INTERNATIONAL STANDARD ISO 10303-108:2005
TECHNICAL CORRIGENDUM 1

Published 2008-12-15

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

Part 108:
Integrated application resource: Parameterization and constraints for explicit geometric product models

TECHNICAL CORRIGENDUM 1

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

Partie 108: Ressources d'application intégrées: Paramétrage et contraintes pour les modèles de produits géométriques explicites

RECTIFICATIF TECHNIQUE 1

iTeh STANDARD PREVIEW

(standards.iteh.ai)

ISO/DIS 10303-239
https://standards.iteh.ai/catalog/standards/sist/a06fe7b8-ae27-4110-b740-77aa6139f058/iso-
dis-10303-239

Technical Corrigendum 1 to ISO 10303-108:2005 was prepared by Technical Committee ISO/TC 184, *Automation systems and integration*, Subcommittee SC 4, *Industrial data.*

Introduction

The modifications made to ISO 10303-108:2005 have three purposes:

a) *to remove an entity name clash with ISO 10303-210:2001 (published earlier than ISO 10303-108:2005 and therefore having prior claim on the name) concerning* **model_parameter**. *This Technical Corrigendum provides for its replacement throughout ISO 10303-108 with* **variational_parameter***;*

b) *to remove the definition of* **non_negative_length_measure***, which has been moved to ISO 10303-41, and to replace it by a reference to that resource;*

c) *to correct minor errors in EXPRESS code.*

The opportunity has also been taken to update the normative reference to ISO 10303-55 (now published) and to correct a few minor editorial errors mainly concerning the numbering of notes and examples.

ICS 25.040.40   **Ref. No. ISO 10303-108:2005/Cor.1:2008(E)**

Published in Switzerland

Modifications to the text of ISO 10303-108:2005

Table of Contents, p. iii ff.
The term **model_parameter** *is being systematically replaced by* **variational_parameter**. *The necessary entity name change requires replacement of the titles of several subclauses.*
Make the following replacements:

Subclause:	Previous title:	New title:
4.2.1	Model parameters	Variational parameters
4.4.1	model_parameter	variational_parameter
4.4.2	bound_model_parameter	bound_variational_parameter
4.4.3	unbound_model_parameter	unbound_variational_parameter
4.4.7	unbound_model_parameter_semantics	unbound_variational_parameter_semantics
5.2.5	Roles of model parameters...	Roles of variational parameters...

Delete the entry 7.3.10 non_negative_length_measure *from the Table of Contents*

Clause 1, p. 1
Replace model parameters *by* variational parameters *in line 1 of the first paragraph, and lines 1 and 3 of the second paragraph.*

Clause 1.1, p. 3
Replace model parameters *by* variational parameters *in the third bulleted item of this subclause.*

Clause 2, p. 5
ISO 10303-55 has now been published. After ISO 10303-55 *delete :—, and also delete the footnote.*

Clause 3.7.24, p. 11
Replace **model parameter** *by* **variational parameter**, *and reposition this definition in the list to follow the definition of* **variational** *(previously clause 3.7.34). The wording of the definition is unchanged, but the text of the notes and the example needs to be changed.*
Replace NOTE 1, EXAMPLE and NOTE 2 as follows:

NOTE 1   Assignment of different values to variational parameters generates different members of a family of models. Variational parameters therefore express design freedom in a model, according to the parameterization scheme imposed by its creator. Limitations may be defined on the allowable ranges of variational parameters.

EXAMPLE   The dimensions of a generic block may be represented by variational parameters L (length), W (width) and H (height). Individual members of the family of blocks are specified by assigning numerical values to the three parameters independently. Alternatively, relationships may be defined between the variational parameters, such as $L = 2W, H = 0.5W$, to restrict the size of the family and define it in terms of the single independent variational parameter W.

