INTERNATIONAL STANDARD

ISO 10303-59

First edition
2008-10-15

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

Part 59: Integrated generic resource — Quality of product shape data

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

Partie 59: Ressource générique intégrée — Qualité des données de forme du produit

Reference number
ISO 10303-59:2008(E)

Contents

page

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 2.

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

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

ISO 10303-59 was prepared by Technical Committee ISO/TC 184, *Automation systems and integration,* Subcommittee SC 4, *Industrial data*.

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

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

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

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

Introduction

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

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

— product data quality definition schema;

— product data quality criteria schema;

— product data quality inspection result schema;

— shape data quality criteria schema;

— shape data quality inspection result schema.

Before going into details of the quality of product data, it will be useful to note that quality of product (physical object), quality of product model, and quality of product model data are different concepts. With reference to the definition of the term 'quality' in ISO 9000, we define the term 'quality of product' as 'a measure of degree of satisfaction of requirements of a product on appearance, functionality and performance'. The conceptual product model is adequately defined in ISO 10303 but there is currently very little provision for the specification of quality of product model. Only tolerance information associated to a product model may be categorized as the information concerning quality of product model. Product model data is a representation of a (conceptual) product model with a finite number of digits. It naturally includes numerical precision problems. Slight inconsistencies amongst geometric entities, topological entities and between geometry and topology may be inappropriate for some applications. All product model data are created by some CAD system and it is well known that the numerical representation method and precision obtained are CAD system dependent. This part of ISO 10303 deals with the erroneous or inappropriate characteristics of product model data.

Other ISO standards that have provisions on the quality of products, including specification, verification and process for improvements, have no information on the quality of product data. Product data quality related specification in ISO 10303 is at present restricted to the overall numerical precision of an entire product model. Provision of more detailed specifications for the representation of the quality of product data, especially that of three dimensional shape data, will promote quality guaranteed (assured) data exchange and will reduce unnecessary rework to adjust the transferred product data to a usable state in the receiving system.

This part of ISO 10303 consists of five mutually related schemas as shown in Figure 1. The first schema, namely the product data quality definition schema, defines high level data elements for managing product data quality information such as the relationship of the product data quality information with the product data inspected.

The second schema, namely the product data quality criteria schema, deals with general product data and specifies how to represent quality criteria and quality measurement requirements. The third schema, namely the product data quality inspection result schema, also deals with general product data and specifies how to represent quality inspection results. Specifications defined in the second and the third schemas are sufficiently general so that quality of various types of product data other than shape data can be dealt with in the future.

Two schemas, the shape data quality criteria schema and the shape data quality inspection result schema, are specializations of the product data quality criteria schema and the product data quality inspection result schema respectively to shape data. The quality of product shape data is modelled in this part of ISO 10303 by classifying practical inconveniences including erroneous data, inappropriate data or other items that hinder successful data exchange. These mainly arise due to inappropriate numerical representation of the underlying mathematical model. The classification is based on the taxonomy of known problems caused by low quality product shape data. Major classifications are ‘erroneous data’ and ‘inapt data’. Problems are further categorized as geometry issues, topology issues, combined geometry/topology issues and geometric model issues.

The target product shape data representation is limited in this part of ISO 10303 to the data conforming to ISO 10303-42 although it may be enhanced to allow other representations including native CAD data in future editions.

Each criterion includes a corresponding measurement requirement. A measurement requirement provides a textual description of how the criterion is to be measured and may have additional attributes and rules to control the test and the element or elements to be tested. It does not provide an algorithm for the measurement process. It is expected that the developer of a product data quality checking program will develop his algorithm for the measurement so that that algorithm satisfies the measurement requirement specified.

NOTE 1 Measurement algorithms are outside the scope of this part of ISO 10303 since it is understood that algorithm development is a competitive arena for engineering system vendors and cannot be standardized by an International Standard.

For the assessment of shape data quality by numerical test, thresholds, which are user definable from application protocols supporting shape models, play a key role. Enhancement of the existing Application Protocols, such as ISO 10303-203[1] and ISO 10303-214[2], to support shape data quality information, or development of new Application Protocols which treat shape data quality will be eased by the use of the procedure suggested in [3]. An example of a typical threshold is a distance threshold for evaluating a gap between a base surface and bounding curves for trimming the effective portion of the surface. That distance threshold implies that if the maximum distance between the surface and the curves is greater than or equal to the specified minimum value, then the gap shall be understood as a quality defect. Appropriate thresholds depend on many factors such as size of a product, design requirements, sensitivity of engineering systems to numerical imprecision, etc. Therefore, what threshold value to use needs to be carefully determined in each business situation based on agreement among business partners.

The shape data quality inspection result schema provides representation of quality inspection results for a specific product shape data with regard to specified quality criteria. The inspection result indicates whether, or not, the product shape data inspected contains quality defects. It may also include detailed information on what type of quality defect is existing, and how serious the defect is, together with the shape element data where the problem is detected. This information is expected to be useful for the quality healing process and to help efficient co-operation between the quality checking process and the healing process of the product shape data.

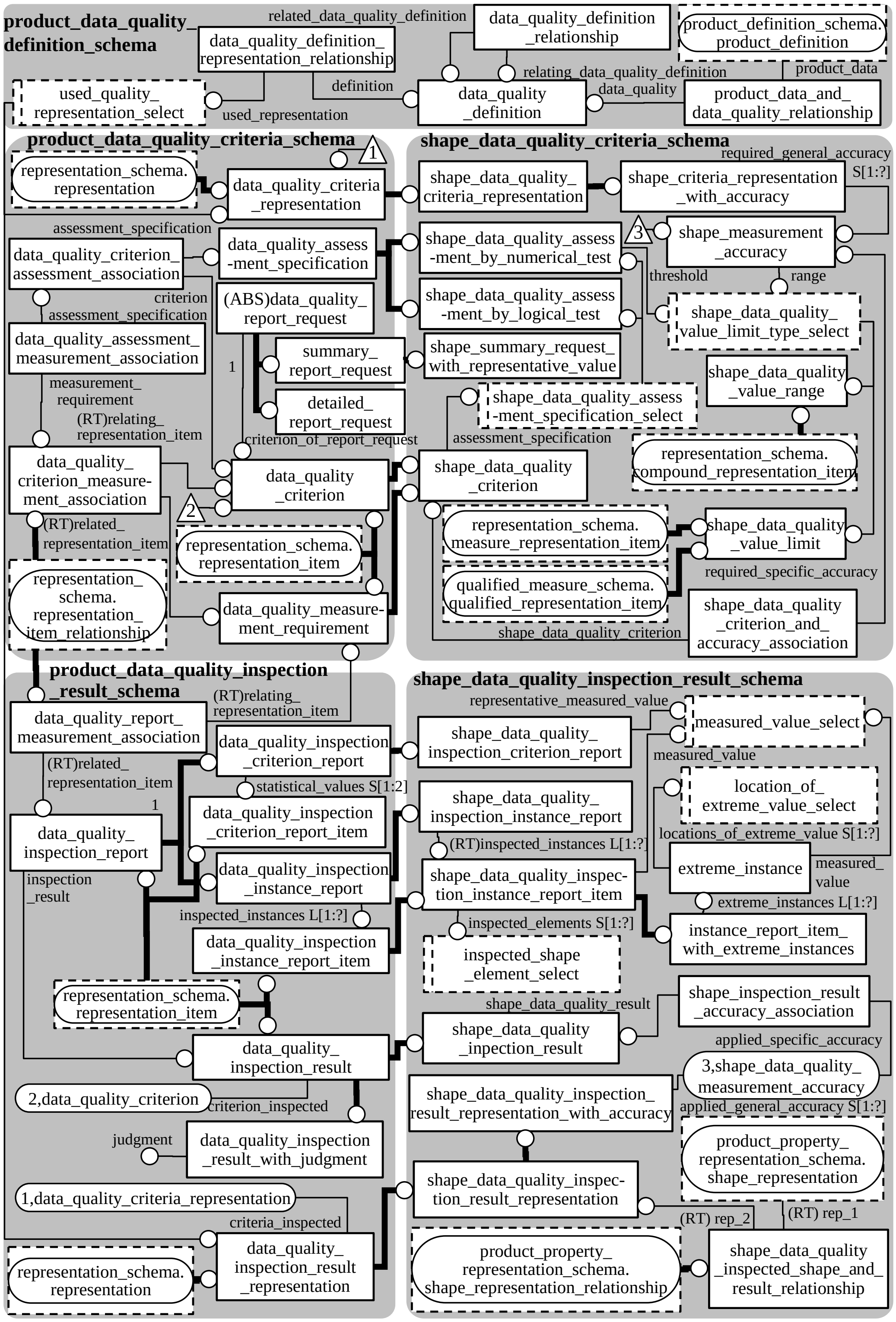

Figure 1 – EXPRESS-G diagram of the overall model structure

NOTE 2 Numbers enclosed in triangles in Figure 1 denote internal references within the figure. This is not standard EXPRESS-G style but is used to prevent complicated long path references. Similar notation is also used in Figures G.2 and G.3.

The selection of instantiated data elements from those shown in Figure 1 depends upon the scenario of quality information usage. Amongst the expected scenarios for the use of this part of ISO 10303 are:

Requirement of quality: The ordering company may require the company receiving the order to create product data that satisfies prescribed quality requirements. Examples are exclusion of infinitesimal geometry smaller than the given tolerance, exclusion of redundant geometry not contributing to the representation of product shape, etc. Very limited information, namely relevant criteria together with required thresholds from those in Figure 1, is necessary in this scenario. The information may be transferred together with the ordering sheet.

Declaration of quality: The creator of a product data may use quality information for explicitly declaring the quality level satisfied by his/her model. Depending on the design method and the CAD system used, the quality of the product data may be unambiguously declared without any inspection. Selective criteria and thresholds for which the model is judged to be quality defect free are required information in this scenario. The quality information may be transferred together with the corresponding product model data.

Assurance of quality: A quality assurance organization may use quality information for representing the results of quality inspection for a particular product model data. This scenario will require inspected quality criteria together with thresholds used, measurement requirements deployed and inspection results obtained. The accuracies used may also be included. The information is transferred together with the corresponding product model data.

Quality information for use in quality improvement: If a quality defect is detected by quality inspection, necessary actions for improving critical data will be required. For that purpose, information on the nature and severity of any quality defects should be provided. Therefore, this scenario will require a detailed inspection result report at the geometric entity instance level. This information is transferred together with the corresponding product model data.

Long term archival of product data: It is desirable that a detailed record of product model data quality is archived with product data. The data requirement for this purpose is similar to that needed for assurance of quality.

Illustrative figures showing required information for these scenarios are presented in Annex G.

Constituent schemas of this part of ISO 10303 are as follows:

Product_data_quality_definition schema: defines high level data elements for managing product data quality information;

Product_data_quality_criteria schema: provides general specifications for the representation of quality criteria, quality measurement requirements and quality assessment specifications for product data;

Product_data_quality_inspection_result schema: provides general specifications for the representation of quality inspection results for a particular product data;

Shape_data_quality_criteria schema: provides representations of shape data quality criteria together with corresponding measurement requirements, thresholds for judging the existence or absence of quality defects and assessment specifications for product shape data;

Shape_data_quality_inspection_result schema: provides representations of quality inspection results for a particular product shape data with regard to specified quality criteria. Detailed information on what type of quality defect is existing, and how serious the defect is, together with the shape element data where the problem is detected can also be represented.

