INTERNATIONAL STANDARD

ISO 10303-43

First edition
1994-12-15

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

Part 43:
Integrated generic resources: Representation structures

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

Partie 43: Ressources génériques intégrées: Structures de représentation

Reference number
ISO 10303-43:1994(E)

Contents

Page

1 Scope 1

2 Normative references 2

3 Definitions 3
3.1 Terms defined in ISO 10303-1 3
4.1 Introduction 4
4.2 Fundamental concepts and assumptions 4
4.2.1 Fundamental concepts and assumptions related to representation 4
4.2.2 Fundamental concepts and assumptions related to the context of representation 5
4.2.3 Fundamental concepts and assumptions related to elements of representation 5
4.2.4 Fundamental concepts and assumptions related to the association of representations . . 6
4.2.5 Fundamental concepts and assumptions related to transformation 6
4.3 Representation_schema type definition: transformation 8
4.4 Representation_schema entity definitions 8
4.4.1 uncertainty_measure_with_unit 8
4.4.2 representation_context 9
4.4.3 global_uncertainty_assigned_context 9
4.4.4 representation_item 10
4.4.5 representation 10
4.4.6 representation_relationship 12
4.4.7 item_defined_transformation 13
4.4.8 functionally_defined_transformation 14
4.4.9 representation_relationship_with_transformation 14
4.4.10 representation_map 15
4.4.11 mapped_item 16
4.4.12 definitional_representation 17
4.4.13 parametric_representation_context 17
4.5 Representation_schema function definitions 18
4.5.1 acyclic_mapped_representation 18
4.5.2 item_in_context 19
4.5.3 using_representations 21

International Organization for Standardization
Case Postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

Annexes

A Short names of entities 23

B Information object registration 24
B.1 Document identification 24
B.2 Schema identification 24

C Computer-interpretable listings 25

D EXPRESS-G figures 26

Index 27

Figures

Figure D.1 - Representation schema subject planning model 26

Tables

Table A.1 - Short names of entities 23

Foreword

The International Organization for Standardization (ISO) 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.

Draft International Standards adopted by 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.

International Standard ISO 10303-43 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC4, *Industrial data and global manufacturing languages*.

ISO 10303 consists of the following parts under the general title *Industrial automation systems and integration - Product data representation and exchange*:

— Part 1, Overview and fundamental principles;

— Part 11, Description methods: The EXPRESS language reference manual;

— Part 21, Implementation methods: Clear text encoding of the exchange structure;

— Part 22, Implementation methods: Standard data access interface;

— Part 31, Conformance testing methodology and framework: General concepts;

— Part 32, Conformance testing methodology and framework: requirements on testing laboratories and clients;

— Part 41, Integrated generic resources: Fundamentals of product description and support;

— Part 42, Integrated generic resources: Geometric and topological representation;

— Part 43, Integrated generic resources: Representation structures;

— Part 44, Integrated generic resources: Product structure configuration;

— Part 45, Integrated generic resources: Materials;

— Part 46, Integrated generic resources: Visual presentation;

— Part 47, Integrated generic resources: Shape variation tolerances;

— Part 49, Integrated generic resources: Process structure and properties;

- Part 101, Integrated application resources: Draughting;
- Part 104, Integrated application resources: Finite element analysis;
- Part 105, Integrated application resources: Kinematics;
- Part 201, Application protocol: Explicit draughting;
- Part 202, Application protocol: Associative draughting;
- Part 203, Application protocol: Configuration controlled design;
- Part 207, Application protocol: Sheet metal die planning and design;
- Part 210, Application protocol: Printed circuit assembly product design data;
- Part 213, Application protocol: Numerical control process plans for machined parts.

The structure of this International Standard is described in ISO 10303-1. The numbering of the parts of this International Standard reflects its structure:

- Part 11 specifies the description method;
- Parts 21 and 22 specify the implementation methods;
- Parts 31 and 32 specify the conformance testing methodology and framework;
- Parts 41 to 49 specify the integrated generic resources;
- Parts 101 to 105 specify the integrated application resources;
- Parts 201 to 213 specify the application protocols.

Should further parts be published, they will follow the same numbering pattern.

Annexes A and B form an integral part of this part of ISO 10303. Annexes C and D are for information only.

Diskette

Users should note that this part of ISO 10303 comprises a diskette:
— the short names of entities given in annex A are also included on the diskette;
— the EXPRESS listings (annex C) are provided on the diskette only;
— a method to enable users to report errors in the documentation is given. Full details are provided in the file.

Introduction

ISO 10303 is an International Standard for the computer-interpretable representation and exchange of product data. The objective is to provide a neutral mechanism capable of describing product data throughout the life cycle of a product, independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving.

This International Standard is organized as a series of parts, each published separately. The parts of ISO 10303 fall into one of the following series: description methods, integrated resources, application protocols, abstract test suites, implementation methods, and conformance testing. The series are described in ISO 10303-1. This part of ISO 10303 is a member of the integrated resources series.

