Recommended Practices

for

STEP AP242 Edition 3
Domain Model XML
Kinematics

Release 1.1

18 November 2022

Contacts:

Organizational	
Jochen Boy	Phil Rosché
PROSTEP AG	ACCR, LLC.
Dolivostraße 11	125 King Charles Circle
64293 Darmstadt / Germany	Summerville, SC 29485 / USA
jochen.boy@prostep.com	phil.rosche@accr-llc.com
Technical	
Guillaume Hirel	Alexandre Cros
T-Systems	Dassault Systèmes
guillaume.hirel@t-systems.com	alexandre.cros@3ds.com

Table of Contents

List of Figures

List of Tables

Document History

Revision	Date	Change
1.0	2021-11-26	Initial public release
1.1	2022-11-18	Final updates for AP242 Ed.3. Incorporated feedback from CAx-IF / JT-IF testing throughout the document.

Working History – Changes since version 0.12 (July 2020)

Revision	Date	Change
1.0	2021-11-26	**Adaptation for AP242 Ed.3 Domain Model**
		Section 1.1.3: Implementation changes to AP242 Ed.2
		Section 1.1.4: Impl. changes to AP242 Ed.2 Minor Rev.
		Throughout: Updated references from AP242 Ed.1 TC BO Model to AP242 Ed.3 Domain Model; Rename 'kinematic joint' with 'kinematic pair' all over the document.
		Section 1.1.2: URL of Domain Model corrected; remark how to find the Ed. 2 MR Domain Model XSD if not published yet by the ISO
		Section 4.2: additional comments regarding Jira issue BS10303-6167 and regarding the use of AxisPlacement
		Sections 4.2, 4.11 and 4.12: remark added, that SpecifiedOccurrences are mostly not supported by the PDM systems => currently out of scope
		Section 4.3: New attribute ActuatedPair.Name new and (Minor Revision) mandatory; new recommendation for ActuatedPair
		Section 4.3.1: LowOrderKinematicPair.Lower/UpperLimitActualTranslationX/Y/Z changed from xsd:string to xsd:double; Introduction of homokinetic_pair; Updates in spherical_pair_with_pin; Updates in universal_pair
		Section 4.3.2: HighOrderKinematicPair.CurveOrSurface1/2 changed from ComplexType/group to a simple reference; Clarification of the usage of HighOrderKinematicPair.orientation
		Section 4.3.3: Ditto for LowOrderKinematicPairWithMotionCoupling. Lower/UpperLimitRackDisplacement, .Pitch and .Radius1/2
		PropertyValueAssignment added to Representation and RepresentationItem for the mapping of validation properties. The relevant objects are KinematicLink (4.2), KinematicPair (section 4.3) Mechanism (section 4.4), LinkMotionAlongPath (section 4.8) and KinematicPathDefinedByNodes (section 4.9)
		New section 4.5 (Mechanism Structure)
		Section 4.12: LinkMotionRelationship.related: type changed; LinkMotionRelationship: attribute RelationType made optional
		Section 4.13: updates in the validation properties
		Some additional consistency rules added as Preprocessor recommendations in section 4.2, 4.3 and 4.4; recommendation for validation properties with value 0 updated
		Section 4.13.2.4: number of actuations now computed over the whole mechanism rather than for each KinematicPair
		New section 4.13.2.5: number of KinematicPairs for each kind of KinematicPair
		Old section 4.13.2.5: Grübler Count / Mobility Formula removed

1.1	2022-11-18	Section 1.1.2: update Ed3 Domain Model URL to official value from the ISO
		Section 4.2: cardinality of relationship between KinematicLink and Occurrence: rule exception for Rigid Groups removed
		Section 4.2, 4.5: update of preprocessor recommendations regarding Rigid Groups
		Section 4.3: typos corrected; new recommendation regarding sharing the same GeometrixCoordinateSpace for the AVP 'notional solids centroid' and the Mechanism
		Section 4.4: model splitting limitation removed (outdated); 'identical matrix' restriction removed regarding BaseLink
		Sections 4.4, 4.6, 4.7: 'fixed part' overworked to BaseLink
		Section 4.5: one red OccurrenceRelationship missing in the figure. Reuse AxisPlacement from Mechanism to build the NAOU under the dummy assembly nodes. Further updates in the mapping of rigid group during import/export
		Section 4.13.2: introduction added with reference to the mapping of validation properties in [AP242-PAS] + XML examples
		Section 4.13.2.3: name of validation property changed from 'number of moving parts' to 'number of moving KinematicLinks'
		Annex B: new issue numbers after switch from PDES Inc. Bugzilla to ISO Jira
		KinematicPair.Name added to all the XML examples

1   Introduction

1.1   Document Overview

1.1.1   Goal and Objectives

The goal of this document is to describe the recommended structure and attribute population for particular instance models created from the entities and attributes defined by the STEP AP242 "Managed Model-based 3D Engineering" Domain Model and populated according to its XML Schema.

The selected instance models illustrate how to encode kinematic data values that need to be exchanged in support of key industry requirements common across the mechanical design domain. The objectives of the usage guide are to:

- Support the short-term needs of the requirements of the Aerospace & Defense and the Automotive industries in the realm of kinematics.

- Prevent the emergence of "flavors", i.e., diverging/conflicting implementations of the AP242 Domain Model XML for different communities

- Ensure consistency with existing Recommended Practices, in particular the joint PDM-IF / MBx-IF / JT-IF Recommended Practices for AP242 Domain Model XML Product and Assembly Structure.

1.1.2   Scope

This document describes the Recommended Practices for the exchange of Kinematic data with external references to geometry files (regardless of file format). It is based on the STEP AP242 Edition 3 Domain Model (ISO 10303-4442:2022) and the corresponding XML schemas, which can be found at:

http://standards.iso.org/iso/ts/10303/-4442/ed-3/tech/xml-schema/domain_model

Therefore, the the XML header of AP242 Domain Model XML files which shall comply to Ed.3 shall look like the following:

```
xmlns:n0="http://standards.iso.org/iso/ts/10303/-4442/ed-3/tech/xml-
schema/domain_model " xsi:schemaLocation="http://stand-
ards.iso.org/iso/ts/10303/-4442/ed-3/tech/xml-schema/domain_model
http://standards.iso.org/iso/ts/10303/-4442/ed-3/tech/xml-schema/do-
main_model/DomainModel.xsd"
```

Note: due to some delays in the publication of the standard, this URL may not be available yet. In this case, please use a local copy of the Domain Model XSD as distributed within the interoperability forums.

AP242 is the first STEP Domain Model that is being implemented. During the development of this document and the related prototyping activities, several issues with the schemas and definitions have been identified. These have been documented in the ISO Jira system as official maintenance issues for AP242 and will dealt with as part of the ISO maintenance procedures. The issues are documented throughout this document and gathered in Annex B for reference.

1.1.3   Implementation Changes between AP242 Ed.1 TC and AP242 Ed.2

For a number of constructs, the implementation changed between the Ed.1 and Ed.2, i.e. ISO 10303-3001 Ed.2 and ISO 10303-4442. These changes became necessary due to deficiencies found in the original definitions of these elements, or to lay the foundation for future extensions of the data model.

The changes include structural changes such as embedding and referencing of XML elements as well as technical changes such as new element types or changed definition of attributes. Except very few of them, all these changes are upward compatible.

The list below provides an overview for the necessary implementation changes to support AP242 Ed.2 in scope of this document:

- New attribute ActuatedPair.Name new => section 4.3

- LinkMotionRelationship.related: type changed from ExternalGeometricModel to ComposedOrExternalGeometricModelSelect => section 4.12 (this has already been assumed in the Rec. Pracs.)

- LinkMotionDecompositionRelationship removed (not used)

- LinkMotionRelationship: attribute RelationType made optional => section 4.12

- PropertyValueAssignment added to Representation for the mapping of validation properties. The relevant objects are KinematicLink (4.2), KinematicPair (4.3), Mechanism (section 4.4), LinkMotionAlongPath (section 4.8) and KinematicPathDefinedByNodes (section 4.9)

- MotionModelAssociation now embedded into AssemblyDefinition (PartView) (section 5.1.4 in [242-PAS])

- HighOrderKinematicPair.CurveOrSurface1/2 changed from ComplexType/group to a simple reference (since the select type ExternalCurveOrSurfaceSelect contains only references) => section 4.3.2

- LowOrderKinematicPair.Lower/UpperLimitActualTranslationX/Y/Z changed from xsd:string to xsd:double (new mapping of EXPRESS type LengthMeasure) => section 4.3.1

- Ditto for LowOrderKinematicPairWithMotionCoupling.Lower/UpperLimitRackDisplacement, .Pitch and .Radius1/2 => section 4.3.3

- Clarification of the usage of HighOrderKinematicPair.orientation => section 4.3.2

1.1.4 Implementation Changes between AP242 Ed.2 and AP242 Ed.3

- Attribute ActuatedPair.Name now mandatory => section 4.3

1.1.5 Intended Audience

This document is intended to be an implementation guide for developers of CAD and file translation application systems that support and exchange Kinematic information with other systems and applications, in support of the design engineering and related downstream business processes.

1.1.6 Intended Use

This document is intended to be a manual and companion to the developer of STEP data exchange and translator software used by applications that support kinematic definitions. It provides guidelines for the consistent preprocessor instance model creation and requirement value encoding to enable meaningful information exchange between different systems and applications using the STEP AP242 Domain model, and guidelines for the consistent interpretation by a postprocessor of the STEP AP242 Domain model exchange file.

1.1.7 Document Style

The overall document proceeds in an incremental, step-by-step fashion to describe, and in parallel to illustrate the recommended instantiations of the XML elements in the STEP AP242 Domain model.

The "template" concept is used in this document. Structures and sub-structures are defined in one section; they are then re-used in other sections of the documents. These templates are represented by the blue boxes in the diagrams.

Templates for common core constructs, such as for instance Product, are defined in the Recommended Practices for AP242 Domain Model XML Product and Assembly Structure and re-used here. References to such templates are prefixed with [242-PAS].

The Instance Model diagram figures are presented using a graphical notation intended to illustrate the instance model.

Following each instance diagram, a table lists all the attributes of each displayed entity according to the XMLschema specification of ISO 10303-3001. The table includes not only the attributes of the EXPRESS schema of the AP242 Domain Model, but also inverse attributes of all possible relations to the element in question. Attributes that are considered important for the scope of these Recommended Practices are in these tables written in black. Attributes that are written in grey are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

Below the table, all recommended attributes (written in black) are listed and recommendations are made for them.

Finally, a STEP AP242 Domain model XML exchange structure example is included. The example exchange file corresponds directly to the instance model diagram and illustrates the very same thing using a different notation, i.e., STEP AP242 Domain model XML syntax versus the graphical instance model notation.

1.1.8 Document Structure

The overall scope of requirements is partitioned into a set of major sections corresponding to the identified units of functionality. Within a major section, there may be sub-sections. These sub-sections further divide the scope into smaller components of coherent functionality (called "feature") that interact with each other to realize the functionality of the entire unit.

There is generally a description of requirements and a corresponding instance diagram associated with each section and sub-section of this document. Each instance diagram is followed by a detailed explanation and specific recommendations for the entities used in the instantiation diagram example. The entity listing and explanation is in turn followed by the corresponding XML exchange structure example.
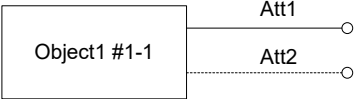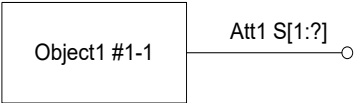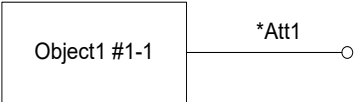
Within a section, diagrams corresponding to sub-sections incrementally build upon one another to finally achieve a complete instance model example that illustrates the entire scope of the unit of functionality.

1.1.9 Instantiation Diagrams

The diagrams are presented using a graphical notation intended to illustrate the instance model.

This notation is not EXPRESS-G and does not illustrate the XML schema; rather it is a graphical illustration of a specific population of a particular instance model of the schema. This notation supports:

- Illustration of entity instances and attribute values (both mapped as XML elements)

- Illustration and identification of referenced entity instances that are either fully illustrated in the current figure, or that refer to another template (if not fully illustrated in the current figure)

- Indication of optional attributes and optional reference entity instances (dashed lines),

- Illustration and identification of groups of functionally related instances (shaded bounding box), showing how XML elements are embedded into each other (the XML elements representing the entity instances placed below are embedded into the XML element representing the entity instance placed above), and

- Identification of specific attribute values (typically string values, may also be enumerated type values or numerical values).

A legend for the diagram notation is shown below:

Object1 #1-1	Object (instance of an EXPRESS ENTITY) After the #, an instance number is given
Object1 #1-1 — Att1 ○ Object1 #1-1 ┈ Att2 ○	Att1: mandatory attribute Att2: optional attribute

Object1 #1-1 — Att1 S[1:?] ○	Aggregate type for the definition of the cardinality constraint:
	
	B: Bag (non-ordered and my contain duplicates)
	
	S: Set (non-ordered and may not contain duplicates)
	
	L: List (ordered)
	
	    [x : y]: lower size : upper size
	
	    ?: unconstrained
	
	A: Array (indexed)
	
	    [x : y]: lower index : upper index

Object1 #1-1 — *Att1 ○	Additional constraint on the object: the attribute(s) depicted with '*' have to contain unique values. *Currently not used in this document*

Diagram	Description
Object1 #1-1 —(DER)Att1 ○	Derived Information from another object or attribute *Currently not used in this document*
STRING / REAL / BOOLEAN	Simple data types
Type1	User-defined data type *Currently not used in this document*
EnumType1	Enumeration Type (consists of a limited list of possible values defined for this type)
SelectType1 — 1 — Object1 #1-1 / Object2 #2-1	Select Type (is used if a relationship from an object may apply either to Object1 or Object2) For a better readability, the members of a select type are displayed using the inheritance link (see below) and the mutually exclusive constraint. This has the same semantic as a select type. *Currently not used in this document*
Object1 #1-1 — Rel1 / Rel2 ○ — Object2 #2-1	Attribute as relationship between two Objects (mandatory or optional), The circle at the end of the line gives the direction. Rel1: mandatory relationship Rel2: optional relationship
Object1 #1-1 — Rel1 / (INV) Rel2 S[1:?] ○ — Object2 #2-1	Rel2: inverse attribute (i.e. in the converse direction to Rel1) with cardinality constraint *Currently not used in this document*

(ABS)Object3 #1-1 ———— Att1 ○ 1 Object1 #1-1 Object2 #2-1 ——— (RT) Att1 ○	Inheritance from a Supertype to its Subtypes An Abstract Supertype (ABS) cannot be instanced without one of its non-abstract Subtypes 1: Only one subtype can be instantiated at a time (the subtypes are mutually exclusive). Per default, all the subtypes may be combined (not supported by XML) RT: an inherited attribute is retyped, for example through restriction of its type, length, value range, cardinality, mandatory instead of optional or DERIVED *Currently not used in this document*
Object1 #1-1 Rel1 ○ Object2 #2-1	Objects shown under each other within a blue colored square are embedded into each other in XML: here Object2 #2-1 is embedded into Object1 #1-1 as its XML element Rel1
Att1 ○ **Template1**	The templates defined in this document are re-used in other sections. This is the simple way to refer to a template (if the object referenced within the template is implicit, for example the object 'Classification' for the template 'Classification'
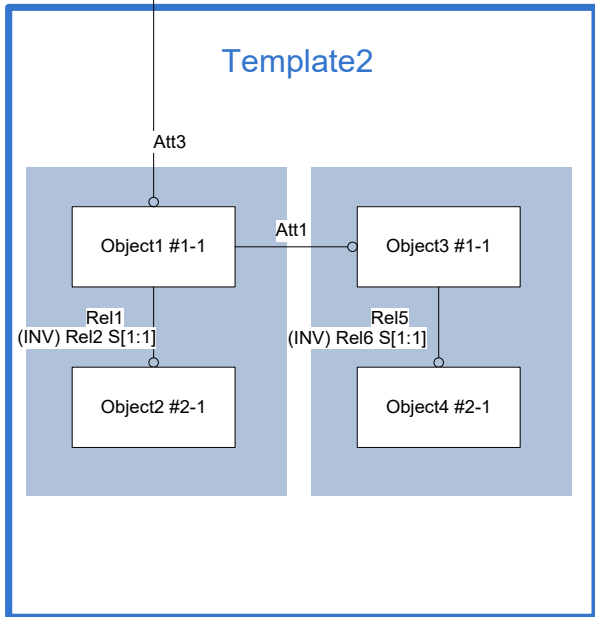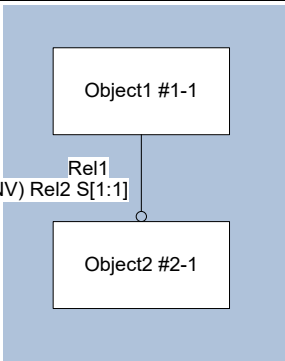
	If the template is more complex and the object referenced within the template is shown explicitly, portions of the reused template are displayed within a blue frame.
	Alternatively, a template may be reused through adding new XML containments to it

Table 1: Instance Diagram Notation

1.2   Organizational Framework

These Recommended Practices for AP242 Domain Model XML Kinematics are jointly developed and supported by several "communities", specifically the vendor and user communities devoted to the development and implementation of AP242 and its associated Domain Model. This section describes those communities' responsibilities.

1.2.1   Vendor Communities

The **CAx-IF** and **JT-IF** will be jointly responsible for the overall organization and development of this document. The CAx-IF and JT-IF will:

- Coordinate the creation of the document
- Verify the approach of the recommended practices in CAx-IF and JT-IF Test Rounds
- Publish result summaries of testing AP242 Domain Model Kinematics
- Ensure the consistency with existing CAx-IF, JT-IF and PDM-IF recommended practices

1.2.2   User Communities

The CAx-IF User Group is the user community for STEP geometry related topics, currently primarily comprised of Aerospace and Defence members, and supports the development of related use cases, their implementation and support by the standard. It will:

- Support the development of the document
- Provide subject matter experts
- Provide A&D requirements and ensure they are fulfilled
- Ensure the consistency with LOTAR standards

The JT Workflow Forum (JT-WF) is the Automotive user community supporting the development of the recommended practices for the ISO JT format. The JT-WF will:

- Support the development of the document
- Provide subject matter experts
- Provide Automotive requirements and ensure they are fulfilled

1.3   Maintenance of this Document

This document describes the recommended practices to implement Kinematic data using the AP242 Domain Model. It is based on the joint PDM-IF / CAx-IF / JT-IF Recommended Practices for AP242 Domain Model XML Product and Assembly Structure. Since Kinematics are an extension to the core scope that is specific to the CAD domain, it is documented in separate document with references to the core document where needed.

It is the responsibility of the CAx-IF and JT-IF to maintain not only this document, but also to ensure its consistency with the core document.

AFNeT, PDES, Inc., prostep ivip Association and VDA as the hosting organizations of the involved implementor forums will maintain and extend the document as long as it provides utility to the vendor community.

2 Scope

The following are within the scope of this document:

- Kinematic Mechanism

- Kinematic Motion

The following are out of scope for this document because they are or will be covered in other documents:

- Basic PDM Capabilities for Product Identification and Assembly Structure [242-PAS]

- Geometric and Assembly Validation Properties [242-PAS]

- External Element References (into Part 21 files or between Domain Model XML files)

- Implementations based on any edition of the AP242 Domain Model (ISO 10303-3001) other than Edition 3 (2022).

The following are out of scope for this document:

- Hierarchies of Mechanisms (GeometricRepresentationRelationship)

- Flexible Parts within Kinematics

- Use of SpecifiedOccurrence to define a different mechanism/motion depending on which multi-level occurrence is referred to (for example the propeller blades in left engine and the propeller blades in right engine)

- Composites

3 Reference to Recommended Practices

3.1 Reference to Core Document [242-PAS]

This document is an extension of the joint PDM-IF / MBx-IF / JT-IF Recommended Practices for AP242 Domain Model Product and Assembly Structure. Definitions of templates and XML elements contained in the core document are not repeated here.

> Where necessary, references to sections of the core document are given in this format:
>
> [242-PAS] 4.6.7

This means a reference to the Recommended Practices for AP242 Domain Model Product and Assembly Structure, section 4.6.7.

This Recommended Practices for AP242 Edition 3 Domain Model XML Kinematics are built on the following version of the core document:

Version 3.1; dated 2022-11-18

This document is publicly available from the CAx-IF and JT-IF web pages.

3.2 Listing of Recommended Practices Version in Exchange Files

For validation purposes, STEP processors shall state which Recommended Practice document and version thereof have been used in the creation of the STEP file. This will not only indicate what information a consumer can expect to find in the file, but even more importantly where to find it in the file.

This shall be done by adding a pre-defined string to the first string element of the `Documentation` attribute of the `Header` element in the XML file (for details see section [242-PAS] 4.1.5). The value follows a specific pattern well established in Part 21 files:

```
Document Type---Document Name---Document Version---Publication Date
```
The string corresponding to this version of this document is:

> ```
> <Documentation>CAx-IF Rec.Pracs.---AP242 Domain Model XML
>         Kinematics---1.1---2022-11-18</Documentation>
> ```

General Postprocessor Recommendation:

If a postprocessor encounters attribute values, or object instantiations different from the ones recommended in this version of the document, a warning shall be recorded. In such case, an additional exchange agreement is supposed to be in place among the parties involved in the data exchange.

4 Kinematics

The scope of this document is to describe the semantics of kinematic constraints between parts.

To describe the kinematic geometric constraints (without a movie), the following templates are involved:

- Curve/Surface

- KinematicLink

- KinematicPair

- Mechanism

- Import Mechanism

- Kinematic Dressup
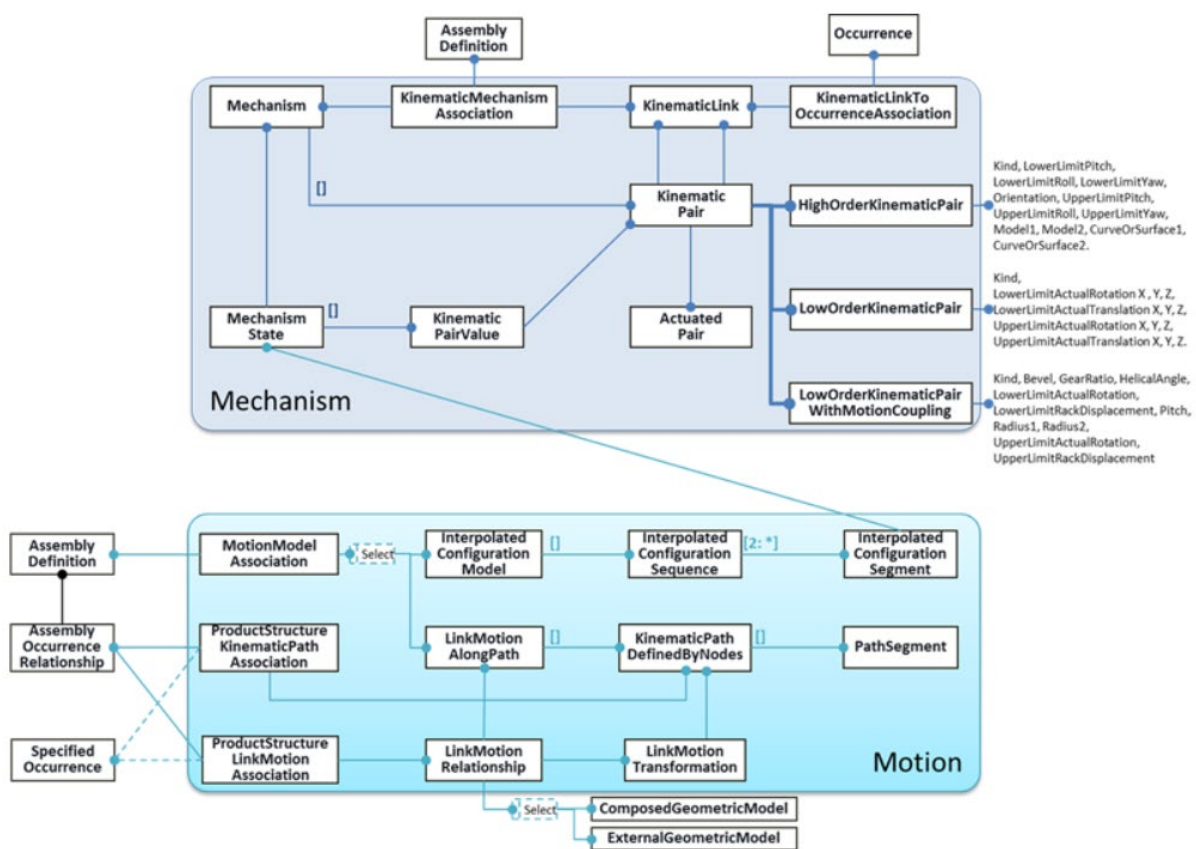
Figure 1: Overview of Kinematic Capabilities in AP242 Ed.2

Kinematic motions may be described in two ways:

- With a number of discrete snapshots, each one defining the position of the parts. Such a motion is computed out of a fully defined motion based on the kinematic geometric constraints. This way, the exact position of the parts in between may not be defined. This involves the following templates:
  - o LinkMotionAlongPath
  - o KinematicPathDefinedByNodes

- o PathSegment

- o LinkMotionRelationship

- o Simplified Kinematic Positioning Representation

- o Full Kinematic Positioning Representation

- Fully defined based on the kinematic geometrical constraints. This is currently out of scope of this recommended practices document.