The relationships of the schemas in this part of ISO 10303 to other schemas as defined as the integrated generic resources of ISO 10303 are illustrated in Figure 2 using EXPRESS-G notation. EXPRESS-G is defined in Annex D of ISO 10303-11:2004. The schemas occurring in Figure 2 are components of ISO 10303 integrated generic resources, and they are specified in the following resource parts:

measure_schema	ISO 10303-41
product_definition_schema	ISO 10303-41
product_property_definition_schema	ISO 10303-41
product_property_representation_schema	ISO 10303-41
support_resource_schema	ISO 10303-41
geometric_model_schema	ISO 10303-42
geometry_schema	ISO 10303-42
topology_schema	ISO 10303-42
representation_schema	ISO 10303-43
qualified_measure_schema	ISO 10303-45

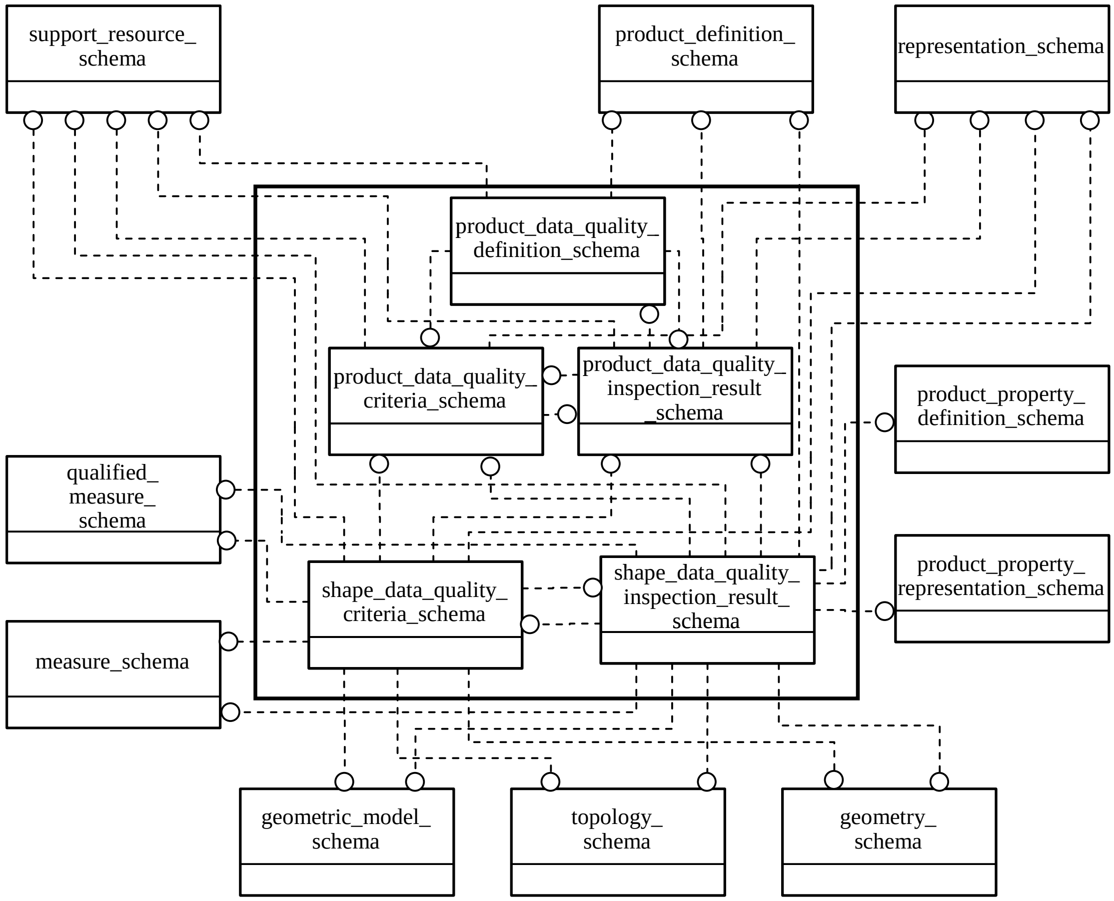

Figure 2 – Schema level diagram of relationships among ISO 10303-59 schemas (inside the box) and with other resource schemas

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

Part 59: Integrated generic resource — Quality of product shape data

1 Scope

This part of ISO 10303 specifies the representation of product data quality especially focusing on three dimensional product shape data. It provides general specifications for the representation of quality criterion, quality measurement requirement, quality assessment specification and quality inspection result of product data. These specifications are provided so that this part of ISO 10303 can be extended to deal with non-shape data quality in the future. By focusing on three dimensional shape data, detailed specifications for the representation of shape data quality criterion together with corresponding measurement requirement, shape data quality assessment specification and detailed results of shape data quality inspection are provided.

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

— representation of high level data elements for managing quality related data;

— representation of general quality criteria for product data;

— representation of general quality criteria for product data associated to the corresponding measurement requirements;

— representation of general quality criteria for product data associated to the corresponding assessment specifications;

— representation of quality inspection results of given product data;

— representation of quality criteria for three dimensional product shape data coupled with measurement requirements for the evaluation of quality criteria and association with the pertinent assessment specifications: quality criteria can be used for the representation of requirement, declaration of, or assurance of, product shape data quality; the target shape data models are equivalent to those defined in ISO 10303-42;

NOTE 1 Detailed requirements for product shape data quality are application context-dependent. This part of ISO 10303 provides a means to select appropriate criteria with required thresholds.

— representation of quality inspection results of given three dimensional product shape data.

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

— degree of satisfaction of design intents;

— quality inspection algorithm;

— relation of shape data quality with tolerance;

NOTE 2 This part of ISO 10303 deals with illegal or inappropriate product (shape) data represented by a finite number of digits where no unique correct solution exists. Therefore, what is specified in this part of ISO 10303 is essentially different from tolerance information.

— detailed information relating to quality of product data other than shape data;

NOTE 3 Though general specifications for the representation of criterion, measurement requirement, assessment specification and inspection result of product data quality are given, detailed specifications are provided only for three dimensional product shape data.

— data model to improve quality of product shape data;

— relationship of design quality and quality of product data;

— aesthetic quality of product shape data.

NOTE 4 Aesthetic quality of product shape is a decisive factor for some types of product, such as passenger cars. However, it is not included in this part of ISO 10303 since technology for the evaluation of the aesthetic quality is not yet well established, even though practical functions for its evaluation such as smooth highlight lines or smooth curvature distribution are deployed.

2 Normative references

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

ISO/IEC 8824-1:2002, *Information technology — Abstract Syntax Notation One (ASN.1) — Part 1: Specification of basic notation*

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

ISO 10303-11:2004, *Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual*

ISO 10303-41:2005, *Industrial automation systems and integration — Product data representation and exchange — Part 41: Integrated generic resource: Fundamentals of product description and support*

ISO 10303-42:2003, *Industrial automation systems and integration — Product data representation and exchange — Part 42: Integrated generic resource: Geometric and topological representation*

ISO 10303-43:2000, *Industrial automation systems and integration — Product data representation and exchange — Part 43: Integrated generic resource: Representation structures*

ISO 10303-45:1998, *Industrial automation systems and integration — Product data representation and exchange — Part 45: Integrated generic resource: Materials*

ISO 9000:2005, *Quality management systems — Fundamentals and vocabulary*

3 Terms, definitions and abbreviations

3.1 Terms defined in ISO 10303-1

For the purposes of this document, the following terms and definitions given in ISO 10303-1 apply:

— application;

— application context;

— application protocol;

— data exchange;

— product;

— product data.

3.2 Terms defined in ISO 10303-42

For the purposes of this document, the following terms and definitions given in ISO 10303-42 apply:

— boundary;

— boundary representation solid model(B-rep);

— bounds;

— curve;

— surface.

3.3 Terms defined in ISO 10303-43

For the purposes of this document, the following terms and definitions given in ISO 10303-43 apply:

— representation;

— context of representation;

— element of representation.

3.4 Terms defined in ISO 9000

For the purposes of this document, the following terms and definitions given in ISO 9000 apply:

— inspection;

— quality;

— quality requirement.

3.5 Other terms and definitions

For the purposes of this document, the following terms and definitions apply:

3.5.1
Hausdorff distance
distance metric between two point sets $\boldsymbol{A}=\{a_1, \ldots, a_m\}$ and $\boldsymbol{B}=\{b_1, \ldots, b_n\}$ defined as:

$$\mathrm{H}(\boldsymbol{A}, \boldsymbol{B}) = \max(\,\mathrm{h}(\boldsymbol{A}, \boldsymbol{B}), \mathrm{h}(\boldsymbol{B}, \boldsymbol{A})\,)$$

where h($\boldsymbol{A}$, $\boldsymbol{B}$) is directed Hausdorff distance from A to B defined as:

$$\mathrm{h}(\boldsymbol{A}, \boldsymbol{B}) = \max_{a \in A} \min_{b \in B} \mathrm{d}(a, b)$$

d(a, b) is Euclidean distance between two points

3.5.2
product data quality
consistency, completeness, and suitability for its purpose of the product data

3.5.3
product data quality criterion
requirement for detecting a quality defect in product data, whose presence can be judged by a logical or a numerical test

3.5.4
product shape data
data representing product shape with geometric and topological information in accordance with ISO 10303-42

3.6 Abbreviations

For the purposes of this document, the following abbreviations apply.

B-rep boundary representation

IR integrated resource (of ISO 10303)

PDQ product data quality

4 Product data quality definition

4.1 Introduction

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

EXPRESS specification:

```
*)
SCHEMA PRODUCT_DATA_QUALITY_DEFINITION_SCHEMA;
 REFERENCE FROM PRODUCT_DATA_QUALITY_CRITERIA_SCHEMA –– ISO 10303-59
  (data_quality_criteria_representation);
 REFERENCE FROM PRODUCT_DATA_QUALITY_INSPECTION_RESULT_SCHEMA –- ISO 10303-59
  (data_quality_inspection_result_representation);
 REFERENCE FROM PRODUCT_DEFINITION_SCHEMA –- ISO 10303-41
  (product_definition);
 REFERENCE FROM SUPPORT_RESOURCE_SCHEMA (text); –- ISO 10303-41
(*
```

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

product_data_quality_criteria_schema	clause 5 of this part of ISO 10303
product_data_quality_inspection_result_schema	clause 6 of this part of ISO 10303
product_definition_schema	ISO 10303-41
support_resource_schema	ISO 10303-41

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

4.2 Fundamental concepts and assumptions

This schema provides the following information;

— identification and definition of product data quality information.

It is understood that data quality information is not a constituent of product data. When requirements on data quality are represented before the actual product design is commenced, they exist without any relationship to a particular product data model. In the case where some product data set is inspected, then the inspection results are related to the particular product data model. Therefore, data quality information shall be identified and defined separately from a product definition with relationship to the pertinent product definition as necessary.

— relationship between product data quality information and product data.

This relationship is required to represent inspection results of a particular product data. Actual relationship will be created between a **data_quality_definition** entity and a **product_definition** entity.

Existence of a **data_quality_definition** entity implies the existence of a set of product data quality criteria. Related inspection results may also exist. The product data quality criteria are represented using entities defined in **product_data_quality_criteria_schema** and the inspection results are represented using entities defined in **product_data_quality_inspection_result_schema**.

4.3 Product data quality definition type definitions

4.3.1 used_quality_representation_select

A **used_quality_representation_select** type allows selection of either **data_quality_criteria-_representation** or **data_quality_inspection_result_representation**.

EXPRESS specification:

```
*)
 TYPE used_quality_representation_select = SELECT
   (data_quality_criteria_representation,
    data_quality_inspection_result_representation);
 END_TYPE;
 (*
```

4.4 Product data quality definition entity definitions

4.4.1 data_quality_definition

A **data_quality_definition** describes aspects of product data quality. Identification of the pertinent product data quality information described will be given by **data_quality_definition_representation-_relationship**. Information on the inspected product data and management data for product data quality shall be related to this entity as required.

EXPRESS specification:

```
*)
 ENTITY data_quality_definition;
    description : text;
 END_ENTITY;
(*
```

Attribute definitions:

description: A textual description characterizing the **data_quality_definition**.