This part of ISO 10303 specifies the overall structure for representation. Collections of elements may be formed to act as the representation of some aspect of product data, such as a property of a product. Each element in such a collection is a representation item. An example of a property that can be represented is the shape of a product. The aspect or property that is being represented is not specified in this part of ISO 10303. Instead the subject of the representation is defined where the capabilities for representation are used in other parts of ISO 10303.

When representation items are collected to participate in a representation, they share a common context which is associated with the representation. This context is referred to as a representation context.

Not all elements of product data participate in representations. Those items which may participate in representations are defined to be representation items. Representation items are those elements that have complete meaning only when associated with a context. As an example, a point is a representation item which is only meaningful within a context (a coordinate space). In contrast, the name of a person is not a representation item because it has meaning separate from any context.

In addition to being an element of representation, a representation item may also support the definition of other representation items. This part of ISO 10303 allows for this distinction.

A collection of product data may contain numerous representation items, each participating in one or more representations. These representations may be related to form a structure which also relates the representation contexts. This structure can then be used to determine which representation items may be related to each other in a meaningful way. As an example, distance between points is only meaningful if the coordinate systems in which the points are defined can be related.

Representations that are unrelated in one context may be related in another. Consider the representation of the shape of a part and its components. The shape of each component may be represented as an independent concept, unrelated to the shape of the other components. In the context of the assembled part, however, the shapes of the components are related.

An aspect of product data may have zero, one, or multiple representations, none of which are the concept itself. For example, the shape of a part may be represented by a collection both of two-dimensional geometry and of constructive solid geometry. Either representation is an idealization of the shape.

Each representation is not necessarily a complete model of some aspect of product data, but it may represent a model of the aspect that is suitable for specific applications. Neither shape representation in the previous paragraph is necessarily a complete representation of the shape concept. Another shape representation might include tolerance information. Rather, each representation is suitable to some specific application's view or approach.

This part of ISO 10303 specifies the structure for transformation of one representation into another.

This part of ISO 10303 includes constructs for the specification of transformations. These transformations may be asserted as defining the relationship between existing representations. The ability to define a new representation by applying the transformation to another representation is provided. The new representation in this case is called a mapped item.

This page intentionally left blank

Industrial automation systems and integration — Product data representation and exchange — Part 43: Integrated generic resources: Representation structures

1 Scope

This part of ISO 10303 specifies the integrated generic resources for associating elements of representation into distinct collections. It provides a basis for distinguishing, within a set of such elements, which elements are related and which elements are not. This part also includes structure for specifying the relationships among these collections, including transformation of one representation as it participates in such a relationship.

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

- the specification of distinct unrelated contexts for representation;
- the specification of elements of representation;
- the association of elements of representation with one or more contexts in which they are combined to represent a concept;
- the association of elements of representation such that one defines another;
- a structure for relating two representations such that one participates in the definition of the other;
- a structure for relating two representations in which one does not participate in the definition of the other;
- constraints to prevent the recursive definition of instances of an element of representation;
- the specification of the process to transform one element of representation into another by specifying an element of the process before the application of the process and that element of the process after application of the process;
- the specification of the process to transform one element of representation into another by specifying the transforming function.

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

- the complete specification of the types of representation, types of elements of representation, and types of representation context;
- the specification of the uses of representation;
- the association of representation with any of its possible uses;
- constraints requiring a directed relationship between representations;

  NOTE - A directed relationship exists between items A and B when the meaning of the relationship of A to B is different from the meaning of B to A. A and B are peers in a non-directed relationship. A directed relationship is a constrained version of the relationships specified in this part of ISO 10303. Such constraints are left to those integrated resources or application protocols specializing the structures presented in this part of ISO 10303.

- constraints forbidding cyclic structures of related representations;
- constraints requiring a directed relationship between the contexts in which related representations exist;
- constraints forbidding cyclic structures of relationships between representation contexts.

2 Normative references

The following standards contain provisions which, through references in this text, constitute provisions of this part of ISO 10303. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO 10303 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 8824-1:–[1], *Information technology - Open Systems Interconnnection - Abstract Syntax Notation One (ASN.1) - Part 1: Specification of Basic Notation.*

ISO 10303-1:1994, *Industrial automation systems and integration - Product data representation and exchange - Part 1: Overview and fundamental principles.*

ISO 10303-11:1994, *Industrial automation systems and integration - Product data representation and exchange - Part 11: Description methods: The EXPRESS language reference manual.*

ISO 10303-41:1994, *Industrial automation systems and integration - Product data representation and exchange - Part 41: Integrated generic resources: Fundamentals of product description and support.*

[1] To be published.

3 Definitions

3.1 Terms defined in ISO 10303-1

This part of ISO 10303 makes use of the following terms defined in ISO 10303-1.

- application;
- application protocol;
- assembly;
- component;
- data;
- information;
- integrated resource;
- product;
- product data;
- structure.

