<AutomationML/>

The Glue for Seamless
Automation Engineering

AutomationML Whitepaper Part 3 – Geometry and Kinematics

Document Identifier: WP Geo, V 2.0.0

State: January 2017

Version 2.0.0, January 2017

Contact: www.automationml.org

Table of content

List of Figures

List of Tables

1 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 AMLis 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.

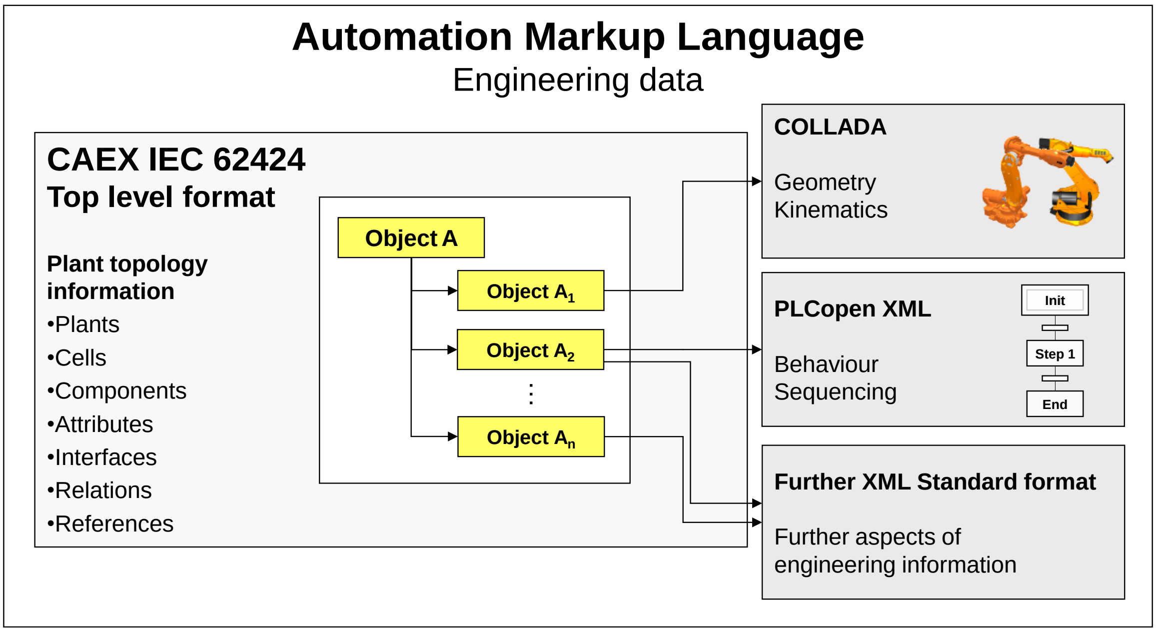

Figure 1: Overview of the engineering data exchange format AML

Due 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 of logics, sequencing, behaviour and control related information.

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

Clause 3 describes the geometry related extensions of the role class libraries.

Clause 4 describes the frame attribute which can be used to represent the geometric position of an InternalElement, InstanceHierarchy, SystemUnitClass, or SystemUnitClassLibrary with respect to another CAEX Object.

Clause 5 gives a normative description regarding referencing COLLADA documents.

Clause 6 specifies the normative provisions for the attachment of two geometric AML objects.

Clause 7 defines how to store meta informations about the source tool directly into the COLLADA document.

Annex A describes the referencing methods for geometric and kinematic models.

Annex B provides an example for modelling of kinematic systems and their combination in AML.

Annex C gives an informative XML representation of the libraries defined in this part of IEC 62714.

1.1 Scope

This part of IEC 62714 specifies the integration of geometry and kinematics information for the exchange between engineering tools in the plant automation area by means of AML.

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

1.2 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 62714-1:2014, Engineering data exchange format for use in industrial automation systems engineering – Automation markup language – Part 1: Architecture and general requirements

IEC 62714-2:2015,Engineering data exchange format for use in industrial automation systems engineering – Automation markup language – Part 2: Role class libraries

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

COLLADA 1.4.1:March 2008 COLLADA – Digital Asset Schema Release 1.4.1
(available at <http://www.khronos.org/files/collada_spec_1_4.pdf>) [access on 23th Juni 2016]

Extensible Markup Language (XML) 1.0:2004, W3C Recommendation
(available at <http://www.w3.org/TR/2004/REC-xml-20040204/>)[access on 23th Juni 2016]

2 Terms, definitions and abbreviations

2.1 Terms and definitions

For the purpose of this document, the terms and definitions of IEC 62714-1:2014and of IEC 62714-2:201X apply and in addition the following terms and definitions.

SID path

For the purposes of this document, the terms and definitions given in IEC 62714-1:2014 and of IEC 62714-2:2015and 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/ [access on 26th September 2016]
- ISO Online browsing platform: available at http://www.iso.org/obp[access on 26th September 2016]

2.2 Abbreviations

For the purposes of this document, the abbreviations of IEC 62714-1:2014 and of IEC 62714-2:2015 apply and in addition the abbreviations listed in Table 1.

SCARA	Selective Compliance Assembly Robot Arm
SID	Scoped Identifier

Table 1: Abbreviations

2.3 Conformity

To claim conformity to the present document with respect to the support of AML, the requirements of Clauses 3, 4, 5, 6 and 7 shall be fulfilled. In the scope of AML, a COLLADA document shall conform to the specification of ISO/PAS 17506 or COLLADA 1.4.1.

3 Extensions of AML libraries for geometry and kinematics

3.1 General

Clause 5 defines extensions of the standard AMLRoleClasses and standard AMLInterfaceClasses. These classes are part of the AML standard libraries and a specific extension of IEC 62714-1 for this part of IEC 62714. All described attributes are part of the standard libraries and may be removed in the InstanceHierarchy if not needed.

3.2 AutomationMLBaseRoleClassLib

The RoleClass “Frame” shall be used as specified in Table 2.

Class name	Frame
Description	This role denotes a Cartesian right handed coordinate system.
Parent Class	AutomationMLBaseRoleClassLib/AutomationMLBaseRole
Path for element reference	AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Frame

Table 2: RoleClass Frame

NOTE 1 An AML object referencing the RoleClass “Frame” is useable to define reference coordinate systems like attachment points.

NOTE 2 To define reference coordinate systems like attachment pointsan AML object referencing the RoleClass “Frame” is useable.

3.3 AutomationMLInterfaceClassLib

3.3.1 InterfaceClass COLLADAInterface

The InterfaceClass “COLLADAInterface” shall be used as specified in Table 3.

Class name	COLLADAInterface	
Description	The InterfaceClass “COLLADAInterface” shall be used in order to reference external COLLADA documents and to publish interfaces that are defined inside an external COLLADA document.	
Parent class	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector	
Path for element reference	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector/ COLLADAInterface	
Attributes	refType(AttributeDataType=”xs:string”)	The attribute “refType” specifies whether the referenced COLLADA document has an explicit or implicit character. The allowed values are “explicit” or “implicit”. The attribute is mandatory.
	target (AttributeDataType=”xs:token”)	The attribute “target” specifies the SID path of a COLLADA element within the referenced document. The attribute is optional.

Table 3: InterfaceClass COLLADAInterface

NOTE 1 An AML object referencing the RoleClass “Frame” is useable to define reference coordinate systems like attachment points.

NOTE 2 To define reference coordinate systems like attachment pointsan AML object referencing the RoleClass “Frame” is useable.

3.3.2 InterfaceClass AttachmentInterface

The InterfaceClass “AttachmentInterface” shall be used as specified in Table 4.

Class name	AttachmentInterface
Description	The InterfaceClass “AttachmentInterface” specifies an interface for geometric or kinematic linksbetween AML objects with RoleClass Frame.
Parent class	AutomationMLInterfaceClassLib/AutomationMLBaseInterface
Path for element reference	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/AttachmentInterface

Table 4: InterfaceClass AttachmentInterface

4 Frame attribute

An InternalElement or a SystemUnitClass may require a frame attribute that represents its geometric position in relation to other objects. For this, the following provisions apply:

- Each frame shall be based on athree dimensional, orthogonal, right-handed coordinate system.
- The elements InstanceHierarchy and SystemUnitClassLib specify a three dimensional, orthogonal, right-handed coordinate system with standard basis. The positive z axis is considered upward, the positive x direction defines the right axis and the negative y direction defines the forward axis.
- The relative translations$x$, y and z as well as the rotationsrx, ry and rz shall be specified as sub-attributes in an AML attribute „Frame“ defined in Table 5 and Table 6. The relative translations$x$, y and z shall be given in relation to the parent coordinate System specified in the previous point. The rotationsrx, ry and rz shall be executed in the order rx, ry and $rz$with respectto the fixed axes of the parent coordinate System. The origin of the translated Frame shall retain its position(x, y, z) during thespatially-fixed rotation. The attribute “Frame” shall affect its containingAML object and all its children.

  NOTE This means, that the point of rotation rotates whereas the frame position remains unchanged. This avoids double changes.

- If the attribute “Frame” is not specified, the default values of x, y, z, rx, ry and rz shall be 0.
- If the attribute “Frame” is specified, all sub-attributes x, y, z, rx, ry and rz shall be listed. Any unused sub-attribute shall have the default value 0.

The attribute “Frame” shall be used as complex attribute specified in Table 5 and Table 6. An example is given in A.1.2.

Attribute	AttributeDataType	Description
Frame	This attribute has no AttributeDataType since attribute has no value.	The attribute “Frame” shall be used as structure element for the storage of the sub-attributes specified in Table 6.

Table 5: Attribute “Frame”

Attribute	AttributeDataType	Description
x	xs:double	The attribute “x” shall be used to specify the relative position in meters along the x axis of the coordinate system specified by the parent element. The value of attribute “Unit” of the CAEX element “Attribute” shall be “m”.
y	xs:double	The attribute “y” shall be used to specify the relative position in meters along the y axis of the coordinate system specified by the parent element. The value of attribute “Unit” of the CAEX element “Attribute” shall be “m”.
z	xs:double	The attribute “z” shall be used to specify the relative position in meters along the z axis of the coordinate system specified by the parent element. The value of attribute “Unit” of the CAEX element “Attribute” shall be “m”.
rx	xs:double	The attribute “rx” shall be used to specify the relative rotation in degrees around the x axis of the coordinate system specified by the parent element. The value of attribute “Unit” of the CAEX element “Attribute” shall be “deg”.
ry	xs:double	The attribute “ry” shall be used to specify the relative rotation in degrees around the y axis of the coordinate system specified by the parent element. The value of attribute “Unit” of the CAEX element “Attribute” shall be “deg”.

rz	xs:double	The attribute "rz" shall be used to specify the relative rotation in degrees around the z axis of the coordinate system specified by the parent element. The value of attribute "Unit" of the CAEX element "Attribute" shall be "deg".

Table 6: Sub-attributes of the attribute "Frame"

5 Integrationof COLLADA Documents

Regarding referencing COLLADA documents, the following provisions apply:

- A reference from an AML object to a COLLADA document shall be modelledby means of a CAEX ExternalInterface derived from the AML InterfaceClass“COLLADAInterface” as defined in3.3.1.
- The COLLADA document shall be referenced by its URI within the attribute “refURI” of this CAEX ExternalInterface.
- The value of the attribute “target” shall point at an element within the COLLADA document and shall follow the syntax of a SIDpath.
- If the attribute “target” is not present or its content is empty and the URI has no fragment, the element “scene” of the COLLADA document shall be considered.
- The COLLADA document and its entry point shall be resolved as specified in Table 7.
- A decision tree to resolve the different entries is shown in Figure A.1.
- The value “explicit” of the attribute “refType” shall be used for AML objects (henceforth named explicit AML objects) that are part of a data representation.

  NOTE 1 An example of explicit AML objects is the geometry of a robot and a conveyor in a common geometric scene.

- The value “implicit” of the attribute “refType” shall be used for AML objects (henceforth named implicit AML objects) that are used for special purposes for referencing objects that are already represented, e.g. for publishing frames or referencing of objects that are already represented.

  NOTE 2 An example of implicit AML objects is the reference of a robot’s geometry that has been already explicitly referenced. The implicit referencing avoids double representation of geometries.

- Objects in the COLLADA document only may be referenced using value “implicit” if they are part of a COLLADA object or identical with a COLLADA object, which already was referenced with value “explicit”.Implicit AML objects shall be modelled as direct or indirect children of explicit AML objects.
- In cases of a COLLADA document (henceforth named main document) referencing other COLLADA documents (henceforth named subdocuments) according to the rules of instantiation and external referencing as specified in ISO/PAS 17506 all references to elements within the referenced COLLADA subdocuments shall only be done via the COLLADA main document. An example is given in A.2.3.
- An AML object shall have zero or one “COLLADAInterface” as direct child.

		Attribute “refURI”	
		URI with fragment	**URI without fragment**
Attribute “target”	**SID path starts with an ID**	URI specifies the document. SID path specifies the entry point.	URI specifies the document. SID path specifies the entry point.
	SID path starts without an ID	URI specifies the document and a corresponding element for the entry point. SID path specifies the entry point.	Undefined status. This combination shall not be used.
	Missing or its value is empty	URI specifies the document and the entry point.	URI specifies the document. The element “scene” specifies the entry point.

Table 7: Rules for resolving document and entry point

6 Attachment of two AML objects

Clause 6 specifies the normative provisions for the attachment of two geometric AML objects. An example is given in Clause A.3.

Regarding attaching two AML objects, the following provisions apply:

- An AML object shall have zero or one “AttachmentInterface” as direct child.
- A link between two ExternalInterfaces assigned to a InterfaceClass “AttachmentInterface” shall be a unidirectional connection with fixed geometric coupling using a CAEX InternalLink element.
- The relative transformation from the AML object referenced by the attribute RefPartnerSideA to the object referenced by the attribute RefPartnerSideBshall remain constant even if the object of RefPartnerSideA is transformed, resulting into a corresponding transformation of RefPartnerSideB in case of a transformation of RefPartnerSideA. A transformation of RefPartnerSideB does not modify the transformation of RefPartnerSideA.

  NOTE 1 RefPartnerSideA and RefPartnerSideB are attributes of the CAEX InternalLink element.
- As an attachment point is part of an AML object, the “AttachmentInterface” shall be part of the corresponding AML object.
- If an “AttachmentInterface” is child of an AML object with the RoleClass “Frame” or an inherited RoleClass, the object to be moved shall be the parent AML object of the AML object with the RoleClass “Frame” or an inherited RoleClass (even if the parent AML object has the RoleClass “Frame” or an inherited RoleClass).
- If an AML object owning an “AttachmentInterface” has the RoleClass “Frame” or an inherited RoleClass and it has an interface of type “COLLADAInterface”, the parent AML object shall have an interface of type “COLLADAInterface” as well. The “COLLADAInterface” of the parent AML object represents the boundary of the geometry of the attached geometry.
- If an attachment refers to a geometric element, a “COLLADAInterface“ shall reference a COLLADA node of the visual scene. The attachment shall move the complete geometry (recursive up to the node attribute of a bind_kinematics_model element) of the attached object without modification of the kinematic state of the attached object.
- If an attachment refers to a kinematic element, a “COLLADAInterface”shall reference a COLLADA element inside a COLLADA kinematics_scene element. The attachment shall affect the joint configuration of the attached kinematic object when moving the parent AML object of the attached object. The importing tool is responsible for the calculation of such coupled kinematic elements.

NOTE 2 To establish a bidirectional connection two unidirectional links are necessary, with each link in opposite direction to the other one.

NOTE 3 It is possibly to link an “AttachmentInterface” more than once.

7 Meta information about the COLLADA source tool

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

- Each COLLADA document shall provide information about the tool which has written the COLLADA document.
- In a data exchange tool chain, the last tool editing the document shall store this information in the COLLADA document.
- This information shall be stored within the child element “asset” of the element “COLLADA”.
- The meta information shall consist of:
    - vendor of the writer tool;
    - URL of the writer tool;
    - name of the writer tool;
    - product version of the writer tool;
    - product release number of the writer tool;
    - creationtimestamp and timestamp of the last modification of the COLLADA document;
    - definition of unit and up axis.
- The meta information shall be stored by means of the elements shown in Table 8.

XML tag name COLLADA 1.4.1	XML tag name ISO/PAS 17506	Type	Level	Example
“author”	“author”	xs:string	Mandatory	ToolX Vendor
“authoring_tool”	“authoring_tool”	xs:string	Mandatory	“ToolX” “0.2” “123 prealpha”
“comments”	“author_website”	xs:anyURI	Mandatory	http://www.ToolX-Vendor.org
“created”	“created”	xs:dateTime	Mandatory	2014-04-14T11:37:18.1875000
“modified”	“modified”	xs:dateTime	Mandatory	2014-04-14T11:37:18.1875000
“unit meter=… name=…”	“unit meter=… name=…”	xs:string	Mandatory	name=”meter” meter=”1.0”
“up_axis”	“up_axis”	xs:string	Mandatory	Z_UP

Table 8: Meta information about the COLLADA source tool

The XML elements shall contain the following information:

- “author” provides the vendor of the writer tool;
- “authoring_tool” provides the name, the product version and the product release number of the writer tool separated by a blank character and each of them enclosed in quotation marks;
- in case of ISO/PAS 17506 “author_website” provides the URL of the writer tool, in case of COLLADA 1.4.1 “comments” provides the URL of the writer tool;
- “created” and “modified” provide the creation timestamp and timestamp of the last modification according to the rules as specified in ISO/PAS 17506;
- “unit” and “up_axis” shall be defined as specified in ISO/PAS 17506 or COLLADA 1.4.1

The required XML text in case of ISO/PAS 17506 and COLLADA 1.4.1 by means of an example is shown inFigure 2 and Figure 3, respectively.

The required XML text in case of ISO/PAS 17506 and COLLADA 1.4.1 by means of an example is shown in Figure 2 and Figure 3, respectively.

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

Figure 2: Required XML text in case of ISO/PAS 17506

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

Figure 3: Required XML text in case of COLLADA 1.4.1

Annex A Referencing methods for geometric / kinematic descriptions

A.1 Integration of a common COLLADA document with explicit referencing

A.1.1 General

Clause A.1 describes the explicit referencing methods for geometric and kinematic models, which are located in COLLADA documents. Figure A.1 shows the different methods used in the following Subclauses by means of a decision tree.

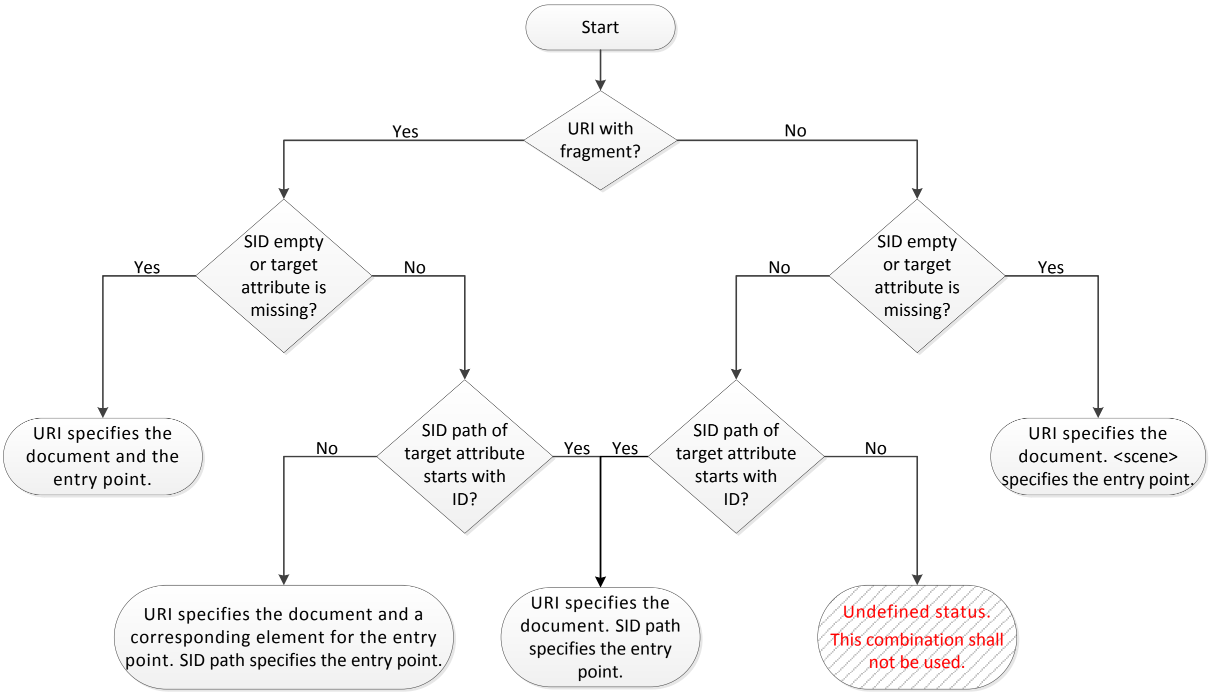

Figure A.1: Decision tree for different referencing methods

NOTE 1 In this document, path or subpath stringsare presented in a short form like “PATH1”, “PATH2”, etc. Thisserves the readability and is acting for a real path or subpath.Same numbers indicate the same path and thus the semantic equality.

NOTE 2 For better readability, only relevant partsof AML/COLLADA documents are shown. Missing parts are denoted with “[…]”.

NOTE 3 For better readability, any RefBaseClassPath attribute which points to the “AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector/COLLADAInterface” class is substituted with “PATH_CI”.

NOTE 4 For better readability, any RefBaseClassPath attribute which points to the “AutomationMLInterfaceClassLib/AutomationMLBaseInterface/AttachmentInterface” class is substituted with “PATH_AI”.

NOTE 5 For better readability, any RefBaseRoleClassPath attribute which points to the “AutomationMLDMIRoleClassLib/DiscManufacturingEquipment/StaticObject” class is substituted with “PATH_SO”.

NOTE 6 For better readability, any RefBaseRoleClassPath attribute which points to the “AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Frame” class is substituted with “PATH_FRAME”.

A.1.2 Definition of the Frame attribute

The following example shows how to define the geometric position and orientation of two CAEX Objects by using the frame attribute mentioned in Clause 6.The correspondingAML document consists of anInstanceHierarchy named “ExampleInstanceHierarchy” with two InternalElements “BaseFrame” and “ChildFrame” (see Figure A.2).

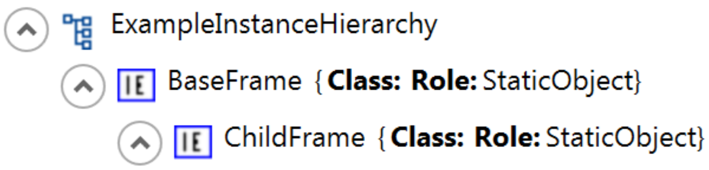

Figure A.2: Two frames represented in theInstanceHierarchy of an AML document

To describe the complete coordinate transformation of the InternalElement “ChildFrame” with respect to its parent Node “BaseFrame”, the former object owns the frame attribute. Figure A.3 shows the relevant part of the AML document and how to define the frame attribute and its subattributes.