NOTE 2   Distinction must be made between the use of the word *parameter* in this part of ISO 10303, in ISO 10303-11, in ISO 10303-42 and in ISO 10303-50. In ISO 10303-11 a parameter is used for the formal representation of an input to, or output from, a function or procedure defined in an EXPRESS schema. In ISO 10303-42 a parameter is a variable used to identify the position of a point on a curve or a surface, so that the parameter may be thought of as an input to a function whose output is a coordinate value. In ISO 10303-50 a parameter is defined as 'a free variable in an expression'. In this part of ISO 10303 the term **variational parameter** is used for a variable that controls dimensions or other gross characteristics of a model, for example the overall shape of a product model. A variational parameter may be thought of as an input to a procedure, in this case a procedure that

computes one instance of a family of shape models. It is unfortunate that the word 'parameter' is in widespread current use for such a variety of purposes. Although at a broad conceptual level the usages within ISO 10303 are similar, there are significant differences in such matters as the way the functions or procedures are defined and in the scope of parameters in a model.

Clause 3.7 more generally

The terms defined in this clause need to be reordered into alphabetical order, and other references to the superseded entity name need to be changed as follows: (a) Renumber clause 3.7.24, as modified above, to 3.7.34; (b) Renumber the current clauses 3.7.25 – 3.7.34 as 3.7.24 – 3.7.33, keeping their sequence the same; (c) In the clause newly numbered 3.7.25, defining the term **parameter**, *replace the existing definition by* variational parameter (in the context of this part of ISO 10303) — see the definition of **variational parameter** given in clause 3.7.34.

Clause 4, p. 15

Clause 4.2 and most of clause 4.4 need to be replaced. Subclauses 4.1 and 4.3 may remain as they are (subject to the correction noted below), and so may subclause 4.5. Clause 4.4.9 needs only one name replacement.

Clause 4.1, p. 15

In NOTE 2*, replace* Figure *by* Figures.

Clause 4.2, p. 15

Replace subclause 4.2 by the following:

4.2 Fundamental concepts and assumptions

This schema provides representation methods for the following:

— Variables, represented by instances of **bound_variational_parameter** or **unbound_variational_-parameter**, expressing variation or design freedom in a representation or model;

— A means for binding a **bound_variational_parameter** instance to an attribute of another entity data type instance in the same **representation**;

— Domains of validity for instances of **bound_variational_parameter** and **unbound_variational_-parameter**;

— A means for fixing the values of attributes of specific entity data type instances in a model, equivalent to the use of **bound_variational_parameter** instances with constant associated values.

These resource constructs are of general utility in the exchange and sharing of ISO 10303 models embodying

— the capability for variation of attribute values in a model following an exchange;

— the capture and transfer of constraint relationships defined in terms of mathematical expressions, functions or procedures. Specifically, variational parameters can participate in instances of **free_-form_constraint** as defined in clause 5.4.4.

Clause 6.3.1 of this part of ISO 10303 defines **variational_representation_item** as a subtype of the ISO 10303-43 entity data type **representation_item**. Variational parameters are defined as subtypes of

variational_representation_item, which is the supertype of all entity data types used to express the variational aspects of models with explicit parameterization and constraints. The type of **representation** in which they participate is a **variational_representation**, as defined in clause 6.3.3.

4.2.1 Variational parameters

An abstract entity data type **variational_parameter** is provided, with two instantiable subtypes, **bound_- variational_parameter** and **unbound_variational_parameter**. These allow for the capture and transmission of permitted aspects of model variation that can be exploited in a receiving system. A **bound_- variational_parameter** is bound to an attribute of an entity data type instance in an ISO 10303 model, in which case it provides a syntactic representation of the value of that attribute, for example a dimensional value. By contrast, an **unbound_variational_parameter** is not directly associated with any model attribute. Either kind of **variational_parameter** may be used in mathematical relationships defined in free-form constraints. The current value of a **variational_parameter** is specified by one of its attributes; in the bound case the value of this attribute is required by an informal proposition to be the same as the value of the attribute to which it is bound.

The entity data type **variational_parameter** is defined as a subtype of **variational_representation_- item**, and the scope of its instantiable subtypes is therefore defined by those instances of **variational_- representation** in which they participate. It is also a subtype of the ISO 10303-50 entity data type **maths_variable**, from which it inherits an attribute **values_space**, of ISO 10303-50 type **maths_space**. This attribute specifies the domain of validity for values of the **variational_parameter**. These may include domains corresponding to those of the EXPRESS data types REAL, INTEGER, BOOLEAN and STRING, together with various bounded subsets of the REAL and INTEGER domains. This part of ISO 10303 does not directly provide the use of parameters having values belonging to aggregate types, but applications may define such extensions if they are required.

