INTERNATIONAL STANDARD

ISO 10303-43

Second edition
2000-08-15

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

Part 43:
Integrated generic resource: 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 10303-43:2000(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

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.ch
Web www.iso.ch

Printed in Switzerland

Contents

Figures

Tables

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 3.

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.

International Standard ISO 10303-43 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 4, *Industrial data*.

This second edition of ISO 10303-43 constitutes a technical revision of the first edition (ISO 10303-43:1994), which is provisionally retained to support continued use and maintenance of implementations based on the first edition, and to satisfy the normative references of other parts of ISO 10303. This edition incorporates the corrections published in ISO 10303-43:1994/Cor.1:1999 and ISO 10303-43:1994/Cor.2:2000.

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 interpreted constructs, application protocols, abstract test suites, implementation methods, and conformance testing. The series are described in ISO 10303-1.

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

<http://www.nist.gov/sc4/editing/step/titles/>.

This part of ISO 10303 is a member of the integrated resources series. The integrated resources specify a single conceptual product data model.

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

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 specifies the **representation_schema**. This schema specifies the resource constructs that group elements of product data into collections in order to describe aspects of products, particularly properties of products. The relationships of the schema in this part of ISO 10303 to other schemas that define the integrated resources of this International Standard are illustrated in Figure 1 using the EXPRESS-G notation. EXPRESS-G is defined in annex D of ISO 10303-11. The **application_context_schema**, **product_definition_schema**, **product_property_definition_schema**, and **product_property_representation_schema** are specified in ISO 10303-41. The schemas illustrated in Figure 1 are components of the integrated resources.

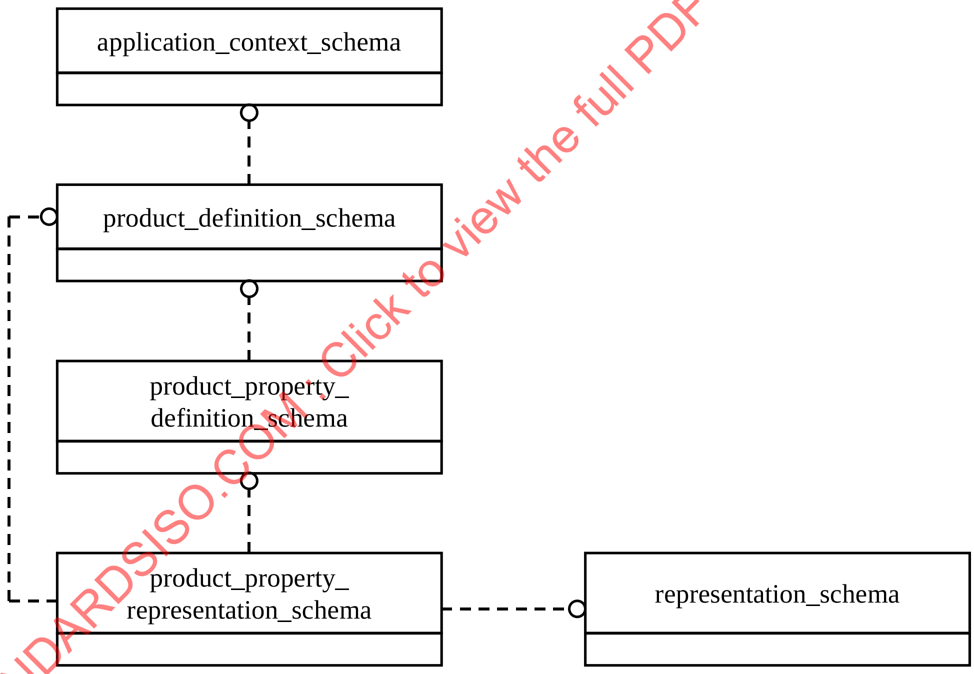

Figure 1 – Relationship of the representation_schema to the ISO 10303 integration architecture

This edition incorporates modifications that are upwardly compatible with the previous edition. Modifications to EXPRESS specifications are upwardly compatible if:

— instances encoded according to ISO 10303-21, and that conform to an ISO 10303 application protocol based on the previous edition of this part, also conform to a revision of that application protocol based on this edition;

— interfaces that conform to ISO 10303-22 and to an ISO 10303 application protocol based on the previous edition of this part, also conform to a revision of that application protocol based on this edition;

— the mapping tables of ISO 10303 application protocols based on the previous edition of this part remain valid in a revision of that application protocol based on this edition.

Technical modifications to ISO 10303-43:1994 are categorized as follows: changes to the EXPRESS declarations, new EXPRESS declarations, and changes to definitions of EXPRESS entity data types.

The following EXPRESS declarations have been modified:

— **acyclic_mapped_representation**;

— **item_in_context**;

— **representation**;

— **uncertainty_measure_with_unit**;

— **using_representations**.

The following EXPRESS declarations have been added:

— **compound_item_definition**;

— **compound_representation_item**;

— **founded_item**;

— **founded_item_select**;

— **list_representation_item**;

— **representation_item_relationship**;

— **set_representation_item**;

— **uncertainty_assigned_representation**;

— **using_items**;

— **valid_measure_value**;

— **value_representation_item**.

The definitions of the following EXPRESS data types have been modified:

— **functionally_defined_transformation**;

— **global_uncertainty_assigned_context**;

— **item_defined_transformation**;

— **mapped_item**;

— **parametric_representation_context**;

— **representation_context**;

— **representation_item**;

— **representation_map**;

— **representation_relationship**;

— **representation_relationship_with_transformation**.

In this International Standard the same English language words may be used to refer to an object in the real world or to a concept, and as the name of an EXPRESS data type that represents this object or concept. The following typographical convention is used to distinguish between these. If a word or phrase occurs in the same typeface as narrative text, the referent is the object or concept. If the word or phrase occurs in a bold typeface, the referent is the EXPRESS data type. Names of EXPRESS schemas also occur in a bold typeface.

The name of an EXPRESS data type may be used to refer to the data type itself, or to an instance of the data type. The distinction between these uses is normally clear from the context. If there is a likelihood of ambiguity, the phrase “entity data type” or “instance(s) of” is included in the text.

Double quotation marks “ ” denote quoted text. Single quotation marks ‘ ’ denote particular text string values.

Numbers in brackets [n] are references to documents listed in the Bibliography.

Several components of this part of ISO 10303 are available in electronic form. This access is provided through the specification of Universal Resource Locators (URLs) that identify the location of these files on the Internet. If there is difficulty accessing these files contact the ISO Central Secretariat, or contact the ISO TC 184/SC4 Secretariat directly at: sc4sec@cme.nist.gov.

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

1 Scope

This part of ISO 10303 specifies the resource constructs that group elements of product data into collections in order to describe aspects of products. This part of ISO 10303 is applicable to the description of properties of products. The following are within the scope of this part of ISO 10303:

- the specification of 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 transformation of one element of representation to another by specifying the input and output of the transformation;
- the specification of the transformation of one element of representation to another by specifying the transforming function.