For more information about the semantics, refer to the STEP Part 105 Application Integrated Resources for Kinematics, which is the basis for this area of the AP242 Domain Model.

4.1 Template "Curve/Surface"

The Curve/Surface template supports the ability to uniquely identify a Curve or a Surface within a DigitalFile.

Figure 2: Template "Curve"

Figure 3: Template "Surface"

The Instance Model: STEP exchange file format (ISO10303 AP242 Domain Model XML syntax)

```
  <RepresentationContext uid="ccs--origin" xsi:type="n0:GeometricCoordi-
nateSpace">
    <Id id="/NULL"/>
    <Items>
      <RepresentationItem xsi:type="n0:HighOrderKinematicPair" uid="kin--
0000000004634060--kpair">
…
        <CurveOrSurface1 uidRef="cv--0000000006D3B150--ei"/>
..
      </RepresentationItem>
      <RepresentationItem xsi:type="n0:Curve" uid="cv--0000000006D3B150--ei">
        <External uid="idal--0000000006D3B150">
          <Id id="/NULL"/>
          <Source>
            <DigitalFile uidRef="df--0000000006D3B150"/>
          </Source>
        </External>
      </RepresentationItem>
    </Items>
    <DimensionCount>3</DimensionCount>
  </RepresentationContext>
```

```xml
<File xsi :type="n0 :DigitalFile" uid="df--0000000006D3B150">
…
  <Id>
    <Identifier uid="dfid--0000000006D3B150--13" id="Kulisse_kinema-
tic_curve.jt" idContextRef="org--0xbc66cf8"/>
  </Id>
  <Locations>
    <ExternalItem uid="idal--0000000006D3B150--ei">
      <Id>
        <Identifier uid="idal--0000000006D3B150--13" id="Kulisse_kine-
matic_curve.jt" idContextRef="org--0xbc66cf8"/>
      </Id>
      <Source>
        <IdentifierString>.\</IdentifierString>
      </Source>
    </ExternalItem>
  </Locations>
</File>
```

Entity Curve/Surface attributes	Attribute type
External	ExternalItem
Name	OPTIONAL DescriptorSelect
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment

Table 2: "Curve/Surface" Attributes

Attribute recommendations

- **External**: the reference to an external item (Id shall contain a unique identifier of the curve within the external item; Source shall reference a DigitalFile where the Curve can be found under the given Id). Use "DigitalFile" template (see [242-PAS] 9.1).

- **Name**: the words or set of words by which the Curve or Surface is known. The value of this attribute need not be specified. Use "Description" template (see [242-PAS] 4.6.7).

Preprocessor Recommendation:

If the referenced external item contains only the given curve or surface, the attribute 'name' may be left empty and the attribute External.Id may be set to '/NULL'. Otherwise, the value of 'name' and External.Id shall be identical and enable to identify uniquely the Curve/Surface within the CAD model

Postprocessor Recommendation:

If the attribute External.Id='/NULL', the referenced external item contains only the given curve or surface.

Remark: depending on the legacy CAD system. There are several ways to identify the Curve/Surface => shall be agreed between the data exchange partners.

Remark: the ExternalItem used for Curve/Surface.External shall not be used as Digital-File.Locations. This is functionally wrong and is not supported by the XSD.

4.2 Template "KinematicLink"

A KinematicLink positions a part/assembly within a kinematic mechanism.
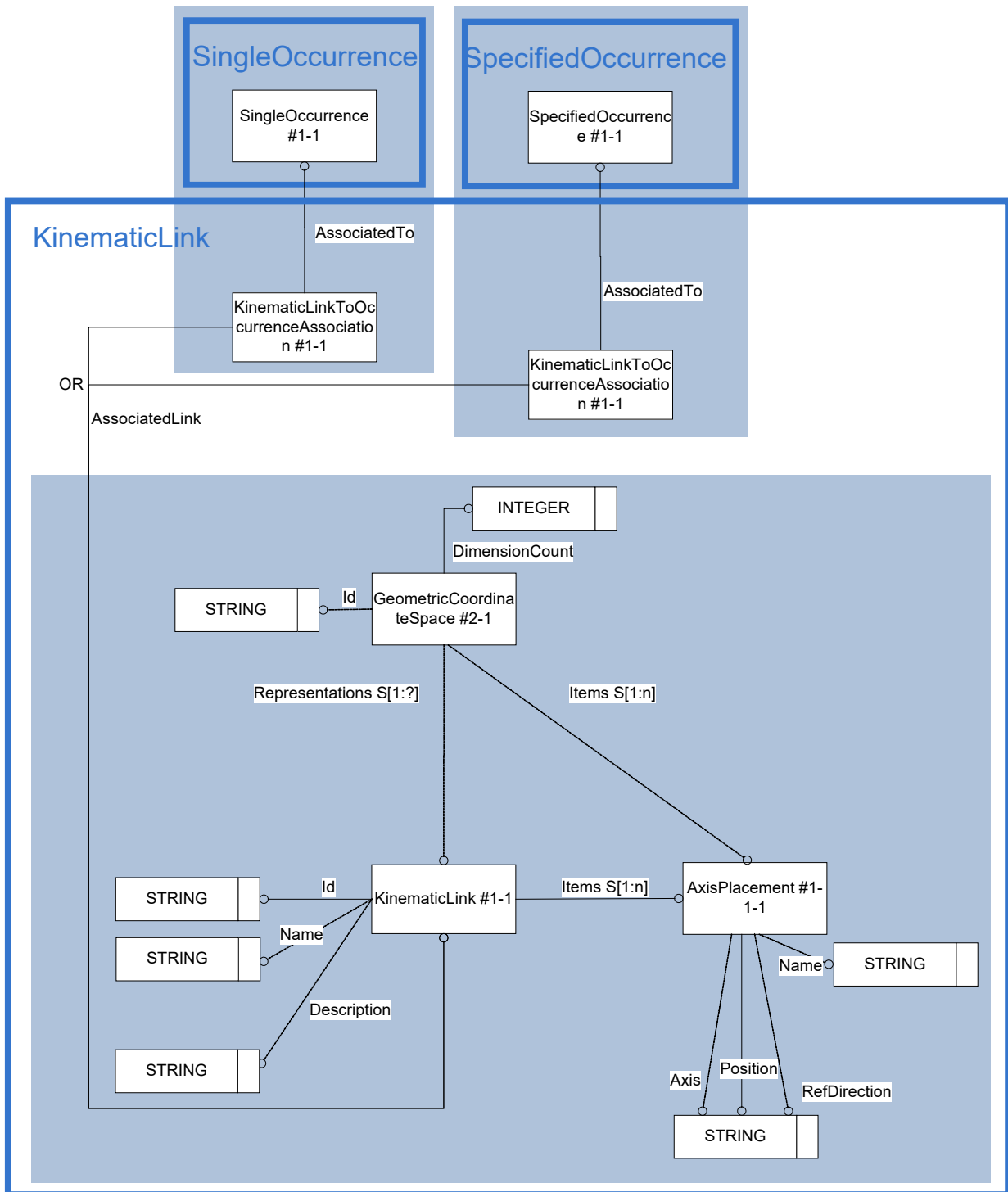
Figure 4: Template "KinematicLink"

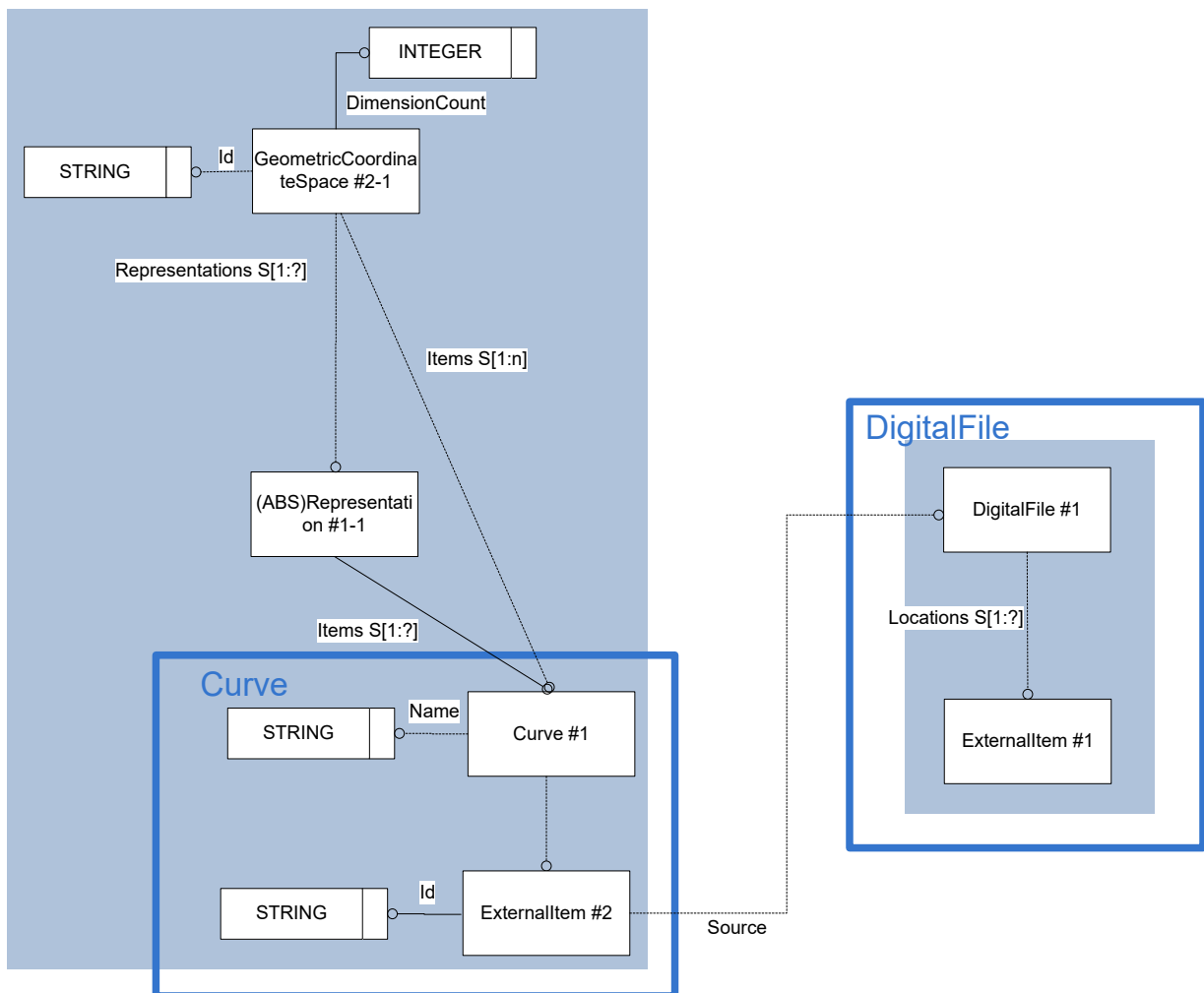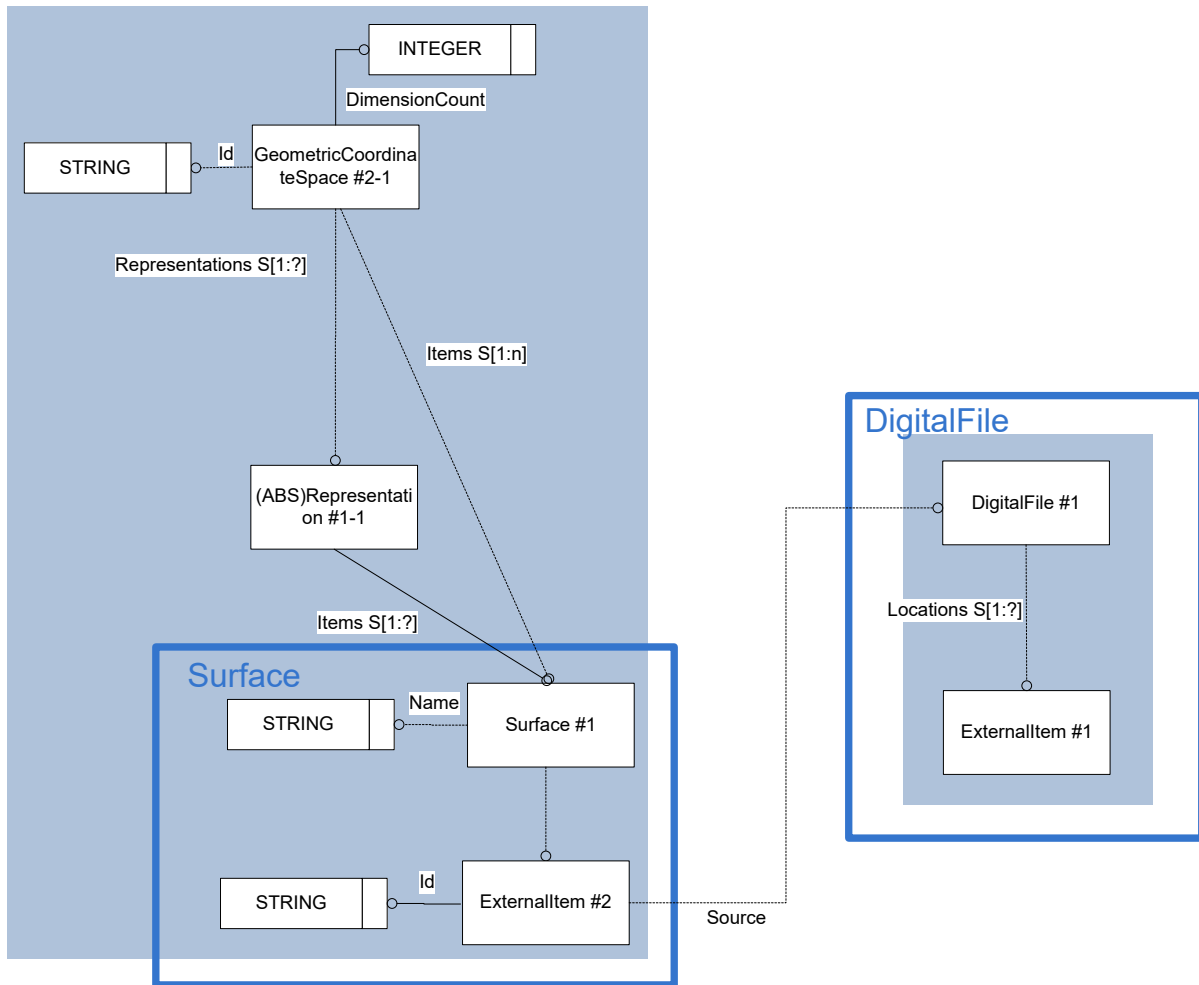## The Instance Model: STEP exchange file format (ISO10303 AP242 Domain Model XML syntax)

```xml
<RepresentationContext uid="ccs--origin" xsi:type="n0:GeometricCoordi-
nateSpace">
  <Id id="/NULL"/>
  <Representations>
    <Representation xsi:type="n0:KinematicLink" uid="kin--0000000004633D60--
kpair--klink--1">
      <Id id="/NULL"/>
      <Items>
        <RepresentationItem uidRef="kin--0000000004633D60--kpair--kframe--
0"/>
      </Items>
    </Representation>
  <Representations>
  <Items>
    <RepresentationItem uid="kin--0000000004633D60--kpair--kframe--0"
xsi:type="n0:AxisPlacement">
      <Axis>-1.0000000000,0.0000000000,0.0000000000</Axis>
      <Position>0.0000000000,-5.0000000000,15.0000000000</Position>
      <RefDirection>0.0000000000,-1.0000000000,0.0000000000</RefDirection>
    </RepresentationItem>
  </Items>
  <DimensionCount>3</DimensionCount>
</RepresentationContext>

<Part uid="p--00000000129ACFC0">
  <Id>
    <Identifier uid="pid--00000000129ACFC0--id3" id="Gestell" idCon-
textRef="org--0xbc66cf8"/>
  </Id>
…
  <Versions>
    <PartVersion uid="pv--00000000129ACFC0--id3">
…
      <Views>
        <PartView xsi:type="n0:AssemblyDefinition" uid="pvv--
00000000129ACFC0--id3">
…
          <Occurrence xsi:type="n0:SingleOccurrence" uid="pi--
00000000129ACFC0--9">
            <Id id="Gestell"/>
            <KinematicLinkToOccurrenceAssociation uid="kin--0000000004633D60-
-kpair--klink--0--ltpo" xsi:type="n0:KinematicLinkToOccurrenceAssociation">
              <AssociatedLink uidRef="kin--0000000004633D60--kpair--kframe--
0"/>
            </KinematicLinkToOccurrenceAssociation>
          </Occurrence>
…..
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
```

Entity KinematicLink attributes	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	Id
Items	SET[1:?] of RepresentationItem
Name	OPTIONAL DescriptorSelect
RepresentationTypes	OPTIONAL SET[1:?] of ClassSelect
VersionId	OPTIONAL Id
RepresentationRelationship	OPTIONAL SET[1:?] of RepresentationRelationship
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DatetimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
OfModel	OPTIONAL ExternalGeometricModel

Table 3: "KinematicLink" Attributes

Attribute recommendations

- **Description**: the reason for the creation of the KinematicLink. The value of this attribute need not be specified. Use "Description" template (see [242-PAS] 4.6.7).

- **Id**: the identifier or set of identifiers for the KinematicLink. Use "Identifier" template (see [242-PAS]).

- **Items**: the set of AxisPlacements that can be used to establish a contact frame for KinematicPairs.

- **Name**: the words or set of words by which the KinematicLink is known. The value of this attribute need not be specified. Use "Description" template (see [242-PAS] 4.6.7).

- **PropertyValueAssignment**: to assign a PropertyValue to the KinematicLink. Use the "PropertyAssignment" template (see [242-PAS] 6.2).

- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

Remark: the usage of OfModel is not recommended, since it would have to reference the same ExternalGeometricModel as on Occurrence or PartView level (via DefiningGeometry).

Preprocessor Recommendations:

- Each AxisPlacement referenced by "Items" is the absolute placement of one occurrence of the part within the Part where the Mechanism ist defined. It is computed (by the sender system) out of its relative placement in the static assembly structure.

- SpecifiedOccurrences are mostly not supported by the PDM systems → currently out of scope

- A KinematicLink shall be referenced by one and only one Part Occurrence
- A Part Occurrence shall be referenced (for a given Mechanism) by at most (optional) one KinematicLink (via one instance of KinematicLinkToOccurrenceAssociation embedded into the Occurrence object, see [242-PAS] 7.1).

  An issue has been created for this purpose on ISO Jira as TCSC410303-817.

This implies: if an occurrence of a part is involved in many KinematicPairs, (for example the elbow (on the one end) and the shoulder (on the other end) kinematics of the human upper arm), an instance of AxisPlacement shall be created for each of both and all referenced via KinematicLink.Items.In very special cases (for example a clock having 3 KinematicPairs frame/hour hand, frame/minute hand and frame/second hand), the same AxisPlacement may be reused for multiple KinematicPairs (here the AxisPlacement of the clock frame).

- A KinematicLink shall be referenced by at least one KinematikPair

- An AxisPlacement defined within KinematicLink.Items[] shall be referenced by at least one KinematikPair

Entity KinematicLinkToOccurrenceAssociation attributes	Attribute type
AssociatedLink	KinematicLink

Table 4: "KinematicLinkToOccurrenceAssociation" Attributes

Attribute recommendations

- **AssociatedLink**: the KinematicLink.

Remarks:

- When the Occurrence is an assembly, the occurrence path from the occurrence of the single part where the AxisPlacement has been computed is currently lost. This would enable to recreate the KinematicLink in the right occurrence of the sinle part in the target system. Otherwise 'any' part under the moving assembly has to be taken… This is not a critical issue, just enables a better mapping.
  This does not affect the fact that the assembly node is used to define the KinematikLink in the target system, just the place where to store the AxisPlacement (some CAD systems do not support to create AxisPlacements in assembly nodes).
  => an issue has been created in ISO Jira as BS10303-6167 to add an optional attribute 'Substructure' of type SpecificOccurrence to KinematicLinkToOccurrenceAssociation, analog to the attribute Substructure in ProductStructureKinematicPathAssociation and ProductStructureLinkMotionAssociation.

- As stated in Part 105: *"kinematic structures are represented by graphs where the links represent the vertices of the graph, and the joints represent the edges. The rigid objects with kinematic representations are defined kinematically in terms of links and geometrically in terms of their associated shape representation. All coordinate systems related to a kinematic link are founded in the link frame, which is the geometric context of the related link representation. These coordinate systems are called frames in this part of ISO 10303. For the purpose of representing the kinematic aspect of a mechanism, the shape of the link is represented by the relative location and orientation of all its pair frames with respect to its link frame."*
  This means the original shape element from which the AxisPlacement has been computed is lost
  => in the target system, the AxisPlacement has to be created into the geometric model of the involved part. If this is unproblematic for pure CAx data exchange scenarios, it may cause trouble for PDM data exchange especially if the target partner shall not edit the geometry but only simulate or work on the kinematics.

4.3 KinematicPair

A KinematicPair specifies a motion constraint (within a kinematic mechanism) between two adjacent KinematicLinks coinciding at a joint.

Before describing the different kinds of KinematicPairs, their common supertype and common features are introduced here under:

Entity KinematicPair attributes	Attribute type
External	OPTIONAL ExternalItem
Name	OPTIONAL DescriptorSelect
Actuation	OPTIONAL ActuatedPair
Link1	KinematicLink
Link2	KinematicLink
PairFrame1	AxisPlacement
PairFrame2	AxisPlacement
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment

Table 5: "KinematicPair" Attributes

Attribute recommendations

- **Name**: the words or set of words by which the KinematicPair is known. Use "Description" template (see [242-PAS] 4.6.7).

- **Actuation**: references an ActuatedPair. The value of this attribute need not be specified, especially if many parts are joined together in a chain => the actuation of one pair causes implicitly the movement of the other pairs (which do not need an own actuation).

- **Link1**: identifies the first KinematicLink.

- **Link2**: identifies the second KinematicLink.

- **PairFrame1**: identifies the first AxisPlacement from the first KinematicLink that coincides at the joint.

- **PairFrame2**: identifies the second AxisPlacement from the second KinematicLink that coincides at the joint.

- **PropertyValueAssignment**: to assign a PropertyValue to the KinematicPair. Use the "PropertyAssignment" template (see [242-PAS] 6.2).

- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

Preprocessor Recommendation:

- Name shall be always set, since a kinematic pair has always a name in the CAD system and this name is relevant for communication between partners.

- The name of the KinematicPair shall be unique within all KinematicPairs of a Mechanism.

- Link1 and Link2 shall not reference the same instance of KinematicLink.

- Usually, an instance of KinematicPair shall be referenced by one and only one Mechanism. In special cases (where multiple Mechanisms are defined on the same product structure (for example several local Mechanisms and an overall Mechanism), the same KinematicPair may be referenced by several Mechanisms.

- The combination of Link1 and Link2 (even of Link2 and Link1) shall be unique over all instances of KinematicPair

- The choice of having Link1 or Link2 for the one or the other part is significant if:
    o an Actuation (for all kind of Pairs if the direction of the actuation is not 'bidirectional')
    o Lower/UpperLimits (for the Low/High/WithMotionCouplingPairs if the lower and upper limits have a different absolute value)
    o or GearRatio/HelicalAngle/Pitch (for the WithMotionCouplingPairs)

  are defined.
  In such a case, these values always apply to the movement of Link2 relatively to Link1.

- PairFrame1 shall reference one of the AxisPlacements defined in KinematicLink for Link1
- PairFrame2 shall reference one of the AxisPlacements defined in KinematicLink for Link2

An ActuatedPair is the indication, along/around which axis the initial movement of Link2 relatively to Link1 shall occur.