EXAMPLE 1   Consider a rectangle, with length x units and width y units. Here x and y are variables or parameters. An explicit constraint relationship $x = y^2 + 2$ relates these dimensions. Valid parameter ranges $10.0 \leq x \leq 30.0$ and $2.0 \leq y \leq 5.0$ are defined. In this case the two variables correspond to instances of **bound_variational_parameter**, both bound to physical quantities in the model, i.e., dimensional attributes of the rectangle. The parameterization and constraint information may be transmitted together with a 'current result' — an explicit model of a rectangle with length 18.0 units and width 4.0 units. These parameter values satisfy the constraint and fall within the required parameter ranges. When model transfer is complete, if one of the parameters is edited the other should adjust accordingly to maintain satisfaction of the constraint, provided the parameters remain within their valid ranges. It is assumed that the necessary functionality for parameter variation and constraint maintenance will be provided by the receiving system.

The following example illustrates the use of an **unbound_variational_parameter**.

EXAMPLE 2   Suppose an instance of **right_circular_cylinder** (as defined in ISO 10303-42), has associated instances of **bound_variational_parameter** associated with its **radius** and **height** attributes, here denoted by r and h respectively. A third parameter, denoted by t, may be used to control the values of both r and h according to the relationships $r = 3t - 2$, $h = t^2 + 1$. In the case when t is not bound to an attribute of any entity data type instance, it will appropriately be modelled in terms of an **unbound_variational_parameter**.

4.2.2 Parameter binding to an instance attribute

A **bound_variational_parameter** is associated with an attribute of an entity data type instance in a populated schema, whose value represents the value of the parameter. This association is defined through the use of an entity data type **instance_attribute_reference** that simply specifies the name of an attribute

and the instance to which it belongs (see clause 4.4.6). A simple example is given below to illustrate the principle, and the intended usage of the mechanism is more fully documented in clause F.1 of annex F. Once the parameter binding has been established, the parameter may participate in a relationship that governs its value if the model is subsequently edited in a receiving system.

EXAMPLE   For the purpose of the example, entity data types defined in the ISO 10303 integrated generic resources are treated as though they are instantiable elements in an application protocol.

It is desired to parameterize one dimension of a **block** solid, as defined in ISO 10303-42. This has three attributes, x, y and z, that prescribe its three principal dimensions. In any instantiation of the block these will have specific real numerical values. Consider now the following fragment of an ISO 10303-21 transfer file:

```
#290 = AXIS2_PLACEMENT_3D(...);
#300 = BLOCK('BLOCK1', #290, 4.0, 6.0, 8.0);
#310 = INSTANCE_ATTRIBUTE_REFERENCE
        ('GEOMETRIC_MODEL_SCHEMA.BLOCK.X', #300);
#320 = FINITE_REAL_INTERVAL(2.0, .CLOSED., 10.0, .CLOSED.);
#330 = BOUND_VARIATIONAL_PARAMETER
        ('XPARAM', #320, 'XPARAM', 'BLOCK X-DIMENSION', *);
#340 = BOUND_PARAMETER_ENVIRONMENT(#310, #330);
```

The instances represented above are explained as follows:

`#290:` defines an ISO 10303-42 axis placement (details omitted) for the next instance;

`#300:` the **block** instance. As a subtype of ISO 10303-43 **representation_item**, this inherits a **name** attribute of type **label**, whose value in this instance is `'block1'`. The block is defined with respect to the axis placement `#290` and has dimensions 4.0, 6.0 and 8.0 units;

`#310:` an instance of **instance_attribute_reference**; `'geometric_model_schema.block.x'` is the specified attribute name and the referenced **block** instance is `#300`. Note that the attribute name appears fully qualified with the name of the owning entity data type and its defining schema. This entry in the file identifies the particular instance whose specified attribute is to be associated with the **bound_variational_parameter** instance;

`#320:` defines the domain of that parameter, a real interval closed at both ends, bounded below by 2.0 and above by 10.0. The entity data type **finite_real_interval** is defined in ISO 10303-50;

`#330:` specifies the **bound_variational_parameter** itself, as defined in clause 4.4.2 of this schema. Its attribute value list contains these entries:

— a label, `'xparam'`, corresponding to the **name** attribute of its **representation_item** supertype;

— a domain `#320`, corresponding to the **values_space** attribute of its **maths_variable** supertype;

— a label, `'xparam'`, corresponding to the **name** attribute of its **maths_variable** supertype — the two inherited **name** attributes are required by a WHERE rule to have the same values;

— a textual description `'block x-dimension'`;

— the value of the **variational_parameter**, given as a derived value, although no formal method is available in EXPRESS for deriving it from instance `#300`;

`#340:` an instance of **bound_parameter_environment**, defined in clause 4.4.4, providing the link between the specified instance attribute, `#310`, and the parameter bound to it, `#330`.