4 Representation_schema

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

EXPRESS Specification:

```
*)
SCHEMA representation_schema;

REFERENCE FROM support_resource_schema
    (identifier,
     label,
     text,
     bag_to_set);

REFERENCE FROM measure_schema
    (measure_with_unit);
(*
```

NOTES

1 - See annex D for a graphical presentation of this schema using the EXPRESS-G notation.

2 - The schema referenced above can be found in the following part of ISO 10303:

support_resource_schema	ISO 10303-41
measures_schema	ISO 10303-41

4.1 Introduction

The subject of the **representation_schema** is the structures for the representation that relates a collection of elements of product data to a context. The use of the representation(s) is not defined in this part of ISO 10303. Other parts of ISO 10303 make use of the structures defined in this part.

Structures are specified that are common to all uses of representation in ISO 10303. Each representation has a context and a collection of elements specified in that context. A context may be related to different or broader contexts.

NOTE - For the representation of geometric elements, the context is a coordinate space.

This part of ISO 10303 also enables the transformation between representations to be specified. The ability to specify the transformation of one representation such that the result forms part of another representation is provided.

4.2 Fundamental concepts and assumptions

4.2.1 Fundamental concepts and assumptions related to representation

The following assumptions apply to the portions of this schema that deal with representation:

- Elements of representation are organized into identifiable collections associated in a context for specific uses. These organizations are called representations. By organizing the elements in this way, the elements are distinguishable as either related or unrelated in the context of the specified use.

NOTES

1 - In this and following notes, most of the examples will be geometric in nature. However, the concepts being illustrated are not restricted to geometry.

2 - Consider two points with coordinate values of (0,0,0) and (1,0,0). It is not possible to assert that these points are one unit apart in space until it is established that they are in the same coordinate space. The specification of a point by itself does not contain enough data to state which coordinate space it is in and what other elements also share that coordinate space.

- In this International Standard, these representations correspond to instances of the entity **representation**.

- A representation may be invoked in more than one use. A representation is independent of its use.

NOTE 3 - Consider a collection of geometry in a coordinate space. Such a collection may be used as the shape of a product. The collection may also be used, possibly with some transformation, as a drawing or picture of the product. Neither of these uses is part of the definition of the collection itself.

4.2.2 Fundamental concepts and assumptions related to the context of representation

The following assumptions apply to the portions of this schema that deal with the context of representation:

– A representation has a context. It is the context in which the elements of the representation are related.

– In this International Standard, the context of a representation corresponds to an instance of the entity **representation_context**.

– Because the context of a representation exists only as a basis for forming the representations that use it, representation contexts are related only if representations using the contexts are related.

– Representation contexts, and the representations that use them, may be specialized further in other parts of ISO 10303. Such specializations may also constrain the collection of representation elements.

NOTE - Possible specializations include contexts for geometry, topology, finite element modelling and kinematic modelling.

4.2.3 Fundamental concepts and assumptions related to elements of representation

The following assumptions apply to the portions of this schema that deal with elements of representation.

– Elements of representation may either participate directly in a representation, or support the definition of some other element of representation.

NOTE 1 - A point may be the only element in the representation of a product location, or it may serve as the end point of a line providing part of the definition of a line that is the only element in the representation of the edge of a product. In the first case the point itself is an element in a representation directly. In the second case, the point serves only to provide definition for the line entity.

– Elements of representation are not used to support the definition of other elements of representation except as they may eventually support a representation.

– Elements of representation may reference each other and form trees of such elements with identifiable roots. The association of a root element with a context shall associate all elements in the tree with the same context.

NOTE 2 - Consider a curve that references a number of points. These points are all assumed to be in the same coordinate space as the curve by virtue of their reference from the curve.

– Elements of representation are associated with contexts as a basis for distinguishing which are related and which are not related among them.

– In this International Standard an element of representation is described as a **representation_item**, and the association of one or more elements of representation with a context is a **representation**.

4.2.4 Fundamental concepts and assumptions related to the association of representations

The following assumptions apply to the portions of this schema that deal with the association of representations:

– A representation may be related to another representation.

– Representation A may be related to representation B such that they both participate in the association, but one does not define the other. In this International Standard, this type of association corresponds to the entity **representation_relationship**.

– If representation A is related to representation B with the meaning that this use of representation A is an element in representation B, then the relationship itself is an element in representation B. In this International Standard, this type of association corresponds to the entities **mapped_item** and **representation_map**.

– Two collections of representation elements may be unrelated in two separate contexts, and yet related in a third context or related only as they both participate in a relating structure.

NOTE - Consider two collections of geometric elements that each represent the shape of a part. Each of those shapes exists in a separate context independent and completely unrelated to the other. A third context may exist for the shape of an assembly in which these parts are components. In this third context, all of the elements are related, either through a direct association of those elements with that context, or an association of the part representations to the assembly representation.

4.2.5 Fundamental concepts and assumptions related to transformation

The following assumptions apply to the portions of this schema that deal with transformations:

– Representation elements participating in a representation may be transformed when and as that representation is related to another representation.

– A representation element may be transformed when and as it is used to define another representation element.

