbody> <tr> <td>Earliest start</td><td>Début le plus précoce</td></tr> <tr> <td>Duration</td><td>Durée</td></tr> <tr> <td>Earliest end</td><td>Fin la plus précoce</td></tr> <tr> <td>Node name</td><td>Nom du nœud</td></tr> <tr> <td>Latest start</td><td>Début le plus tardif</td></tr> <tr> <td>Buffer</td><td>Mémoire tampon</td></tr> <tr> <td>Latest end</td><td>Fin la plus tardive</td></tr> </tbody> </table>	Anglais	Français	Earliest start	Début le plus précoce	Duration	Durée	Earliest end	Fin la plus précoce	Node name	Nom du nœud	Latest start	Début le plus tardif	Buffer	Mémoire tampon	Latest end	Fin la plus tardive
Anglais	Français																
Earliest start	Début le plus précoce																
Duration	Durée																
Earliest end	Fin la plus précoce																
Node name	Nom du nœud																
Latest start	Début le plus tardif																
Buffer	Mémoire tampon																
Latest end	Fin la plus tardive																
Élément IML	<pre> <xsd:complexType name="IMLTransition"> <xsd:complexContent> <xsd:extension base="ppx:Transition"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- Il doit s'agir d'une représentation textuelle de l'UUID, qui peut être GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- ID global du point de connexion extérieur de l'élément "Nom de bar1" --> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- ID global du point de connexion extérieur de l'élément "Nom de bar2" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> </pre>																

Type	Représentation
	<pre> </xsd:element> </xsd:complexType> </xsd:element> </xsd:complexType> </pre>

8.3.5 Archivage de nœuds de successeur

Le Tableau 10 spécifie le mapping des activités de successeur de réseau de représentation nodale à des éléments IML.

Tableau 10 – Mapping de nœuds de successeur de réseau de représentation nodale à des éléments IML

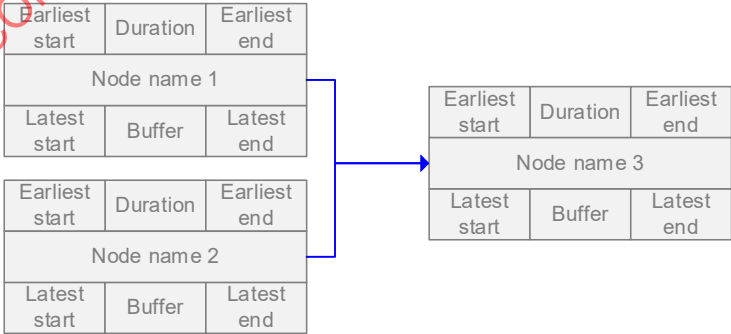
Type	Représentation																
Élément du réseau de représentation nodale	 <table border="1"> <thead> <tr> <th>Anglais</th><th>Français</th></tr> </thead> <tbody> <tr> <td>Earliest start</td><td>Début le plus précoce</td></tr> <tr> <td>Duration</td><td>Durée</td></tr> <tr> <td>Earliest end</td><td>Fin la plus précoce</td></tr> <tr> <td>Node name</td><td>Nom du nœud</td></tr> <tr> <td>Latest start</td><td>Début le plus tardif</td></tr> <tr> <td>Buffer</td><td>Mémoire tampon</td></tr> <tr> <td>Latest end</td><td>Fin la plus tardive</td></tr> </tbody> </table>	Anglais	Français	Earliest start	Début le plus précoce	Duration	Durée	Earliest end	Fin la plus précoce	Node name	Nom du nœud	Latest start	Début le plus tardif	Buffer	Mémoire tampon	Latest end	Fin la plus tardive
Anglais	Français																
Earliest start	Début le plus précoce																
Duration	Durée																
Earliest end	Fin la plus précoce																
Node name	Nom du nœud																
Latest start	Début le plus tardif																
Buffer	Mémoire tampon																
Latest end	Fin la plus tardive																
Élément IML	<pre> <xsd:complexType name="IMLTransition"> <xsd:complexContent> <xsd:extension base="ppx:Transition"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- Il doit s'agir d'une représentation textuelle de l'UUID, qui peut être GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- ID global du point de connexion extérieur de l'élément "Nom de bar1" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- ID global du point de connexion extérieur de l'élément "IMLTransition" --> </xsd:complexType> </xsd:element> </xsd:complexType> <xsd:complexType name="IMLSimultaneousDivergence"> <xsd:complexContent> <xsd:extension base="ppx:IMLSimultaneousDivergence"> </pre>																

Type	Représentation
	<pre> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- Il doit s'agir d'une représentation textuelle de l'UUID, qui peut être GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- ID global du point de connexion extérieur de l'élément "IMLTransition" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- ID global du point de connexion intérieur de l'élément "Nom de bar2" --> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- ID global du point de connexion intérieur de l'élément "Nom de bar3" --> </xsd:complexType> </xsd:element> </xsd:complexType> </pre>

8.3.6 Archivage de nœuds de prédécesseur

Le Tableau 11 spécifie le mapping des nœuds de prédécesseur de réseau de représentation nodale à des éléments IML.

Tableau 11 – Mapping de nœuds de prédécesseur de réseau de représentation nodale à des éléments IML

Type	Représentation																
Élément du réseau de représentation nodale	 <table border="1"> <thead> <tr> <th>Anglais</th><th>Français</th></tr> </thead> <tbody> <tr> <td>Earliest start</td><td>Début le plus précoce</td></tr> <tr> <td>Duration</td><td>Durée</td></tr> <tr> <td>Earliest end</td><td>Fin la plus précoce</td></tr> <tr> <td>Node name</td><td>Nom du nœud</td></tr> <tr> <td>Latest start</td><td>Début le plus tardif</td></tr> <tr> <td>Buffer</td><td>Mémoire tampon</td></tr> <tr> <td>Latest end</td><td>Fin la plus tardive</td></tr> </tbody> </table>	Anglais	Français	Earliest start	Début le plus précoce	Duration	Durée	Earliest end	Fin la plus précoce	Node name	Nom du nœud	Latest start	Début le plus tardif	Buffer	Mémoire tampon	Latest end	Fin la plus tardive
Anglais	Français																
Earliest start	Début le plus précoce																
Duration	Durée																
Earliest end	Fin la plus précoce																
Node name	Nom du nœud																
Latest start	Début le plus tardif																
Buffer	Mémoire tampon																
Latest end	Fin la plus tardive																
Élément IML	<pre> <xsd:complexType name="IMLTransition"> <xsd:complexContent> <xsd:extension base="ppx:Transition"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre>																

Type	Représentation
	<pre> <!-- Il doit s'agir d'une représentation textuelle de l'UUID, qui peut être GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--ID global du point de connexion intérieur de l'élément "IMLTransition" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--ID global du point de connexion intérieur de "Nom de bar3" --> </xsd:complexType> </xsd:element> </xsd:complexType> <xsd:complexType name="IMLSimultaneousConvergence"> <xsd:complexContent> <xsd:extension base="ppx:SimultaneousConvergence"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- Il doit s'agir d'une représentation textuelle de l'UUID, qui peut être GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--ID global du point de connexion extérieur de "Nom de bar1" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--ID global du point de connexion extérieur de "Nom de bar2" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--ID global du point de connexion intérieur de l'élément "IMLTransition" --> </xsd:complexType> </xsd:element> </xsd:complexType> </pre>

8.4 Archivage des chronogrammes dans un document XML de logique AML

8.4.1 Règles communes

Pour archiver un chronogramme dans un document XML de logique AML, les dispositions suivantes s'appliquent:

- Un chronogramme doit être mappé à un élément logique IML de type "TimingDiagram" selon les dispositions suivantes.
- L'IML doit commencer par la définition de la ligne chronologique (voir 8.4.2).
- Les ressources et leurs états dans un chronogramme doivent être représentés par des éléments IML (voir 8.4.3).
- La ligne de vie de chaque ressource doit être définie par ses segments (des éléments IMLStep associés à des éléments IMLTransition, par exemple) (voir 8.4.4).
- Les signaux dans un chronogramme doivent être définis par des éléments IMLTransition (voir 8.4.5).

NOTE Les signaux peuvent être retirés à tout moment par la ligne chronologique ou par une ressource, après être restés dans un état de ressource pendant une durée prédéfinie.

8.4.2 Archivage de la ligne chronologique d'un chronogramme

Le Tableau 12 spécifie le mapping de la ligne chronologique d'un chronogramme à des éléments IML.

Tableau 12 – Mapping de la ligne chronologique d'un chronogramme à des éléments IML

Type	Représentation				
Élément du chronogramme	 <table border="1"> <thead> <tr> <th>Anglais</th><th>Français</th></tr> </thead> <tbody> <tr> <td>Time</td><td>Temps</td></tr> </tbody> </table>	Anglais	Français	Time	Temps
Anglais	Français				
Time	Temps				
Élément IML	<pre> <xsd:element name="TimeInformation" minOccurs="0"> <xsd:complexType> <xsd:attribute name="timeFormat" type="aml:TimeFormatEnum" use="required" /> <!-- Il doit s'agir de la valeur du type simple "TimeFormatEnum" --> <xsd:attribute name="startOfScale" type="aml:TimeUnion" use="optional" default="PT0S"/> <!-- valeur de t0 au format défini par l'attribut "timeFormat" --> <xsd:attribute name="endOfScale" type="aml:TimeUnion" use="optional"/> <!-- valeur de tn au format défini par l'attribut "timeFormat" --> <xsd:attribute name="scaleInterval" type="aml:TimeUnion" use="optional"/> <!-- valeur de l'échelle de temps au format défini par le type xsd:duration --> </xsd:complexType> </xsd:element> </pre>				

8.4.3 Archivage des ressources et des états de ressource

Le Tableau 13 spécifie le mapping des ressources et états de ressource du chronogramme à des éléments IML.