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

- the complete specification of 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 if 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 can be specified in an annotated EXPRESS schema that uses or specializes this schema.

- 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 normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO 10303. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO 10303 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/IEC 8824-1:1995, *Information technology – Abstract Syntax Notation One (ASN.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:—[1], *Industrial automation systems and integration – Product data representation and exchange – Part 41: Integrated generic resource: Fundamentals of product description and support.*

3 Terms and definitions

3.1 Terms defined in ISO 10303-1

For the purposes of this part of ISO 10303, the following terms defined in ISO 10303-1 apply:

- application;
- application protocol;
- assembly;
- data;
- information;
- integrated resource;
- product;

[1] To be published. (Revision of ISO 10303-41:1994)

- product data;
- structure.

3.2 Terms defined in ISO 10303-41

For the purposes of this part of ISO 10303, the following terms defined in ISO 10303-41 apply:

- agreement of common understanding;
- annotated EXPRESS schema.

3.3 Other terms and definitions

For the purposes of this part of ISO 10303, the following terms and definitions apply:

3.3.1
context of representation
the basis through which elements of a representation are related to each other

3.3.2
element of representation
a data element that participates in the description of a representation, either directly or by describing other elements of representation

3.3.3
founded
the property of an element of representation that results from its association, direct or indirect, with a context of representation

3.3.4
representation
an organized collection of associated data elements, collected together for one or more specific uses

4 Representation

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

<u>EXPRESS specification:</u>

```
*)
SCHEMA representation_schema;

REFERENCE FROM basic_attribute_schema -- ISO 10303-41
  (get_description_value,
   get_id_value);

REFERENCE FROM measure_schema -- ISO 10303-41
  (measure_value,
   measure_with_unit);
```

```
REFERENCE FROM support_resource_schema -- ISO 10303-41
  (bag_to_set,
   identifier,
   label,
   text);

 (*
```

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

basic_attribute_schema	ISO 10303-41
measure_schema	ISO 10303-41
support_resource_schema	ISO 10303-41

NOTE 2 See annex D for a graphical presentation of this schema using the EXPRESS-G notation.

NOTE 3 A listing of the complete EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text, is available from the Internet – see annex C.

4.1 Introduction

The subjects of the **representation_schema** are the structures that relate a collection of elements of product data to a context. These structures are representations. Representations are used to describe aspects of products. The associations between representations and the aspects of products they describe are specified in annotated EXPRESS schemas that use or specialize this schema. Representations can be used to:

— describe a property or a relationship between two properties, in which the properties are associated with a complete product or with a part of a product;

NOTE 1 The use of representations to collect elements of product data in order to describe the properties of a product is specified in ISO 10303-41, and can be specified in annotated EXPRESS schemas that use or specialize the constructs specified in this part of ISO 10303 and in ISO 10303-41.

— describe a picture.

NOTE 2 The use of representations to collect elements of product data in order to describe a picture of a product is specified in ISO 10303-46 [6], and can be specified in annotated EXPRESS schemas that use or specialize the constructs specified in this part of ISO 10303 and in ISO 10303-46.

One representation can be part of another representation.

EXAMPLE 1 A collection of lines and points describes the shape of a wall. This representation can be used as part of the description of the shape of the building, of which the wall is a part.

Each representation has a context and a collection of elements specified in that context

EXAMPLE 2 For the representation of geometric elements, the context is a coordinate space.

A context can be related to other contexts.

EXAMPLE 3 Local coordinate spaces can be defined for each building in a factory complex. These coordinate spaces can be related to each other, and to the coordinate space of the factory complex itself.

Transformations between representations can be specified.

4.2 Fundamental concepts and assumptions

4.2.1 Representation

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

a) A representation consists of a collection of elements of representation and a context. A representation corresponds to an instance of the **representation** entity data type. By structuring the elements in this way, relationships between elements can be established. Elements of representation are related if:

   1) they are elements in the same representation, or

   2) they are elements in different representations that have the same context, or

   3) they are elements in different representations that have different contexts, if the contexts are related.

EXAMPLE 1 Consider two points with coordinate values of (0,0,0) and (1,0,0). It is not possible to calculate the distance between these points 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 part of ISO 10303, a point is an example of an element of representation, and a coordinate space is an example of a context.

b) A representation can be used more than once. A representation is separate from its use.

EXAMPLE 2 Consider a collection of points and lines in a coordinate space. This collection can be used to describe the shape of a product. The collection can also be used, possibly with some transformation, to describe a drawing or picture of the product. Neither of these uses is part of the definition of the collection itself.

NOTE Representations can be specialized further in annotated EXPRESS schemas that use or specialize this schema.

4.2.2 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. The context of a representation corresponds to an instance of the **representation_context** entity data type.

— The context of a representation exists only as a basis for the representations that use it. Therefore, representation contexts are related only if representations using the contexts are related.

NOTE Representation contexts can be specialized further in annotated EXPRESS schemas that use or specialize this schema.

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

4.2.3 Elements of representation

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

- Elements of representation participate directly in a representation, or support the definition of another element of representation, or both.

EXAMPLE 1 A point could be the only element in the representation of the location of a product, or it could serve as the end point 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.

EXAMPLE 2 An element of representation that specifies the presentation of a text string could be part of the description of a drawing. In this case it participates directly in a representation. The same element of representation could, alternatively, be part of a dimension callout. In this second case the text is part of the callout, and participates indirectly in the description of the drawing.

NOTE Elements of representation that describe the presentation of text strings are specified in ISO 10303-46 [6]. Elements of representation that describe dimension callouts are described in ISO 10303-101 [8].

- Elements of representation can refer to each other, thereby forming graphs of such elements in which each graph has an identifiable root. The association of a root element with a context associates all elements in the graph with the context.

EXAMPLE 3 A curve is defined by a number of points. These points are all in the same coordinate space as the curve by virtue of their reference from the curve.

- Elements of representation are collected into representations and associated with contexts as a basis for establishing relationships between elements.

- An element of representation corresponds to an instance of the **representation_item** entity data type.

- An association of one or more elements of representation with a context corresponds to an instance of the **representation** entity data type.

4.2.4 Association of representations

The following concepts and assumptions apply to the portions of this schema that deal with the association of representations.

- A representation can be related to another representation.

- One representation can be related to another representation such that they both participate in the association, but one does not define the other. This type of association corresponds to the **representation_relationship** entity data type.

- One representation can be related to another representation such that the first is part of the definition of the second. This type of association corresponds to the **mapped_item** and **representation_map** entity data types.

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

EXAMPLE Two collections of points and lines 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 can exist for the shape of an assembly of 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 through an association of the representations of the parts with the representation of the assembly.

4.2.5 Transformation

The following concepts and assumptions apply to the portions of this schema that deal with transformations.

a) Elements in different representations can be compared if

   1) the representations have the same context, or

   2) a transformation is defined that relates the representations to each other.

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

   1) the set of elements a to be transformed;

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

   3) the definition of the context A that is common to the set of elements a;

   4) the definition of the context B that is common to the set of elements b;

   5) the function f.