At this point the binding of the parameter to the desired attribute is complete. The intention is that the value attribute of the **bound_variational_parameter** instance `#330` is equal to the value 4.0 associated with the `'x'` attribute in the **block** instance, and lies within the domain of validity represented by `#320`. However, because the

EXPRESS language provides no formal way of asserting this the value attribute of the parameter is recorded as indeterminate, and an informal proposition in clause 4.4.2 requires that the two values shall be equal on completion of the transfer. The achievement of this is the responsibility of the translation software. The parametric relationship having been captured in an exchange file as shown above, the x-dimension of the block may now be controlled in terms of the parameter associated with it if the model is edited following transfer into a receiving system.

Clause 4.4, p. 19
Replace subclause 4.4 up to and including 4.4.8 by the following:

4.4 Parameterization entity definitions

4.4.1 variational_parameter

The **variational_parameter** entity data type is a type of **variational_representation_item** that represents a variable quantity in a **variational_representation** (see clause 6). It is also a type of **maths_variable** as defined in ISO 10303-50, and can therefore participate in mathematical relationships. Its attributes include an optional textual description of the significance of the parameter, and a current parameter value. A **variational_parameter** instance inherits a **name** attribute from both its supertypes; for consistency, the **maths_variable** name is required to be the same as the **representation_item** name. It also inherits an attribute **values_space** from its **maths_variable** supertype, specifying the domain (permissible set of values) of the **variational_parameter**.

NOTE 1   The fact that **variational_parameter** is a type of **maths_variable** restricts its underlying domain of values to subsets of the real or integer numbers, Booleans or strings. Future editions of this part of ISO 10303 have the possibility of extending that spectrum of domains, if applications require it, by making use of the more general capabilities of ISO 10303-50.

Because **variational_parameter** is a type of **maths_variable**, and hence ultimately of **generic_variable** as defined in ISO 13584-20, each instance of it is required to have an associated instance of the ISO 13584-20 entity data type **environment**, which links the parameter with its associated semantics. For that purpose, this schema provides appropriate subtypes of **environment**, namely **bound_parameter_environment** and **unbound_parameter_environment**, as defined in clauses 4.4.4 and 4.4.5 respectively. The key relationships between these ISO 10303-108 and ISO 13584-20 entity data types are shown in clause F.1 of annex F.

EXPRESS specification:

```
*)
ENTITY variational_parameter
  ABSTRACT SUPERTYPE OF (ONEOF (bound_variational_parameter,
                                unbound_variational_parameter))
  SUBTYPE OF (variational_representation_item, maths_variable);
  parameter_description : OPTIONAL text;
  parameter_current_value : maths_value;
WHERE
  WR1 : member_of(parameter_current_value,
    SELF\maths_variable.values_space);
  WR2 : SELF\maths_variable.name = SELF\representation_item.name;
END_ENTITY;
(*
```

Attribute definitions:

parameter_description: An optional description, for human interpretation, of the significance of the **variational_parameter** instance.

parameter_current_value: The current value associated with the **variational_parameter** instance.

SELF\maths_variable.values_space: The domain of validity of the current value associated with the **variational_parameter** instance.

SELF\maths_variable.name: The **name** attribute of the **maths_variable** supertype.

SELF\representation_item.name: The **name** attribute of the **representation_item** supertype, whose value is required to be the same as for the previous attribute.

Formal propositions:

WR1: The current value of the **variational_parameter** instance shall lie within the domain specified by the attribute **SELF\maths_variable.values_space**.

WR2: The **name** attributes of supertypes **maths_variable** and **representation_item** shall be the same.