Tableau 13 – Mapping des ressources et états de ressource du chronogramme à des éléments IML

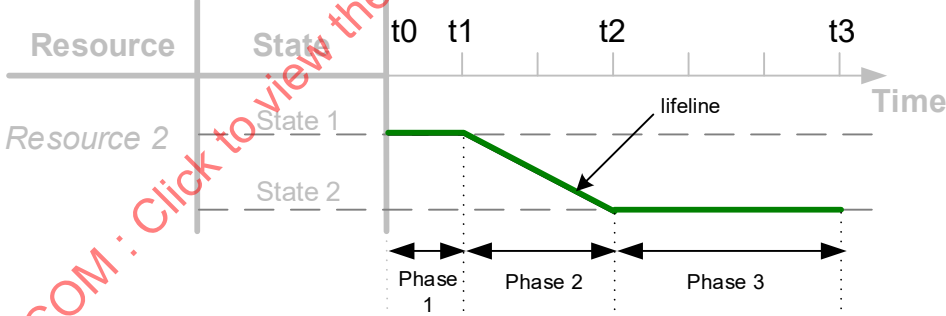
Type	Représentation											
Élément du chronogramme	<table border="1"> <thead> <tr> <th>Resource</th><th>State</th></tr> </thead> <tbody> <tr> <td rowspan="2">Resource 1</td><td>State 1</td></tr> <tr> <td>State 2</td></tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Anglais</th><th>Français</th></tr> </thead> <tbody> <tr> <td>Resource</td><td>Ressource</td></tr> <tr> <td>State</td><td>État</td></tr> </tbody> </table>	Resource	State	Resource 1	State 1	State 2	Anglais	Français	Resource	Ressource	State	État
Resource	State											
Resource 1	State 1											
	State 2											
Anglais	Français											
Resource	Ressource											
State	État											

Type	Représentation
Élément IML	<pre> <xsd:element name="Resource" minOccurs="0" maxOccurs="unbounded"> <xsd:complexType> <xsd:sequence> <xsd:element name="Status" maxOccurs="unbounded"> <xsd:complexType> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- Il doit s'agir d'une représentation textuelle de l'UUID, qui peut être GUID. --> <xsd:attribute name="name" type="xsd:string" use="required"/> <!-- Nom de l'état dans la représentation textuelle de type xsd:string --> <xsd:attribute name="order" type="xsd:integer" use="optional"/> <!-- Ordre de l'état dans la représentation xsd:integer--> </xsd:complexType> </xsd:element> </xsd:sequence> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- Il doit s'agir d'une représentation textuelle de l'UUID, qui peut être GUID. --> <xsd:attribute name="name" type="xsd:string" use="required"/> <!-- Nom de la ressource dans la représentation textuelle de type xsd:string --> <xsd:attribute name="order" type="xsd:integer" use="optional"/> <!-- Ordre de la ressource dans la représentation xsd:integer--> </xsd:complexType> </xsd:element> </pre>

8.4.4 Archivage des lignes de vie

Le Tableau 14 spécifie le mapping des lignes de vie du chronogramme à des éléments IML.

Tableau 14 – Mapping des lignes de vie du chronogramme à des éléments IML

Type	Représentation										
Élément du chronogramme	 <table border="1"> <thead> <tr> <th>Anglais</th><th>Français</th></tr> </thead> <tbody> <tr> <td>Resource</td><td>Ressource</td></tr> <tr> <td>State</td><td>État</td></tr> <tr> <td>lifeline</td><td>ligne de vie</td></tr> <tr> <td>Time</td><td>Temps</td></tr> </tbody> </table>	Anglais	Français	Resource	Ressource	State	État	lifeline	ligne de vie	Time	Temps
Anglais	Français										
Resource	Ressource										
State	État										
lifeline	ligne de vie										
Time	Temps										
Élément IML	<pre> <xsd:complexType name="IMLStep"> <xsd:complexContent> <xsd:extension base="ppx:Step"> <xsd:attribute name="uuid" type="aml:UuidString"/> <!-- Il doit s'agir d'une représentation textuelle de l'UUID, qui peut être GUID. --> <xsd:attribute name="name" type="xsd:string" use="required"/> <!-- Segment "Phase 1" dans la représentation textuelle de type xsd:string --> <xsd:attribute name="startTime" type="aml:TimeUnion"/> <!-- valeur de t0 au format de type "aml:TimeUnion" --> <xsd:attribute name="endTime" type="aml:TimeUnion"/> <!-- valeur de t1 au format de type "aml:TimeUnion" --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- ID global du point de connexion extérieur de "Phase 1" --> </xsd:complexType> </xsd:element> </xsd:complexType> </pre>										

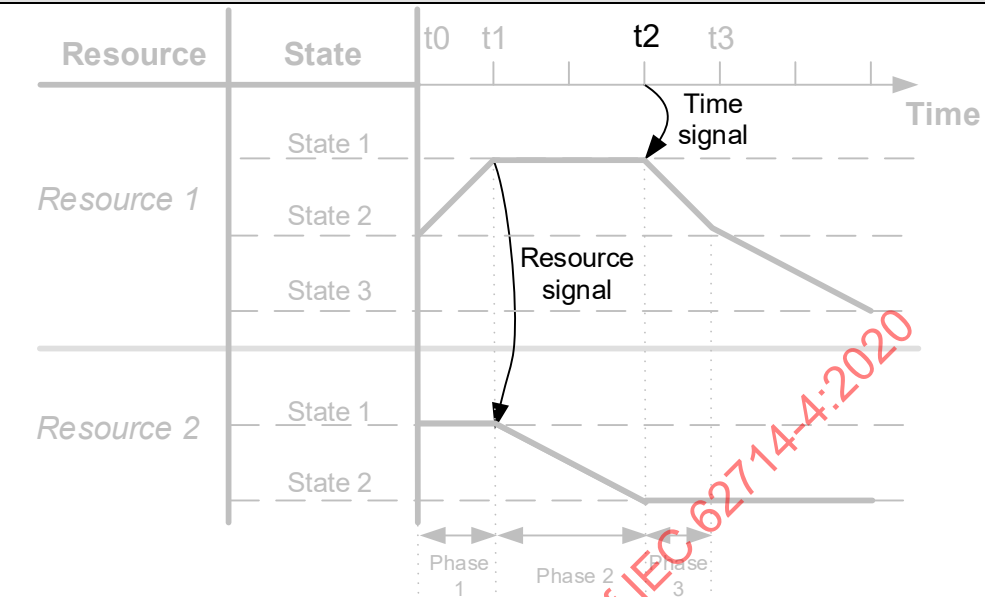
Type	Représentation
	<pre> <xsd:complexType name="IMLStep"> <xsd:complexContent> <xsd:extension base="ppx:Step"> <xsd:attribute name="uuid" type="aml:UuidString"/> <!-- Il doit s'agir d'une représentation textuelle de l'UUID, qui peut être GUID. --> <xsd:attribute name="name" type="xsd:string" use="required"/> <!-- Segment "Phase 2" dans la représentation textuelle de type xsd:string --> <xsd:attribute name="startTime" type="aml:TimeUnion"/> <!-- valeur de t1 au format de type "aml:TimeUnion " --> <xsd:attribute name="endTime" type="aml:TimeUnion"/> <!-- valeur de t2 au format de type "aml:TimeUnion " --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--ID global du point de connexion intérieur de "Phase 2"--> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--ID global du point de connexion extérieur de "Phase 2" --> </xsd:complexType> </xsd:element> </xsd:complexType> <xsd:complexType name="IMLStep"> <xsd:complexContent> <xsd:extension base="ppx:Step"> <xsd:attribute name="uuid" type="aml:UuidString"/> <!-- Il doit s'agir d'une représentation textuelle de l'UUID, qui peut être GUID. --> <xsd:attribute name="name" type="xsd:string" use="required"/> <!-- Segment "Phase 3" dans la représentation textuelle de type xsd:string --> <xsd:attribute name="startTime" type="aml:TimeUnion"/> <!-- valeur de t2 au format de type "aml:TimeUnion " --> <xsd:attribute name="endTime" type="aml:TimeUnion"/> <!-- valeur de t3 au format de type "aml:TimeUnion " --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--ID global du point de connexion intérieur de "Phase 3" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> </xsd:complexType> <xsd:complexType name="IMLTransition"> <xsd:complexContent> <xsd:extension base="ppx:Transition"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- Il doit s'agir d'une représentation textuelle de l'UUID, qui peut être GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> </pre>

Type	Représentation
	<pre> <!--ID global du point de connexion extérieur de "Phase 1" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--ID global du point de connexion intérieur de "Phase 2" --> </xsd:complexType> </xsd:element> </xsd:complexType> <xsd:complexType name="IMLTransition"> <xsd:complexContent> <xsd:extension base="ppx:Transition"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- Il doit s'agir d'une représentation textuelle de l'UUID, qui peut être GUID. --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--ID global du point de connexion extérieur de "Phase 2" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--ID global du point de connexion intérieur de "Phase 3" --> </xsd:complexType> </xsd:element> </xsd:complexType> </pre>

8.4.5 Archivage du signal de temps et du signal de ressource

Le Tableau 15 spécifie le mapping du signal de temps et du signal de ressource à des éléments IML.