   The domain A and the range B are instances of the **representation_context** entity data type. The elements a and b are instances of the **representation_item** entity data type. The relationships between a and A, and between b and B, are instances of the **representation** entity data type: a is an element in a representation whose context is A, b is an element in a representation whose context is B.

c) Two different approaches are used in this part of ISO 10303 to specify transformations.

   1) The function f can be specified. This type of transformation corresponds to the **functionally_defined_transformation** entity data type.

EXAMPLE 1 Two representations are related such that one is rotated and skewed with respect to the other. This transformation can be specified by a matrix.

NOTE 1 The data structures for particular kinds of transformation functions, such as matrices, are not specified in this part of ISO 10303.

EXAMPLE 2 Points on a map are related to points on the surface of the earth by a function that transforms the three dimensional surface to a two dimensional picture, and applies a scaling factor.

   2) An element a in context A and an element b in context B that are sufficient to derive the function can be specified. This type of transformation corresponds to the **item_defined_transformation** entity data type, or to the mapping defined by the **mapped_item** entity data type.

EXAMPLE 3 A translation between coordinate spaces can be uniquely determined by two instances of **axis2_placement_3d** a1 and b1 (one in each coordinate space), such that f takes a1 and transforms it to b1.

NOTE 2 The **axis2_placement_3d** entity data type is defined in ISO 10303-42 [3,4].

4.2.6 Uncertainty

Numeric values that are measured or calculated can be imprecise. Uncertainty is a measure of the interval of confidence associated with this imprecision. In this International Standard, uncertainty can be specified for:

a) multiple representations that share a common context;

b) individual representations;

c) individual elements of representation.

This part of ISO 10303 supports (a) and (b).

NOTE 1 (c) is supported by ISO 10303-45 [5].

NOTE 2 Uncertainty is unrelated to the subject of tolerances or permitted variations. These are supported by ISO 10303-47 [7].

The following concepts and assumptions apply to the portions of this schema that deal with uncertainty:

— The uncertainty for numeric values can be specified for all the representations that share a context. This is specified using the **global_uncertainty_assigned_context** entity data type.

— The uncertainty for numeric values can be specified for a representation in a given context. This is specified using the **uncertainty_assigned_representation** entity data type.

If uncertainties are specified more than once, the following precedence rules shall apply. The uncertainty specified for an individual element of representation shall have precedence over the uncertainty specified by any **uncertainty_assigned_representation** in which the item participates. The uncertainty specified by an **uncertainty_assigned_representation** shall have precedence over the uncertainty specified by any **global_uncertainty_assigned_context** in which the representation participates.

NOTE 3 Uncertainty for an individual element of representation is specified by the **qualified_-representation_item** entity data type, defined in ISO 10303-45 [5].

4.3 Representation type definitions

4.3.1 compound_item_definition

A **compound_item_definition** is a selection between different aggregations of **representation_item** instances.

EXPRESS specification:

```
*)
TYPE compound_item_definition = SELECT
  (list_representation_item,
   set_representation_item);
END_TYPE;
(*
```

4.3.2 founded_item_select

A **founded_item_select** is a selection between a **founded_item** and a **representation_item**.

EXPRESS specification:

```
*)
TYPE founded_item_select = SELECT
  (founded_item,
   representation_item);
END_TYPE;
(*
```

4.3.3 list_representation_item

A **list_representation_item** is an ordered aggregation of **representation_item** instances.

EXPRESS specification:

```
*)
TYPE list_representation_item = LIST [1:?] OF representation_item;
END_TYPE;
(*
```

4.3.4 set_representation_item

A **set_representation_item** is an unordered aggregation of **representation_item** instances.

EXPRESS specification:

```
*)
TYPE set_representation_item = SET [1:?] OF representation_item;
END_TYPE;
(*
```

4.3.5 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 entity definitions

4.4.1 compound_representation_item

A **compound_representation_item** is a type of **representation_item** that is defined by an aggregation of other instances of **representation_item**. This aggregation is either ordered or unordered.

NOTE 1 A **compound_representation_item** supports the description of aspects of product data using structured collections.

NOTE 2 The meaning and usage of a structured collection of instances of **representation_item** can be specified in an annotated EXPRESS listing that uses or specializes this entity data type.

EXAMPLE In an application protocol in the domain of ship design, hydrostatic properties of the ship hull can be represented using a tabular structure composed of instances of **list_representation_item**.

EXPRESS specification:

```
*)
ENTITY compound_representation_item
  SUBTYPE OF (representation_item);
  item_element : compound_item_definition;
END_ENTITY;
(*
```

Attribute definitions:

item_element: the **list_representation_item** or **set_representation_item** whose component instances of **representation_item** define the **compound_representation_item**.

4.4.2 definitional_representation

A **definitional_representation** is a type of **representation** that has 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**.

4.4.3 founded_item

The **founded_item** entity data type represents part of an element of representation. A **founded_item** can be used only as part of the definition of a **representation_item**, and is founded through the participation of the **representation_item** in a representation. Further, any instances of **representation_item** that are part of a **founded_item** are founded through that participation. A **founded_item** cannot be an item in a **representation**.

NOTE 1 This entity data type allows errors in other parts of ISO 10303 to be corrected in an upwardly compatible manner. Specifically, it is a supertype of the entity data types **composite_curve_segment** and **surface_patch**, defined in ISO 10303-42 [3,4], and **view_volume**, defined in ISO 10303-46 [6].

NOTE 2 This entity data type is not intended to be instantiated as a complex instance of two or more of its defined subtypes **composite_curve_segment, surface_patch**, and **view_volume**.

NOTE 3 This entity data type is not intended for any other use apart from the correction of the errors in other parts of ISO 10303 listed in NOTE 1 above.

EXAMPLE A **bounded_curve** that is the parent curve of a **composite_curve_segment** is founded as a result of the **composite_curve_segment** being a **founded_item** that is part of a **representation_item** (a **composite_-curve**). This provides the necessary relationship between the **bounded_curve** and a **geometric_-representation_context**.

NOTE 4 The **bounded_curve**, **composite_curve**, and **geometric_representation context** entity types are defined in ISO 10303-42 [3,4].

EXPRESS specification:

```
*)
ENTITY founded_item;
END_ENTITY;
(*
```

Informal propositions:

IP1: Each **founded_item** shall participate in the definition of a **representation_item**.

4.4.4 functionally_defined_transformation

The **functionally_defined_transformation** entity data type represents a transformation that is defined by an explicit transformation function, the function f between a domain A and range B.

NOTE 1 Let f be the transformation function between 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, i.e., f(a)=b.

NOTE 2 The transformation function can be specified in an annotated EXPRESS schema that uses or specializes this entity data type, or in an agreement of common understanding between the partners sharing this information.

NOTE 3 The function f can be specified in specializations of this entity type, or through constraints on the population and usage of the **description** attribute.

EXAMPLE 1 An annotated EXPRESS schema could define a **subtype x_y_plane_mirror_transformation** of this entity type; the definition of the subtype would state that the nature of the functionally defined transformation is to mirror all instances of **representation_item** in the x-y plane.