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

Figure A.3: XML representation of the AML document

Assuming that the coordinate system $K_1(x_1,y_1,z_1)$ belongs to the InternalElement “ChildFrame”, Figure A.4 depictsits relative translation by x, y and $z$as well as its spatially fixed rotation by rx, ry and rz with respect to the three axes of the coordinate system $K_0(x_0,y_0,z_0)$ specified by the parent element “BaseFrame”. By the use a homogeneous transformationmatrix $\mathbf{H}_1^0$ and the six subattributes mentioned in Table 6, the relative translation and orientation between K_1 and K_0can be described in a compact format as follows:

$$\mathbf{H}_1^0(x,y,z,rx,ry,rz)=\begin{bmatrix}\mathbf{R}_1^0 & \mathbf{d}_1^0\\ \mathbf{0} & 1\end{bmatrix}\epsilon \mathrm{R}^{4\times 4}. \qquad (A.1)$$

Hereby the Rotationmatrix$\mathbf{R}_1^0$ expresses the orientation of K_1 relative to K_0. Clause 4 determines the order of rotation by rx, ry and rz. In the context of spatially fixed rotations, it is recalled that the appropriate rotation matrices have to be multiplied from the left hand side. Thus the rotational part of $\mathbf{H}_1^0$ is given as

$$\mathbf{R}_1^0= \mathbf{Rot}_z(rz)\mathbf{Rot}_y(ry)\mathbf{Rot}_x(rx)\ . \qquad (A.2)$$

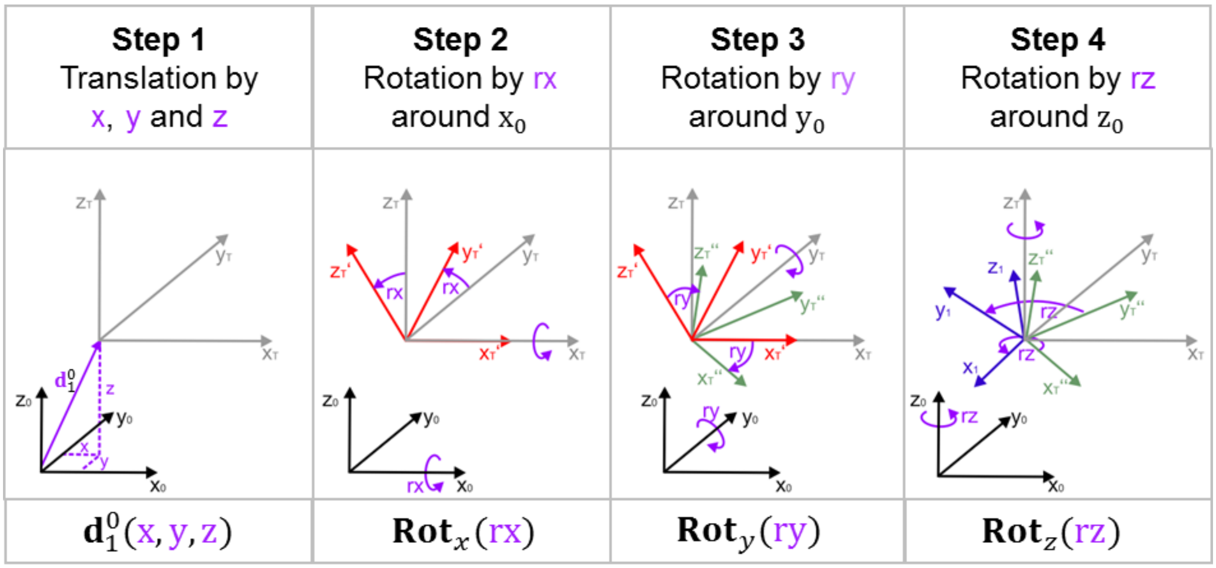

Figure A.4: Translation and spatially fixed rotation

Each rotationmatrix is characterized as

$$\mathbf{Rot}_x(rx)=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos(rx) & -\sin(rx)\\ 0 & \sin(rx) & \cos(rx)\end{bmatrix},\ \mathbf{Rot}_y(ry)=\begin{bmatrix}\cos(ry) & 0 & \sin(ry)\\ 0 & 1 & 0\\ -\sin(ry) & 0 & \cos(ry)\end{bmatrix} \text{ and}$$

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

The displacement vector is described by

$$\mathbf{d}_1^0=\begin{bmatrix}x\\ y\\ z\end{bmatrix}.$$

As shown in Figure A.4, the origin of the translated Frame retains its Cartesian coordinates x, y and z during the spatially-fixed rotation. By substituting $\sin(ri) = sr_i$ and $\cos(ri) = cr_i$, equation (A.2) and the multiplication in equation (A.1) yields the compact format

$$\mathbf{H}_1^0=\begin{bmatrix}cr_z cr_y & cr_z sr_y sr_x - sr_z cr_x & cr_z sr_y cr_x + sr_z sr_x & x\\ sr_z cr_y & sr_z sr_y sr_x + cr_z cr_x & sr_z sr_y cr_x - cr_z sr_x & y\\ -sr_y & cr_y sr_x & cr_y sr_x & z\\ 0 & 0 & 0 & 1\end{bmatrix}$$

which describesthe translation and orientation of the InternalElement “ChildFrame” with respect to its parent Node “BaseFrame”.

A.1.3 Structure of the COLLADA documents

The scene used in the following samples consists of two cubes, one on the top of the other. In the visual scene a node for each cube is instantiated. Each node references the same geometry element, but applies a different name (“RedCube”, “BlueCube”), material (red, blue) and transformation. Figure A.5 shows the rendered scene with the corresponding COLLADA scene graph.

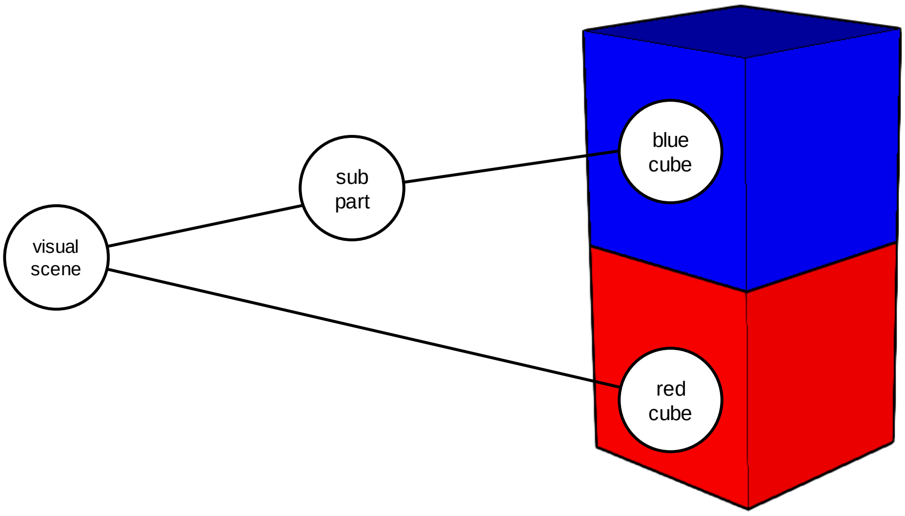

Figure A.5: COLLADA scene used in this example

In this example, two COLLADA documents are used to define the complete visual scene. The node hierarchy is defined in the COLLADA main document “red_blue_cubes.dae”. The geometry of the cube is defined in a library within the separate COLLADA subdocument “cube.dae”. Each node references the subdocument “cube.dae” and instantiates a cube geometry according to the rules of instantiation and external referencing as specified in ISO/PAS 17506. Figure A.6 depicts the structure and references used in this example.

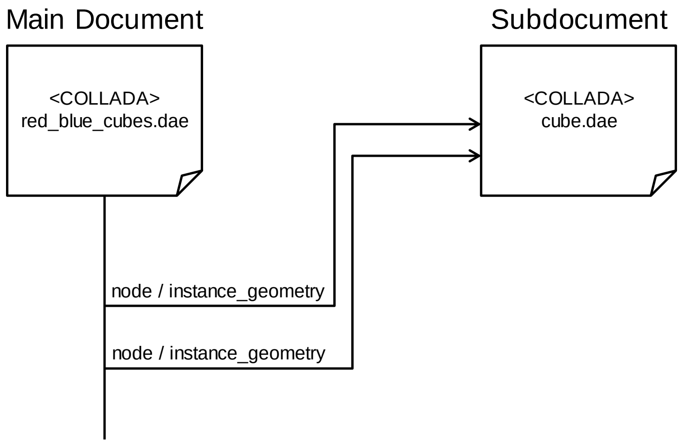

Figure A.6: Structure and References

The content of the COLLADA subdocument “cube.dae” is shown in Figure A.7.

```
<COLLADA version="1.5.0">
  <asset>
    <contributor>
      <author>WriterVendor</author>
      <author_website>http://www.WriterVendor.com</author_website>
```

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

```
    </geometry>
  </library_geometries>
</COLLADA>
```

Figure A.7: Content of the COLLADA document cube.dae

The main document (“red_blue_cubes.dae”) including the node hierarchy and external references to the cube geometry (“cube.dae”) is shown in Figure A.8. Each instantiated geometry overwrites the material symbol with its custom material definition.

```
<COLLADA version="1.5.0">
  <asset>
    <contributor>
      <author>WriterVendor</author>
      <author_website>http://www.WriterVendor.com</author_website>
      <authoring_tool>"WriterName" "WriterVersion" "WriterRelease"</authoring_tool>
      <comments>This document defines the node hierarchy and instantiates cubes from subdocuments</comments>
    </contributor>
    <created>2014-04-14T11:37:18.1875000</created>
    <modified>2014-04-14T11:37:18.1875000</modified>
    <unit meter="1.0" name="meter" />
    <up_axis>Z_UP</up_axis>
  </asset>
  <library_effects>
    <effect id="red-fx">
      <profile_COMMON>
        <technique sid="COMMON">
          <phong>
            <diffuse>
              <color>1 0 0 1</color>
            </diffuse>
          </phong>
        </technique>
      </profile_COMMON>
    </effect>
    <effect id="blue-fx">
      <profile_COMMON>
        <technique sid="COMMON">
          <phong>
            <diffuse>
              <color>0 0 1 1 </color>
            </diffuse>
          </phong>
        </technique>
      </profile_COMMON>
    </effect>
  </library_effects>
  <library_materials>
```

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

Figure A.8: Content of the COLLADA document red_blue_cubes.dae

A.1.1 Referencing using URI and fragments without a targetand ID

The following example AML document consists of anInstanceHierarchy named “ExampleInstanceHierarchy” with one single InternalElement“RedCube”. The latter includes an CAEX ExternalInterface of type “COLLADAInterface” named “FileLinkExplicit”, which is part of the standard InterfaceClassLibrary “AutomationMLInterfaceClassLib” as depicted in Figure A.9.

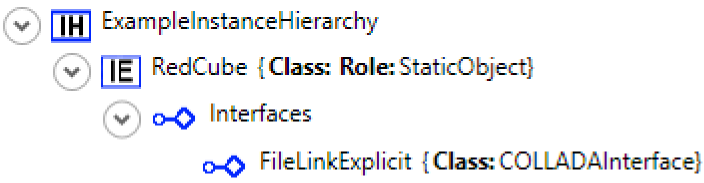

Figure A.9: “RedCube” – Hierarchy of the AML document

Figure A.10 shows the relevant part of the AML document and how to define attributes for referencing.

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

Figure A.10: XML representation of the AML document

The ExternalInterface “FileLinkExplicit” points to the COLLADA document, which is shown in Figure A.11. The red cube is referenced unambiguously by the use of the relative URI “./red_blue_cubes.dae#redcube”.

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

Figure A.11: Referencing the red cube by ID

A.1.2 Referencing using URI and fragments including a target without an ID

The AML document consists of an InstanceHierarchynamed “ExampleInstanceHierarchy”with a single InternalElement“BlueCube”. The latter one includes an ExternalInterface “COLLADAInterface” named “FileLinkExplicit”, which is part of the standard InterfaceClassLibrary “AutomationMLInterfaceClassLib” as depicted in Figure A.12.

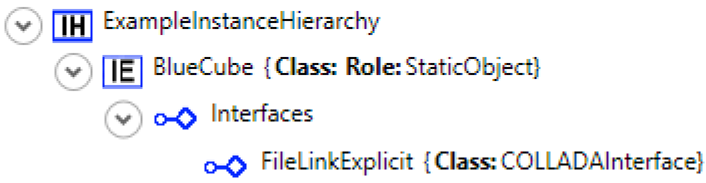

Figure A.12: "BlueCube" – Hierarchy of the AML document

Figure A.13 shows the relevant part of the AML document and how to define attributes for referencing.

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

Figure A.13: XML representation of the AML document

The ExternalInterface "FileLinkExplicit" points to the COLLADA document, which is shown in Figure A.14. First the SID's entry point is specified by the relative URI "./red_blue_cubes.dae#subpart". The target element defined in the COLLADA document named "bluecube" is then resolved by browsing from the referenced element "subpart" according to the rules of a SID.

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

Figure A.14: Referencing the blue cube

A.1.3 Referencing using URI without a fragment, including a target and an ID

The AML document consists of an InstanceHierarchynamed "ExampleInstanceHierarchy"with a single InternalElement"BlueCube". The latter one includes an ExternalInterface "COLLADAInterface" named "FileLinkExplicit", which is part of the standard InterfaceClassLibrary "AutomationMLInterfaceClassLib" as depicted in Figure A.15.

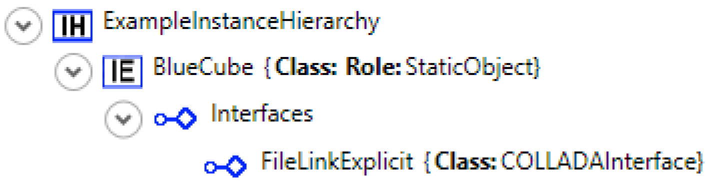

Figure A.15: Hierarchy of the AML document

Figure A.16 shows the relevant part of the AML document and how to define attributes for referencing.

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

Figure A.16: XML representation of the AML document

The ExternalInterface “FileLinkExplicit” points to the COLLADA document, which is shown in Figure A.17. First the SID’s entry point is specified by the relative URI “./red_blue_cubes.dae”. The target element defined in the COLLADA document named “bluecube” is then resolved by browsing from the entry point of the SID “subpart/bluecube”.

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

Figure A.17: Referencing the blue cube starting from the element “subpart”

A.1.4 Referencing using URI and fragments including a target and an ID

The AML document consists of an InstanceHierarchynamed “ExampleInstanceHierarchy”with a single InternalElement“BlueCube”. The latter one includes an ExternalInterface “COLLADAInterface” named “FileLinkExplicit”, which is part of the standard InterfaceClassLibrary “AutomationMLInterfaceClassLib” as depicted in Figure A.18.

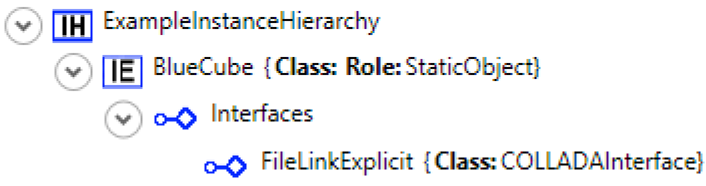

Figure A.18: Hierarchy of the AML document

Figure A.19 shows the relevant part of the AML document and how to define attributes for referencing.

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

Figure A.19: XML representation of the AML document

The ExternalInterface “FileLinkExplicit” points to the COLLADA document, which is shown in Figure A.20. In this example the blue cube is addressed as already shown in A.1.3. Initially the SID’s entry point is specified by the relative URI “./red_blue_cubes.dae#redcube”. At this point it is demonstrated that the fragment of the specified URI is actually not necessary for referencing the blue cube. In this case the blue cube is resolved by evaluating the SID beginning with an ID. As the ID of any SID must be unique across the entire document, the proposed referencing is unambiguous as well.

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

Figure A.20: Referencing the blue cube

A.1.5 Referencing using URI without afragment, target and ID

The AML document consists of an InstanceHierarchy named “ExampleInstanceHierarchy”with a single InternalElement“RedAndBlueCube”. The latter one includes an ExternalInterface “COLLADAInterface” named “FileLinkExplicit”, which is part of the standard InterfaceClassLibrary “AutomationMLInterfaceClassLib” as depicted in Figure A.21.

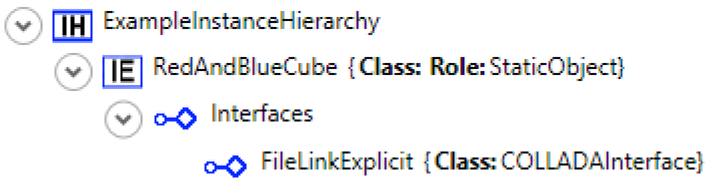

Figure A.21: Hierarchy of the AML document

Figure A.22 shows the relevant part of the AML document and how to define attributes for referencing.

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

Figure A.22: XML representation of the AML document

The ExternalInterface “FileLinkExplicit” points to the COLLADA document, which is shown in Figure A.23. In this sample the complete COLLADA element “scene” is addressed. Both the red and the blue cube are referenced unambiguously by the use of the relative URI “./red_blue_cubes.dae” without any additional fragment or target.

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

```
            </technique_common>
          </bind_material>
        </instance_geometry>
      </node>
    </node>
  </visual_scene>
</library_visual_scenes>
<scene>
  <instance_visual_scene url="#visualscene" />
</scene>
</COLLADA>
```

Figure A.23: Referencing the complete COLLADA scene

A.2 Implicit referencing of COLLADA elements

A.2.1 General

Clause A.2 describes the implicit referencing method for geometric and kinematic models, which are located in external COLLADA documents. Implicit referencing is used to refer from a CAEX object to a COLLADA object and to provide additional information about the COLLADA object at CAEX level, e. g. a frame for providing attachment capabilities. In this case the COLLADA object is already known from a different context, e. g. from an explicit reference (see Clause A.1). Subclause A.2.2 describes the implicit referencing to a single COLLADA document, whereas A.2.3 describes the implicit referencing tonested COLLADA documents. Subclause A.2.4 describes how to publish elements respectively frames of a COLLADA document.

A.2.2 Implicit referencing

The following sample uses the visual scene taken from Subclause A.1.2 (COLLADA document “red_blue_cubes.dae”). The AML document consists of an InstanceHierarchy named “ExampleInstanceHierarchy”with three InternalElements as depicted in Figure A.24. The InternalElement “RedAndBlueCubeExplicit” includes an ExternalInterface “COLLADAInterface” named “FileLinkExplicit” Both InternalElements “BlueCubeImplicit” and “RedCubeImplicit” include an ExternalInterface “COLLADAInterface” named “FileLinkImplicit”, which are part of the standard InterfaceClassLibrary “AutomationMLInterfaceClassLib”.

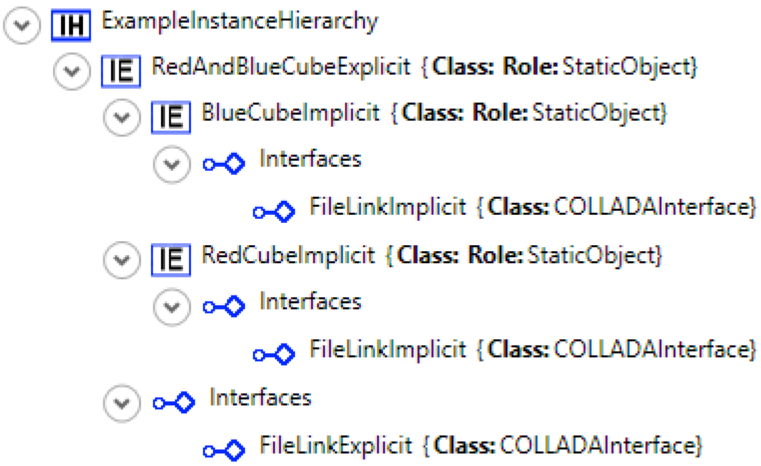

Figure A.24: Implicit Referencing: Hierarchy of the AML document

The ExternalInterface “FileLinkExplicit” of the InternalElement“RedAndBlueCubeExplicit” points to the COLLADA document and addresses the complete COLLADA scene explicitly (see subclause A.1.5).

Both the red and the blue cube are referenced unambiguously by the use of the relative URI "./red_blue_cubes.dae" without any additional fragment or target.

The InternalElements "RedCubeImplicit" and "BlueCubeImplicit" reference the red and blue cube implicitly. Figure A.25 shows the relevant part of the AML document and how to define attributes for implicit referencing.

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

Figure A.25: XML representation of the AML document

In the case of visualization it is not necessary to render both the red and blue cube once more becauseboth were visualised already via the explicit reference of the "RedAndBlueCubeExplicit" element. In this case, the explicitly referenced scene ("RedAndBlueCubeExplicit") is visualised.

A.2.3 Implicit referencing to COLLADA subdocuments

According to external referencing as specified in ISO/PAS 17506 or COLLADA 1.4.1 the geometric description of a scene may be divided into separate COLLADA documents. In such a case a main COLLADA document references elements which are stored in referenced COLLADA subdocuments. Subclause A.2.3 describes the implicit referencing to COLLADA elements, which are located in COLLADA subdocuments and referenced from a main COLLADA document. Figure A.26 depicts the structure and relations used in this example.

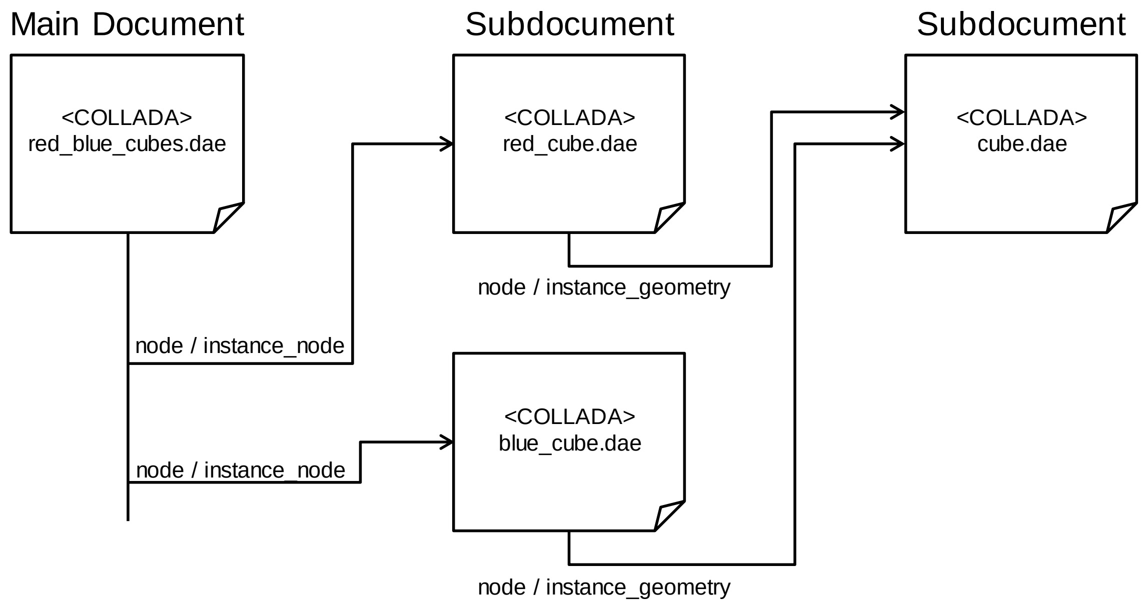

Figure A.26: Structure and relations of referenced COLLADA subdocuments

The following sample uses the visual scene taken from Subclause A.1.2 with a modified structure of COLLADA documents. The main COLLADA document (“red_blue_cubes.dae”) consists of two nodes, which each instantiate a node provided in a COLLADA subdocument (“red_cube.dae” and “blue_cube.dae”). The latter COLLADA subdocuments consist of a library, which defines a single node and a material. Each node of both documents instantiate a cube geometry, which is defined in aCOLLADA subdocument (“cube.dae”). While the definition of the geometry remains the same as used in A.1.2 the main COLLADA document (“red_blue_cubes.dae”) is modified and two additional COLLADA subdocuments are introduced (“red_cube.dae” and “blue_cube.dae”) in this example. Figure A.27 depicts the content of the modified main COLLADA document (“red_blue_cubes.dae”).

```
<COLLADA version="1.5.0">
  <asset>
    <contributor>
      <author>WriterVendor</author>
      <author_website>http://www.WriterVendor.com</author_website>
      <authoring_tool>"WriterName" "WriterVersion" "WriterRelease"</authoring_tool>
      <comments>This document defines the node hierarchy and instantiates nodes from subdocuments</comments>
    </contributor>
    <created>2014-04-14T11:37:18.1875000</created>
    <modified>2014-04-14T11:37:18.1875000</modified>
    <unit meter="1.0" name="meter" />
    <up_axis>Z_UP</up_axis>
  </asset>
  <library_visual_scenes>
```

```
    <visual_scene id="visualscene">
      <node id="redcube" name="RedCubeInstance">
        <instance_node sid="redcuberef" url="./red_cube.dae#redcube" />
      </node>
      <node id="bluecube" name="blueCubeInstance">
        <translate>0 0 1</translate>
        <instance_node sid="bluecuberef" url="./blue_cube.dae#bluecube" />
      </node>
    </visual_scene>
  </library_visual_scenes>
  <scene>
    <instance_visual_scene url="#visualscene" />
  </scene>
</COLLADA>
```

Figure A.27: Content of the modified COLLADA document red_blue_cubes.dae

The visual scene consists of two node elements (redcube and bluecube) which each instantiate a node in asubdocument ("red_cube.dae" and "blue_cube.dae"). The content of both COLLADA subdocuments are shown in Figure A.28 and Figure A.29.