Tableau 15 – Mapping du signal de temps et du signal de ressource à des éléments IML

Type	Représentation												
Élément du chronogramme	 <table border="1" data-bbox="391 929 1300 1086"> <thead> <tr> <th>Anglais</th><th>Français</th></tr> </thead> <tbody> <tr> <td>Resource</td><td>Ressource</td></tr> <tr> <td>State</td><td>État</td></tr> <tr> <td>Time</td><td>Temps</td></tr> <tr> <td>Time signal</td><td>Signal de temps</td></tr> <tr> <td>Resource signal</td><td>Signal de ressource</td></tr> </tbody> </table>	Anglais	Français	Resource	Ressource	State	État	Time	Temps	Time signal	Signal de temps	Resource signal	Signal de ressource
Anglais	Français												
Resource	Ressource												
State	État												
Time	Temps												
Time signal	Signal de temps												
Resource signal	Signal de ressource												
Élément IML	<pre> <xsd:complexType name="IMLTransition"> <xsd:complexContent> <xsd:extension base="ppx:Transition"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- Il doit s'agir d'une représentation textuelle de l'UUID, qui peut être GUID. --> <xsd:attribute name="refTime" type="aml:TimeUnion" use="optional"/> <!-- valeur de t2 au format de type "aml:TimeUnion" --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:element name="Connection" type="ppx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- ID global du point de connexion extérieur de "Phase 2 Ressource 1" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="ppx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!-- ID global du point de connexion intérieur de "Phase 3 Ressource 1" --> </xsd:complexType> </xsd:element> </xsd:complexType> <xsd:complexType name="IMLTransition"> <xsd:complexContent> <xsd:extension base="ppx:Transition"> <xsd:attribute name="uuid" type="aml:UuidString" use="required"/> <!-- Il doit s'agir d'une représentation textuelle de l'UUID, qui peut être GUID. --> <xsd:attribute name="refEndSegmentId" type="aml:UuidString" use="optional"/> <!-- Il doit s'agir d'une représentation textuelle de l'UUID, qui peut être GUID, de l'IMLStep pour le segment "Phase 1 Ressource 1" --> </xsd:extension> </xsd:complexContent> <xsd:element name="ConnectionPointIn" type="ppx:connectionPointIn" minOccurs="0" maxOccurs="1"> </pre>												

Type	Représentation
	<pre> <xsd:complexType> <xsd:element name="Connection" type="pdx:connection" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="refConnectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--ID global du point de connexion extérieur de "Phase 1 Ressource 2" --> </xsd:complexType> </xsd:element> </xsd:complexType> </xsd:element> <xsd:element name="ConnectionPointOut" type="pdx:connectionPointOut" minOccurs="0" maxOccurs="1"> <xsd:complexType> <xsd:attribute name="connectionPointOutId" type="xsd:unsignedLong" use="required"/> <!--ID global du point de connexion intérieur de "Phase 2 Ressource 2" --> </xsd:complexType> </xsd:element> </xsd:complexType> </pre>

8.5 Archivage des diagrammes fonctionnels en séquence dans un document XML de logique AML

8.5.1 Règles communes

Pour archiver un diagramme fonctionnel en séquence dans un document XML de logique AML, les dispositions suivantes s'appliquent:

- L'ensemble du diagramme fonctionnel en séquence doit être mappé à un élément "FunctionBlock" de type "SFC" dans un élément "AMLLogic".
- Les éléments d'un diagramme fonctionnel en séquence doivent être utilisés conformément à l'IEC 61131-10.

8.5.2 Archivage des variables

Le Tableau 16 spécifie l'archivage des variables d'un diagramme fonctionnel en séquence dans un document XML de logique AML.

Tableau 16 – Archivage des variables d'un diagramme fonctionnel en séquence dans un document XML de logique AML

Type	Représentation
Diagramme fonctionnel en séquence	Aucune représentation graphique disponible.
Élément XML de logique AML	<pre> <xsd:element name="Parameters" type="ppx:ParameterSet" minOccurs="0" maxOccurs="1"/> <xsd:complexType name="ParameterSet"> <xsd:sequence> <xsd:choice minOccurs="0" maxOccurs="unbounded"> <xsd:element name="InputVars"> <xsd:complexType> <xsd:sequence> <xsd:element name="Variable" minOccurs="0" maxOccurs="unbounded"> <xsd:complexType> <xsd:complexContent> <xsd:extension base="ppx:VariableDecl"> <xsd:attribute name="orderWithinParamSet" type="xsd:unsignedInt" use="required"/> <!-- Doit indiquer l'ordre des paramètres. --> <xsd:attribute name="edgeDetection" type="ppx:EdgeModifierType" use="optional" default="none"/> <!-- Ce paramètre peut être caractérisé par rapport à la détection des contours. -- </xsd:extension> </xsd:complexContent> </xsd:complexType> </xsd:element> </xsd:sequence> </xsd:complexType> </xsd:element> <xsd:element name="OutputVars"> <xsd:complexType> <xsd:sequence> <xsd:element name="Variable" minOccurs="0" maxOccurs="unbounded"> <xsd:complexType> <xsd:complexContent> <xsd:extension base="ppx:VariableDecl"> <xsd:attribute name="orderWithinParamSet" type="xsd:unsignedInt" use="required"/> <!-- Doit indiquer l'ordre des paramètres. --> </xsd:extension> </xsd:complexContent> </xsd:complexType> </xsd:element> </xsd:sequence> </xsd:complexType> </xsd:element> </xsd:choice> </xsd:sequence> </xsd:complexType> </pre>

8.6 Archivage des diagrammes de bloc fonctionnel dans un document XML de logique AML

8.6.1 Règles communes

Pour archiver un diagramme de bloc fonctionnel dans un document XML de logique AML, les dispositions suivantes s'appliquent:

- L'ensemble du diagramme de bloc fonctionnel doit être mappé à un élément "FunctionBlock" de type "FBD" dans un élément "AMLLogic".
- Les éléments d'un diagramme de bloc fonctionnel doivent être utilisés conformément à l'IEC 61131-10.

8.6.2 Archivage des variables

Le Tableau 17 spécifie l'archivage des variables d'un diagramme de bloc fonctionnel dans un document XML de logique AML.

Tableau 17 – Archivage des variables d'un diagramme de bloc fonctionnel dans un document XML de logique AML

Type	Représentation
Diagramme de bloc fonctionnel	Aucune représentation graphique disponible.
Élément XML de logique AML	<pre> <xsd:element name="Parameters" type="ppx:ParameterSet" minOccurs="0" maxOccurs="1"/> <xsd:complexType name="ParameterSet"> <xsd:sequence> <xsd:choice minOccurs="0" maxOccurs="unbounded"> <xsd:element name="InputVars"> <xsd:complexType> <xsd:sequence> <xsd:element name="Variable" minOccurs="0" maxOccurs="unbounded"> <xsd:complexType> <xsd:complexContent> <xsd:extension base="ppx:VariableDecl"> <xsd:attribute name="orderWithinParamSet" type="xsd:unsignedInt" use="required"/> <!-- Doit indiquer l'ordre des paramètres. --> <xsd:attribute name="edgeDetection" type="ppx:EdgeModifierType" use="optional" default="none"/> <!-- Ce paramètre peut être caractérisé par rapport à la détection des contours. --> </xsd:extension> </xsd:complexContent> </xsd:complexType> </xsd:element> </xsd:sequence> </xsd:complexType> </xsd:element> <xsd:element name="OutputVars"> <xsd:complexType> <xsd:sequence> <xsd:element name="Variable" minOccurs="0" maxOccurs="unbounded"> <xsd:complexType> <xsd:complexContent> <xsd:extension base="ppx:VariableDecl"> <xsd:attribute name="orderWithinParamSet" type="xsd:unsignedInt" use="required"/> <!-- Doit indiquer l'ordre des paramètres. --> </xsd:extension> </xsd:complexContent> </xsd:complexType> </xsd:element> </xsd:sequence> </xsd:complexType> </xsd:element> </xsd:choice> </xsd:sequence> </xsd:complexType> </pre>

8.7 Archivage des expressions mathématiques dans un document XML de logique AML

8.7.1 Règles communes

Pour archiver une expression mathématique dans un document XML de logique AML, les dispositions suivantes s'appliquent:

- L'ensemble de l'expression mathématique doit être mappé à un élément "FunctionBlock" de type "MathematicalExpression" dans un élément "AMLLogic".
- Les variables de l'élément "FunctionBlock" doivent être mappées en conséquence aux éléments "InputVars" ou "OutputVars" dans l'élément "Parameters".
- L'expression mathématique doit être archivée dans l'élément "MathML" conformément à la partie de contenu MathML 2.0.

- La manière de corréler les variables de l'élément "MathML" avec celles de l'élément "FunctionBlock" doit être modélisée dans l'élément "VariableMapping".
 - L'élément "VariableMapping" doit avoir les attributs suivants:
 - L'attribut "refMathMLVariableName" doit archiver le nom de la variable dans l'élément "MathML".
 - L'attribut "refFBVariableUUID" doit archiver l'UUID de la variable dans l'élément "FunctionBlock".
 - L'attribut "direction" doit indiquer s'il s'agit d'une variable d'entrée ou de sortie pour l'expression mathématique archivée dans l'élément "MathML". La valeur "In" doit indiquer une variable d'entrée. La valeur "Out" doit indiquer une variable de sortie.

8.7.2 Archivage des variables

Le Tableau 18 spécifie l'archivage des variables d'une expression mathématique dans un document XML de logique AML.

