INTERNATIONAL STANDARD

ISO 10303-109

First edition
2004-12-01

Industrial automation systems and integration — Product data representation and exchange —

Part 109:
Integrated application resource: Kinematic and geometric constraints for assembly models

Systèmes d'automatisation industrielle et intégration — Représentation et échange de données de produits —

Partie 109: Ressources d'application intégrées: Contraintes cinématiques et géométriques pour les modèles d'assemblage

Reference number
ISO 10303-109:2004(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

1 Scope 1

2 Normative references 2

3 Terms, definitions, and abbreviations 3
3.1 Terms defined in ISO 10303-1 3
3.2 Terms defined in ISO 10303-11 3
3.3 Terms defined in ISO 10303-41 3
3.4 Terms defined in ISO 10303-42 3
3.5 Terms defined in ISO 10303-43 3
3.6 Terms defined in ISO 10303-44 4
3.7 Terms defined in ISO 10303-108 4
3.8 Abbreviations 4

4 Assembly feature relationship 5
4.1 Introduction 5
4.2 Fundamental concepts and assumptions 6
4.3 Assembly feature relationship type definitions 7
4.3.1 Representing relationship 7
4.4 Assembly feature relationship entity definitions 7
4.4.1 Shape aspect relationship representation association 7
4.4.2 Representative shape representation 9
4.4.3 Free kinematic motion representation 10
4.4.4 Constrained kinematic motion representation 12
4.5 Assembly feature relationship function definitions 13
4.5.1 Assembly root 13
4.5.2 Find assembly root 14
4.5.3 Find shape representation of product definition 16
4.5.4 Find shape representation of shape aspect 17
4.5.5 Find representative shape representation of product definition 18
4.5.6 Find representative shape representation of shape aspect 18
4.5.7 Unique in product definition 19
4.5.8 Unique in shape aspect 20
4.5.9 Using product definition of shape aspect 21
4.5.10 Using product definition of shape representation 22
4.5.11 Using shape aspect of shape representation 23
4.5.12 Using representations with mapping 24

5 Assembly constraint 26
5.1 Introduction 26
5.2 Fundamental concepts and assumptions 27

5.3 Assembly constraint entity definitions 29
5.3.1 Assembly geometric constraint...29
5.3.2 Binary assembly constraint ……………………………………………………….29
5.3.3 Fixed constituent assembly constraint 31
5.3.4 Parallel assembly constraint 32
5.3.5 Parallel assembly constraint with dimension...33
5.3.6 Surface distance assembly constraint with dimension..33
5.3.7 Angle assembly constraint with dimension 34
5.3.8 Perpendicular assembly constraint 34
5.3.9 Incidence assembly constraint 35
5.3.10 Coaxial assembly constraint 35
5.3.11 Tangent assembly constraint...36
5.4 Assembly constraint function definitions 36
5.4.1 Assembly leaf 36

Annex A (normative) Short names of entities...38

Annex B (normative) Information object registration..39
B.1 Document identification 39
B.2 Schema identification 39
B.2.1 Identification of the schema assembly_feature_relationship……………………….39
B.2.2 Identification of the schema assembly_constraint…………………………….……..39

Annex C (informative) Computer-interpretable listings 40

Annex D (informative) EXPRESS-G diagrams 41

Annex E (informative) Informative figures……………………………………………………...44

Index...46

Figures

Figure 1 Schema level diagram of relationships between ISO 10303-109 schemas (inside the box) and other resource schemas……………………………………....………viii
Figure D.1 EXPRESS-G diagram of the assembly_feature_relationship_schema (1 of 1)……….....42
Figure D.2 EXPRESS-G diagram of the assembly_constraint_schema (1 of 1)…………………43
Figure E.1 Relationships of schemas in this part of ISO 10303 and related modules with existing resource entities (overall structure)…………………………...……………………45

Tables

Table A.1 Short names of entities…………………………………………………………38

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75% of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this part of ISO 10303 may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 10303-109 was prepared by Technical Committee ISO/TC184, *Industrial automation systems and integration*, Subcommittee SC4, *Industrial data*.

ISO 10303 is organized as a series of parts, each published separately. The structure of ISO 10303 is described in ISO 10303-1.

Each part of ISO 10303 is a member of one of the following series: description methods, implementation methods, conformance testing methodology and framework, integrated generic resources, integrated application resources, application protocols, abstract test suites, application interpreted constructs, and application modules. This part is a member the of integrated application resources series. The integrated generic resources and the integrated application resources specify a single conceptual product data model.

A complete list of parts of ISO 10303 is available from the Internet:

`<http://www.tc184-sc4.org/titles/>`

Introduction

ISO 10303 is an International Standard for the computer-interpretable representation of product information and for the exchange of product data. The objective is to provide a neutral mechanism capable of describing products throughout their life cycle. This mechanism is suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases, and as a basis for archiving.

This part of ISO 10303 is a member of the integrated application resources series. Major subdivisions of this part of ISO 10303 are:

— Assembly feature relationship schema;

— Assembly constraint schema.

This part of ISO 10303 provides general representations for geometric relationships between component parts within an assembly model. The distinction of assembly, subassembly and component part is context dependent. An assembly in some engineering context could be a component part in another engineering context. In order to cope with this variety, this part of ISO 10303 uses the term 'constituent' to represent a generic concept that implies assembly or subassembly or component part in case these distinctions are not necessary.

EXAMPLE 1 For some applications, an electric motor is regarded as a single component part of a washing machine rather than as an assembly in its own right.

Detailed geometric relationships between constituents of an assembly are required in applications such as parametric representation of geometric constraints between constituents, motion animation of an assembly product, assembly/disassembly process planning, kinematics analysis and tolerance analysis.

Existing STEP resource parts support limited assembly model representations. They capture hierarchical relationship, alternative constituent and mutual position and orientation between two constituents. This part of ISO 10303 is intended to fill in missing information to enable the above mentioned applications.

An important concept newly introduced in this part of ISO 10303 is 'assembly feature'. An assembly feature is a portion of a constituent meaningful for representing the connecting relation between constituents. The detailed geometric relationship between two constituents can be represented via the necessary number of pairs of assembly features one belonging to one constituent and the other belonging to the other constituent. The **assembly_feature_relationship_schema** has been created to capture feature level correspondence between constituents.

In most assembly related applications, not only correspondence of assembly features but also more detailed geometric constraint information such as parallelism, coincidence, tangency, and co-axial relationships are required at the geometric entity level.

EXAMPLE 2 Two plates belonging to different constituents are constrained to be parallel with distance-L.

These geometric constraint specifications applied between two constituents are summarised in the **assembly_constraint_schema**. In assembly related applications, the position and orientation of at least one constituent within an assembly model should be fixed to prevent infinite number of solutions. This constituent plays the role of an anchor. The necessary constraint is also included in the **assembly-_constraint_schema**.

This part of ISO 10303 is a member of a set of standards newly introduced for representing detailed geometric relationships among constituents of an assembly model of a product.
They are:

— ISO 10303-109: Integrated application resources: Kinematic and geometric constraints for assembly models
— ISO 10303-1101: Application Module: Product property feature definition module
— ISO 10303-1102: Application Module: Assembly feature definition module

The relationships between schemas in this part of ISO 10303 and existing integrated resource schemas of ISO 10303 are illustrated in Figure 1. The schemas occurring in Figure 1 are components of ISO 10303 integrated resources, and they are specified in the following resource parts:

`explicit_constraint_schema`	ISO 10303-108
`explicit_geometric_constraint_schema`	ISO 10303-108
`geometric_model_schema`	ISO 10303-42
`geometry_schema`	ISO 10303-42
`kinematic_motion_representation_schema`	ISO 10303-105
`kinematic_structure_schema`	ISO 10303-105
`parameterization_schema`	ISO 10303-108
`product_definition_schema`	ISO 10303-41
`product_property_definition_schema`	ISO 10303-41
`product_property_representation_schema`	ISO 10303-41
`product_structure_schema`	ISO 10303-44
`representation_schema`	ISO 10303-43