4.4.2 data_quality_definition_representation_relationship

A **data_quality_definition_representation_relationship** relates a **data_quality_definition** to a representation of quality data. The **used_representation** may be a **data_quality_criteria_-representation** as defined in **product_data_quality_criteria_schema** or a **data_quality_inspection-_result_representation** as defined in **product_data_quality_inspection_result_schema.**

EXPRESS specification:

```
*)
 ENTITY data_quality_definition_representation_relationship;
   description      : text;
   definition       : data_quality_definition;
   used_representation : used_quality_representation_select;
 END_ENTITY;
(*
```

Attribute definitions:

description: A textual description characterizing the **data_quality_definition_representation_-relationship**.

definition: **data_quality_definition** that is related to its corresponding quality information in representation level.

used_representation: **used_quality_representation_select** that allows selection of either **data_-quality_criteria_representation** or **data_quality_inspection_result_representation**.

4.4.3 product_data_and_data_quality_relationship

A **product_data_and_data_quality_relationship** relates a **data_quality_definition** to a **product_-definition**.

NOTE When only requirements of product data quality are exchanged, this entity will not be created.

EXPRESS specification:

```
*)
 ENTITY product_data_and_data_quality_relationship;
   description : text;
   product_data : product_definition;
   data_quality : data_quality_definition;
 END_ENTITY;
 (*
```

Attribute definitions:

description: A textual description characterizing the **product_data_and_data_quality_relationship**.

product_data: A **product_definition** whose data is subject to inspection.

data_quality: A **data_quality_definition** that describes the product data quality information.

4.4.4 data_quality_definition_relationship

A **data_quality_definition_relationship** describes a relationship between two **data_quality_-definition**s.

EXAMPLE Product data quality for one product will be checked multiple times along with the advancement of the design of the product with slightly or very different requirements. This entity enables to relate these checks by relating representative **data_quality_definition**s.

EXPRESS specification:

```
*)
 ENTITY data_quality_definition_relationship;
    description                : text;
    relating_data_quality_definition : data_quality_definition;
    related_data_quality_definition  : data_quality_definition;
 END_ENTITY;
(*
```

Attribute definitions:

description: A textual description characterizing the **data_quality_definition_relationship**.

relating_data_quality_definition: One **data_quality_definition** that relates to **related_data_-quality_definition**.

related_data_quality_definition: Another **data_quality_definition** that is related with **relating_-data_quality_definition**.

4.4.5 software_for_data_quality_check

A **software_for_data_quality_check** identifies software required to be used, or used in data quality inspection.

EXPRESS specification:

```
*)
 ENTITY software_for_data_quality_check;
    description    : text;
    software_name   : text;
```



```
    software_version : text;
    data_quality    : data_quality_definition_representation_relationship;
 END_ENTITY;
(*
```

Attribute definitions:

description: A textual description characterizing the **software_for_data_quality_check**.

software_name: The name of the software.

software_version: The version of the software.

data_quality: The product data quality information pertinent to the software identified.

NOTE Other management information concerning data quality check such as who is responsible for the check or when the check was done can be appended in AP level by the use of **management_resource_schema** included in ISO 10303-41 as is done in the existing APs.

EXPRESS specification:

```
*)
END_SCHEMA; -- end product_data_quality_definition_schema
(*
```

5 Product data quality criteria

5.1 Introduction

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

EXPRESS specification:

```
*)
SCHEMA PRODUCT_DATA_QUALITY_CRITERIA_SCHEMA;
  REFERENCE FROM PRODUCT_DATA_QUALITY_INSPECTION_RESULT_SCHEMA -- ISO 10303-59
    (data_quality_inspection_result_representation);
  REFERENCE FROM REPRESENTATION_SCHEMA -- ISO 10303-43
    (representation,
     representation_item,
     representation_item_relationship);
  REFERENCE FROM SUPPORT_RESOURCE_SCHEMA (text ,bag_to_set); -- ISO 10303-41
(*
```

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

product_data_quality_inspection_result_schema	clause 6 of this part of ISO 10303
representation_schema	ISO 10303-43
support_resource_schema	ISO 10303-41

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

5.2 Fundamental concepts and assumptions

In order to unambiguously share product data quality information, the following are required:

a) A clear statement of each quality requirement. Such a requirement will be referred to as ‘quality criterion’ in this standard.

b) Specification of necessary conditions for the measurement to be used for ensuring that a quality criterion is met. Such conditions are referred to as ‘measurement requirement’ in this standard. Each measurement requirement shall specify the types of entities to be checked, the measurement process to be used (if definable), and the required accuracy of the measurement. Measurement processes are specified in descriptive terms only — the actual measurement algorithms are outside the scope of this standard.

c) Specification of assessment for judging the acceptability of the quality measurements. Such a specification is referred to as ‘assessment specification’ in this standard. Assessment specifications for the quality of 3D shape models fall into two classes, the first involving numerical test with one or two thresholds for acceptability, and the second involving logical test (details will be presented in clause 7).

d) Provision of a method for selecting required inspection report type.

e) Specification for the representation of inspection results.

This schema provides representations for items a) – d) that are mandatory in any application.

The information corresponding to e) above is specified in **product_data_quality_inspection_-result_schema**.

NOTE 1 Scenarios of the expected use of this standard are described in Introduction and annex G.

5.2.1 Representation of product data quality criteria

Product data quality requirement can be unambiguously represented by a combination of quality criteria. Each criterion may have one or more associated measurement requirements and assessment specifications.

It is anticipated that there are quality criteria for which it is not possible to specify a measurement requirement. For this reason, this schema provides both quality criteria with or without relationship to the corresponding measurement requirement.

EXAMPLE 1 'unmanufacturable data is not acceptable' is a clear and important product data quality requirement. In this case, 'existence of unmanufacturable data' will become a criterion. Therefore, satisfaction of the criterion implies the existence of quality defect. But, the definition of measurement requirement for completely detecting unmanufacturable data will be very difficult.

EXAMPLE 2 A significant gap between a surface and bounding curves for a face on the surface can cause problems for the receiving system. In order to detect that type of low quality data, the criterion '**gap_between_-edge_and_base_surface**' is defined. In order to judge the quality, the related measurement shall be based on the calculation of the distance between a curve and a surface. Ideally the distance shall be calculated as the minimum distance to the surface from a point on the curve. After calculating this distance for all points on the curve, the maximum value amongst them shall be obtained. Then this maximum distance shall be compared with the threshold defined for this criterion for evaluating if quality defect exists. This example shows how a combination of quality criterion, associated measurement requirement and assessment specification forms a core part of the quality model representation.

Quality requirement is application dependent and target design requirement dependent. This schema allows a user to select a set of necessary criteria from the standardized criteria and combine them to represent his requirements on product data quality.

Specifications in this schema are sufficiently general to be applicable for quality of any property of product data. Specialization of this schema to three dimensional product shape data is given in clause 7. Application to other types of product data quality will require further specialization of this schema to be defined.

5.2.2 Representation of the requirement on measurement of product data quality

Where an inspection report is required, a reliable measurement shall be associated with each quality criterion and used to determine whether, or not, the data is defective with respect to that criterion.
The following are requirements for the reliable measurement:
(1) Appropriate data element shall be measured.

(2) Necessary conditions for appropriate measurement shall be clarified such as mandatory measurement process if definable, what physical quantity shall be measured, what special cases shall be taken into account, required precision of measurement result, etc.
(3) A robust algorithm that satisfies all the requirements represented by (2) above shall be deployed.

NOTE 1 It is understood that inspection algorithms belong to a competitive arena for engineering system vendors and therefore shall not be standardized. For this reason, the requirement (3) above is outside the scope of this standard.

The representation of measurement requirement is designed so that it can be referenced both from **data_quality_criteria_representation** and **data_quality_inspection_result_representation**.

5.2.3 Control of inspection result report

Type of information and level of detail of inspection result report may differ depending the scenario of its use or the purpose of the inspection. The entity **data_quality_report_request** provides the representation of a request for an inspection report with regard to a particular criterion. It has two subtypes, **summary_report_request and detailed_report_request**. The use of **summary_report_-request** is recommended to capture overall quality level such as how many entities were checked regarding the criterion and how many entities among them were judged as having quality defect. The use of **detailed_report_request** is suited to obtain more detailed report such as what entity instance caused quality defect and how serious it is. It has attributes to request a report on all the measured elements or only on elements with inferior quality. Expected number of elements to be reported and expected sequence of report may also be specified. Since these specifications affect other schemas in this standard, it is recommended to look into Figure 1 of Introduction to understand related entities in other schemas. **data_quality_inspection_report** of **product_data_inspection_result_schema** corresponds to **data_quality_report_request** of this schema. It has two subtypes, **data_quality_-inspection_criterion_report** which corresponds to **summary_report_request** of this schema and **data_quality_inspection_instance_report** which corresponds to **detailed_report_request** of this schema. **data_quality_inspection_instance_report** is a collection of **data_quality_inspection_-instance_report_item** which corresponds to one instance or one pair of instances of the given product data.

NOTE In **shape_data_quality_inspection_result_schema**, **data_quality_inspection_instance_report_item** is specialized to **shape_data_quality_inspection_instance_report_item** which has a subtype **instance_report-_item_with_extreme_instances** for representing quality defect in shape entity instance level and how serious it is.

EXAMPLE The following explanation assumes the case where only report on inferior quality elements is requested. In the case of '**short_length_edge**' criterion, two or more edges may be detected as having shorter length than the specified threshold. The entity **data_quality_inspection_instance_report** (or **shape_data_-quality_inspection_instance_report**) will report all the detected edges. The entity **data_quality_inspection_-instance_report_item** (or **shape_data_quality_inspection_instance_report_item**) corresponds to one detected edge. In the case of '**multiply_defined_faces**' criterion, two or more pairs of **face_surface**s may be detected as having duplication. **data_quality_inspection_instance_report** (or **shape_data_quality_inspection_instance_-report**) will report all the pairs of **face_surface**s detected. **data_quality_inspection_instance_report_item** (or **shape_data_quality_inspection_instance_report_item**) corresponds to one pair of **face_surface**s.

5.3 Product data quality criteria type definitions

5.3.1 detailed_report_type

A **detailed_report_type** is a type to allow selection of the required type of detailed inspection result report. It has two enumerated values. One is **MEASURED_ELEMENT** that requests instance report on inspected data element irrespective of whether, or not, it has inferior quality. The other is **INFERIOR_QUALITY_ELEMENT** that requests instance report only for instances having inferior quality. The selection of this type affects to **data_quality_inspection_instance_report_item** as defined in clause 6.4.8.

NOTE This type is used in the definition of **detailed_report_request** entity defined in clause 5.4.10.

EXPRESS specification:

```
*)
 TYPE detailed_report_type = ENUMERATION OF
   (MEASURED_ELEMENT,
    INFERIOR_QUALITY_ELEMENT);
 END_TYPE;
(*
```

Enumerated item definitions:

MEASURED_ELEMENT: A detailed report is required on all the inspected elements represented by **data_quality_inspection_instance_report_item**.

INFERIOR_QUALITY_ELEMENT: A detailed report is required only on elements with quality defects represented by **data_quality_inspection_instance_report_item**.

5.3.2 preferred_ordering

A **preferred_ordering** has two enumerated values, which determines in what order to present detailed report.

NOTE 1 This type is used in the definition of **detailed_report_request** entity defined in clause 5.4.10.

EXPRESS specification:

```
*)
 TYPE preferred_ordering = ENUMERATION OF
   (EXTREMITY_ORDER,
    DETECTED_ORDER);
 END_TYPE;
 (*
```

Enumerated item definitions:

EXTREMITY_ORDER: The inspected data shall be reported in extremity order. The most extreme data shall be reported first.