EXAMPLE 2 An annotated EXPRESS schema could associate different natural language translation functions with the values ‘English to French’ and ‘French to English’, where these are populations of the **description** attribute of **functionally_defined_transformation**.

EXPRESS specification:

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

Attribute definitions:

name: the **label** by which the **functionally_defined_transformation** is known.

description: the **text** that characterizes the **functionally_defined_transformation**. The value of the description need not be specified.

4.4.5 global_uncertainty_assigned_context

A **global_uncertainty_assigned_context** is a **representation_context** that specifies uncertainty for the elements of representation that are associated with it. The uncertainty is specified by instances of **uncertainty_measure_with_unit** (see 4.4.17), and applies to all elements of representation that are expressed in the same measure and that are associated with a **representation** that has this **global_uncertainty_assigned_context**.

NOTE The precedence rules that apply to uncertainties in numeric quantities are specified in 4.2.6.

EXAMPLE An instance of **global_uncertainty_assigned_context** specifies uncertainty of 0.01m with respect to lengths. Unless modified by the precedence rules specified in 4.2.6, this uncertainty applies to each length that occurs in each **representation_item** associated with a **representation** that has this **global_uncertainty_assigned_context**.

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 instances of **uncertainty_measure_with_unit** that apply in the **representation_context**.

4.4.6 item_defined_transformation

An **item_defined_transformation** is a transformation that is defined by two instances of **representation_item** in which one instance of **representation_item** is the result of applying the transformation function to the other. The transformation function is not explicitly provided, but it is derived from the relationship between the instances of **representation_item**.

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, i.e., f(a)=b.

NOTE 1 In special cases, the inverse transformation function g:B→A can also be derived as appropriate.

EXPRESS specification:

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

Attribute definitions:

name: the **label** by which the **item_defined_transformation** is known.

NOTE 2 The name could designate a particular instance of **item_defined_transformation**, or the kind of information that an instance of **item_defined_transformation** conveys.

description: the **text** that characterizes the **item_defined_transformation**. The value of the **description** need not be specified.

transform_item_1: the first instance of **representation_item** that describes the transformation function.

transform_item_2: the second instance of **representation_item** that describes the transformation function.

EXAMPLE Consider one **representation** having a set of instances of **representation_item** and a context that is a cartesian coordinate space and a second **representation** with another set of instances of **representation_item** and a context that is a second cartesian coordinate space. These instances of **representation** 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 space. The meaning of such a **representation_relationship** would be to relate the two instances of **representation** such that the transformation between the two instances of **axis2_placement_3d** applies to every element in the representations.

4.4.7 mapped_item

A **mapped_item** is a type of **representation_item** that specifies the mapping of a **representation** as an element of the items of a second **representation**.

NOTE 1 See 4.2.4 for the fundamental concepts and assumptions that apply to this entity data type.

NOTE 2 The mapping includes a transformation that is derived from the **mapping_source.mapping_origin** and the **mapping_target** attributes. See 4.2.5 for the fundamental concepts and assumptions that apply to transformation.

NOTE 3 The precise meaning of the mapping can be specified in an annotated EXPRESS schema that uses or specializes this entity data type and the **representation_map** entity data type, or in an agreement of common understanding between the partners sharing this information.

EXAMPLE Figure 2 illustrates the use of the **mapped_item** and **representation_map** entity data types. Three instances of representation are shown. The first representation R_1 consists of geometry G_1 and an **axis2_placement_3d** A_1. The second **representation** R_2 consists of geometry G_2 and an **axis2_placement_3d** A_2. For the purpose of this example, the nature and structure of G_1 and G_2 are not relevant. R_1 represents the shape of a roof. R_2 represents the shape of some walls.

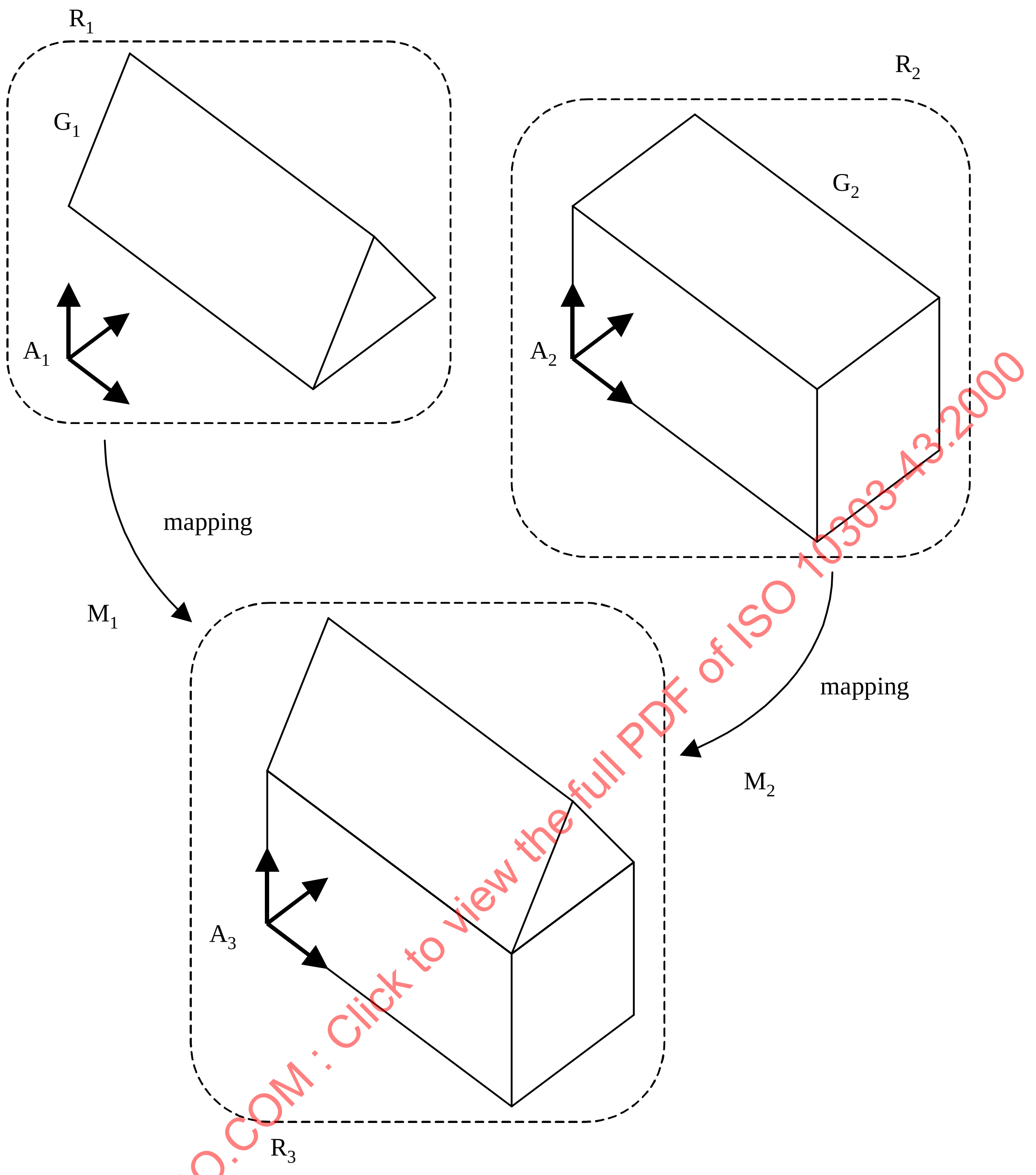

Figure 2 – Example of the use of mapped_item and representation_map

Two instances of **representation_map** (see 4.4.13) allow R_1 and R_2 to be used as elements in a third **representation**, R_3. R_3 represents the shape of the building. The first instance of **representation_map** RM_1 references R_1 as its mapped_representation and A_1 as its mapping_origin. The second instance of **representation_map** RM_2 references R_2 as its mapped_representation and A_2 as its mapping_origin.