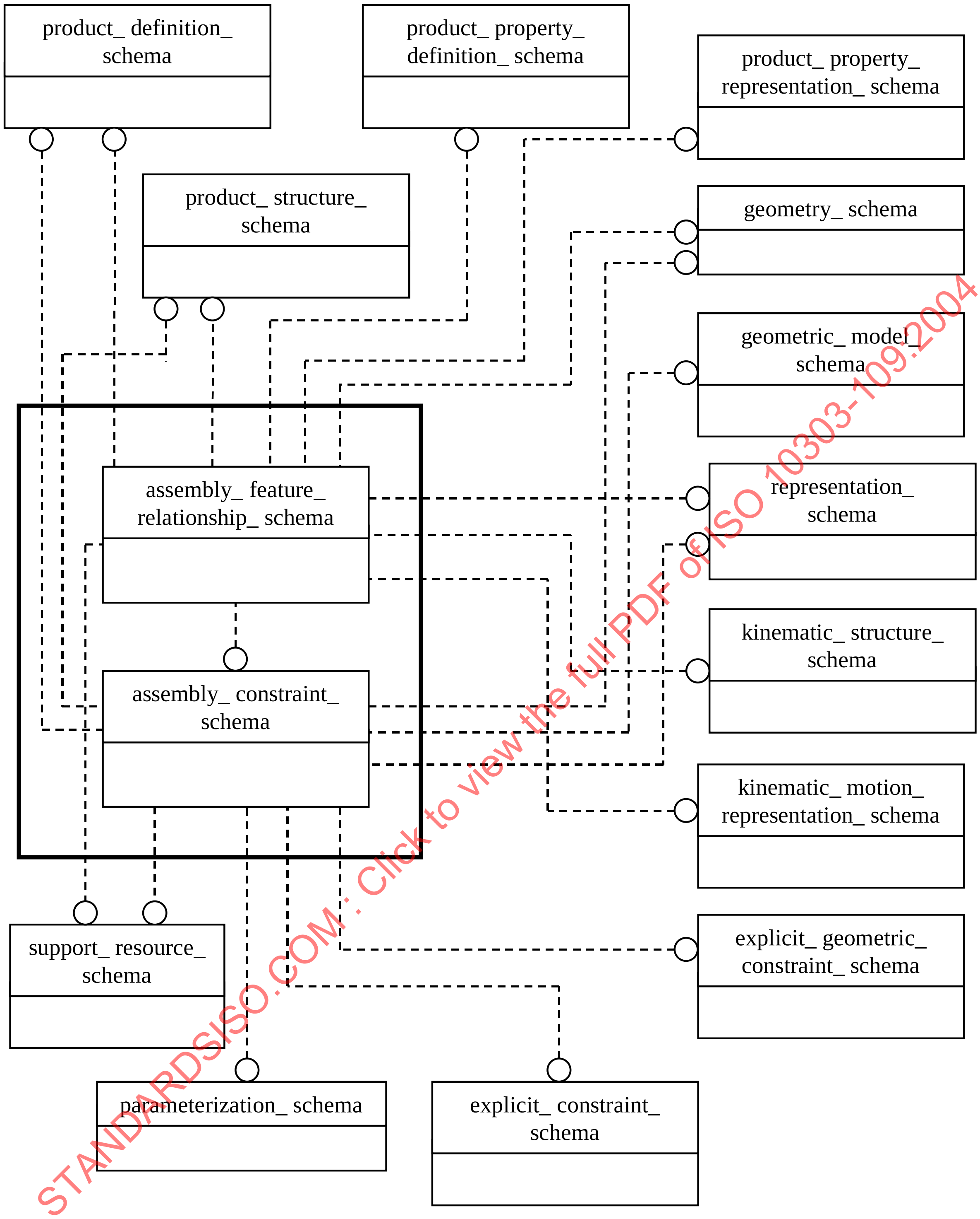

Figure 1 – Schema level diagram of relationships between ISO 10303-109 schemas (inside the box) and other resource schemas

Industrial automation systems and integration – Product data representation and exchange –

Part 109:

Integrated application resource: Kinematic and geometric constraints for assembly models

1 Scope

This part of ISO 10303 specifies the resource constructs for the representation of detailed geometric relationships between constituents of an assembly model including geometric constraints between constituents.

The following are within the scope of this part of ISO 10303:

— The association of shape aspect relationship with its representation;

— The association of instanced assembly feature relationship with its representation;

— The representation of detailed geometric information of instanced assembly feature relationship at the **geometric_representation_item** level in terms of elements such as assembly geometric constraints, kinematics pair and kinematics path;

— The representation of the fixed constituent which plays the anchor role in the assembly model.

The following are outside the scope of this part of ISO 10303:

— Product structure configurations of assemblies and their constituents.

— Tolerance information. See clause 5.2 of this part of ISO 10303.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 8824-1, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*.

ISO 10303-1, *Industrial automation systems and integration – Product data representation and exchange – Part 1: Overview and fundamental principles*.

ISO 10303-11, *Industrial automation systems and integration – Product data representation and exchange – Part 11: Description methods: The EXPRESS language reference manual*.

ISO 10303-41, *Industrial automation systems and integration – Product data representation and exchange – Part 41: Integrated generic resource: Fundamentals of product description and support*.

ISO 10303-42, *Industrial automation systems and integration – Product data representation and exchange – Part 42: Integrated generic resource: Geometric and topological representation*.

ISO 10303-43, *Industrial automation systems and integration – Product data representation and exchange – Part 43: Integrated generic resource: Representation structures*.

ISO 10303-44, *Industrial automation systems and integration – Product data representation and exchange – Part 44: Integrated generic resource: Product structure configuration*.

ISO 10303-105, *Industrial automation systems and integration – Product data representation and exchange – Part 105: Integrated application resource: Kinematics*.

ISO 10303-108:2004, *Industrial automation systems and integration – Product data representation and exchange – Part 108: Integrated application resource: Parameterization and constraints for explicit geometric product models*.

3 Terms, definitions, and abbreviations

3.1 Terms defined in ISO 10303-1

For the purposes of this document, the following terms defined in ISO 10303-1 apply.

— assembly;

— component.

3.2 Terms defined in ISO 10303-11

For the purposes of this document, the following terms defined in ISO 10303-11 apply.

— entity;

— entity data type;

— instance.

3.3 Terms defined in ISO 10303-41

For the purposes of this document, the following term defined in ISO 10303-41 applies.

— shape aspect.

3.4 Terms defined in ISO 10303-42

For the purposes of this document, the following terms defined in ISO 10303-42 apply.

— boundary representation solid model (B-rep);

— constructive solid geometry (CSG);

— geometric representation item.

3.5 Terms defined in ISO 10303-43

For the purposes of this document, the following terms defined in ISO 10303-43 apply.

— representation;

— representation context.

3.6 Terms defined in ISO 10303-44

For the purposes of this document, the following terms defined in ISO 10303-44 apply.

— constituent;

— sub-assembly.

3.7 Terms defined in ISO 10303-108

For the purposes of this document, the following terms defined in ISO 10303-108 apply.

— constrained element;

— constraint;

— constraint solver;

— defined constraint;

— dimensional constraint;

— directed constraint;

— explicit geometric constraint;

— logical constraint;

— reference element;

— sketch;

— undirected constraint.

3.8 Abbreviations

AP application protocol (of ISO 10303)

B-rep boundary representation

4 Assembly feature relationship

4.1 Introduction

The following EXPRESS declaration begins the **assembly_feature_relationship_schema** and identifies the necessary references.

<u>EXPRESS specification:</u>

```
*)
SCHEMA assembly_feature_relationship_schema;

  REFERENCE FROM assembly_constraint_schema; -- 10303-109
  REFERENCE FROM geometry_schema; -- 10303-42
  REFERENCE FROM kinematic_motion_representation_schema; -- 10303-105
  REFERENCE FROM kinematic_structure_schema; -- 10303-105
  REFERENCE FROM product_definition_schema; -- 10303-41
  REFERENCE FROM product_property_definition_schema; -- 10303-41
  REFERENCE FROM product_property_representation_schema; -- 10303-41
  REFERENCE FROM product_structure_schema; -- 10303-44
  REFERENCE FROM representation_schema; -- 10303-43
  REFERENCE FROM support_resource_schema; -- 10303-41
(*
```

NOTE 1 The schemas referenced above can be found in the following parts of ISO 10303:

assembly_constraint_schema	ISO 10303-109
geometry_schema	ISO 10303-42
kinematic_motion_representation_schema	ISO 10303-105
kinematic_structure_schema	ISO 10303-105
product_definition_schema	ISO 10303-41
product_property_definition_schema	ISO 10303-41
product_property_representation_schema	ISO 10303-41
product_structure_schema	ISO 10303-44
representation_schema	ISO 10303-43
support_resource_schema	ISO 10303-41

NOTE 2 See annex D, Figure D.1, for a graphical presentation of this schema.

4.2 Fundamental concepts and assumptions

An important concept in this part of ISO 10303 is **assembly_feature**. It is a portion of a constituent used for defining a connectivity relationship between constituents of an assembly model. It is actually defined in ISO 10303-1102: Assembly feature definition module as an entity named **instanced-_assembly_feature** that is a subtype of **product_property_instanced_feature** defined in ISO 10303-1101: Product property feature definition module. As described in the introduction paragraph of this part of ISO 10303, these two modules in combination with this part of ISO 10303 support the concept of **assembly_feature**. See annex E, Figure E.1 for a graphical presentation of these relationships.

The **product_property_instanced_feature** itself is a subtype of **shape_aspect** defined in ISO 10303-41 as shown in annex E. The necessary number of pairs of **assembly_feature** instances one belonging to one constituent and the other belonging to the other constituent can represent the detailed geometric relationship between two constituents within an assembly model.