NOTE 2 Whether the largest value or the smallest value is the most extreme data depends on the criterion.

DETECTED_ORDER: The inspected data shall be reported in detected order.

5.3.3 summary_report_type

A **summary_report_type** is a type to allow selection of required type of summarized inspection result report. It has five enumerated values. The selection of this type affects **data_quality_inspection_-criterion_report_item** as defined in clause 6.4.6.

NOTE This type is used in the definition of **summary_report_request** entity defined in clause 5.4.9.

EXPRESS specification:

```
*)
 TYPE summary_report_type = ENUMERATION OF
  (UNSPECIFIED,
   CONCLUSION_ONLY,
   NUMBER_OF_QUALITY_DEFECTS,
   NUMBER_OF_INSPECTIONS_AND_CONCLUSION,
   FULL_STATISTICS);
 END_TYPE;
(*
```

Enumerated item definitions:

UNSPECIFIED: Report type is not specified.

CONCLUSION_ONLY: Only conclusion showing defect free or not is requested.

NUMBER_OF_QUALITY_DEFECTS: Report on the number of quality defects is requested.

NUMBER_OF_INSPECTIONS_AND_CONCLUSION: Report on the number of inspected elements and conclusion is requested.

FULL_STATISTICS: All types of information, namely, conclusion, the number of inspected elements and the number of quality defects are requested.

5.4 Product data quality criteria entity definitions

5.4.1 data_quality_criteria_representation

A **data_quality_criteria_representation** represents a selected set of criteria on product data quality. An instance of this entity may be used independently to represent only quality requirements or may be used together with **data_quality_inspection_result_representation** as defined in clause 6.3.1 to represent specified requirements and the corresponding inspection results against particular product data.

EXPRESS specification:

```
*)
 ENTITY data_quality_criteria_representation
  SUBTYPE OF(representation);
  WHERE
   WR1 : SIZEOF( QUERY( q <* SELF\representation.items|
                    'PRODUCT_DATA_QUALITY_CRITERIA_SCHEMA.' +
                    'DATA_QUALITY_CRITERION' IN TYPEOF(q))) > 0;
 END_ENTITY;
(*
```

Formal propositions:

WR1: At least one instance of **representation_item** referenced from a **data_quality_criteria_-representation** shall be of type **data_quality_criterion**.

5.4.2 data_quality_criterion

A **data_quality_criterion** represents a criterion that is an aspect of a requirement on product data quality.

EXPRESS specification:

```
*)
 ENTITY data_quality_criterion
  SUBTYPE OF(representation_item);
  WHERE
   WR1 : SIZEOF(QUERY(q <* bag_to_set( USEDIN(SELF,
            'REPRESENTATION_SCHEMA.REPRESENTATION.ITEMS'))
           |'PRODUCT_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
            'DATA_QUALITY_INSPECTION_RESULT_REPRESENTATION'
            IN TYPEOF(q)))=0;
 END_ENTITY;
(*
```

Formal propositions:

WR1: An instance of this entity shall not be used as an **item** of the **data_quality_inspection_-result_representation**.

5.4.3 data_quality_assessment_specification

A **data_quality_assessment_specification** represents a specification to be applied to evaluate inspection results for each criterion.

EXPRESS specification:

```
*)
 ENTITY data_quality_assessment_specification;
   description : text;
 END_ENTITY;
(*
```

Attribute definitions:

description: A textual description characterizing the **data_quality_assessment_specification**.

5.4.4 data_quality_measurement_requirement

A **data_quality_measurement_requirement** represents requirements on acceptable measurement for testing whether the criterion on data quality is satisfied (quality defect exists) or not.

EXPRESS specification:

```
*)
 ENTITY data_quality_measurement_requirement
  SUBTYPE OF(representation_item);
  WHERE
   WR1 : SIZEOF(QUERY(q <*
         bag_to_set( USEDIN(SELF,
            'REPRESENTATION_SCHEMA.REPRESENTATION.ITEMS')) |
          'PRODUCT_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
          'DATA_QUALITY_INSPECTION_RESULT_REPRESENTATION'
          IN TYPEOF(q)))=0;
 END_ENTITY;
(*
```

Formal propositions:

WR1: An instance of this entity shall not be used as an **item** of the **data_quality_inspection_result_-representation**.

5.4.5 data_quality_criterion_assessment_association

A **data_quality_criterion_assessment_association** represents a relationship between a criterion and an assessment specification.

NOTE For a criterion that does not require an assessment specification, this entity will not be instantiated. For a criterion that requires two or more assessment specifications, multiple instances of this entity will be created to represent relationship to each assessment specification. Multiple instances of this entity may also be used to relate an assessment specification to two or more criteria sharing that assessment specification.

EXPRESS specification:

```
*)
 ENTITY data_quality_criterion_assessment_association;
    criterion  : data_quality_criterion;
    assessment_specification : data_quality_assessment_specification;
 END_ENTITY;
(*
```

Attribute definitions:

criterion: A **data_quality_criterion** to be associated.

assessment_specification: The **assessment_specification** associated to the **criterion**.

5.4.6 data_quality_criterion_measurement_association

A **data_quality_criterion_measurement_association** represents a relationship between a criterion and its corresponding measurement requirement.

NOTE For a criterion that does not require a measurement requirement, this entity will not be instantiated. For a criterion that requires two or more measurement requirements, multiple instances of this entity will be created to represent relationship to each measurement requirement. Multiple instances of this entity may also be used to relate a measurement requirement to two or more criteria sharing that measurement requirement.

EXPRESS specification:

```
*)
 ENTITY data_quality_criterion_measurement_association
   SUBTYPE OF(representation_item_relationship);
    SELF\representation_item_relationship.relating_representation_item :
        data_quality_criterion;
    SELF\representation_item_relationship.related_representation_item  :
        data_quality_measurement_requirement;
 END_ENTITY;
(*
```

Attribute definitions:

SELF\representation_item_relationship.relating_representation_item: A **data_quality_criterion** to be associated

SELF\representation_item_relationship.related_representation_item: The measurement requirement associated to the **criterion**.

5.4.7 data_quality_assessment_measurement_association

A **data_quality_assessment_measurement_association** represents a relationship between a measurement requirement and its corresponding assessment specification.

NOTE This entity will be used for specifying the assessment specification and its corresponding measurement requirement when multiple measurement requirements or multiple assessment specifications are specified for a criterion.

EXPRESS specification:

```
*)
 ENTITY data_quality_assessment_measurement_association;
   assessment_specification :
         data_quality_criterion_assessment_association;
   measurement_requirement :
         data_quality_criterion_measurement_association;
  WHERE
   WR1 : SELF.measurement_requirement\representation_item_relationship.
        relating_representation_item :=:
        SELF.assessment_specification.criterion;
 END_ENTITY;
(*
```

Attribute definitions:

assessment_specification: The relationship in which the assessment specification concerned participates.

measurement_requirement: The relationship in which the measurement requirement concerned participates.

Formal propositions:

WR1: The **data_quality_criterion** identified by the attribute **measurement_requirement** and that identified by the attribute **assessment_specification** shall be identical.

5.4.8 data_quality_report_request

A **data_quality_report_request** represents the specification of a report to be generated as a result of inspection for a **data_quality_criterion**. It is either a **summary_report_request** requiring a summarized report or a **detailed_report_request** requiring a report on the details of the inspection result.

EXPRESS specification:

```
*)
 ENTITY data_quality_report_request
  ABSTRACT SUPERTYPE OF (ONEOF(summary_report_request,
                         detailed_report_request));
    description : text;
    criterion_of_report_request : data_quality_criterion;
 END_ENTITY;
(*
```

Attribute definitions:

description: A textual description characterizing the **data_quality_report_request**.

criterion_of_report_request: A **data_quality_criterion** concerned.

5.4.9 summary_report_request

A **summary_report_request** requests a summarized inspection report on the criterion concerned. An entity **data_quality_inspection_criterion_report** as defined in **product_data_quality_-inspection_result_schema** shall be instantiated as a summarized report.

EXPRESS specification:

```
*)
 ENTITY summary_report_request
  SUBTYPE OF(data_quality_report_request);
    summary_report_request_type : summary_report_type;
 END_ENTITY;
(*
```

Attribute definitions:

summary_report_request_type: **summary_report_type** to select one of **UNSPECIFIED, CONCLUSION_ONLY, NUMBER_OF_QUALITY_DEFECTS, NUMBER_OF_-INSPECTIONS_AND_CONCLUSION** and **FULL_STATISTICS**.

5.4.10 detailed_report_request

A **detailed_report_request** requests a detailed inspection report on the criterion concerned. In order to control the volume of the report and report sequence, two attributes are provided.

EXAMPLE What entity instance caused quality defect and what is the measured value calculated will be a typical detailed report that may be used for improving the product model concerned.

NOTE **data_quality_inspection_instance_report** defined in **product_data_quality_inspection_result_-schema** shall be instantiated as the detailed report corresponding to the specification defined in this entity.

<u>EXPRESS specification:</u>

```
*)
 ENTITY detailed_report_request
   SUBTYPE OF(data_quality_report_request);
     detailed_report_request_type : detailed_report_type;
     report_sequence        : preferred_ordering;
 END_ENTITY;
 (*
```

<u>Attribute definitions:</u>

detailed_report_request_type: **detailed_report_type** to select either **MEASURED_ELEMENT** or **INFERIOR_QUALITY_ELEMENT**.

report_sequence: **preferred_ordering** to select either **EXTREMITY_ORDER** or **DETECTED_-ORDER**.

5.4.11 detailed_report_request_with_number_of_data

A **detailed_report_request_with_number_of_data** is a type of **detailed_report_request** that controls maximum number of instances reported.

<u>EXPRESS specification:</u>

```
*)
 ENTITY detailed_report_request_with_number_of_data
   SUBTYPE OF(detailed_report_request);
     number_of_data : INTEGER;
   WHERE
     WR1 : SELF\detailed_report_request.detailed_report_request_type
         <> INFERIOR_QUALITY_ELEMENT;
 END_ENTITY;
(*
```

Attribute definitions:

number_of_data: An INTEGER value to control maximum number of instances reported.

Formal propositions:

WR1: The value of the attribute **detailed_report_request_type** of the supertype **detailed_report_-request** shall not be **INFERIOR_QUALITY_ELEMENT**.

EXPRESS specification:

```
*)
END_SCHEMA; -- end product_data_quality_criteria_schema
(*
```

6 Product data quality inspection result

6.1 Introduction

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

EXPRESS specification:

```
*)
SCHEMA PRODUCT_DATA_QUALITY_INSPECTION_RESULT_SCHEMA;
 REFERENCE FROM PRODUCT_DATA_QUALITY_CRITERIA_SCHEMA; -- ISO 10303-59
 REFERENCE FROM PRODUCT_DATA_QUALITY_DEFINITION_SCHEMA -- ISO 10303-59
   (data_quality_definition_representation,
    product_data_and_data_quality_relationship);
 REFERENCE FROM PRODUCT_DEFINITION_SCHEMA -- ISO 10303-41
   (product_definition);
 REFERENCE FROM REPRESENTATION_SCHEMA -- ISO 10303-43
   (representation,
    representation_item,
    representation_item_relationship);
 REFERENCE FROM SUPPORT_RESOURCE_SCHEMA (bag_to_set); -- ISO 10303-41
(*
```

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

product_data_quality_criteria_schema	clause 5 of this part of ISO 10303
product_data_quality_definition_schema	clause 4 of this part of ISO 10303
product_definition_schema	ISO 10303-41
representation_schema	ISO 10303-43
support_resource_schema	ISO 10303-41

NOTE 2 See annex D, Figure D.3, for a graphical presentation of this schema.