Tableau 18 – Archivage des variables d'une expression mathématique dans un document XML de logique AML

Type	Représentation
Élément d'expression mathématique	Aucune représentation graphique disponible.
Élément XML de logique AML	<pre> <xsd:element name="Parameters" type="ppx:ParameterSet" minOccurs="0" maxOccurs="1"/> <xsd:complexType name="ParameterSet"> <xsd:sequence> <xsd:choice minOccurs="0" maxOccurs="unbounded"> <xsd:element name="InputVars"> <xsd:complexType> <xsd:sequence> <xsd:element name="Variable" minOccurs="0" maxOccurs="unbounded"> <xsd:complexType> <xsd:complexContent> <xsd:extension base="ppx:VariableDecl"> <xsd:attribute name="orderWithinParamSet" type="xsd:unsignedInt" use="required"/> <!-- Doit indiquer l'ordre des paramètres. --> <xsd:attribute name="edgeDetection" type="ppx:EdgeModifierType" use="optional" default="none"/> <!-- Ce paramètre peut être caractérisé par rapport à la détection des contours. --> </xsd:extension> </xsd:complexContent> </xsd:complexType> </xsd:element> </xsd:sequence> </xsd:complexType> </xsd:element> <xsd:element name="OutputVars"> <xsd:complexType> <xsd:sequence> <xsd:element name="Variable" minOccurs="0" maxOccurs="unbounded"> <xsd:complexType> <xsd:complexContent> <xsd:extension base="ppx:VariableDecl"> <xsd:attribute name="orderWithinParamSet" type="xsd:unsignedInt" use="required"/> <!-- Doit indiquer l'ordre des paramètres. --> </xsd:extension> </xsd:complexContent> </xsd:complexType> </xsd:element> </xsd:sequence> </xsd:complexType> </xsd:element> </xsd:choice> </xsd:sequence> </xsd:complexType> </pre>

8.7.3 Archivage des mappings de variable

Le Tableau 19 spécifie l'archivage des mappings de variable d'une expression mathématique dans un document XML de logique AML.

Tableau 19 – Archivage des mappings de variable d'une expression mathématique dans un document XML de logique AML

Type	Représentation
Élément d'expression mathématique	Aucune représentation graphique disponible.
Élément XML de logique AML	<pre> <xsd:element name="VariableMapping" minOccurs="0" maxOccurs="unbounded"> <xsd:complexType> <xsd:attribute name="refMathMLVariableName" type="xsd:string" use="required"/> <!--Il doit représenter le nom de la variable définie dans MathML.--> <xsd:attribute name="refFBVariableUUID" type="aml:UuidString" use="required"/> <!--Il doit représenter l'UUID of de la variable déclarée dans le même bloc fonctionnel.--> <xsd:attribute name="direction" use="required"> <xsd:simpleType> <xsd:restriction base="xsd:string"> <xsd:enumeration value="In"/> <xsd:enumeration value="Out"/> </xsd:restriction> </xsd:simpleType> </xsd:attribute> <!--Il doit représenter la direction du mapping avec la chaîne émulée qui ne permet que "In" ou "Out". --> </xsd:complexType> </xsd:element> </pre>

8.7.4 Archivage des expressions mathématiques

Le Tableau 20 spécifie l'archivage d'une expression mathématique dans un document XML de logique AML.

Tableau 20 – Archivage d'une expression mathématique dans un document XML de logique AML

Type	Représentation
Élément d'expression mathématique	Expression mathématique
Élément XML de logique AML	<pre> <xsd:element name="MathML"> <xsd:complexType> <xsd:sequence> <xsd:any namespace="##any" processContents="lax"/> </xsd:sequence> </xsd:complexType> </xsd:element> </pre>

9 Méta-informations relatives aux outils d'écriture XML de logique AML

Pour simplifier l'échange de données entre un outil source et un outil de destination, il est utile d'archiver des informations relatives à l'outil d'écriture directement dans le document XML de logique AML. Ainsi, les dispositions suivantes s'appliquent:

NOTE Dans le processus d'ingénierie, il est utile de transférer les informations du projet auquel appartiennent les données et la durée de l'export des données. Cela vient à l'appui de la mise en œuvre du processus des interfaces.

- Un document XML de logique AML doit donner des informations relatives à chaque outil d'écriture avec lequel il a été écrit.

- Dans la chaîne d'outils d'échange de données, les derniers outils d'écriture participants doivent archiver ces informations dans le document XML de logique AML.
- Chaque document XML de logique AML doit contenir au moins un élément XML "WriterHeader".
- Dans la chaîne d'outils d'échange de données, tous les outils participants doivent archiver de la même manière ces informations dans le document XML de logique AML. Par conséquent, le document peut contenir des informations relatives à plusieurs outils d'une chaîne d'outils d'échange de données. Un outil peut supprimer les informations d'écriture d'autres outils. Cela peut gêner l'échange de données itératives avec les autres outils: le retrait des informations d'écriture des autres outils n'est donc pas recommandé.
- Ces informations doivent être archivées dans les éléments XML de l'élément XML "WriterHeader".
- Les méta-informations doivent donner des informations concernant:
 - le nom de l'outil d'écriture;
 - l'ID unique de l'outil d'écriture (par exemple: UUID, URL de l'outil);
 - Le nom du fournisseur de l'outil d'écriture;
 - l'URL du fournisseur de l'outil d'écriture;
 - les informations complètes relatives à la version de l'outil d'écriture (y compris les révisions majeure, mineure et de version); et
 - la date et l'heure de la dernière écriture du document XML de logique AML.
- De plus, les informations facultatives suivantes peuvent être incluses:
 - les contextes culturels de l'outil d'écriture conformément au RFC 5646;
 - le nom du projet associé aux données exportées;
 - l'ID unique du projet associé aux données exportées; et
 - les informations supplémentaires.
- Les informations ci-dessus doivent être archivées à l'aide des éléments XML présentés dans le Tableau 21 pour chaque outil d'écriture.

Tableau 21 – Méta-informations relatives à chaque outil d'écriture XML de logique AML

Nom d'élément XML	Type	Cardinalité	Exemple
WriterName	xs:string	1	"ToolX"
WriterID	xs:string	1	"4F1D5716-1C2A-4CBC-8C40-91A91F7E09A4"
WriterVendor	xs:string	1	"ToolX Vendor"
WriterVendorURL	xs:string	1	"http://www.ToolX-Vendor.org"
WriterVersion	xs:string	1	"0.2.123 prealpha"
LastWritingDateTime	xs:dateTime	1	"2011-05-25T09:30:47"
WriterCulturalSetting	xs:string	0..1	"en-US"
WriterProjectTitle	xs:string	0..1	"ProjectY"
WriterProjectID	xs:string	0..1	"6BA300EA-DF77-4E78-80F9-2B050E2C6799"
AdditionalInformation	xs:string	0..1	"Ce fournisseur d'outil s'appelait auparavant ToolYVendor."

10 Extensions des classes AML pour la logique

10.1 Généralités

Le présent article définit les extensions des classes de rôles AML et des classes d'interfaces AML. Ces classes font partie des bibliothèques ou sont des spécialisations des classes normalisées AML définies dans l'IEC 62714-1. Tous les attributs et interfaces décrits font partie de ces bibliothèques et peuvent être retirés dans InstanceHierarchy s'ils ne sont pas nécessaires.

10.2 AutomationMLLogicRoleClassLib

10.2.1 Généralités

La Figure 23 représente l'AutomationMLLogicRoleClassLib normative sous la forme d'un arbre d'objets. AutomationMLLogicRoleClassLib spécifie toutes les classes de rôles liées à la logique. Les détails de chaque classe de rôles sont donnés de 10.2.2 à 10.2.4. Le texte XML d' AutomationMLLogicRoleClassLib est donné à l'Article D.2.

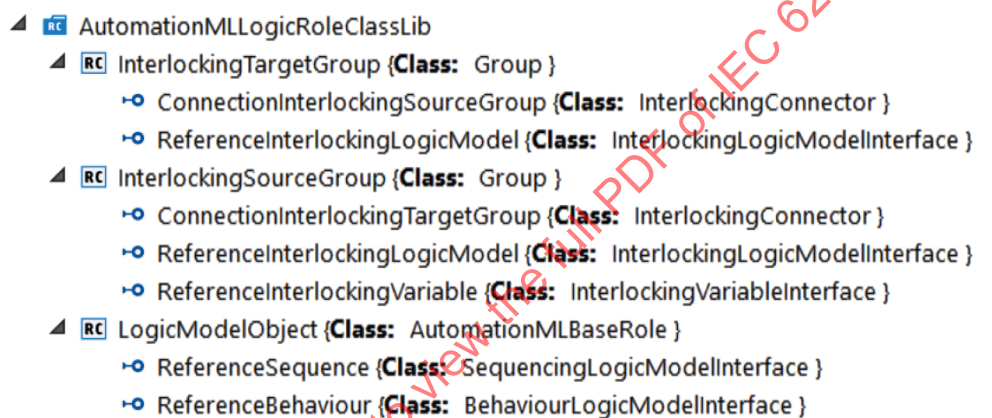


Figure 23 – AutomationMLLogicRoleClassLib

10.2.2 RoleClass InterlockingTargetGroup

La classe de rôles "InterlockingTargetGroup" doit être utilisée comme cela est spécifié dans le Tableau 22.

Tableau 22 – RoleClass InterlockingTargetGroup

Nom de classe	InterlockingTargetGroup	
Description	La classe de rôles "InterlockingTargetGroup" doit être utilisée pour les groupes cibles de verrouillage. Les détails sont spécifiés à l'Article 12. Pour des exemples, voir l'Annexe C.	
Classe parente	AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Group	
Chemin pour la référence de l'élément	AutomationMLLogicRoleClassLib/InterlockingTargetGroup	
Attributs	aucun	
Interfaces	Interface de type="InterlockingConnector"	Les interfaces doivent être du type "InterlockingConnector" spécifié en 10.5.2. Cette interface doit être exigée pour le concept de verrouillage spécifié à l'Article 12. Le nom est défini par l'utilisateur. L'utilisation de l'interface doit être obligatoire.
	Interface de type="InterlockingLogicModelInterface"	Les interfaces doivent être du type "InterlockingLogicModelInterface" spécifié en 10.3.5. Cette interface doit être utilisée pour le concept de verrouillage spécifié à l'Article 12 pour le référencement du modèle logique de verrouillage. Le nom est défini par l'utilisateur. L'utilisation de l'interface doit être facultative.