This schema provides resource constructs for linking **shape_aspect_relationship** with its corresponding **representation_relationship** and for detailing geometric information of the **representation_relationship**.

The entity **shape_aspect** defined in ISO 10303-41 is an identifiable element of the shape of a product, and is used to specify a portion of a product shape. The entity **shape_aspect_relationship** also defined in ISO 10303-41 relates two instances of **shape_aspect**.

EXAMPLE 1 If a product named “gear box” has a set of bearing holes for supporting a set of journals of a product named “shaft”, the bearing holes and the journals are **shape_aspect** instances. The instances of **shape_aspect** representing bearing holes may have some relationships with the instances of the **shape_aspect** representing journals. These relationships can be described by the use of **shape_aspect_relationship**.

The geometric shape of an instance of **shape_aspect** is represented by using an instance of **shape_-representation** defined in ISO 10303-41. If a pair of **shape_aspect** instances are related with each other, the **shape_representation** instance of one **shape_aspect** instance may be specified in the context of the **shape_representation** instance of the other **shape_aspect** instance. An entity data type **shape_aspect-_relationship_representation_association** is introduced to relate an instance of **shape_aspect_-relationship** with an instance of **representation_relationship** representing the geometric relationship between two **shape_aspect** instances.

EXAMPLE 2 In the above example consisting of a gear box and a shaft, **shape_aspect_relationship_-representation_association** is used to identify the **representation_relationship** representing the relative position and orientation between a bearing hole and a journal.

As for the representation of detailed geometric information of **instanced_assembly_feature_-relationship**, this schema enables the selection from among **binary_assembly_constraint**, **constrained_kinematic_motion_representation** and **free_kinematic_motion_representation**.

4.3 Assembly feature relationship type definitions

4.3.1 Representing relationship

The **representing_relationship** type is used to distinguish those major subtypes of **instanced_-assembly_feature_relationship_representation_association** which are **binary_assembly_-constraint, constrained_kinematic_motion_representation**, and **free_kinematic_motion_-representation**.

EXPRESS specification:

```
*)
TYPE representing_relationship = SELECT
(binary_assembly_constraint,
constrained_kinematic_motion_representation,
free_kinematic_motion_representation);
END_TYPE;-- representing_relationship
(*
```

4.4 Assembly feature relationship entity definitions

4.4.1 Shape aspect relationship representation association

A **shape_aspect_relationship_representation_association** identifies an instance of **representation_-relationship** representing the geometric information of an instance of **shape_aspect_relationship**. This entity is used to describe the relative position and orientation between a pair of **shape_aspect** instances.

EXPRESS specification:

```
*)
ENTITY shape_aspect_relationship_representation_association;
  represented_shape_aspect_relationship: shape_aspect_relationship;
  representing_representation_relationship : representation_relationship;
WHERE
WR1: ('REPRESENTATIVE_SHAPE_REPRESENTATION' IN
      TYPEOF(representing_representation_relationship\
      representation_relationship.rep_1)) AND
     ('REPRESENTATIVE_SHAPE_REPRESENTATION' IN
      TYPEOF(representing_representation_relationship\
      representation_relationship.rep_2));
```

```
WR2: (represented_shape_aspect_relationship.relating_shape_aspect IN
      using_shape_aspect_of_shape_representation
     (representing_representation_relationship.rep_1))
     AND(represented_shape_aspect_relationship.related_shape_aspect IN
      using_shape_aspect_of_shape_representation
    (representing_representation_relationship.rep_2));
WR3: ((find_representative_shape_representation_of_product_definition
     (using_product_definition_of_shape_aspect
     (represented_shape_aspect_relationship.relating_shape_aspect)).
      context_of_items) :=:
     (find_representative_shape_representation_of_shape_aspect
     (represented_shape_aspect_relationship.relating_shape_aspect).
      context_of_items)) AND
     ((find_representative_shape_representation_of_product_definition
     (using_product_definition_of_shape_aspect
     (represented_shape_aspect_relationship.related_shape_aspect)).
      context_of_items) :=:
     (find_representative_shape_representation_of_shape_aspect
     (represented_shape_aspect_relationship.related_shape_aspect).
      context_of_items));
WR4: using_product_definition_of_shape_aspect
     (represented_shape_aspect_relationship.relating_shape_aspect) :<>:
     using_product_definition_of_shape_aspect
     (represented_shape_aspect_relationship.related_shape_aspect);
WR5: find_assembly_root ([using_product_definition_of_shape_aspect
     (represented_shape_aspect_relationship.relating_shape_aspect)]) :=:
     find_assembly_root ([using_product_definition_of_shape_aspect
     (represented_shape_aspect_relationship.related_shape_aspect)]);
END_ENTITY; -- shape_aspect_relationship_representation_association
(*
```

Attribute definitions:

represented_shape_aspect_relationship: An instance of **shape_aspect_relationship**.

representing_representation_relationship: An instance of **representation_relationship** that represents the geometric information of the **shape_aspect_relationship** instance specified by **represented_shape_aspect_relationship**.

Formal propositions:

WR1: The **representing_representation_relationship** shall be a relationship between a pair of **representative_shape_representation** instances.

WR2: The **relating_shape_aspect** of the **shape_aspect_relationship** specified by the first attribute shall be represented by an instance of **representative_shape_representation** indicated by **rep_1** of the **representing_representation_relationship** specified by the second attribute**.** The **related_shape_aspect** of the **shape_aspect_relationship** specified by the first attribute shall be represented by an instance of **representative_shape_representation** indicated by **rep_2** of the **representing_representation_relationship** specified by the second attribute.

WR3: Two instances of **representative_shape_representation** representing an instance of **shape_aspect** and an instance of **product_definition** respectively shall have the same value for **context_of_items**.

WR4: The **relating_shape_aspect** and the **related_shape_aspect** of an instance of **shape_aspect_relationship** shall not belong to the same instance of **product_definition**.

WR5: Two **product_definition** instances, which include the **relating_shape_aspect** instance and the **related_shape_aspect** instance of an instance of **shape_aspect_relationship** respectively, are required to be constituents of the same assembly.

4.4.2 Representative shape representation

A **representative_shape_representation** is a subtype of **shape_representation** that is a representative shape of a **product_definition** or a **shape_aspect**.

NOTE A 'representative shape' is a **shape_representation** that represents the referenced **product_definition** or **shape_aspect** with the most detailed geometric representation. Identification of the most detailed geometric representation becomes necessary as this resource part represents the geometric information of connectivity between constituents at the most detailed **geometric_representation_item** level, and as one **product_definition** or one **shape_aspect** may be represented by two or more **shape_representations**.

EXPRESS specification:

```
*)
ENTITY representative_shape_representation
  SUBTYPE OF (shape_representation);
WHERE
WR1: unique_in_product_definition (SELF) OR unique_in_shape_aspect (SELF);
END_ENTITY; -- representative_shape_representation
(*
```

Formal propositions:

WR1: If existing, a **representative_shape_representation** shall be unique for the corresponding **shape_aspect** or **product_definition**.

4.4.3 Free kinematic motion representation

A **free_kinematic_motion_representation** represents a free kinematics motion in three dimensional space of an assembly feature with respect to another assembly feature. The motion is represented by a relative kinematics motion between a pair of **geometric_representation_context** instances of the **shape_representation** instances representing the assembly features. Detailed information about the constraints is described by applying the entity **kinematic_path** defined in ISO 10303-105.

EXPRESS specification:

```
*)
ENTITY free_kinematic_motion_representation
  SUBTYPE OF (representation_relationship_with_transformation);
  SELF\representation_relationship.rep_1:
     representative_shape_representation;
  SELF\representation_relationship.rep_2:
     representative_shape_representation;
  motion : kinematic_path;
WHERE
WR1: 'ITEM_DEFINED_TRANSFORMATION' IN TYPEOF
     (SELF\representation_relationship_with_transformation.
      transformation_operator);
WR2: ('GEOMETRY_SCHEMA.GEOMETRIC_REPRESENTATION_ITEM' IN TYPEOF
     (SELF\representation_relationship_with_transformation.
     transformation_operator\item_defined_transformation.transform_item_1))
     AND('GEOMETRY_SCHEMA.GEOMETRIC_REPRESENTATION_ITEM' IN TYPEOF
     (SELF\representation_relationship_with_transformation.
    transformation_operator\item_defined_transformation.transform_item_2));
WR3: ((dimension_of
     (SELF\representation_relationship_with_transformation.
      transformation_operator\item_defined_transformation.
      transform_item_1\geometric_representation_item) = 3 ) AND
     (dimension_of
     (SELF\representation_relationship_with_transformation.
      transformation_operator\item_defined_transformation.
      transform_item_2\geometric_representation_item) = 3 ));
```



```
WR4:(SELF\representation_relationship.rep_1 IN
     (using_representations
     (SELF\representation_relationship_with_transformation.
      transformation_operator\item_defined_transformation.transform_item_1)
      + using_representation_with_mapping
     (SELF\representation_relationship_with_transformation.
    transformation_operator\item_defined_transformation.transform_item_1)))
     AND(SELF\representation_relationship.rep_2 IN
     (using_representations
     (SELF\representation_relationship_with_transformation.
      transformation_operator\item_defined_transformation.transform_item_2)
      + using_representation_with_mapping
     (SELF\representation_relationship_with_transformation.
   transformation_operator\item_defined_transformation.transform_item_2)));
END_ENTITY; -- free_kinematic_motion_representation
(*
```

Attribute definitions:

SELF\representation_relationship.rep_1: An instance of **shape_representation** of the type **representative_shape_representation** to which the transformation applies.