Entity ActuatedPair attributes	Attribute type
Name	DescriptorSelect
Rx	OPTIONAL ActuatedDirectionEnum
Ry	OPTIONAL ActuatedDirectionEnum
Rz	OPTIONAL ActuatedDirectionEnum
Tx	OPTIONAL ActuatedDirectionEnum
Ty	OPTIONAL ActuatedDirectionEnum
Tz	OPTIONAL ActuatedDirectionEnum

Table 6: "ActuatedPair" Attributes

Attribute recommendations

- Depending on the Kind of kinematic pair (see below), one or many of the following attributes may be specified:
    - **Name:** the words or set of words by which the Actuation is known. Use "Description" template (see [242-PAS] 4.6.7). *This is the name of the command (in CATIA) resp. called driver (in NX).*
    - **Rx**: indicates that the actuation is around the x-axis. The value of this attribute need not be specified.
    - **Ry**: indicates that the actuation is around the y-axis. The value of this attribute need not be specified.
    - **Rz**: indicates that the actuation is around the z-axis. The value of this attribute need not be specified.
    - **Tx**: indicates that the actuation is along the x-axis. The value of this attribute need not be specified.
    - **Ty**: indicates that the actuation is along the y-axis. The value of this attribute need not be specified.
    - **Tz**: indicates that the actuation is along the z-axis. The value of this attribute need not be specified.
- Where applicable, the following values shall be used:

Enumeration value	Explanation
Bidirectional	the actuation is in both directions, positive and negative
positive_only	the actuation is in positive direction only
negative_only	the actuation is in negative direction only
not_actuated	the translation or rotation axis is not actuated

Preprocessor Recommendation:

- If not actuated along a given axle, the actuation should be left empty (rather than use the value 'not_actuated'.
- An ActuatedPair shall have at least one actuation around or along X, Y or Z.

Remarks:

- Even if both a rotational actuation and a translational actuation may be given (for example for a cylindrical_pair), most CAD systems can only actuate one of both at a time => the other actuation attributes shall be unset.
- Depending on the legacy system, not all kinds of actuation are supported. For example, CATIA only supports 'bidirectional' (default) and the AxisPlacements are calculated in such a way, that only Rz and Tz are needed.
- The name of the ActuatedPair shall be unique within all KinematicPairs of a Mechanism.

In the following sections, the subtypes of KinematicPair are described.

According to Part 105: low order pairs can be represented as sliding high order pairs. If a pair can be represented as a low order pair, it should be so represented and not as a high order pair.

4.3.1 Template "LowOrderKinematicPair"

For a LowOrderKinematicLink, the motion is constrained by simple translation or rotational parameters.

Preprocessor Recommendations:

- The use of LowOrderKinematicLink shall be done whenever possible, rather than the other subtypes of KinematicPair (used only for the more complex cases).
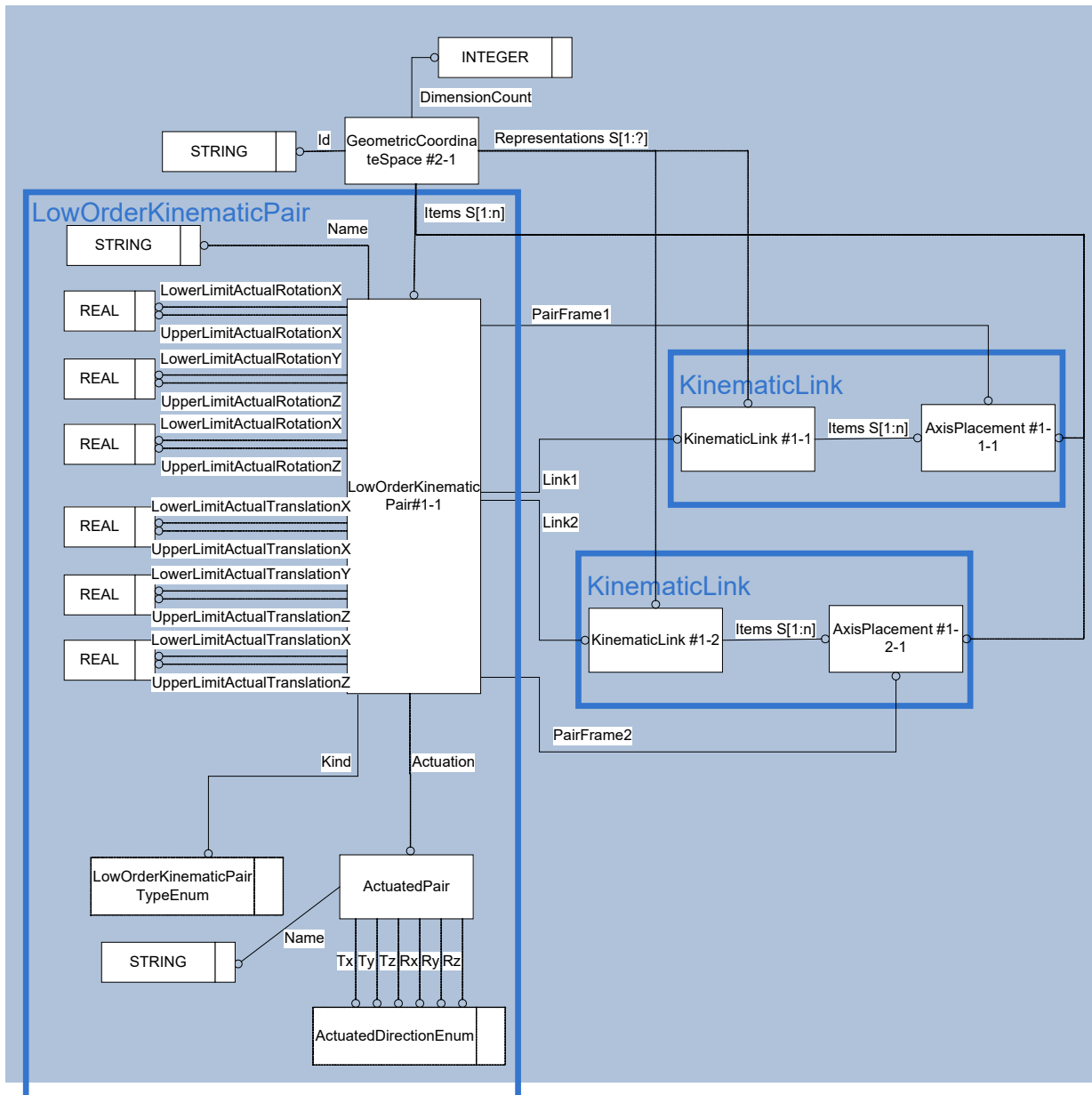

Figure 5: Template "LowOrderKinematicPair"

The Instance Model: STEP exchange file format (ISO10303 AP242 Domain Model XML syntax)

```xml
<RepresentationContext uid="ccs--origin" xsi:type="n0:GeometricCoordi-
nateSpace">
  <Id id="/NULL"/>
  <Representations>
    <Representation xsi:type="n0:Mechanism" uid="kin--0000000005F89EE0--m">
…
    <Items>
…
      <RepresentationItem uidRef="kin--0000000004634660--kpair"/>
</Items>
…
    </Representation>
  <Representations>
  <Items>
    <RepresentationItem xsi:type="n0:LowOrderKinematicPair" uid="kin--
0000000004634660--kpair">
      <Name>
          <CharacterString>Plunger-Disc-Spring</CharacterString>
      </Name>
      <Actuation uid="act--kin--0000000004634660--kpair">
        <Name>
          <CharacterString>command #1</CharacterString>
        </Name>
        <Tz>bidirectional</Tz>
      </Actuation>
      <Link1 uidRef="kin--0000000004634660--kpair--klink--1"/>
      <Link2 uidRef="kin--0000000004634660--kpair--klink--2"/>
      <PairFrame1 uidRef="kin--0000000004634660--kpair--kframe--0"/>
      <PairFrame2 uidRef="kin--0000000004634660--kpair--kframe--1"/>
      <Kind>prismatic_pair</Kind>
      <LowerLimitActualTranslationZ>-100.0000000000</LowerLimitActualTransla-
tionZ>
      <UpperLimitActualTranslationZ>100.0000000000</UpperLimitActualTransla-
tionZ>
    </RepresentationItem>
  </Items>
  <DimensionCount>3</DimensionCount>
</RepresentationContext>
```
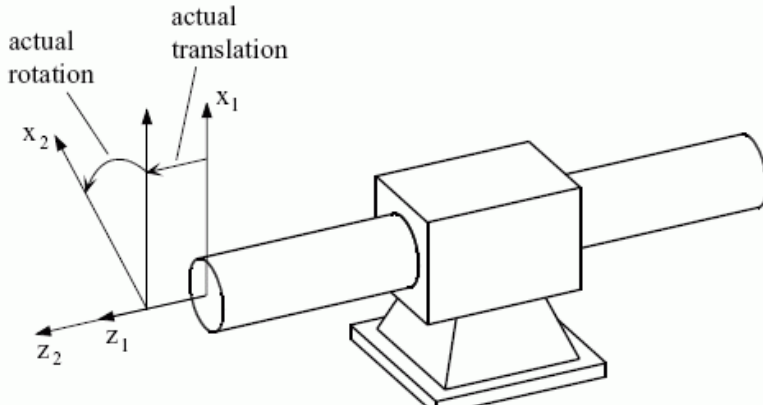
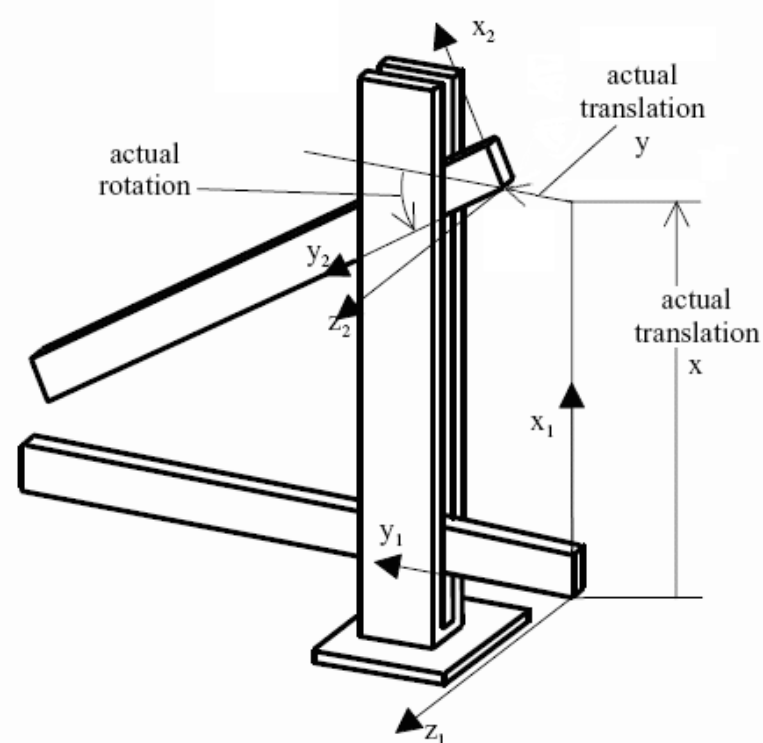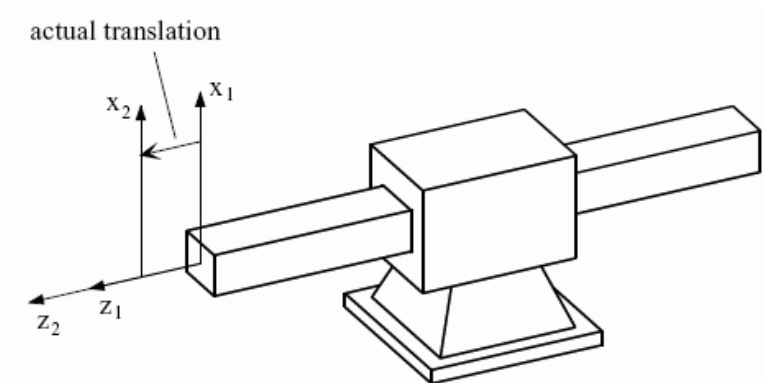| Entity LowOrderKinematicPair attributes	Attribute type
External	OPTIONAL ExternalItem
Name	OPTIONAL DescriptorSelect
Actuation	OPTIONAL ActuatedPair
Link1	KinematicLink
Link2	KinematicLink
PairFrame1	AxisPlacement
PairFrame2	AxisPlacement
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
Kind	LowOrderKinematicPairTypeEnum
LowerLimitActualRotationX	OPTIONAL REAL

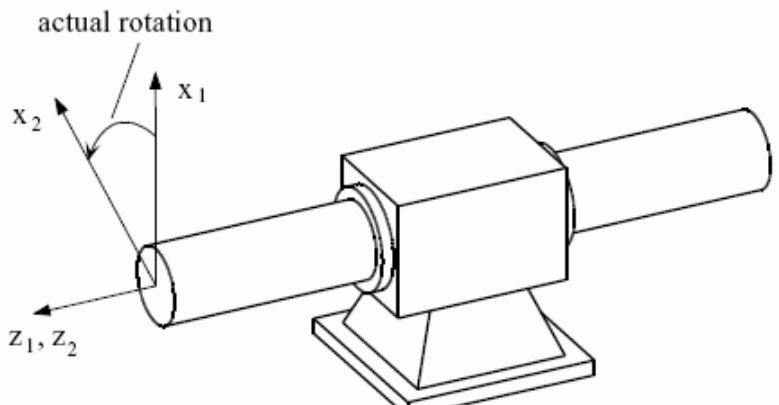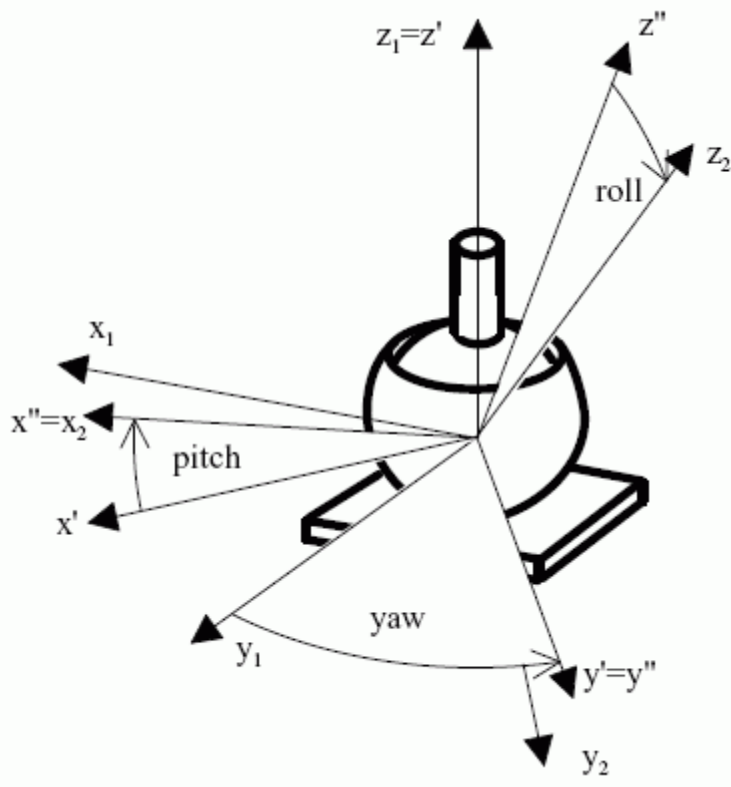| Entity LowOrderKinematicPair attributes	Attribute type
LowerLimitActualRotationY	OPTIONAL REAL
LowerLimitActualRotationZ	OPTIONAL REAL
LowerLimitActualTranslationX	OPTIONAL REAL
LowerLimitActualTranslationY	OPTIONAL REAL
LowerLimitActualTranslationZ	OPTIONAL REAL
UpperLimitActualRotationX	OPTIONAL REAL
UpperLimitActualRotationY	OPTIONAL REAL
UpperLimitActualRotationZ	OPTIONAL REAL
UpperLimitActualTranslationX	OPTIONAL REAL
UpperLimitActualTranslationY	OPTIONAL REAL
UpperLimitActualTranslationZ	OPTIONAL REAL

Table 7: "LowOrderKinematicPair" Attributes
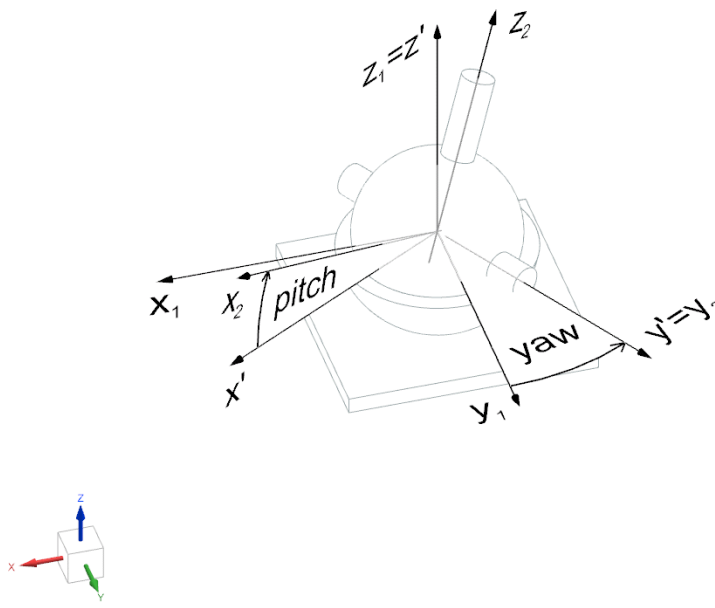
Attribute recommendations

- **Kind**: the kind of pair. Where applicable, the following values shall be used:
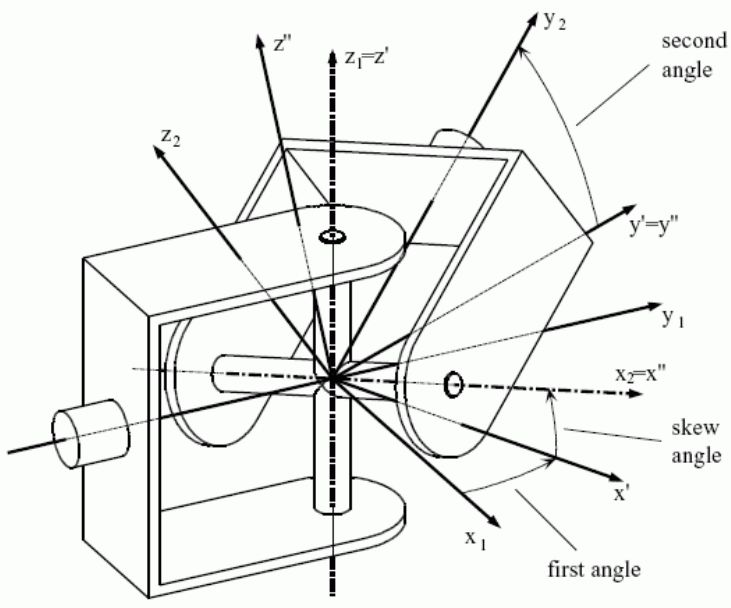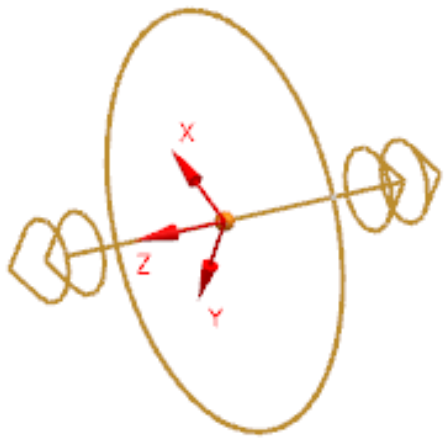
Enumeration value	Explanation
cylindrical_pair	a pair that constrains the motion between two adjacent links to a translation along a common axis and a rotation about it. Remark: both a rotational and a translational actuation may be specified. According to Part 105, the z-axes shall coincide, and the positive directions shall agree; both Rotation and Translation Lower/UpperLimits may be specified (around/along the Z axis)
fully_con-strained_pair ('fixed' in NX, 'rigid' in CATIA)	a pair that prevents any relative motion between two adjacent links. The two pair frames are always forced to coincide. Remark: no actuation may be specified. According to Part 105, no Lower/UpperLimit may be specified
planar_pair	a pair that constrains the motion between two adjacent links to translations along the x-axis and y-axis and a rotation about the z-axis. To measure the translations and the angle of rotation, a frame is defined on each of the links such that the x-, y-, and z-axes coincide, and the positive directions agree. The translations in x- and y-directions are defined as the displacements required to translate the origin of the first pair frame in positive direction of the x- and y-axes

until it coincides with the origin of the second pair frame. The angle of rotation is defined as the angle required to rotate the x-axis of the first pair frame in positive direction around the common z-axis until its direction coincides with the direction of the x-axis of the second pair frame.

Remark: no actuation may be specified. According to Part 105, both rotation and Translation Lower/UpperLimits may be specified: around the Z axis (Rotation) and along the X and Y axis (Translation) (not supported by NX)

prismatic_pair ('slider' in NX)	a pair that constrains the motion between two adjacent links to a translation along a common axis. To measure the distance of translation a frame is defined on each of the links such that their corresponding coordinate axes coincide and their positive directions agree. The motion of the second link with respect to the first link is defined as the distance required to translate the xy-plane of the first pair frame in positive direction of the common z-axis until it coincides with the xy-plane of the second pair frame.

Remark: a translational actuation may be specified along the Z axis. According to Part 105, a Translation Lower/UpperLimit may be specified along the Z axis

revolute_pair	a pair that constrains the motion between two adjacent links to a rotation about a common axis. To measure the angle of rotation a frame is defined on each of the links such that local origins and the z-axes coincide, and their positive directions agree. The motion of the second link with respect to the first link is defined as the angle required to rotate the x-axis of the first pair frame in positive direction around the common z-axis until it matches the x-axis of the second pair frame. Remark: a rotational actuation may be specified around the Z axis. According to Part 105, a Rotation Lower/UpperLimit may be specified around the Z axis

spherical_pair	a pair that constrains the motion between two adjacent links to the rotation about three independent axes that intersect in a common point. To measure the three angles of rotation, a frame system is defined on each of the links such that the origins of the frames coincide. The three angles of rotation are defined as the yaw, pitch, and roll angles. They are required to rotate the x-, y-, and z-axes of the first pair frame until the axis directions of this frame coincide with the axis directions of the second pair frame.
	Remark: no actuation may be specified. According to Part 105, Rotation Lower/UpperLimits may be specified around all 3 axis (not supported by NX)

spheri-cal_pair_with_pin *(not supported by CATIA nor NX)*	 a pair that constrains the motion between two adjacent links to the rotation about two independent axes that intersect in a common point. It can be compared to the human elbows: the joint between upper arm and forearm. The forearm can rotate around the direction of the forearm and it can also rotate towards the upper arm. To measure the two angles of rotation, a frame system is defined on each of the links such that the origins of the frames coincide. The two angles of rotation are defined as the yaw and pitch angles. They are required to rotate the y-, and z-axes of the first pair frame until the axis directions of this frame coincide with the axis directions of the second pair frame. Remark: actuation may be specified in rotation around Y- and Z-axis. According to Part 105, Rotation Lower/UpperLimits may be specified around the Y and Z axis
Unconstrained_pair *(not supported by NX and CATIA)*	a pair that does not constrain the relative motion between two adjacent links. The **unconstrained_pair** may be helpful in early design stages or for kinematic analysis systems, because it allows to eliminate the constraints between two links without changing the topological structure. Remark: no actuation may be specified. According to Part 105, Lower/UpperLimit may be specified around/along all axes

universal_pair	a pair that constrains the motion between two adjacent links to two rotations about two intersecting axes.
	Remark: actuation may be specified in rotation around X- and Z-axis. According to Part 105, a Rotation Lower/UpperLimit may be specified around the X and Z axis (not supported by CATIA and NX)
Homokinetic_pair *('constant velocity' in NX, but **not** 'CV joint' in CATIA (icon is defined as 2 universal pairs having parallel ends))*	A homokinetic_pair is a type of universal_pair, where the rotation of input axis and output axis are uniform.
	Remark: actuation may be specified in rotation around X- and Z-axis. According to Part 105, a Rotation Lower/UpperLimit may be specified around the X and Z axis (not supported by CATIA)

- Depending on the Kind of kinematic pair, one or many of the following attributes pairs (lower/upper for X, Y or Z) can be set at a time (Lower or Upper have both to be set).

    o **LowerLimitActualRotationX**: the minimum value of rotation around the x-axis. The value of this attribute need not be specified.

    o **UpperLimitActualRotationX**: the maximum value of rotation around the x-axis. The value of this attribute need not be specified.

    o **LowerLimitActualRotationY**: the minimum value of rotation around the y-axis. The value of this attribute need not be specified.

    o **UpperLimitActualRotationY**: the maximum value of rotation around the y-axis. The value of this attribute need not be specified.

    o **LowerLimitActualRotationZ**: the minimum value of rotation around the z-axis. The value of this attribute need not be specified.

    o **UpperLimitActualRotationZ**: the maximum value of rotation around the z-axis. The value of this attribute need not be specified.

    o **LowerLimitActualTranslationX**: the minimum value of translation along the x-axis. The value of this attribute need not be specified.

    o **UpperLimitActualTranslationX**: the maximum value of translation along the x-axis. The value of this attribute need not be specified.

    o **LowerLimitActualTranslationY**: the minimum value of translation along the y-axis. The value of this attribute need not be specified.

    o **UpperLimitActualTranslationY**: the maximum value of translation along the y-axis. The value of this attribute need not be specified.

    o **LowerLimitActualTranslationZ**: the minimum value of translation along the z-axis. The value of this attribute need not be specified.

    o **UpperLimitActualTranslationZ**: the maximum value of translation along the z-axis. The value of this attribute need not be specified.