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

```
    <material id="red" name="Red">
      <instance_effect url="#red-fx" />
    </material>
  </library_materials>
  <library_nodes>
    <node id="redcube" name="RedCube">
      <instance_geometry url="./cube.dae#cube-geom">
        <bind_material>
          <technique_common>
            <instance_material symbol="mat" target="#red" />
          </technique_common>
        </bind_material>
      </instance_geometry>
    </node>
  </library_nodes>
</COLLADA>
```

Figure A.28: Content of the COLLADA document red_cube.dae

The COLLADA document "red_cube.dae" defines a node named "RedCube" within a library which may be referenced and instantiated. Additionally, a red material is defined to attach to the instantiated geometry in the COLLADA subdocument "cube.dae". In a similar manner the node and geometry of the blue cube named "BlueCube" is defined in the COLLADA subdocument "blue_cube.dae" as depicted in Figure A.29.

```
<COLLADA version="1.5.0">
  <asset>
    <contributor>
      <author>WriterVendor</author>
      <author_website>http://www.WriterVendor.com</author_website>
      <authoring_tool>"WriterName" "WriterVersion" "WriterRelease"</authoring_tool>
      <comments>This document defines a node which instantiates a cube geometry from a COLLADA
subdocument and applies a blue material</comments>
    </contributor>
    <created>2014-04-14T11:37:18.1875000</created>
    <modified>2014-04-14T11:37:18.1875000</modified>
    <unit meter="1.0" name="meter" />
    <up_axis>Z_UP</up_axis>
  </asset>
  <library_effects>
    <effect id="blue-fx">
      <profile_COMMON>
        <technique sid="COMMON">
          <phong>
            <diffuse>
              <color>0 0 1 1 </color>
            </diffuse>
          </phong>
        </technique>
```

```
      </profile_COMMON>
    </effect>
  </library_effects>
  <library_materials>
    <material id="blue" name="Blue">
      <instance_effect url="#blue-fx" />
    </material>
  </library_materials>
  <library_nodes>
    <node id="bluecube" name="BlueCube">
      <instance_geometry url="./cube.dae#cube-geom">
        <bind_material>
          <technique_common>
            <instance_material symbol="mat" target="#blue" />
          </technique_common>
        </bind_material>
      </instance_geometry>
    </node>
  </library_nodes>
</COLLADA>
```

Figure A.29: Content of the COLLADA document blue_cube.dae

To reference the blue or red cube in the COLLADA subdocuments from an AML document the full path starting from the root COLLADA document shall be specified. The following example shows an extract of an AML document to demonstrate the implicit referencing to elements located in referenced COLLADA subdocuments.

The AML document consists of an InstanceHierarchy named “ExampleInstanceHierarchy” with three InternalElements as depicted in Figure A.30. All InternalElements include an ExternalInterface of type “COLLADAInterface”, which is part of the standard InterfaceClassLibrary “AutomationMLInterfaceClassLib”.

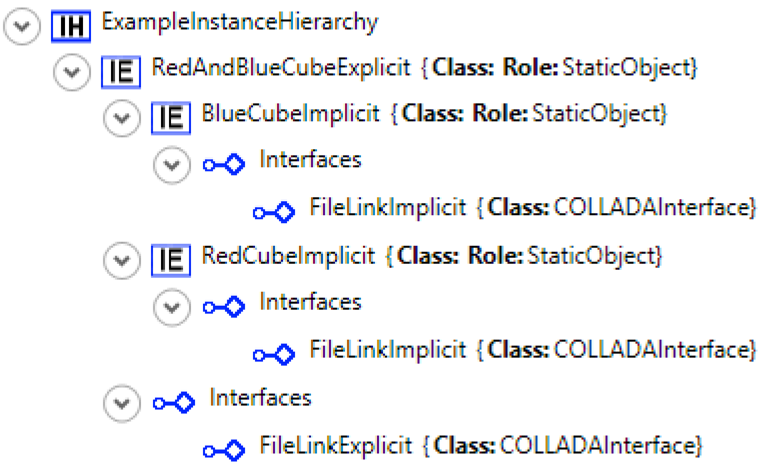

Figure A.30: Implicit Referencing: Hierarchy of the AML document

The ExternalInterface “FileLinkExplicit” of the InternalElement “RedAndBlueCubeExplicit” points to the COLLADA document and addresses the complete COLLADA scene explicitly (see A.1.5). Both the red and the blue cube are referenced unambiguously by the use of the relative URI “./red_blue_cubes.dae” without any additional fragment or target.The InternalElements

"RedCubeImplicit" and "BlueCubeImplicit" reference the red and blue cube implicitly, which both are located in a COLLADA subdocument. Figure A.31 shows the relevant part of the AML document for implicit referencing.

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

Figure A.31: XML representation of the AML document

A.2.4 Publishing elements of a COLLADA document in CAEX

Subclause A.2.4 shows a use case where both (explicit and implicit) referencing methodsare relevant.This Subclause describes how to publish frames of COLLADA documents and its referencing within an AML document.

Geometric and kinematic COLLADA elements can be published at CAEX level to make them available for usage. This makes it possible to reference them and assign further information. In this example the COLLADA document and its visual scene taken from A.1.2 (COLLADA document “red_blue_cubes.dae”) with an additional COLLADA element is used. The additional nodenamed “bluecubeframe” is attached to the blue cubeand represents a frame.

Figure A.32 shows the modified visual scene with the correspondingCOLLADAscene graph.

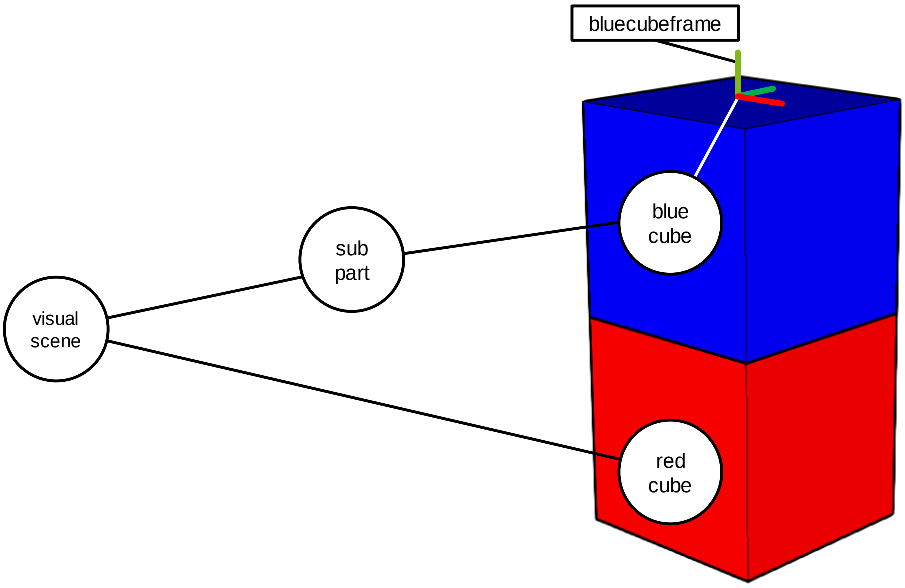

Figure A.32: Modified COLLADA scene with additional node

The relevant part of the COLLADA document, in particular the additional frame node, is shown in Figure A.33.

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

```
        <bind_material>
          <technique_common>
            <instance_material symbol="mat" target="#blue" />
          </technique_common>
        </bind_material>
      </instance_geometry>
    </node>
  </node>
</visual_scene>
```

Figure A.33: Additional frame element in COLLADA document

In this example the AML document consists of an InstanceHierarchy named “ExampleInstanceHierarchy”with an InternalElement “RedAndBlueCubeExplicit”, which includes an ExternalInterface “COLLADAInterface” named “FileLinkExplicit”. This ExternalInterface is used to reference the complete scene explicitly as depicted in Figure A.34.

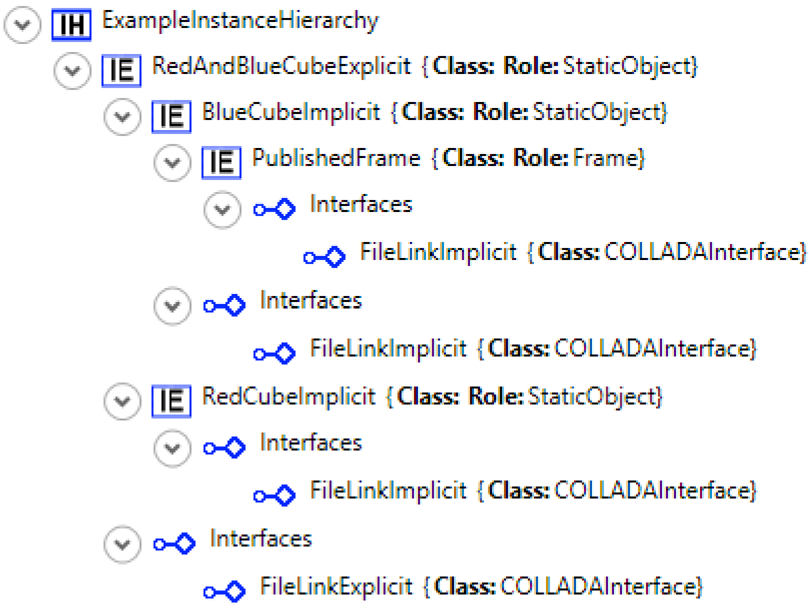

Figure A.34: Publishing frames: Hierarchy of the AML document

The InternalElement“RedAndBlueCubeExplicit” has two InternalElements named “BlueCubeImplicit” and “RedCubeImplicit”. The InternalElement“BlueCubeImplicit” has an InternalElement “PublishedFrame” with assigned RoleClass “Frame”. The ExternalElement “FileLinkImplicit” of the InternalElement “PublishedFrame” references the COLLADA node “bluecubeframe” in the COLLADA document node implicitly.

Figure A.35 shows the relevant part of the AML document and how to define attributes for implicit referencing.

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

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

```
    </InternalElement>
    <RoleRequirements RefBaseRoleClassPath="PATH_SO" />
  </InternalElement>
</InstanceHierarchy>
```

Figure A.35: XML representation of the AML document

NOTE 1 A published CAEX object with assigned RoleClass “Frame” does not have to point to a COLLADA geometry object (it may point to a COLLADA object without geometry). In former case the transformation information is taken from the COLLADA object and applied to the published CAEX object with assigned RoleClass “Frame”, even if the CAEX frame object provides a CAEX frame attribute. If the COLLADA object does not provide any transformation, the one from the frame attribute of the CAEX object with assigned RoleClass “Frame” is used.

A.3 Attachmentbetweenobjects in CAEX

Clause A.3 describes how to attach a geometric AML object to anothergeometric AML object (both of type InternalElement). Both AML objects have a reference to any COLLADA geometry objectsby using explicit references. Furthermore both AML objects have a child InternalElement Frame with assigned RoleClass“Frame”. Each InternalElement Frame has an implicit ExternalInterface derived from the InterfaceClass “COLLADAInterface”which references a node defined in the COLLADA document. Additionally each InternalElement Frame has ExternalInterface derived from the InterfaceClass “AttachmentInterface”, which may be connected or linked. The structure is shown in Figure A.36.

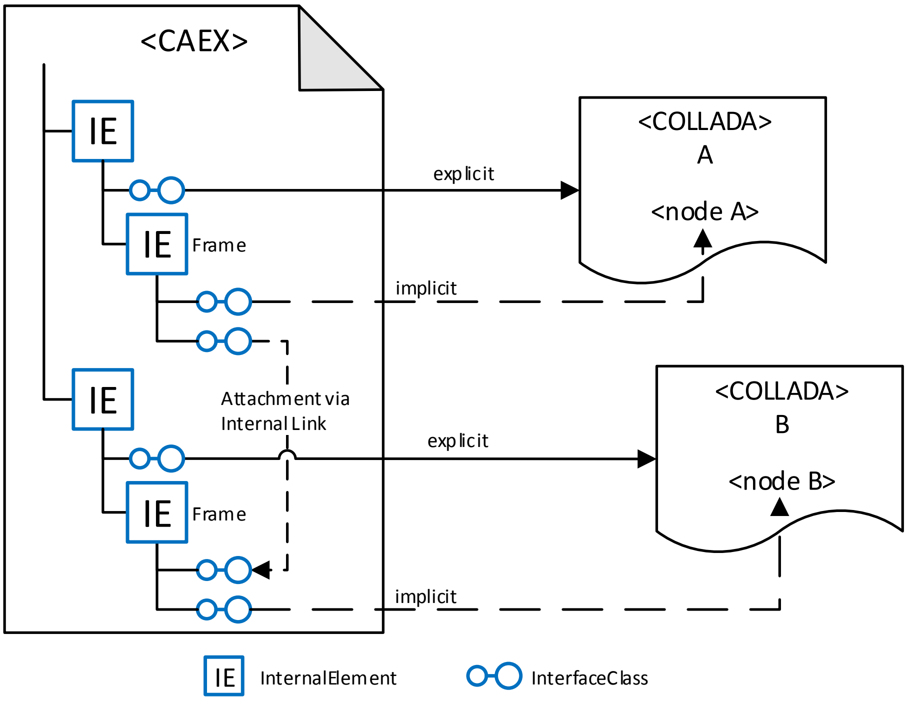

Figure A.36: Structure for attachments between objects in CAEX

In this example the COLLADA document and its visual scene taken from the SubclauseA.2.3 with an additional node is used. The additional node, which is attached to the blue cube, represents the published frame. An additional yellow cube is defined in a separate COLLADA document (“yellow_cube.dae”) with an additional node, which is published. The connection between both additional nodes (attachment frames) is established within the AML document. Figure A.37 shows the visualization of the yellow cube (“YellowCube”) and its additional frame.

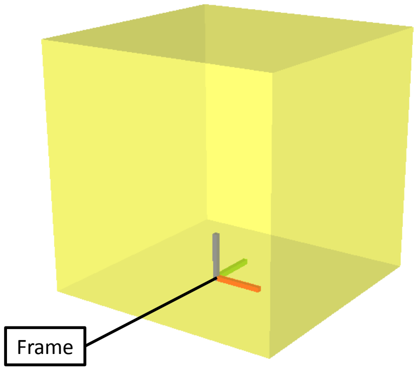

Figure A.37: Visualization of yellow cube with additional frame

The complete COLLADA document of the yellow cube("yellow_cube.dae") is shown in Figure A.38.It uses the geometry definition from the external COLLADA document "cube.dae" taken from Subclause A.1.2.

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

```
  <library_visual_scenes>
    <visual_scene id="visualscene">
      <node id="yellowcube" name="YellowCube">
        <translate>0 0 2</translate>
        <node sid="yellowcubeframe">
          <translate>0.5 0.5 0</translate>
        </node>
        <instance_geometry url="./cube.dae#cube-geom">
          <bind_material>
            <technique_common>
              <instance_material symbol="mat" target="#yellow" />
            </technique_common>
          </bind_material>
        </instance_geometry>
      </node>
    </visual_scene>
  </library_visual_scenes>
  <scene>
    <instance_visual_scene url="#visualscene" />
  </scene>
</COLLADA>
```

Figure A.38: COLLADA document of yellow cube with additional frame

The AML document consists of an InstanceHierarchy named “ExampleInstanceHierarchy”with two InternalElements (“RedAndBlueCubeExplicit” and “YellowCubeExplicit”). In this example the yellow cube (“YellowCube”) from the separate COLLADA document is attached to the blue one (“BlueCube”). Both InternalElements include an ExternalInterface “COLLADAInterface” named “FileLinkExplicit”, which ispart of the standard InterfaceClassLibrary “AutomationMLInterfaceClassLib”.

Each ExternalInterface named “FileLinkExplicit“ ofboth InternalElements points to the separate COLLADA documents and addresses the complete COLLADA scene explicitly (see also A.1.5). In addition bothInternalElements have a child InternalElement (“PublishedFrame”) with applied RoleClass “Frame”, which is part of the standard RoleClassLibrary as well.Each InternalElement “PublishedFrame” has two ExternalInterfaces: an ExternalInterface derived from the InterfaceClass “COLLADAInterface” to reference a COLLADA node implicitly (“FileLinkImplicit”) and an ExternalInterface derived from the InterfaceClass “AttachmentInterface” to define an attachment (“Attachment”). The structure is depicted in Figure A.39.

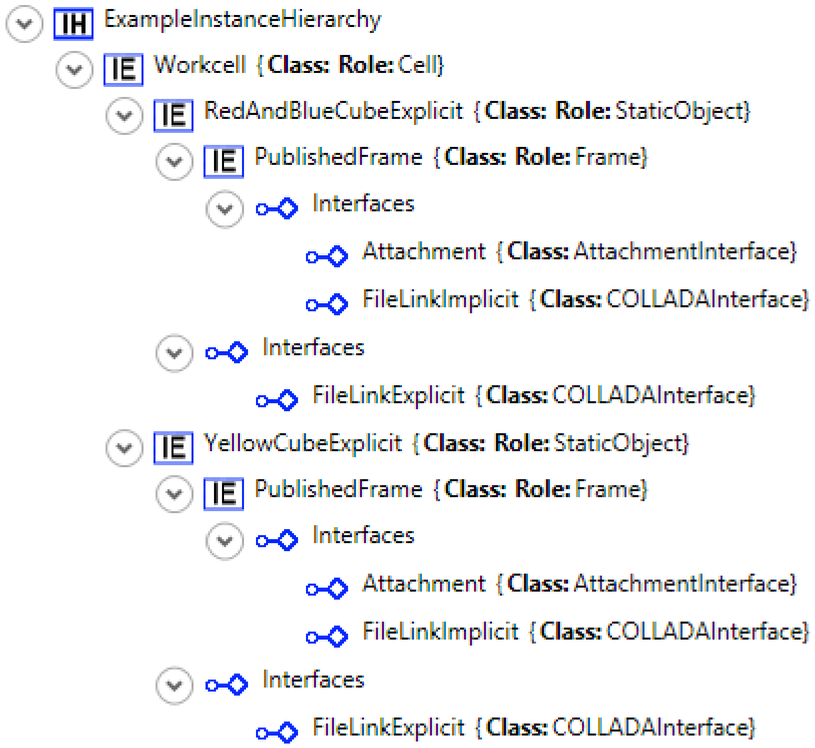

Figure A.39: Hierarchy of the AML document

Figure A.40 shows the relevant part of the AML document and how to publish frames for attaching objects, which are defined in AML.

```
<InstanceHierarchy Name="ExampleInstanceHierarchy">
  <InternalElement Name="Workcell" ID="GUID0">
    <InternalElement Name="RedAndBlueCubeExplicit" ID="GUID1">
      <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID2">
        <Attribute Name="refURI" AttributeDataType="xs:anyURI">
          <Value>./red_blue_cubes.dae</Value>
        </Attribute>
        <Attribute Name="refType" AttributeDataType="xs:string">
          <Value>explicit</Value>
        </Attribute>
      </ExternalInterface>
      <InternalElement Name="PublishedFrame" ID="GUID3">
        <ExternalInterface Name="Attachment" RefBaseClassPath="PATH_AI" ID="GUID4" />
        <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID5">
          <Attribute Name="refType" AttributeDataType="xs:string">
            <Value>implicit</Value>
          </Attribute>
          <Attribute Name="refURI" AttributeDataType="xs:anyURI">
            <Value>./red_blue_cubes.dae#subpart</Value>
          </Attribute>
          <Attribute Name="target" AttributeDataType="xs:token">
            <Value>./bluecube/bluecubeframe</Value>
          </Attribute>
```

```
        </ExternalInterface>
        <RoleRequirements RefBaseRoleClassPath="PATH_FRAME" />
      </InternalElement>
      <RoleRequirements RefBaseRoleClassPath="PATH_SO" />
    </InternalElement>
    <InternalElement Name="YellowCubeExplicit" ID="GUID6">
      <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID7">
        <Attribute Name="refType" AttributeDataType="xs:string">
          <Value>explicit</Value>
        </Attribute>
        <Attribute Name="refURI" AttributeDataType="xs:anyURI">
          <Value>./yellow_cube.dae</Value>
        </Attribute>
      </ExternalInterface>
      <InternalElement Name="PublishedFrame" ID="GUID8">
        <ExternalInterface Name="Attachment" RefBaseClassPath="PATH_AI" ID="GUID9" />
        <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID10">
          <Attribute Name="refType" AttributeDataType="xs:string">
            <Value>implicit</Value>
          </Attribute>
          <Attribute Name="refURI" AttributeDataType="xs:anyURI">
            <Value>./yellow_cube.dae#yellowcube</Value>
          </Attribute>
          <Attribute Name="target" AttributeDataType="xs:token">
            <Value>./yellowcubeframe</Value>
          </Attribute>
        </ExternalInterface>
        <RoleRequirements RefBaseRoleClassPath="PATH_FRAME" />
      </InternalElement>
      <RoleRequirements RefBaseRoleClassPath="PATH_SO" />
    </InternalElement>
    <InternalLink Name="AttachmentBlueMovesYellow" RefPartnerSideA="GUID3:Attachment" RefPartnerSide
B="GUID7:Attachment" />
    <RoleRequirements RefBaseRoleClassPath="PATH5/Cell" />
  </InternalElement>
</InstanceHierarchy>
```

Figure A.40: XML representation of the AML document

As a result the yellow cube is attached to the blue one as depicted in Figure A.41.In opposite to mounting the attaching does not include any additional transformation of the attached object. The connection is unidirectional. If the blue cube is translated or rotated, the transformation of the blue cubeaffects the yellow cube as well.

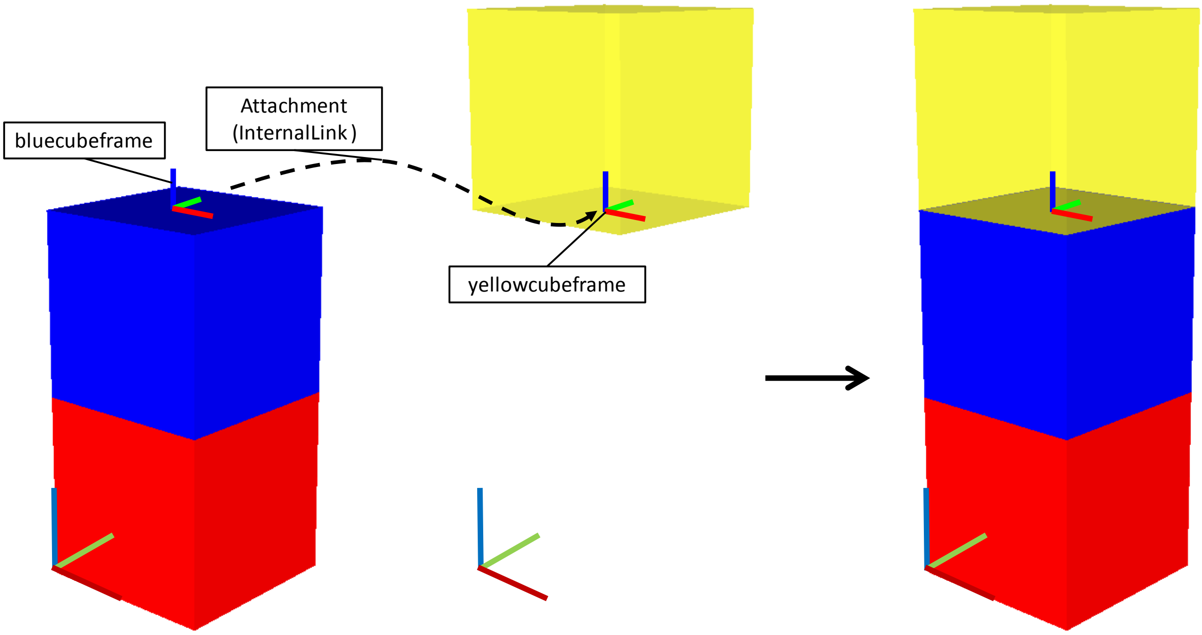

Figure A.41: Attachment betweengeometric AML objects

Finally, the connection is established by the use of an InternalLink, which connects both published frames. Figure A.42 shows the relevant part of the AML document and how to define an InternalLink. The link is stored at the CAEX InternalElement “Workcell“,which is the lowest commonparent of the connected CAEX objects. This is according to IEC62714-1.

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

Figure A.42: XML representation of the AML document

Annex B Modelling of kinematic systems and their combination in AML

B.1 General

The following clauses describe the modelling of kinematic systems and the combination in AML. In the clauses a scene will be composed stepwise containing a linear unit (Clause B.2) and a robot (Clause B.3) fixed to the linear unit (Clause B.4). In addition a gripper will be defined and mounted to the robot (Clause B.5). Finally a workpiece will be attached to the gripper (Clause B.6).

NOTE 1 For better readability all substitutions for RefBaseRoleClassPath attributes from A.1 apply.

NOTE 2 For better readability any RefBaseRoleClassPath attribute which points to the “AutomationMLMIRoleClassLib/ManufacturingEquipment/Robot” class is substituted with “PATH_ROB”.

NOTE 3 For better readability any RefBaseRoleClassPath attribute which points to the “AutomationMLMIRoleClassLib/ManufacturingEquipment/Carrier” class is substituted with “PATH_CAR”.

B.2 Modelling an AML document of a linear unit in CAEX and COLLADA

B.2.1 General

Clause B.2 describes the modelling of a linear unit. In B.2.2the visualscene is described. In the subsequent Subclauses B.2.3 – B.2.7the scene is complemented stepwise with kinematic elements. Finally, the AML document will be defined and combined with the kinematic system in B.2.8.

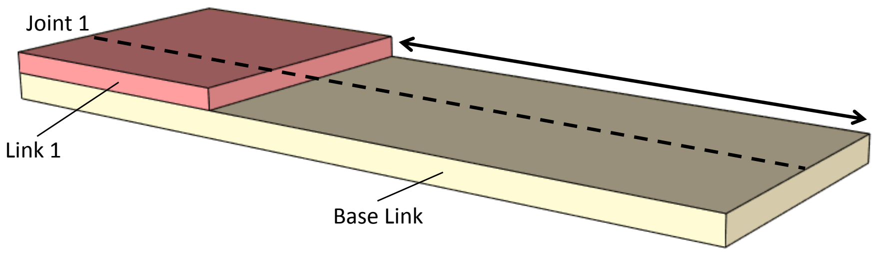

Figure B.1: Visualization of the linear unit

B.2.2 Definition of the visual scene

The visual scene of the linear unit in this example consists of two geometric elements, one on the top of the other as depicted in Figure B.1.To keep the COLLADA contentto a minimum, the geometries of the linear unit are represented by simple rectangular cuboids, which are defined within the COLLADA document "cube.dae" as introduced in Figure A.7. Each cube has a different scaling and provides a custom material. The lower cube represents the base link of the linear unit while the upper cube represents the slider.