SELF\representation_relationship.rep_2: An instance of **shape_representation** of the type **representative_shape_representation** which is related to **rep_1** by the transformation.

motion: Description of the free kinematics motion. The motion is described by the transformation between two instances of **geometric_representation_item** belonging to two different instances of **shape_representation**.

Formal propositions:

WR1: **SELF\representation_relationship_with_transformation.transformation_operator** shall be of type **item_defined_transformation**.

WR2: Both **transform_item_1** and **transform_item_2** shall be of type **geometric_representation_-item**.

WR3: Both **transform_item_1** and **transform_item_2** shall have a dimensionality of three.

WR4: **transform_item_1** shall be used, directly or indirectly, by **rep_1** and **transform_item_2** shall be used, directly or indirectly, by **rep_2**.

4.4.4 Constrained kinematic motion representation

A **constrained_kinematic_motion_representation** represents a kinematics motion constrained by a kinematic pair between a pair of assembly features. The relationship is represented as a relative kinematics motion between a pair of **geometric_representation_item** instances belonging to two **representative_shape_representation** instances which represent target assembly features. Detailed information about the constraints is derived by applying the entity **kinematic_pair** defined in ISO 10303-105.

EXPRESS specification:

```
*)
ENTITY constrained_kinematic_motion_representation
  SUBTYPE OF (representation_relationship_with_transformation);
  SELF\representation_relationship.rep_1:
    representative_shape_representation;
  SELF\representation_relationship.rep_2:
    representative_shape_representation;
WHERE
WR1: ('GEOMETRY_SCHEMA.AXIS2_PLACEMENT_3D' IN TYPEOF
     (SELF\representation_relationship_with_transformation.
     transformation_operator\item_defined_transformation.transform_item_1))
     AND('GEOMETRY_SCHEMA.AXIS2_PLACEMENT_3D' IN TYPEOF
     (SELF\representation_relationship_with_transformation.
    transformation_operator\item_defined_transformation.transform_item_2));
WR2: ((dimension_of
     (SELF\representation_relationship_with_transformation.
     transformation_operator\item_defined_transformation.
     transform_item_1\geometric_representation_item) = 3 ) AND
     (dimension_of
     (SELF\representation_relationship_with_transformation.
     transformation_operator\item_defined_transformation.
     transform_item_2\geometric_representation_item) = 3 ));
WR3:(SELF\representation_relationship.rep_1 IN
     (using_representations
     (SELF\representation_relationship_with_transformation.
     transformation_operator\item_defined_transformation.transform_item_1)
      + using_representation_with_mapping
     (SELF\representation_relationship_with_transformation.
    transformation_operator\item_defined_transformation.transform_item_1)))
     AND(SELF\representation_relationship.rep_2 IN
     (using_representations
     (SELF\representation_relationship_with_transformation.
```



```
     transformation_operator\item_defined_transformation.transform_item_2)
      + using_representation_with_mapping
     (SELF\representation_relationship_with_transformation.
   transformation_operator\item_defined_transformation.transform_item_2)));
WR4: 'KINEMATIC_PAIR' IN TYPEOF
     (SELF\representation_relationship_with_transformation.
      transformation_operator);
END_ENTITY; -- constrained_kinematic_motion_representation
(*
```

Attribute definitions:

SELF\representation_relationship.rep_1: An instance of **shape_representation** of the type **representative_shape_representation** to which the transformation applies.

SELF\representation_relationship.rep_2: An instance of **shape_representation** of the type **representative_shape_representation** which is related to **rep_1** by the transformation.

Formal propositions:

WR1: The two **representation_item** instances specified by the inherited attribute **transformation_-operator** shall be of type **axis2_placement_3d**.

WR2: The two **geometric_representation_item** instances specified by the inherited attribute **transformation_operator** shall have a dimensionality of three.

WR3: **transform_item_1** shall be used, directly or indirectly, by **rep_1** and **transform_item_2** shall be used, directly or indirectly, by **rep_2**.

WR4: SELF\representation_relationship_with_transformation.transformation_operator shall be a **kinematic_pair** specifying a pair of **representation_items** for its representation.

4.5 Assembly feature relationship function definitions

4.5.1 Assembly root

The function **assembly_root** determines if an instance of **product_definition** is the root node in the tree structure defined by an instance of **assembly_component_usage** as defined in ISO 10303-44. This means that the instance of **product_definition** is a highest level constituent of an assembly.

EXPRESS specification:

```
*)
FUNCTION assembly_root
  (item: product_definition) : BOOLEAN;
-- extraction of related assembly_component_relationships --
IF (SIZEOF(USEDIN (item,
   'PRODUCT_STRUCTURE_SCHEMA.ASSEMBLY_COMPONENT_USAGE.' +
   'RELATED_PRODUCT_DEFINITION')) = 0) THEN RETURN(TRUE);
ELSE RETURN (FALSE);
END_IF;
END_FUNCTION;
(*
```

Argument definitions:

item: An instance of **product_definition** to be checked to ascertain whether it is a root node of an assembly structure.

4.5.2 Find assembly root

The function **find_assembly_root** finds a set of **product_definition** instances that are the common root of specified **product_definition** instances in the tree structure defined by instances of **assembly_-component_usage** as defined in ISO 10303-44.

EXPRESS specification:

```
*)
FUNCTION find_assembly_root
  (constituent: SET OF product_definition) : SET OF product_definition;
LOCAL
  local_relation: SET OF assembly_component_usage := [];
  local_relation2: BAG OF assembly_component_usage := [];
  local_parent: SET OF product_definition := [];
  root : SET OF product_definition;
  i : INTEGER := 0;
  j : INTEGER := 0;
END_LOCAL;

-- Is constituent a root ? --
IF ((SIZEOF (constituent) = 1) AND assembly_root (constituent[1]))
  THEN RETURN ([constituent [1]]);
-- ERROR constituent is vacant --
  ELSE IF (SIZEOF (constituent) = 0 ) THEN RETURN ([]);
```



```
-- extraction of related assembly_component_relationships --
    ELSE
      REPEAT j:= 1 TO HIINDEX(constituent);
        local_relation2 := local_relation2 + (USEDIN (constituent[j],
                    'PRODUCT_STRUCTURE_SCHEMA.ASSEMBLY_COMPONENT_USAGE.'
                    +'RELATED_PRODUCT_DEFINITION'));
      END_REPEAT;
      local_relation := bag_to_set (local_relation2);
      IF (SIZEOF(local_relation) = 0) THEN
        IF (SIZEOF(constituent) = 1) THEN RETURN ([constituent[1]]);
          ELSE RETURN ([]);
        END_IF;
      ELSE
-- extraction of a set of parents --
        REPEAT i :=1 TO HIINDEX(local_relation);
          REPEAT j := 1 TO HIINDEX(constituent);
          IF (local_relation[i].relating_product_definition <>
              constituent[j]) THEN
            local_parent := local_parent +
                            local_relation[i].relating_product_definition;
          END_IF;
          END_REPEAT;
        END_REPEAT;
        IF ((SIZEOF (local_parent) = 1 ) AND
             assembly_root (local_parent[1]))
          THEN RETURN ([local_parent[1]]);
          ELSE IF (SIZEOF (local_parent) = 0) THEN RETURN ([]);
-- try again --
            ELSE
              root := find_assembly_root(local_parent);
              IF (SIZEOF (root) =1) THEN RETURN (root);
                ELSE IF (SIZEOF (root) = 0) THEN RETURN ([]);
                END_IF;
              END_IF;
          END_IF;
        END_IF;
      END_IF;
  END_IF;
END_IF;
END_FUNCTION;
(*
```

Argument definitions:

constituent: A set of **product_definition** instances whose common root shall be sought in the tree structure defined by instances of **assembly_component_usage** as defined in ISO 10303-44.

4.5.3 Find shape representation of product definition

The function **find_shape_representation_of_product_definition** finds **shape_representation** instances representing the shape of a specified **product_definition** instance.