6.2 Fundamental concepts and assumptions

This schema provides general specifications for the representation of the results of quality inspection of product data. Inspection results show the level of product data quality of particular product data instance with regard to specified quality criteria. Therefore, identification of the product data instance inspected and the relationship to corresponding quality criteria specified are provided.

The requirement for inspection results depends upon the usage scenario of product data quality information. When only quality requirements are transferred or declaration of satisfied quality is transferred, inspection results are not required.

NOTE See annex G for details on what kind of quality related information is required for each expected usage scenario.

An instance of **data_quality_inspection_result** represents the inspection result corresponding to exactly one data quality criterion. There are cases that judgement on whether to regard obtained inspection result as quality defect or not is application dependent. Therefore, two entities with and without judgement are provided.

To represent entire inspection result corresponding to a set of specified product data quality criteria, **data_quality_inspection_result_representation** is provided. It is a collection of **data_quality-_inspection_result**s. The relationship between the former and the latter is identical with that between **representation** and **representation_item**.

There are two types of inspection result report. One is a summary report that identifies the quality criterion checked and describes the inspection results without giving detailed information about the extent of the defect and what entity instance was responsible. This information may be used to judge the overall quality level of the target product data with respect to one criterion. The other is a detailed report that represents the inspection result at entity instance level to show which particular entity instance caused what kind and degree of quality defect. This information may be used for improving the quality of the target product data instance.

Specifications in this schema are enough general so that they can be applied for the representation of inspection result of any property of a product data. Specialization of this schema to three dimensional product shape data is given in clause 8. Specialization to inspection result of any property other than product shape will also be possible in the future as a specialization of this schema.

6.3 Product data quality inspection result type definitions

6.3.1 statistical_value_type

A **statistical_value_type** is a type to allow selection of the statistical value for use in the entity **data_quality_inspection_criterion_report_item** defined in clause 6.4.6. It has two enumerated values, one **NUMBER_OF_INSPECTED_INSTANCES**, the other **NUMBER_OF_QUALITY_-DEFECTES_DETECTED**.

EXPRESS specification:

```
*)
 TYPE statistical_value_type = ENUMERATION OF
  (NUMBER_OF_INSPECTED_INSTANCES,
   NUMBER_OF_QUALITY_DEFECTS_DETECTED);
 END_TYPE;
(*
```

Enumerated item definitions:

NUMBER_OF_INSPECTED_INSTANCES: The statistical value to be represented by **data_-quality_inspection_criterion_report_item** is the number of entity instances inspected.

NUMBER_OF_QUALITY_DEFECTS_DETECTED: The statistical value to be represented by **data_quality_inspection_criterion_report_item** is the number of entity instances detected as having quality defect.

6.4 Product data quality inspection result entity definitions

6.4.1 data_quality_inspection_result_representation

A **data_quality_inspection_result_representation** represents inspection results for the specified product data quality criteria against specific product data instance. It consists of one or more **data_quality_inspection_result**s, each corresponding to a specific criterion.

EXPRESS specification:

```
*)
 ENTITY data_quality_inspection_result_representation
  SUBTYPE OF(representation);
   criteria_inspected : data_quality_criteria_representation;
  WHERE
   WR1 : SIZEOF( QUERY( q <* SELF\representation.items|
                   'PRODUCT_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.' +
                   'DATA_QUALITY_INSPECTION_RESULT' IN TYPEOF(q))) > 0;
   WR2 : SIZEOF(inspected_product_definition(SELF)) > 0;
 END_ENTITY;
(*
```

Attribute definitions:

criteria_inspected: The criteria for which the inspection was performed.

Formal propositions:

WR1: At least one instance of **representation_item** referenced from a **data_quality_inspection_-result_representation** shall be of type **data_quality_inspection_result**.

WR2: An instance of this entity shall be referenced from at least one instance of **product_definition** defined in **product_definition_schema** via an instance of **data_quality_definition** defined in **product_data_quality_definition_schema**.

6.4.2 data_quality_inspection_result

A **data_quality_inspection_result** represents the result of inspection regarding a single specified criterion.

EXPRESS specification:

```
*)
 ENTITY data_quality_inspection_result
   SUBTYPE OF(representation_item);
     criterion_inspected : data_quality_criterion;
   WHERE
     WR1 : SIZEOF(QUERY(q <* bag_to_set( USEDIN(SELF,
                      'REPRESENTATION_SCHEMA.REPRESENTATION.ITEMS'))
                    |'PRODUCT_DATA_QUALITY_CRITERIA_SCHEMA.'+
                     'DATA_QUALITY_CRITERIA_REPRESENTATION'
                     IN TYPEOF(q)))=0;
 END_ENTITY;
(*
```

Attribute definitions:

criterion_inspected: **data_quality_criterion** for which the inspection was performed.

Formal propositions:

WR1: An instance of this entity shall not be used as an **item** of the **data_quality_criteria_-representation**.

6.4.3 data_quality_inspection_result_with_judgement

A **data_quality_inspection_result_with_judgement** is a type of **data_quality_inspection_result** that contains information on whether the data inspected have passed or failed the test for the criterion.

EXPRESS specification:

```
*)
 ENTITY data_quality_inspection_result_with_judgement
   SUBTYPE OF(data_quality_inspection_result);
     judgement : BOOLEAN;
 END_ENTITY;
(*
```

Attribute definitions:

judgement: A boolean value representing the fact whether the result of inspection satisfies the data quality criterion specified by **criterion_inspected** or not. The value is TRUE when a quality concern represented by the criterion is detected and FALSE otherwise.

NOTE 1 When the judgement is not required or is not available for some reasons such as incomplete inspection, an instance of **data_quality_inspection_result** shall be used instead of this entity.

NOTE 2 Expression of judgement depends on each application. This attribute is provided as a default representation of judgement. Developers of Application Protocols using this standard may create their own judgement representation based on the detailed information available from **data_quality_inspection_report**.

6.4.4 data_quality_inspection_report

A **data_quality_inspection_report** represents an inspection report with respect to one specific criterion.

EXPRESS specification:

```
*)
 ENTITY data_quality_inspection_report
  SUPERTYPE OF (ONEOF(data_quality_inspection_criterion_report,
                 data_quality_inspection_instance_report))
   SUBTYPE OF(representation_item);
    inspection_result : data_quality_inspection_result;
   WHERE
    WR1 : SIZEOF(QUERY(q <* bag_to_set( USEDIN(SELF,
                   'REPRESENTATION_SCHEMA.REPRESENTATION.ITEMS'))
                 |'PRODUCT_DATA_QUALITY_CRITERIA_SCHEMA.'+
                  'DATA_QUALITY_CRITERIA_REPRESENTATION'
                  IN TYPEOF(q)))=0;
 END_ENTITY;
(*
```

Attribute definitions:

inspection_result: **data_quality_inspection_result** for which the inspection report is represented.

Formal propositions:

WR1: An instance of this entity shall not be used as an **item** of the **data_quality_criteria_-representation**.

6.4.5 data_quality_inspection_criterion_report

A **data_quality_inspection_criterion_report** is a type of **data_quality_inspection_report** that represents summary information of the inspection results for one specific criterion. This entity may report how many instances were inspected and how many instances among them were found to have quality defects with regard to the criterion via **data_quality_inspection_criterion_report_item**.

EXPRESS specification:

```
*)
 ENTITY data_quality_inspection_criterion_report
   SUBTYPE OF(data_quality_inspection_report);
    statistical_values : SET [1:2] OF
                    data_quality_inspection_criterion_report_item;
   WHERE
    WR1 : SIZEOF(QUERY(i1<*bag_to_set(USEDIN(
```

```
          SELF\data_quality_inspection_report.
          inspection_result.criterion_inspected,
          'PRODUCT_DATA_QUALITY_CRITERIA_SCHEMA.'+
          'DATA_QUALITY_REPORT_REQUEST.CRITERION_OF_REPORT_REQUEST'))|
          'PRODUCT_DATA_QUALITY_CRITERIA_SCHEMA.'+
          'SUMMARY_REPORT_REQUEST'
         IN TYPEOF(i1)))>0;
    WR2 : (SIZEOF(SELF.statistical_values) = 1 ) OR
         (SELF.statistical_values[1].value_type <>
          SELF.statistical_values[2].value_type);
 END_ENTITY;
(*
```

Attribute definitions:

statistical_values: One or two **data_quality_inspection_criterion_report_item**s that shows the number of inspected instances and or the number of defects detected.

Formal propositions:

WR1: **data_quality_criterion** associated with an instance of this entity shall be referenced by at least one **summary_report_request**.

WR2: The size of **statistical_values** of this entity shall be one or two. If it is two, **value_type** of two **statistical_values** shall not be identical.

6.4.6 data_quality_inspection_criterion_report_item

A **data_quality_inspection_criterion_report_item** is a type of **representation_item** that represents an integer value that implies **NUMBER_OF_INSPECTED_INSTANCES** or **NUMBER_OF_-QUALITY_DEFECTS_DETECTED**.

EXPRESS specification:

```
*)
 ENTITY data_quality_inspection_criterion_report_item
   SUBTYPE OF(representation_item);
    statistical_value : INTEGER;
    value_type      : statistical_value_type;
   WHERE
    WR1 : SIZEOF(QUERY(q <* bag_to_set( USEDIN(SELF,
                  'REPRESENTATION_SCHEMA.REPRESENTATION.ITEMS'))
                 |'PRODUCT_DATA_QUALITY_CRITERIA_SCHEMA.'+
                  'DATA_QUALITY_CRITERIA_REPRESENTATION'
                  IN TYPEOF(q)))=0;
 END_ENTITY;
(*
```

Attribute definitions:

statistical_value: An integer value.

value_type: statistical_value_type that shows whether **statistical_value** relates to **NUMBER_OF_-INSPECTED_INSTANCES** or to **NUMBER_OF_QUALITY_DEFECTES_DETECTED**.

Formal propositions:

WR1: An instance of this entity shall not be used as an **item** of the **data_quality_criteria_-representation**.

6.4.7 data_quality_inspection_instance_report

A **data_quality_inspection_instance_report** is a type of **data_quality_inspection_report** that represents detailed information of inspection result in entity instance level for one specific criterion. Inspection is made against each instance element of the product data concerned. This entity contains a list of **data_quality_inspection_instance_report_item**s, each representing the inspection result for an instance or a pair of instances of the inspected product data.

NOTE The order of **data_quality_inspection_instance_report_item**s in the list of **inspected_instances** attribute is specified by **data_quality_report_request** as defined in **product_data_quality_criteria_schema**.

EXPRESS specification:

```
*)
 ENTITY data_quality_inspection_instance_report
   SUBTYPE OF(data_quality_inspection_report);
     inspected_instances : LIST [1:?] OF
                         data_quality_inspection_instance_report_item;
   WHERE
     WR1 : SIZEOF(QUERY(i1<*bag_to_set(USEDIN(
             SELF\data_quality_inspection_report.
             inspection_result.criterion_inspected,
             'PRODUCT_DATA_QUALITY_CRITERIA_SCHEMA.'+
             'DATA_QUALITY_REPORT_REQUEST.CRITERION_OF_REPORT_REQUEST'))|
             'PRODUCT_DATA_QUALITY_CRITERIA_SCHEMA.'+
             'DETAILED_REPORT_REQUEST'
            IN TYPEOF(i1)))>0;
 END_ENTITY;
(*
```

Attribute definitions:

inspected_instances: A list of **data_quality_inspection_instance_report_item**s, each representing the inspection result for an instance or a pair of instances of the inspected product data.

Formal propositions:

WR1: **data_quality_criterion** associated with an instance of this entity shall be referenced by at least one **detailed_report_request**.

6.4.8 data_quality_inspection_instance_report_item

A **data_quality_inspection_instance_report_item** represents a detailed inspection result for an inspected element or a pair of inspected elements of the specified product data.