Figure B.2 depicts the content of the COLLADA document "linear_unit.dae" which defines the geometric representation of the unit.

```
<COLLADA version="1.5.0">
  <asset>
    <contributor>
      <author>WriterVendor</author>
      <author_website>http://www.WriterVendor.com</author_website>
      <authoring_tool>"WriterName" "WriterVersion" "WriterRelease"</authoring_tool>
      <comments>This document defines the geometry and kinematic of a linear unit</comments>
    </contributor>
    <created>2014-04-14T11:37:18.1875000</created>
    <modified>2014-04-14T11:37:18.1875000</modified>
    <unit meter="1.0" name="meter" />
    <up_axis>Z_UP</up_axis>
  </asset>
  <library_effects>
    <effect id="base-fx">
      <profile_COMMON>
        <technique sid="COMMON">
          <phong>
            <diffuse>
              <color>1 0.9 0.8 1</color>
            </diffuse>
          </phong>
        </technique>
```

```
    </profile_COMMON>
  </effect>
  <effect id="link-fx">
    <profile_COMMON>
      <technique sid="COMMON">
        <phong>
          <diffuse>
            <color>1 0.6 0.6 1</color>
          </diffuse>
        </phong>
      </technique>
    </profile_COMMON>
  </effect>
</library_effects>
<library_materials>
  <material id="base-mat" name="BaseMat">
    <instance_effect url="#base-fx" />
  </material>
  <material id="link-mat" name="LinkMat">
    <instance_effect url="#link-fx" />
  </material>
</library_materials>
<library_visual_scenes>
  <visual_scene id="visualscene">
    <node id="linearunit" name="LinearUnit">
      <node id="geombase" name="GeomBase">
        <scale>3.0 1.0 0.1</scale>
        <instance_geometry url="./cube.dae#cube-geom">
          <bind_material>
            <technique_common>
              <instance_material symbol="mat" target="#base-mat" />
            </technique_common>
          </bind_material>
        </instance_geometry>
        <node id="geomlink1" name="GeomLink1">
          <translate>0 0 1</translate>
          <scale>0.33 1 0.8</scale>
          <rotate sid="transform">0 0 0 0</rotate>
          <instance_geometry url="./cube.dae#cube-geom">
            <bind_material>
              <technique_common>
                <instance_material symbol="mat" target="#link-mat" />
              </technique_common>
            </bind_material>
          </instance_geometry>
        </node>
```

```
        </node>
      </node>
    </visual_scene>
  </library_visual_scenes>
   <!-- Step 1: Definition of joint elements -->
   <!-- Step 2: Definition of kinematic model element -->
   <!-- Step 3: Definition of articulated system (kinematics) element -->
   <!-- Step 4: Definition of kinematics scene element -->
  <scene>
    <instance_visual_scene url="#visualscene" />
     <!-- Step 5: Instantiation of kinematics scene -->
  </scene>
</COLLADA>
```

Figure B.2: Definition of the visual scene

Each cuboid is assigned to a node (geombase and geomlink1). In the subclauses B.3.2 - B.3.7 the required kinematic elements will be defined and applied to this scene to complete the kinematic system.

B.2.3 Definition of the joint

In this case the kinematic system has a single prismatic jointnamed “Joint1”, which is defined in a library and which will be instantiated in a subsequent step.Prismatic joints are defined by the element “prismatic”. Additionally, the direction of the joint’s axis and the optional minimum and maximum limit values are specified according to the rules as specified in ISO/PAS 17506 or COLLADA 1.4.1. The extract of the COLLADA document is shown in Figure B.3.

NOTE 1 Physical units are definable according to ISO/PAS 17506 or COLLADA 1.4.1. By default degrees (angles) and meters (distances) shall are useable. Latter one is redefineable within the “unit” element.

```
<library_joints>
  <joint id="joint1" name="Joint1">
    <prismatic sid="axis0">
      <axis>0 0 1</axis>
      <limits>
        <min>0</min>
        <max>3</max>
      </limits>
    </prismatic>
  </joint>
</library_joints>
```

Figure B.3: Definition of the joint

B.2.4 Definition of the kinematic model

In subclause B.2.4 the kinematic model and its hierarchy is defined.First, for the kinematic model all joints are defined or instantiated from a joints library. The following kinematic chain consists of links and a joint and its hierarchical relationship. Any link or joint may provide a transformation. The transformation of a link or a joint is always relative to its predecessor and is described through the elements “translate” and “rotate”. The kinematic model is defined within a library and will be instantiated in a subsequent step. Figure B.4 shows the relevant part of the COLLADA document.

```
<library_kinematics_models>
  <kinematics_model id="linearunit_kinmodel">
    <technique_common>
```

```
      <instance_joint sid="inst_joint1" url="#joint1" />
      <link name="BaseLink" sid="baselink">
        <attachment_full joint="linearunit_kinmodel/inst_joint1">
          <rotate>0 1 0 90</rotate>
          <link name="Link1" sid="link1" />
        </attachment_full>
      </link>
    </technique_common>
  </kinematics_model>
</library_kinematics_models>
```

Figure B.4: Definition of kinematic model

In this example two links named “BaseLink” and “Link1” are defined, which are connected through a single joint, which references a joint defined in the library as described in the previous subclause.

B.2.5 Definition of the articulatedsystem

Kinematic and dynamic properties are assembled within “articulated_system” elements. Since each articulated system definition only contains either kinematic or dynamic information, two separate “articulated_system” systems are defined within a library.

```
<library_articulated_systems>
  <articulated_system id="linearunit_kinematics">
    <kinematics>
       <!-- ... ARTICULATED SYSTEM KINEMATICS ... -->
    </kinematics>
  </articulated_system>
  <articulated_system id="linearunit_motion">
    <motion>
       <!-- ... ARTICULATED SYSTEM MOTION ... -->
    </motion>
  </articulated_system>
</library_articulated_systems>
```

Figure B.5: Definition of the articulated system library

Figure B.5 depicts the structure of the library definition. The comments in the two articulated system elements denote the missing definitions, which will be explained in subclauses B.2.6 and B.2.7.

The “kinematics” section of an articulated system contains information about the kinematic behaviour of an articulated model, for instance activity of joints, limits, or dependencies between joints. Figure B.6 shows the relevant part of the COLLADA document and its additional kinematic parameters used in this example.

```
<articulated_system id="linearunit_kinematics">
  <kinematics>
    <instance_kinematics_model sid="inst_linearunit_kinmodel" url="#linearunit_kinmodel">
      <newparam sid="linearunit_kinematics_inst_linearunit_kinmodel">
        <SIDREF>linearunit_kinematics/inst_linearunit_kinmodel</SIDREF>
      </newparam>
      <newparam sid="linearunit_kinematics_inst_linearunit_kinmodel_inst_joint1_axis0">
        <SIDREF>linearunit_kinematics/inst_linearunit_kinmodel/inst_joint1/axis0</SIDREF>
      </newparam>
      <newparam sid="linearunit_kinematics_inst_linearunit_kinmodel_inst_joint1_value">
```

```
        <float>0.0</float>
      </newparam>
    </instance_kinematics_model>
    <technique_common>
      <axis_info axis="linearunit_kinematics/inst_linearunit_kinmodel/inst_joint1/axis0" sid="inst_j
oint1_info">
        <active>
          <bool>true</bool>
        </active>
        <locked>
          <bool>false</bool>
        </locked>
        <limits>
          <min>
            <float>0.0</float>
          </min>
          <max>
            <float>3.0</float>
          </max>
        </limits>
      </axis_info>
      <frame_origin link="linearunit_kinematics/inst_linearunit_kinmodel/baselink" sid="baseframe"
/>
      <frame_tip link="linearunit_kinematics/inst_linearunit_kinmodel/link1" sid="flangeframe">
        <translate>1 2 3</translate>
      </frame_tip>
    </technique_common>
  </kinematics>
</articulated_system>
```

Figure B.6: Definition of the kinematic articulated system

To provide dynamic information an additional articulated system shall be defined. The motion section of an articulated system contains information about the dynamic behaviour of an articulated model, for instance velocities of joints, acceleration, or deceleration of joints. Figure B.7 depicts the relevant part of the COLLADA document and its additional dynamics parameters used in this example.

```
<articulated_system id="linearunit_motion">
  <motion>
    <instance_articulated_system sid="inst_linearunit_kinematics" url="#linearunit_kinematics">
      <newparam sid="linearunit_motion_linearunit_kinematics_inst_LinearUnit_kinematics_model">
        <SIDREF>linearunit_kinematics/linearunit_kinematics_inst_linearunit_kinmodel</SIDREF>
      </newparam>
      <newparam sid="linearunit_motion_linearunit_kinematics_inst_linearunit_kinmodel_inst_joint1_ax
is0">
        <SIDREF>linearunit_kinematics/linearunit_kinematics_inst_linearunit_kinmodel_inst_joint1_axi
s0</SIDREF>
      </newparam>
      <newparam sid="linearunit_motion_linearunit_kinematics_inst_linearunit_kinmodel_inst_joint1_va
lue">
        <SIDREF>linearunit_kinematics/linearunit_kinematics_inst_linearunit_kinmodel_inst_joint1_val
```

```
ue</SIDREF>
      </newparam>
    </instance_articulated_system>
    <technique_common>
      <axis_info axis="linearunit_kinematics/inst_joint1_info">
        <speed>
          <float>0.0</float>
        </speed>
        <acceleration>
          <float>0.0</float>
        </acceleration>
        <deceleration>
          <float>0.0</float>
        </deceleration>
        <jerk>
          <float>0.0</float>
        </jerk>
      </axis_info>
    </technique_common>
  </motion>
</articulated_system>
```

Figure B.7: Definition of the motion articulated system

B.2.6 Definition of the kinematic scene

A kinematic scene assembles different kinematic systems. All parameters that are needed for the whole scene are published within a kinematic scene. In this example the kinematic scene instantiates the articulated system, which is defined in B.2.5. Additionally, the kinematic model, the jointaxis, and its valuesare published. Figure B.8 shows the relevant part of the COLLADA document.

```
<library_kinematics_scenes>
  <kinematics_scene id="linearunit_kinscene">
    <instance_articulated_system sid="inst_linearunit_motion" url="#linearunit_motion">
      <bind symbol="linearunit_kinscene_linearunit_motion_linearunit_kinematics_inst_LinearUnit_kine
matics_model">
        <param ref="linearunit_motion/linearunit_motion_linearunit_kinematics_inst_LinearUnit_kinema
tics_model" />
      </bind>
      <bind symbol="linearunit_kinscene_linearunit_motion_linearunit_kinematics_inst_linearunit_kinm
odel_inst_joint1_axis0">
        <param ref="linearunit_motion/linearunit_motion_linearunit_kinematics_inst_linearunit_kinmod
el_inst_joint1_axis0" />
      </bind>
      <bind symbol="linearunit_kinscene_linearunit_motion_linearunit_kinematics_inst_linearunit_kinm
odel_inst_joint1_value">
        <param ref="linearunit_motion/linearunit_motion_linearunit_kinematics_inst_linearunit_kinmod
el_inst_joint1_value" />
      </bind>
    </instance_articulated_system>
  </kinematics_scene>
</library_kinematics_scenes>
```

Figure B.8: Definition of the kinematic scene

B.2.7 Assembling of the scene

In the element “scene”, all kinematics scenes are instantiated that are used. In addition the linkage to the visualscene is done here, as depicted in Figure B.9. This linkage defines the flow of the joint value from a kinematic calculation to the transformation in the geometry.

```
<scene>
  <instance_visual_scene url="#visualscene" />
  <instance_kinematics_scene sid="inst_linearunit_kinscene" url="#linearunit_kinscene">
    <bind_kinematics_model node="linearunit">
      <param>linearunit_kinscene_linearunit_motion_linearunit_kinematics_inst_LinearUnit_kinematics_
model</param>
    </bind_kinematics_model>
    <bind_joint_axis target="geomlink1/transform">
      <axis>
        <param>linearunit_kinscene_linearunit_motion_linearunit_kinematics_inst_linearunit_kinmodel_
inst_joint1_axis0</param>
      </axis>
      <value>
        <param>linearunit_kinscene_linearunit_motion_linearunit_kinematics_inst_linearunit_kinmodel_
inst_joint1_value</param>
      </value>
    </bind_joint_axis>
  </instance_kinematics_scene>
</scene>
```

Figure B.9: Instantiation of the kinematic scene

B.2.8 Combination of CAEX and COLLADA into AML

The CAEX document consists of an InstanceHierarchy named “ExampleInstanceHierarchy” with a single InternalElement “LinearUnit”. The latter one includes an ExternalInterface named “FileLinkExplicit” derived from the InterfaceClass “COLLADAInterface”, which is part of the standard InterfaceClassLibrary “AutomationMLInterfaceClassLib”. This interface is used to reference the complete scene explicitly and in particular the linear unit asdepicted in Figure B.10.

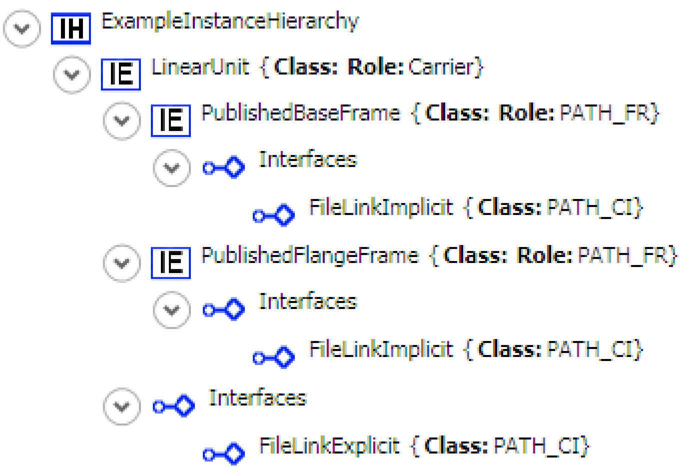

Figure B.10: Hierarchy of the AML document

The InternalElement “LinearUnit” has two additional InternalElements named “PublishedBaseFrame” and “PublishedFlangeFrame” with assigned RoleClass “Frame”. Both InternalElements reference kinematic elements in the COLLADA document implicitly. Figure B.11 shows the relevant part of the AML document.

```xml
<InstanceHierarchy Name="ExampleInstanceHierarchy">
  <InternalElement Name="LinearUnit" ID="GUID1">
    <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID2">
      <Attribute Name="refType" AttributeDataType="xs:string">
        <Value>explicit</Value>
      </Attribute>
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>./linear_unit.dae</Value>
      </Attribute>
    </ExternalInterface>
    <InternalElement Name="PublishedBaseFrame" ID="GUID3">
      <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID4">
        <Attribute Name="refType" AttributeDataType="xs:string">
          <Value>implicit</Value>
        </Attribute>
        <Attribute Name="refURI" AttributeDataType="xs:anyURI">
          <Value>./linear_unit.dae#linearunit_kinematics</Value>
        </Attribute>
        <Attribute Name="target" AttributeDataType="xs:token">
          <Value>./baseframe</Value>
        </Attribute>
      </ExternalInterface>
      <RoleRequirements RefBaseRoleClassPath="PATH_FR" />
    </InternalElement>
    <InternalElement Name="PublishedFlangeFrame" ID="GUID5">
      <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID6">
        <Attribute Name="refType" AttributeDataType="xs:string">
          <Value>implicit</Value>
        </Attribute>
        <Attribute Name="refURI" AttributeDataType="xs:anyURI">
          <Value>./linear_unit.dae#linearunit_kinematics</Value>
        </Attribute>
        <Attribute Name="target" AttributeDataType="xs:token">
          <Value>./flangeframe</Value>
        </Attribute>
      </ExternalInterface>
      <RoleRequirements RefBaseRoleClassPath="PATH_FR" />
    </InternalElement>
    <RoleRequirements RefBaseRoleClassPath="PATH_CAR" />
  </InternalElement>
</InstanceHierarchy>
```

Figure B.11: XML representation of the AML document

Each published frame references unambiguously the “articulated_system” element within the COLLADA document by the use of the relative URI “./linear_unit.dae#linearunit_kinematics”. The target elements “frame_tip” and “frame_origin”(see Figure B.6) are resolved by browsing from the entry point by the use of a relative path “./flangeframe” and “./baseframe” according to the rules of a SID.

B.3 Modelling an AML document of a robot in CAEX and COLLADA

B.3.1 General

Clause B.3 describes the modelling of a Selective Compliance Assembly Robot Arm(SCARA), which is a specialized type of an industrial robot. In subclause B.3.2 the visualscene is described. In the subsequent subclauses the scene is complemented stepwise with kinematic elements. Finally the AML document will be defined and combined with the kinematic system.

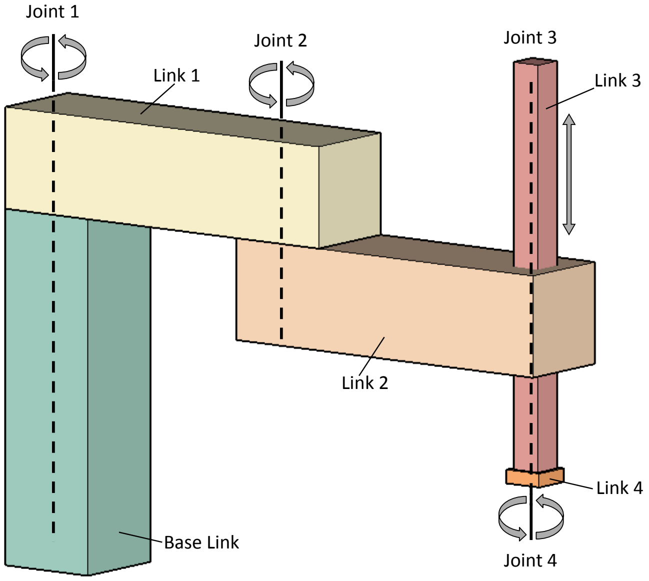

Figure B.12: Visualization of the robot

B.3.2 Definition of the visual scene

The visual scene of the SCARA robot in this example consists of fivegeometric elements as depicted in Figure B.12. Each geometric element represents a link of the robot. To keep the COLLADA content to a minimum, each link is represented by a rectangular cuboid, which is defined within the COLLADA document “cube.dae” as introduced Figure A.7.To obtain the needed shape each cube has a different scaling. The first cube represents the base link of the SCARA robot while the remaining cubes represent the subsequent links.

Figure B.13 depicts the content of the COLLADA document “robot.dae” which defines the geometric representation of the robot.

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

```
  <up_axis>Z_UP</up_axis>
</asset>
<library_effects>
  <effect id="base-fx">
    <profile_COMMON>
      <technique sid="COMMON">
        <phong>
          <diffuse>
            <color>0.60 0.77 0.72 1</color>
          </diffuse>
        </phong>
      </technique>
    </profile_COMMON>
  </effect>
  <effect id="link1-fx">
    <profile_COMMON>
      <technique sid="COMMON">
        <phong>
          <diffuse>
            <color>0.94 0.91 0.77 1</color>
          </diffuse>
        </phong>
      </technique>
    </profile_COMMON>
  </effect>
  <effect id="link2-fx">
    <profile_COMMON>
      <technique sid="COMMON">
        <phong>
          <diffuse>
            <color>0.93 0.80 0.69 1</color>
          </diffuse>
        </phong>
      </technique>
    </profile_COMMON>
  </effect>
  <effect id="link3-fx">
    <profile_COMMON>
      <technique sid="COMMON">
        <phong>
          <diffuse>
            <color>0.85 0.59 0.57 1</color>
          </diffuse>
        </phong>
      </technique>
    </profile_COMMON>
```

```
  </effect>
  <effect id="link4-fx">
    <profile_COMMON>
      <technique sid="COMMON">
        <phong>
          <diffuse>
            <color>0.95 0.64 0.41 1</color>
          </diffuse>
        </phong>
      </technique>
    </profile_COMMON>
  </effect>
</library_effects>
<library_materials>
  <material id="base-mat" name="BaseMat">
    <instance_effect url="#base-fx" />
  </material>
  <material id="link1-mat" name="Link1Mat">
    <instance_effect url="#link1-fx" />
  </material>
  <material id="link2-mat" name="Link2Mat">
    <instance_effect url="#link2-fx" />
  </material>
  <material id="link3-mat" name="Link3Mat">
    <instance_effect url="#link3-fx" />
  </material>
  <material id="link4-mat" name="Link4Mat">
    <instance_effect url="#link4-fx" />
  </material>
</library_materials>
<library_visual_scenes>
  <visual_scene id="visualscene">
    <node id="robot" name="Robot">
      <scale>2 2 2</scale>
      <node id="geombase" name="GeomBase">
        <translate>-0.05 -0.05 0</translate>
        <scale>0.1 0.1 0.35</scale>
        <instance_geometry url="./cube.dae#cube-geom">
          <bind_material>
            <technique_common>
              <instance_material symbol="mat" target="#base-mat" />
            </technique_common>
          </bind_material>
        </instance_geometry>
        <node id="geomlink1" name="GeomLink1">
          <translate>0 0 1</translate>
```

```
          <scale>3.8 1 0.2857</scale>
          <instance_geometry url="./cube.dae#cube-geom">
            <bind_material>
              <technique_common>
                <instance_material symbol="mat" target="#link1-mat" />
              </technique_common>
            </bind_material>
          </instance_geometry>
          <node id="geomlink2" name="GeomLink2">
            <translate>0.736848 0 -1</translate>
            <scale>0.947376 1 1</scale>
            <instance_geometry url="./cube.dae#cube-geom">
              <bind_material>
                <technique_common>
                  <instance_material symbol="mat" target="#link2-mat" />
                </technique_common>
              </bind_material>
            </instance_geometry>
            <node id="geomlink3" name="GeomLink3">
              <translate>0.861 0.35 -1</translate>
              <scale>0.08333 0.3 4</scale>
              <instance_geometry url="./cube.dae#cube-geom">
                <bind_material>
                  <technique_common>
                    <instance_material symbol="mat" target="#link3-mat" />
                  </technique_common>
                </bind_material>
              </instance_geometry>
              <node id="geomlink4" name="GeomLink4">
                <translate>-0.1666 -0.166 -0.0333</translate>
                <scale>1.333 1.333 0.0333</scale>
                <instance_geometry url="./cube.dae#cube-geom">
                  <bind_material>
                    <technique_common>
                      <instance_material symbol="mat" target="#link4-mat" />
                    </technique_common>
                  </bind_material>
                </instance_geometry>
              </node>
            </node>
          </node>
        </node>
      </node>
    </node>
  </visual_scene>
</library_visual_scenes>
```

```
  <!-- Step 1: Definition of joint elements -->
  <!-- Step 2: Definition of kinematic model element -->
  <!-- Step 3: Definition of articulated system (kinematics) element -->
  <!-- Step 4: Definition of kinematics scene element -->
  <scene>
    <instance_visual_scene url="#visualscene" />
     <!-- Step 5: Instantiation of kinematics scene -->
  </scene>
</COLLADA>
```

Figure B.13: Definition of the visual scene

Each geometryis assigned to a node (geombase, geomlink1, geomlink2, geomlink3, geomlink4). The next subclauses define the required kinematic elements and apply them to this scene to complete the kinematic system.