R_3 contains as its items an **axis2_placement_3d** and two instances of **mapped_item**, M_1 and M_2. M_1 references RM_1 as its mapping_source and A_3 as its mapping target. M_2 references RM_2 as its mapping_source and A_3 as its mapping target. The result is that R_3 uses R_1 and R_2 as parts of its definition. As it is used in R_3, R_1 is transformed so that A_1 is mapped onto A_3. As it is used in R_3, R_2 is transformed so that A_2 is mapped onto A_3.

This example shows how the **mapped_item** and **representation_map** entity data types can be used to describe the composition of one **representation** from other instances of **representation**. See 4.4.14 for an example of the use of **representation_relationship** to describe whole-part associations between instances of representation.

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: the **representation_map** that specifies the source and the origin of the **mapped_item**.

mapping_target: the **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.

4.4.8 parametric_representation_context

A **parametric_representation_context** is a **representation_context** in which instances of **representation_item** are defined in a parametric space.

NOTE 1 The definition of the parametric space can be specified in an annotated EXPRESS schema that uses or specializes this entity data type.

NOTE 2 In a **parametric_representation_context**, length units are dimensionless.

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**.

NOTE 3 the **global_unit_assigned_context** and **length_unit** entity data types are defined in ISO 10303-41.

4.4.9 representation

A **representation** is a collection of one or more **representation_item** instances that are related in a specified **representation_context**.

NOTE 1 The use of a **representation**, i.e., that which is being represented, is not specified in this part of ISO 10303. It can be specified in an annotated EXPRESS schema that uses or specializes this entity data type.

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

EXAMPLE 1 Two cartesian points P and Q (described by instances of **representation_item**) are related in a context A (they are elements in the same representation in context A, or are elements in different representations that share context A). It is therefore possible to calculate the distance between these points. A third cartesian point R (also described by an instance of **representation_item**) is not related to context A. It is not possible to determine the distance between R and P, or between R and Q.

A **representation_item** can be related to a **representation_context** directly, when it occurs as an element in a representation, or indirectly, when it is referenced through any number of intervening entities, each of type **representation_item** or **founded_item**.

A **representation** relates a **representation_context** to trees of **representation_item** instances each tree being rooted in one member of the set of **items**. A **representation_item** or **founded_item** is one node in the tree; a relationship between one **representation_item** or **founded_item** and another is an edge.

NOTE 2 Instances of **representation_item_relationship** (see 4.4.12) do not form either nodes or edges in this tree; an instance of **representation_item** is not part of the tree solely because it is associated with an element in the tree by an instance of **representation_item_relationship**.

NOTE 3 A **representation** can be incomplete in that it need not fully model the concept that is represented, although it could be adequate for a given application.

EXAMPLE 2 Consider a collection of two-dimensional **representation_item** instances 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.

NOTE 4 Two instances of **representation** are not related solely because the same instance of **representation_item** is referenced directly or indirectly from their sets of **items**.

EXAMPLE 3 Consider a surface that is used in the respective 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 instances of **representation_context** (i.e., coordinate spaces): one for the die and one for the part by the two instances of **representation**. However, the two instances of **representation** are not related; they simply share a common **representation_item**.

NOTE 5 Two instances of **representation** are not related solely because instances of **representation_item** in their sets of items are related by an instance of **representation_item_relationship** (see 4.4.12).

EXPRESS specification:

```
*)
ENTITY representation;
  name              : label;
  items             : SET[1:?] OF representation_item;
  context_of_items : representation_context;
DERIVE
  id                : identifier := get_id_value (SELF);
  description       : text := get_description_value (SELF);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                             'ID_ATTRIBUTE.IDENTIFIED_ITEM'))
       <= 1;
  WR2: SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                             'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM'))
       <= 1;
 END_ENTITY;
(*
```

Attribute definitions:

name: the **label** by which the **representation** is known.

NOTE 6 The name could designate a particular instance of **representation,** or the kind of information that an instance of **representation** conveys.

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.

id: the **identifier** that distinguishes the **representation**.

NOTE 7 This attribute is an upwardly compatible addition to **representation** as specified in ISO 10303-43:1994.

description: the **text** that characterizes the **representation**.

NOTE 8 This attribute is an upwardly compatible addition to **representation** as specified in ISO 10303-43:1994.

Formal propositions:

WR1: Each **representation** shall be the identified_item in at most one **id_attribute**.

NOTE 9 The **id_attribute** entity data type is defined in the **basic_attribute_schema** in ISO 10303-41.

WR2: Each **representation** shall be the named_item in at most one **name_attribute**.

NOTE 10 The **name_attribute** entity data type is defined in the **basic_attribute_schema** in ISO 10303-41.

NOTE 11 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E of ISO 10303-41.

4.4.10 representation_context

A **representation_context** is a context in which instances of **representation_item** are related.

NOTE 1 Two instances of **representation_context** are separate and have no relationship unless a relationship is explicitly specified between them in an annotated EXPRESS schema that uses or specializes this entity data type.

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: the **identifier** that distinguishes the **representation_context**.

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

NOTE 2 Constraints on the uniqueness of **context_identifier**, or the allowed values of **context_type,** can be specified in an annotated EXPRESS schema that uses or specializes this entity data type.

representations_in_context: the instances of **representation** that refer to the **representation_-context**.

4.4.11 representation_item

A **representation_item** is an element of representation. A **representation_item** participates in one or more instances of **representation**, or contributes to the definition of another **representation_item**.

NOTE 1 One **representation_item** contributes to the definition of a second **representation_item** if the first is referenced by the second.

NOTE 2 The same **representation_item** could be related multiple times to the same **representation_context** by being used directly or indirectly in several instances of **representation**, 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 the use of the same instance of **representation_item** in the **representation_context** for different uses.

EXAMPLE 1 Consider two instances of **representation**, 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 instances of **representation_item** in the **representation_context**. Rather, the use of the **line** in that **geometric_representation_context** has been asserted twice, once in each **representation**.