– A transformation may be defined as a function f between a domain A and range B. The function f:A→B takes each element a in A and maps it to an element b in B, f(a)=b. The complete specification of a use of a transformation requires the following:

a) the set of elements a to be transformed;

b) the set of elements b resulting from the transformations;

c) the definition of the context A which is common to the set of elements a;

d) the definition of the context B which is common to the set of elements b;

e) the function f.

– In this International Standard, the contexts A and B are **representation_contexts**. The elements a and b are **representation_items**. The relationships between a and A and between b and B are specified as a **representation**.

- Two different approaches are used in this part of ISO 10303 to specify transformations. These are characterized by:

a) defining the function f

b) by specifying an element from each context which are sufficient to determine the transformation.

A transformation specified by a defined function is a **functionally_defined_transformation**. A transformation defined by specifying elements is an **item_defined_transformation**.

– Where a transformation is defined as function, the form of the function is specified as normative text within this International Standard. The information needed to derive the function is explicitly included in the data specification. This information may be provided by one or more methods. The appropriate data specification may be selected depending on the use of the transformation.

NOTES

1 - A transformation that rotates an element of representation about the x, y, and z axes of a Cartesian coordinate space is specified by three normalized vectors defining the columns of an orthogonal matrix T. The matrix T may be used directly to compute the desired transformation.

2 - A transformation may scale all elements of representation related in context A by some factor.

– Where two elements are used to specify the transformation, these are an element a in A and an element b in B from which the required transformation function f may be derived.

NOTE 3 - If the required transformation function is to effect rigid motion in a coordinate space, then the function f could be uniquely determined by an **axis2_placement_3d** a1 and an **axis2_placement_3d** b1 such that f takes a1 and transforms it onto b1. Entity **axis2_placement_3d** is defined in ISO 10303-42.

– Another function, similar to a transformation, is one where all of the elements a in A are used as a single element b in B. This asserts the existence of a new **representation_item**, b, rather than using a and b (and the related contexts) to define a transformation. In this International Standard, this kind of assertion is a **mapped_item**. The elements a in A are the **mapped_item.mapped_representation**. The element b is the **mapped_item**. The two elements are specified to allow a transformation to be determined in the same manner as for the **item_defined_transformation**. These elements are the **mapped_item.mapping_target** and the **mapped_item.mapping_source.mapping_origin**.

NOTE 4 - **Mapped_item** may be specialized in other parts of ISO 10303 to define the role played by the **mapped_item** in the **representation** into which it is mapped. Such specializations may specify the form of transformation involved. The transformation need not be geometric and may provide positioning or orientation information. It does not necessarily imply that the representation_items from the mapped representation are included in the target representation.

4.3 Representation_schema type definition: transformation

A **transformation** is a selection between types of transformation function specifications.

EXPRESS specification:

```
*)
TYPE transformation = SELECT
  (item_defined_transformation,
   functionally_defined_transformation);
END_TYPE;
(*
```

4.4 Representation_schema entity definitions

This subclause contains the EXPRESS entity definitions in the **representation_schema**.

4.4.1 uncertainty_measure_with_unit

An **uncertainty_measure_with_unit** is a **measure_with_unit** that specifies the amount a physical quantity may vary from its specified value and the unit in which that amount is specified.

EXPRESS specification:

```
*)
ENTITY uncertainty_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  name        : label;
  description : text;
WHERE
  WR1: (NOT ('NUMBER' IN TYPEOF
       (SELF\measure_with_unit.value_component))) OR
       (SELF\measure_with_unit.value_component >= 0);
END_ENTITY;
(*
```

Attribute definitions:

name: the word or group of words by which the **uncertainty_measure_with_unit** is referred to.

description: an informal description of the reason for the specification of the **uncertainty_measure_with_unit**.

Formal propositions:

WR1: the **value_component** of the **uncertainty_measure_with_unit** shall be a positive number if the type of the **value_component** is a number.

4.4.2 representation_context

A **representation_context** is a context in which a collection of **representation_items** are related.

Two **representation_contexts** are separate and have no relationship unless a relationship is separately specified.

EXPRESS specification:

```
*)
ENTITY representation_context;
  context_identifier : identifier;
  context_type       : text;
INVERSE
  representations_in_context : SET [1:?] OF representation
    FOR context_of_items;
END_ENTITY;
(*
```

Attribute definitions:

context_identifier: an identifier of the **representation_context**.

context_type: a description of the type of a **representation_context**.

> NOTE - Constraints on the uniqueness of **context_identifier**, or the allowed values of **context_type**, are specified by application protocols.

representations_in_context: at least one **representation** shall be associated with each **representation_context**.

4.4.3 global_uncertainty_assigned_context

A **global_uncertainty_assigned_context** is a **representation_context** in which specified physical quantities are permitted to vary or be varied. This variation shall be in accordance with the specified **uncertainty_measure_with_units**.