B.3.3 Definition of joints

The robot used in this example consists of four joints (rotational, rotational, prismatic, rotational). All joints are defined in a library and instantiated in a subsequent step. Prismatic joints are defined by the element “prismatic” whereas rotational joints are defined by the element “revolute”. While in this example the direction of the joint axis is specified, the lower and upper joint limits are defined exemplarily at a different location. The extract of the COLLADA document is shown in Figure B.14.

```
<library_joints>
  <joint id="joint1" name="joint1">
    <revolute sid="axis0">
      <axis>0 0 1</axis>
    </revolute>
  </joint>
  <joint id="joint2" name="joint2">
    <revolute sid="axis0">
      <axis>0 0 1</axis>
    </revolute>
  </joint>
  <joint id="joint3" name="joint3">
    <prismatic sid="axis0">
      <axis>0 0 1</axis>
    </prismatic>
  </joint>
  <joint id="joint4" name="joint4">
    <revolute sid="axis0">
      <axis>0 0 1</axis>
    </revolute>
  </joint>
</library_joints>
```

Figure B.14: Definition of joints

B.3.4 Definition of the kinematic model

In subclause B.3.4 the kinematic model and its hierarchy is defined. For the kinematic model first all joints are instantiated from a joints library. The following kinematic chain consists of links and joints and its hierarchical relationship. Any link or joint may provide a transformation. The transformation of a link or a joint is always relative to its predecessor and is described through the elements “translate” and “rotate”. The kinematic model is defined within a library and will be instantiated in a subsequent step. Figure B.15 shows the relevant part of the COLLADA document.

```
<library_kinematics_models>
  <kinematics_model id="robot_kinmodel">
    <technique_common>
      <instance_joint sid="inst_joint1" url="#joint1" />
      <instance_joint sid="inst_joint2" url="#joint2" />
      <instance_joint sid="inst_joint3" url="#joint3" />
      <instance_joint sid="inst_joint4" url="#joint4" />
      <link name="BaseLink" sid="baselink">
        <attachment_full joint="robot_kinmodel/inst_joint1">
          <translate>0.0 0.0 0.3</translate>
          <link name="Link1" sid="link1">
            <attachment_full joint="robot_kinmodel/inst_joint2">
              <translate>0.28 0.0 0.0</translate>
              <link name="Link2" sid="link2">
                <attachment_full joint="robot_kinmodel/inst_joint3">
                  <translate>0.26 0.0 0.0</translate>
                  <link name="Link3" sid="link3">
                    <attachment_full joint="robot_kinmodel/inst_joint4">
                      <translate>0.0 0.0 -0.15</translate>
                      <link name="Link4" sid="link4" />
                    </attachment_full>
                  </link>
                </attachment_full>
              </link>
            </attachment_full>
          </link>
        </attachment_full>
      </link>
    </technique_common>
  </kinematics_model>
</library_kinematics_models>
```

Figure B.15: Definition of kinematic model

In this example five links (BaseLink, Link1, Link2, Link3, Link4) are defined which are connected through joints referencing joints defined in the library as described in the previous subclause.

B.3.5 Definition of the articulated system

Kinematic and dynamic properties are assembled within “articulated_system” elements. Since each articulated system definition only contains either kinematic or dynamic information, two separate “articulated_system” systems are defined within a library. Figure B.16 depicts the structure of the articulated system library definition. The comments in the two articulated system elements denote the missing definitions which will be explained in the subclauses B.3.6 and B.3.7.

```
<library_articulated_systems>
  <articulated_system id="robot_kinematics">
    <kinematics>
```

```
      <!-- ... ARTICULATED SYSTEM KINEMATICS ... -->
    </kinematics>
  </articulated_system>
  <articulated_system id="robot_motion">
    <motion>
      <!-- ... ARTICULATED SYSTEM MOTION ... -->
    </motion>
  </articulated_system>
</library_articulated_systems>
```

Figure B.16: Definition of the articulated system library

The kinematic section of an articulated system contains information about the kinematic behaviour of an articulated model, for instance activity of joints, limits, or dependencies between joints. Figure B.17 shows the relevant part of the COLLADA document and its additional kinematic parameters used in this example.

```
<articulated_system id="robot_kinematics">
  <kinematics>
    <instance_kinematics_model sid="inst_robot_kinmodel" url="#robot_kinmodel">
      <newparam sid="robot_kinematics_inst_robot_kinmodel">
        <SIDREF>robot_kinematics/inst_robot_kinmodel</SIDREF>
      </newparam>
      <newparam sid="robot_kinematics_inst_robot_kinmodel_inst_joint1_axis0">
        <SIDREF>robot_kinematics/inst_robot_kinmodel/inst_joint1/axis0</SIDREF>
      </newparam>
      <newparam sid="robot_kinematics_inst_robot_kinmodel_inst_joint1_value">
        <float>0.0</float>
      </newparam>
      <newparam sid="robot_kinematics_inst_robot_kinmodel_inst_joint2_axis0">
        <SIDREF>robot_kinematics/inst_robot_kinmodel/inst_joint2/axis0</SIDREF>
      </newparam>
      <newparam sid="robot_kinematics_inst_robot_kinmodel_inst_joint2_value">
        <float>0.0</float>
      </newparam>
      <newparam sid="robot_kinematics_inst_robot_kinmodel_inst_joint3_axis0">
        <SIDREF>robot_kinematics/inst_robot_kinmodel/inst_joint3/axis0</SIDREF>
      </newparam>
      <newparam sid="robot_kinematics_inst_robot_kinmodel_inst_joint3_value">
        <float>0.0</float>
      </newparam>
      <newparam sid="robot_kinematics_inst_robot_kinmodel_inst_joint4_axis0">
        <SIDREF>robot_kinematics/inst_robot_kinmodel/inst_joint4/axis0</SIDREF>
      </newparam>
      <newparam sid="robot_kinematics_inst_robot_kinmodel_inst_joint4_value">
        <float>0.0</float>
      </newparam>
    </instance_kinematics_model>
    <technique_common>
```

```
    <axis_info axis="robot_kinematics_model/inst_joint1/axis0" sid="inst_joint1_info">
      <active>
        <bool>true</bool>
      </active>
      <locked>
        <bool>false</bool>
      </locked>
      <limits>
        <min>
          <float>-180</float>
        </min>
        <max>
          <float>180</float>
        </max>
      </limits>
    </axis_info>
    <axis_info axis="robot_kinematics_model/inst_joint2/axis0" sid="inst_joint2_info">
      <active>
        <bool>true</bool>
      </active>
      <locked>
        <bool>false</bool>
      </locked>
      <limits>
        <min>
          <float>-140</float>
        </min>
        <max>
          <float>140</float>
        </max>
      </limits>
    </axis_info>
    <axis_info axis="robot_kinematics_model/inst_joint3/axis0" sid="inst_joint3_info">
      <active>
        <bool>true</bool>
      </active>
      <locked>
        <bool>false</bool>
      </locked>
      <limits>
        <min>
          <float>0.1</float>
        </min>
        <max>
          <float>-0.2</float>
        </max>
```

```
        </limits>
      </axis_info>
      <axis_info axis="robot_kinematics_model/inst_joint4/axis0" sid="inst_joint4_info">
        <active>
          <bool>true</bool>
        </active>
        <locked>
          <bool>false</bool>
        </locked>
        <limits>
          <min>
            <float>-360</float>
          </min>
          <max>
            <float>360</float>
          </max>
        </limits>
      </axis_info>
      <frame_origin link="robot_kinematics/inst_robot_kinmodel/baselink" sid="baseframe" />
      <frame_tip link="robot_kinematics/inst_robot_kinmodel/link4" sid="flangeframe">
        <translate>1 2 3</translate>
      </frame_tip>
    </technique_common>
  </kinematics>
</articulated_system>
```

Figure B.17: Definition of the kinematic articulated system

To provide dynamics information an additional articulated system shall be defined. The motion section of an articulated system contains information about the dynamics behaviour of an articulated model, for instance velocities of joints, acceleration, or deceleration of joints. Figure B.18 depicts the relevant part of the COLLADA document and its additional dynamics parameters used in this example.

```
<articulated_system id="robot_motion">
  <motion>
    <instance_articulated_system sid="inst_robot_kinematics" url="#robot_kinematics">
      <newparam sid="robot_motion_robot_kinematics_inst_robot_kinmodel">
        <SIDREF>robot_kinematics/robot_kinematics_inst_robot_kinmodel</SIDREF>
      </newparam>
      <newparam sid="robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint1_axis0">
        <SIDREF>robot_kinematics/robot_kinematics_inst_robot_kinmodel_inst_joint1_axis0</SIDREF>
      </newparam>
      <newparam sid="robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint1_value">
        <SIDREF>robot_kinematics/robot_kinematics_inst_robot_kinmodel_inst_joint1_value</SIDREF>
      </newparam>
      <newparam sid="robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint2_axis0">
        <SIDREF>robot_kinematics/robot_kinematics_inst_robot_kinmodel_inst_joint2_axis0</SIDREF>
      </newparam>
      <newparam sid="robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint2_value">
```

```
        <SIDREF>robot_kinematics/robot_kinematics_inst_robot_kinmodel_inst_joint2_value</SIDREF>
      </newparam>
      <newparam sid="robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint3_axis0">
        <SIDREF>robot_kinematics/robot_kinematics_inst_robot_kinmodel_inst_joint3_axis0</SIDREF>
      </newparam>
      <newparam sid="robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint3_value">
        <SIDREF>robot_kinematics/robot_kinematics_inst_robot_kinmodel_inst_joint3_value</SIDREF>
      </newparam>
      <newparam sid="robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint4_axis0">
        <SIDREF>robot_kinematics/robot_kinematics_inst_robot_kinmodel_inst_joint4_axis0</SIDREF>
      </newparam>
      <newparam sid="robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint4_value">
        <SIDREF>robot_kinematics/robot_kinematics_inst_robot_kinmodel_inst_joint4_value</SIDREF>
      </newparam>
    </instance_articulated_system>
    <technique_common>
      <axis_info axis="robot_kinematics/inst_joint1_info">
        <speed>
          <float>0.0</float>
        </speed>
        <acceleration>
          <float>0.0</float>
        </acceleration>
        <deceleration>
          <float>0.0</float>
        </deceleration>
        <jerk>
          <float>0.0</float>
        </jerk>
      </axis_info>
      <axis_info axis="robot_kinematics/inst_joint2_info">
        <speed>
          <float>0.0</float>
        </speed>
        <acceleration>
          <float>0.0</float>
        </acceleration>
        <deceleration>
          <float>0.0</float>
        </deceleration>
        <jerk>
          <float>0.0</float>
        </jerk>
      </axis_info>
      <axis_info axis="robot_kinematics/inst_joint3_info">
        <speed>
```

```
          <float>0.0</float>
        </speed>
        <acceleration>
          <float>0.0</float>
        </acceleration>
        <deceleration>
          <float>0.0</float>
        </deceleration>
        <jerk>
          <float>0.0</float>
        </jerk>
      </axis_info>
      <axis_info axis="robot_kinematics/inst_joint4_info">
        <speed>
          <float>0.0</float>
        </speed>
        <acceleration>
          <float>0.0</float>
        </acceleration>
        <deceleration>
          <float>0.0</float>
        </deceleration>
        <jerk>
          <float>0.0</float>
        </jerk>
      </axis_info>
    </technique_common>
  </motion>
</articulated_system>
```

Figure B.18: Definition of the motion articulated system

B.3.6 Definition of the kinematic scene

A kinematic scene assembles different kinematic systems. All parameters needed for the whole scene are published within a kinematic scene. In this example the kinematic scene instantiates the articulated system which has been defined in B.3.5. Additionally, the kinematic model, the joint axis, and its valuesare published. Figure B.19 shows the relevant part of the COLLADA document.

```
<library_kinematics_scenes>
  <kinematics_scene id="robot_kinscene">
    <instance_articulated_system sid="inst_robot_motion" url="#robot_motion">
      <bind symbol="robot_kinscene_robot_motion_robot_kinematics_inst_robot_kinmodel">
        <param ref="robot_motion/robot_motion_robot_kinematics_inst_robot_kinmodel" />
      </bind>
      <bind symbol="robot_kinscene_robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint1_axi
s0">
        <param ref="robot_motion/robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint1_axis0
" />
      </bind>
```

```
      <bind symbol="robot_kinscene_robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint1_val
ue">
        <param ref="robot_motion/robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint1_value
" />
      </bind>
      <bind symbol="robot_kinscene_robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint2_axi
s0">
        <param ref="robot_motion/robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint2_axis0
" />
      </bind>
      <bind symbol="robot_kinscene_robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint2_val
ue">
        <param ref="robot_motion/robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint2_value
" />
      </bind>
      <bind symbol="robot_kinscene_robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint3_axi
s0">
        <param ref="robot_motion/robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint3_axis0
" />
      </bind>
      <bind symbol="robot_kinscene_robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint3_val
ue">
        <param ref="robot_motion/robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint3_value
" />
      </bind>
      <bind symbol="robot_kinscene_robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint4_axi
s0">
        <param ref="robot_motion/robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint4_axis0
" />
      </bind>
      <bind symbol="robot_kinscene_robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint4_val
ue">
        <param ref="robot_motion/robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint4_value
" />
      </bind>
    </instance_articulated_system>
  </kinematics_scene>
</library_kinematics_scenes>
```

Figure B.19: Definition of the kinematic scene

B.3.7 Assembling of the scene

In the element “scene” all kinematic scenes are instantiated that are used. In addition the linking to the visualscene is done here, as depicted in Figure B.20. This linking defines the flow of the joint value from a kinematic calculation to the transformation in the geometry.

```
<scene>
  <instance_visual_scene url="#visualscene" />
  <instance_kinematics_scene sid="inst_robot_kinscene" url="#robot_kinscene">
    <bind_kinematics_model node="robot">
      <param>robot_kinscene_robot_motion_robot_kinematics_inst_robot_kinmodel</param>
    </bind_kinematics_model>
    <bind_joint_axis target="joint1/transform">
      <axis>
```

```
        <param>robot_kinscene_robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint1_axis0</p
aram>
      </axis>
      <value>
        <param>robot_kinscene_robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint1_value</p
aram>
      </value>
    </bind_joint_axis>
    <bind_joint_axis target="joint2/transform">
      <axis>
        <param>robot_kinscene_robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint2_axis0</p
aram>
      </axis>
      <value>
        <param>robot_kinscene_robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint2_value</p
aram>
      </value>
    </bind_joint_axis>
    <bind_joint_axis target="joint3/transform">
      <axis>
        <param>robot_kinscene_robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint3_axis0</p
aram>
      </axis>
      <value>
        <param>robot_kinscene_robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint3_value</p
aram>
      </value>
    </bind_joint_axis>
    <bind_joint_axis target="joint4/transform">
      <axis>
        <param>robot_kinscene_robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint4_axis0</p
aram>
      </axis>
      <value>
        <param>robot_kinscene_robot_motion_robot_kinematics_inst_robot_kinmodel_inst_joint4_value</p
aram>
      </value>
    </bind_joint_axis>
  </instance_kinematics_scene>
</scene>
```

Figure B.20: Instantiation of the kinematic scene

B.3.8 Combination of CAEX and COLLADA into AML

The AML document consists of an InstanceHierarchy named “ExampleInstanceHierarchy” with a single InternalElement “SCARA Robot”. The latter one includes an ExternalInterface named “FileLinkExplicit” derived from the InterfaceClass “COLLADAInterface”, which is part of the standard InterfaceClassLibrary “AutomationMLInterfaceClassLib”. This interface is used to reference the complete scene explicitly and in particular the robot as depicted in Figure B.21.

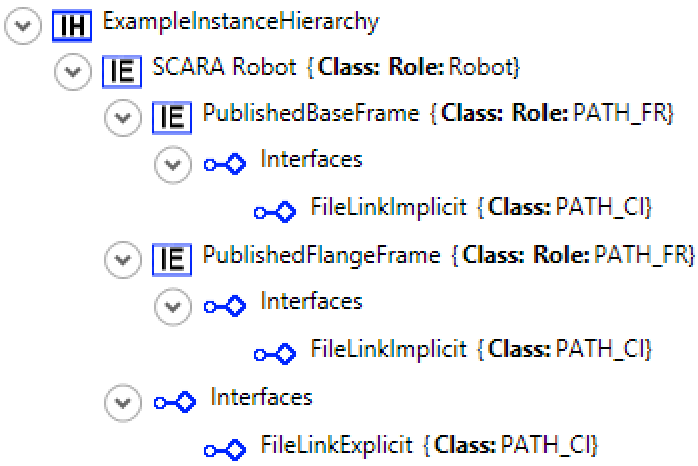

Figure B.21: Hierarchy of the AML document

The InternalElement “SCARA Robot” has two additional InternalElements named “PublishedBaseFrame” and “PublishedFlangeFrame” with assigned RoleClass “Frame”. Both InternalElements reference kinematic elements in the COLLADA document implicitly. Figure B.22 shows the relevant part of the AML document.

```
<InstanceHierarchy Name="ExampleInstanceHierarchy">
  <InternalElement Name="SCARA Robot" ID="GUID7">
    <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID8">
      <Attribute Name="refType" AttributeDataType="xs:string">
        <Value>explicit</Value>
      </Attribute>
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>./robot.dae</Value>
      </Attribute>
    </ExternalInterface>
    <InternalElement Name="PublishedBaseFrame" ID="GUID9">
      <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID10">
        <Attribute Name="refType" AttributeDataType="xs:string">
          <Value>implicit</Value>
        </Attribute>
        <Attribute Name="refURI" AttributeDataType="xs:anyURI">
          <Value>./robot.dae#robot_kinematics</Value>
        </Attribute>
        <Attribute Name="target" AttributeDataType="xs:token">
          <Value>./baseframe</Value>
        </Attribute>
      </ExternalInterface>
      <RoleRequirements RefBaseRoleClassPath="PATH_FR" />
    </InternalElement>
    <InternalElement Name="PublishedFlangeFrame" ID="GUID11">
      <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID12">
        <Attribute Name="refType" AttributeDataType="xs:string">
          <Value>implicit</Value>
```

```
        </Attribute>
        <Attribute Name="refURI" AttributeDataType="xs:anyURI">
          <Value>./robot.dae#robot_kinematics</Value>
        </Attribute>
        <Attribute Name="target" AttributeDataType="xs:token">
          <Value>./flangeframe</Value>
        </Attribute>
      </ExternalInterface>
      <RoleRequirements RefBaseRoleClassPath="PATH_FR" />
    </InternalElement>
    <RoleRequirements RefBaseRoleClassPath="PATH_ROB" />
  </InternalElement>
</InstanceHierarchy>
```

Figure B.22: XML representation of the AML document

Each published frame references unambiguously the “articulated_system” element within the COLLADA document by the use of the relative URI “./robot.dae#robot_kinematics”. The target elements “frame_tip” and “frame_origin” (see Figure B.22) are resolved by browsing from the entry point by the use of a relative path “./flangeframe” and “./baseframe” according to the rules of a SID.

B.4 Modelling an AML document of a combined system including a robot and a linear axis in CAEX and COLLADA

Clause B.4 demonstrates how to attach a robot to a linear unit. Therefore the definition of the robot is taken from clause B.3 and the definition of the linear unit is taken from clause B.2. In this example both definitions are extended by additional elements needed for the attachment which is realized in the top level AML document. In general, there are different ways how to specify the combined system as depicted in Figure B.23. Firstly, robot and linear unit may be defined in separate COLLADA documents and the offset transformation for the attachment may be defined in the AML document. Secondly, a combined main COLLADA document is used to reference both the robot and the linear unit. In the latter case the offset transformation for the attachment is defined in the main COLLADA document.The example of clause B.4 describes the first case.

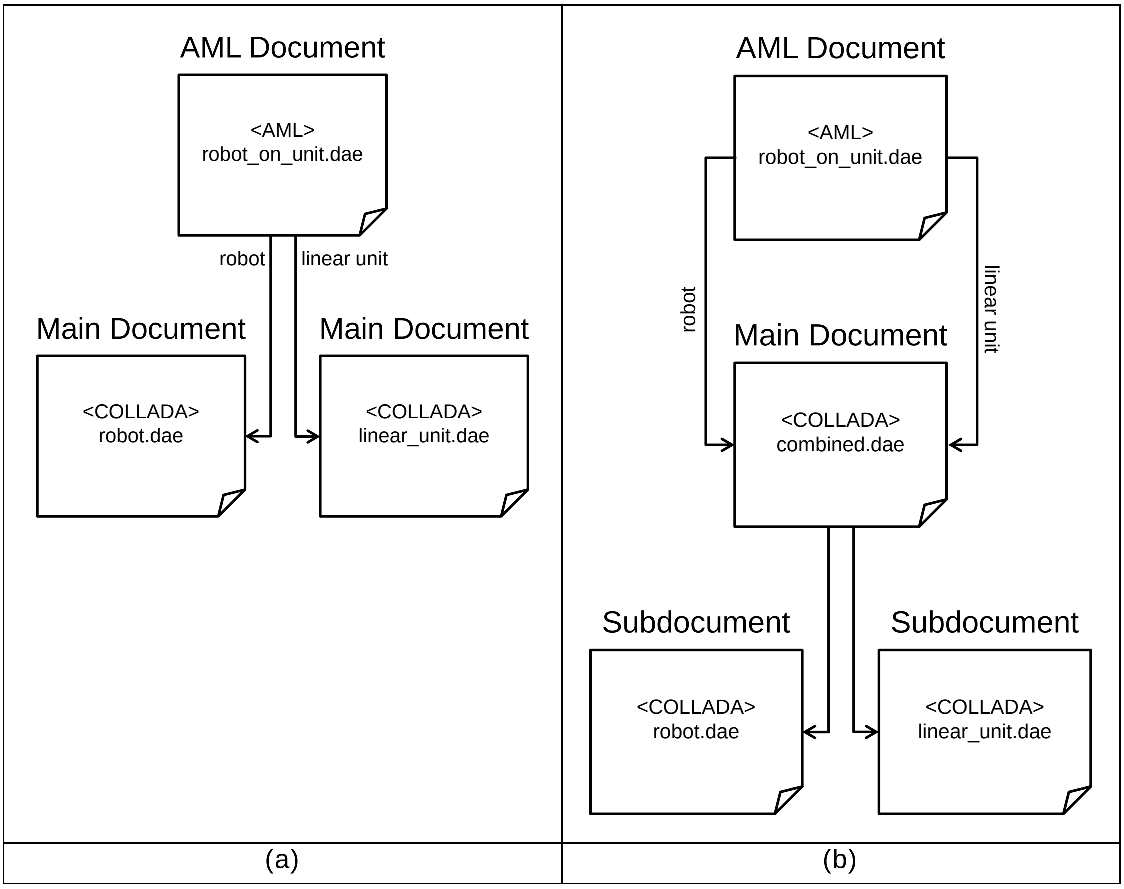

Figure B.23: Structure of referencing COLLADA subdocuments

The AML document consists of an InstanceHierarchy named “ExampleInstanceHierarchy” with a root InternalElement named “RobotOnLinearUnit”. This InternalElement has two InternalElements (“LinearUnit”, “SCARA Robot”). In this example the robot is attached to the linear unit. Both InternalElements include an ExternalInterface named “FileLinkExplicit” derived from the InterfaceClass “COLLADAInterface”, which is part of the standard InterfaceClassLibrary “AutomationMLInterfaceClassLib”. Each ExternalInterface “FileLinkExplicit” of both InternalElements points to a separate COLLADA document defined in clause B.2 and clause B.3 and addresses the complete COLLADA scene explicitly. In addition both InternalElements have a child element (“PublishedBaseFrame”, “PublishedFlangeFrame”) with applied RoleClass “Frame”, which is part of the standard RoleClassLibrary as well.The InternalElements “PublishedBaseFrame” and “PublishedFlangeFrame” have two ExternalInterfaces: an ExternalInterface derived from the InterfaceClass “COLLADAInterface” to reference a COLLADA node implicitly (“FileLinkImplicit”) and an ExternalInterface derived from the InterfaceClass “AttachmentInterface” to define an attachment (“Attachment”). The structure is depicted in Figure B.24.