EXAMPLE 2 The **compound_representation_item** entity data type (see 4.4.1) provides a general capability to define one **representation_item** using other instances of **representation_item**. This can be used to create tabular structures: each cell in the table is a **representation_item** (such as a **measure_representation_item** (see ISO 10303-45 [5]) that provides a name-value-unit tuple), and the table is itself a **representation_item** that participates in the representation of a product property. The **representation_item** instances that are the cells in the table do not participate directly in the representation.

EXPRESS specification:

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

Attribute definitions:

name: the **label** by which the **representation_item** is known.

NOTE 3 The name could designate a particular instance of **representation_item** or the kind of information that an instance of **representation_item** conveys.

Formal propositions:

WR1: the **representation_item** shall participate in at least one **representation**, either as an item in that **representation** or being directly or indirectly referenced by an item in the **representation**.

4.4.12 representation_item_relationship

A **representation_item_relationship** is an association between two instances of **representation_item**. A **representation_item_relationship** can associate two instances of **representation_item** in the same **representation** or in two different instances of **representation**. In this association the two instances of **representation_item** are independent; neither is founded with respect to the other or the instance(s) of **representation** in which they participate. If the instances of **representation_item** that are associated are in different instances of **representation**, this association does not relate the instances of representation.

NOTE 1 The meaning of the association can be specified in an annotated EXPRESS schema that uses or specializes this entity, or in an agreement of common understanding between the partners sharing this information.

EXPRESS specification:

```
*)
ENTITY representation_item_relationship;
  name : label;
  description : OPTIONAL text;
  relating_representation_item : representation_item;
  related_representation_item : representation_item;
END_ENTITY;
(*
```

Attribute definitions:

name: the **label** by which the **representation_item_relationship** is known.

NOTE 2 The name could designate a particular instance of **representation_item_relationship** or the kind of information that an instance of **representation_item_relationship** conveys.

description: the **text** that characterizes the **representation_item_relationship**. The value of the **description** need not be specified.

relating_representation_item: one of the instances of **representation_item** that participates in the association.

NOTE 3 The role of this attribute can be defined in an annotated EXPRESS schema that uses or specializes this entity data type.

related_representation_item: the other instance of **representation_item** that participates in the association. If one element of the association is dependent upon the other, this attribute shall be the dependent one.

NOTE 4 The role of this attribute can be defined in an annotated EXPRESS schema that uses or specializes this entity data type.

4.4.13 representation_map

A **representation_map** is the identification of a **representation**, and of a **representation_item** that is an element in the **representation**, for the purpose of mapping. The **representation_item** defines the origin of the mapping.

NOTE An instance of **representation_map** is used as the source of a mapping by an instance of **mapped_item**. See 4.4.7 for an example of the use of these two entity data types.

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** from which the **mapped_representation** is to be mapped.

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

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

Formal propositions:

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

4.4.14 representation_relationship

A **representation_relationship** is the association of two instances of **representation**. One **representation** is not made part of the definition of the other by participation in a **representation_relationship**.

NOTE 1 The meaning of the association can be specified in an annotated EXPRESS schema that uses or specializes this entity, or in an agreement of common understanding between the partners sharing this information.

EXAMPLE Two instances of **representation** each describe a property of a product. The second of these instances is a more complete and accurate description of the property. An instance of **representation_relationship** can be used to denote the second instance of **representation** as being the successor of the first.

NOTE 2 A combination of instances of **representation** and **representation_relationship** can form a graph of related instances of **representation**. In such a graph, instances of **representation** are the nodes and instances of **representation_relationship** are the branches connecting the nodes.

NOTE 3 Constraints can be specified that ensure that such a graph is acyclic, i.e., that no instance of **representation** is its own ancestor. Such constraints are outside the scope of this part of ISO 10303.

EXPRESS specification:

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

Attribute definitions:

name: the **label** by which the **representation_relationship** is known.

NOTE 4 The name could designate a particular instance of **representation_relationship** or the kind of information that an instance of **representation_relationship** conveys.

description: the **text** that characterizes the **representation_relationship**. The value of the **description** need not be specified.

rep_1: the first of two **representation**s that are related.

rep_2: the second of two **representation**s that are related.

NOTE 5 There is no significance to the ordering of the two related instances of **representation**. The names **rep_1** and **rep_2** serve only to distinguish the attributes. If any significance to the ordering is needed in any specialization of **representation_relationship**, this significance can be defined in the specialization.

4.4.15 representation_relationship_with_transformation

A **representation_relationship_with_transformation** is an association between two **representation**s such that the contexts of the **representation**s are related through the transformation

NOTE 1 The existence of a **representation_relationship_with_transformation** instance does not necessarily imply that there is any correspondence between instances of **representation_item** in the two instances of **representation** that are related. Any such correspondence and other constraints can be defined in specializations of this entity data type.

EXAMPLE 1 Figure 3 illustrates the use of the **representation_relationship** entity data type. Three instances of representation are shown. The first representation R_1 contains some geometry G_1 and an **axis2_placement_3d** A_1. The second **representation** contains some geometry G_2 and an **axis2_placement_3d** A_2. For the purpose of this example, the nature and structure of G_1 and G_2 are not relevant. R_1 represents the shape of a roof. R_2 represents the shape of some walls.

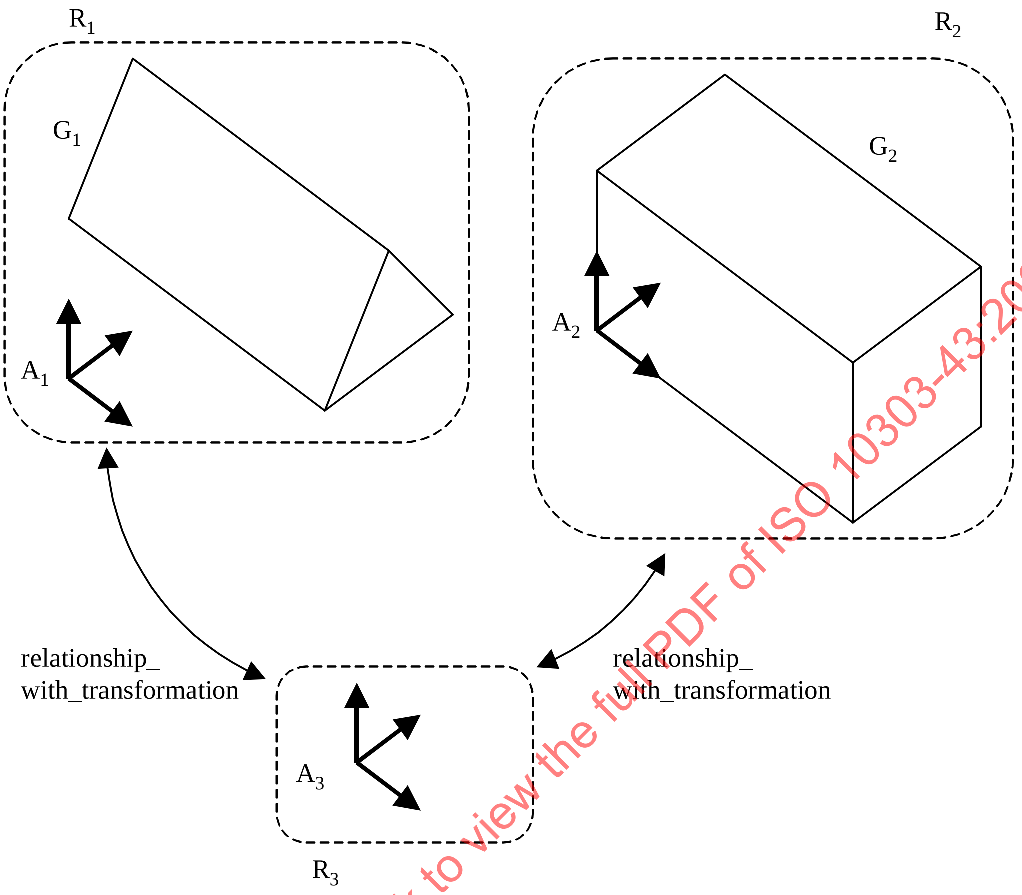

Figure 3 – Example of the use of representation_relationship_with_-transformation