Preprocessor Recommendations:

- The value in **Lower…** shall be smaller than the value in **Upper…**.
- If a value is given in Lower…, the corresponding Upper… attribute shall also have a value.
- Both KinematicLinks of a LowOrderKinematicPair have to be defined within the same GeometricCoordinateSpace than the Mechanism itself.
- The same GeometricCoordinateSpace shall be used for the Mechanism and the "notional solids centroid" assembly validation property defined in [242-PAS] 13.1.2 of the assembly node on which the mechanism is defined.
- The values 0 for rotation and 0 for translation means that the axis placement of the kinematic links are the same.
- Multiple 360° are possible, negative values also => for example [-360, 1080]
- The CATIA CV joint shall be mapped as the combination of two universal pairs (as mapped internally by CATIA).

Postprocessor Recommendations:

- If the universal pair is not supported by the target CAD system (for example NX), it shall be imported as the combination of two cylindrical pairs (between the cross and both brackets of the universal pair)

Remark: depending on the legacy system, not all Kinds are supported, and not all Limits are supported. For example, from CATIA, the AxisPlacements are calculated in such a way, that only Lower/UpperLimitActualRotationZ and Lower/UpperLimitActualTranslationZ are needed.

4.3.2 Template "HighOrderKinematicPair"

For a HighOrderKinematicLink, the motion constraint requires references to one or more surfaces or curves in order to be defined cinematically.
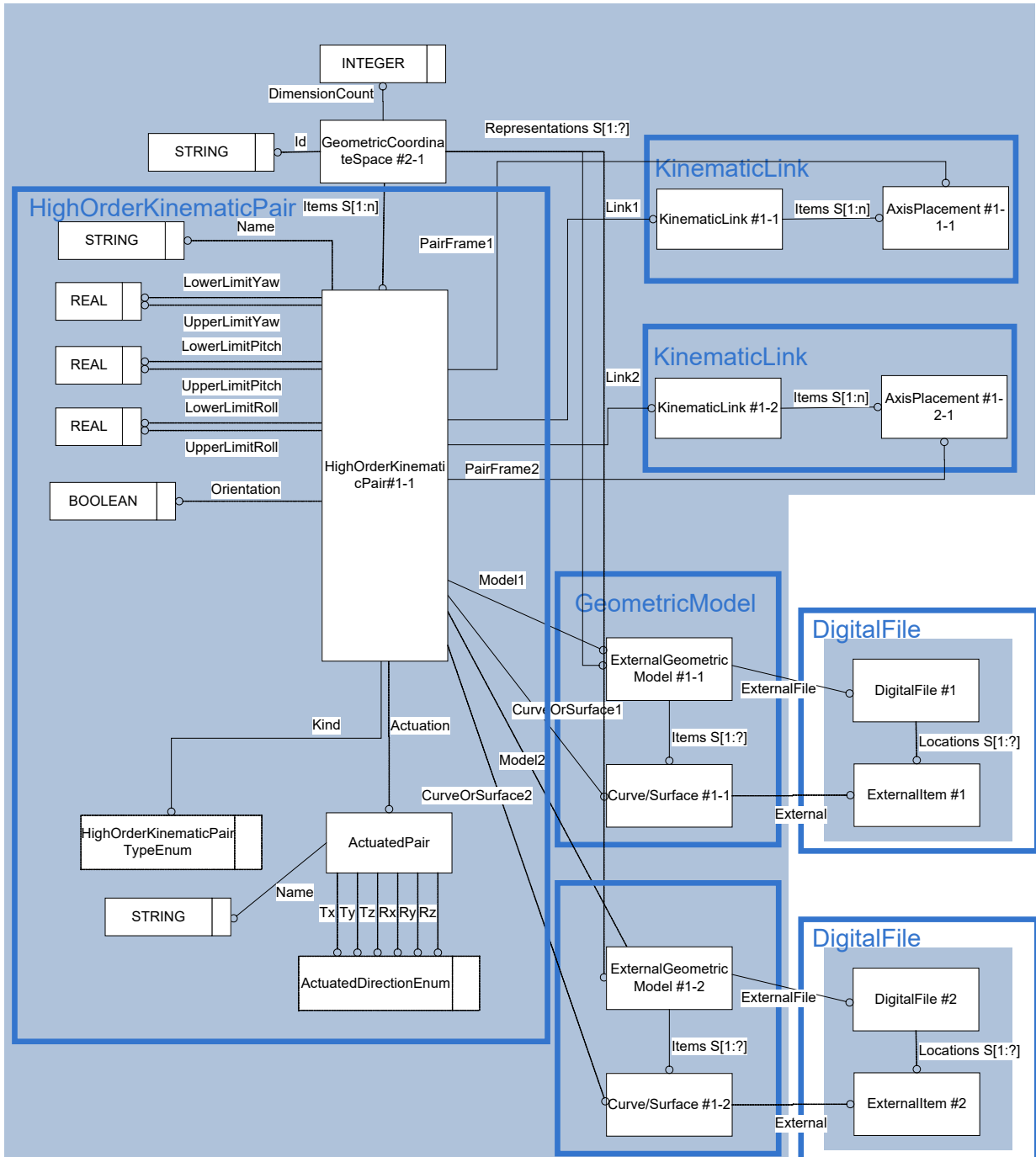
Figure 6: Template "HighOrderKinematicPair"

The Instance Model: STEP exchange file format (ISO10303 AP242 Domain Model XML syntax)

```xml
<RepresentationContext uid="ccs--origin" xsi:type="n0:GeometricCoordi-
nateSpace">
  <Id id="/NULL"/>
  <Representations>
    <Representation xsi:type="n0:Mechanism" uid="kin--0000000005FC5FC0--m">
…
    <Items>
      <RepresentationItem uidRef="kin--0000000004634060--kpair"/>
    </Items>
…
    </Representation>
  <Representations>
  <Items>
…
    <RepresentationItem xsi:type="n0:HighOrderKinematicPair" uid="kin--
0000000004634060--kpair">
      <Name>
          <CharacterString>Plunger-Disc-Spring</CharacterString>
      </Name>
      <Actuation uid="act--0000000006D3B150">
        <Name>
          <CharacterString>command #1</CharacterString>
        </Name>
        <Tz>bidirectional</Tz>
      </Actuation>
      <Link1 uidRef="kin--0000000004634060--kpair--klink--0"/>
      <Link2 uidRef="kin--0000000004634060--kpair--klink--1"/>
      <PairFrame1 uidRef="kin--0000000004634060--kpair--kframe--0"/>
      <PairFrame2 uidRef="kin--0000000004634060--kpair--kframe--1"/>
      <CurveOrSurface1 uidRef="cv--0000000006D3B150--ei"/>
      <Kind>point_on_planar_curve_pair</Kind>
      <Model1 uidRef="egm--0000000006D3B150"/>
    </RepresentationItem>
    <RepresentationItem xsi:type="n0:Curve" uid="cv--0000000006D3B150--ei">
      <External uid="idal--0000000006D3B150--ei2" xsi:type="n0:ExternalItem">
        <Id>
          <Identifier uid="idal--0000000006D3B150--132" id="#1234" idContex-
tRef="org--0xbc66cf8"/>
        </Id>
        <Source>
          <DigitalFile uidRef="df--0000000006D3B150"/>
        </Source>
      </External>
    </RepresentationItem>
…
  </Items>
  <DimensionCount>3</DimensionCount>
</RepresentationContext>
```

Entity HighOrderKinematicPair attributes	Attribute type
External	OPTIONAL ExternalItem
Name	OPTIONAL DescriptorSelect
Actuation	OPTIONAL ActuatedPair
Link1	KinematicLink
Link2	KinematicLink
PairFrame1	AxisPlacement
PairFrame2	AxisPlacement
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
CurveOrSurface1	ExternalCurveOrSurfaceSelect
CurveOrSurface2	OPTIONAL ExternalCurveOrSurfaceSelect
Kind	HighOrderKinematicPairTypeEnum
LowerLimitPitch	OPTIONAL REAL
LowerLimitRoll	OPTIONAL REAL
LowerLimitYaw	OPTIONAL REAL
Model1	ExternalGeometricModel
Model2	OPTIONAL ExternalGeometricModel
Orientation	OPTIONAL BOOLEAN
Range1	OPTIONAL ExternalTrimmedCurveOrSurfaceSelect
Range2	OPTIONAL ExternalTrimmedCurveOrSurfaceSelect
UpperLimitPitch	OPTIONAL REAL
UpperLimitRoll	OPTIONAL REAL
UpperLimitYaw	OPTIONAL REAL
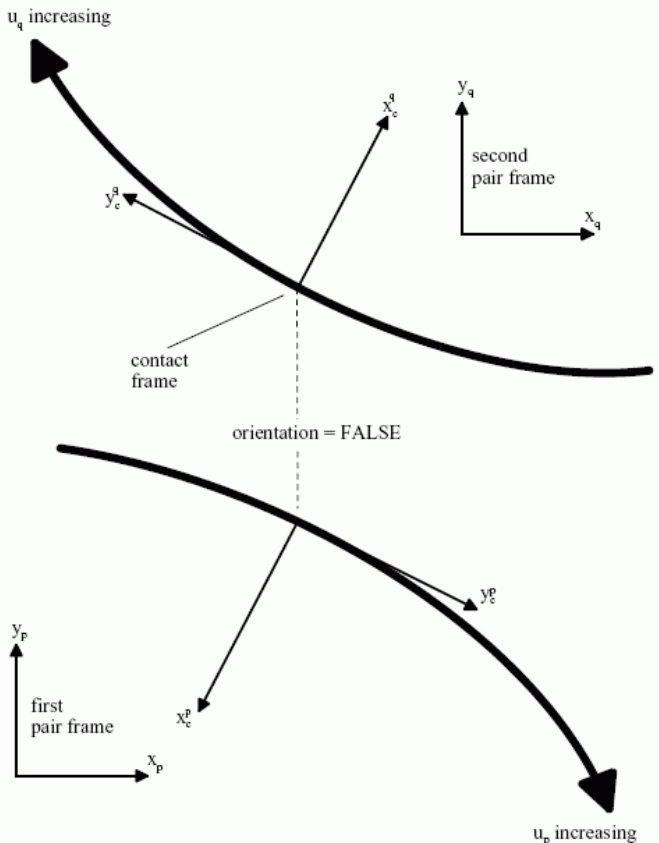
Table 8: "HighOrderKinematicPair" Attributes
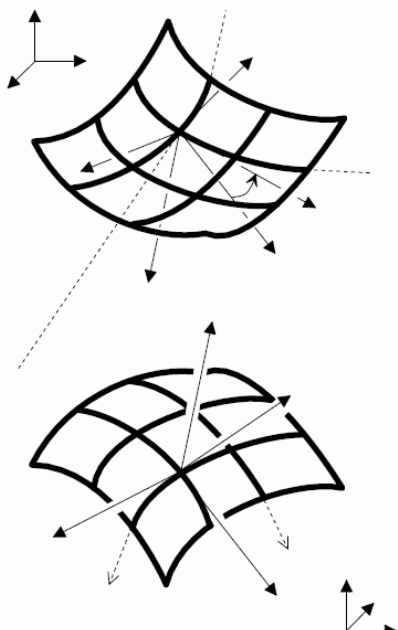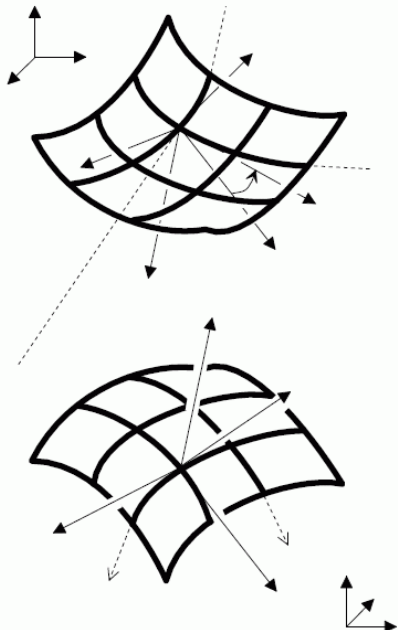
Attribute recommendations

- **CurveOrSurface1:** the first curve or surface that coincide at the joint. Use the "Curve/Surface" template (see 4.1)

- **CurveOrSurface2:** the second curve or surface that coincide at the joint. Use the "Curve/Surface" template (see 4.1). Depending on the value of 'Kind', the value of this attribute need not be specified.

- **Kind**: the kind of pair. Where applicable, the following values shall be used:

Enumeration value	Explanation
linear_flexible_and_pla-nar_curve_pair	the pair describes a linear flexible link that moves (slides) around a rotating curve where the rotation axis of the curve is perpendicular to the direction of the linear flexible link and the curve. *Remark*: Definition of Actuations needs to be clarified. According to Part 105, no lower/UpperLimits may be specified

point_on_surface_pair	the pair constrains the motion of two links such that a point defined on the second link always lies on a surface defined on the first link. The actual location of this point on the surface is called the contact point. The contact point specifies an origin of a frame called the contact frame. The local normal to the surface in the contact point coincides with the z-axis of the contact frame. The x-axis of the contact frame is parallel to the tangent to the iso-parameter line of the surface for the first surface parameter, which passes the contact point, and points into the direction where the first surface parameter increases.
	Remark: no actuation may be specified. According to Part 105, lower/UpperLimits may be specified on Yaw, Pitch and Roll (not supported by NX)
point_on_planar_curve_pair	a pair that constrains the motion of two links such that a point defined on the second link always lies on a planar curve defined on the first link. The actual location of this point on the curve is called the contact point.

	Remark: a rotational actuation may be specified. According to Part 105, lower/UpperLimits may be specified on Yaw, Pitch and Roll (not supported by NX)
planar_curve_pair	a pair that constrains the motion of two links along a planar curve on each of the links. Both curves lie in a common plane. Remark: no actuation may be specified. According to Part 105, no Lower/UpperLimit may be specified
rolling_curve_pair (not supported by NX)	a kind of a planar_curve_pair that specifies the rolling motion between the curves on the two links. Remark: a rotational actuation may be specified. According to Part 105, no Lower/UpperLimit may be specified Symbol in NX (curve on curve?):
sliding_curve_pair ('curve on curve' in NX)	a kind of a planar_curve_pair that specifies the sliding motion between the curves on the two links. Remark: no actuation may be specified. According to Part 105, no Lower/UpperLimit may be specified

rolling_surface_pair (not supported in NX and CATIA)	a pair that specifies a rolling motion between the two contact surfaces on the two links. Remark: no actuation may be specified. According to Part 105, lower/UpperLimit may be specified around one axis
sliding_surface_pair (not supported in NX and CATIA)	a pair that specifies a sliding motion between the two contact surfaces on the two links. Remark: no actuation may be specified. According to Part 105, lower/UpperLimit may be specified around one axis

- Depending on the kind of kinematic pair, one or many of the following attribute pairs (lower/upper for roll, pitch or yaw) can be set at a time (Lower / Upper have both to be set).
    - **LowerLimitPitch**: the minimum value of the (Y) pitch angle. The value of this attribute need not be specified.
    - **UpperLimitPitch**: the maximum value of the (Y) pitch angle. The value of this attribute need not be specified.
    - **LowerLimitRoll**: the minimum value of the roll (Z) angle. The value of this attribute need not be specified.
    - **UpperLimitRoll**: the maximum value of the roll (Z) angle. The value of this attribute need not be specified.
    - **LowerLimitYaw**: the minimum value of the yaw (X) angle. The value of this attribute need not be specified.
    - **UpperLimitYaw**: the maximum value of the yaw (X) angle. The value of this attribute need not be specified.
- **Model1:** the ExternalGeometricModel that contains curveOrSurface1
- **Model2:** the ExternalGeometricModel that contains curveOrSurface2. Depending on the value of 'Kind', the value of this attribute need not be specified.
- **Orientation:** a flag indicating axis-directions agreements. The value depends on the kind:

    - 'linear_flexible_and_planar_curve_pair', 'planar_curve_pair', 'sliding_curve_pair', 'rolling_curve_pair': a flag indicating whether the x-axis-directions agree;
    - 'point_on_planar_curve_pair': a flag indicating whether the y-axis of the unrotated contact frame agrees with the z-axis of the pair frame on the first link;
    - 'rolling_surface_pair', 'sliding_surface_pair': an indication of whether the z-direction of the second surface agrees with the z-direction of the first surface.
- **Range1:** the admissible range for the positional pair parameter value on CurveOrSurface1. The value of this attribute need not be specified.
- **Range2:** the admissible range for the positional pair parameter value on CurveOrSurface2. Depending on the value of 'Kind', the value of this attribute may be specified.

Preprocessor Recommendations:

- In case only one Model/Curve/Surface is involved, it shall be mapped into Model1, CurveOrSurface1 and Range1.
- The value in **Lower…** shall be smaller than the value in **Upper…**.
- If a value is given in Lower…, the corresponding Upper… attribute shall also have a value.
- Both KinematicLinks of a HighOrderKinematicPair have to be defined within the same GeometricCoordinateSpace than the Mechanism itself. Dito for the CurveOrSurface1/2 and Model1/2.
- The CurveOrSurface1/2 may be defined in distinct GeometricModels or within the GeometricModel of one of the parts.

Remark: depending on the legacy system, not all Kinds are supported, and not all Limits/Ranges are supported.

4.3.3 Template "LowOrderKinematicPairWithMotionCoupling"

This is a type of KinematicPair whose motion is constrained by simple translation or rotation and by one or more geometric parameters. This kind of Kinematic pair is used for gear, linear flexible pinion, rack pinion and screw pairs.

Remarks:

- a LowOrderKinematicPairWithMotionCoupling corresponds to a Gear, Cable, Rack or Screw joint in CATIA and NX

Figure 7: Template "LowOrderKinematicPairWithMotionCoupling"

The Instance Model: STEP exchange file format (ISO10303 AP242 Domain Model XML syntax) for gear_pair

```
<RepresentationContext uid="ccs--origin" xsi:type="n0:GeometricCoordi-
nateSpace">
  <Id id="/NULL"/>
  <Representations>
    <Representation xsi:type="n0:Mechanism" uid="kin--000000000BB93740--m">
…
      <Items>
        <RepresentationItem uidRef="kin--0000000007EA8550--kpair"/>
      </Items>
      <Name>
            <CharacterString>Ventiltrieb</CharacterString>
      </Name>
…
    </Representation>
  <Representations>
  <Items>
…
      <RepresentationItem xsi:type="bom:LowOrderKinematicPairWithMotionCou-
pling" uid="kin--0000000007EA8550--kpair">
      <Name>
            <CharacterString>Plunger-Disc-Spring</CharacterString>
      </Name>
      <Actuation uid="act--kin--0000000007EA8550--kpair">
        <Name>
          <CharacterString>command #1</CharacterString>
        </Name>
        <Rz>bidirectional</Rz>
      </Actuation>
      <Link1 uidRef="kin--0000000007EA8550--kpair--klink--0"/>
      <Link2 uidRef="kin--0000000007EA8550--kpair--klink--1"/>
      <PairFrame1 uidRef="kin--0000000007EA8550--kpair--kframe--0"/>
      <PairFrame2 uidRef="kin--0000000007EA8550--kpair--kframe--1"/>
      <GearRatio>0.5000000000</GearRatio>
      <Kind>gear_pair</Kind>
      <LowerLimitActualRotation>0.0000000000</LowerLimitActualRotation>
      <UpperLimitActualRotation>360.0000000000</UpperLimitActualRotation>
    </RepresentationItem>
…
  </Items>
  <DimensionCount>3</DimensionCount>
</RepresentationContext>
```

The Instance Model: STEP exchange file format (ISO10303 AP242 Domain Model XML syntax) for rack_and_pinion_pair

```
<RepresentationContext uid="ccs--origin" xsi:type="n0:GeometricCoordi-
nateSpace">
  <Id id="/NULL"/>
  <Representations>
    <Representation xsi:type="n0:Mechanism" uid="kin--000000000C3F5180--m">
…
      <Items>
        <RepresentationItem uidRef"kin--0000000008018840--kpair"/>
      </Items>
      <Name>
            <CharacterString>Mechanism.1</CharacterString>
      </Name>
```

```
…
    </Representation>
  <Representations>
  <Items>
…
    <RepresentationItem xsi:type="bom:LowOrderKinematicPairWithMotionCou-
pling" uid="kin--0000000008018840--kpair">
      <Name>
          <CharacterString>Plunger-Disc-Spring</CharacterString>
      </Name>
      <Actuation uid="act--kin--0000000008018840--kpair">
        <Name>
          <CharacterString>command #1</CharacterString>
        </Name>
        <Tz>bidirectional</Tz>
      </Actuation>
      <Link1 uidRef="kin--0000000008018840--kpair--klink--0"/>
      <Link2 uidRef="kin--0000000008018840--kpair--klink--1"/>
      <PairFrame1 uidRef="kin--0000000008018840--kpair--kframe--0"/>
      <PairFrame2 uidRef="kin--0000000008018840--kpair--kframe--1"/>
      <GearRatio>15.0000000000</GearRatio>
      <Kind>rack_and_pinion_pair</Kind>
      <LowerLimitRackDisplacement>-100.000000000</LowerLimitRackDisplacement>
      <UpperLimitRackDisplacement>100.000000000</UpperLimitRackDisplacement>
    </RepresentationItem>
…
  </Items>
  <DimensionCount>3</DimensionCount>
</RepresentationContext>
```