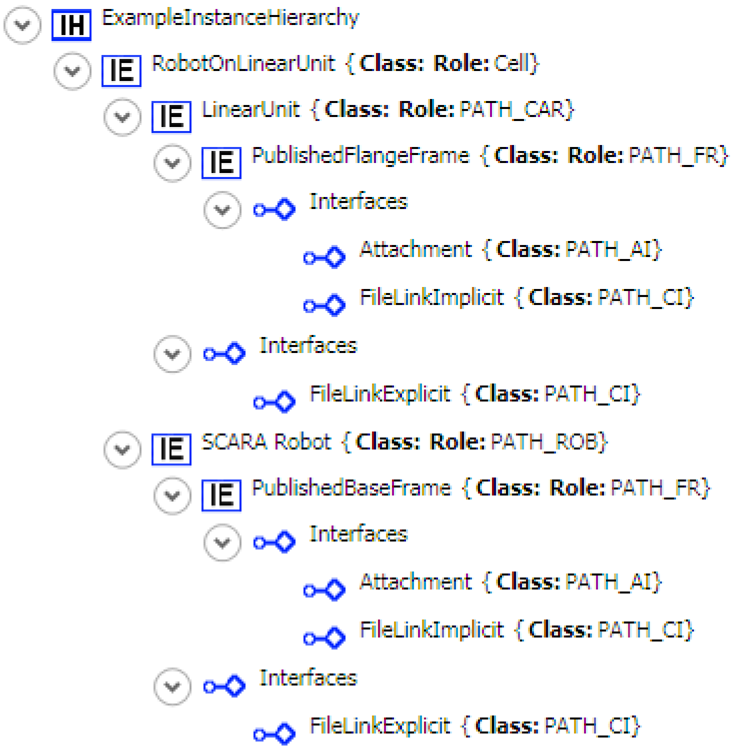

Figure B.24: Hierarchy of the AML document

Figure B.25 shows the relevant part of the AML document and how to publish frames for attaching the robot to the linear unit, which are defined in AML.

```
<InstanceHierarchy Name="ExampleInstanceHierarchy">
  <InternalElement Name="RobotOnLinearUnit" ID="GUID0">
    <InternalElement Name="LinearUnit" ID="GUID1">
      <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID2">
        <Attribute Name="refType" AttributeDataType="xs:string">
          <Value>explicit</Value>
        </Attribute>
        <Attribute Name="refURI" AttributeDataType="xs:anyURI">
          <Value>./linear_unit.dae</Value>
        </Attribute>
      </ExternalInterface>
      <InternalElement Name="PublishedFlangeFrame" ID="GUID3">
        <ExternalInterface Name="Attachment" RefBaseClassPath="PATH_AI" ID="GUID4" />
        <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID5">
          <Attribute Name="refType" AttributeDataType="xs:string">
            <Value>implicit</Value>
          </Attribute>
          <Attribute Name="refURI" AttributeDataType="xs:anyURI">
            <Value>./linear_unit.dae#linearunit_kinematics</Value>
          </Attribute>
          <Attribute Name="target" AttributeDataType="xs:token">
            <Value>./flangeframe</Value>
          </Attribute>
```

```xml
        </ExternalInterface>
        <RoleRequirements RefBaseRoleClassPath="PATH_FR" />
      </InternalElement>
      <RoleRequirements RefBaseRoleClassPath="PATH_CAR" />
    </InternalElement>
    <InternalElement Name="SCARA Robot" ID="GUID6">
      <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID7">
        <Attribute Name="refType" AttributeDataType="xs:string">
          <Value>explicit</Value>
        </Attribute>
        <Attribute Name="refURI" AttributeDataType="xs:anyURI">
          <Value>./robot.dae</Value>
        </Attribute>
      </ExternalInterface>
      <InternalElement Name="PublishedBaseFrame" ID="GUID8">
        <ExternalInterface Name="Attachment" RefBaseClassPath="PATH_AI" ID="GUID9" />
        <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID10">
          <Attribute Name="refType" AttributeDataType="xs:string">
            <Value>implicit</Value>
          </Attribute>
          <Attribute Name="refURI" AttributeDataType="xs:anyURI">
            <Value>./robot.dae#robot_kinematics</Value>
          </Attribute>
          <Attribute Name="target" AttributeDataType="xs:token">
            <Value>./baseframe</Value>
          </Attribute>
        </ExternalInterface>
        <RoleRequirements RefBaseRoleClassPath="PATH_FR" />
      </InternalElement>
      <RoleRequirements RefBaseRoleClassPath="PATH_ROB" />
    </InternalElement>
  </InternalElement>
</InstanceHierarchy>
```

Figure B.25: XML representation of the AML document

Finally, the connection is established by the use of an InternalLink, which connects both published frames. Figure B.26 shows the relevant part of the AML document and how to define an InternalLink. The link is stored in the InternalElement “RobotOnLinearUnit“, which is the lowest common parent of the connected AML objectsaccording to IEC62714-1.

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

Figure B.26: XML representation of the AML document

Figure B.27 depicts the resulting scene with the robot attached to the linear unit.

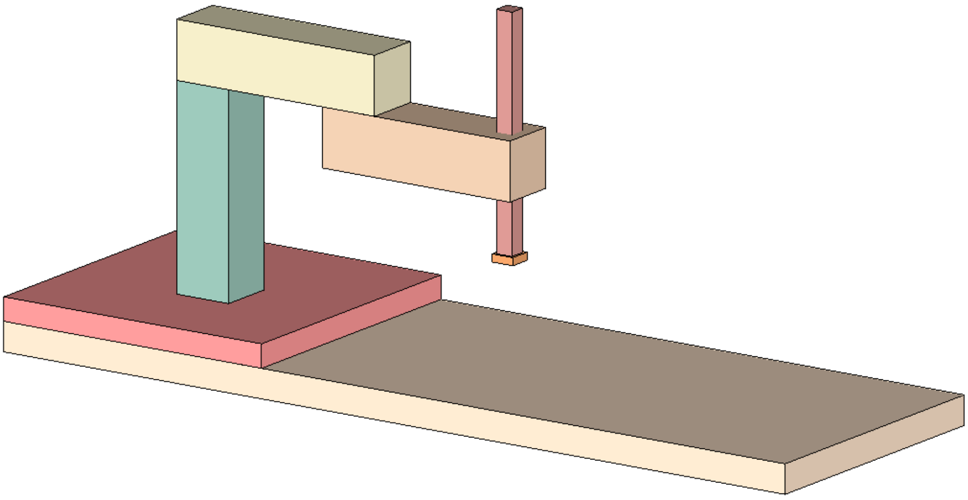

Figure B.27: Visualization of the robot attached to the linear unit

B.5 Modelling an AML document of a gripper connected to robot in CAEX and COLLADA

B.5.1 General

Clause B.5 shows how to model a gripping device which represents a simplified two jaw parallel gripper. To open or close the gripper by means of moving the fingers, a kinematic system will be applied to the gripper. To achieve a parallel movement of both fingers, the first finger can be moved actively and the second finger will be dependent on the first one. This dependency and the direction of the movement will be expressed using MathML.

In this example three COLLADA documents are used: a main document and two subdocuments. The geometry and kinematic definition is separated into two subdocuments. The main document defines the scene and combines the geometry and kinematic representation. Figure B.28 depicts the hierarchy of the documents.

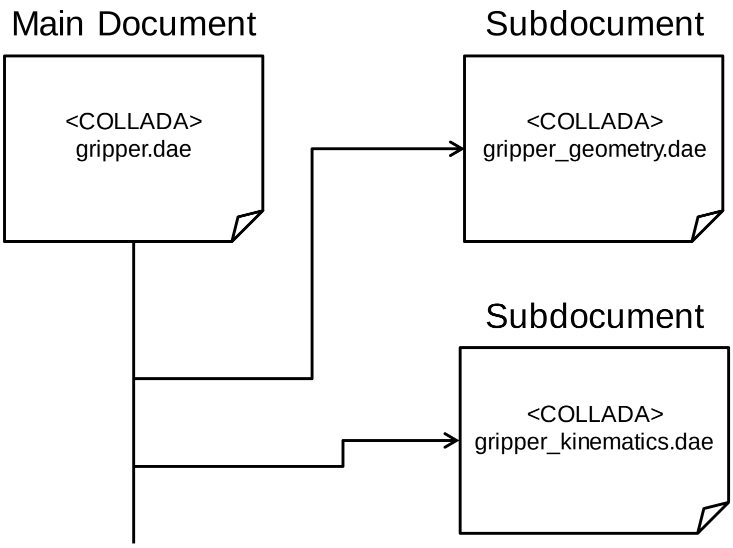

Figure B.28: Hierarchy of COLLADA documents

According to clauses B.2 and B.5.2, the first Subclause describes the visual scene, which is defined in the subdocument “gripper_geometry.dae”. In B.5.3 the kinematics scene is defined, which is part of the subdocument “gripper_kinematics”. The combination of both the geometry and kinematic and in particular the definition of the scene is defined in the main document “gripper.dae” and is subject of B.5.3.5. In B.5.5 the combination of CAEX and COLLADA into AML is shown.

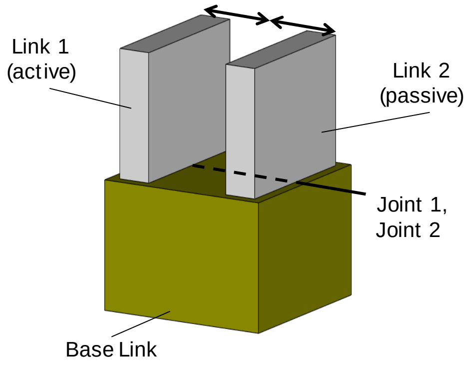

Figure B.29: Visualization of the gripper

B.5.2 Definition of the visual scene

The visual scene of the gripper consists of three geometric elements: a gripping module (“BaseLink”) and two fingers (“Link1”, “Link2”) as depicted in Figure B.29. To keep the COLLADA content to a minimum, the geometries of the gripper are represented by rectangular cuboids, which are defined within the COLLADA document “cube.dae” as introduced in Figure A.7. Figure B.30 depicts the content of the COLLADA document “gripper_geometry.dae” which defines the geometric representation of the gripper.

```
<COLLADA version="1.5.0">
  <asset>
    <contributor>
      <author>WriterVendor</author>
      <author_website>http://www.WriterVendor.com</author_website>
      <authoring_tool>"WriterName" "WriterVersion" "WriterRelease"</authoring_tool>
      <comments>This document defines the node hierarchy of a gripper with two jaws</comments>
    </contributor>
    <created>2014-04-14T11:37:18.1875000</created>
    <modified>2014-04-14T11:37:18.1875000</modified>
    <unit meter="1.0" name="meter" />
    <up_axis>Z_UP</up_axis>
  </asset>
  <library_effects>
    <effect id="base-fx">
      <profile_COMMON>
        <technique sid="COMMON">
          <phong>
            <diffuse>
              <color>0.5 0.5 0 1</color>
            </diffuse>
          </phong>
        </technique>
      </profile_COMMON>
```

```
    </effect>
    <effect id="link-fx">
      <profile_COMMON>
        <technique sid="COMMON">
          <phong>
            <diffuse>
              <color>0.75 0.75 0.75 1</color>
            </diffuse>
          </phong>
        </technique>
      </profile_COMMON>
    </effect>
  </library_effects>
  <library_materials>
    <material id="base-mat" name="BaseMat">
      <instance_effect url="#base-fx" />
    </material>
    <material id="link-mat" name="LinkMat">
      <instance_effect url="#link-fx" />
    </material>
  </library_materials>
  <library_visual_scenes>
    <visual_scene id="visualscene">
      <node id="gripper" name="Gripper">
        <node id="geombase" name="GeomBase">
           <!-- the base link of the gripper -->
          <translate>-0.02 -0.02 0</translate>
          <scale>0.04 0.04 0.03</scale>
          <instance_geometry url="./cube.dae#cube-geom">
            <bind_material>
              <technique_common>
                <instance_material symbol="mat" target="#base-mat" />
              </technique_common>
            </bind_material>
          </instance_geometry>
           <!-- the first (active) link of the gripper -->
          <node id="geomlink1" name="GeomLink1">
            <scale>25 25 33.3333</scale>
            <translate>0.0025 0.0025 0.03</translate>
            <scale>0.0075 0.035 0.03</scale>
            <instance_geometry url="./cube.dae#cube-geom">
              <bind_material>
                <technique_common>
                  <instance_material symbol="mat" target="#link-mat" />
                </technique_common>
              </bind_material>
```

```xml
          </instance_geometry>
        </node>
         <!-- the second (passive) link of the gripper -->
        <node id="geomlink2" name="GeomLink2">
          <scale>25 25 33.3333</scale>
          <translate>0.03 0.0025 0.03</translate>
          <scale>0.0075 0.035 0.03</scale>
          <instance_geometry url="./cube.dae#cube-geom">
            <bind_material>
              <technique_common>
                <instance_material symbol="mat" target="#link-mat" />
              </technique_common>
            </bind_material>
          </instance_geometry>
        </node>
      </node>
    </node>
  </visual_scene>
 </library_visual_scenes>
</COLLADA>
```

Figure B.30: Definition of the visual scene

B.5.3 Definition of the kinematic system

In subclause B.5.3 the kinematic system will be defined.According to clause B.2 or clause B.3 the steps include the definition of joints, the kinematic model, articulated system, and the kinematic scene. The exact procedure is not explained any more in detail as it is already explained in clauses B.2 and B.3. Figure B.31 shows the template of the COLLADA document “gripper.dae”. Missing elements are denoted by XML comments and will be defined in the subsequent subclauses.

```xml
<COLLADA version="1.5.0">
  <asset>
    <contributor>
      <author>WriterVendor</author>
      <author_website>http://www.WriterVendor.com</author_website>
      <authoring_tool>"WriterName" "WriterVersion" "WriterRelease"</authoring_tool>
      <comments>This document defines the kinematics of a gripper with two jaws</comments>
    </contributor>
    <created>2014-04-14T11:37:18.1875000</created>
    <modified>2014-04-14T11:37:18.1875000</modified>
    <unit meter="1.0" name="meter" />
    <up_axis>Z_UP</up_axis>
  </asset>
  <library_joints>
     <!-- Step 1: Definition of joint elements -->
  </library_joints>
  <library_kinematics_models>
     <!-- Step 2: Definition of kinematic model element -->
  </library_kinematics_models>
```

```
  <library_articulated_systems>
     <!-- Step 3: Definition of articulated system (kinematics) element -->
     <!-- Step 4: Definition of articulated system (motion) element -->
  </library_articulated_systems>
  <library_kinematics_scenes>
     <!-- Step 5: Definition of kinematics scene element -->
  </library_kinematics_scenes>
  <library_formulas>
     <!-- Step 6: Definition of formula elements (joint dependencies) -->
  </library_formulas>
</COLLADA>
```

Figure B.31: Definition of the kinematics

B.5.3.1 Definition of joints

In this example, the kinematic system has two prismatic joints(“Joint1”, “Joint2”) which are defined in a library and which will be instantiated in a subsequent step. “Joint1” will represent the active, leading joint while “Joint2” will represent the passive, dependent one. The extract of the COLLADA document is shown in Figure B.32.

```
<joint id="joint1" name="Joint1">
  <prismatic sid="axis0">
    <axis>0 0 1</axis>
    <limits>
      <min>0</min>
      <max>3</max>
    </limits>
 </prismatic>
</joint>
<joint id="joint2" name="Joint2">
  <prismatic sid="axis0">
    <axis>0 0 1</axis>
    <limits>
      <min>0</min>
      <max>3</max>
    </limits>
  </prismatic>
</joint>
```

Figure B.32: Definition of joints

B.5.3.2 Definition of the kinematic model

In subclause B.5.3.2 the kinematic model and its hierarchy is defined. Firstly all joints are defined or instantiated from a joints library. The following kinematic chain consists of links and joints and their hierarchical relationship. Any link or joint may provide a transformation. The transformation of a link or a joint is always relative to its predecessor and is described through the elements “translate” and “rotate”. The kinematic model is defined within a library and will be instantiated in a subsequent step. Figure B.33 shows the relevant part of the COLLADA document.

```
<kinematics_model id="gripper_kinmodel">
  <technique_common>
    <instance_joint sid="inst_joint1" url="#joint1" />
    <instance_joint sid="inst_joint2" url="#joint2" />
    <link name="BaseLink" sid="baselink">
      <attachment_full joint="gripper_kinmodel/inst_joint1">
        <rotate>0 1 0 90</rotate>
        <link name="Link1" sid="link1" />
      </attachment_full>
      <attachment_full joint="gripper_kinmodel/inst_joint2">
        <rotate>0 1 0 90</rotate>
        <link name="Link2" sid="link2" />
      </attachment_full>
    </link>
  </technique_common>
</kinematics_model>
```

Figure B.33: Definition of kinematic model

B.5.3.3 Definition of the articulated system

Kinematic and dynamic properties are assembled within “articulated_system” elements. Since each articulated system definition only contains either kinematic or dynamic information, two separate “articulated_system” systems are defined within a library.

The kinematic section of an articulated system contains information about the kinematic behaviour of an articulated model, for instance activity of joints, limits or dependencies between joints. Figure B.34 shows the relevant part of the COLLADA document and its additional kinematic parameters used in this example.

```
<articulated_system id="gripper_kinematics">
  <kinematics>
    <instance_kinematics_model sid="inst_gripper_kinmodel" url="#gripper_kinmodel">
      <newparam sid="gripper_kinematics_inst_gripper_kinmodel">
        <SIDREF>gripper_kinematics/inst_gripper_kinmodel</SIDREF>
      </newparam>
      <newparam sid="gripper_kinematics_inst_gripper_kinmodel_inst_joint1_axis0">
        <SIDREF>gripper_kinematics/inst_gripper_kinmodel/inst_joint1/axis0</SIDREF>
      </newparam>
      <newparam sid="gripper_kinematics_inst_gripper_kinmodel_inst_joint1_value">
        <float>0.0</float>
      </newparam>
      <newparam sid="gripper_kinematics_inst_gripper_kinmodel_inst_joint2_axis0">
        <SIDREF>gripper_kinematics/inst_gripper_kinmodel/inst_joint2/axis0</SIDREF>
      </newparam>
      <newparam sid="gripper_kinematics_inst_gripper_kinmodel_inst_joint2_value">
        <float>0.0</float>
      </newparam>
    </instance_kinematics_model>
    <technique_common>
      <axis_info axis="gripper_kinematics/inst_gripper_kinmodel/inst_joint1/axis0" sid="inst_joint1_
```

```
info">
        <active>
          <bool>true</bool>
        </active>
        <locked>
          <bool>false</bool>
        </locked>
        <limits>
          <min>
            <float>0.0</float>
          </min>
          <max>
            <float>3.0</float>
          </max>
        </limits>
      </axis_info>
      <axis_info axis="gripper_kinematics/inst_gripper_kinmodel/inst_joint2/axis0" sid="inst_joint2_
info">
        <active>
          <bool>false</bool>
        </active>
        <locked>
          <bool>false</bool>
        </locked>
        <limits>
          <min>
            <float>0.0</float>
          </min>
          <max>
            <float>3.0</float>
          </max>
        </limits>
      </axis_info>
      <frame_origin link="gripper_kinematics/inst_gripper_kinmodel/baselink" sid="baseframe" />
      <frame_tip link="gripper_kinematics/inst_gripper_kinmodel/link1" sid="flangeframe">
        <translate>1 2 3</translate>
      </frame_tip>
    </technique_common>
  </kinematics>
</articulated_system>
```

Figure B.34: Definition of the articulated system

To provide dynamics information, an additional articulated system shall be defined. The motion section of an articulated system contains information about the dynamics behaviour of an articulated model, for instance velocities of joints, acceleration or deceleration of joints. Figure B.35 depicts the relevant part of the COLLADA document and its additional dynamics parameters used in this example.

```
<articulated_system id="gripper_motion">
  <motion>
    <instance_articulated_system sid="inst_gripper_kinematics" url="#gripper_kinematics">
      <newparam sid="gripper_motion_gripper_kinematics_inst_gripper_kinematics_model">
        <SIDREF>gripper_kinematics/gripper_kinematics_inst_gripper_kinmodel</SIDREF>
      </newparam>
      <newparam sid="gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_joint1_axis0">
        <SIDREF>gripper_kinematics/gripper_kinematics_inst_gripper_kinmodel_inst_joint1_axis0</SIDREF>
      </newparam>
      <newparam sid="gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_joint1_value">
        <SIDREF>gripper_kinematics/gripper_kinematics_inst_gripper_kinmodel_inst_joint1_value</SIDREF>
      </newparam>
      <newparam sid="gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_joint2_axis0">
        <SIDREF>gripper_kinematics/gripper_kinematics_inst_gripper_kinmodel_inst_joint2_axis0</SIDREF>
      </newparam>
      <newparam sid="gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_joint2_value">
        <SIDREF>gripper_kinematics/gripper_kinematics_inst_gripper_kinmodel_inst_joint2_value</SIDREF>
      </newparam>
    </instance_articulated_system>
    <technique_common>
      <axis_info axis="gripper_kinematics/inst_joint1_info">
        <speed>
          <float>0.0</float>
        </speed>
        <acceleration>
          <float>0.0</float>
        </acceleration>
        <deceleration>
          <float>0.0</float>
        </deceleration>
        <jerk>
          <float>0.0</float>
        </jerk>
      </axis_info>
      <axis_info axis="gripper_kinematics/inst_joint2_info">
        <speed>
          <float>0.0</float>
        </speed>
        <acceleration>
          <float>0.0</float>
        </acceleration>
        <deceleration>
          <float>0.0</float>
```

```
        </deceleration>
        <jerk>
          <float>0.0</float>
        </jerk>
      </axis_info>
    </technique_common>
  </motion>
</articulated_system>
```

Figure B.35: Definition of the articulated system

B.5.3.4 Definition of the kinematic scene

A kinematic scene assembles different kinematic systems. All parameters needed for the whole scene are published within a kinematic scene. In this example the kinematic scene instantiates the articulated system which has been defined in B.5.3.3. Additionally, the kinematic model, the joint axis, and its values are published. Figure B.36 shows the relevant part of the COLLADA document.

```
<kinematics_scene id="gripper_kinscene">
  <instance_articulated_system sid="inst_gripper_motion" url="#gripper_motion">
    <bind symbol="gripper_kinscene_gripper_motion_gripper_kinematics_inst_gripper_kinematics_model">
      <param ref="gripper_motion/gripper_motion_gripper_kinematics_inst_gripper_kinematics_model" />
    </bind>
    <bind symbol="gripper_kinscene_gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_join
t1_axis0">
      <param ref="gripper_motion/gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_joint1
_axis0" />
    </bind>
    <bind symbol="gripper_kinscene_gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_join
t1_value">
      <param ref="gripper_motion/gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_joint1
_value" />
    </bind>
    <bind symbol="gripper_kinscene_gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_join
t2_axis0">
      <param ref="gripper_motion/gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_joint2
_axis0" />
    </bind>
    <bind symbol="gripper_kinscene_gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_join
t2_value">
      <param ref="gripper_motion/gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_joint2
_value" />
    </bind>
  </instance_articulated_system>
</kinematics_scene>
```

Figure B.36: Definition of the kinematic scene

B.5.3.5 Definition of the joint dependency using MathML

For opening and closing the gripper, both fingers need to move towards one another simultaneously. To move the passive joint “Joint2” when moving the active joint “Joint1” a dependency is required, that is modelled using a mathematical formula.Formula B.1 shows the according formula to define the value on “Joint2” depending on the value of “Joint1”.

$$Joint2(Joint1) = (-1) * Joint1 \quad \text{(B.1)}$$

The sign depends on the orientation of the joint defined in the kinematic model. Figure B.37 depicts the relevant MathML section to define a formula within the COLLADA document. First, the target is defined based on the SID “gripper_kinmodel/inst_joint2”. Using the SID of “Joint1” within the formula the value of the active joint “Joint1” is applied to the passive joint“Joint2” according to the mathematical expression.

```
<formula sid="front_dir.formula">
  <target>
    <param>gripper_kinmodel/inst_joint2</param>
  </target>
  <technique_common>
    <mathml:math>
      <mathml:apply>
        <mathml:times />
        <mathml:cn type="integer">-1</mathml:cn>
        <mathml:ci>gripper_kinmodel/inst_joint1</mathml:ci>
      </mathml:apply>
    </mathml:math>
  </technique_common>
</formula>
```