Two instances of **representation_relationship_with_transformation** allow R_1 and R_2 to be associated with a third **representation**, R_3. R_3 represents the shape of the building. R_3 contains a single item: an **axis2_-placement_3d**. The associations between R_1 and R_3, and between R_2 and R_3, do not make R_1 and R_2 parts of R_3. However, the associations between R_1 and R_3, and between R_2 and R_3, allow an application to infer that G_1 and G_2 can be combined and used to describe the shape of the building. An application can make use of the specified transformations to compose the overall shape from G_1 and G_2.

See 4.4.7 for an example of the use of the **mapped_item** and **representation_map** entity data types to describe the composition of one **representation** from other instances of **representation**.

EXAMPLE 2 If the related representations both have geometric contexts, the **transformation** can be used to calculate the distances between instances of **geometric_representation_item** in the two instances of **representation**. The **geometric_representation_item** entity data type is defined in ISO 10303-42 [3,4].

EXAMPLE 3 Two instances of **representation** contain instances of **descriptive_representation_item**, an element of representation that specifies the use of a text string for representation. The first instance of **representation** contains descriptions in the English language; and the second contains descriptions in the French language. An instance of **representation_relationship_with_transformation** could be used to assert an association between these instances of representation, where the referenced **functionally_defined_-transformation** is a language translation table. The **descriptive_representation_item** entity data type is defined in ISO 10303-45 [5].

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 that is the range of the transformation.

SELF\representation_relationship.rep_2: the **representation** with a context that is the domain of the transformation.

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

NOTE 2 The transformation that relates the **representation.items** of one **representation** with the **representation.items** in a second **representation** can be specified by operating with the graphs of instances of **representation_item** that define the respective **representation.items**.

Formal propositions:

WR1: the two related **representations** shall not have the same **representation_context**.

Informal propositions:

IP1: if the transformation is an **item_defined_transformation**, the ordering of the instances of **representation** given for the inherited attributes of **representation_relationship** shall be consistent with the ordering of the two instances of **representation_item** given as attributes of **item_defined_transformation**.

NOTE 3 The **item_defined_transformation** entity data type is defined in 4.4.6.

4.4.16 uncertainty_assigned_representation

An **uncertainty_assigned_representation** is a type of **representation** that specifies uncertainty with respect to the elements of representation that it collects. The uncertainty is specified by instances of **uncertainty_measure_with_unit** (see 4.4.17), and applies to all elements of representation that are expressed in the same measures.

EXAMPLE This entity data type can be used to specify the uncertainties that apply to the numeric values used to represent a property. For example, in an application protocol whose domain is acoustical engineering, there could be an uncertainty of 1 decibel in the values that represent the relative loudness of sounds.

NOTE The precedence rules that apply to uncertainty in numeric quantities are specified in 4.2.6.

EXPRESS specification

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

Attribute definitions

uncertainty: the set of instances of **uncertainty_measure_with_unit** that apply to instances of **representation_item** in the **uncertainty_assigned_representation**.

4.4.17 uncertainty_measure_with_unit

An **uncertainty_measure_with_unit** is a **measure_with_unit** that specifies the uncertainty that applies to a type of measure. An **uncertainty_measure_with_unit** applies to each **representation_item** that uses the type of measure specified in the **value_component** of the **uncertainty_measure_with_unit**, in the following cases.

— the **representation_item** is part of an **uncertaintly_assigned_representation**;

— the **representation_item** is founded in a **global_unit_assigned_context**; or

— the **representation_item** is a **qualified_representation_item**.

NOTE The qualified_representation_item entity data type is defined in ISO 10303-45 [5].

EXPRESS specification:

```
*)
ENTITY uncertainty_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  name        : label;
  description : OPTIONAL text;
WHERE
  WR1: valid_measure_value (SELF\measure_with_unit.value_component);
END_ENTITY;
(*
```

Attribute definitions:

name: the **label** by which the **uncertainty_measure_with_unit** is known.

description: the **text** that characterizes the **uncertainty_measure_with_unit**. The value of the **description** need not be specified.

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.

4.4.18 value_representation_item

A **value_representation_item** is a type of **representation_item** that specifies only a measure value. The unit that applies to the specified value is provided by a **global_unit_assigned_context** of the **representation** containing the **value_representation_item**.

NOTE 1 The **global_unit_assigned_context** entity data type, subtype of **representation_context**, is defined in ISO 10303-41.

NOTE 2 The **measure_representation_item** entity data type, specified in ISO 10303-45 [5], is an element of representation that consists of a value and a unit.

EXPRESS specification:

```
*)
ENTITY value_representation_item
SUBTYPE OF (representation_item);
  value_component : measure_value;
WHERE
  WR1: SIZEOF (QUERY (rep <* using_representations (SELF) |
       NOT ('MEASURE_SCHEMA.GLOBAL_UNIT_ASSIGNED_CONTEXT'
       IN TYPEOF (rep.context_of_items)
       ))) = 0;
END_ENTITY;
(*
```

Attribute definitions:

value_component: the value of the element of representation when expressed with in the unit specified in the **global_unit_assigned_context**.

Formal propositions:

WR1: every **value_representation_item** shall be an item in a **representation** whose **context_of_-items** attribute references an instance of **representation_context** that is a **global_unit_assigned_-context**, or shall participate in the definition of such a **representation_item**.

Informal propositions:

IP1: If a **value_representation_item** is an item in more than one **representation**, each **representation** shall specify the same unit, either by referencing the same **global_unit_assigned_-context**, or by referencing different instances of **global_unit_assigned_context** that specify the same unit.

4.5 Representation function definitions

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 checks both the **mapped_representation** and the **mapped_representation.items** recursively for any instances of **mapped_item**, or **representation_item** 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**.

NOTE This function is used to constrain the entity **mapped_item**. See 4.4.7.

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;
  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_representation
      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 instances of **representation** in which the **mapped_item** is used. This is input to the function. On initial input, this is the set of instances of **representation** in which the **mapped_item** being checked is used.

children_set: the set of instances of **representation_item** 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.