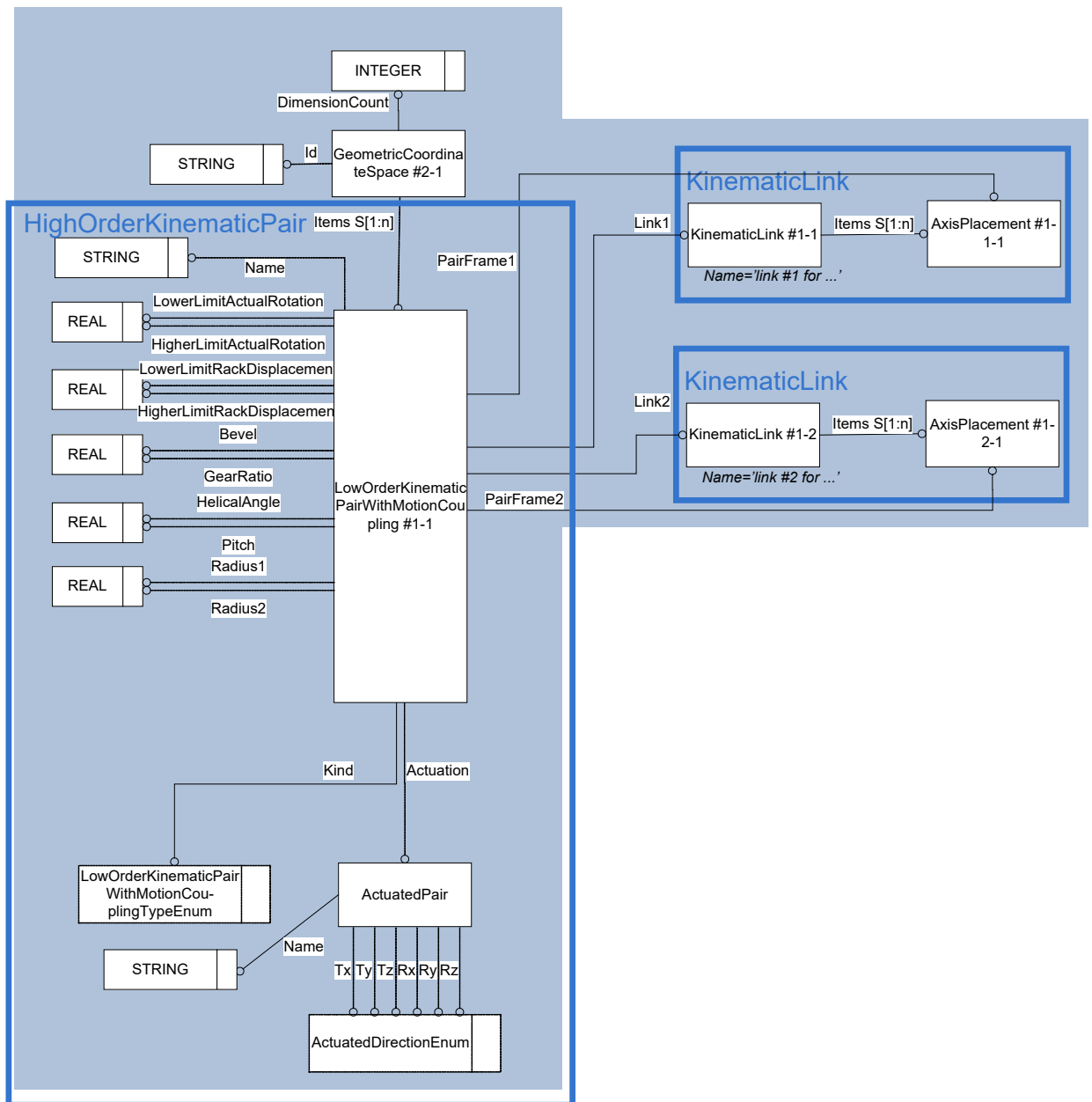## The Instance Model: STEP exchange file format (ISO10303 AP242 Domain Model XML syntax) for screw_pair

```
<RepresentationContext uid="ccs--origin" xsi:type="n0:GeometricCoordi-
nateSpace">
  <Id id="/NULL"/>
  <Representations>
    <Representation xsi:type="n0:Mechanism" uid="kin--000000000C0F70C0--m">
…
    <Items>
      <RepresentationItem uidRef"kin--0000000008050310--kpair"/>
    </Items>
    <Name>
        <CharacterString>Mechanism.1</CharacterString>
    </Name>
…
    </Representation>
  <Representations>
  <Items>
…
    <RepresentationItem xsi:type="bom:LowOrderKinematicPairWithMotionCou-
pling" uid="kin--0000000008050310--kpair">
      <Name>
          <CharacterString>Plunger-Disc-Spring</CharacterString>
      </Name>
      <Actuation uid="act--kin--0000000008050310--kpair">
        <Name>
          <CharacterString>command #1</CharacterString>
        </Name>
        <Rz>bidirectional</Rz>
```

```xml
        </Actuation>
        <Link1 uidRef="kin--0000000008050310--kpair--klink--0"/>
        <Link2 uidRef="kin--0000000008050310--kpair--klink--1"/>
        <PairFrame1 uidRef="kin--0000000008050310--kpair--kframe--0"/>
        <PairFrame2 uidRef="kin--0000000008050310--kpair--kframe--1"/>
        <Kind>screw_pair</Kind>
        <LowerLimitActualRotation>0.0000000000</LowerLimitActualRotation>
        <Pitch>25.0000000000</Pitch>
        <UpperLimitActualRotation>180.0000000000</UpperLimitActualRotation>
      </RepresentationItem>
…
    </Items>
    <DimensionCount>3</DimensionCount>
</RepresentationContext>
```
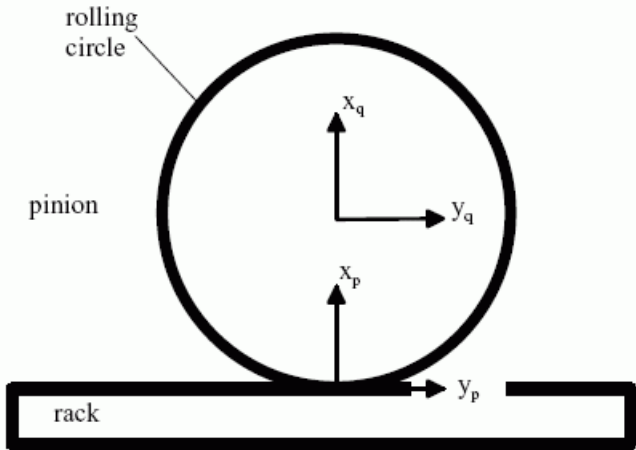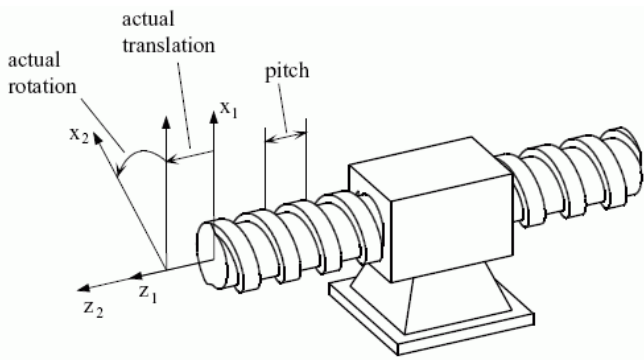
Entity LowOrderKinematicPairWithMotionCoupling attributes	Attribute type
External	OPTIONAL ExternalItem
Name	OPTIONAL DescriptorSelect
Actuation	OPTIONAL ActuatedPair
Link1	KinematicLink
Link2	KinematicLink
PairFrame1	AxisPlacement
PairFrame2	AxisPlacement
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
Bevel	OPTIONAL PlaneAngleMeasure (REAL)
GearRatio	OPTIONAL REAL
HelicalAngle	OPTIONAL PlaneAngleMeasure (REAL)
Kind	LowOrderKinematicPairWithMotionCouplingTypeEnum
LowerLimitActualRotation	OPTIONAL PlaneAngleMeasure (REAL)
LowerLimitRackDisplacement	OPTIONAL LengthMeasure (REAL)
Pitch	OPTIONAL LengthMeasure (REAL)
Radius1	OPTIONAL LengthMeasure (REAL)
Radius2	OPTIONAL LengthMeasure (REAL)
UpperLimitActualRotation	OPTIONAL PlaneAngleMeasure (REAL)
UpperLimitRackDisplacement	OPTIONAL LengthMeasure (REAL)

Table 9: "LowOrderKinematicPairWithMotionCoupling" Attributes

Attribute recommendations

- **Bevel:** the angle between the two x-axes. The value of this attribute need not be specified.

- **GearRatio:** the gear ratio of the pair. The value of this attribute need not be specified.

- **HelicalAngle:** the angle between the two y-axes. The value of this attribute need not be specified.

- **Kind**: the kind of pair. Where applicable, the following values shall be used:

Enumeration value	Explanation
gear_pair	a pair that constrains the motion between two adjacent links to a rolling (and potentially sliding) motion of the second link along the first link. To measure the motion of the second link with respect to the first link, a frame is defined on each of the links such that the origin lies in the center of the rolling circle of the respective link. *Remark*: AP242 does not allow to define actuations on this type of pair, while this seems to be possible in CAD systems. Needs to be discussed when this is tested. According to Part 105, Lower/UpperLimits may be specified on the rotation; Radius1/2, Bevel, GearRatio and HelicalAngle are mandatory. Pitch may not be specified.

linear_flexible_and_pinion_pair ('cable' 🖼 in NX 🖼 in CATIA)	a pair that consists of a linear flexible link that moves around a rolling pinion where the rotation axis of the pinion is perpendicular to the direction of the linear flexible link *Remark*: AP242 does not allow to define actuations on this type of pair, while this seems to be possible in CAD systems. Needs to be discussed when this is tested. According to Part 105, Radius1 is mandatory and no lower/UpperLimits may be specified, nor Bevel, GearRatio(!?), HelicalAngle, Pitch, Radius2.
rack_and_pinion_pair	a pair that describes a pinion rolling on a rack where the rotation axis of the pinion is perpendicular to the direction of the rack. *Remark*: AP242 only allows defining rotational actuations, while some CAD systems also support translational actuations. Needs to be discussed when this is tested. According to Part 105, Radius1 is mandatory and no lower/UpperLimits may be specified, nor Bevel, GearRatio, HelicalAngle, Pitch, Radius2. These restrictions also need to be discussed when this is tested.
screw_pair	a pair that constrains the motion between two adjacent links to a rotation about, and translation along, a common axis, where the translation is proportional to the rotation. The factor of proportionality is given by pitch which defines the translational displacement for one full rotation. The frames are defined on each of the links such that the z-axes coincide and the positive directions agree. *Remark*: AP242 does not allow to define actuations on this type of pair, while this seems to be possible in CAD systems. Needs to be discussed when this is tested. According to Part 105, Pitch is mandatory and no lower/UpperLimits may be specified, nor Bevel, GearRatio, HelicalAngle, Radius1/2. These restrictions also need to be discussed when this is tested.

- Depending on the kind of kinematic pair, one of the following attributes pairs (lower/upper for rotation or displacement) can be set at a time (Lower / Upper have both to be set).

    o **LowerLimitActualRotation**: the minimum value of the pair parameter of the first link. The value of this attribute need not be specified.

    o **UpperLimitActualRotation**: the maximum value of the pair parameter of the first link. The value of this attribute need not be specified.

    o **LowerLimitRackDisplacement**: the minimum value of the pair parameter for the rack. The value of this attribute need not be specified.

    o **UpperLimitRackDisplacement**: the maximum value of the pair parameter for the rack. The value of this attribute need not be specified.

- **Pitch:** the pitch of the screw in the case that the **kind** is of type screw_pair. The value of this attribute need not be specified.

- **Radius1:** the radius of the rolling circle of the first link. The value of this attribute need not be specified.

- **Radius2:** the radius of the rolling circle of the second link. The value of this attribute need not be specified.

Preprocessor Recommendations:

- In case only one Pinion (for example in Rack and Pinion) is involved, it shall be mapped into Radius1.
- The value in **Lower…** shall be smaller than the value in **Upper…**.
- If a value is given in Lower…, the corresponding Upper… attribute shall also have a value.
- Both KinematicLinks of a LowOrderKinematicPairWithMotionCoupling have to be defined within the same GeometricCoordinateSpace than the Mechanism itself.

Remark: depending on the legacy system, not all Kinds are supported, and not all Limits/Ranges are supported.

4.4 Template "Mechanism"

A Mechanism contains all constraints that describe the kinematics of a given assembly.
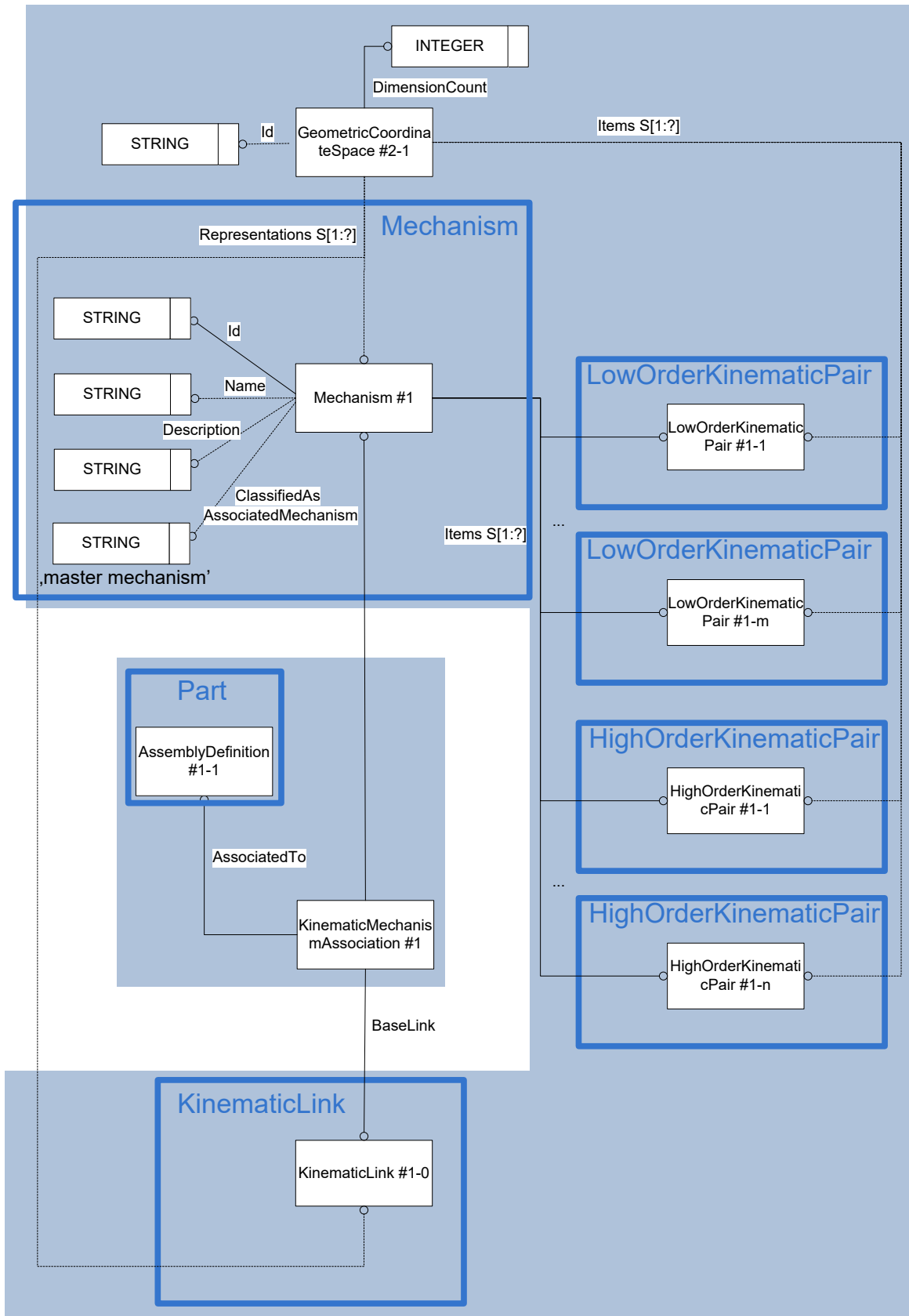
Figure 8: Template "Mechanism"

The Instance Model: STEP exchange file format (ISO10303 AP242 Domain Model XML syntax)

```xml
<RepresentationContext uid="ccs--origin" xsi:type="n0:GeometricCoordinateSpace">
  <Id id="/NULL"/>
  <Representations>
    <Representation xsi:type="n0:Mechanism" uid="kin--0000000005FC5FC0--m">
      <Id id="/NULL"/>
      <Items>
        <RepresentationItem uidRef="kin--0000000004633D60--kpair"/>
        <RepresentationItem uidRef="kin--0000000004633EE0--kpair"/>
        <RepresentationItem uidRef="kin--0000000004634060--kpair"/>
      </Items>
      <Name>
        <CharacterString>Kurvengetriebe</CharacterString>
      </Name>
    </Representation>
    <Representation xsi:type="n0:KinematicLink" uid="kin--0000000004633D60--kpair--klink--1">
      <Id id="/NULL"/>
      <Items>
        <RepresentationItem uidRef="kin--0000000004633D60--kpair--kframe--0"/>
      </Items>
    </Representation>
  <Representations>
  <Items>
    <RepresentationItem xsi:type="n0:LowOrderKinematicPair" uid="kin--0000000004633D60--kpair">
…
    </RepresentationItem>
    <RepresentationItem xsi:type="n0:LowOrderKinematicPair" uid="kin--0000000004633EE0--kpair">
…
    </RepresentationItem>
    <RepresentationItem xsi:type="n0:HighOrderKinematicPair" uid="kin--0000000004634060--kpair">
…
    </RepresentationItem>
…
    <RepresentationItem uid="kin--0000000004633D60--kpair--kframe--0" xsi:type="n0:AxisPlacement">
      <Axis>-1.0000000000,0.0000000000,0.0000000000</Axis>
      <Position>0.0000000000,-5.0000000000,15.0000000000</Position>
      <RefDirection>0.0000000000,-1.0000000000,0.0000000000</RefDirection>
    </RepresentationItem>
  </Items>
  <DimensionCount>3</DimensionCount>
</RepresentationContext>


<Part uid="p--0000000012B9CF70">
…
  <Versions>
    <PartVersion uid="pv--0000000012B9CF70--id1">
…
      <Views>
        <PartView xsi:type="n0:AssemblyDefinition" uid="pvv--0000000012B9CF70--id1">
…
```

```
            <KinematicMechanismAssociation uid="kin--0000000005FC5FC0--kma"
xsi:type="n0:KinematicMechanismAssociation">
                <AssociatedMechanism uidRef="kin--0000000005FC5FC0--m"/>
                <BaseLink uidRef="kin--0000000004633D60--kpair--klink--1"/>
            </KinematicMechanismAssociation>
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
```

Entity Mechanism attributes	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	Id
Items	SET[1:?] of KinematicPair
Name	OPTIONAL DescriptorSelect
RepresentationTypes	OPTIONAL SET[1:?] of ClassSelect
RetentionPeriod	OPTIONAL SET[1:?] RetentionPeriod
VersionId	OPTIONAL IdentifierSelect
RepresentationRelationship	OPTIONAL SET[1:?] OF RepresentationRelationship
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DatetimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
MechanismState	OPTIONAL SET[1:?] of MechanismState

Table 10: "Mechanism" Attributes

Attribute recommendations

- **ClassifiedAs**: the classifications of the Mechanism. The value of this attribute shall be 'master mechanism' (normal case) or 'imported dressup mechanism' (in case of 4.6) or 'local dressup mechanism' for a dressup if defined locally on the master mechanism (special case of 4.7 where both 'master mechanism' and 'local dressup mechanism' reference to the same assembly node via AssociatedTo.

- **Description**: if a dressup is defined on this mechanism, the name of the dressup, otherwise: need not be specified. Use "Description" template (see [242-PAS] 4.6.7).

- **Id**: the identifier or set of identifiers for the Mechanism. Use "Identifier" template (see [242-PAS] 4.6.6).

- **Items**: the set of KinematicPairs of the Mechanism

- **Name**: the words or set of words by which the Mechanism is known. The value of this attribute need not be specified. Use "Description" template (see [242-PAS] 4.6.7).

- **PropertyValueAssignment**: to assign a PropertyValue to the Mechanism. Use the "PropertyAssignment" template (see [242-PAS] 6.2).

- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

Entity KinematicMechanismAssociation attributes	Attribute type
AssociatedMechanism	Mechanism
BaseLink	KinematicLink

Table 11: "KinematicMechanismAssociation" Attributes

Attribute recommendations

- **AssociatedMechanism:** the Mechanism. Use "Mechanism" template (see 4.4).

- **BaseLink**: a KinematicLink that plays the role of a base (fixed part(s)) for the Mechanism. ***Preprocessor Recommendations***:

- Each Mechanism shall have a unique Baselink. If the sending systems supports multiple fixed parts belonging to different KinematicLinks, one of the KinematicLinksl be mapped as BaseLink and the other ones shall be constrained to it via KinematicPairs of kind fully_constrained_pair.

- The KinematicLink referenced by BaseLink shall be involved as Link1 (or Link2) of at least one KinematicPair associated to the Mechanism with an AxisPlacement as PairFraime1 (or PairFrame2) ~~being an identity matrix~~

- According to the issue created on ISO Jira as TCSC410303-817, all the Occurrences referenced by the KinematicLinks of the KinematicPairs of the Mechanism shall be component parts of the assembly structure on which the Mechanism is defined (AssociatedTo).

- All the KinematicPairs of a Mechanism have to be defined within the same GeometricCoordinateSpace than the Mechanism itself.

- Therefore, it is also recommended that the length unit and the angle unit used are the same throughout the kinematic structure of the mechanism, and be same as the length unit used for the assembly structure and the geometry itself.

- According [242-PAS] 6.1, the angle unit used for the Lower/UpperLimits of KinematicPairs shall be defined in GeometricCoordinateSpace.Units, while the LengthUnit can be overtaken from the one defined in ExchangeContext.DefaultUnit rather than being defined again in the GeometricCoordinateSpace.

- It is not recommended to exchange kinematic constraints based on single bodies within a CAD model, since these cannot be supported with the current mapping.

4.5 Mechanism Structure

Note: Some concepts in this section, namely the definition and transfer of Rigid Groups, are included for completeness but have not yet been fully tested. Hence, these definitions might change in the future.

Most CAD systems require that the assembly nodes involved in KinematicPair may not have further KinematicPairs between their components:
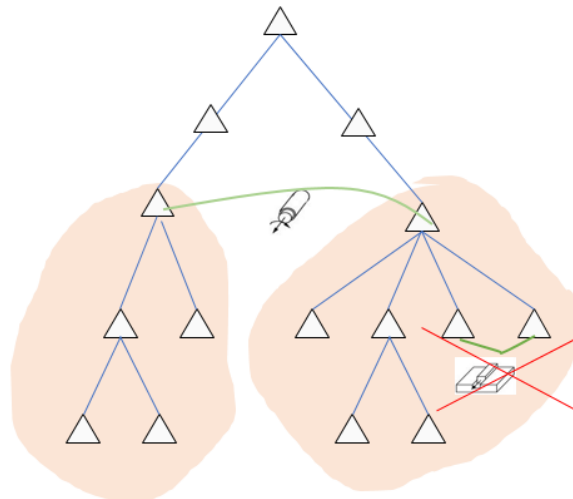
Figure 9: Violation of rigidity rule within an assembly involved in a KinematicPair

Exceptions are so-called 'flexible assemblies' where each occurrence may have a different shape (currently out of scope, even if described in [242-PAS] 7.1 regarding Flexible Parts) and defining a separate mechanism for the whole assembly node or a subset of it (currently out of scope).

1. In simple cases:
   - In NX each Rigid Group contains only one occurrence of an assembly node or single part and can be mapped to a KinematikLink => the number of Rigid Groups is the number of moving parts/assemblies.
   - In CATIA the mechanism can be built upon of the original BoM Structure
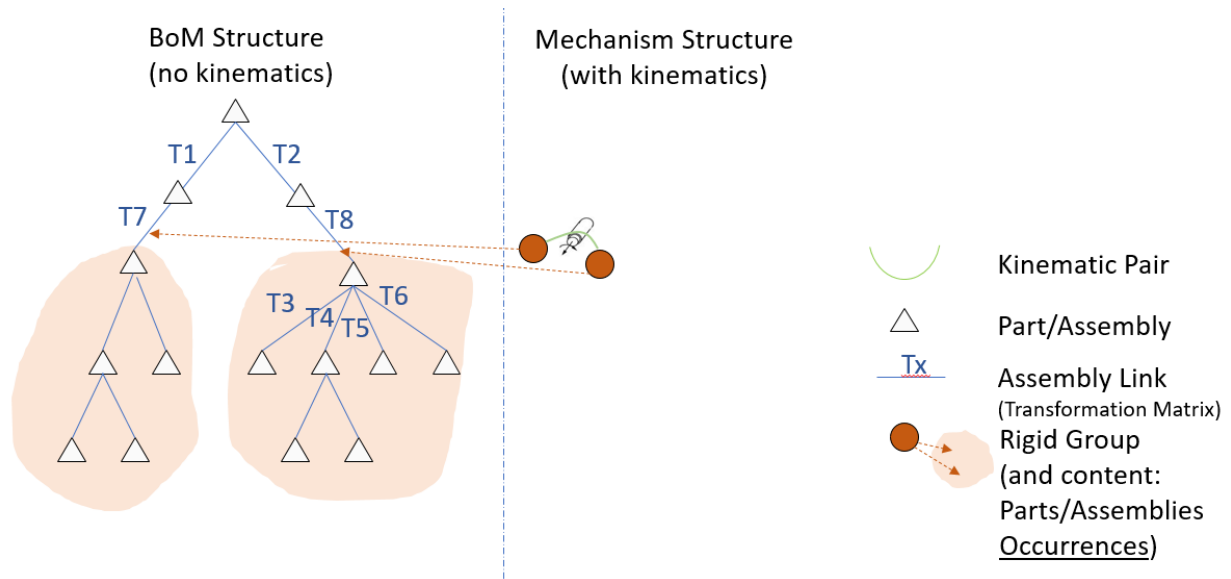
Figure 10: Simple case: rigid groups contain only one part/assembly

2. In complex cases, Rigid Groups (like in NX) or additional Assembly Nodes (like in CATIA) are required to contain only those components of the assembly that shall move according to the same kinematic law (i.e. that a rigid among eachother). All these Rigid Groups or assemblies build a flat structure called here the Kinematic Mechanism Structure.
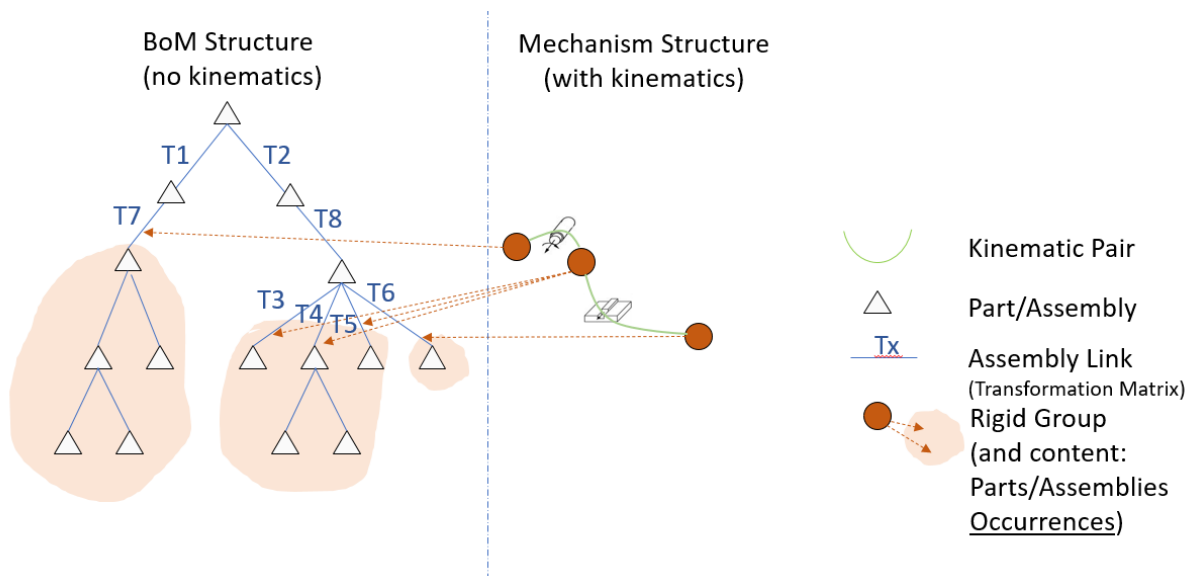
Figure 11: Complex case: Rigid Groups made of more than one part/assembly