EXPRESS specification:

```
*)
FUNCTION find_shape_representation_of_product_definition
    (item:product_definition) : SET OF shape_representation;
LOCAL
  local_p_d_s: SET OF product_definition_shape := [];
  local_p_d_s2: BAG OF product_definition_shape := [];
  local_s_d_r: SET OF shape_definition_representation := [];
  local_s_d_r2: BAG OF shape_definition_representation := [];
  local_s_r: SET OF shape_representation := [];
  i : INTEGER;
END_LOCAL;
-- find product_definition_shape representing the product_definiton
  local_p_d_s2 := local_p_d_s2 + (USEDIN (item,
'PRODUCT_PROPERTY_DEFINITION_SCHEMA.PRODUCT_DEFINITION_SHAPE.DEFINITION'));
  local_p_d_s := bag_to_set(local_p_d_s2);
-- find shape_definition_representations refereing to the local_p_d_s
  REPEAT i := 1 to HIINDEX (local_p_d_s);
    local_s_d_r2 := local_s_d_r2 + (USEDIN (local_p_d_s[i],
                    'PRODUCT_PROPERTY_REPRESENTATION_SCHEMA.' +
                    'SHAPE_DEFINITION_REPRESENTATION.DEFINITION'));
  END_REPEAT;
  local_s_d_r := bag_to_set (local_s_d_r2);
  REPEAT i := 1 to HIINDEX (local_s_d_r);
    IF('PRODUCT_PROPERTY_REPRESENTATION_SCHEMA.SHAPE_REPRESENTATION'
        IN TYPEOF (local_s_d_r[i].used_representation)) THEN
        local_s_r := local_s_r + local_s_d_r[i].used_representation;
    END_IF;
  END_REPEAT;
  RETURN (local_s_r);
END_FUNCTION;
(*
```


Argument definitions:

item: An instance of **product_definition** whose **shape_representation** instances are to be sought.

4.5.4 Find shape representation of shape aspect

The function **find_shape_representation_of_shape_aspect** finds a set of **shape_representation** instances representing the shape of a specified **shape_aspect** instance.

EXPRESS specification:

```
*)
FUNCTION find_shape_representation_of_shape_aspect
    (item:shape_aspect) : SET OF shape_representation;
LOCAL
  local_p_d: SET OF property_definition:= [];
  local_s_d_r: SET OF shape_definition_representation := [];
  local_s_d_r2: BAG OF shape_definition_representation := [];
  local_s_r: SET OF shape_representation := [];
  i : INTEGER;
END_LOCAL;
-- find property_definition representing the shape_aspect
  local_p_d := bag_to_set (USEDIN (item,
   'PRODUCT_PROPERTY_DEFINITION_SCHEMA.PROPERTY_DEFINITION.DEFINITION'));
-- find shape_definition_representations refereing to the local_p_d
  REPEAT i := 1 to HIINDEX (local_p_d);
    local_s_d_r2 := local_s_d_r2 + (USEDIN (local_p_d[i],
                   'PRODUCT_PROPERTY_REPRESENTATION_SCHEMA.' +
                   'SHAPE_DEFINITION_REPRESENTATION.DEFINITION'));
  END_REPEAT;
    local_s_d_r := bag_to_set (local_s_d_r2);
  REPEAT i := 1 to HIINDEX (local_s_d_r);
    IF('PRODUCT_PROPERTY_REPRESENTATION_SCHEMA.SHAPE_REPRESENTATION'
        IN TYPEOF (local_s_d_r[i].used_representation)) THEN
        local_s_r := local_s_r + local_s_d_r[i].used_representation;
    END_IF;
  END_REPEAT;
  RETURN (local_s_r);
END_FUNCTION;
(*
```

Argument definitions:

item: An instance of **shape_aspect** whose **shape_representation** instances are to be sought.

4.5.5 Find representative shape representation of product definition

The function **find_representative_shape_representation_of_product_definition** finds an instance of **representative_shape_representation** representing the shape of a specified **product_definition** instance.

EXPRESS specification:

```
*)
FUNCTION find_representative_shape_representation_of_product_definition
    (item:product_definition) : shape_representation;
LOCAL
  local_s_r: SET OF shape_representation := [];
END_LOCAL;
-- find representative_shape_representation of the product_definition
  local_s_r := QUERY
    ( z <* find_shape_representation_of_product_definition (item)|
      'REPRESENTATIVE_SHAPE_REPRESENTATION' IN TYPEOF(z));
IF (SIZEOF (local_s_r) = 1) THEN RETURN (local_s_r[1]);
   ELSE local_s_r := []; RETURN(local_s_r[1]);
END_IF;
END_FUNCTION;
(*
```

Argument definitions:

item: An instance of **product_definition** whose **representative_shape_representation** instance is to be sought.

4.5.6 Find representative shape representation of shape aspect

The function **find_representative_shape_representation_of_shape_aspect** finds an instance of **representative_shape_representation** representing the shape of a specified **shape_aspect** instance.

EXPRESS specification:

```
*)
FUNCTION find_representative_shape_representation_of_shape_aspect
```



```
    (item:shape_aspect) : shape_representation;
LOCAL
  local_s_r: SET OF shape_representation := [];
END_LOCAL;
-- find representative_shape_representation of the shape_aspect
  local_s_r := QUERY ( z <* find_shape_representation_of_shape_aspect (item)|
                'REPRESENTATIVE_SHAPE_REPRESENTATION' IN TYPEOF(z));
IF (SIZEOF (local_s_r) = 1) THEN RETURN (local_s_r[1]);
   ELSE local_s_r := []; RETURN(local_s_r[1]);
END_IF;
END_FUNCTION;
(*
```

Argument definitions:

item: An instance of **shape_aspect** whose **representative_shape_representation** instance is to be sought.

4.5.7 Unique in product definition

The function **unique_in_product_definition** checks if a specified instance of **representative_shape_-representation** is unique in the corresponding **product_definition** instance. This function returns TRUE only if the instance of **product_definition** corresponding to the specified **representative_-shape_representation** instance does not have references to other **representative_shape_-representation** instances.

EXPRESS specification:

```
*)
FUNCTION unique_in_product_definition
    (item:representative_shape_representation) : BOOLEAN;
LOCAL
  local_p_d: SET OF product_definition := [];
  local_s_r: SET OF shape_representation := [];
  i : INTEGER;
  j : INTEGER;
END_LOCAL;
-- find product_definitions represented by the input
-- representative_shape_representation
  local_p_d := using_product_definition_of_shape_representation (item);
-- ERROR
  IF (SIZEOF (local_p_d) <> 1) THEN RETURN (FALSE);
```

```
    ELSE
-- find shape_representation representing the product_definitions
    REPEAT i := 1 to HIINDEX (local_p_d);
      local_s_r := find_shape_representation_of_product_definition
                   (local_p_d[i]);
      REPEAT j := 1 to HIINDEX (local_s_r);
        IF (('REPRESENTATIVE_SHAPE_REPRESENTATION' IN TYPEOF (local_s_r[j]))
             AND (local_s_r[j] :<>: item)) THEN RETURN (FALSE);
        END_IF;
      END_REPEAT;
    END_REPEAT;
  END_IF;
  RETURN (TRUE);
END_FUNCTION;
(*
```

Argument definitions:

item: An instance of **representative_shape_representation** whose uniqueness with regard to the corresponding **product_definition** instance is to be checked.

4.5.8 Unique in shape aspect

The function **unique_in_shape_aspect** checks if a specified **representative_shape_representation** instance is unique in the corresponding **shape_aspect** instance. This function returns TRUE only if the **shape_aspect** instance corresponding to the specified **representative_shape_representation** instance does not have references to other **representative_shape_representation** instances.

EXPRESS specification:

```
*)
FUNCTION unique_in_shape_aspect
    (item:representative_shape_representation) : BOOLEAN;
LOCAL
  local_s_a: SET OF shape_aspect := [];
  local_s_r: SET OF shape_representation := [];
  i : INTEGER;
  j : INTEGER;
END_LOCAL;
-- find shape_aspects represented by the input
-- representative_shape_representation
  local_s_a := using_shape_aspect_of_shape_representation (item);
```



```
-- ERROR
  IF (SIZEOF (local_s_a) <> 1) THEN RETURN (FALSE);
    ELSE
-- find shape_representation representing the shape_aspect
    REPEAT i := 1 to HIINDEX (local_s_a);
      local_s_r := find_shape_representation_of_shape_aspect
                     (local_s_a[i]);
      REPEAT j := 1 to HIINDEX (local_s_r);
        IF ((' REPRESENTATIVE_SHAPE_REPRESENTATION ' IN TYPEOF (local_s_r[j]))
           AND (local_s_r[j] :<>: item)) THEN RETURN (FALSE);
        END_IF;
      END_REPEAT;
    END_REPEAT;
  END_IF;
  RETURN (TRUE);
END_FUNCTION;
(*
```

Argument definitions:

item: An instance of **representative_shape_representation** whose uniqueness with regard to the corresponding **shape_aspect** instance is to be checked.