10.2.3 RoleClass InterlockingSourceGroup

La classe de rôles "InterlockingSourceGroup" doit être utilisée comme cela est spécifié dans le Tableau 23.

Tableau 23 – RoleClass InterlockingSourceGroup

Nom de classe	InterlockingSourceGroup	
Description	La classe de rôles "InterlockingSourceGroup" doit être utilisée pour les groupes sources de verrouillage. Des détails sont spécifiés à l'Article 12. Pour des exemples, voir l'Annexe C.	
Classe parente	AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Group	
Chemin pour la référence de l'élément	AutomationMLLogicRoleClassLib/InterlockingSourceGroup	
Attributs	aucun	
Interfaces	Interface de type="InterlockingConnector"	Les interfaces doivent être du type "InterlockingConnector" spécifié en 10.5.2. Cette interface doit être exigée pour le concept de verrouillage spécifié à l'Article 12. Le nom est défini par l'utilisateur. L'utilisation de l'interface doit être obligatoire.
	Interface de type="InterlockingLogicModelInterface"	Les interfaces doivent être du type "InterlockingLogicModelInterface" spécifié en 10.3.5. Cette interface doit être utilisée pour le concept de verrouillage spécifié à l'Article 12 pour le référencement du modèle logique de verrouillage. Le nom est défini par l'utilisateur. L'utilisation de l'interface doit être facultative.
	Interface de type="InterlockingVariableInterface"	Les interfaces doivent être du type "InterlockingVariableInterface" spécifié en 10.3.8. Cette interface doit être utilisée pour le concept de verrouillage spécifié à l'Article 12 pour le référencement de la variable de sortie dans un modèle logique contenant des informations de verrouillage. Le nom est défini par l'utilisateur. L'utilisation de l'interface doit être facultative.

10.2.4 RoleClass LogicModelObject

La classe de rôles "LogicModelObject" doit être utilisée comme cela est spécifié dans le Tableau 24.

Tableau 24 – RoleClass LogicModelObject

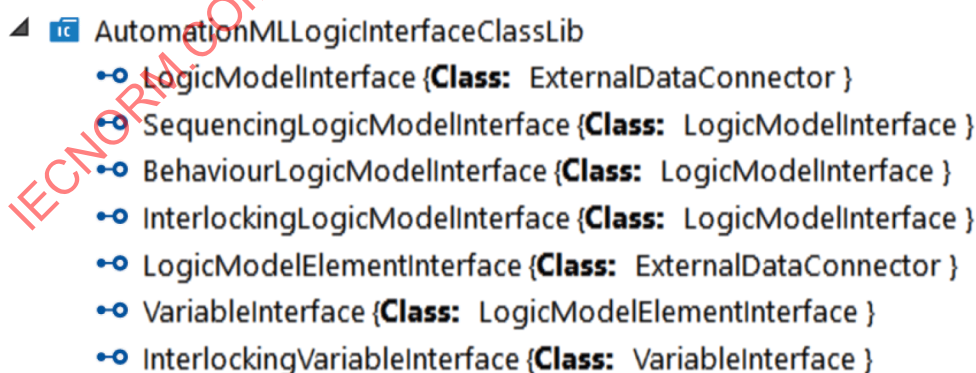
Nom de classe	LogicModelObject	
Description	La classe de rôles "LogicModelObject" doit indiquer un objet logique AML qui instancie un modèle logique.	
Classe parente	AutomationMLBaseRoleClassLib/AutomationMLBaseRole	
Chemin pour la référence de l'élément	AutomationMLLogicRoleClassLib/LogicModelObject	
Attributs	aucun	
Interfaces	Interface of type="SequencingLogicModelInterface"	Les interfaces doivent être du type "SequencingLogicModelInterface" spécifié en 10.3.3. Cette interface doit être utilisée pour spécifier la sémantique de modèle logique spécifiée à l'Article 11. Le nom est défini par l'utilisateur. L'utilisation de l'interface doit être facultative.
	Interface de type="BehaviourLogicModelInterface"	Les interfaces doivent être du type "BehaviourLogicModelInterface" spécifié en 10.3.4. Cette interface doit être utilisée pour spécifier la sémantique de modèle logique spécifiée à l'Article 11. Le nom est défini par l'utilisateur. L'utilisation de l'interface doit être facultative.

NOTE Ces objets peuvent être enrichis et référencés dans une plus large mesure dans CAEX.

10.3 AutomationMLLogicInterfaceClassLib

10.3.1 Généralités

La Figure 24 représente l'AutomationMLLogicInterfaceClassLib normative sous la forme d'un arbre d'objets. AutomationMLLogicInterfaceClassLib spécifie toutes les classes d'interfaces liées à la logique pour le référencement des documents XML de logique AML. Les détails de chaque classe d'interfaces sont donnés de 10.3.2 à 10.3.8. Le texte XML d'AutomationMLInterfaceClassLib est donné à l'Article D.3.



IEC

Figure 24 – AutomationMLLogicInterfaceClassLib

10.3.2 InterfaceClass LogicModelInterface

La classe d'interfaces "LogicModelInterface" doit être utilisée comme cela est spécifié dans le Tableau 25.

Tableau 25 – InterfaceClass LogicModelInterface

Nom de classe	LogicModelInterface
Description	Toutes les déductions de la classe d'interfaces "LogicModelInterface" doivent être utilisées pour référencer un modèle logique archivé dans un FunctionBlock à l'intérieur d'un document XML de logique AML externe. Les détails sont spécifiés à l'Article 11. Pour des exemples, voir l'Annexe B.
Classe parente	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector
Chemin pour la référence de l'élément	AutomationMLLogicInterfaceClassLib/LogicModelInterface
Attributs	aucun

10.3.3 InterfaceClass SequencingLogicModelInterface

La classe d'interfaces "SequencingLogicModelInterface" doit être utilisée comme cela est spécifié dans le Tableau 26.

Tableau 26 – InterfaceClass SequencingLogicModelInterface

Nom de classe	SequencingLogicModelInterface
Description	La classe d'interfaces "SequencingLogicModelInterface" doit être utilisée pour référencer un modèle logique contenant les informations de séquençement archivées dans un FunctionBlock à l'intérieur d'un document XML de logique AML externe. Les détails sont spécifiés à l'Article 11. Pour des exemples, voir l'Annexe B.
Classe parente	AutomationMLLogicInterfaceClassLib/LogicModelInterface
Chemin pour la référence de l'élément	AutomationMLLogicInterfaceClassLib/SequencingLogicModelInterface
Attributs	aucun

10.3.4 InterfaceClass BehaviourLogicModelInterface

La classe d'interfaces "BehaviourLogicModelInterface" doit être utilisée comme cela est spécifié dans le Tableau 27.

Tableau 27 – InterfaceClass BehaviourLogicModelInterface

Nom de classe	BehaviourLogicModelInterface
Description	La classe d'interfaces "BehaviourLogicModelInterface" doit être utilisée pour référencer un modèle logique contenant les informations de comportement archivées dans un FunctionBlock à l'intérieur d'un document XML de logique AML externe. Les détails sont spécifiés à l'Article 11. Pour des exemples, voir l'Annexe B.
Classe parente	AutomationMLLogicInterfaceClassLib/LogicModelInterface
Chemin pour la référence de l'élément	AutomationMLLogicInterfaceClassLib/BehaviourLogicModelInterface
Attributs	aucun

10.3.5 InterfaceClass InterlockingLogicModelInterface

La classe d'interfaces "InterlockingLogicModelInterface" doit être utilisée comme cela est spécifié dans le Tableau 28.

Tableau 28 – InterfaceClass InterlockingLogicModelInterface

Nom de classe	InterlockingLogicModelInterface
Description	La classe d'interfaces "InterlockingLogicModelInterface" doit être utilisée pour référencer un modèle logique contenant les informations de verrouillage archivées dans un FunctionBlock à l'intérieur d'un document XML de logique AML externe. Cette classe d'interfaces doit uniquement être utilisée par les groupes sources de verrouillage et les groupes cibles de verrouillage. Les détails sont spécifiés à l'Article 12. Pour des exemples, voir l'Annexe C.
Classe parente	AutomationMLLogicInterfaceClassLib/LogicModelInterface
Chemin pour la référence de l'élément	AutomationMLLogicInterfaceClassLib/InterlockingLogicModelInterface
Attributs	aucun

10.3.6 InterfaceClass LogicModelElementInterface

La classe d'interfaces "LogicModelElementInterface" doit être utilisée comme cela est spécifié dans le Tableau 29.

Tableau 29 – InterfaceClass LogicModelElementInterface

Nom de classe	LogicModelElementInterface
Description	La classe d'interfaces "LogicModelElementInterface" doit être utilisée pour référencer un élément dans un modèle logique archivé dans un FunctionBlock à l'intérieur d'un document XML de logique AML externe. Les détails sont spécifiés à l'Article 11. Pour des exemples, voir l'Annexe B.
Classe parente	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector
Chemin pour la référence de l'élément	AutomationMLLogicInterfaceClassLib/LogicModelElementInterface
Attributs	aucun

10.3.7 InterfaceClass VariableInterface

La classe d'interfaces "VariableInterface" doit être utilisée comme cela est spécifié dans le Tableau 30.

Tableau 30 – InterfaceClass VariableInterface

Nom de classe	VariableInterface
Description	La classe d'interfaces "VariableInterface" doit être utilisée pour référencer une variable dans un modèle logique archivé dans un FunctionBlock à l'intérieur d'un document XML de logique AML externe. Les détails sont spécifiés à l'Article 11. Pour des exemples, voir l'Annexe B.
Classe parente	AutomationMLLogicInterfaceClassLib/LogicModelElementInterface
Chemin pour la référence de l'élément	AutomationMLLogicInterfaceClassLib/VariableInterface
Attributs	Direction (type="xs:string") Cet attribut doit être utilisé pour décrire une direction de VariableInterface. Valeurs admises: "In", "Out" ou "InOut". La valeur d'attribut doit être "In" si la variable est consommée par CAEX. La valeur d'attribut doit être "Out" si le flux d'informations part de CAEX vers la variable. La valeur d'attribut doit être "InOut" si les deux directions sont possibles. L'utilisation de l'attribut doit être facultative.