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. 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:

— referenced in the set of **items** of a **representation** where **cntxt** appears as the **context_of_items**, or

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

NOTE 1 The second 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 instances of **representation_item**. The tree is rooted in an entity that is related to a **representation_context** by fulfilling the first or second condition.

NOTE 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
    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 bag is not empty
      IF SIZEOF(y) > 0 THEN
      -- For each element in the bag
      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;
(*
```

Argument definitions:

item: the **representation_item** checked for relationship in **cntxt**. This is input to the function.

cntxt: the **representation_context** for which relationship to **item** is determined. This is input to the function.

4.5.3 using_items

The function **using_items** returns the set of instances of **representation_item** or **founded_item** that reference a **representation_item** directly or indirectly.

EXPRESS specification:

```
*)
FUNCTION using_items (item : founded_item_select;
                      checked_items: SET OF founded_item_select)
                    : SET OF founded_item_select;
  LOCAL
    new_check_items    : SET OF founded_item_select;
    result_items       : SET OF founded_item_select;
    next_items         : SET OF founded_item_select;
  END_LOCAL;
  result_items := [];
  new_check_items := checked_items + item;
  -- Find the set of representation_items or founded_items
  -- in which item is used directly.
  next_items := QUERY(z <* bag_to_set( USEDIN(item , '')) |
    ('REPRESENTATION_SCHEMA.REPRESENTATION_ITEM' IN TYPEOF(z)) OR
    ('REPRESENTATION_SCHEMA.FOUNDED_ITEM'        IN TYPEOF(z)));
  -- If the set of next_items is not empty;
  IF SIZEOF(next_items) > 0 THEN
    -- For each element in the set, find the using_items recursively
    REPEAT i := 1 TO HIINDEX(next_items);
      -- Check for loop in data model, i.e. one of the next_items
      -- occurred earlier in the set of check_items;
      IF NOT(next_items[i] IN new_check_items) THEN
        result_items := result_items + next_items[i] +
                        using_items(next_items[i],new_check_items);
      END_IF;
    END_REPEAT;
  END_IF;
  -- return the set of representation_items or founded_items
  -- in which the input item is used directly and indirectly.
  RETURN (result_items);
END_FUNCTION;
(*
```

Argument definitions:

item: the **representation_item** for which the referencing instances of **representation_item** and **founded_item** are determined. This is input to the function.

checked_items: the set of instances of **representation_item** and **founded_item** that have been checked already in order to ensure termination of the recursive function. This is input to the function.

4.5.4 using_representations

The function **using_representations** returns the set of representations in which a **representation_-item** is used.

A **representation_item** is used in a **representation** if it is:

— referenced in the set of items of the representation,

— referenced by a **representation_item** used in the **representation**, or

— referenced by a **founded_item** used in the **representation**.

NOTE The second and third conditions are checks allowing for a **representation_item** to be used in a **representation** by being part of a tree of related **representation_items** or **founded_items**. The tree is rooted in an entity used in a **representation** by fulfilling the first condition.

A **founded_item** or a **representation_item** is used in a **representation** if it is referenced directly, or indirectly, by a **representation_item** in the set of items of the **representation**.

EXPRESS specification:

```
*)
FUNCTION using_representations (item : founded_item_select)
  : SET OF representation;
  LOCAL
    results            : SET OF representation;
    result_bag         : BAG OF representation;
    intermediate_items : SET OF founded_item_select;
  END_LOCAL;
  -- Find the representations in which the item is used and add to the
  -- results set.
  results := [];
  result_bag :=
USEDIN(item,'REPRESENTATION_SCHEMA.REPRESENTATION.ITEMS');
  IF SIZEOF(result_bag) > 0 THEN
    REPEAT i := 1 TO HIINDEX(result_bag);
      results := results + result_bag[i];
    END_REPEAT;
  END_IF;
  -- Find all representation_items or founded_items
  -- by which item is referenced directly or indirectly.
  intermediate_items := using_items(item,[]);
  -- If the set of intermediate items is not empty;
  IF SIZEOF(intermediate_items) > 0 THEN
    -- For each element in the set, add the
    -- representations of that element.
    REPEAT i := 1 TO HIINDEX(intermediate_items);
      result_bag := USEDIN(intermediate_items[i],
                    'REPRESENTATION_SCHEMA.REPRESENTATION.ITEMS');
```

```
   IF SIZEOF(result_bag) > 0 THEN
    REPEAT j := 1 TO HIINDEX(result_bag);
     results := results + result_bag[j];
    END_REPEAT;
   END_IF;
  END_REPEAT;
 END_IF;
 -- Return the set of representation in which the input item is
 -- used directly and indirectly (through intervening
 -- representation_items or founded items).
 RETURN (results);
END_FUNCTION;
(*
```

Argument definitions:

item: the **representation_item** or **founded_item** for which using instances of **representation** are determined. This is input to the function.

4.5.5 valid_measure_value

The function **valid_measure_value** determines whether a **measure_value** is valid. The function returns TRUE if the **measure_value** is numeric and is positive, or if it is textual. Function **valid_measure_value** returns FALSE otherwise.

EXPRESS specification:

```
*)
FUNCTION valid_measure_value
 (m : measure_value) : BOOLEAN;
 IF ('REAL' IN TYPEOF (m)) THEN
 RETURN (m > 0.0);
 ELSE
  IF ('INTEGER' IN TYPEOF (m)) THEN
  RETURN (m > 0);
  ELSE
   RETURN (TRUE);
  END_IF;
 END_IF;
END_FUNCTION;
(*
```

Argument definitions:

m: the **measure_value** to be checked. This is the input to the function.

```
*)
END_SCHEMA; -- representation_schema
(*
```

Annex A
(normative)

Short names of entities

Table A.1 provides the short names of entities specified in this part of ISO 10303. Requirements on the use of the short names are found in the implementation methods included in ISO 10303.

NOTE The short names are available from the Internet – see Annex C.

Table A.1 – Short names of entities

Entity names	Short names
compound_representation_item	CMRPIT
definitional_representation	DFNRPR
founded_item	FNDITM
functionally_defined_transformation	FNDFTR
global_uncertainty_assigned_context	GC
item_defined_transformation	ITDFTR
mapped_item	MPPITM
parametric_representation_context	PRRPCN
representation	RPRSNT
representation_context	RPRCNT
representation_item	RPRITM
representation_item_relationship	RPITRL
representation_map	RPRMP
representation_relationship	RPRRLT
representation_relationship_with_transformation	RRWT
uncertainty_assigned_representation	UNASRP
uncertainty_measure_with_unit	UMWU
value_representation_item	VLRPIT

Annex B
(normative)

Information object registration

B.1 Document identification

To provide for unambiguous identification of an information object in an open system, the object identifier

{ iso standard 10303 part(43) version(4) }

is assigned to this part of ISO 10303. The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

B.2 Schema identification

To provide for unambiguous identification of the **representation_schema** in an open information system, the object identifier

{ iso standard 10303 part(43) version(3) object(1) representation-schema(1) }

is assigned to the **representation_schema** (see clause 4). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.