4.5.9 Using product definition of shape aspect

The function **using_product_definition_of_shape_aspect** finds an instance of **product_definition** that uses a specified **shape_aspect** instance, if such a one exists.

EXPRESS specification:

```
*)
FUNCTION using_product_definition_of_shape_aspect
    (item:shape_aspect) : product_definition ;
IF ('PRODUCT_DEFINITION_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF
    (item.of_shape.definition)) THEN
    RETURN (item.of_shape.definition);
END_IF;
END_FUNCTION;
(*
```

Argument definitions:

item: An instance of **shape_aspect** of which the corresponding **product_definition** instance is to be sought.

4.5.10 Using product definition of shape representation

The function **using_product_definition_of_shape_representation** finds all the **product_definition** instances that use a specified **shape_representation** instance.

EXPRESS specification:

```
*)
FUNCTION using_product_definition_of_shape_representation
    (item: shape_representation) : SET OF product_definition;
LOCAL
  local_s_d_r: SET OF shape_definition_representation := [];
  local_p_d_s: SET OF product_definition_shape := [];
  local_p_d: SET OF product_definition := [];
  i : INTEGER;
END_LOCAL;
-- find shape_definition_representations
  local_s_d_r  := local_s_d_r + bag_to_set (USEDIN (item,
 'PRODUCT_PROPERTY_REPRESENTATION_SCHEMA.' +
 'SHAPE_DEFINITION_REPRESENTATION.USED_REPRESENTATION'));
-- find product_definition_shape & product_definitions
REPEAT i := 1 TO HIINDEX (local_s_d_r);
   IF (('PRODUCT_PROPERTY_DEFINITION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
     TYPEOF (local_s_d_r[i].definition) AND
     ('PRODUCT_DEFINITION_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF
       (local_s_d_r[i].definition.definition)))
   THEN local_p_d := local_p_d + local_s_d_r[i].definition.definition;
   ELSE IF (('PRODUCT_PROPERTY_DEFINITION_SCHEMA.PROPERTY_DEFINITION' IN
     TYPEOF (local_s_d_r[i].definition)) AND
     ('PRODUCT_DEFINITION_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF
       (local_s_d_r[i].definition.definition)))
   THEN local_p_d := local_p_d + local_s_d_r[i].definition.definition;
   END_IF;
   END_IF;
END_REPEAT;
RETURN (local_p_d);
END_FUNCTION;
(*
```


Argument definitions:

item: An instance of **shape_representation** whose corresponding **product_definition** instances are to be sought.

4.5.11 Using shape aspect of shape representation

The function **using_shape_aspect_of_shape_representation** finds all the **shape_aspect** instances that are represented by a specified **shape_representation** instance.

EXPRESS specification:

```
*)
FUNCTION using_shape_aspect_of_shape_representation
    (item: shape_representation) : SET OF shape_aspect;
LOCAL
  local_s_d_r: SET OF shape_definition_representation := [];
  local_s_d_r2: BAG OF shape_definition_representation := [];
  local_s_a: SET OF shape_aspect := [];
  i : INTEGER;
END_LOCAL;
-- find shape_definition_representations
  local_s_d_r2  := local_s_d_r2 + (USEDIN (item,
  'PRODUCT_PROPERTY_REPRESENTATION_SCHEMA.' +
  'SHAPE_DEFINITION_REPRESENTATION.USED_REPRESENTATION'));
  local_s_d_r  := bag_to_set (local_s_d_r2);
-- find shape_aspects
REPEAT i := 1 TO HIINDEX (local_s_d_r);
   IF ('PRODUCT_PROPERTY_DEFINITION_SCHEMA.SHAPE_ASPECT' IN TYPEOF
      (local_s_d_r[i].definition))
   THEN local_s_a := local_s_a + local_s_d_r[i].definition;
   END_IF;
END_REPEAT;
RETURN (local_s_a);
END_FUNCTION;
(*
```

Argument definitions:

item: An instance of **shape_representation** whose corresponding **shape_aspect** instances are to be sought.

4.5.12 Using representations with mapping

The function **using_representations_with_mapping** returns all the **representation** instances where a specified **representation_item** instance is used through mapping.

EXPRESS specification:

```
*)
FUNCTION using_representation_with_mapping
  (item : representation_item) : SET OF representation;
LOCAL
  results: SET OF representation := [];
  local_results : SET OF representation := [];
  local_representation_map : SET OF representation_map := [];
  intermediate_items : SET OF representation_item := [];
  i : INTEGER;
  j : INTEGER;
END_LOCAL;
-- find the representations in which the item is used
-- and which is referenced from the representation_map
   local_results := QUERY (z <* using_representations (item)|
           SIZEOF (USEDIN (z, 'REPRESENTATION_SCHEMA.REPRESENTATION_MAP.' +
                             'MAPPED_REPRESENTATION')) > 0 );
  IF (SIZEOF (local_results) = 0) THEN RETURN ([]);
  ELSE
-- find the set of representation_map in which the local_results are used
  REPEAT i := 1 TO HIINDEX(local_results);
  local_representation_map := local_representation_map +
      bag_to_set (USEDIN (local_results[i],

'REPRESENTATION_SCHEMA.REPRESENTATION_MAP.MAPPED_REPRESENTATION'));
  END_REPEAT;
-- find the set of mapped_item
  REPEAT i := 1 TO HIINDEX(local_representation_map);
  intermediate_items := intermediate_items +
          bag_to_set (USEDIN (local_representation_map[i],
                 'REPRESENTATION_SCHEMA.MAPPED_ITEM.MAPPING_SOURCE'));
  END_REPEAT;
-- repeat same manner
  REPEAT j := 1 to HIINDEX(intermediate_items);
        results := results + bag_to_set (using_representation_with_mapping
                              (intermediate_items[j]));
```



```
    END_REPEAT;
  END_IF;
  RETURN (results);
END_FUNCTION;
(*
```

Argument definitions:

item: An instance of **representation_item** for which the corresponding **representation** instances that are used through mapping, are to be sought.

EXPRESS specification:

```
*)
END_SCHEMA;  -- assembly_feature_relationship_schema
(*
```

5 Assembly constraint

5.1 Introduction

The following EXPRESS declaration begins the **assembly_constraint_schema** and identifies the necessary external references.

EXPRESS specification:

```
*)
SCHEMA assembly_constraint_schema;

  REFERENCE FROM assembly_feature_relationship_schema; -- ISO 10303-109
  REFERENCE FROM explicit_constraint_schema; -- ISO 10303-108
  REFERENCE FROM explicit_geometric_constraint_schema; -- ISO 10303-108
  REFERENCE FROM geometric_model_schema; -- ISO 10303-42
  REFERENCE FROM geometry_schema; -- ISO 10303-42
  REFERENCE FROM parameterization_schema; -- ISO 10303-108
  REFERENCE FROM product_definition_schema; -- 10303-41
  REFERENCE FROM product_structure_schema; -- ISO 10303-44
  REFERENCE FROM representation_schema; -- ISO 10303-43
  REFERENCE FROM support_resource_schema; -- 10303-41

(*
```

NOTE 1 The schemas referenced above, unless otherwise stated, can be found in the following parts of ISO 10303:

assembly_feature_relationship_schema	ISO 10303-109
explicit_constraint_schema	ISO 10303-108
explicit_geometric_constraint_schema	ISO 10303-108
geometric_model_schema	ISO 10303-42
geometry_schema	ISO 10303-42
parameterization_schema	ISO 10303-108
product_definition_schema	ISO 10303-41
product_structure_schema	ISO 10303-44
representation_schema	ISO 10303-43
support_resource_schema	ISO 10303-41

NOTE 2 See annex D, Figures D.2, for a graphical presentation of this schema.

5.2 Fundamental concepts and assumptions

This schema provides the resources for the specification of explicit geometric constraints between constituents of an assembly model. Although all the constraints defined in this schema are subtypes of constraints defined in ISO 10303-108, there is a significant limitation in their semantics. They are intended to control position and orientation between exactly two constituents in an assembly, and for this reason their interpretation is as follows:

— An **assembly_constraint** asserts relationships involving rigid constituents, which control transformation matrices that position and orient models of those constituents.

NOTE A constrained model using ISO 10303-108 may, under other circumstances, be concerned with relationships between elements of a component part model.

— All constraints in this schema with the exception of the **fixed_constituent_assembly_constraint** are binary constraints controlling the relationship between precisely two constituents in an assembly. They each therefore assert relationships between two geometric elements, one of which shall be specified by **reference_elements** and the other shall be specified by **constrained_elements** which are attributes inherited from **explicit_constraint_schema** defined in clause 5 of ISO 10303-108. In the case of **fixed_constituent_assembly_constraint**, the target constituent is fixed with regard to the context of the highest constituent, namely the assembly root.

— In the case of binary constraints, related geometric elements shall belong to different constituents of an assembly model. Since a **geometric_representation_item** may belong to two or more **shape_representation** instances, all constraints are specified by the relation between two pairs of **shape_representation** and **geometric_representation_item**.