10.3.8 InterfaceClass InterlockingVariableInterface

La classe d'interfaces "InterlockingVariableInterface" doit être utilisée comme cela est spécifié dans le Tableau 31.

Tableau 31 – InterfaceClass InterlockingVariableInterface

Nom de classe	InterlockingVariableInterface	
Description	La classe d'interfaces "InterlockingVariableInterface" doit être utilisée pour référencer la variable de sortie d'un modèle logique contenant les informations de verrouillage archivées dans un FunctionBlock à l'intérieur d'un document XML de logique AML externe. Cette variable représente le résultat d'une fonction décrivant la condition de verrouillage. Cette classe d'interfaces doit uniquement être utilisée par les groupes sources de verrouillage et les groupes cibles de verrouillage. Les détails sont spécifiés à l'Article 12. Pour des exemples, voir l'Annexe C.	
Classe parente	AutomationMLLogicInterfaceClassLib/VariableInterface	
Chemin pour la référence de l'élément	AutomationMLLogicInterfaceClassLib/InterlockingVariableInterface	
Attributs	SafeConditionEquals (type="xs:boolean")	L'attribut "SafeConditionEquals" doit être utilisé pour spécifier le résultat de la fonction décrivant la condition de verrouillage. Les valeurs admises doivent être "true" ou "false" et doivent indiquer la valeur d'une variable Boolean unique qui représente l'état de sécurité d'un groupe source de verrouillage. L'utilisation de l'attribut doit être facultative. La valeur par défaut doit être "true".

10.4 AutomationMLPLCopenXMLInterfaceClassLib

10.4.1 Généralités

La Figure 25 représente l'AutomationMLPLCopenXMLInterfaceClassLib normative sous la forme d'un arbre d'objets. AutomationMLPLCopenXMLInterfaceClassLib spécifie la classe d'interfaces liées à la logique pour le référencement des documents IEC 61131-10. Les détails de la classe d'interfaces sont donnés en 10.4.2. Le texte XML d'AutomationMLInterfaceClassLib est donné à l'Article D.4.

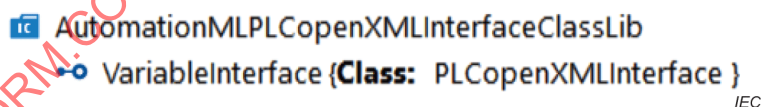


Figure 25 – AutomationMLPLCopenXMLInterfaceClassLib

10.4.2 InterfaceClass VariableInterface

La classe d'interfaces "VariableInterface" doit être utilisée comme cela est spécifié dans le Tableau 32.

Tableau 32 – InterfaceClass VariableInterface

Nom de classe	VariableInterface
Description	La classe d'interfaces "VariableInterface" doit être utilisée pour publier les signaux ou variables défini(e)s à l'intérieur d'un document XML PLCopen de PLCopenXML version 2.0, 2.01 ou IEC 61131-10.
Classe parente	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector/PLCopenXMLInterface
Chemin pour la référence de l'élément	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector/PLCopenXMLInterface/VariableInterface
Attributs	aucun

10.5 AutomationMLInterfaceClassLib

10.5.1 Généralités

Le Paragraphe 10.5 définit les spécifications des classes d'interfaces normalisées AML. Ces classes font partie intégrante de la bibliothèque normalisée AML. Tous les attributs et interfaces décrits font partie de la bibliothèque normalisée et peuvent être retirés dans InstanceHierarchy s'ils ne sont pas nécessaires. Les détails des classes d'interfaces sont donnés en 10.5.3. Le texte XML d'AutomationMLInterfaceClassLib est donné dans l'IEC 62714-1.

10.5.2 InterfaceClass InterlockingConnector

La classe d'interfaces "InterlockingConnector" est une classe de la bibliothèque normalisée de classes d'interfaces AML, définie dans l'IEC 62714-1, et doit être utilisée comme cela est spécifié dans le Tableau 33.

Tableau 33 – InterfaceClass InterlockingConnector

Nom de classe	InterlockingConnector
Description	La classe d'interfaces "InterlockingConnector" doit être utilisée pour modéliser les relations entre un groupe source de verrouillage et un groupe cible de verrouillage. Les détails sont spécifiés à l'Article 12. Pour des exemples, voir l'Annexe C.
Classe parente	AutomationMLInterfaceClassLib/AutomationMLBaseInterface
Chemin pour la référence de l'élément	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/InterlockingConnector
Attributs	aucun

10.5.3 InterfaceClass PLCopenXMLInterface

La classe d'interfaces "PLCopenXMLInterface" est une classe de la bibliothèque normalisée de classes d'interfaces AML, définie dans l'IEC 62714-1, et doit être utilisée comme cela est spécifié dans le Tableau 34.

Tableau 34 – InterfaceClass PLCopenXMLInterface

Nom de classe	PLCopenXMLInterface
Description	La classe d'interfaces "PLCopenXMLInterface" doit être utilisée pour référencer les documents XML PLCopen externes de PLCopenXML version 2.0, 2.01 ou IEC 61131-10.
Classe parente	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector
Chemin pour la référence de l'élément	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector/ PLCopenXMLInterface
Attributs	aucun

11 Référencement de documents XML de logique AML

11.1 Généralités

Selon la structure de documents répartie d'AML, il est nécessaire de placer les différentes informations dans les différents documents en relation les uns avec les autres. Le présent article porte sur le référencement des objets AML (dans un document CAEX) avec les informations logiques archivées dans des documents XML de logique AML. Une vue d'ensemble informative des méthodes de référencement est fournie à l'Annexe B.

11.2 Référencement des informations concernant la logique

En ce qui concerne le référencement des informations concernant la logique, les dispositions suivantes s'appliquent:

- Une référence entre un objet AML et des informations logiques dans un document XML de logique AML doit être modélisée par un CAEX ExternalInterface utilisant l'AML InterfaceClasses "LogicModelInterface", "LogicModelElementInterface" et leurs déductions spécifiés en 10.3.
 - Les informations logiques doivent être référencées par leur URI dans l'attribut "refURI" de ces CAEX ExternalInterfaces.
 - La valeur de l'attribut "refURI" doit pointer vers les informations logiques dans le document XML de logique AML et doit suivre la syntaxe de l'URI, en utilisant le même attribut XML de logique AML "uuid" que le fragment.

NOTE 1 Exemple de valeur de refURI: file:///behaviour.xml#74e94dc8-b5a2-490c-bf0f-f0fdaa4515ef

NOTE 2 Dans la présente partie de l'IEC 62714, les UUID sont présentés sous une forme courte ("UUID1", "UUID100", etc.). Cela permet la lisibilité et agit comme un UUID réel.

NOTE 3 Il ne s'agit pas simplement de faire référence au document XML de logique AML lui-même, étant donné que le document est susceptible de contenir plusieurs modèles logiques contenant des informations logiques.

- Chaque "LogicModelElementInterface" et ses déductions doivent faire référence à une déduction correspondante de "LogicModelInterface" par l'intermédiaire d'un InternalLink.

NOTE 4 Une déduction de "LogicModelInterface" peut être associée à plusieurs "LogicModelElementInterface" et ses déductions par l'intermédiaire d'un InternalLink.

NOTE 5 Les variables peuvent ensuite être attribuées, par exemple, à des signaux d'autres objets AML ou à des variables d'autres modèles logiques, à l'aide d'InternalLinks.

- Les modèles logiques doivent se distinguer, c'est-à-dire qu'un objet logique AML ne doit pas avoir plusieurs CAEX ExternalInterface de la même AML InterfaceClass "SequencingLogicModelInterface", "BehaviourLogicModelInterface" ou l'une de leurs déductions. Chaque groupe source de verrouillage ou groupe cible de verrouillage ne doit avoir qu'un CAEX ExternalInterface de l'AML InterfaceClass "InterlockingLogicModelInterface".

NOTE 6 Par exemple, différents niveaux de détail d'un modèle logique ne sont pas admis au niveau d'un objet de logique AML.

12 Liaison d'objets AML à des informations de verrouillage

12.1 Généralités

Les informations de verrouillage sont une partie importante de l'ingénierie des systèmes de production permettant d'assurer un comportement sûr. Le présent article porte sur le concept de verrouillage qui est composé de deux niveaux de détail pour modéliser et archiver les informations concernant le verrouillage dans AML. Une vue d'ensemble informative du concept de verrouillage est fournie à l'Annexe C.

NOTE Dans la présente partie de l'IEC 62714, le verrouillage concerne les dépendances en matière de sécurité entre les matériels (cellules de détection qui évitent d'être blessé par un robot, par exemple). Dans la présente partie de l'IEC 62714, le verrouillage ne concerne pas les dépendances fonctionnelles pertinentes entre les matériels, par exemple celui de deux robots collaborants dont il est nécessaire qu'il ait terminé son opération.

12.2 Référencement des informations concernant le verrouillage

Sur le premier niveau de détail du concept de verrouillage, les groupes sources de verrouillage et les groupes cibles de verrouillage sont liés pour décrire les dépendances fonctionnelles entre les groupes d'objets AML (les objets AML respectivement). Pour ce faire, les dispositions suivantes s'appliquent:

- Un groupe cible de verrouillage doit être créé en utilisant le concept AML Group de l'IEC 62714-1.
 - Pour identifier le groupe comme un groupe cible de verrouillage, la classe de rôles "InterlockingTargetGroup" doit être utilisée. Cela est décrit à l'Article 11.