EXPRESS specification:

```
*)
ENTITY global_uncertainty_assigned_context
  SUBTYPE OF (representation_context);
  uncertainty : SET [1:?] OF uncertainty_measure_with_unit;
END_ENTITY;
(*
```

Attribute definitions:

uncertainty: the **uncertainty_measure_with_units** that apply in the **representation_context**.

4.4.4 representation_item

A **representation_item** is an element of product data that participates in one or more **representations** or contributes to the definition of another **representation_item**.

A **representation_item** contributes to the definition of another **representation_item** when it is referenced by that **representation_item**.

EXPRESS specification:

```
*)
ENTITY representation_item;
  name : label;
WHERE
  WR1: SIZEOF(using_representations(SELF)) > 0;
END_ENTITY;
(*
```

Attribute definitions:

name: an identifier of the **representation_item**.

Formal propositions:

WR1: every **representation_item** shall be associated with at least one **representation_context**.

4.4.5 representation

A **representation** is a collection of one or more **representation_items** that are related in a specified **representation_context**. The use of a **representation**, i.e., that which is being represented, is not specified in this part of ISO 10303.

The relationship of **representation_item** to **representation_context** is the basis for distinguishing which **representation_item** entities are related.

NOTE 1 - Consider the context in which a set of **representation_items** is related to represent the shape of something. The **items** related in this context are used in this **representation**. Other **representation_items** are specifically not included. This is the basis for distinguishing which **geometric_representation_items** in the set **representation_items** are related. This distinction is not otherwise included in the specification of **representation_items**.

The members of the set of **items** plus all **representation_items** indirectly referenced by that set are related to the **context_of_items** by a **representation**. Indirect reference to a **representation_item** occurs when it is referenced through any number of intervening entities, each of type **representation_item**.

NOTE 2 - A **representation** relates a **representation_context** to trees of **representation_items** with each tree rooted on one member of the set of **items**. A **representation_item** is one node in the tree, and the reference of one **representation_item** to another is a branch.

The set of **representations** directly referenced as **items**, related to the **context_of_items**, is the **representation**. The **representation_items** indirectly referenced support the definition of the **items** and are all related in the same **representation_context**.

> NOTE 3 - In the **representation** of the shape of a cube by a set of **line** entities, the set of **lines** is the only **representation_item** representing the shape. A **line** in turn references a **cartesian_point** and a **direction** that support the **line** definition and are related with each other and the **line** in the referenced **representation_context**. The shape, however, is not represented by these **cartesian_points** and **directions**.

A **representation** is specified to meet the needs of an application. Often a **representation** is incomplete in that it need not fully model the concept that is represented, although it may be adequate for a given application.

> NOTE 4 - Consider a collection of two dimensional **representation_items** used to represent the shape of a machined part. It is not a complete description of the shape, but is suitable for certain applications such as computer-aided draughting.

One **representation_item** may be related to more than one **representation_context**. Two **representations** are not related solely because the same **representation_item** is referenced directly or indirectly from their sets of **items**.

> NOTE 5 - Consider a **surface** that is used in the **representations** of the shape of a casting die and of the shape of the part cast in that die. The same **surface** is related to two distinct **representation_contexts** (i.e., coordinate spaces): one for the die and one for the part by the **representations**. However, the **representations** are not related. The two **representations** simply share a common **representation_item**.

The same **representation_item** may be related multiple times to the same **representation_context** by being used directly or indirectly in several **representations**, each referencing the same **representation_context**. This does not have the meaning that each **representation** is creating a new instance of the same **representation_item** in the same **representation_context**. Rather, each **representation** reasserts one instance of the **representation_item** in the **representation_context** for different uses.

> NOTE 6 - Consider two **representations**, each having the same value for **context_of_items**. One is a **representation** of the shape of a cube and indirectly references a **line** as one of its edges. The second simply references the **line** as one of its **items**. There are not two occurrences of the **line** and its sub-tree of referenced **representation_items** in the **representation_context**. Rather, the use of the **line** in that **geometric_representation_context** has been asserted twice, once in each **representation**. The first might exist as representing the shape of the whole cube. The other might exist as representing the shape of an edge of the same cube.

<u>EXPRESS specification</u>:

```
*)
ENTITY representation;
  name               : label;
  items              : SET[1:?] OF representation_item;
  context_of_items   : representation_context;
END_ENTITY;
(*
```

Attribute definitions:

name: an identifier of the **representation**.

items: a set of **representation_items** that are related in the **context_of_items**.

context_of_items: a **representation_context** in which the **items** are related to form a **representation** of some concept.

4.4.6 representation_relationship

A **representation_relationship** is the association of two **representations**.

NOTES

1 - One **representation** is not made part of the definition of the other by participation in a **representation_relationship**.

2 - A combination of **representations** and **representation_relationships** may be used to model a tree of related **representations**. In such a tree, **representations** are the nodes and **representation_relationships** are the branches connecting the nodes.

3 - It may occur that certain structures of **representation_relationship** should be acyclic, i.e., a tree or a complex structure of **representations** in which no **representation** is its own ancestor. Such constraints are outside the scope of this part of ISO 10303.