— All constraints in this schema are three-dimensional.

The geometric elements that may appear in constraint instances in this schema are all subtypes of **geometric_representation_item.** The particular sets of valid subtypes of **geometric_representation-_item** differ from one constraint type to another.

The provision of the unary **fixed_constituent_assembly_constraint** (see clause 5.3.3) allows the position and orientation of one constituent to be fixed, thus anchoring its owning assembly in space.

The fundamental concepts and assumptions stated in clause 5 of the **explicit_constraint_schema** and in clause 7 of the **explicit_geometric_constraint_schema** of ISO 10303-108 also apply in this schema.

All the constraints defined in this schema are of type **defined_constraint** as specified in clause 5 of ISO 10303-108. Their specification is descriptive rather than mathematical. As in clause 7 of ISO 10303-108, dimensional constraints are defined as subtypes of logical constraints, the rationale being that the logical relationship between geometric elements needs to be established before the semantics of certain types of dimension become unambiguous.

The three-dimensional constraints defined in this schema are appropriate for the later stages of design. In early design, where the outlines of component parts may be depicted in a two-dimensional model, the use of the sketch entity data type (as defined in clause 8 of ISO 10303-108) will be appropriate. If the sketch is constrained, two dimensional instances of constraints from clause 7 of ISO 10303-108 shall be used.

The current edition of this part of ISO 10303 treats nominal geometry and does not include tolerance information. Tolerancing decisions can profoundly impact the quality and cost of electro-mechanical assemblies. Existing approaches to tolerance analysis and synthesis in design entail detailed knowledge of geometry of the assemblies and are mostly applicable during advanced stages of design, leading to a less than optimal design process. During the design process of assemblies, both the assembly structure and associated tolerance information evolve continuously. Therefore, significant gains can be achieved by effectively using this information to influence the design of assemblies and advance tolerancing decisions to the earliest possible stages of design. This issue raises the need for effective representation of tolerancing information during different stages of design and for effective assembly modelling. The assembly model should be able to represent the function, form, and behaviour of the assembly and define both a system level conceptual model and associated hierarchical relationships. The model should describe assembly as a concept and as a data structure and could provide a way for tolerance representation and propagation, kinematics representation, and engineering analysis at the system level.

Tolerance information may be added in the future edition of this part of ISO 10303 or in AP using this part as follows:
Target entities to be enhanced (with the addition of tolerance information) are those classified under **binary_assembly_constraint**. It is expected that **parallel_assembly_constraint** can be further specialized by creating a new subtype **toleranced_parallel_assembly_constraint** and by adding a new attribute for defining tolerance zone. It is expected that **perpendicular_assembly_constraint** and **coaxial_assembly_constraint** could also be specialized in this way. Similarly, **surface_distance_-assembly_constraint_with_dimension** and **angle_assembly_constraint** may be specialized by creating new subtype entities and additionally defining attributes for specifying dimensional tolerance respectively. Specialization of **parallel_assembly_constraint_with_dimension** will require creation of a new subtype entity where attributes for specifying both tolerance zone and dimensional tolerance are defined. As for **coincidence_assembly_constraint** and **tangent_assembly_constraint**, further consideration is required since simple specialization as described above does not seem to apply.

The modification described above may require harmonization with deliverables of ISO TC213 such as the GSP(Geometrical Product Specification) standard.

5.3 Assembly constraint entity definitions

5.3.1 Assembly geometric constraint

An **assembly_geometric_constraint** is the supertype of all geometric assembly constraints. It is a subtype of **explicit_constraint** as defined in clause 5 of ISO 10303-108.

EXPRESS specification:

```
*)
ENTITY assembly_geometric_constraint
  ABSTRACT SUPERTYPE OF (ONEOF
           (binary_assembly_constraint,
           fixed_constituent_assembly_constraint))
  SUBTYPE OF (explicit_constraint);

END_ENTITY;-- assembly_geometric_constraint
(*
```

5.3.2 Binary assembly constraint

A **binary_assembly_constraint** is the generic supertype of all binary assembly constraints controlling the relationship between precisely two constituents in an assembly. It is a common subtype of **assembly_geometric_constraint** as defined in this part of ISO 10303 and **representation_-relationship_with_transformation** as defined in ISO 10303-43. It specifies a binary relationship between just two geometric elements. Only directed cases are allowed. Any instance of an **binary_assembly_constraint** is specified between exactly two pairs of **shape_representation** and **geometric_representation_item** to guarantee that it is a constraint between two different constituents.

EXPRESS specification:

```
*)
ENTITY binary_assembly_constraint
  ABSTRACT SUPERTYPE OF (ONEOF
           (parallel_assembly_constraint,
            surface_distance_assembly_constraint_with_dimension,
            angle_assembly_constraint_with_dimension,
            perpendicular_assembly_constraint,
            incidence_assembly_constraint,
            coaxial_assembly_constraint,
            tangent_assembly_constraint))
```



```
  SUBTYPE OF (assembly_geometric_constraint,
              representation_relationship_with_transformation);
SELF\representation_relationship.rep_1:
  representative_shape_representation;
SELF\representation_relationship.rep_2:
  representative_shape_representation;
WHERE
WR1: (SELF\explicit_constraint.constrained_elements[1]
      \geometric_representation_item.dim = 3) AND
     (SELF\explicit_constraint.reference_elements[1]
      \geometric_representation_item.dim = 3);
WR2: (SELF\representation_relationship.rep_1 IN (using_representations
     (SELF\representation_relationship_with_transformation.
      transformation_operator\item_defined_transformation.transform_item_1)
      + using_representation_with_mapping
     (SELF\representation_relationship_with_transformation.
    transformation_operator\item_defined_transformation.transform_item_1)))
     AND(SELF\representation_relationship.rep_2 IN (using_representations
     (SELF\representation_relationship_with_transformation.
    transformation_operator\item_defined_transformation.transform_item_2) +
      using_representation_with_mapping
     (SELF\representation_relationship_with_transformation.
   transformation_operator\item_defined_transformation.transform_item_2)));
WR3: (SIZEOF(SELF\explicit_constraint.constrained_elements)=1) AND
     (SIZEOF(SELF\explicit_constraint.reference_elements)=1);
WR4: (SELF\representation_relationship_with_transformation.
     transformation_operator\item_defined_transformation.transform_item_1 =
      SELF\explicit_constraint.reference_elements[1]) AND
     (SELF\representation_relationship_with_transformation.
     transformation_operator\item_defined_transformation.transform_item_2 =
      SELF\explicit_constraint.constrained_elements[1]);
END_ENTITY; -- binary_assembly_constraint
(*
```


Attribute definitions:

SELF\representation_relationship.rep_1: An instance of **representative_shape_representation** including the instance specified by **SELF\explicit_constraint.constrained_elements**.

SELF\representation_relationship.rep_2: An instance of **representative_shape_representation** including the instance specified by **SELF\explicit_constraint.reference_elements**.

Formal propositions:

WR1: The dimensionality of instances specified by **constrained_elements** and **reference_elements** shall be 3.

WR2: This entity has the attribute **transformation_operator** that specifies **item_defined_-transformation** that has two attributes **transform_item_1** and **transform_item_2**. The **transform_-item_1** shall be used, directly or indirectly, by **rep_1** and **transform_item_2** shall be used, directly or indirectly, by **rep_2** .

WR3: The number of elements in **explicit_constraint.constrained_elements** shall be one. The number of elements in **explicit_constraint.reference_elements** shall also be one.

WR4: **SELF\representation_relationship_with_transformation.transformation_operator\item_-defined_transformation.transform_item_1** shall be equal to **SELF\explicit_constraint.reference_-elements**, and **SELF\representation_relationship_with_transformation.transformation_-operator\item_defined_transformation.transform_item_2** shall be equal to **SELF\explicit_-constraint.constrained_elements**.

5.3.3 Fixed constituent assembly constraint

A **fixed_constituent_assembly_constraint** is a common subtype of **assembly_geometric_constraint** as defined in this part of ISO 10303 and **fixed_element_geometric_constraint** as defined in clause 7 of ISO 10303. An instance of **fixed_constituent_assembly_constraint** constrains specified constituent instance fixed in space.

NOTE This constraint asserts that one constituent shape model instance has a fixed position and orientation with regard to the context of the highest level constituent, namely assembly root, to anchor the assembly in space. Without its use, assembly constraint equations may have infinitely many solutions due to the existence of translational and rotational degrees of freedom of the entire assembly. The implication of the constraint is that at the time of data transfer, the sending system constrained the specified constituent as fixed for solving assembly constraint equations. The receiving system, after knowing this fact, may substitute the constituent to be fixed for checking various design alternatives.