Since a KinematicLink can be associated to only one part/assembly occurrence, these Rigid Groups cannot be mapped as usual, but rather as an additional Assembly Node (like in CATIA) on which the KinematicLink can apply.
The flat structure mentioned above shall be mapped by a dedicated top Assembly Node having all the (original and additional) Assembly Nodes + Single Parts associated to KinematicLinks as its components.
As soon as one complex rigid group is present, all the simple rigid groups shall be built as childs of the top level assembly.
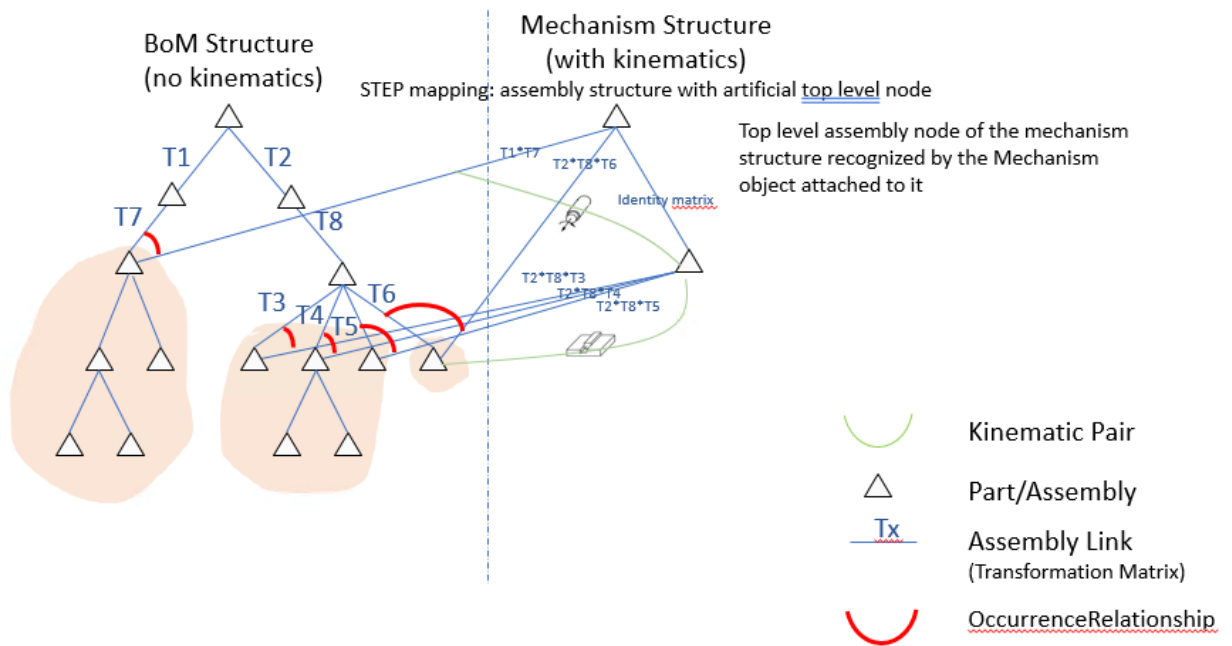
Figure 12: Complex case: use of additional assembly nodes

All the components (direct or indirect via sub-assemblies) of the components of the additional Assembly Node introduced for the Rigid Group are implicitly fully constrained to eachother (rigid).
Kinematic pairs are only allowed between rigid groups.
All the assembly nodes and single parts from the BoM Structure that are not built into the Kinematic Mechanism Structure do not move at all.

In some CAD systems (like NX) using the concept of Rigid Groups, the Mechanism always has implicitly the same initial position than in the BoM Structure.

Since according to the [242-PAS] 7.1, each assembly component shall have its own <u>Occurrence</u> for each usage, the relevant occurrences from the BoM Structure shall be duplicated to Occurrences in the Mechanism Structure:

- the 3D positioning of the components of a complex rigid group within the new assembly nodes shall be copied from the AxisPlacements used for their placement within the Mechanism (flat structure). Doing so, no 3D positioning (identity matrix) is necessary for the new assembly nodes within the top assembly node of the mechanism structure.
- the 3D positioning of simple rigid groups within the top assembly node of the mechanism structure shall be copied from the AxisPlacements used for their placement within the Mechanism (flat structure).
- each occurrence from the BoM Structure and its corresponding occurrence in the Mechanism Structure shall be related via an OccurrenceRelationship having RelationType = 'kinematic definition'. Relating shall refer to the occurrence in the Mechanism Structure and Related to the occurrence in the BoM Structure.

In some CAD systems (like CATIA), the Kinematic Mechanism Structure can be defined and maintained independently from the BoM Structure. Therefore, it may have its own 3D positioning => the Mechanism may have another initial position than in the BoM Structure.
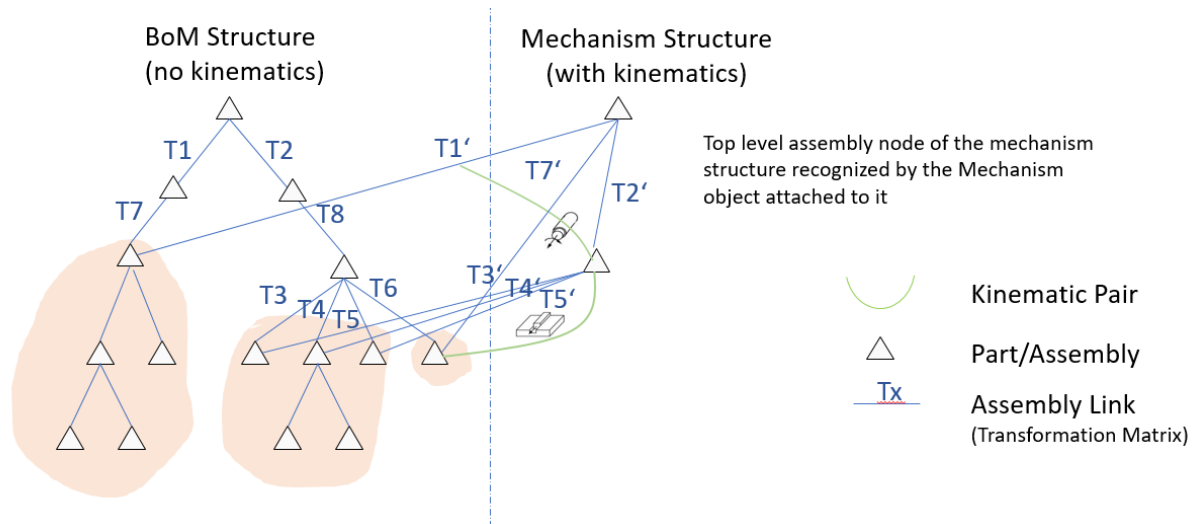
Figure 13: Complex case: different initial position in BoM and Mechanism Structures

Preprocessor Recommendations:

Since the conversion of the 3D positioning within the Mechanism Structure (transformation matrices used to position the 'points of contact' CartesianPoints of the KinematicLinks) so that it fits to the initial position of the BoM Structure (conform to NX) can be very complex:

- A CAD system like CATIA shall export only the mechanism structure (not the BoM structure) so that a CAD system like NX can build rigid groups out of it. In this case, the CATIA mechanism structure becomes the BoM Structure in NX. Otherwise, the BoM Structure shall be imported in NX as a distinct structure, without kinematics (as it is in CATIA).
- A CAD system like NX shall export the name of the rigid groups als KinematicLink.Id (not '/NULL').

Remark: a single part / assembly node occurrence shall *not* be associated to multiple rigid groups, neither directly nor indirectly (through intermediate assembly nodes). This is indeed possible in NX but may lead to a conflict.

Postprocessor Recommendations:

- For a complex case as explained in 2., a CAD system like CATIA shall import at least the mechanism structure and ignore the OccurrenceRelationships having RelationType = 'kinematic definition'
- For a complex case as explained in 2., a CAD system like NX shall build the rigid groups as following:
  - One for each KinematikLink
  - If the Assembly Node is not built into the BoM Structure and if for each component, an OccurrenceRelationship with RelationType 'kinematic definition' exists, do not map the Assembly Node itself but associate its components (occurrences) referenced by Relating<-OccurrenceRelationship.Relates (created during export from NX) to the Rigid Group.

4.6   Template "Import Mechanism"

A Mechanism defined on a given assembly node A1 may be imported on another assembly node A2 placed (within the product structure) above A1. The original mechanism is called the master mechanism, while the other is called the imported mechanism. All the kinematic constraints of the master mechanism may be used by the imported dressup mechanism (but may only be modified in the master mechanism).

This feature is especially used for dressups (which reuse the kinematic pairs defined in the master mechanism on a part occurrence of A1 and associate them to occurrences of further parts or assemblies built beyond the A2.
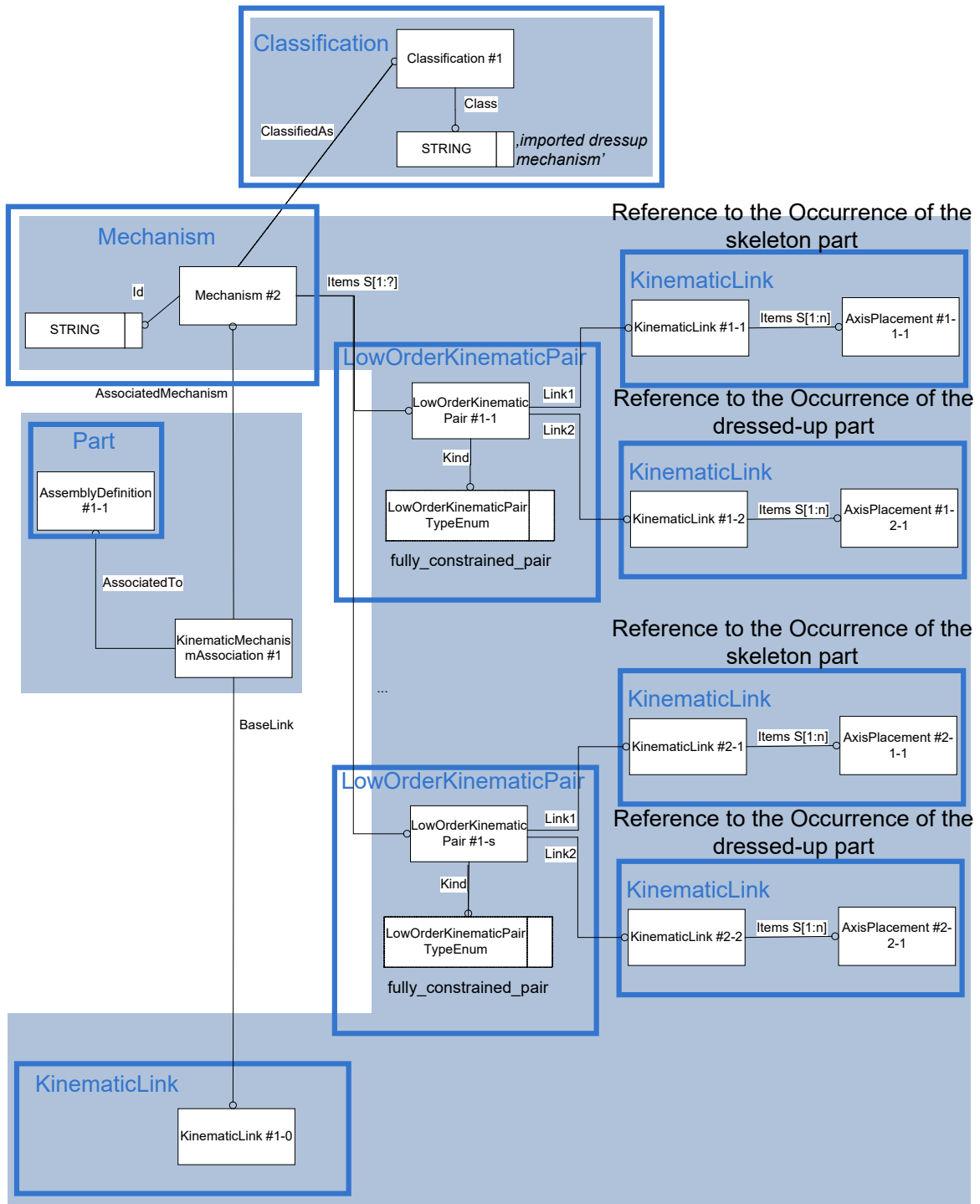
Figure 14: Template "Import Mechanism"

In the special case where the dressup mechanism is defined on the local Mechanism, both are defined on the same assembly node A1. The original mechanism is called the master mechanism, while the other is called the local dressup mechanism. All the kinematic constraints of the master mechanism may be used by the dressup mechanism (but may only be modified in the master mechanism).
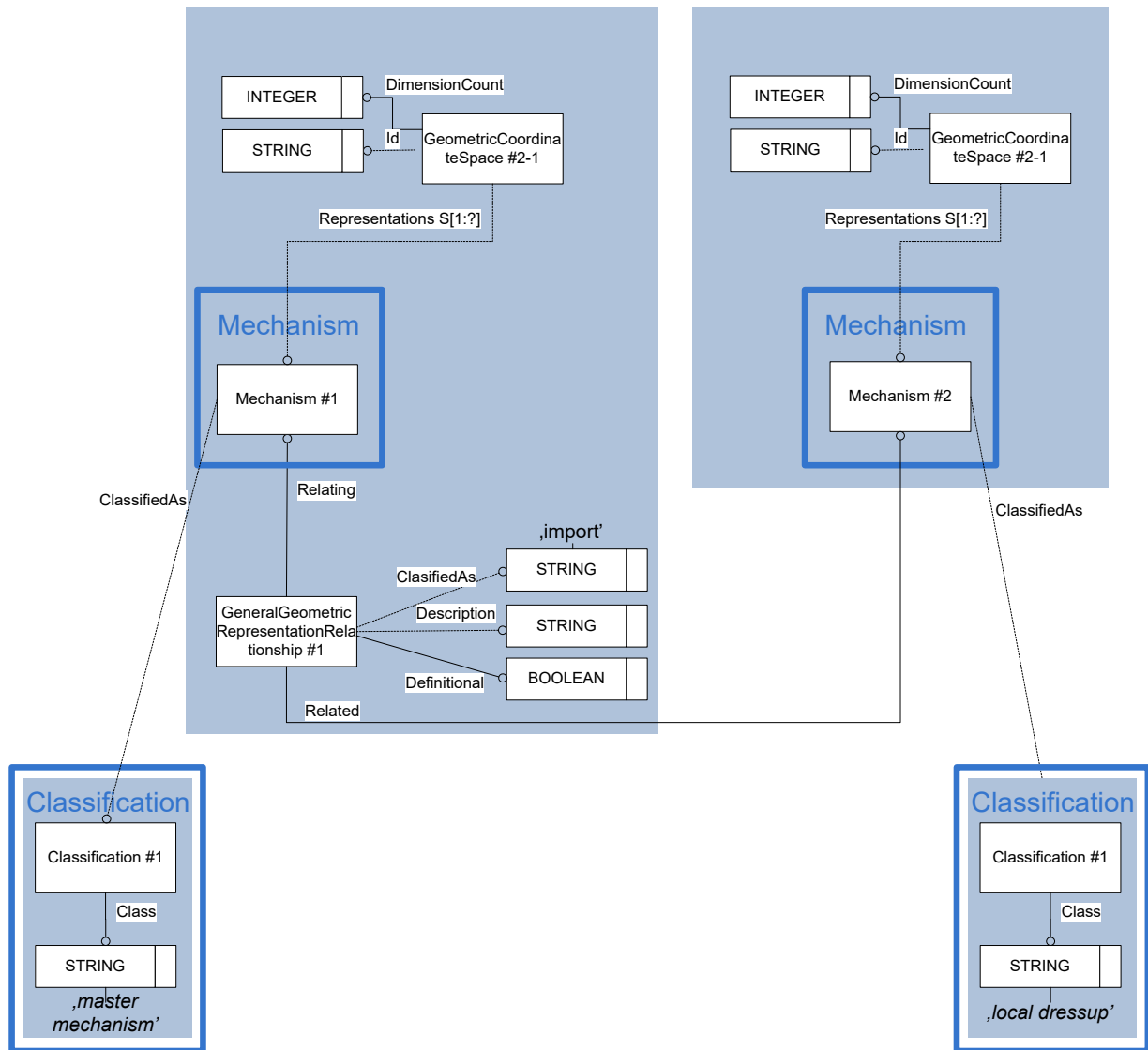
Figure 15: Template "Import Mechanism" Special Case for Local Dressup

The Instance Model: STEP exchange file format (ISO10303 AP242 Domain Model XML syntax)

```
<RepresentationContext uid="ccs--origin" xsi:type="n0:GeometricCoordi-
nateSpace">
  <Id id="/NULL"/>
  <Representations>
    <Representation xsi:type="n0:Mechanism" uid="kin--0000000006039F00--m">

…

      <Id id="kin--0000000006039F00--m"/>

…

      <RepresentationRelationship uid="kin--0000000006039F00--mrel"
xsi:type="n0:GeneralGeometricRepresentationRelationship">

        <Definitional>true</Definitional>

        <Related uidRef="kin--0000000006039F00--m2"/>

        <RelationType>import</RelationType>

      </RepresentationRelationship>

    </Representation>
  <Representations>
  <DimensionCount>3</DimensionCount>
</RepresentationContext>
```

Preprocessor Recommendations:

- The master and the imported/local dressup mechanism usually do not share the same instance of GeometricCartesianSpace, since the BaseLink of each mechanism are usually defined in different coordinate spaces.
- For a given mechanism (master, local or imported), only one dressup shall be defined. A mechanism may be imported in different assembly nodes (each of them having one and only one dressup).
- In the special case of a 'local dressup mechanism', the dressup if defined locally on the master mechanism (special case of 4.5 where both 'master mechanism' and 'local dressup mechanism' reference to the same assembly node via AssociatedTo

4.7   Template "Kinematic Dressup"

A Dressup enables to associate fully constrained pairs of part occurrences involved in a mechanism with part or assembly occurrences not involved in a mechanism. For each dressup association, two KinematicLinks and a LowKinematicPair of kind fully_constrained_pair is specified.

Normally the kinematic is defined within a master Mechanism (involving a minimum number of parts having simplified wired geometry) and used on the same assembly node ('local') or imported on another assembly node placed above in the product structure (see 4.5) => in this case, the dressup links are defined within the local or imported dressup mechanism. Doing so, the performance of kinematics is improved within the CAD systems.
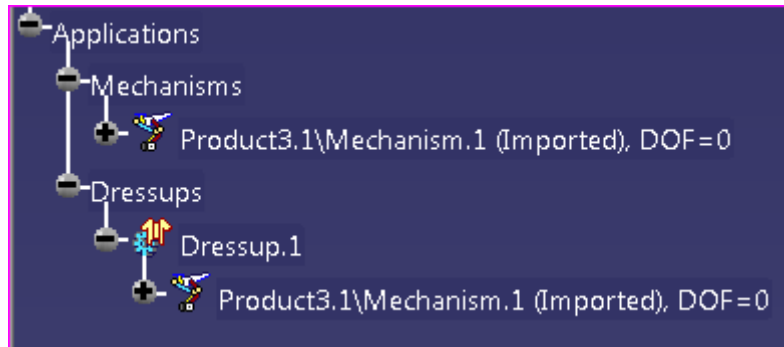
But a dressup may be defined within one mechanism, if the Kinematic pairs involving one part occurrence shall apply in the same way to several further parts or assemblies.

In the following CATIA window, an example is shown how for each kinematic Link of the master mechanism (chosen from the select box place above the two lists), one or multiple part/assembly occurrences of the assembly (on which the dressup mechanism is attached) may be chosen (left list) and added to the right list. All these selected part occurrences are fully constrained with the above Kinematic Link:
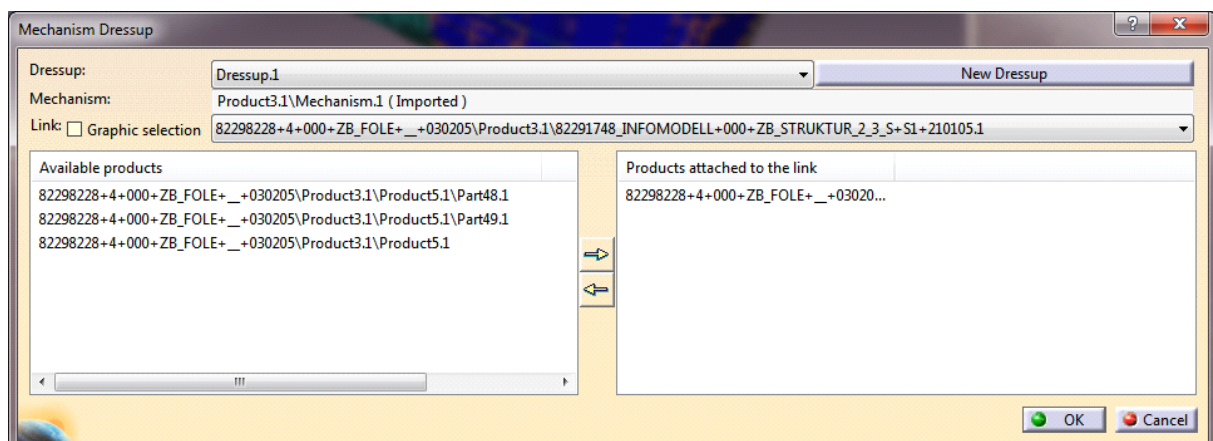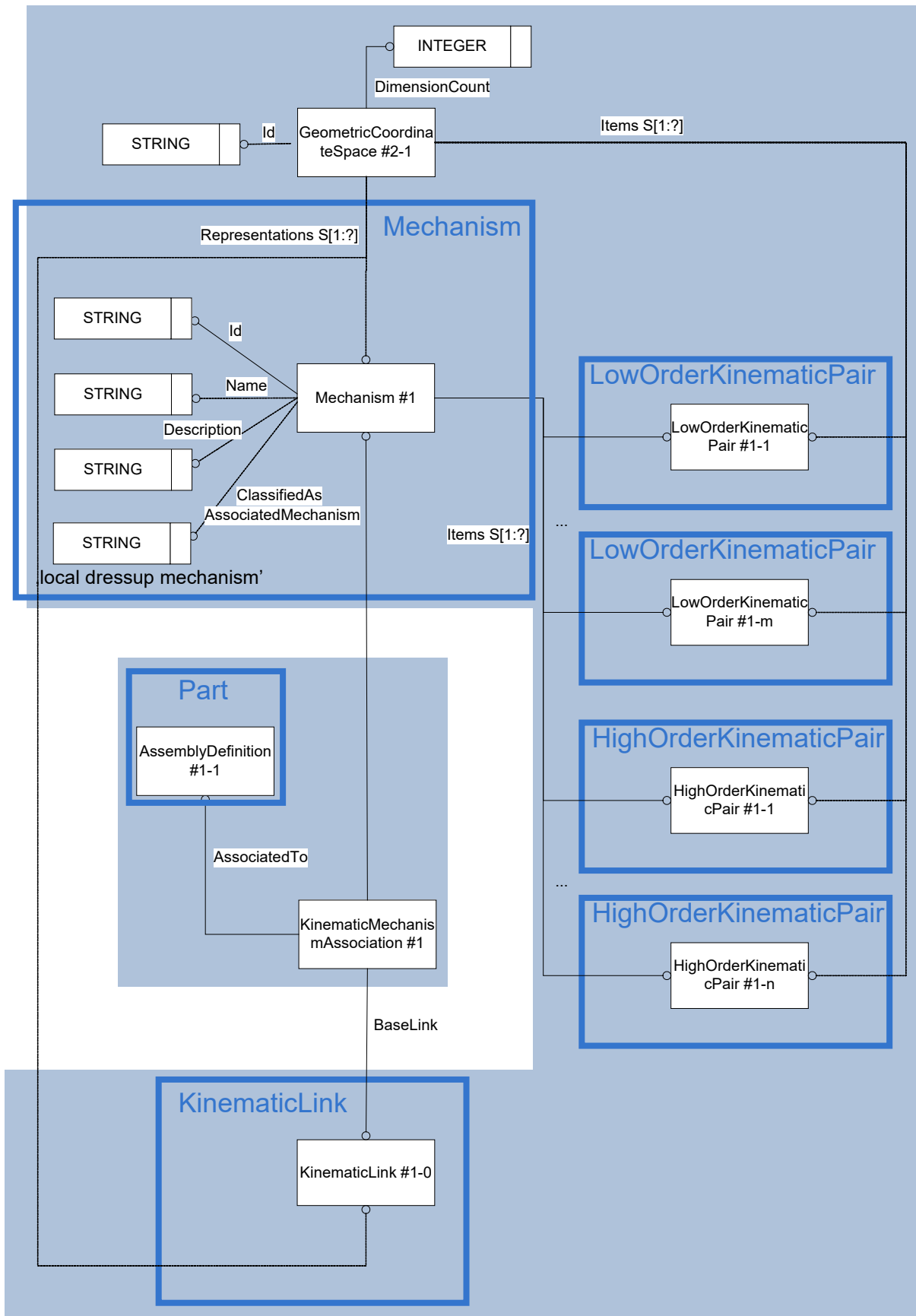
Figure 16: Template "KinematicDressup"