NOTE 2 No requirement has been imposed for the **name** attribute of a **variational_parameter** instance to have a value that is unique in any **representation** it participates in. This is because it is not anticipated that **variational_parameter** instances will be referred to by their name attributes. In general, **variational_parameter** instances used in an exchange are referenced by ephemeral identifiers created during the translation process and discarded when the exchange is complete, by which time system-dependent identifiers have been generated in the receiving system. However, if uniqueness of name attributes is required for some application purpose the necessary restriction can be imposed in schemas that specialize definitions from this part of ISO 10303.

4.4.2 bound_variational_parameter

The **bound_variational_parameter** entity data type is a type of **variational_parameter** whose instances can be bound to (associated with) explicit attributes of entity instances participating in a **variational_representation**. The current value of any instance of **bound_variational_parameter** is indeterminate during an exchange, but is required by an informal proposition to be set equal in the receiving system to the value of the attribute to which it is bound. That attribute is therefore required to have an explicit value in a populated schema, which rules out the association of a **bound_variational_parameter** with a derived or inverse attribute.

NOTE 1 This approach to the association of a value with a **bound_variational_parameter** instance is necessary because the EXPRESS language provides no means for the formal derivation of the value from the referenced entity data type instance in a populated schema.

EXPRESS specification:

```
*)
ENTITY bound_variational_parameter
  SUBTYPE OF (variational_parameter);
DERIVE
  SELF\variational_parameter.parameter_current_value : maths_value := ?;
WHERE
```

```
WR1 : 'PARAMETERIZATION_SCHEMA.BOUND_PARAMETER_ENVIRONMENT'
    IN TYPEOF(SELF\generic_variable.interpretation);
END_ENTITY;
(*
```

Attribute definitions:

parameter_current_value: The current value of the attribute to which the parameter is bound, always derived as indeterminate for an instance of this entity data type (see notes 1, 2 and 3).

SELF\generic_variable.interpretation: The instance of **bound_parameter_environment** that links a **bound_variational_parameter** instance to a particular entity instance attribute.

Formal propositions:

WR1: Every instance of **bound_variational_parameter** shall reference an instance of type **bound_-parameter_environment**.

NOTE 2   The indeterminate value of the the attribute **parameter_current_value** does not give rise to a violation of WR1 of the **variational_parameter** supertype, which for an instance of **bound_variational_parameter** will evaluate to UNKNOWN rather than FALSE. Clause 9.2.2.2 of ISO 10303-11 states that this does not constitute a violation of the rule. In practice it is the responsibility of the translator software to check that the value of the referenced attribute lies within the domain of the parameter.

NOTE 3   Because the indeterminate value of **parameter_current_value** is derived — in this case, by a simple assigment — an instance of **bound_variational_parameter** in an ISO 10303-21 exchange file will represent it by an asterisk, ∗, as illustrated in the examples in clause 4.2.2 and annex F.

NOTE 4   The ISO 13584-20 entity data type **environment** has two attributes, **semantics** and **syntactic_repre-sentation**. As a subtype of **environment**, **bound_parameter_environment** also possesses these attributes, which are treated as follows:

> **semantics:** This attribute is of type **instance_attribute_reference** (see clause 4.4.6).

> **syntactic_representation:** A WHERE rule applying to **bound_parameter_environment** requires that the value of this attribute shall be of type **bound_variational_parameter**.

The ISO 13584-20 entity data type **generic_variable** has an inverse attribute **interpretation**, corresponding to the **syntactic_representation** attribute of **environment**. For **bound_variational_parameter**, a subtype of **generic_-variable**, this inverse attribute is required by WR1 to be of type **bound_parameter_environment**.

The entity data types **environment** and **variable_semantics** are subtyped in this part of ISO 10303 to satisfy a requirement of ISO 13584-20 regarding the binding of values to variables. An EXPRESS-G representation of their relationships with entity data types defined in this schema is given in clause F.1 of annex F.

Informal propositions:

IP1: The **parameter_current_value** attribute shall have the same value as the entity data type instance attribute referenced via the inverse attribute **SELF\generic_variable.interpretation**, and shall be type-compatible with it.

NOTE 5   It will be crucial for implementations to ensure that the foregoing informal proposition is satisfied. Means cannot be provided in this schema for checking its validity because no formal mechanism exists for accessing the value of the attribute with which the **bound_variational_parameter** is associated.