NOTE 1 Les objets d'un groupe cible de verrouillage exécutent les actions nécessaires pour assurer la sécurité fonctionnelle.

- Un groupe source de verrouillage doit être créé en utilisant le concept AML Group de l'IEC 62714-1.
 - Pour identifier le groupe comme un groupe source de verrouillage, la classe de rôles "InterlockingSourceGroup" doit être utilisée. Cela est décrit en 10.2.3.

NOTE 2 Les objets d'un groupe source de verrouillage indiquent à quel moment des actions sont nécessaires pour assurer la sécurité fonctionnelle.

- Chaque groupe cible de verrouillage et chaque groupe source de verrouillage doivent avoir le CAEX ExternalInterface de l'AML InterfaceClass "InterlockingConnector". Cela est décrit en 10.5.2.
 - Le CAEX ExternalInterface du groupe source de verrouillage doit être lié au CAEX ExternalInterface du groupe cible de verrouillage correspondant par l'intermédiaire d'un InternalLink.
- Un objet AML doit se trouver dans aucun ou plusieurs groupes de verrouillage.
- Il doit être admis qu'un objet AML se trouve en même temps dans un groupe source de verrouillage et dans un groupe cible de verrouillage.
- Il doit être admis qu'au moins un groupe source de verrouillage soit relié à au moins un groupe cible de verrouillage (c'est-à-dire que l'attribution entre deux groupes n'est pas limitée à une association de type 1:1)
- Les relations entre les différents groupes doivent être indépendantes.

NOTE 3 L'état du groupe source de verrouillage affecte le groupe cible de verrouillage.

Sur le second niveau de détail du concept de verrouillage, le simple regroupement d'objets AML dans des groupes sources de verrouillage et des groupes cibles de verrouillage (du premier niveau de détail) est étendu par une fonction décrivant la condition de verrouillage des groupes sources de verrouillage. Pour ce faire, les dispositions suivantes s'appliquent:

- La fonction doit être représentée par un FunctionBlock de type "FBD", qui est décrit à l'Article 7.

- Ce FunctionBlock doit être référencé par le CAEX ExternalInterface de la classe d'interfaces AML "InterlockingLogicModelInterface", comme cela est spécifié en 10.3.5.
- Le FunctionBlock doit donner lieu à une variable Boolean unique décrivant le résultat d'évaluation respectif de la sortie de ce FunctionBlock.

NOTE 4 Cela indique si l'état de sécurité du groupe source de verrouillage est représenté par une valeur "true" ou "false".

- La variable Boolean unique doit être référencée par un groupe source de verrouillage modélisant un CAEX ExternalInterface de l'AML InterfaceClass "InterlockingVariableInterface", comme cela est spécifié en 10.3.8.
 - Ce CAEX ExternalInterface peut comporter un attribut "SafeConditionEquals" qui doit archiver la valeur Boolean indiquant l'état de sécurité. La valeur par défaut est "true".
- Après le mécanisme décrit en 10.4.2, les ExternalInterfaces des variables d'entrée et de sortie du FunctionBlock doivent être associés à l'ExternalInterface du modèle logique de verrouillage par des InternalLinks.
 - La modélisation de la réaction des groupes cibles de verrouillage correspondants doit suivre les mêmes dispositions que celles définies pour le groupe source de verrouillage.

IECNORM.COM : Click to view the full PDF of IEC 62714-4:2020

Annexe A (Informative)

Exemples d'archivage des modèles logiques dans un document XML de logique AML

A.1 Exemple de diagrammes de Gantt

A.1.1 Généralités

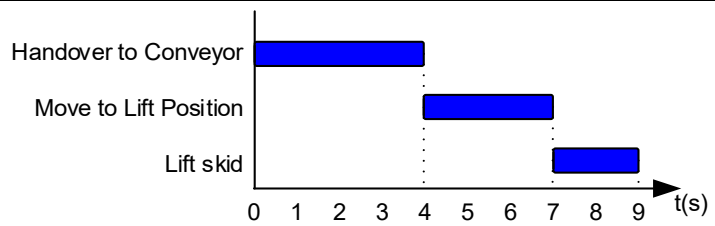
Les exemples donnés du Tableau A.1 au Tableau A.4 présentent l'utilisation des règles de mapping pour transformer des diagrammes de Gantt en IML.

A.1.2 Archivage des activités sans relation prédécesseur/successeur

L'exemple de diagramme de Gantt dans le Tableau A.1 ne présente aucune relation prédécesseur/successeur entre les barres.

IECNORM.COM : Click to view the full PDF of IEC 62714-4:2020

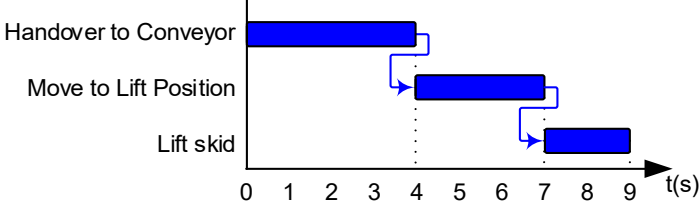
Tableau A.1 – Exemple d'archivage de diagramme de Gantt "activités sans relation prédécesseur/successeur"

Type	Représentation								
Diagramme de Gantt	 <table border="1" data-bbox="383 582 1300 683"> <thead> <tr> <th>Anglais</th><th>Français</th></tr> </thead> <tbody> <tr> <td>Handover to Conveyor</td><td>Transfert vers le transporteur</td></tr> <tr> <td>Move to lift position</td><td>Déplacement en position de relevage</td></tr> <tr> <td>Lift skid</td><td>Relever le patin</td></tr> </tbody> </table>	Anglais	Français	Handover to Conveyor	Transfert vers le transporteur	Move to lift position	Déplacement en position de relevage	Lift skid	Relever le patin
Anglais	Français								
Handover to Conveyor	Transfert vers le transporteur								
Move to lift position	Déplacement en position de relevage								
Lift skid	Relever le patin								
XML de logique AML	<pre> <?xml version="1.0" encoding="utf-8"?> <aml:AMLLogic xmlns="http://www.iec.ch/public/TC65SC65BWG7TF10" xmlns:aml="http://www.automationml.org/IEC62714-4Ed1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.automationml.org/IEC62714-4Ed1 AMLLogicXML.xsd" schemaVersion="1.0"> <aml:WriterHeader> <aml:WiterName>ToolA</aml:WiterName> <aml:WiterID>010990003</aml:WiterID> <aml:WriterVender>Vendor1</aml:WriterVender> <aml:WriterVenderURL>https://www.automationml.org/</aml:WriterVenderURL> <aml:WriterVersion>1.00</aml:WriterVersion> <aml:LastWritingDateTime>2002-05-30T09:00:00</aml:LastWritingDateTime> <aml:WriterCulturalSetting>en-US</aml:WriterCulturalSetting> <aml:WriterProjectTitle>SampleXML</aml:WriterProjectTitle> <aml:WriterProjectID>829e1797-dd62-44ac-b29c-c6786d93ca93</aml:WriterProjectID> <aml:AdditionalInformation>Modèle de document XML.</aml:AdditionalInformation> </aml:WriterHeader> <aml:Types> <aml:FunctionBlock uuid="2edd1b2e-64e1-4d2f-9fa6-12022d2ea768" name="Gantt-Chart_Example"> <aml:MainBody xsi:type="aml:IML" logicModelType="GanttChart"> <aml:TimeInformation timeFormat="xsd:duration" startOfScale="PT0S" endOfScale="PT9S" scaleInterval="PT1S"/> <aml:IMLStep name="Handover to Conveyor" uuid="f103c162-c087-48b8-a755-3293f63d8fbf" startTime="PT0S" endTime="PT4S"/> <aml:IMLStep name="Move to Lift Position" uuid="415d47ed-c13d-4727-8f13-bc8f65921ee6" startTime="PT4S" endTime="PT7S"/> <aml:IMLStep name="Lift skid" uuid="d60104de-ec05-46df-967b-5ab846e85888" startTime="PT7S" endTime="PT9S"/> </aml:MainBody> </aml:FunctionBlock> </aml:Types> </aml:AMLLogic> </pre>								

A.1.3 Archivage d'une séquence d'activités

Le Tableau A.2 décrit un diagramme de Gantt avec une séquence prédécesseur/successeur entre les barres.

Tableau A.2 – Exemple d'archivage de diagramme de Gantt "séquence d'activités"