Figure B.37: Definition of the joint dependency using MathML

B.5.3.6 FinalCOLLADA document

Figure B.38 depicts the resulting COLLADA document “gripper_kinematics.dae”.

```
<COLLADA version="1.5.0">
  <asset>
    <contributor>
      <author>WriterVendor</author>
      <author_website>http://www.WriterVendor.com</author_website>
      <authoring_tool>"WriterName" "WriterVersion" "WriterRelease"</authoring_tool>
      <comments>This document defines the kinematics of a gripper with two jaws</comments>
    </contributor>
    <created>2014-04-14T11:37:18.1875000</created>
    <modified>2014-04-14T11:37:18.1875000</modified>
    <unit meter="1.0" name="meter" />
    <up_axis>Z_UP</up_axis>
  </asset>
  <library_joints>
     <!-- Step 1: Definition of joint elements -->
    <joint id="joint1" name="Joint1">
      <prismatic sid="axis0">
        <axis>0 0 1</axis>
        <limits>
          <min>0</min>
          <max>3</max>
        </limits>
      </prismatic>
    </joint>
```

```
    <joint id="joint2" name="Joint2">
      <prismatic sid="axis0">
        <axis>0 0 1</axis>
        <limits>
          <min>0</min>
          <max>3</max>
        </limits>
      </prismatic>
    </joint>
  </library_joints>
  <library_kinematics_models>
     <!-- Step 2: Definition of kinematic model element -->
    <kinematics_model id="gripper_kinmodel">
      <technique_common>
        <instance_joint sid="inst_joint1" url="#joint1" />
        <instance_joint sid="inst_joint2" url="#joint2" />
        <link name="BaseLink" sid="baselink">
          <attachment_full joint="gripper_kinmodel/inst_joint1">
            <rotate>0 1 0 90</rotate>
            <link name="Link1" sid="link1" />
          </attachment_full>
          <attachment_full joint="gripper_kinmodel/inst_joint2">
            <rotate>0 1 0 90</rotate>
            <link name="Link2" sid="link2" />
          </attachment_full>
        </link>
      </technique_common>
    </kinematics_model>
  </library_kinematics_models>
  <library_articulated_systems>
     <!-- Step 3: Definition of articulated system (kinematics) element -->
    <articulated_system id="gripper_kinematics">
      <kinematics>
        <instance_kinematics_model sid="inst_gripper_kinmodel" url="#gripper_kinmodel">
          <newparam sid="gripper_kinematics_inst_gripper_kinmodel">
            <SIDREF>gripper_kinematics/inst_gripper_kinmodel</SIDREF>
          </newparam>
          <newparam sid="gripper_kinematics_inst_gripper_kinmodel_inst_joint1_axis0">
            <SIDREF>gripper_kinematics/inst_gripper_kinmodel/inst_joint1/axis0</SIDREF>
          </newparam>
          <newparam sid="gripper_kinematics_inst_gripper_kinmodel_inst_joint1_value">
            <float>0.0</float>
          </newparam>
          <newparam sid="gripper_kinematics_inst_gripper_kinmodel_inst_joint2_axis0">
            <SIDREF>gripper_kinematics/inst_gripper_kinmodel/inst_joint2/axis0</SIDREF>
          </newparam>
```

```
          <newparam sid="gripper_kinematics_inst_gripper_kinmodel_inst_joint2_value">
            <float>0.0</float>
          </newparam>
        </instance_kinematics_model>
        <technique_common>
          <axis_info axis="gripper_kinematics/inst_gripper_kinmodel/inst_joint1/axis0" sid="inst_joi
nt1_info">
            <active>
              <bool>true</bool>
            </active>
            <locked>
              <bool>false</bool>
            </locked>
            <limits>
              <min>
                <float>0.0</float>
              </min>
              <max>
                <float>3.0</float>
              </max>
            </limits>
          </axis_info>
          <axis_info axis="gripper_kinematics/inst_gripper_kinmodel/inst_joint2/axis0" sid="inst_joi
nt2_info">
            <active>
              <bool>false</bool>
            </active>
            <locked>
              <bool>false</bool>
            </locked>
            <limits>
              <min>
                <float>0.0</float>
              </min>
              <max>
                <float>3.0</float>
              </max>
            </limits>
          </axis_info>
          <frame_origin link="gripper_kinematics/inst_gripper_kinmodel/baselink" sid="baseframe" />
          <frame_tip link="gripper_kinematics/inst_gripper_kinmodel/link1" sid="flangeframe">
            <translate>1 2 3</translate>
          </frame_tip>
        </technique_common>
      </kinematics>
    </articulated_system>
     <!-- Step 4: Definition of articulated system (motion) element -->
```

```
    <articulated_system id="gripper_motion">
      <motion>
        <instance_articulated_system sid="inst_gripper_kinematics" url="#gripper_kinematics">
          <newparam sid="gripper_motion_gripper_kinematics_inst_gripper_kinematics_model">
            <SIDREF>gripper_kinematics/gripper_kinematics_inst_gripper_kinmodel</SIDREF>
          </newparam>
          <newparam sid="gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_joint1_axis0">
            <SIDREF>gripper_kinematics/gripper_kinematics_inst_gripper_kinmodel_inst_joint1_axis0</S
IDREF>
          </newparam>
          <newparam sid="gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_joint1_value">
            <SIDREF>gripper_kinematics/gripper_kinematics_inst_gripper_kinmodel_inst_joint1_value</S
IDREF>
          </newparam>
          <newparam sid="gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_joint2_axis0">
            <SIDREF>gripper_kinematics/gripper_kinematics_inst_gripper_kinmodel_inst_joint2_axis0</S
IDREF>
          </newparam>
          <newparam sid="gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_joint2_value">
            <SIDREF>gripper_kinematics/gripper_kinematics_inst_gripper_kinmodel_inst_joint2_value</S
IDREF>
          </newparam>
        </instance_articulated_system>
        <technique_common>
          <axis_info axis="gripper_kinematics/inst_joint1_info">
            <speed>
              <float>0.0</float>
            </speed>
            <acceleration>
              <float>0.0</float>
            </acceleration>
            <deceleration>
              <float>0.0</float>
            </deceleration>
            <jerk>
              <float>0.0</float>
            </jerk>
          </axis_info>
          <axis_info axis="gripper_kinematics/inst_joint2_info">
            <speed>
              <float>0.0</float>
            </speed>
            <acceleration>
              <float>0.0</float>
            </acceleration>
            <deceleration>
              <float>0.0</float>
```

```
          </deceleration>
          <jerk>
            <float>0.0</float>
          </jerk>
        </axis_info>
      </technique_common>
    </motion>
  </articulated_system>
</library_articulated_systems>
<library_kinematics_scenes>
   <!-- Step 5: Definition of kinematics scene element -->
  <kinematics_scene id="gripper_kinscene">
    <instance_articulated_system sid="inst_gripper_motion" url="#gripper_motion">
      <bind symbol="gripper_kinscene_gripper_motion_gripper_kinematics_inst_gripper_kinematics_mod
el">
        <param ref="gripper_motion/gripper_motion_gripper_kinematics_inst_gripper_kinematics_model
" />
      </bind>
      <bind symbol="gripper_kinscene_gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_
joint1_axis0">
        <param ref="gripper_motion/gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_jo
int1_axis0" />
      </bind>
      <bind symbol="gripper_kinscene_gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_
joint1_value">
        <param ref="gripper_motion/gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_jo
int1_value" />
      </bind>
      <bind symbol="gripper_kinscene_gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_
joint2_axis0">
        <param ref="gripper_motion/gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_jo
int2_axis0" />
      </bind>
      <bind symbol="gripper_kinscene_gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_
joint2_value">
        <param ref="gripper_motion/gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_jo
int2_value" />
      </bind>
    </instance_articulated_system>
  </kinematics_scene>
</library_kinematics_scenes>
<library_formulas>
   <!-- Step 6: Definition of formula elements (joint dependencies) -->
  <formula sid="front_dir.formula">
    <target>
      <param>gripper_kinmodel/inst_joint2</param>
    </target>
    <technique_common>
      <mathml:math>
        <mathml:apply>
```

```
            <mathml:times />
            <mathml:cn type="integer">-1</mathml:cn>
            <mathml:ci>gripper_kinmodel/inst_joint1</mathml:ci>
          </mathml:apply>
        </mathml:math>
      </technique_common>
    </formula>
  </library_formulas>
</COLLADA>
```

Figure B.38: XML representation of the COLLADA document gripper_kinematics.dae

B.5.4 Assembling of the scene

In the element “scene”, all kinematic scenes are instantiated that are used. In addition the linking to the visualscene is done here, as depicted in Figure B.39. The scene in this example instantiates elements from the subdocuments “gripper_geometry.dae” and “gripper_kinematics.dae” as defined in B.5.2 and B.5.3. This linking defines the flow of the joint value from a kinematic calculation to the transformation in the geometry.

```
<COLLADA version="1.5.0">
  <asset>
    <contributor>
      <author>WriterVendor</author>
      <author_website>http://www.WriterVendor.com</author_website>
      <authoring_tool>"WriterName" "WriterVersion" "WriterRelease"</authoring_tool>
      <comments>This document defines the combination of the gripper's geometry and kinematics</comments>
    </contributor>
    <created>2014-04-14T11:37:18.1875000</created>
    <modified>2014-04-14T11:37:18.1875000</modified>
    <unit meter="1.0" name="meter" />
    <up_axis>Z_UP</up_axis>
  </asset>
  <scene>
    <instance_visual_scene url="./gripper_geometry.dae#visualscene" />
    <instance_kinematics_scene sid="inst_linearunit_kinscene" url="./gripper_kinematics.dae#gripper_kinscene">
      <bind_kinematics_model node="gripper">
        <param>gripper_kinscene_gripper_motion_gripper_kinematics_inst_gripper_kinematics_model</param>
      </bind_kinematics_model>
      <bind_joint_axis target="geomlink1/transform">
        <axis>
          <param>gripper_kinscene_gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_joint1_axis0</param>
        </axis>
        <value>
          <param>gripper_kinscene_gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_joint1_value</param>
        </value>
      </bind_joint_axis>
```

```
      <bind_joint_axis target="geomlink1/transform">
        <axis>
          <param>gripper_kinscene_gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_joint
2_axis0</param>
        </axis>
        <value>
          <param>gripper_kinscene_gripper_motion_gripper_kinematics_inst_gripper_kinmodel_inst_joint
2_value</param>
        </value>
      </bind_joint_axis>
    </instance_kinematics_scene>
  </scene>
</COLLADA>
```

Figure B.39: XML representation of the COLLADA document gripper.dae

B.5.5 Combination of CAEX and COLLADA into AML

This example extends the AML document from clause B.4. In addition to the existing InternalElements “LinearUnit” and “SCARA Robot” the AML document has an InternalElement “JawGripper” including an ExternalInterface named “FileLinkExplicit” derived from the InterfaceClass “COLLADAInterface”, which is part of the standard InterfaceClassLibrary “AutomationMLInterfaceClassLib”. This ExternalInterface is used to reference the complete scene explicitly and in particular the gripper as depicted in Figure B.40.

The InternalElement “SCARA Robot” has an additional child InternalElement “PublishedFlangeFrame” with applied RoleClass “Frame”. This InternalElement has two ExternalInterfaces: an ExternalInterface derived from the InterfaceClass “COLLADAInterface” to reference a COLLADA node implicitly (“FileLinkImplicit”) and an ExternalInterface derived from the InterfaceClass “AttachmentInterface” to define an attachment (“Attachment”).

The InternalElement “JawGripper” has two child InternalElements (“PublishedBaseFrame”, “PublishedFlangeFrame”) with applied RoleClass “Frame”, which is part of the standard RoleClassLibrary as well. The InternalElements “PublishedBaseFrame” and “PublishedFlangeFrame” have two ExternalInterfaces: an ExternalInterface derived from the InterfaceClass “COLLADAInterface” to reference a COLLADA node implicitly (“FileLinkImplicit”) and an ExternalInterface derived from the InterfaceClass “AttachmentInterface” to define an attachment (“Attachment”).

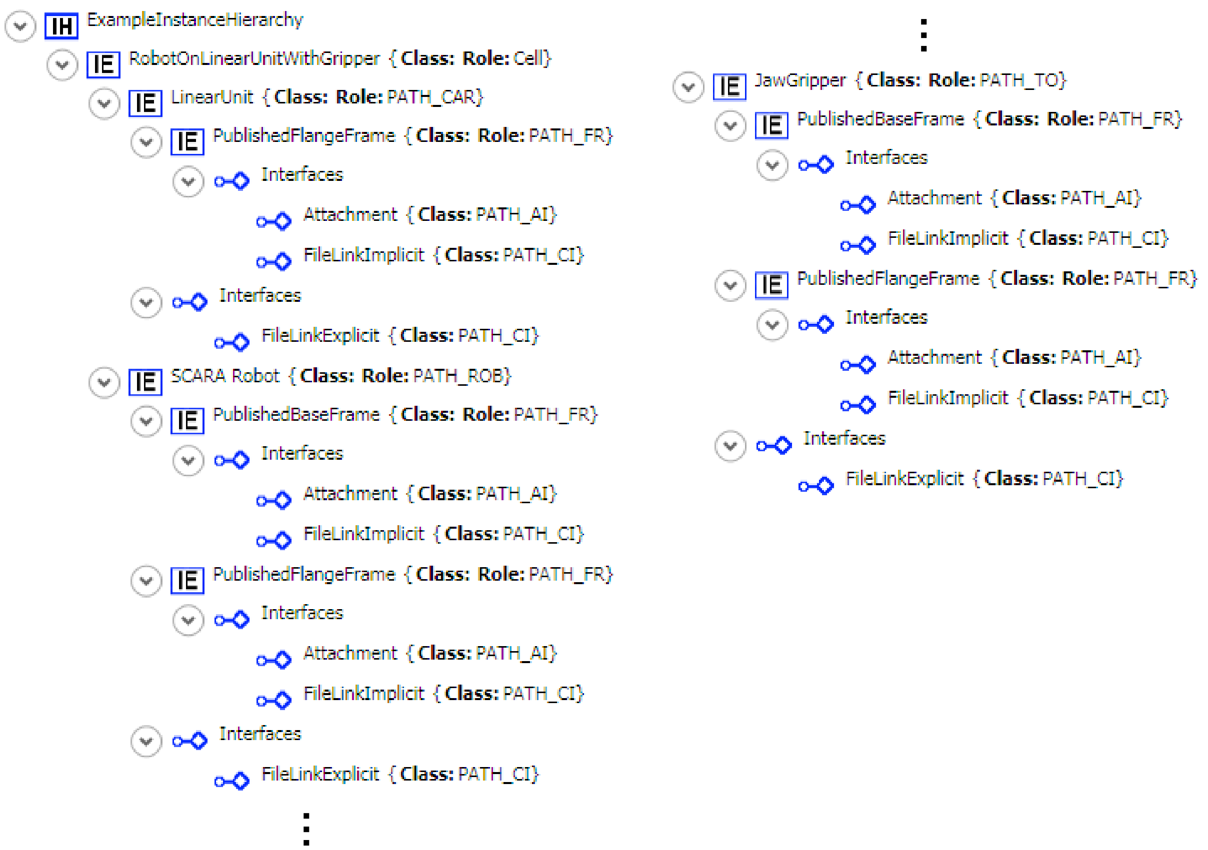

Figure B.40: Hierarchy of the AML document

Figure B.41 shows the relevant part of the AML document and how to publish frames for attaching the gripper to the robot, which are defined in AML.

```
<InstanceHierarchy Name="ExampleInstanceHierarchy">
  <InternalElement Name="RobotOnLinearUnitWithGripper" ID="GUID0">
    <InternalElement Name="LinearUnit" ID="GUID1">
      <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID2">
        <Attribute Name="refType" AttributeDataType="xs:string">
          <Value>explicit</Value>
        </Attribute>
        <Attribute Name="refURI" AttributeDataType="xs:anyURI">
          <Value>./linear_unit.dae</Value>
        </Attribute>
      </ExternalInterface>
      <InternalElement Name="PublishedFlangeFrame" ID="GUID3">
        <ExternalInterface Name="Attachment" RefBaseClassPath="PATH_AI" ID="GUID4" />
        <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID5">
          <Attribute Name="refType" AttributeDataType="xs:string">
            <Value>implicit</Value>
          </Attribute>
          <Attribute Name="refURI" AttributeDataType="xs:anyURI">
            <Value>./linear_unit.dae#linearunit_kinematics</Value>
```

```
        </Attribute>
        <Attribute Name="target" AttributeDataType="xs:token">
          <Value>./flangeframe</Value>
        </Attribute>
      </ExternalInterface>
      <RoleRequirements RefBaseRoleClassPath="PATH_FR" />
    </InternalElement>
    <RoleRequirements RefBaseRoleClassPath="PATH_CAR" />
  </InternalElement>
  <InternalElement Name="SCARA Robot" ID="GUID6">
    <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID7">
      <Attribute Name="refType" AttributeDataType="xs:string">
        <Value>explicit</Value>
      </Attribute>
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>./robot.dae</Value>
      </Attribute>
    </ExternalInterface>
    <InternalElement Name="PublishedBaseFrame" ID="GUID8">
      <ExternalInterface Name="Attachment" RefBaseClassPath="PATH_AI" ID="GUID9" />
      <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID10">
        <Attribute Name="refType" AttributeDataType="xs:string">
          <Value>implicit</Value>
        </Attribute>
        <Attribute Name="refURI" AttributeDataType="xs:anyURI">
          <Value>./robot.dae#robot_kinematics</Value>
        </Attribute>
        <Attribute Name="target" AttributeDataType="xs:token">
          <Value>./baseframe</Value>
        </Attribute>
      </ExternalInterface>
      <RoleRequirements RefBaseRoleClassPath="PATH_FR" />
    </InternalElement>
    <InternalElement Name="PublishedFlangeFrame" ID="GUID11">
      <ExternalInterface Name="Attachment" RefBaseClassPath="PATH_AI" ID="GUID12" />
      <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID13">
        <Attribute Name="refType" AttributeDataType="xs:string">
          <Value>implicit</Value>
        </Attribute>
        <Attribute Name="refURI" AttributeDataType="xs:anyURI">
          <Value>./robot.dae#robot_kinematics</Value>
        </Attribute>
        <Attribute Name="target" AttributeDataType="xs:token">
          <Value>./flangeframe</Value>
        </Attribute>
      </ExternalInterface>
```

```xml
        <RoleRequirements RefBaseRoleClassPath="PATH_FR" />
      </InternalElement>
      <RoleRequirements RefBaseRoleClassPath="PATH_ROB" />
    </InternalElement>
    <InternalElement Name="JawGripper" ID="GUID14">
      <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID15">
        <Attribute Name="refType" AttributeDataType="xs:string">
          <Value>explicit</Value>
        </Attribute>
        <Attribute Name="refURI" AttributeDataType="xs:anyURI">
          <Value>./gripper.dae</Value>
        </Attribute>
      </ExternalInterface>
      <InternalElement Name="PublishedBaseFrame" ID="GUID16">
        <ExternalInterface Name="Attachment" RefBaseClassPath="PATH_AI" ID="GUID17" />
        <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID18">
          <Attribute Name="refType" AttributeDataType="xs:string">
            <Value>implicit</Value>
          </Attribute>
          <Attribute Name="refURI" AttributeDataType="xs:anyURI">
            <Value>./robot.dae#robot_kinematics</Value>
          </Attribute>
          <Attribute Name="target" AttributeDataType="xs:token">
            <Value>./baseframe</Value>
          </Attribute>
        </ExternalInterface>
        <RoleRequirements RefBaseRoleClassPath="PATH_FR" />
      </InternalElement>
      <InternalElement Name="PublishedFlangeFrame" ID="GUID19">
        <ExternalInterface Name="Attachment" RefBaseClassPath="PATH_AI" ID="GUID20" />
        <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID21">
          <Attribute Name="refType" AttributeDataType="xs:string">
            <Value>implicit</Value>
          </Attribute>
          <Attribute Name="refURI" AttributeDataType="xs:anyURI">
            <Value>./robot.dae#robot_kinematics</Value>
          </Attribute>
          <Attribute Name="target" AttributeDataType="xs:token">
            <Value>./flangeframe</Value>
          </Attribute>
        </ExternalInterface>
        <RoleRequirements RefBaseRoleClassPath="PATH_FR" />
      </InternalElement>
      <RoleRequirements RefBaseRoleClassPath="PATH_TO" />
    </InternalElement>
    <InternalLink Name="AttachmentLinearUnitMovesRobot" RefPartnerSideA="GUID3:Attachment" RefPartne
```

```
rSideB="GUID8:Attachment" />
    <InternalLink Name="AttachmentRobotMovesGripper" RefPartnerSideA="GUID11:Attachment" RefPartnerSideB="GUID16:Attachment" />
  </InternalElement>
</InstanceHierarchy>
```

Figure B.41: XML representation of the AML document

Finally, the connection is established by the use of an InternalLink, which connects both published frames. Figure B.42 shows the relevant part of the AML document and how to define an InternalLink. The link is stored in the InternalElement “RobotOnLinearUnitWithGripper“, which is the lowest common parent of the connected AML objectsaccording to IEC 62714-1.

```
<InstanceHierarchy Name="ExampleInstanceHierarchy">
  <InternalElement Name="RobotOnLinearUnitWithGripper" ID="GUID0">
    [...]
    <InternalLink Name="AttachmentLinearUnitMovesRobot"
       RefPartnerSideA="GUID3:Attachment"
       RefPartnerSideB="GUID8:Attachment" />
    <InternalLink Name="AttachmentRobotMovesGripper"
       RefPartnerSideA="GUID11:Attachment"
       RefPartnerSideB="GUID16:Attachment" />
  </InternalElement>
</InstanceHierarchy>
```

Figure B.42: XML representation of the AML document

Figure B.43 depicts the resulting scene with the robot attached to the linear unit.

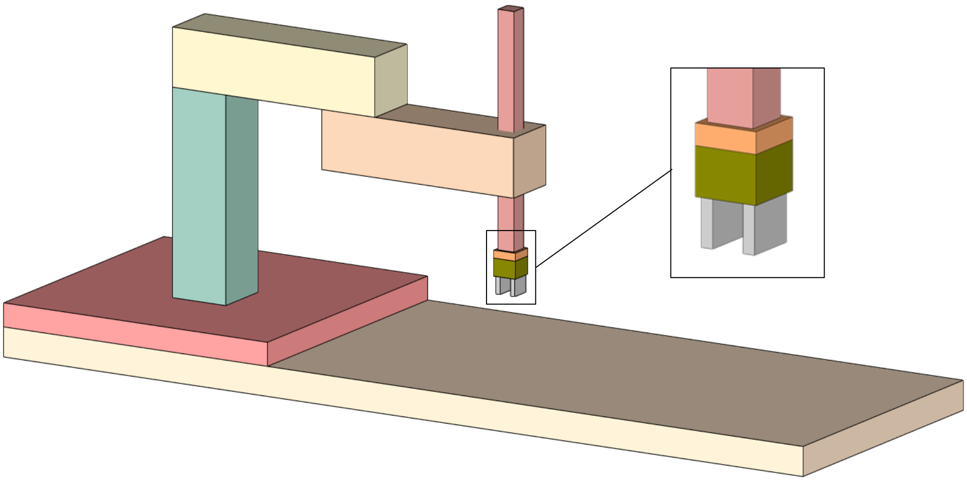

Figure B.43: Visualization of the robot on a linear unit and attached gripper

B.6 Modelling an AML document of a work piece connected to a gripper in CAEX and COLLADA

B.6.1 General

Clause B.6 describes a specialized use case of a geometric attachment between a parent and a child AML object(both of type InternalElement). In general any descendant geometry of an attached child AML object is transformed (moved or rotated) by the parent AML object due to the parent child relationship. If the attached child AML object references a COLLADA node whose geometry is defined in an ancestor COLLADA node, some additional requirements shall be met to move the geometry together with the attachment. In this clause, the example from clauses B.2, B.3, B.4, and B.4 is supplemented by a further work object, which will be attached to the gripper defined in B.5.

B.6.2 Implicit upper boundary

For a better understanding, Figure B.44 shows a simplified example of a robot with two frames (“A”, “B”) and a box with two frames (“C”,”D”). Frame “D” represents a child node of frame “C” and should be attached to frame “B” of the robot. The geometric shapeof the box is defined in frame “C” and thus it is not explicitly part of the attachment.

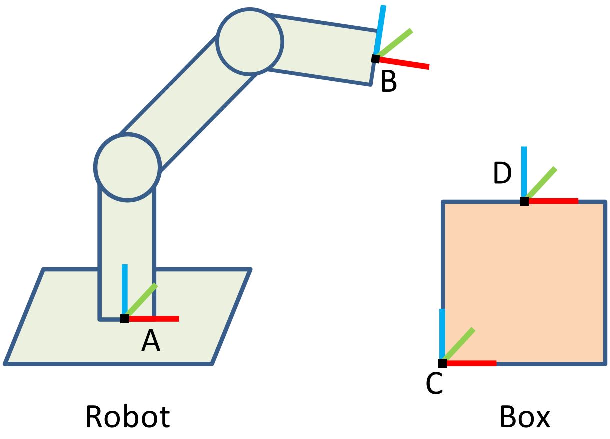

Figure B.44: Example for implicit upper boundary

In this example, an AML and two COLLADA documents are used. The robot and the box are defined in separated COLLADA documents while the topology and the geometric attachment are defined in a top level AML document. The structure is depicted in Figure B.45 .

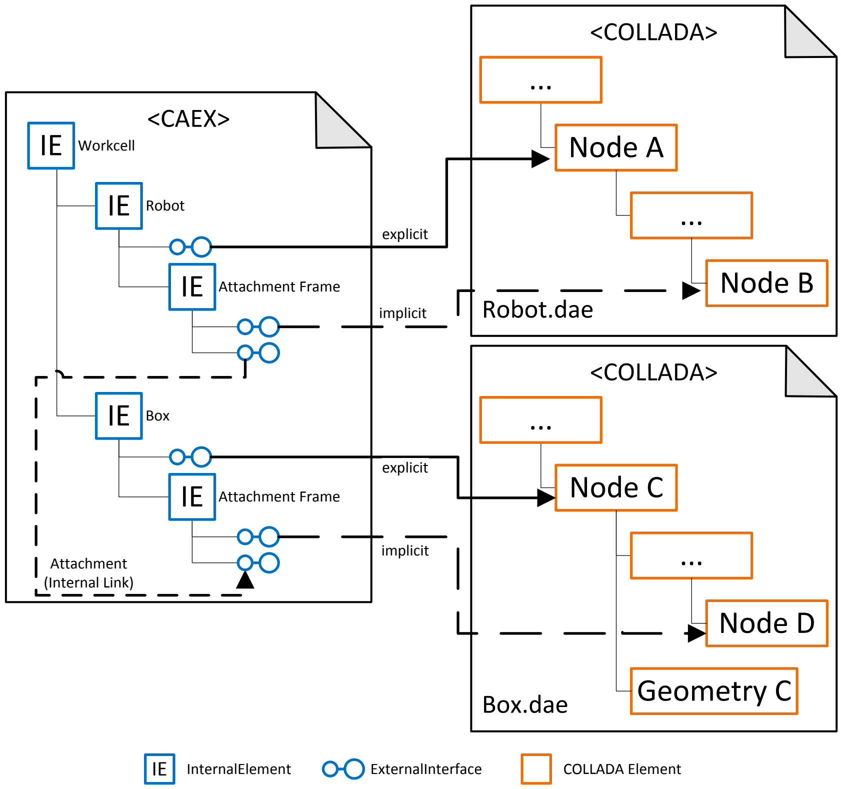

Figure B.45: Structure for attachments between objects in CAEX

The AML document has two InternalElements (“Robot”, “Box”) by means of AML objects.Both AML objects have a reference to a COLLADA geometry object by using explicit references. The AML object “Robot” references the COLLADA document Robot.dae and the AML object “Box” references the COLLADA document Box.dae.Furthermore both AML objects have a child InternalElement “Attachment Frame” with assigned RoleClass “Frame”. Each InternalElement “Attachment Frame” has an implicit ExternalInterface derived from the InterfaceClass “COLLADAInterface” which references a node defined in the COLLADA document. Additionally each InternalElement “Attachment Frame” has ExternalInterface derived from the InterfaceClass “AttachmentInterface”, which may be connected or linked. In this example the implicit reference of the AML object “Box” addresses the COLLADA node“Node D” which is a child of the COLLADA node “Node C”. The geometry of the box is defined in the scope of the COLLADA node “Node C”.