NOTE 6   A local rule in the definition of **variational_representation** (see clause 6.3.3) ensures that any instance of **bound_variational_parameter** shall belong to the same **variational_representation** as the entity data type instance to whose specified attribute it is bound.

NOTE 7   No restriction is imposed in this schema to prevent the binding of more than one **bound_variational_parameter** to a single attribute of an entity data type instance. However, if such a restriction is required for some application purpose it can be specified in schemas that specialize definitions from this part of ISO 10303.

NOTE 8   The mechanism defined in this schema does not allow the direct association of a **bound_variational_parameter** instance with more than one entity data type instance attribute. The effect of such a multiple binding can be achieved through the use of multiple **bound_variational_parameter** instances related by the **equal_parameter_constraint** as defined in clause 5.4.3.

4.4.3 unbound_variational_parameter

The **unbound_variational_parameter** entity data type is a type of **variational_parameter** representing a variable that is not bound to an attribute of any entity instance in the model. The value attribute of an **unbound_variational_parameter** instance is specified explicitly, rather than by association with an attribute of some other instance in the model.

NOTE 1   An instance of **unbound_variational_parameter** may be used in mathematical expressions in free-form constraints that govern values associated with instances of **bound_variational_parameter**. Examples of this usage are given in clause 4.2.1 and clause F.1 of annex F.

EXPRESS specification:

```
*)
ENTITY unbound_variational_parameter
  SUBTYPE OF (variational_parameter);
WHERE
  WR1: 'PARAMETERIZATION_SCHEMA.UNBOUND_PARAMETER_ENVIRONMENT'
    IN TYPEOF(SELF\generic_variable.interpretation);
END_ENTITY;
(*
```

Attribute definitions:

SELF\generic_variable.interpretation: The instance of **unbound_parameter_environment** providing the link between the **unbound_variational_parameter** instance and its associated instance of **unbound_variational_parameter_semantics**. The definitions of these entities are given in clauses 4.4.5 and 4.4.7.

Formal propositions:

WR1: Every instance of **unbound_variational_parameter** shall be referenced by an instance of **unbound_parameter_environment**.

NOTE 2   The ISO 13584-20 entity data type **environment** has two attributes, **semantics** and **syntactic_representation**. As a subtype of **environment**, **unbound_parameter_environment** also possesses these attributes, which are treated as follows:

> **semantics:** The value of this attribute is required to be of type **unbound_variational_parameter_semantics** (see clause 4.4.7).

> **syntactic_representation:** A WHERE rule applying to **unbound_parameter_environment** requires that the value of this attribute shall be of type **unbound_variational_parameter**.

The ISO 13584-20 entity data type **generic_variable** has an inverse attribute **interpretation**, corresponding to the **syntactic_representation** attribute of **environment**. For **unbound_variational_parameter**, a subtype of **generic_variable**, this inverse attribute is required by WR1 to be of type **unbound_parameter_environment**.

The entity data types **environment** and **variable_semantics** are subtyped in this part of ISO 10303 to satisfy a requirement of ISO 13584-20 regarding the binding of values to variables. An EXPRESS-G representation of their relationships with entity data types defined in this schema is given in clause F.1 of annex F.

4.4.4 bound_parameter_environment

The **bound_parameter_environment** entity data type is a type of **environment** as defined in ISO 13584-20. It provides a link between the syntactic and semantic aspects of a **bound_variational_parameter** instance.

NOTE 1   ISO 13584-20 requires an instance of **environment** to be defined for every instance of **generic_variable**, and since **bound_variational_parameter** as defined in this schema is a subtype of **generic_variable** it has been necessary to provide an appropriate subtype of **environment** in this schema.

EXPRESS specification:

```
*)
ENTITY bound_parameter_environment
  SUBTYPE OF (environment);
WHERE
  WR1: ('PARAMETERIZATION_SCHEMA.BOUND_VARIATIONAL_PARAMETER' IN
    TYPEOF(SELF\environment.syntactic_representation)) AND
    ('PARAMETERIZATION_SCHEMA.INSTANCE_ATTRIBUTE_REFERENCE' IN
    TYPEOF(SELF\environment.semantics));
END_ENTITY;
(*
```

Formal propositions:

WR1: For every instance of **bound_parameter_environment**, the **syntactic_representation** attribute of the **environment** supertype shall be of type **bound_variational_parameter** and the **semantics** attribute shall be of type **instance_attribute_reference**.