Type	Représentation								
Diagramme de Gantt	 <table border="1"> <thead> <tr> <th>Anglais</th><th>Français</th></tr> </thead> <tbody> <tr> <td>Handover to Conveyor</td><td>Transfert vers le transporteur</td></tr> <tr> <td>Move to lift position</td><td>Déplacer en position de relevage</td></tr> <tr> <td>Lift skid</td><td>Relever le patin</td></tr> </tbody> </table>	Anglais	Français	Handover to Conveyor	Transfert vers le transporteur	Move to lift position	Déplacer en position de relevage	Lift skid	Relever le patin
Anglais	Français								
Handover to Conveyor	Transfert vers le transporteur								
Move to lift position	Déplacer en position de relevage								
Lift skid	Relever le patin								
XML de logique AML	<pre> <?xml version="1.0" encoding="utf-8"?> <aml:AMLLogic xmlns="http://www.iec.ch/public/TC65SC65BWG7TF10" xmlns:aml="http://www.automationml.org/IEC62714-4Ed1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.automationml.org/IEC62714-4Ed1 AMLLogicXML.xsd" schemaVersion="1.0"> <aml:WriterHeader> <aml:WiterName>ToolA</aml:WiterName> <aml:WiterID>010990003</aml:WiterID> <aml:WriterVender>Vendor1</aml:WriterVender> <aml:WriterVenderURL>https://www.automationml.org/</aml:WriterVenderURL> <aml:WriterVersion>1.00</aml:WriterVersion> <aml:LastWritingDateTime>2002-05-30T09:00:00</aml:LastWritingDateTime> <aml:WriterCulturalSetting>en-US</aml:WriterCulturalSetting> <aml:WriterProjectTitle>SampleXML</aml:WriterProjectTitle> <aml:WriterProjectID>829e1797-dd62-44ac-b29c-c6786d93ca93</aml:WriterProjectID> <aml:AdditionalInformation>Modèle de document XML.</aml:AdditionalInformation> </aml:WriterHeader> <aml:Types> <aml:FunctionBlock uuid="2edd1b2e-64e1-4d2f-9fa6-12022d2ea768" name="Gantt-Chart_Example"> <aml:MainBody xsi:type="aml:IML" logicModelType="GanttChart"> <aml:TimeInformation timeFormat="xsd:duration" startOfScale="PT0S" endOfScale="PT9S" scaleInterval="PT1S"> </aml:TimeInformation> <aml:IMLStep name="Handover to Conveyor" uuid="f103c162-c087-48b8-a755-3293f63d8bf" startTime="PT0S" endTime="PT4S"> <ConnectionPointOut connectionPointOutId="1"/> </aml:IMLStep> <aml:IMLStep name="Move to Lift Position" uuid="415d47ed-c13d-4727-8f13-bc8f65921ee6" startTime="PT4S" endTime="PT7S"> <ConnectionPointOut connectionPointOutId="2"/> <ConnectionPointIn> <Connection refConnectionPointOutId="3"/> </ConnectionPointIn> </aml:IMLStep> <aml:IMLStep name="Lift skid" uuid="d60104de-ec05-46df-967b-5ab846e85888" startTime="PT7S" endTime="PT9S"> <ConnectionPointIn> <Connection refConnectionPointOutId="4"/> </ConnectionPointIn> <ConnectionPointIn> <Connection refConnectionPointOutId="3"/> </ConnectionPointIn> </aml:IMLStep> <aml:IMLTransition uuid="78cb4068-b9ea-4492-ab64-587839949a8e"> <ConnectionPointIn> <Connection refConnectionPointOutId="1"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="3"/> </aml:IMLTransition> <aml:IMLTransition uuid="909b2b5b-8575-41ea-986e-b20d81d302d7"> <ConnectionPointIn> <Connection refConnectionPointOutId="2"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="4"/> </aml:IMLTransition> </aml:MainBody> </aml:FunctionBlock> </aml:Types> </aml:AMLLogic> </pre>								

Type	Représentation
	<pre> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="4"/> </aml:IMLTransition> </aml:MainBody> </aml:FunctionBlock> </aml:Types> </aml:AMLogic> </pre>

A.1.4 Archivage d'une séquence d'activités avec divergences

Dans le Tableau A.3, une divergence simultanée est introduite pour décrire plusieurs successeurs d'une barre dans l'IML.

Tableau A.3 – Exemple d'archivage de diagramme de Gantt "séquence d'activités avec divergence"

Type	Représentation								
Diagramme de Gantt	<table border="1"> <thead> <tr> <th>Anglais</th><th>Français</th></tr> </thead> <tbody> <tr> <td>Initialize Robot 1</td><td>Initialiser Robot 1</td></tr> <tr> <td>Execute Manufacturing Robot 1</td><td>Fabrication Robot 1</td></tr> <tr> <td>Initialize Robot 2</td><td>Initialiser Robot 2</td></tr> </tbody> </table>	Anglais	Français	Initialize Robot 1	Initialiser Robot 1	Execute Manufacturing Robot 1	Fabrication Robot 1	Initialize Robot 2	Initialiser Robot 2
Anglais	Français								
Initialize Robot 1	Initialiser Robot 1								
Execute Manufacturing Robot 1	Fabrication Robot 1								
Initialize Robot 2	Initialiser Robot 2								
XML de logique AML	<pre> <?xml version="1.0" encoding="utf-8"?> <aml:AMLogic xmlns="http://www.iec.ch/public/TC65SC65BWG7TF10" xmlns:aml="http://www.automationml.org/IEC62714-4Ed1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.automationml.org/IEC62714-4Ed1 AMLLogicXML.xsd" schemaVersion="1.0"> <aml:WriterHeader> <aml:WiterName>ToolA</aml:WiterName> <aml:WiterID>010990003</aml:WiterID> <aml:WriterVender>Vendor1</aml:WriterVender> <aml:WriterVenderURL>https://www.automationml.org/</aml:WriterVenderURL> <aml:WriterVersion>1.00</aml:WriterVersion> <aml:LastWritingDateTime>2002-05-30T09:00:00</aml:LastWritingDateTime> <aml:WriterCulturalSetting>en-US</aml:WriterCulturalSetting> <aml:WriterProjectTitle>SampleXML</aml:WriterProjectTitle> <aml:WriterProjectID>829e1797-dd62-44ac-b29c-c6786d93ca93</aml:WriterProjectID> <aml:AdditionalInformation>Modèle de document XML.</aml:AdditionalInformation> </aml:WriterHeader> <aml:Types> <aml:FunctionBlock uuid="2edd1b2e-64e1-4d2f-9fa6-12022d2ea768" name="Gantt- Chart_Example"> <aml:MainBody xsi:type="aml:IML" logicModelType="GanttChart"> <aml:TimeInformation timeFormat="xsd:duration" startOfScale="PT0S" endOfScale="PT18S" scaleInterval="PT1S"> </aml:TimeInformation> <aml:IMLStep name="Initialiser Robot 1" uuid="f103c162-c087-48b8-a755-3293f63d8bf" startTime="PT0S" endTime="PT6S"> <ConnectionPointOut connectionPointOutId="1"/> </aml:IMLStep> <aml:IMLStep name="Fabrication Robot 1" uuid="415d47ed-c13d-4727-8f13-bc8f65921ee6" startTime="PT9S" endTime="PT18S"> <ConnectionPointIn> <Connection refConnectionPointOutId="3"/> </ConnectionPointIn> </aml:IMLStep> <aml:IMLStep name="Initialiser Robot 2" </pre>								

Type	Représentation
	<pre> uuid="d60104de-ec05-46df-967b-5ab846e85888" startTime="PT6S" endTime="PT10S"> <ConnectionPointIn> <Connection refConnectionPointOutId="4"/> </ConnectionPointIn> </aml:IMLStep> <aml:IMLTransition uuid="78cb4068-b9ea-4492-ab64-587839949a8e"> <ConnectionPointIn> <Connection refConnectionPointOutId="1"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="2"/> </aml:IMLTransition> <aml:IMLSimultaneousDivergence uuid="ffc63a33-5f4f-48fe-98d3-6a49deaaec1"> <ConnectionPointIn> <Connection refConnectionPointOutId="2"/> </ConnectionPointIn> <ConnectionPointOut connectionPointOutId="3"/> <ConnectionPointOut connectionPointOutId="4"/> </aml:IMLSimultaneousDivergence> </aml:MainBody> </aml:FunctionBlock> </aml:Types> </aml:AMLLogic> </pre>

A.1.5 Archivage d'une séquence d'activités avec convergences

L'exemple de diagramme de Gantt dans le Tableau A.4 inclut une convergence avec la séquence de prédécesseur entre les barres du diagramme de Gantt. Cela donne une convergence simultanée dans l'IML.

Tableau A.4 – Exemple d'archivage de diagramme de Gantt "séquence d'activités avec convergences"

Type	Représentation								
Diagramme de Gantt	<table border="1"> <thead> <tr> <th>Anglais</th><th>Français</th></tr> </thead> <tbody> <tr> <td>Lift skid</td><td>Relever le patin</td></tr> <tr> <td>Initialize Robot 1</td><td>Initialiser Robot 1</td></tr> <tr> <td>Execute Manufacturing Robot 1</td><td>Fabrication Robot 1</td></tr> </tbody> </table>	Anglais	Français	Lift skid	Relever le patin	Initialize Robot 1	Initialiser Robot 1	Execute Manufacturing Robot 1	Fabrication Robot 1
Anglais	Français								
Lift skid	Relever le patin								
Initialize Robot 1	Initialiser Robot 1								
Execute Manufacturing Robot 1	Fabrication Robot 1								
XML de logique AML	<pre> <?xml version="1.0" encoding="utf-8"?> <aml:AMLLogic xmlns="http://www.iec.ch/public/TC65SC65BWG7TF10" xmlns:aml="http://www.automationml.org/IEC62714-4Ed1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.automationml.org/IEC62714-4Ed1 AMLLogicXML.xsd" schemaVersion="1.0"> <aml:WriterHeader> <aml:WiterName>ToolA</aml:WiterName> <aml:WiterID>010990003</aml:WiterID> <aml:WriterVender>Vendor1</aml:WriterVender> <aml:WriterVenderURL>https://www.automationml.org/</aml:WriterVenderURL> <aml:WriterVersion>1.00</aml:WriterVersion> <aml:LastWritingDateTime>2002-05-30T09:00:00</aml:LastWritingDateTime> <aml:WriterCulturalSetting>en-US</aml:WriterCulturalSetting> <aml:WriterProjectTitle>SampleXML</aml:WriterProjectTitle> <aml:WriterProjectID>829e1797-dd62-44ac-b29c-c6786d93ca93</aml:WriterProjectID> <aml:AdditionalInformation>Modèle de document XML.</aml:AdditionalInformation> </aml:WriterHeader> <aml:Types> <aml:FunctionBlock uuid="2edd1b2e-64e1-4d2f-9fa6-12022d2ea768" name="Gantt-Chart_Example"> <aml:MainBody xsi:type="aml:IML" logicModelType="GanttChart"> <aml:TimeInformation timeFormat="xsd:duration" startOfScale="PT0S" </pre>								