NOTE An instance of this entity may be created for each entity instance inspected. Usage of instanes of this entity can be controlled by **detailed_report_request** defined in **data_quality_criteria_schema** that has attributes to require a report on all measured elements or only on elements with inferior quality, number of elements to be reported and expected report sequence.

EXPRESS specification:

```
*)
 ENTITY data_quality_inspection_instance_report_item
   SUBTYPE OF(representation_item);
   WHERE
     WR1 : SIZEOF(QUERY(q <* bag_to_set( USEDIN(SELF,
                       'REPRESENTATION_SCHEMA.REPRESENTATION.ITEMS'))
                     |'PRODUCT_DATA_QUALITY_CRITERIA_SCHEMA.'+
                      'DATA_QUALITY_CRITERIA_REPRESENTATION'
                      IN TYPEOF(q)))=0;
 END_ENTITY;
(*
```

Formal propositions:

WR1: An instance of this entity shall not be used as an **item** of the **data_quality_criteria_-representation**.

6.4.9 data_quality_report_measurement_association

A **data_quality_report_measurement_association** represents the correspondence between a **data_quality_inspection_report** and the **data_quality_measurement_requirement** used for the inspection.

EXPRESS specification:

```
*)
 ENTITY data_quality_report_measurement_association
   SUBTYPE OF(representation_item_relationship);
     SELF\representation_item_relationship.related_representation_item :
         data_quality_inspection_report;
     SELF\representation_item_relationship.relating_representation_item :
         data_quality_measurement_requirement;
```

```
 END_ENTITY;
(*
```

Attribute definitions:

SELF\representation_item_relationship.related_representation_item: Relating **data_quality_-inspection_report.**

SELF\representation_item_relationship.relating_representation_item: Related **data_quality_-measurement_requirement.**

6.5 Product data quality inspection result function definitions

6.5.1 inspected_product_definition

This function finds all **product_definition**s that is referenced from **data_quality_inspection_-result_representation** given as an input argument of this function via **data_quality_definition_-representation_relationship**, **data_quality_definition** and **product_data_and_data_quality_relationship** defined in **product_data_quality_definition_schema**.

EXPRESS specification:

```
*)
 FUNCTION inspected_product_definition
  (dqir:data_quality_inspection_result_representation)
  :BAG OF product_definition;
  LOCAL
   dqdr :BAG OF data_quality_definition_representation_relationship:= [];
   pdqdr :BAG OF product_data_and_data_quality_relationship:= [];
   pd  :BAG OF product_definition:= [];
  END_LOCAL;
  dqdr := USEDIN(dqir,'PRODUCT_DATA_QUALITY_DEFINITION_SCHEMA.'
               +'DATA_QUALITY_DEFINITION_REPRESENTATION_RELATIONSHIP.'
               +'USED_REPRESENTATION');
  REPEAT i:= 1 TO SIZEOF(dqdr);
   pdqdr := USEDIN(dqdr[i].definition,
              'PRODUCT_DATA_QUALITY_DEFINITION_SCHEMA.'
             +'PRODUCT_DATA_AND_DATA_QUALITY_RELATIONSHIP.'
             +'DATA_QUALITY');
   REPEAT j:= 1 TO SIZEOF(pdqdr);
     pd := pd + pdqdr[j].product_data;
   END_REPEAT;
  END_REPEAT;
  RETURN(pd);
 END_FUNCTION;
(*
```

Argument definitions:

dqir: An instance of **data_quality_inspection_result_representation** of which corresponding **product_definition** instances are to be sought.

pd (Rerturn): A BAG containing all instances of **product_definition** that are referenced by **dqir** via **data_quality_definition_representation_relationship, data_quality_definition** and **product_data-_and_data_quality_relationship.**

EXPRESS specification:

```
*)
END_SCHEMA; -- end product_data_quality_inspection_result_schema
(*
```

7 Shape data quality criteria

7.1 Introduction

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

EXPRESS specification:

```
*)
SCHEMA SHAPE_DATA_QUALITY_CRITERIA_SCHEMA;

 REFERENCE FROM GEOMETRIC_MODEL_SCHEMA; -- ISO 10303-42
 REFERENCE FROM GEOMETRY_SCHEMA; -- ISO 10303-42
 REFERENCE FROM MEASURE_SCHEMA; –- ISO 10303-41
 REFERENCE FROM PRODUCT_DATA_QUALITY_CRITERIA_SCHEMA; –- ISO 10303-59
 REFERENCE FROM PRODUCT_DATA_QUALITY_INSPECTION_RESULT_SCHEMA
  (data_quality_inspection_result_representation, –- ISO 10303-59
   data_quality_report_measurement_association,
   data_quality_inspection_report);
 REFERENCE FROM QUALIFIED_MEASURE_SCHEMA –- ISO 10303-45
  (qualified_representation_item,
   value_qualifier,
   type_qualifier,
   measure_representation_item);
 REFERENCE FROM REPRESENTATION_SCHEMA –- ISO 10303-43
  (compound_representation_item,
   value_representation_item,
   set_representation_item);
 REFERENCE FROM SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA; –- ISO 10303-59
 REFERENCE FROM SUPPORT_RESOURCE_SCHEMA –- ISO 10303-41
  (text,bag_to_set);
 REFERENCE FROM TOPOLOGY_SCHEMA; -- ISO 10303-42
(*
```

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

geometric_model_schema	ISO 10303-42
geometry_schema	ISO 10303-42
measure_schema	ISO 10303-41
product_data_quality_criteria_schema	clause 5 of this part of ISO 10303
product_data_quality_inspection_result_schema	clause 6 of this part of ISO 10303
qualified_measure_schema	ISO 10303-45
representation_schema	ISO 10303-43
shape_data_quality_inspection_result_schema	clause 8 of this part of ISO 10303
support_resource_schema	ISO 10303-41

topology_schema	ISO 10303-42

NOTE 2 See annex D, Figure D.4 – D.16, for a graphical presentation of this schema.

7.2 Fundamental concepts and assumptions

This schema provides representation of quality criteria for three dimensional product shape data. It does not directly define what is good quality data but rather classifies and enumerates quality concerns for detecting low quality data. This schema is a specialization of **product_data_quality_criteria_- schema** presented in clause 5 to the quality of three dimensional product shape data. Therefore, all the fundamental concepts and assumptions described in 5.2 also apply. This schema provides classified shape data quality criteria together with corresponding measurement requirements and thresholds for the assessment of shape data quality. Accuracy is also related to each criterion when appropriate.

7.2.1 Representation of the criteria of product shape data quality and its classification

The quality of product shape data is modelled in this part of ISO 10303 by classifying practical inconveniences which can require remedial action in the receiving system. This identification and classification can potentially improve the efficiency of the data exchange process by reducing the required reworking. These inconveniences mainly arise due to inappropriate numerical representation or inappropriate numerical manipulation of underlying mathematical model, that are highly CAD system dependent. The classification is based on taxonomy of known problems caused by low quality product shape data. They are classified into two major categories that are ‘erroneous data’ and ‘inapt data’.

NOTE 1 It is understood that good quality shape data shall represent design result entirely with acceptable precision where degree of satisfaction of design requirements and design intent is explicitly represented. Though use of 3D engineering systems is expanding, no system fully supports creation and maintenance of good quality shape data as defined above. Satisfaction of the requirement ‘degree of satisfaction of design requirements and design intent is explicitly represented’ is still a big unresolved research issue. This is the reason why we decided to adopt above described classification.

NOTE 2 The product shape data considered shall be represented using entities defined in ISO 10303-42. Amongst the various representation methods possible, such as boundary representation (B-Rep), algebraic representation and CSG, the focus in this part of ISO 10303 is on boundary representation (B-Rep) models.

Problems classified as ‘erroneous data’ are those that violate logical consistency of the product data structure. In other words, it could be said as mathematically invalid data. In contrast, problems classified as ‘inapt data’ are those that are obviously unfavourable for most applications though they are not mathematically invalid. Each category of problems is further classified into topology related ones, geometry related ones, those related to inappropriate relationship between geometry and topology, and those related to shape model.

NOTE 3 Detailed criteria are classified as subtypes of **shape_data_quality_criterion** that is a type of **data_quality_criterion** as defined in clause 5:**product_data_quality_criteria_schema** of this part of ISO 10303.

Aesthetic quality of product shape is a decisive factor for some type of product such as passenger car. But, rigorous definition of aesthetic quality is not included in this standard since technology for its evaluation is not yet well established though practical functions for its evaluation such as smooth highlight lines or smooth curvature distribution are deployed. Only criteria requiring smooth free form curves and surfaces are included.

Not all product shape data accepted in design phase is acceptable in down stream applications such as die design or NC machining. Examples in automobile panel design are spring back compensation and over crown compensation. Draft angle consideration is another example in mould design area. These quality requirements from down stream applications will be typical ‘inapt data’, but inclusion in this standard is hesitated since there is no consensus on their representation. They could be included in the future versions.

7.2.2 Representation of the requirement on measurement of product shape data quality

Each criterion has an associated measurement requirement. A measurement requirement provides a textual description of how the criterion is to be measured and may have additional attributes and rules to control the test and the element or elements to be tested. It does not provide an algorithm for the measurement process. It may be said that a measurement requirement plays a role of external specification for reliable measurement algorithm.

7.2.3 Specialization of data quality assessment specification

data_quality_assessment_specification defined in **product_data_quality_criteria_schema** is specialized to shape data quality by **shape_data_quality_assessment_specification** in this schema that has two subtypes **shape_data_quality_assessment_by_numerical_test** and **shape_data_-quality_assessment_by_logical_test**. For the assessment of **shape_data_quality_assessment_by_-numerical_test**, attribute **threshold**s which are user definable from AP (Application Protocol) play a key role. An example of a typical threshold is a distance threshold for evaluating a gap between a base surface and bounding curves for trimming the effective portion of the surface. That distance threshold implies that if the maximum distance between the surface and the curves is greater than or equal to the specified minimum value, then the gap shall be understood as a quality defect. Appropriate thresholds depend on many factors such as size of a product, design requirements, robustness of engineering systems on numerical precision, etc. Therefore, what threshold value to use shall be carefully determined in each business situation based on agreement among business partners.

7.2.4 Representation of required accuracy for shape data quality inspection

In most cases, especially when free form geometry is involved, the measurement algorithm calculates an approximate solution not an exact solution. In the above example, surface and curves consist of an infinite number of points. Since calculation on an infinite number of points is impossible, any algorithm tries to calculate the solution using a sufficient number of finite points. In order to require the difference of the approximate solution and the exact solution, even when it is unknown, smaller than the expected value, accuracy specification is provided. The intended interpretation of the accuracy specification is that an approximate solution is acceptable if the difference between the approximate solution and any other approximate solution obtained by calculation with a finer distribution of sampling points is smaller than the given accuracy. There are two types of accuracies, one is general accuracy applied to all the measurements and the other is specific accuracy applied only to specified measurement.

NOTE 1 ISO 10303-45 includes **qualified_measure_schema** where specifications to qualify measurement results of physical object from accuracy point of view are defined. Though there are some similarities between the concept defined in that schema and the accuracy concept defined in this standard, the latter concerns with the accuracy of numerical calculation. It therefore includes accuracy of value without any unit such as accuracy of parameter value that is not taken into account in the **qualified_measure_schema**. The accuracy information in this standard is effective for representing requirement of accuracy of numerical calculation even when the target product data is not yet identified.-These are the reasons why the accuracy concept in this standard makes use of only appropriate portion of the **qualified_measure_schema**.

NOTE 2 Default accuracy applicable to all the measurements is represented by the attribute **required_-general_accuracy** of the entity **shape_criteria_representation_with_accuracy**. The accuracy specific to each measurement can be represented by the attribute **required_specific_accuracy** of the entity **shape_-data_quality_criterion_and_accuracy_association**. Actual accuracy applied by quality inspection can be represented using entities in the **shape_data_quality_inspection_result** schema.