NOTE 2   The relationships between the ISO 13584-20 entity data type **environment** and other entity data types defined in this schema are illustrated by the EXPRESS-G diagram in clause F.1 of annex F.

4.4.5 unbound_parameter_environment

The **unbound_parameter_environment** entity data type is a type of **environment** as defined in ISO 13584-20. It provides a link between the syntactic and semantic aspects of an **unbound_variational_- parameter** instance.

NOTE 1   ISO 13584-20 requires an instance of **environment** to be defined for every instance of **generic_- variable**, and since **unbound_variational_parameter** as defined in this schema is a subtype of **generic_variable** it has been necessary to provide an appropriate subtype of **environment** in this schema.

EXPRESS specification:

```
*)
ENTITY unbound_parameter_environment
  SUBTYPE OF (environment);
WHERE
  WR1: ('PARAMETERIZATION_SCHEMA.UNBOUND_VARIATIONAL_PARAMETER' IN
    TYPEOF(SELF\environment.syntactic_representation)) AND
    ('PARAMETERIZATION_SCHEMA.UNBOUND_VARIATIONAL_PARAMETER_SEMANTICS' IN
    TYPEOF(SELF\environment.semantics));
END_ENTITY;
(*
```

Formal propositions:

WR1: For any instance of **unbound_parameter_environment**, the **syntactic_representation** attribute of the **environment** supertype shall be of type **unbound_variational_parameter** and the **semantics** attribute shall be of type **unbound_variational_parameter_semantics**.

NOTE 2   The relationships between the ISO 13584-20 entity data type **environment** and other entity data types defined in this schema are illustrated by the EXPRESS-G diagram in clause F.1 of annex F.

4.4.6 instance_attribute_reference

The **instance_attribute_reference** entity data type is a type of **variable_semantics** (see ISO 13584-20). It identifies a named explicit attribute of a specific **representation_item** instance in a populated EXPRESS schema. The name of the attribute is specified as an **attribute_identifier** (see clause 4.3.1). Derived or inverse attributes shall not be referenced by the use of **instance_attribute_reference**.

NOTE 1   This entity data type is used in the definition of an association between a bound **variational_parameter** and an attribute of an instance of a **representation_item**. It is defined as a subtype of the ISO 13584-20 entity data type **variable_semantics** to satisfy a requirement of that standard regarding the binding of values to variables. The intention is to provide an interpretation of the role of the variable in its modelling context.

Examples of the use of this entity data type are provided in clause F.1 of annex F.

EXPRESS specification:

```
*)
ENTITY instance_attribute_reference
  SUBTYPE OF (variable_semantics);
  attribute_name  : attribute_identifier;
  owning_instance : representation_item;
END_ENTITY;
(*
```

Attribute definitions:

attribute_name: An **attribute_identifier** specifying the name of the referenced attribute.

owning_instance: The **represention_item** instance owning the referenced attribute.

Informal propositions:

IP1: Any attribute referenced by an instance of **instance_attribute_reference** shall be specified as a fully qualified attribute in the form `'SCHEMA_NAME.ENTITY_NAME.ATTRIBUTE_NAME'` (see ISO 10303-11).

NOTE 2   The foregoing informal proposition is imposed to ensure that ambiguities are avoided, for example in the case of complex instances, and to enable effective checking for compatibility of the referenced attribute identifier and its owning entity data type instance.

4.4.7 unbound_variational_parameter_semantics

The **unbound_variational_parameter_semantics** entity data type is a type of **variable_semantics** as defined in ISO 13584-20. It represents the semantics of an unbound **variational_parameter**.

NOTE   ISO 13584-20 requires a subtype of **variable_semantics** to be defined for any subtype of **generic_-variable**. This is intended to provide an interpretation of the role of the variable in its modelling context. The definition of **unbound_variational_parameter_semantics** given below reflects the fact that an unbound **variational_parameter** has no semantics beyond what is implied by its use in an instance of **free_form_constraint** (see clause 5.4.4). It does not necessarily have any immediate physical significance in its own right, though it may indirectly control quantities that do have physical significance.

EXPRESS specification:

```
*)
ENTITY unbound_variational_parameter_semantics
  SUBTYPE OF (variable_semantics);
END_ENTITY;
(*
```