The Instance Model: STEP exchange file format (ISO10303 AP242 Domain Model XML syntax)

```xml
<RepresentationContext uid="ccs--origin" xsi:type="n0:GeometricCoordi-
nateSpace">
  <Id id="/NULL"/>
  <Representations>
<Representation xsi:type="n0:Mechanism" uid="kin--0000000006039F00--m">

…

  <Id id="kin--0000000006039F00--m"/>

…

  <RepresentationRelationship uid="kin--0000000006039F00--mrel"
xsi:type="n0:GeneralGeometricRepresentationRelationship">

    <Definitional>true</Definitional>

    <Related uidRef="kin--0000000006039F00--m2"/>

    <RelationType>import</RelationType>

  </RepresentationRelationship>

</Representation>

  <DimensionCount>3</DimensionCount>
</RepresentationContext>

<Classification uid="cla--im">
  <Class>
    <ClassString>imported dressup mechanism</ClassString>
  </Class>
</Classification>
<RepresentationContext uid="ccs--origin" xsi:type="n0:GeometricCoordi-
nateSpace">
  <Id id="/NULL"/>
  <Representations>
<Representation xsi:type="n0:Mechanism" uid="kin--0000000006039F00--m2">

  <ClassifiedAs>

    <Classification uidRef="cla--im"/>

  </ClassifiedAs>

  <ContextOfItems uidRef="ccs--origin"/>

  <Id id="kin--0000000006039F00--m2"/>

  <Items>

        <RepresentationItem uidRef="kin--0000000006039F00--m2--kpair1"/>
</Items>

…

  <Name>

    <CharacterString>Training Dressup</CharacterString>

  </Name>

</Representation>

  <Representations>
  <Items>

    <RepresentationItem xsi:type="n0:LowOrderKinematicPair" uid="kin--
0000000006039F00--m2--kpair1">
```

```xml
      <Name>
            <CharacterString>Plunger-Disc-Spring</CharacterString>
      </Name>
      <Link1 uidRef="kin--00000000129AA190--17--1"/>
      <Link2 uidRef="kin--00000000124E1D30--21--1"/>
      <PairFrame1 uidRef="kin--00000000129AA190--17--1--klfp--dummy"/>
      <PairFrame2 uidRef="kin--00000000124E1D30--21--1--klfp--dummy"/>
      <Kind>fully_constrained_pair</Kind>
   </RepresentationItem>
  </Items>
  <DimensionCount>3</DimensionCount>
</RepresentationContext>


<RepresentationContext uid="ccs--origin" xsi:type="n0:GeometricCoordi-
nateSpace">
  <Id id="/NULL"/>
  <Representations>
<Representation xsi:type="n0:KinematicLink" uid="kin--0000000006039F00--klfp-
-m2">
  <Description>
    <CharacterString>Fixed Part for &gt;Training Dressup&lt;</Character-
String>
  </Description>
  <Id id="kin--0000000006039F00--klfp--id--m2"/>
  <Items>
        <RepresentationItem uidRef="kin--0000000006039F00--klfp--dummy--m2"/>
  </Items>
</Representation>
  <Representations>
  <Items>
    <RepresentationItem uid="kin--0000000006039F00--klfp--dummy--m2"
xsi:type="n0:AxisPlacement">
      <Axis>0.0000000000,0.0000000000,1.0000000000</Axis>
      <Position>0.0000000000,0.0000000000,0.0000000000</Position>
      <RefDirection>1.0000000000,0.0000000000,0.0000000000</RefDirection>
    </RepresentationItem>
  </Items>
  <DimensionCount>3</DimensionCount>
</RepresentationContext>


<RepresentationContext uid="ccs--origin" xsi:type="n0:GeometricCoordinateS-
pace">
  <Id id="/NULL"/>
  <Representations>
<Representation xsi:type="n0:KinematicLink" uid="kin--00000000129AA190--17--
1">
  <Id id="kin--00000000129AA190--17--1--id"/>
```

```xml
  <Items>

  </Items>

</Representation>

  <Representations>
  <Items>
    <RepresentationItem uid="kin--00000000129AA190--17--1--klfp--dummy"
xsi:type="n0:AxisPlacement">

      <Axis>0.0000000000,0.0000000000,1.0000000000</Axis>

      <Position>0.0000000000,0.0000000000,0.0000000000</Position>

      <RefDirection>1.0000000000,0.0000000000,0.0000000000</RefDirection>

    </RepresentationItem>

  </Items>
  <DimensionCount>3</DimensionCount>
</RepresentationContext>


<RepresentationContext uid="ccs--origin" xsi:type="n0:GeometricCoordinateS-
pace">
  <Id id="/NULL"/>
  <Representations>
<Representation xsi:type="n0:KinematicLink" uid="kin--00000000124E1D30--21--
1">

  <ContextOfItems uidRef="ccs--origin"/>

  <Id id="kin--00000000124E1D30--21--1--id"/>

  <Items>

  </Items>

</Representation>

  <Representations>
  <Items>
    <RepresentationItem uid="kin--00000000124E1D30--21--1--klfp--dummy"
xsi:type="n0:AxisPlacement">

      <Axis>0.0000000000,0.0000000000,1.0000000000</Axis>

      <Position>0.0000000000,0.0000000000,0.0000000000</Position>

      <RefDirection>1.0000000000,0.0000000000,0.0000000000</RefDirection>

    </RepresentationItem>

  </Items>
  <DimensionCount>3</DimensionCount>
</RepresentationContext>
```

Preprocessor Recommendations:

- Link1 references a new instance of KinematicLink that refers to the Occurrence of part that has a simplified wired geometry

- Link2 references a new instance of KinematicLink that refers to the Occurrence of the part with full geometry

- All the Occurrences referenced over Link1 shall be component parts of the assembly structure of Mechanism#1.AssociatedTo.

- All the Occurrences referenced over Link2 shall be component parts or assembly nodes of the assembly structure of Mechanism#2.AssociatedTo.

- Since only one dressup may be defined for a given mechanism (local or imported) at a given assembly node, the name of the dressup may be mapped to Mechanism.Description.

- Since all the kinematic pairs defined within a dressup mechanism are fully constrained, the BaseLink can be chosen out of any of them.

- In the special case of a 'local dressup mechanism', the BaseLink of the dressup shall be the same as the BaseLink of the local master mechanism

4.8 Template "LinkMotionAlongPath"

A LinkMotionAlongPath is a type of GeometricRepresentation for the definition of parameter or time coordinated movements of one or several links along paths in space.

Remark: a LinkMotionAlongPath corresponds to an Animation in CATIA and to a set of FrameSequences in PLMXML.
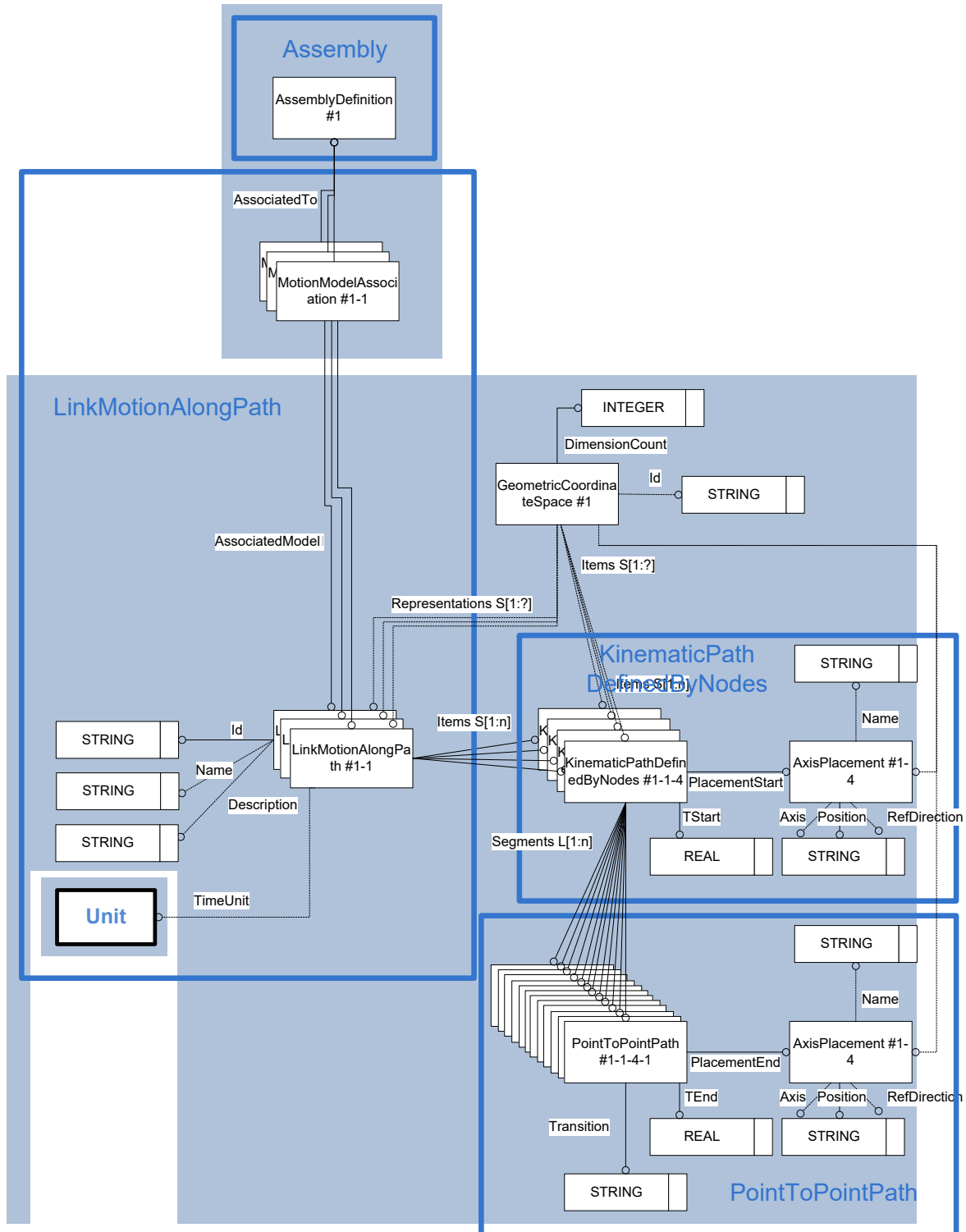
Figure 17: Templates "LinkMotionAlongPath", "KinematicPathDefinedByNodes" and "PointToPointPath"

The Instance Model: STEP exchange file format (ISO10303 AP242 Domain Model XML syntax)

Entity LinkMotionAlongPath attributes	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	Id
Items	SET[1:?] of KinematicPathDefinedByNodes
Name	OPTIONAL DescriptorSelect
RepresentationTypes	OPTIONAL SET[1:?] of ClassSelect
RetentionPeriod	OPTIONAL SET[1:?] RetentionPeriod
VersionId	OPTIONAL IdentifierSelect
RepresentationRelationship	OPTIONAL SET[1:?] OF RepresentationRelationship
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DatetimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
TimeUnit	OPTIONAL Unit
LinkMotionDecompositionRelationship	OPTIONAL SET[1:?] OF LinkMotionDecompositionRelationship
LinkMotionRelationship	OPTIONAL SET[1:?] OF LinkMotionRelationship

Table 12: "LinkMotionAlongPath" Attributes

Attribute recommendations

- **Description**: the reason for the creation of the LinkMotionAlongPath. The value of this attribute need not be specified. Use "Description" template (see [242-PAS] 4.6.7).

- **Id**: the identifier or set of identifiers for the LinkMotionAlongPath. Use "Identifier" template (see [242-PAS]).

- **Items**: the set of KinematicPathDefinedByNodes that that constitute the motion path.

- **Name**: the words or set of words by which the LinkMotionAlongPath is known. The value of this attribute need not be specified. Use "Description" template (see [242-PAS] 4.6.7).

- **PropertyValueAssignment**: to assign a PropertyValue to the Motion. Use the "PropertyAssignment" template (see [242-PAS] 6.2).

- **TimeUnit**: the unit for measures of time related to this motion definition. The value of this attribute need not be specified. Use "Unit" template (see [242-PAS]).

- **LinkMotionRelationship**: the associated GeometricModels involved by this Motion. See template "Full kinematic positioning representation" in 4.12.

- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

Preprocessor Recommendations:

- The LinkMotionAlongPath(s) associated to an AssemblyDefinition shall be defined within the same GeometricCoordinateSystem than the AssemblyDefinition.

- The LinkMotionAlongPaths associated to the same AssemblyDefinition and sharing one of many KinematicPathDefinedByNodes shall have the same TimeUnit.

- If the AssemblyDefinition has a GeometricModel attached to it, this geometry is not part of the motion, except it is explicitly added to the motion via the full kinematic positioning (see next sections). Therefore, there is no obligation to define the AssemblyDefinition and the PlacementStart in the same GeometricCoordinateSystem.

- The AssemblyDefinition associated to the LinkMotionAlongPaths shall be a common top-level node of all parts/geometries that are defined within the paths of the motion.

- Multiple LinkMotionAlongPaths can be defined on one AssemblyDefinition (typically alternative/variants to each other).

- The AxisPlacement of all the PointToPointPaths of a KinematicPathDefinedByNodes shall be in the same GeometricCoordinateSystem.

- TimeUnit may be left unset. In this case, no speed indication is given to the motion, but just the relative motion of the paths to each other => the unit chosen to run the motion shall be the same for all paths.

Entity MotionModelAssociation attributes	Attribute type
AssociatedModel	MotionModelSelect

Table 13: "MotionModelAssociation" Attributes

Attribute recommendations

- **AssociatedModel**: the LinkMotionAlongPath.

4.9 KinematicPathDefinedByNodes

A KinematicPathDefinedByNodes is a type of RepresentationItem that is a parameter or time coordinated path in space along which motion shall take place. The path is defined by a start position and orientation in space at a particular parameter or time and a list of segments to traverse through.

Remark: a KinematicPathDefinedByNodes corresponds to a Channel in CATIA and to a FrameSequence in PLMXML.

The Instance Model: STEP exchange file format (ISO10303 AP242 Domain Model XML syntax)

Entity KinematicPathDefinedByNodes attributes	Attribute type
External	OPTIONAL ExternalItem
Name	OPTIONAL DescriptorSelect
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
PlacementStart	AxisPlacement
Segments	LIST [1:?] OF PathSegment
TStart	REAL

Table 14: "KinematicPathDefinedByNodes" Attributes

Attribute recommendations

- **PropertyValueAssignment**: to assign a KinematicPathDefinedByNodes to the Motion. Use the "PropertyAssignment" template (see [242-PAS] 6.2).

- **PlacementStart**: the AxisPlacement that defines the reference coordinate system and, thus, the location and orientation of the start-point of the KinematicPathDefinedByNodes. Use "AxisPlacement" definition within the "GeometricModel" template (see [242-PAS]).

- **Segments***:* the ordered list of PathSegment objects that the KinematicPathDefinedByNodes consists of. The travel along the KinematicPathDefinedByNodes starts at the first element in the list. Use "PointToPointPath" template (see 4.10).

- **TStart:** the parameter value that represents the start location of a travel along the KinematicPathDefinedByNodes

- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

Preprocessor Recommendations:

- Multiple KinematicPathDefinedByNodes can be defined on one LinkMotionAlongPaths (to see the orchestrated motion of several paths).

- All the KinematicPathDefinedByNodes associated to a LinkMotionAlongPath shall be in the same GeometricCoordinateSystem as the LinkMotionAlongPath

- The AxisPlacement (StartPlacement) is the absolute placement of the part within the AssemblyDefinition. It is computed out of its relative placement in the static assembly structure.

- TStart shall be zero or greater and shall be expressed in the (optional) TimeUnit.
- The motion is expected to start at the time 0 and to end at the highest TEnd of all Paths.
- Each Path may have a different TStart and their highest TEnd do not need to have the same value.

4.10 Template "PointToPointPath"

A PathSegment is a portion of a complete kinematic path and describes the movement of a link from a previous position and orientation to a new position and orientation in space. A PathSegment is either a CircularPath, a LinearPath or a PointToPointPath.

For the moment, only PointToPointPaths are in scope of this document.

Remark: a PointToPointPath corresponds to a Channel Step in CATIA and to a FrameStep in PLMXML.

The Instance Model: STEP exchange file format (ISO10303 AP242 Domain Model XML syntax)

Entity PointToPointPath attributes	Attribute type
PlacementEnd	AxisPlacement
TEnd	REAL
Transition	TransitionCodeEnum

Table 15: "PointToPointPath" Attributes

Attribute recommendations

- **PlacementEnd**: the AxisPlacement to which the parameters of the PathSegment refer. It is the location and orientation of the end-point of the PathSegment. Use "AxisPlacement" definition within the "GeometricModel" template (see [242-PAS]).

- **TEnd:** the parameter value that represents the end location of a travel along the PathSegment

- **Transition:** the type of smoothness of the motion along the path. Where applicable, the following values shall be used:

Enumeration value	Explanation
cont same gradient	the motion occurs continuously with a constant gradient
cont same gradient same curvature	gradient and curvature of the motion are constant
continuous	the motion is continuous
discontinuous	the position between the path segments is not defined

For the moment, the use of the value 'discontinuous' is recommended.

Preprocessor Recommendations:

- The PlacementEnd shall be defined in the same GeometricCoordinateSystem than the PlacementStart of the KinematicPathDefinedByNodes => incremental positioning (in which each Segment has its own GeometricCoordinateSystem and the rotation/translation of a given segment builds on the GeometricCoordinateSystem of the predecessor segment) is not supported.

- The AxisPlacement (EndPlacement) is the absolute placement of the part within the AssemblyDefinition. It is computed out of its relative placement in the static assembly structure.

- In order to tell that a path shall be immobile during a given time interval, the KinematicPathDefinedByNodes.PlacementStart and the Segments[1].PlacementEnd, resp. the Segments[ix].PlacementEnd and Segments[ix+1].PlacementEnd shall be identical.

- The segments in the list must have increasing TEnds: Segments[1].Tend > TStart resp. Segments[ix+1].Tend > Segments[ix].Tend => endless loops are not supported (can be activated within the viewer that runs the motion).

- If the source system supports Segments having different duration, they shall be converted to segments all having the same duration.

Postprocessor Recommendations:

- If the target system requires that each Segment has its own GeometricCoordinateSystem and is positioned in the GeometricCoordinateSystem of its predecessor Segment (so-called 'incremental' positioning), the PlacementEnds have to be transformed into the appropriate GeometricCoordinateSystems.

4.11 Simplified Kinematic Positioning Representation

This enables to relate the Kinematic path to the part (or to a SpecificOccurrence of this part) in the case no GeometricModel is involved.

Figure 18: Template "ProductStructureKinematicPathAssociation"

The Instance Model: STEP exchange file format (ISO10303 AP242 Domain Model XML syntax)

Entity ProductStructureKinematicPathAssociation attributes	Attribute type
AssociatedMotionRepresentation	LinkMotionAlongPath
AssociatedMotionPath	KinematicPathdefinedByNode
SubStructure	OPTIONAL SpecifiedOccurrence

Table 16: "ProductStructureKinematicPathAssociation" Attributes

Attribute recommendations

- **AssociatedMotionRepresentation**: the LinkMotionAlongPath that specifies a motion path. Use "LinkMotionAlongPath" template (see 4.8).

- **AssociatedMotionPath***:* the KinematicPathDefinedByNodes object that is being transformed. Use "KinematicPathDefinedByNodes" template (see 4.9).

- **SubStructure:** the SpecifiedOccurrence to which the kinematic path applies. The value of this attribute need not be specified. Use "SpecifiedOcurrence" template (see [242-PAS]).

Preprocessor Recommendation:

- If SubStructure is unset, the kinematic path applies to the SingleOccurrence attached as 'Related' to the NextAssemblyOccurrenceUsage) in which the current ProductStructure-KinematicPathAssociation is embedded.

  If set, the kinematic path applies to one particular SpecifiedOccurrence of one of its components.
  Remark: SpecifiedOccurrences are mostly not supported by the PDM systems => currently out of scope

- ProductStructureKinematicPathAssociation.AssociatedMotionRepresentation and ProductStructureKinematicPathAssociation.AssociatedTo shall reference the same AssemblyDefinition.