7.3 Shape data quality criteria type definitions

7.3.1 shape_data_quality_assessment_specification_select

A **shape_data_quality_assessment_specification_select** type allows selection of either **shape_data-_quality_assessment_by_logical_test** or **shape_data_quality_assessment_by_numerical_test**.

<u>EXPRESS specification:</u>

```
*)
 TYPE shape_data_quality_assessment_specification_select = SELECT
   (shape_data_quality_assessment_by_logical_test,
    shape_data_quality_assessment_by_numerical_test);
 END_TYPE;
(*
```

7.3.2 shape_data_quality_value_limit_type_select

A **shape_data_quality_value_limit_type_select** type allows selection of either **shape_data_-quality_value_range** or **shape_data_quality_value_limit**.

<u>EXPRESS specification:</u>

```
*)
 TYPE shape_data_quality_value_limit_type_select = SELECT
   (shape_data_quality_value_range,
    shape_data_quality_value_limit);
 END_TYPE;
(*
```

7.4 Shape data quality criteria entity definitions

7.4.1 shape_data_quality_criteria_representation

A **shape_data_quality_criteria_representation** represents requirements on shape data quality for a single product shape model. It is a collection of one or more criteria on shape data quality represented by **shape_data_quality_criterion**. An instance of this entity may be used independently to represent quality requirements before product data is created, or may be used together with **shape_data_-quality_inspection_result_representation**, that represents inspection results, to represent what requirement is satisfied or not to what degree by the particular product shape data.

EXPRESS specification:

```
*)
 ENTITY shape_data_quality_criteria_representation
  SUBTYPE OF(data_quality_criteria_representation);
  WHERE
   WR1 : SIZEOF( QUERY( q <* SELF\representation.items|
                    'SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
                    'SHAPE_DATA_QUALITY_CRITERION' IN TYPEOF(q))) > 0;
 END_ENTITY;
(*
```

Formal propositions:

WR1: At least one instance of **representation_item** referenced from a **shape_data_quality_-criteria_representation** shall be of type **shape_data_quality_criterion**.

7.4.2 shape_criteria_representation_with_accuracy

A **shape_criteria_representation_with_accuracy** is a type of **shape_data_quality_criteria_-representation** that contains accuracy information to be applied for measurement.

EXPRESS specification:

```
*)
 ENTITY shape_criteria_representation_with_accuracy
  SUBTYPE OF(shape_data_quality_criteria_representation);
   required_general_accuracy : SET [1:?] OF shape_measurement_accuracy;
 END_ENTITY;
(*
```

Attribute definitions:

required_general_accuracy: A set of **shape_measurement_accuracy** for measurements. This set of accuracies is applied as default values in all the measurements for the criteria identified by this entity.

NOTE Accuracy applied for measurement against a specific criterion can be specified using **shape_data_-quality_criterion_and_accuracy_association** and it supersedes the accuracy defined by this entity if present.

7.4.3 shape_data_quality_criterion

A **shape_data_quality_criterion** represents a criterion that is an aspect of the requirements on shape data quality for a single product shape model together with a measurement requirement representing the necessary conditions for appropriate measurement(s) for the judgement of satisfaction or dissatisfaction of the criterion. It is a supertype of all the detailed criterion classified as **erroneous_-data** and **inapt_data**.

EXPRESS specification:

```
*)
 ENTITY shape_data_quality_criterion
   SUPERTYPE OF (ONEOF(erroneous_data, inapt_data))
   SUBTYPE OF(data_quality_criterion, data_quality_measurement_requirement);
        assessment_specification :
          shape_data_quality_assessment_specification_select;
   WHERE
     WR1 : SIZEOF(USEDIN(SELF,
          'PRODUCT_DATA_QUALITY_CRITERIA_SCHEMA.'+
          'DATA_QUALITY_CRITERION_ASSESSMENT_ASSOCIATION.'+
          'CRITERION'
          )) = 0;
     WR2 : SIZEOF(QUERY(i1<*bag_to_set(
           USEDIN(SELF,
                'REPRESENTATION_SCHEMA.'+
                'REPRESENTATION_ITEM_RELATIONSHIP.'+
                'RELATED_REPRESENTATION_ITEM'))|
          'PRODUCT_DATA_QUALITY_CRITERIA_SCHEMA.'+
          'DATA_QUALITY_CRITERION_MEASUREMENT_ASSOCIATION'
           IN TYPEOF(i1))) = 0;
     WR3 : SIZEOF(QUERY(i1<*bag_to_set(
           USEDIN(SELF,
                'REPRESENTATION_SCHEMA.'+
                'REPRESENTATION_ITEM_RELATIONSHIP.'+
                'RELATING_REPRESENTATION_ITEM'))|
          'PRODUCT_DATA_QUALITY_CRITERIA_SCHEMA.'+
          'DATA_QUALITY_CRITERION_MEASUREMENT_ASSOCIATION'
           IN TYPEOF(i1))) = 0;
 END_ENTITY;
(*
```

Attribute definitions:

assessment_specification: **shape_data_quality_assessment_specification_select** that allows selection of either **shape_data_quality_assessment_by_logical_test** or **shape_data_quality_-**

assessment_by_numerical_test to be applied for the inspection of the criterion.

Formal propositions:

WR1: This entity shall not be used as the **criterion** attribute of **data_quality_criterion_assessment_-association**.

NOTE 1 As this entity has the attribute **assessment_specification** that directly specifies the corresponding assessment specification, it is not required nor recommended to create an instance of **data_quality_criterion_-assessment_association** as defined in clause 5.4.5 for an instance of this entity.

WR2: This entity shall not be used as the **related_representation** attribute of **data_quality_-criterion_measurement_association**.

WR3: This entity shall not be used as the **relating_representation** attribute of **data_quality_criterion_-measurement_association**.

NOTE 2 As this entity is a common subtype of **data_quality_criterion** and **data_quality_measurement_-requirement**, it is not required nor recommended to create an instance of **data_quality_criterion_-measurement _association** as defined in clause 5.4.6 for an instance of this entity.

7.4.4 shape_data_quality_assessment_by_logical_test

A **shape_data_quality_assessment_by_logical_test** represents an assessment specification by a logical test.

EXPRESS specification:

```
*)
 ENTITY shape_data_quality_assessment_by_logical_test
  SUBTYPE OF(data_quality_assessment_specification);
 END_ENTITY;
 (*
```

7.4.5 shape_data_quality_assessment_by_numerical_test

A **shape_data_quality_assessment_by_numerical_test** represents an assessment specification by a numerical test.

EXPRESS specification:

```
*)
 ENTITY shape_data_quality_assessment_by_numerical_test
  SUBTYPE OF(data_quality_assessment_specification);
    threshold : shape_data_quality_value_limit_type_select;
 END_ENTITY;
(*
```

Attribute definitions:

threshold: **shape_data_quality_value_limit_type_select** to select either **shape_data_quality_value-_range** or **shape_data_quality_value_limit** for evaluating the measured value. If the measured value is within the specified range or limit, then the quality concern represented by the criterion is satisfied, namely, quality defect is existing.

7.4.6 shape_data_quality_criterion_and_accuracy_association

A **shape_data_quality_criterion_and_accuracy_association** represents a relationship between a **shape_data_quality_criterion** and a **shape_measurement_accuracy** required for the measurement of the criterion.

EXPRESS specification:

```
*)
 ENTITY shape_data_quality_criterion_and_accuracy_association;
   required_specific_accuracy  : shape_measurement_accuracy;
   shape_data_quality_criterion : shape_data_quality_criterion;
 END_ENTITY;
(*
```

Attribute definitions:

required_specific_accuracy: **shape_measurement_accuracy** for use in the measurement of the criterion concerned.

shape_data_quality_criterion: **shape_data_quality_criterion** to be associated.

NOTE This entity is used for specifying the accuracy of measurement to be used for a specific criterion. When an instance of this entity exists, it supersedes the default accuracy common for every criteria specified by **shape_criteria_representation_with_accuracy**.

7.4.7 shape_measurement_accuracy

A **shape_measurement_accuracy** represents the accuracy of the numerical calculation for use in the measurement. It is either the accuracy required for use in the measurement or the accuracy actually applied in the measurement. It may be specified either as a general accuracy effective for all the measurements or as a specific accuracy effective only for the identified measurement.

EXPRESS specification:

```
*)

 ENTITY shape_measurement_accuracy;
   description : text;
   range     : shape_data_quality_value_limit_type_select;
```

```
 END_ENTITY;
(*
```

Attribute definitions:

description: A textual description characterizing **shape_measurement_accuracy.**

range: **shape_data_quality_value_limit_type_select** to select a range or a limit of a value for representing the accuracy.

7.4.8 shape_data_quality_value_range

A **shape_data_quality_value_range** is a specification of numerical interval which has various subtypes for use in specifying thresholds of criteria or accuracies for measurements.

EXPRESS specification:

```
*)
 ENTITY shape_data_quality_value_range
   SUBTYPE OF(compound_representation_item);
   WHERE
    WR1 : ('REPRESENTATION_SCHEMA.SET_REPRESENTATION_ITEM'
          IN TYPEOF (SELF\compound_representation_item.item_element)) AND
         ((SIZEOF(SELF\compound_representation_item.item_element) = 2)
          AND ((SIZEOF(QUERY (i1 <*
                      SELF\compound_representation_item.item_element |
              ('QUALIFIED_MEASURE_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
                                               TYPEOF(i1)))) = 2)
              OR (SIZEOF(QUERY (i2 <*
                      SELF\compound_representation_item.item_element |
                  ('REPRESENTATION_SCHEMA.VALUE_REPRESENTATION_ITEM'      IN
                                             TYPEOF(i2)))) = 2))) ;
    WR2 : (SIZEOF(QUERY (i <* SELF\compound_representation_item.item_element|
          (i.name = 'upper limit'))) = 1)
         AND(SIZEOF(QUERY (i <*
             SELF\compound_representation_item.item_element |
         (i.name = 'lower limit'))) = 1);
    WR3 : (SIZEOF(QUERY(i1 <*
             SELF\compound_representation_item.item_element |
          ('QUALIFIED_MEASURE_SCHEMA.MEASURE_REPRESENTATION_ITEM'
                                         IN TYPEOF (i1)) AND
          (SIZEOF (QUERY (i2 <*
                 SELF\compound_representation_item.item_element |
               ('QUALIFIED_MEASURE_SCHEMA.MEASURE_REPRESENTATION_ITEM'
                                           IN TYPEOF (i2)) AND
                 (i1 :<>: i2) AND
                 (i1\measure_with_unit.unit_component :=:
                 i2\measure_with_unit.unit_component))) = 1))) = 2)
```

```
        OR (SIZEOF(QUERY(i1 <*
               SELF\compound_representation_item.item_element |
           ('REPRESENTATION_SCHEMA.VALUE_REPRESENTATION_ITEM'
                                          IN TYPEOF (i1)) AND
           (SIZEOF (QUERY (i2 <*
                   SELF\compound_representation_item.item_element |
                  ('REPRESENTATION_SCHEMA.VALUE_REPRESENTATION_ITEM'
                                          IN TYPEOF (i2)) AND
                 (i1 :<>: i2) AND
                 (TYPEOF(i1\value_representation_item.value_component) =
                 TYPEOF(i2\value_representation_item.value_component)))
            ) = 1))) = 2);
 END_ENTITY;
(*
```

Formal propositions:

WR1: **item_element** attribute is of type **set_representation_item** and its size shall be 2. These two elements shall both be of type **measure_representation_item** or of type **value_representation_item.**

WR2: The **name**s of the **representation_item**s appearing within **item_element** shall be ‘upper limit’ and ‘lower limit’ respectively.