In accordance with the requirements for an attachment defined in Clause 6, this example is valid and the geometry of the box is moved with the attachment since it fulfils the requirements:

- The “AttachmentInterface” is child of an AML object with the RoleClass “Frame” and the object to be moved is the parent AML object “Box”.
- The AML object owning an “AttachmentInterface” has the RoleClass “Frame” and it has an ExternalInterface derived from the InterfaceClass “COLLADAInterface”. Its parent AML object “Box” has anderived from the InterfaceClass “COLLADAInterface” as well. Applying this, the “COLLADAInterface” of the parent AML object represents the boundary of the geometry of the attached geometry.

In the subclauses B.6.3 and B.6.4 , the example from B.2, B.3, B.4, and B.5 is continued. According to the definition of the box a work piece is defined and attached to a gripper through an additional frame. The geometry of the work piece is defined in an ancestor COLLADA node.

B.6.3 Definition of the work piece

The work piece used in this example is defined in the COLLADA document “workpiece.dae”. To keep the COLLADA content to a minimum, the geometry of the work piece is represented by a rectangular cuboid, which is defined within the COLLADA document “cube.dae” as introduced in Figure A.7. The work piece has a different scaling and provides a custom material. Figure B.46 shows the visualization of the work piece (“WorkPiece”) and its additional frame.

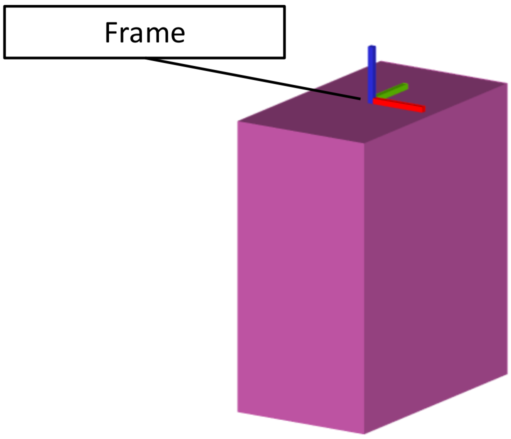

Figure B.46: Visualization of the work piece with additional frame

The complete COLLADA document of the work piece (“workpiece.dae”) is shown in Figure B.47. It uses the geometry definition from the external COLLADA document “cube.dae” taken from Figure A.7.

```
<COLLADA version="1.5.0">
  <asset>
    <contributor>
      <author>WriterVendor</author>
      <author_website>http://www.WriterVendor.com</author_website>
```

```
    <authoring_tool>"WriterName" "WriterVersion" "WriterRelease"</authoring_tool>
    <comments>This document defines the geometry of a work piece.
              An additional node representing a frame is provided.</comments>
  </contributor>
  <created>2014-04-14T11:37:18.1875000</created>
  <modified>2014-04-14T11:37:18.1875000</modified>
  <unit meter="1.0" name="meter" />
  <up_axis>Z_UP</up_axis>
</asset>
<library_effects>
  <effect id="wpeffect-fx">
    <profile_COMMON>
      <technique sid="COMMON">
        <phong>
          <diffuse>
            <color>0.71 0.31 0.61 1</color>
          </diffuse>
        </phong>
      </technique>
    </profile_COMMON>
  </effect>
</library_effects>
<library_materials>
  <material id="wpeffect" name="WorkPieceEffect">
    <instance_effect url="#wpeffect-fx" />
  </material>
</library_materials>
<library_visual_scenes>
  <visual_scene id="visualscene">
    <node id="workpiece" name="WorkPiece">
      <scale>0.04 0.07 0.08</scale>
      <node sid="workpieceframe">
        <translate>0.5 0.5 1</translate>
      </node>
      <instance_geometry url="./cube.dae#cube-geom">
        <bind_material>
          <technique_common>
            <instance_material symbol="mat" target="#wpeffect" />
          </technique_common>
        </bind_material>
      </instance_geometry>
    </node>
  </visual_scene>
</library_visual_scenes>
<scene>
  <instance_visual_scene url="#visualscene" />
```

```
  </scene>
</COLLADA>
```

Figure B.47: COLLADA document of work piece with additional frame

B.6.4 Combination of CAEX and COLLADA into AML

This example extends the AML document from clause B.5. In addition to the existing InternalElements "LinearUnit", "SCARA Robot" and "JawGripper" the AML document has an InternalElement "WorkPiece" including an ExternalInterface named "FileLinkExplicit" derived from the InterfaceClass "COLLADAInterface", which is part of the standard InterfaceClassLibrary "AutomationMLInterfaceClassLib". This ExternalInterface is used to reference the complete scene explicitly and in particular the work piece as depicted in Figure B.48.

The InternalElement "WorkPiece" has a child InternalElement ("PublishedFrame") with applied RoleClass "Frame", which is part of the standard RoleClassLibrary. The InternalElement "PublishedFrame" has an ExternalInterface derived from the InterfaceClass "COLLADAInterface" to reference a COLLADA node implicitly ("FileLinkImplicit") and an ExternalInterface derived from the InterfaceClass "AttachmentInterface" to define an attachment ("Attachment") as depicted in Figure B.48.

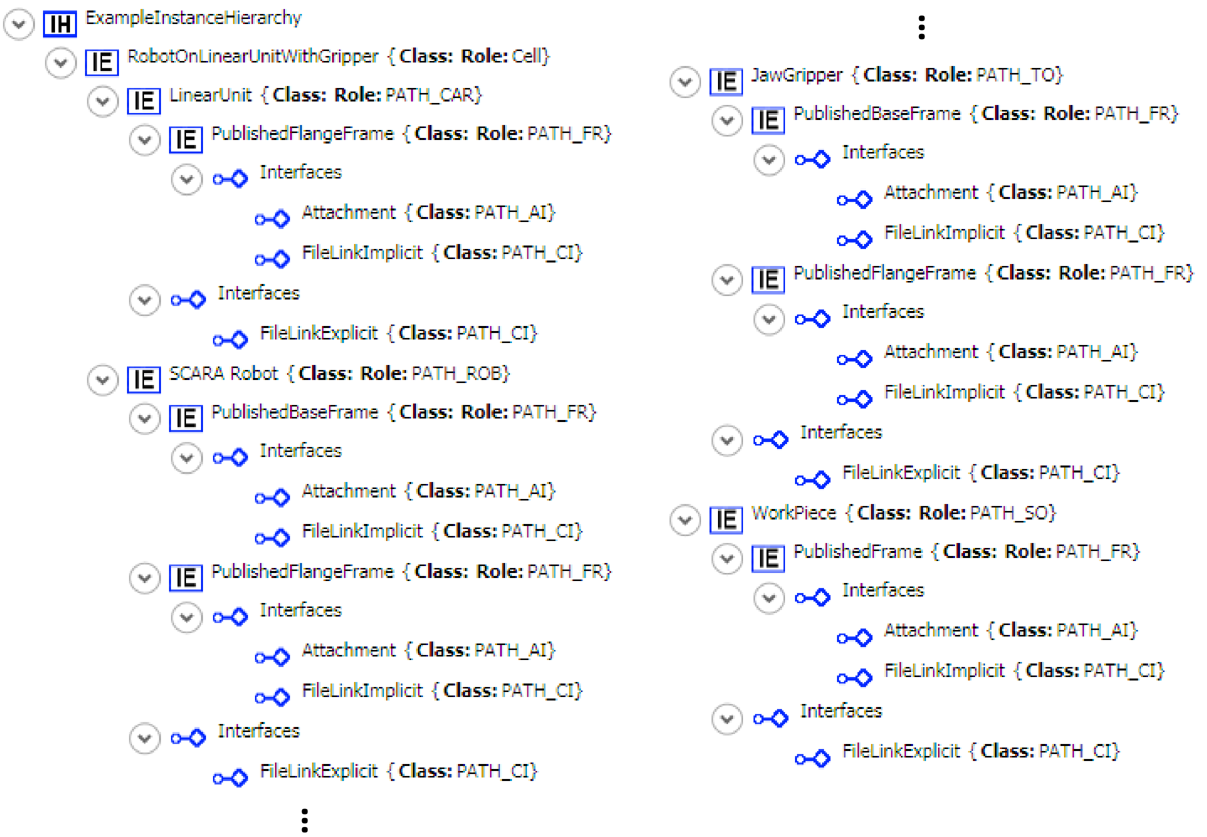

Figure B.48: Hierarchy of the AML document

Figure B.49 shows the relevant part of the AML document.

```
<InstanceHierarchy Name="ExampleInstanceHierarchy">
  <InternalElement Name="RobotOnLinearUnitWithGripperAndWorkpiece" ID="GUID0">
    <InternalElement Name="LinearUnit" ID="GUID1">
      <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID2">
        <Attribute Name="refType" AttributeDataType="xs:string">
          <Value>explicit</Value>
```

```xml
      </Attribute>
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>./linear_unit.dae</Value>
      </Attribute>
    </ExternalInterface>
    <InternalElement Name="PublishedFlangeFrame" ID="GUID3">
      <ExternalInterface Name="Attachment" RefBaseClassPath="PATH_AI" ID="GUID4" />
      <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID5">
        <Attribute Name="refType" AttributeDataType="xs:string">
          <Value>implicit</Value>
        </Attribute>
        <Attribute Name="refURI" AttributeDataType="xs:anyURI">
          <Value>./linear_unit.dae#linearunit_kinematics</Value>
        </Attribute>
        <Attribute Name="target" AttributeDataType="xs:token">
          <Value>./flangeframe</Value>
        </Attribute>
      </ExternalInterface>
      <RoleRequirements RefBaseRoleClassPath="PATH_FR" />
    </InternalElement>
    <RoleRequirements RefBaseRoleClassPath="PATH_CAR" />
  </InternalElement>
  <InternalElement Name="SCARA Robot" ID="GUID6">
    <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID7">
      <Attribute Name="refType" AttributeDataType="xs:string">
        <Value>explicit</Value>
      </Attribute>
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>./robot.dae</Value>
      </Attribute>
    </ExternalInterface>
    <InternalElement Name="PublishedBaseFrame" ID="GUID8">
      <ExternalInterface Name="Attachment" RefBaseClassPath="PATH_AI" ID="GUID9" />
      <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID10">
        <Attribute Name="refType" AttributeDataType="xs:string">
          <Value>implicit</Value>
        </Attribute>
        <Attribute Name="refURI" AttributeDataType="xs:anyURI">
          <Value>./robot.dae#robot_kinematics</Value>
        </Attribute>
        <Attribute Name="target" AttributeDataType="xs:token">
          <Value>./baseframe</Value>
        </Attribute>
      </ExternalInterface>
      <RoleRequirements RefBaseRoleClassPath="PATH_FR" />
    </InternalElement>
```

```xml
    <InternalElement Name="PublishedFlangeFrame" ID="GUID11">
      <ExternalInterface Name="Attachment" RefBaseClassPath="PATH_AI" ID="GUID12" />
      <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID13">
        <Attribute Name="refType" AttributeDataType="xs:string">
          <Value>implicit</Value>
        </Attribute>
        <Attribute Name="refURI" AttributeDataType="xs:anyURI">
          <Value>./robot.dae#robot_kinematics</Value>
        </Attribute>
        <Attribute Name="target" AttributeDataType="xs:token">
          <Value>./flangeframe</Value>
        </Attribute>
      </ExternalInterface>
      <RoleRequirements RefBaseRoleClassPath="PATH_FR" />
    </InternalElement>
    <RoleRequirements RefBaseRoleClassPath="PATH_ROB" />
  </InternalElement>
  <InternalElement Name="JawGripper" ID="GUID14">
    <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID15">
      <Attribute Name="refType" AttributeDataType="xs:string">
        <Value>explicit</Value>
      </Attribute>
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>./gripper.dae</Value>
      </Attribute>
    </ExternalInterface>
    <InternalElement Name="PublishedBaseFrame" ID="GUID16">
      <ExternalInterface Name="Attachment" RefBaseClassPath="PATH_AI" ID="GUID17" />
      <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID18">
        <Attribute Name="refType" AttributeDataType="xs:string">
          <Value>implicit</Value>
        </Attribute>
        <Attribute Name="refURI" AttributeDataType="xs:anyURI">
          <Value>./robot.dae#robot_kinematics</Value>
        </Attribute>
        <Attribute Name="target" AttributeDataType="xs:token">
          <Value>./baseframe</Value>
        </Attribute>
      </ExternalInterface>
      <RoleRequirements RefBaseRoleClassPath="PATH_FR" />
    </InternalElement>
    <InternalElement Name="PublishedFlangeFrame" ID="GUID19">
      <ExternalInterface Name="Attachment" RefBaseClassPath="PATH_AI" ID="GUID20" />
      <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID21">
        <Attribute Name="refType" AttributeDataType="xs:string">
          <Value>implicit</Value>
```

```
        </Attribute>
        <Attribute Name="refURI" AttributeDataType="xs:anyURI">
          <Value>./robot.dae#robot_kinematics</Value>
        </Attribute>
        <Attribute Name="target" AttributeDataType="xs:token">
          <Value>./flangeframe</Value>
        </Attribute>
      </ExternalInterface>
      <RoleRequirements RefBaseRoleClassPath="PATH_FR" />
    </InternalElement>
    <RoleRequirements RefBaseRoleClassPath="PATH_TO" />
  </InternalElement>
  <InternalElement Name="WorkPiece" ID="GUID22">
    <ExternalInterface Name="FileLinkExplicit" RefBaseClassPath="PATH_CI" ID="GUID23">
      <Attribute Name="refType" AttributeDataType="xs:string">
        <Value>explicit</Value>
      </Attribute>
      <Attribute Name="refURI" AttributeDataType="xs:anyURI">
        <Value>./workpiece.dae</Value>
      </Attribute>
    </ExternalInterface>
    <InternalElement Name="PublishedFrame" ID="GUID24">
      <ExternalInterface Name="Attachment" RefBaseClassPath="PATH_AI" ID="GUID25" />
      <ExternalInterface Name="FileLinkImplicit" RefBaseClassPath="PATH_CI" ID="GUID26">
        <Attribute Name="refType" AttributeDataType="xs:string">
          <Value>implicit</Value>
        </Attribute>
        <Attribute Name="refURI" AttributeDataType="xs:anyURI">
          <Value>./workpiece.dae#workpiece</Value>
        </Attribute>
        <Attribute Name="target" AttributeDataType="xs:token">
          <Value>./workpieceframe</Value>
        </Attribute>
      </ExternalInterface>
      <RoleRequirements RefBaseRoleClassPath="PATH_FR" />
    </InternalElement>
    <RoleRequirements RefBaseRoleClassPath="PATH_SO" />
  </InternalElement>
</InternalElement>
</InstanceHierarchy>
```

Figure B.49: XML representation of the AML document

As a result, the work piece is attached to the gripper as depicted in Figure B.50. In opposite to mounting the attaching does not include any additional transformation of the attached object.

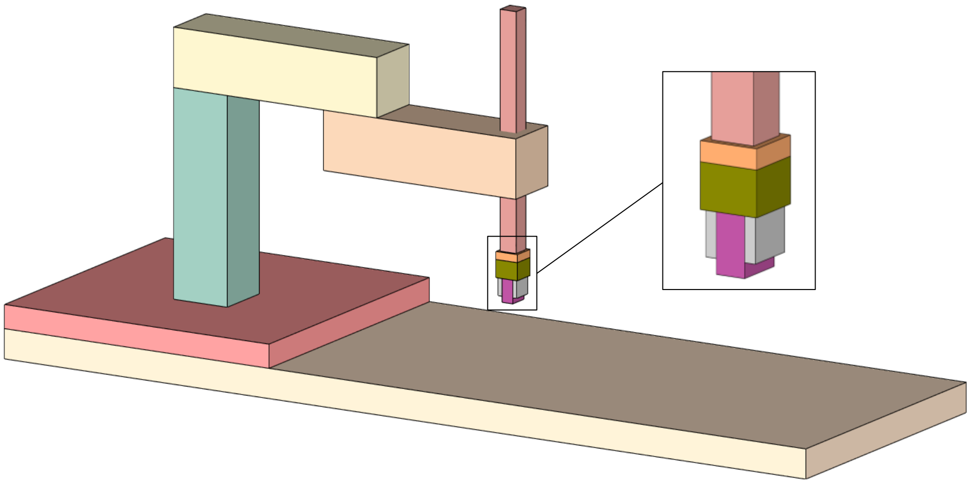

Figure B.50: Attachment between geometric AML objects

Finally, the connection is established by the use of an InternalLink, which connects both published frames. Figure B.51 shows the relevant part of the AML document and how to define an InternalLink. The link is stored at the CAEX InternalElement “Workcell“, which is the lowest common parent of the connected CAEX objects according to IEC 62714-1.

```
<InstanceHierarchy Name="ExampleInstanceHierarchy">
  <InternalElement Name="RobotOnLinearUnitWithGripperAndWorkpiece" ID="GUID0">
    <InternalLink Name="AttachmentLinearUnitMovesRobot"
       RefPartnerSideA="GUID3:Attachment"
       RefPartnerSideB="GUID8:Attachment" />
    <InternalLink Name="AttachmentRobotMovesGripper"
       RefPartnerSideA="GUID11:Attachment"
       RefPartnerSideB="GUID16:Attachment" />
    <InternalLink Name="AttachmentGripperMovesWorkpiece"
       RefPartnerSideA="GUID19:Attachment"
       RefPartnerSideB="GUID24:Attachment" />
  </InternalElement>
</InstanceHierarchy>
```

Figure B.51: XML representation of the AML document

Annex C XML representation of AML libraries

C.1 AutomationMLBaseRoleClassLib

Figure C.1 shows the AML libraries AutomationMLBaseRoleClassLib which contains the RoleClass “Frame” specified in this document.

```
<CAEXFile xsi:noNamespaceSchemaLocation="CAEX_ClassModel_V2.15.xsd" FileName="AutomationMLBaseRoleCl
assLib.aml" SchemaVersion="2.15">
  <AdditionalInformation AutomationMLVersion="2.0" />
  <AdditionalInformation>
    <WriterHeader>
      <WriterName>AutomationML e.V.</WriterName>
      <WriterID>AutomationML e.V.</WriterID>
      <WriterVendor>AutomationML e.V.</WriterVendor>
      <WriterVendorURL>www.automationml.org</WriterVendorURL>
      <WriterVersion>1.0</WriterVersion>
      <WriterRelease>1.0.0</WriterRelease>
      <LastWritingDateTime>2013-03-01</LastWritingDateTime>
      <WriterProjectTitle>Automation Markup Language Standard Libraries</WriterProjectTitle>
      <WriterProjectID>Automation Markup Language Standard Libraries</WriterProjectID>
    </WriterHeader>
  </AdditionalInformation>
  <ExternalReference Path="AutomationMLInterfaceClassLib.aml" Alias="AutomationMLInterfaceClassLib"
/>
  <RoleClassLib Name="AutomationMLBaseRoleClassLib">
    <Description>Automation Markup Language Base Role class Library - Part 1 Content extended with
Part 3 Content</Description>
    <Version>2.2.1</Version>
    <RoleClass Name="AutomationMLBaseRole">
      <RoleClass Name="Group" RefBaseClassPath="AutomationMLBaseRole">
        <Attribute Name="AssociatedFacet" AttributeDataType="xs:string" />
      </RoleClass>
      <RoleClass Name="Facet" RefBaseClassPath="AutomationMLBaseRole" />
      <RoleClass Name="Port" RefBaseClassPath="AutomationMLBaseRole">
        <Attribute Name="Direction" AttributeDataType="xs:string" />
        <Attribute Name="Cardinality">
          <Attribute Name="MinOccur" AttributeDataType="xs:unsignedInt" />
          <Attribute Name="MaxOccur" AttributeDataType="xs:unsignedInt" />
        </Attribute>
        <Attribute Name="Category" AttributeDataType="xs:string" />
        <ExternalInterface Name="ConnectionPoint" ID="9942bd9c-c19d-44e4-a197-
11b9edf264e7" RefBaseClassPath="AutomationMLInterfaceClassLib@AutomationMLInterfaceClassLib/Automati
onMLBaseInterface/PortConnector" />
      </RoleClass>
      <RoleClass Name="Resource" RefBaseClassPath="AutomationMLBaseRole" />
      <RoleClass Name="Product" RefBaseClassPath="AutomationMLBaseRole" />
      <RoleClass Name="Process" RefBaseClassPath="AutomationMLBaseRole" />
      <RoleClass Name="Structure" RefBaseClassPath="AutomationMLBaseRole">
```

```xml
        <RoleClass Name="ProductStructure" RefBaseClassPath="Structure" />
        <RoleClass Name="ProcessStructure" RefBaseClassPath="Structure" />
        <RoleClass Name="ResourceStructure" RefBaseClassPath="Structure" />
      </RoleClass>
      <RoleClass Name="PropertySet" RefBaseClassPath="AutomationMLBaseRole" />
      <RoleClass Name="Frame" RefBaseClassPath="AutomationMLBaseRole" />
    </RoleClass>
  </RoleClassLib>
</CAEXFile>
```

Figure C.1: XML representation of AML libraries AutomationMLBaseRoleClassLib

C.2 AutomationMLInterfaceClassLib

Figure C.2 shows the AML libraries AutomationMLInterfaceClassLib which contains the InterfaceClass “COLLADAInterface” specified in this document.

```xml
<CAEXFile xsi:noNamespaceSchemaLocation="CAEX_ClassModel_V2.15.xsd" FileName="AutomationMLInterfaceClassLib.aml" SchemaVersion="2.15">
  <AdditionalInformation AutomationMLVersion="2.0" />
  <AdditionalInformation>
    <WriterHeader>
      <WriterName>AutomationML e.V.</WriterName>
      <WriterID>AutomationML e.V.</WriterID>
      <WriterVendor>AutomationML e.V.</WriterVendor>
      <WriterVendorURL>www.automationml.org</WriterVendorURL>
      <WriterVersion>1.0</WriterVersion>
      <WriterRelease>1.0.0</WriterRelease>
      <LastWritingDateTime>2013-03-01</LastWritingDateTime>
      <WriterProjectTitle>Automation Markup Language Standard Libraries</WriterProjectTitle>
      <WriterProjectID>Automation Markup Language Standard Libraries</WriterProjectID>
    </WriterHeader>
  </AdditionalInformation>
  <InterfaceClassLib Name="AutomationMLInterfaceClassLib">
    <Description>Standard Automation Markup Language Interface Class Library - Part 1 Content extended with Part 3 Content</Description>
    <Version>2.2.1</Version>
    <InterfaceClass Name="AutomationMLBaseInterface">
      <InterfaceClass Name="Order" RefBaseClassPath="AutomationMLBaseInterface">
        <Attribute Name="Direction" AttributeDataType="xs:string" />
      </InterfaceClass>
      <InterfaceClass Name="PortConnector" RefBaseClassPath="AutomationMLBaseInterface" />
      <InterfaceClass Name="InterlockingConnector" RefBaseClassPath="AutomationMLBaseInterface" />
      <InterfaceClass Name="PPRConnector" RefBaseClassPath="AutomationMLBaseInterface" />
      <InterfaceClass Name="ExternalDataConnector" RefBaseClassPath="AutomationMLBaseInterface">
        <Attribute Name="refURI" AttributeDataType="xs:anyURI" />
        <InterfaceClass Name="COLLADAInterface" RefBaseClassPath="ExternalDataConnector">
          <Attribute Name="refType" AttributeDataType="xs:string" />
          <Attribute Name="target" AttributeDataType="xs:token" />
```

```
        </InterfaceClass>
        <InterfaceClass Name="PLCopenXMLInterface" RefBaseClassPath="ExternalDataConnector" />
      </InterfaceClass>
      <InterfaceClass Name="Communication" RefBaseClassPath="AutomationMLBaseInterface">
        <InterfaceClass Name="SignalInterface" RefBaseClassPath="Communication" />
      </InterfaceClass>
      <InterfaceClass Name="AttachmentInterface" RefBaseClassPath="AutomationMLBaseInterface" />
    </InterfaceClass>
  </InterfaceClassLib>
</CAEXFile>
```

Figure C.2: XML representation of AML libraries AutomationMLInterfaceClassLib