4.12 Full Kinematic Positioning Representation

This enables to relate the Kinematic path to the part (or to a SpecificOccurrence of this part) in the case a GeometricModel is involved.
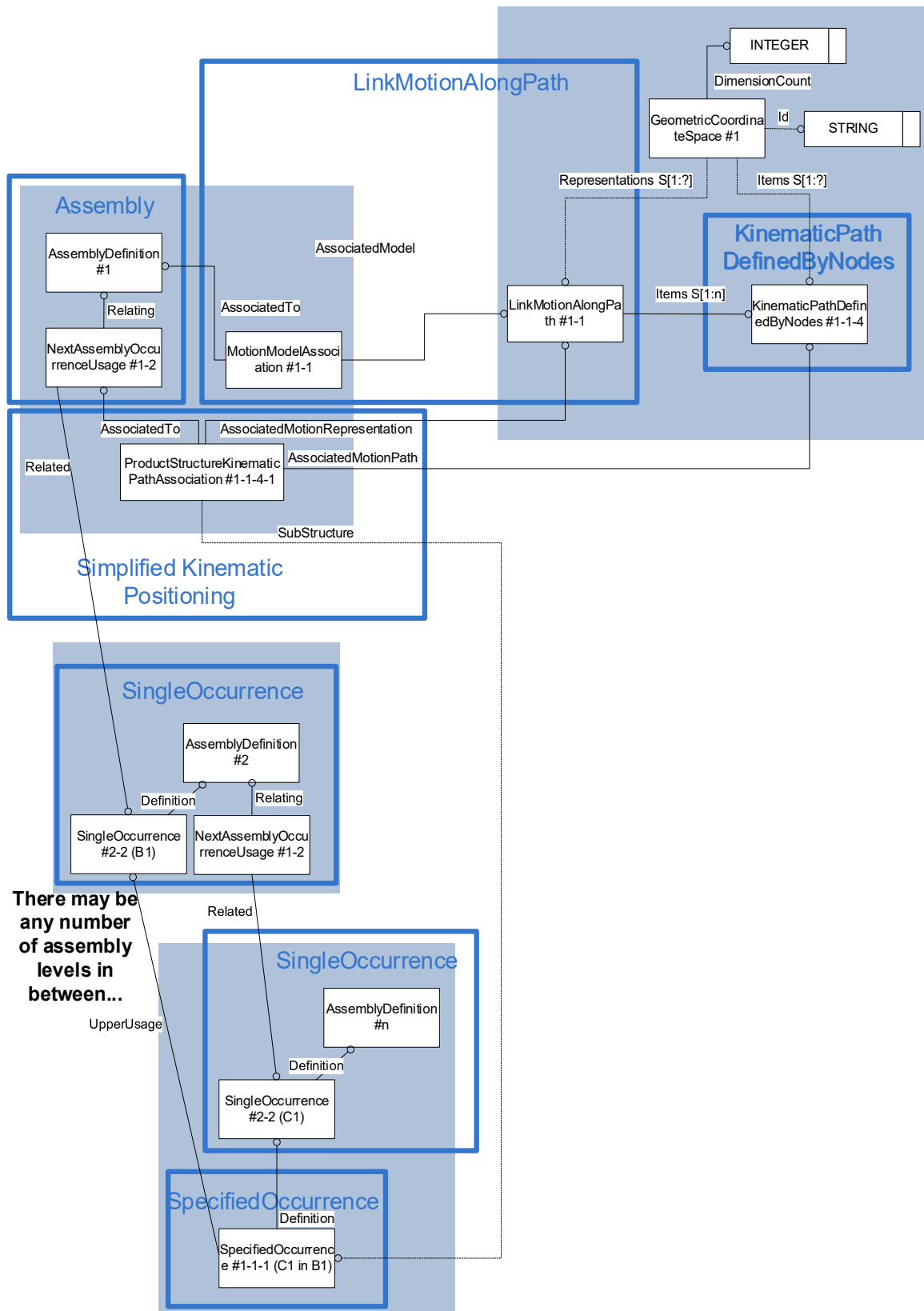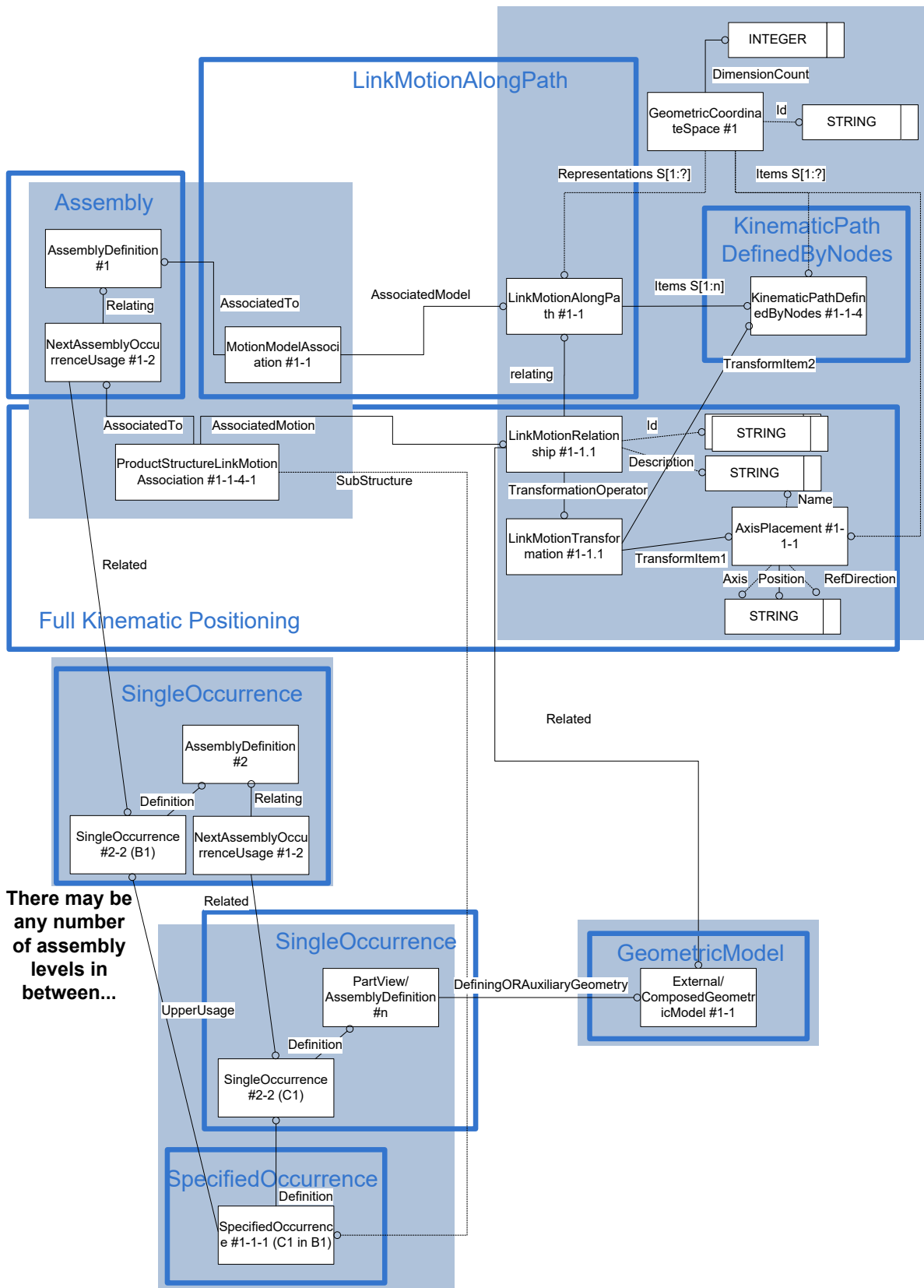

Figure 19: Template "ProductStructureLinkMotionAssociation"

The Instance Model: STEP exchange file format (ISO10303 AP242 Domain Model XML syntax)

Entity ProductStructureLinkMotionAssociation attributes	Attribute type
AssociatedMotion	LinkMotionRelationship
SubStructure	OPTIONAL SpecifiedOccurrence

Table 17: "ProductStructureLinkMotionAssociation" Attributes

Attribute recommendations

- **AssociatedMotion**: the LinkMotionRelationship that specifies the GeometricModel and the motion path.

- **SubStructure:** the SpecifiedOccurrence to which the kinematic path applies. The value of this attribute need not be specified. Use "SpecifiedOcurrence" template (see [242-PAS]).

Entity LinkMotionRelationship attributes	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL Id
Related	ComposedOrExternalGeometricModelSelect
RelationType	OPTIONAL ClassSelect
TransformationOperator	LinkMotionTransformation
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment

Table 18: "LinkMotionRelationship" Attributes

Attribute recommendations

- **Description**: the text or the set of texts that provide further information about the Link-MotionRelationship: need not be specified. Use "Description" template (see [242-PAS] 4.6.7).

- **Id**: the identifier or set of identifiers for the LinkMotionRelationship: need not be specified. Use "Identifier" template (see [242-PAS] 4.6.6).

- *Related:* the GeometricModel.

- *RelationType*: the meaning of the relationship. Use ClassString type if one of the values below is used, otherwise use "Class" template (see [242-PAS] 4.6.4): need not be specified.

- **TransformationOperator:**: the LinkMotionTransformation that specifies the parameter or time dependent transformation of the link geometry within the LinkMotionAlongPath.

- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

Entity LinkMotionTransformation attributes	Attribute type
TransformItem1	AxisPlacement
TransformItem2	KinematicPathDefinedByNodes

Table 19: "LinkMotionTransformation" Attributes

Attribute recommendations

- **TransformItem1**: the AxisPlacement that is the operator of the transformation.

- **TransformItem2**: the KinematicPathDefinedByNodes object that is being transformed.

Preprocessor Recommendation:

- ProductStructureLinkMotionAssociation.AssociatedMotion and ProductStructureLinkMotionAssociation.AssociatedTo shall reference to the same AssemblyDefinition.
- The AxisPlacement for TransformItem1 shall be in the same GeometricCoordinateSystem than the KinematicPathDefinedByNodes referenced by TransformItem2
- The AxisPlacement for TransformItem1 shall have neither a rotation nor a translation component.
- If SubStructure is unset, the kinematic path applies to the SingleOccurrence attached as 'Related' to the AssemblyOccurrenceRelationship (NextAssemblyOccurrenceUsage) in which the current ProductStructureLinkMotionAssociation is embedded. If set, the kinematic path applies to one particular SpecifiedOccurrence of one of its components.
  Remark: SpecifiedOccurrences are mostly not supported by the PDM systems => currently out of scope
- The GeometricModel for LinkMotionRelationship.Related can be either an ExternalGeometricModel (of a piece part) or a ComposedGeometricModel (of an assembly). It has to be one of the GeometricModels associated via DefiningGeometry or via AuxiliaryGeometry to the PartView/AssemblyDefinition where the SingleOccurrence/SpecifiedOccurrence belongs to.

Therefore, the GeometricCoordinateSystem of the GeometricModel maybe different from the GeometricCoordinateSystem of the LinkMotionAlongPath, but this has no effect since the positioning of the GeometricModel is done via the PlacementStart.

4.13 Kinematic Statistics and Validation Properties

As the development and testing of Kinematic capabilities is just starting, no established statistics for testing or validation properties are available yet. It is a task for the test rounds in the involved implementor forums to define and improve these.

This section is intended to gather initial ideas for such statistics and properties, and to facilitate the discussion of these across the two groups.

Note: like in [242-PAS] 6.2, It is recommended that all the properties attached to an object are spread over two instances of PropertyValueAssignment. One instance shall collect the properties that describe the object (having PropertyValueAssignment.ClassifiedAs='<object> properties'); the other instance shall collect the properties that describe the validation properties of the same object (having PropertyDefinition.PropertyType='kinematic validation property' and PropertyValueAssignment.ClassifiedAs='validation properties').

Note: For some of the proposed validation properties, it is necessary to assign properties to some Relationship or specific objects (like PathSegment, ProductStructureKinematicPathAssociation). It is currently unclear if and how this can be supported in the AP242 Domain Model XML schema (see section 1.1.2).

4.13.1 Statistics and Validation Properties for Kinematic Motion

4.13.1.1 Number of Kinematic Motions

For each assembly part having Motions, calculate the number of Kinematic Motions by counting the MotionModelAssociations in the STEP file. These define alternative/variant motions.

For use as a validation property, store the result in a property called 'number of kinematic motions', similar to 'number of children', attached to the AssemblyDefinition.

As all properties that are counts, this is intended as a completeness check.

A value 0 shall be omitted and no validation property mapped.

4.13.1.2 Number of Kinematic Paths per Motion

For each LinkMotionAlongPath, calculate the number of Kinematic Paths by counting the instances of KinematicPathDefinedByNodes orchestrated within the same Motion.

For use as a validation property, store the result in a property called 'number of kinematic paths', attached to the LinkMotionAlongPath. A value 0 shall not be omitted and shall be mapped as validation property.

4.13.1.3 Number of Steps per Kinematic Path

For each KinematicPathByNodes, calculate the number of steps by counting the instances of PointToPointPath.

For use as a validation property, store the result in a property called 'number of kinematic steps', attached to the KinematicPathDefinedByNodes.

A value 0 shall not be omitted and shall be mapped as validation property.

4.13.1.4 Volume of the related geometric model

Total volume of the moving parts within the assembly node linked to the ProductStructureKinematicPathAssociation or ProductStructureLinkMotionAssociation. This is intended to ensure that the right parts of the assembly are moving, similar to the "affected geometry" validation property for PMI.

An issue has been created on ISO Jira as TCSC410303-836 to assign a property value to ProductStructureKinematicPathAssociation and ProductStructureLinkMotionAssociation.

4.13.1.5 Centroid and Length of the Notional Kinematic Path Polyline

The "notional kinematic path polyline" is a virtual polyline, built from the KinematicPathDefinedByNodes as follows: For each axis system of the path, compute the point (10, 10, 10) and join all these points according to the path order. This definition is inspired by the definition of the notional centroid for assembly validation properties. It seems simple to compute and representative for the motion.

For use as validation properties,

- store the result in a property (as length measure) called 'length of notional kinematic path polyline', attached to the LinkMotionAlongPath

- store the result in a property (as cartesian point) called 'centroid of notional kinematic path polyline', attached to the LinkMotionAlongPath

Additionally, the Centroid of the part/assembly moved by the motion could be computed for each PathSegment. For use as a validation property, store the result in a property (as cartesian point) called 'centroid of motioned part/assembly', attached to the PathSegment.

An issue has been created on ISO Jira as TCSC410303-836 to assign a property value to PathSegment.

4.13.2 Statistics and Validation Properties for Kinematic Mechanisms

These validation properties shall be mapped similarly to the assembly validation properties defined in the section 13.1 in [242-PAS].

The PropertyDefinition.PropertyType shall be 'kinematics validation property' and the PropertyDefinition.Id shall be 'quality property'.

If an object has several validation properties (for example number of children and some kinematics validation property), both PropertyValues shall be associated to the same PropertyValueAssignment (with Class 'validation properties').

4.13.2.1 Number of Kinematic Mechanisms

For each assembly part having Mechanisms, calculate the number of Mechanisms by counting the instances of KinematicMechanismAssociations. These define alternative/variant mechanisms and may all apply together.

For use as a validation property, store the result in a property called 'number of kinematic mechanisms', attached to the AssemblyDefinition

A value 0 shall be omitted and no validation property mapped.

The Instance Model: STEP exchange file format (ISO10303 AP242 Domain Model XML syntax)

```
<Part uid="ID_62">
…
  <Versions>
    <PartVersion uid="ID_64">
…
      <Views>
        <PartView uid="ID_65" xsi:type="bom:AssemblyDefinition">
          <InitialContext uidRef="ID_60"/>
          <ShapeDependentProperty uid="ID_106" xsi:type="bom:CentreOfMass">
```

```xml
            <DefinedIn uidRef="ID_108"/>
            <Id id="notional solids centroid"/>
            <Role>
              <ClassString>assembly validation property</ClassString>
            </Role>
            <CentrePoint uidRef="ID_107"/>
          </ShapeDependentProperty>
          <PropertyValueAssignment uid="ID_66">
            <AssignedPropertyValues>
…
              <PropertyValue uid="ID_59-1" xsi:type="bom:NumericalValue">
                <Definition>
                  <PropertyDefinition uidRef="ID_57-1"/>
                </Definition>
                <Name>
                  <CharacterString>number of kinematic mechanisms</CharacterString>
                </Name>
                <Unit uidRef="ID_x17d1fa886a0"/>
                <ValueComponent>6</ValueComponent>
              </PropertyValue>
            </AssignedPropertyValues>
            <ClassifiedAs>
              <Classification uidRef="ID_58"/>
            </ClassifiedAs>
          </PropertyValueAssignment>
…
          <ViewOccurrenceRelationship uid="ID_73" xsi:type="bom:NextAssemblyOccurrenceUsage">
…
        </PartView>
      </Views>
…
    </PartVersion>
  </Versions>
</Part>
<PropertyDefinition uid="ID_57">
  <Id id="quality property"/>
  <PropertyType>
    <ClassString>assembly validation property</ClassString>
```

```xml
    </PropertyType>
  </PropertyDefinition>


<Unit uid="ID_x17d1fa886a0">
  <Kind>
    <ClassString>unspecified</ClassString>
  </Kind>
  <Name>
    <ClassString>each</ClassString>
  </Name>
  <Quantity>
    <ClassString>number of elements</ClassString>
  </Quantity>
</Unit>


<Classification uid="ID_58">
  <Class>
    <ClassString>validation properties</ClassString>
  </Class>
  <Role>kind of properties</Role>
</Classification>


<PropertyDefinition uid="ID_57-1">
  <Id id="quality property"/>
  <PropertyType>
    <ClassString>kinematics validation property</ClassString>
  </PropertyType>
</PropertyDefinition>
```


4.13.2.2 Number of Kinematic Pairs per Mechanism

For each Mechanism, calculate the number of Low/High Order Kinematic Pairs by counting the instances of LowOrderKinematicPair combined within the same mechanism.

For use as a validation property, store the result in two properties called 'number of low order kinematic pairs' and 'number of high order kinematic pairs', attached to the Mechanism.

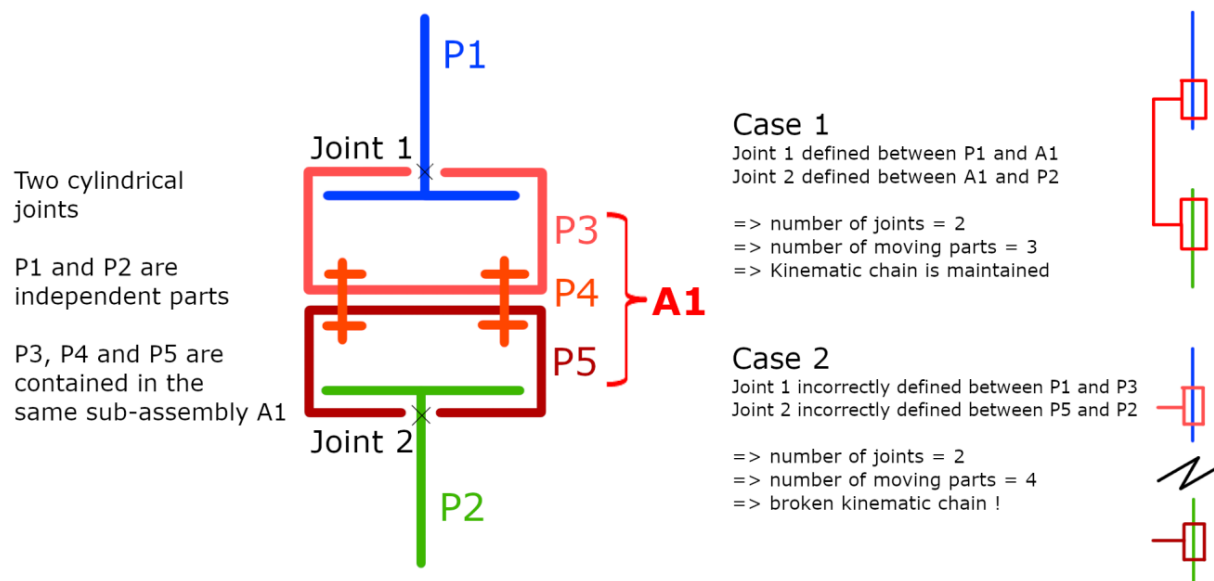A value 0 shall not be omitted and shall be mapped as validation property.

4.13.2.3 Number of Moving KinematicLinks per Mechanism

For each Mechanism, calculate the number of moving parts by counting the instances of KinematicLink involved via KinematicPairs within the same mechanism (except the one used as KinematicMechanismAssociation.BaseLink, see 4.4).

If each KinematicLink is associated to one (Single/Specified)Occurrence, since one Occurrence shall be associated at most to one KinematicLink per Mechanism (see 4.2), these define also the number of moving parts/assemblies. But in case KinematicLinks are associate to multiple (Single/Specified)Occurrence (see complex case of rigid groups in section 4.5), even more parts/assemblies will be moving.

For use as a validation property, store the result in a property called 'number of moving KinematicLinks', attached to the Mechanism.

Here an example of a wrong import (the kinematic pairs are created on the single parts P3/P5 instead of on an assembly A1) where this validation property helps detect the problem:

A value 0 shall not be omitted and no validation property mapped.

4.13.2.4 Number of Actuations per Mechanism

For each Mechanism, calculate the number of actuations over all KinematicPairs, by counting those attributes (Rx/y/z and Tx/y/z) in KinematicPair.Actuation that have a non-zero value (see 4.3). This captures the pairs where an initial movement can occur.

For use as a validation property, store the result in a property called 'number of actuations', attached to the Mechanism.

A value 0 shall not be omitted and shall be mapped as validation property.

4.13.2.5 Number of KinematicPairs for each kind of KinematicPair per Mechanism

For each Mechanism, calculate the number of KinematicPairs for each kind of KinematicPair according to the recommended values for LowOrderKinematicPair.Kind, HighOrderKinematicPair.Kind and LowOrderKinematicPairWithMotionCoupling.Kind.

The name of the validation property shall be overtaken from the attribute Kind (revolute_pair, spherical_pair, …), for example 'number of revolute_pairs', 'number of spherical_pairs', …).

 A value 0 shall be omitted and shall not be mapped as validation property.

The Instance Model: STEP exchange file format (ISO10303 AP242 Domain Model XML syntax)

```xml
<RepresentationContext      uid="ccs--origin"      xsi:type="bom:GeometricCoordi-
nateSpace">
  <Id id="/NULL"/>
  <Representations>
…
    <Representation xsi:type="bom:Mechanism" uid="kin--00000198446F1EF0--m">
      <Id id="/NULL"/>
      <Items>
…
      </Items>
      <Name>
…
      </Name>
      <PropertyValueAssignment uid="ID_66-2">
        <AssignedPropertyValues>
          <PropertyValue uid="ID_59-2" xsi:type="bom:NumericalValue">
            <Definition>
              <PropertyDefinition uidRef="ID_57-1"/>
            </Definition>
            <Name>
              <CharacterString>number of low order kinematic pairs</Character-
String>
            </Name>
            <Unit uidRef="ID_x17d1fa886a0"/>
            <ValueComponent>16</ValueComponent>
          </PropertyValue>
          <PropertyValue uid="ID_59-3" xsi:type="bom:NumericalValue">
            <Definition>
              <PropertyDefinition uidRef="ID_57-1"/>
            </Definition>
            <Name>
              <CharacterString>number of high order kinematic pairs</Charac-
terString>
            </Name>
            <Unit uidRef="ID_x17d1fa886a0"/>
            <ValueComponent>0</ValueComponent>
          </PropertyValue>
          <PropertyValue uid="ID_59-4" xsi:type="bom:NumericalValue">
            <Definition>
              <PropertyDefinition uidRef="ID_57-1"/>
            </Definition>
```

```xml
        <Name>
          <CharacterString>number of moving parts</CharacterString>
        </Name>
        <Unit uidRef="ID_x17d1fa886a0"/>
        <ValueComponent>12</ValueComponent>
      </PropertyValue>
      <PropertyValue uid="ID_59-5" xsi:type="bom:NumericalValue">
        <Definition>
          <PropertyDefinition uidRef="ID_57-1"/>
        </Definition>
        <Name>
          <CharacterString>number of actuations</CharacterString>
        </Name>
        <Unit uidRef="ID_x17d1fa886a0"/>
        <ValueComponent>2</ValueComponent>
      </PropertyValue>
      <PropertyValue uid="ID_59-6" xsi:type="bom:NumericalValue">
        <Definition>
          <PropertyDefinition uidRef="ID_57-1"/>
        </Definition>
        <Name>
          <CharacterString>number of revolute_pairs</CharacterString>
        </Name>
        <Unit uidRef="ID_x17d1fa886a0"/>
        <ValueComponent>2</ValueComponent>
      </PropertyValue>
…
    </AssignedPropertyValues>
    <ClassifiedAs>
      <Classification uidRef="ID_58"/>
    </ClassifiedAs>
  </PropertyValueAssignment>
 </Representation>
…
 </Representations>
 <Items>
…
 </Items>
 <DimensionCount>3</DimensionCount>
</RepresentationContext>
```

Annex A   List of Templates

This chapter gives a summary of the templates described in [242-PAS]:

Name	Paragraph
Template "ExchangeContext"	4.6.1
Template "Organization"	4.6.2
Template "Unit"	4.6.3
Template "Class"	4.6.4
Template "Classification"	4.6.5
Template "Identifier"	4.6.6
Template "Description"	4.6.7
Template "ViewContext"	4.6.8
Template "NumericalValue"	4.6.9
Template "StringValue"	4.6.10
Template "DateTime"	4.6.11
Template "Approval"	4.6.12
Template "Person"	4.6.13
Template "PersonInOrganization"	4.6.14
Template "Part"	5.1
Template "Assembly"	5.1
Template "GeometricModel"	6.1
Template "PartProperty"	6.2
Template "ShapeDependentProperty"	6.3
Template "SingleOccurrence"	7.1
Template "SpecifiedOccurrence"	7.2
Template "Simplified Positioning Representation"	7.3.1
Template "Document"	8.1
Template "DigitalFile"	9.1
Template "FormatProperty"	10.1
Template "ContentProperty"	10.2
Template "CreationProperty"	10.3
Template "SizeProperty"	10.4
Template "DocumentFileProperty"	10.5
Template "CAx Representation for DocumentAssignment"	11.1
Template "PDM Representation for DocumentAssignment"	11.2
Template "GeneralProperty"	12.2

Annex B   Known Issues

This section lists known issues with the AP242 Domain Model, both related to the assembly structure and to other domains within the Domain Model. These issues concern errors in the XSD, mismatches between the EXPRESS and XML schemas, deficiencies in the documentation and other issues that have already been communicated to the AP242 maintenance / development team for resolution. Many of these have been resolved by the Technical Corrigendum 1 (TC1) of AP242, which was published in 2016, by the Ed.2 (2019) and by the Ed.3 (2022). The table below lists only Kinematic-relevant issues which are yet unresolved. For issues against the AP242 Domain Model in general, please refer to [242-PAS, Annex B].

Since 2022, the Bugzilla list hosted by PDES Inc. has been transferred to ISO JIRA with new issue numbers. The old Bugzilla issue numbers are still mentioned in the short description.

ISO Jira Issue	Summary	Target Milestone	Status
TCSC410303-817	Constraints on KinematicLinkToOccur-renceAssociation (Bugzilla #6271)	10303:4442 (Ed4)	In Review
BS10303-5564	Add Link3+Link4 to LowOrderKinematicPairWithMotionCoupling (Bugzilla #7908)	242-3-CD	Under Discussion
BS10303-6167	Add attribute SubStructure to KinematicLinkToOccurrenceAssociation (Bugzilla #8578)	242-3-CD	Open
TCSC410303-836	Add ProductStructureKinematicPathAssociation and ProductStructureLinkMotionAssociation to PropertyAssignmentSelect (Bugzilla #9073)	10303:4442 (Ed4)	In Review

Annex C   Reference Documents

This recommended practices document is based on and derived from various other documents, schemas, and technical presentations. Those resources are listed below:

- Recommended Practices for AP242 Ed.3 Domain Model XML Product & Assembly Structure
    - Version 3.1; dated 18 November 2022
- AP242 Ed.3 Domain Model XML / EXPRESS Schema
    - Dated 11 May 2021 (version N10590)