4 - It may occur that certain structures of **representation_relationship** are directed. The meaning of the relationship in one direction is defined. In the other direction no meaning or a meaning other than the inverse of the first direction is intended. No such distinction is made in this part of ISO 10303.

EXPRESS specification:

```
*)
ENTITY representation_relationship;
  name        : label;
  description : text;
  rep_1       : representation;
  rep_2       : representation;
END_ENTITY;
(*
```

Attribute definitions:

name: an identifier of the representation_relationship.

description: a description of the representation_relationship.

rep_1: the first of two **representations** that are related.

rep_2: the second of two **representations** that are related.

NOTE 5 - There is no significance to the ordering of the two related **representations**. The names serve only to distinguish the attributes. If any significance to the ordering is needed in any specializations of **representation_relationship**, the significance will be defined in the specialization.

4.4.7 item_defined_transformation

An **item_defined_transformation** is a transformation performed by defining two **representation_items** where one **representation_item** is the result of applying the transformation function to the other. The transformation function is not explicitly provided, but it is derived through its relationship to the **representation_items**. The transformation function is a function f between a domain A and range B. The function f:A→B takes each element a in A and maps it to an element b in B, f(a)=b.

There is no significance to the order of the two related **representation_items**. In the general case, the inverse transformation function f′:B→A may also be derived as appropriate.

NOTE 1 - The use of the ordinal number in the attribute names serves only to distinguish the names. If any significance in the ordering is needed for usage of **item_defined_transformation**, the significance will be defined with the specialization or usage.

EXPRESS specification:

```
*)
ENTITY item_defined_transformation;
  name              : label;
  description       : text;
  transform_item_1  : representation_item;
  transform_item_2  : representation_item;
END_ENTITY;
(*
```

Attribute definitions:

name: an identifier of the **item_defined_transformation**.

description: a description of the **item_defined_transformation**.

transform_item_1: the first of two **representation_items** that describe the transformation function.

transform_item_2: the second of two **representation_items** that describe the transformation function.

NOTE 2 - Consider one **representation** having a set of **representation_items** and a context which is a Cartesian coordinate space and a second **representation** with another set of **representation_items** and a context which is a second Cartesian coordinate space. These **representations** are related by rigid motion in a **representation_relationship_with_transformation** that uses an **item_defined_transformation**. The **transform_item_1** and **transform_item_2** might each be an **axis2_placement_3d** where each is in the respective Cartesian coordinate spaces. The meaning of such a **representation_relationship** would be to relate the two **representations** by overlaying the two **axis2_placements**.

4.4.8 functionally_defined_transformation

A **functionally_defined_transformation** is a transformation performed by defining an explicit transformation function, the function f between a domain A and range B. The function f:A→B takes each element a in A and maps it to an element b in B, f(a)=b.

EXPRESS specification:

```
*)
ENTITY functionally_defined_transformation;
  name        : label;
  description : text;
END_ENTITY;
(*
```

Attribute definitions:

name: an identifier of the **functionally_defined_transformation**.

description: a description of the **functionally_defined_transformation**.

4.4.9 representation_relationship_with_transformation

A **representation_relationship_with_transformation** is an association between two **representations** such that the contexts of the **representations** are related through the transformation. This transformation may be used to determine the relationship between **representation.items** in the respective **representations**.

NOTES

1 - Where the related representations both have geometric contents, the **transformation** would be used to calculate the distances between **representation_items** in the two **representations**.

2 - The existence of a **related_representation_with_transformation** entity does not necessarily imply that there is any correspondence between **representation_items** in the two **representations**. Any such correspondence and other constraints may be defined in specializations of this entity.

EXPRESS specification:

```
*)
ENTITY representation_relationship_with_transformation
  SUBTYPE OF (representation_relationship);
  transformation_operator : transformation;
WHERE
  WR1:
   SELF\representation_relationship.rep_1.context_of_items
   :<>: SELF\representation_relationship.rep_2.context_of_items;
END_ENTITY;
(*
```

Attribute definitions:

SELF\representation_relationship.rep_1: the representation with a context to which the transformation applies.

SELF\representation_relationship.rep_2: the representation with a context which is the result of the transformation.

transformation_operator: a **transformation** that relates the **representation.context_of_items** of the two related **representations**.

NOTE 3 - The transformation that relates the **representation.items** of one **representation** with the **representation.items** in a second **representation** may be specified by operating with the trees of **representation_items** that define the respective **representation.items**.

Formal propositions:

WR1: the contexts of the two related **representations** shall not be the same.

Informal propositions:

IP1: when the transformation is an **item_defined_transformation**, the ordering of the **representations** given for the inherited attributes of **representation_relationship** shall be consistent with the ordering of the two **representation_items** given as attributes of **item_defined_transformation**.

4.4.10 representation_map

A **representation_map** is the identification of a **representation** and a **representation_item** in that **representation** for the purpose of mapping. The **representation_item** defines the origin of the mapping. The **representation_map** is used as the source of a mapping by a **mapped item**.