An instance of **representative_shape_representation** representing the **product_definition** of the constituent to be fixed shall be specified. A **geometric_constraint_element** instance inherited from **fixed_element_geometric_constraint**, which corresponds to the geometric entity picked from a screen by the user for identifying fixed constituent, shall be consistent with the **representative_shape_-representation** specified. Namely, the **geometric_constraint_element** instance shall be included in the **representative_shape_representation** instance. The **geometric_constraint_element** shall be either a point or a curve or a surface. For details of its definition, see clause 7 of ISO 10303-108.

EXPRESS specification:

```
*)
ENTITY fixed_constituent_assembly_constraint
SUBTYPE OF (assembly_geometric_constraint,
            fixed_element_geometric_constraint);
        fixed_constituent: representative_shape_representation;
WHERE
WR1: SELF\explicit_constraint.constrained_elements[1]
     \geometric_representation_item.dim = 3;
WR2: (assembly_leaf (using_product_definition_of_shape_representation
     (fixed_constituent)[1]));
WR3: (SIZEOF(SELF\explicit_constraint.constrained_elements) = 1) AND
     (SIZEOF(SELF\explicit_constraint.reference_elements) = 0);
WR4: fixed_constituent IN
   (using_representations(SELF\explicit_constraint.constrained_elements[1]
     \representation_item) + using_representation_with_mapping
   (SELF\explicit_constraint.constrained_elements[1]\representation_item));
END_ENTITY;-- fixed_constituent_assembly_constraint
(*
```

Formal propositions:

WR1: The dimensionality of an instance specified by **constrained_elements** shall be 3.

WR2: The **representative_shape_representation** specified by the **fixed_constituent** attribute shall represent a **product_definition** that is a leaf node in the assembly structure.

WR3: The number of elements in **explicit_constraint.constrained_elements** shall be one and the number of elements in **explicit_constraint.reference_elements** shall be zero.

WR4: The instance of **representation_item** specified by the inherited attribute **constrained_-elements**[1] shall be included, directly or indirectly with mapping, in the **fixed_constituent**.

5.3.4 Parallel assembly constraint

A **parallel_assembly_constraint** specifies that two **linear_geometry_constraint_element** instances, each belonging to different constituents, are constrained to be parallel. The **linear_geometry_-constraint_element** shall be either a line or a plane. For details of its definition, see clause 7 of ISO 10303-108.

EXPRESS specification:

```
*)
ENTITY parallel_assembly_constraint
  SUBTYPE OF (binary_assembly_constraint,
              parallel_geometric_constraint);
END_ENTITY;-- parallel_assembly_constraint
(*
```

5.3.5 Parallel assembly constraint with dimension

A **parallel_assembly_constraint_with_dimension** is a subtype of the **parallel_assembly_constraint** that is used for constraining distances between two parallel **linear_geometry_constraint_element** instances. The **linear_geometry_constraint_element** shall be either a line or a plane. For details of its definition, see clause 7 of ISO 10303-108.

The actual distance value is an attribute of the supertype **pgc_with_dimension** as defined in clause 7 of ISO 10303-108. Distances between lines and planes are measured in the directions of their common normals.

EXPRESS specification:

```
*)
ENTITY  parallel_assembly_constraint_with_dimension
  SUBTYPE OF (parallel_assembly_constraint,
              pgc_with_dimension);
END_ENTITY;-- parallel_assembly_constraint_with_dimension
(*
```

5.3.6 Surface distance assembly constraint with dimension

A **surface_distance_assembly_constraint_with_dimension** constrains the minimum distance between two surfaces belonging to different constituents. The actual distance value is an attribute of the supertype **sdgc_with_dimension** as defined in clause 7 of ISO 10303-108.

The optional **near_points** attribute of the supertype **sdgc_with_dimension** as defined in clause 7 of ISO 10303-108 may be used to provide approximate locations of the closest points on each surface in cases where multiple solutions of this constraint may arise.

This constraint does not exclude the case when two surfaces specified are both planes. However, it is appropriate to use **parallel_assembly_constraint_with_dimension** when two planes are parallel, otherwise the constraint is meaningless.

EXPRESS specification:

```
*)
ENTITY surface_distance_assembly_constraint_with_dimension
  SUBTYPE OF (binary_assembly_constraint,
              sdgc_with_dimension);
END_ENTITY;-- surface_distance_assembly_constraint_with_dimension
(*
```

5.3.7 Angle assembly constraint with dimension

An **angle_assembly_constraint_with_dimension** specifies that an instance specified by **constrained-_elements** is constrained to make a specified angle with a given instance specified by **reference_-elements**. The two elements concerned shall belong to different constituents. The actual angle value is an attribute of the supertype **angle_geometric_constraint_with_dimension** as defined in clause 7 of ISO 10303-108.

If both elements involved in the constraint are lines, then it is not necessary for them to be co-planar. If both elements are not in a plane then the angle is measured in a plane that is spanned by the normals of both elements.

EXPRESS specification:

```
*)
ENTITY angle_assembly_constraint_with_dimension
  SUBTYPE OF (binary_assembly_constraint,
              agc_with_dimension);
END_ENTITY;-- angle_assembly_constraint_with_dimension
(*
```


5.3.8 Perpendicular assembly constraint

A **perpendicular_assembly_constraint** specifies that two **linear_geometry_constraint_element** instances belonging to different constituents be constrained to be mutually perpendicular. The **linear_-geometry_constraint_element** shall be either a line or a plane. For details of its definition, see clause 7 of ISO 10303-108.

EXPRESS specification:

```
*)
ENTITY perpendicular_assembly_constraint
```



```
  SUBTYPE OF (binary_assembly_constraint,
              perpendicular_geometric_constraint);
END_ENTITY;-- perpendicular_assembly_constraint
(*
```

5.3.9 Incidence assembly constraint

An **incidence_assembly_constraint** specifies that two instances of **geometric_constraint_element** belonging to different constituents have the relationship that one of them, regarded as a point set, is entirely included in the other. There are two cases: that the instance specified by **constrained_elements** is entirely included in the instance specified by **reference_elements** or inversely, that the instance specified by **reference_elements** is entirely included in the instance specified by **constrained_-elements**. Complete coincidence of the two instances is a special case of inclusion. The **geometric_-constraint_element** shall be either a point or a curve or a surface. For details of its definition, see clause 7 of ISO 10303-108.

EXAMPLE If two unbounded lines are constrained to be incident, they must have the same or opposite directions, and each line must include the defining point of the other. In this case the two point sets completely coincide. If one of the lines is trimmed but the other is unbounded then the first point set is a subset of the second.

<u>EXPRESS specification:</u>

```
*)
ENTITY incidence_assembly_constraint
  SUBTYPE OF (binary_assembly_constraint,
              incidence_geometric_constraint);
END_ENTITY;-- incidence_assembly_constraint
(*
```


NOTE In some cases the use of a dimensional constraint with zero dimension value is more appropriate than the use of an **incidence_assembly_constraint**. The mating of planar surfaces of two parts in an assembly may be taken as an example. Control of the distance between two parts in an assembly is a typical assembly design requirement and mating can be regarded as its special case, where the distance value is zero. Therefore, engineers will prefer the use of **parallel_assembly_constraint_with_dimension** rather than **incidence_assembly_-constraint** for representing the mate relation between two parts. The use of **parallel_assembly_constraint_-with_dimension** is advantageous in this case as it allows the imposition of an additional constraint on the relative orientations of the planes concerned, that is the two normals of these planes should be opposite.

5.3.10 Coaxial assembly constraint

A **coaxial_assembly_constraint** specifies that the axes of two elements belonging to different constituents are constrained to be identical. The constraint is valid for entities of **axial_geometry_-**

constraint_element type. The **axial_geometry_constraint_element** type includes points, lines, circles, surfaces and solids having an axis of rotational symmetry. For details of its definition, see clause 7 of ISO 10303-108.

EXPRESS specification:

```
*)
ENTITY coaxial_assembly_constraint
  SUBTYPE OF (binary_assembly_constraint,
              coaxial_geometric_constraint);
END_ENTITY;-- coaxial_assembly_constraint
(*
```

5.3.11 Tangent assembly constraint

A **tangent_assembly_constraint** specifies that two geometric elements belonging to different constituents are constrained to be tangent to each other.

EXPRESS specification:

```
*)
ENTITY tangent_assembly_constraint
  SUBTYPE OF (binary_assembly_constraint,
              tangent_geometric_constraint);
END_ENTITY;-- tangent_assembly_constraint
(*
```

5.4 Assembly constraint function definitions

5.4.1 Assembly leaf

This function determines whether or not an instance of **product_definition** is a leaf node in the tree structure of an instance of **assembly_component_usage** as defined in ISO 10303-44.

EXPRESS specification:

```
*)
FUNCTION assembly_leaf
  (item: product_definition) : BOOLEAN;
LOCAL
  local_relation: SET OF assembly_component_usage := [];
  local_relation2: BAG OF assembly_component_usage := [];
END_LOCAL;
```