NOTE 1 - The definition of a mapping which is used to specify a new representation_item comprises a **representation_map** and a **mapped_item** entity. Without both entities, the mapping is not fully defined. Two entities are specified to allow the same source representation (**representation_map.mapped_representation**) to be mapped into multiple new representations (**mapped_items**).

EXPRESS specification:

```
*)
ENTITY representation_map;
  mapping_origin        : representation_item;
  mapped_representation : representation;
INVERSE
  map_usage : SET[1:?] OF mapped_item FOR mapping_source;
WHERE
  WR1: item_in_context(SELF.mapping_origin,
        SELF.mapped_representation.context_of_items);
END_ENTITY;
(*
```

Attribute definitions:

mapping_origin: a **representation_item** about which the **mapped_representation** is mapped.

NOTE 2 - Consider the Cartesian mapping of one geometric **representation** to another. The **mapping_origin** might be an **axis2_placement_3d** in the context of **mapped_representation** that defines the position about which it is mapped.

mapped_representation: a **representation** that is mapped to at least one **mapped_item**.

map_usage: the set of one or more **mapped_items** to which the **representation_map** is mapped.

Formal propositions:

WR1: the **mapping_origin** shall be in the **representation_context** of the **mapped_representation**.

4.4.11 mapped_item

A **mapped_item** is the use of an existing representation (the **mapping_source.mapped_representation**) as a **representation_item** in a second **representation**.

NOTE 1 - A **mapped_item** is a subtype of **representation_item**. It enables a **representation** to be used as a **representation_item** in one or more other **representations**. The **mapped_item** allows for the definition of a **representation** using other **representations**.

The mapping is achieved through an operator that is implicitly defined by the **mapping_source.-mapping_origin** and the **mapping_target** attributes. In this respect, the mapping is specified in the same way as for an **item_defined_transformation**.

EXPRESS specification:

```
*)
ENTITY mapped_item
  SUBTYPE OF (representation_item);
  mapping_source : representation_map;
  mapping_target : representation_item;
WHERE
  WR1: acyclic_mapped_representation(using_representations(SELF), [SELF]);
END_ENTITY;
(*
```

Attribute definitions:

mapping_source: a **representation_map** that is the source of the **mapped_item**;

mapping_target: a **representation_item** that is the target onto which the **mapping_source** is mapped.

Formal propositions:

WR1: a **mapped_item** shall not be self-defining by participating in the definition of the **representation** being mapped. See 4.5.1.

NOTES

2 - The details of how any particular mapping is achieved is left to various specializations of **mapped_item** and **representation_map**.

3 - For example, consider the Cartesian mapping of one geometric **representation** to another. The **mapping_source** might be a **representation_map** referencing a **representation** and an **axis_placement_3d** founded in the **geometric_representation_context** of the referenced **representation**. The **mapped_item** might be a reference to this **representation_map** and a second **axis_placement_3d**. The **mapped_item** would then be a **representation_item** that is a mapping of the referenced **representation** such that the **representation_map.mapping_origin** is overlaid onto the **mapped_item.mapping_target**.

4.4.12 definitional_representation

A **definitional_representation** is a **representation** defined in a **parametric_representation_context**.

EXPRESS specification:

```
*)
ENTITY definitional_representation
  SUBTYPE OF ( representation );
WHERE
  WR1: 'REPRESENTATION_SCHEMA.PARAMETRIC_REPRESENTATION_CONTEXT' IN
        TYPEOF( SELF\representation.context_of_items );
END_ENTITY;
(*
```

Formal propositions:

WR1: The context of the definitional_representation shall be a **parametric_representation_context**.

4.4.13 parametric_representation_context

A **parametric_representation_context** is a **representation_context** in which the **representation_items** are defined in some parametric space. The length units in this space are dimensionless.

NOTE - The definition of the parametric space is given with specializations and use of this entity in other parts of ISO 10303.

EXPRESS specification:

```
*)
ENTITY parametric_representation_context
  SUBTYPE OF (representation_context);
END_ENTITY;
(*
```

Informal propositions:

IP1: If the entity instance is also an instance of **global_unit_assigned_context**, then the **global_unit_assigned_context.units** attribute shall not include a **length_unit**. See ISO 10303-41. In the **parametric_representation_context** all length measures are in unitless parameter space.

4.5 Representation_schema function definitions

This subclause contains the EXPRESS function definitions in the **representation_schema.**

4.5.1 acyclic_mapped_representation

The function **acyclic_mapped_representation** determines if a given **mapped_item** is self-defining by virtue of mapping a **representation** in which the **mapped_item** is used. The function is extended to check both the **mapped_representation** and the **mapped_representation.items** recursively for any **mapped_-items** or **representation_items** referencing a **mapped_item** that might cause a self defining reference. This function returns TRUE if the input candidate **representation_item** does not cause self definition. It returns FALSE otherwise. The type of the function is **BOOLEAN**.

This function is used to constrain the entity **mapped_item**. See 4.4.11.

EXPRESS specification:

```
*)
FUNCTION acyclic_mapped_representation
 (parent_set    : SET OF representation;
  children_set : SET OF representation_item) : BOOLEAN;
 LOCAL
   x,y : SET OF representation_item;
   i   : INTEGER;
 END_LOCAL;

 -- Determine the subset of children_set that are mapped_items.

 x := QUERY(z <* children_set ¦ 'REPRESENTATION_SCHEMA.MAPPED_ITEM'
      IN TYPEOF(z));

 -- Determine that the subset has elements.

 IF SIZEOF(x) > 0 THEN

   -- Check each element of the set.

   REPEAT i := 1 TO HIINDEX(x);

     -- If the selected element maps a representation in the
     --   parent_set, then return false.

     IF x[i]\mapped_item.mapping_source.mapped_representation
       IN parent_set THEN
       RETURN (FALSE);
     END_IF;

     -- Recursive check of the items of mapped_rep.
     IF NOT acyclic_mapped_representation
       (parent_set + x[i]\mapped_item.mapping_source.mapped_representation,
       x[i]\mapped_item.mapping_source.mapped_representation.items) THEN
       RETURN (FALSE);
     END_IF;
   END_REPEAT;
 END_IF;

 -- Determine the subset of children_set that are not
 -- mapped_items.
```

```
  x := children_set - x;

  -- Determine that the subset has elements.

  IF SIZEOF(x) > 0 THEN

    -- For each element of the set:

    REPEAT i := 1 TO HIINDEX(x);

      -- Determine the set of representation_items referenced.

      y := QUERY(z <* bag_to_set( USEDIN(x[i], '')) |
           'REPRESENTATION_SCHEMA.REPRESENTATION_ITEM' IN TYPEOF(z));

      -- Recursively check for an offending mapped_item.
      -- Return false for any errors encountered.

      IF NOT acyclic_mapped_representation(parent_set, y) THEN
        RETURN (FALSE);
      END_IF;
    END_REPEAT;
  END_IF;

  -- Return true when all elements are checked and
  -- no error conditions found.
  RETURN (TRUE);
END_FUNCTION;
(*
```

Argument definitions:

parent_set: the set of **representations** in which the **mapped_item** is used. This is input to the function. On initial input, this is the set of **representations** in which the **mapped_item** being checked is used and is modified in recursive calls.

children_set: the set of **representation_items** that might possibly be a **mapped_item** and are referenced directly or indirectly through the **items** of the **representations** in the **parent_set**. This is input to the function. On initial input this is the **mapped_item** being checked and is modified in recursive calls.

4.5.2 item_in_context

The function **item_in_context** determines if a **representation_item** is related to a **representation_context**. The function returns TRUE if:

- the **item** argument is related by a **representation** to the input **cntxt** argument.

- the **item** argument is related by a **definitional_representation** to the input **cntxt** argument.

Function **item_in_context** returns FALSE otherwise. The type of the function is **BOOLEAN**.

A **representation_item** is related to a **representation_context** if it is any of the following:

a) referenced in the set of **items** of a **representation** where **cntxt** appears as the **context_of_items**;

b) referenced in the set of **items** of a **definitional_representation_item** where **cntxt** appears as the **context_of_items**;

c) referenced by a **representation_item** that is an **item_in_context** of the **cntxt**.

NOTES

1 - The third condition is a recursive check allowing for a **representation_item** to be related to a **representation_context** by being part of a tree of related **representation_items**. The tree is rooted in an entity that is related to a **representation_context** by fulfilling the first or second condition.

2 - The function **item_in_context** only determines if an **item** is related to a specific **representation_context**. The relationship of the **item** to some other **representation_context** is not determined.

EXPRESS specification:

```
*)
FUNCTION item_in_context
  (item   : representation_item;
   cntxt : representation_context) : BOOLEAN;
  LOCAL
    i : INTEGER;
    y : BAG OF representation_item;
  END_LOCAL;

  -- If there is one or more representation using both the item
  -- and cntxt return true.

  IF SIZEOF(USEDIN(item,'REPRESENTATION_SCHEMA.REPRESENTATION.ITEMS')
    * cntxt.representations_in_context) > 0 THEN
    RETURN (TRUE);

    -- Determine the bag of representation_items that reference
    -- item.

    ELSE
    y := QUERY(z <* USEDIN (item , '') |
         'REPRESENTATION_SCHEMA.REPRESENTATION_ITEM' IN TYPEOF(z));

     -- Ensure that the set is not empty.

      IF SIZEOF(y) > 0 THEN

      -- For each element in the set

      REPEAT i := 1 TO HIINDEX(y);

        -- Check to see it is an item in the input cntxt.

        IF item_in_context(y[i], cntxt) THEN
          RETURN (TRUE);
        END_IF;
      END_REPEAT;
    END_IF;
  END_IF;

  -- Return false when all possible branches have been checked
  -- with no success.
  RETURN (FALSE);
END_FUNCTION;
(*
```