WR3: If **item_element** attribute refers to two **measure_representation_item**s, the **unit**s of the corresponding **measure_with_unit**s shall be consistent. If it refers to two **value_representation_-item**s, their types shall be consistent.

7.4.9 shape_data_quality_value_limit

A **shape_data_quality_value_limit** represents a limit of value used in assessment specification.

EXPRESS specification:

```
*)
 ENTITY shape_data_quality_value_limit
  ABSTRACT SUPERTYPE OF (ONEOF(
   shape_data_quality_upper_value_limit,
   shape_data_quality_lower_value_limit))
  SUBTYPE OF(
   measure_representation_item,
   qualified_representation_item);
 END_ENTITY;
(*
```

7.4.10 shape_data_quality_upper_value_limit

A **shape_data_quality_upper_value_limit** represents a upper limit of value used in assessment specification.

EXPRESS specification:

```
*)
 ENTITY shape_data_quality_upper_value_limit
   SUBTYPE OF(shape_data_quality_value_limit);
   WHERE
    WR1 : SELF\qualified_representation_item.qualifiers[1].name
          = 'maximum';
 END_ENTITY;
(*
```

Formal propositions:

WR1: The inherited **qualified_representation_item.qualifiers**[1] **name** attribute shall have the value 'maximum'.

7.4.11 shape_data_quality_lower_value_limit

A **shape_data_quality_lower_value_limit** represents a lower limit of value used in assessment specification.

EXPRESS specification:

```
*)
 ENTITY shape_data_quality_lower_value_limit
   SUBTYPE OF(shape_data_quality_value_limit);
   WHERE
    WR1 : SELF\qualified_representation_item.qualifiers[1].name
     = 'minimum';
 END_ENTITY;
(*
```

Formal propositions:

WR1: The inherited **qualified_representation_item.qualifiers**[1] **name** attribute shall have the value 'minimum'.

7.4.12 shape_summary_request_with_representative_value

A **shape_summary_request_with_representative_value** requests a summarized inspection report with representative measured value on the criterion concerned. An entity **shape_data_quality_-inspection_criterion_report** as defined in **shape_data_quality_inspection_result_schema** shall be instantiated as a summarized report.

EXPRESS specification:

```
*)
 ENTITY shape_summary_request_with_representative_value
```

```
   SUBTYPE OF(summary_report_request);
  END_ENTITY;
(*
```

7.4.13 erroneous_data

An **erroneous_data** is a type of **shape_data_quality_criterion** that is a supertype of criteria for detecting mathematically invalid shape data. It is a supertype of **erroneous_topology**, **erroneous_-geometry**, **erroeous_topology_and_geometry_relationship** and **erroneous_manifold_solid_- brep**.

EXPRESS specification:

```
*)
 ENTITY erroneous_data
  ABSTRACT SUPERTYPE OF (ONEOF(
      erroneous_topology,
      erroneous_geometry,
      erroneous_topology_and_geometry_relationship,
      erroneous_manifold_solid_brep)
    )
  SUBTYPE OF(shape_data_quality_criterion);
 END_ENTITY;
 (*
```

7.4.14 erroneous_topology

An **erroneous_topology** is a type of **erroneous_data** that is a supertype of criteria for detecting topologically invalid shape data.

EXPRESS specification:

```
*)
 ENTITY erroneous_topology
  ABSTRACT SUPERTYPE OF (ONEOF(
    open_edge_loop, open_closed_shell,
    inconsistent_adjacent_face_normals, disconnected_face_set))
  SUBTYPE OF(erroneous_data);
 END_ENTITY;
(*
```

7.4.15 open_edge_loop

An **open_edge_loop** asserts that vertices and **oriented_edge**s constituting of an instance of **edge_loop** do not form topologically closed loop. The measurement requirement corresponding to this entity requires that the measurement shall check all **oriented_edge**s in the **edge_loop** and shall detect the case where the vertex for **edge_start** of an **oriented_edge** differs from the vertex for **edge_end** of the previous **oriented_edge**. The measurement shall also be performed between the last **oriented_edge** and the first **oriented_edge** in the same rule. The case where an **edge_loop** consists of a single **edge** and its **edge_start** and **edge_end** vertices are different shall also be detected.

NOTE 1 This criterion corresponds to the violation of WR1 of **edge_loop** as defined in ISO 10303-42.

NOTE 2 Figure 3 illustrates the case where adjacent **edge_curve**s in an **edge_loop** do not share the same vertex at the location indicated by a dashed circle. Figure 4 illustrates the case where **orientation** value of the **oriented_edge** is inconsistent with the direction of the **edge_loop**. Both cases shall be detected by the measurement of this criterion.

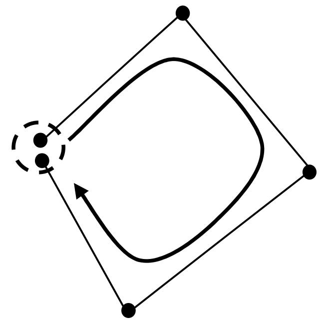

Figure 3 – Incomplete edge loop

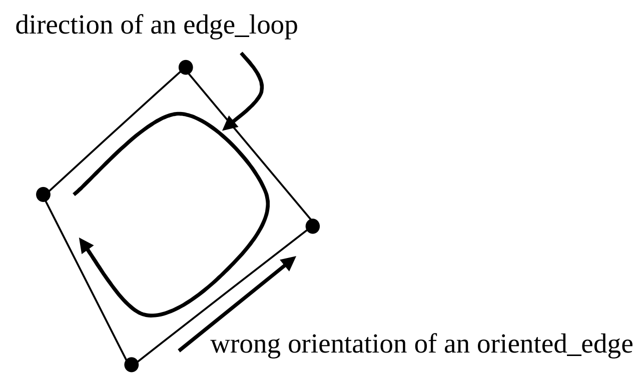

Figure 4 – Wrong orientation of oriented_edge

<u>EXPRESS specification:</u>

```
*)
  ENTITY open_edge_loop
  SUBTYPE OF(erroneous_topology);
   SELF\shape_data_quality_criterion.assessment_specification :
                        shape_data_quality_assessment_by_logical_test;
  WHERE
   WR1 : validate_measured_data_type(SELF,
        'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
```

```
    WR2 : validate_inspected_elements_type(SELF,
        ['TOPOLOGY_SCHEMA.EDGE_LOOP']);
    WR3 : validate_locations_of_extreme_value_type(SELF,
        ['TOPOLOGY_SCHEMA.ORIENTED_EDGE']);
    WR4 : validate_accuracy_types(SELF,
        []);
  END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_by_logical_test** that tests if non connected consecutive edges are included in the **edge_loop**, which is the case 'TRUE' or not.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_report_item** associated with this entity shall be of type **edge_loop**.

WR3: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_report_item_with_extreme_instances** associated with this entity shall be of type **oriented_edge**. It implies that **edge_end** of that **oriented_edge** is different from the **edge_start** of the next **oriented_edge** appearing in the **edge_loop** under inspection.

WR4: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_association**.

7.4.16 open_closed_shell

An **open_closed_shell** asserts that a **closed_shell** contains one or more **edge**s that are referred from faces only once. The measurement requirement corresponding to this entity requires that the measurement shall check if there exist one or more **edge**s that are referred from faces only once.

EXPRESS specification:

```
*)
  ENTITY open_closed_shell
    SUBTYPE OF(erroneous_topology);
      SELF\shape_data_quality_criterion.assessment_specification :
                         shape_data_quality_assessment_by_logical_test;
```

```
   WHERE
     WR1 : validate_measured_data_type(SELF,
          'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
     WR2 : validate_inspected_elements_type(SELF,
          ['TOPOLOGY_SCHEMA.CLOSED_SHELL']);
     WR3 : validate_locations_of_extreme_value_type(SELF,
          ['TOPOLOGY_SCHEMA.EDGE_CURVE']);
     WR4 : validate_accuracy_types(SELF,
          []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests if there exist one or more **edge**s that are referred from faces only once, which is the case ‘TRUE’, or not.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **closed_shell**.

WR3: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance-_report_item_with_extreme_instances** associated with this entity shall be of type **edge_curve**.

WR4: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.17 inconsistent_adjacent_face_normals

An **inconsistent_adjacent_face_normals** asserts that normals of two **face**s that share an **edge** are not consistent with each other in an **open_shell** or a **closed_shell**. The measurement requirement corresponding to this entity requires that the measurement shall check orientation for all **edge**s related to the **face**s in a **closed_shell** or an **open_shell**. If two adjacent faces share an edge with the same orientation, then the topological normals to the faces are in opposite senses and therefore inconsistent.

NOTE In Figure 5, an **edge** is shared by two **face**s, face 1 and face 2, in the same orientation. Therefore, topological normals of the two faces are opposite.

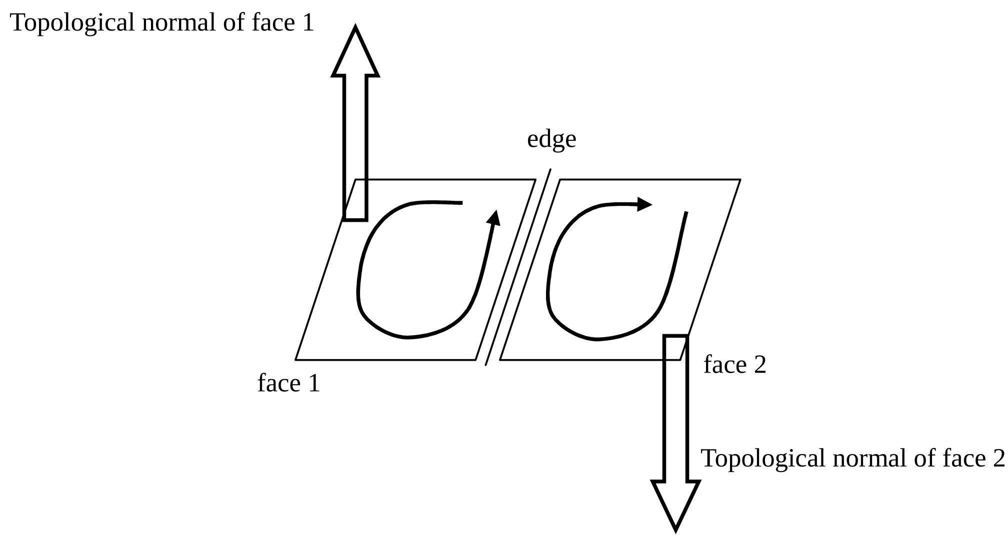

Figure 5 – Inconsistent adjacent face normals

EXPRESS specification:

```
*)
 ENTITY inconsistent_adjacent_face_normals
  SUBTYPE OF(erroneous_topology);
   SELF\shape_data_quality_criterion.assessment_specification :
                        shape_data_quality_assessment_by_logical_test;
  WHERE
   WR1 : validate_measured_data_type(SELF,
         'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
   WR2 : validate_inspected_elements_type(SELF,
         ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
          'CLOSED_OR_OPEN_SHELL_SELECT']);
   WR3 : validate_locations_of_extreme_value_type(SELF,
         ['TOPOLOGY_SCHEMA.EDGE_CURVE']);
   WR4 : validate_accuracy_types(SELF,
         []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical test** that tests if there exist adjacent faces which share an edge with the same orientation, which is the case ‘TRUE’, or not.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **closed_shell** or **open_shell**.

WR3: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **edge_curve**. It implies that normals of the **face**s related to that **edge_curve** are inconsistent.

WR4: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.18 disconnected_face_set

A **disconnected_face_set** asserts that **face**s in a **connected_face_set** are not arcwise connected. The measurement requirement corresponding to this entity requires that the measurement shall create groups of faces in the **connected_face_set** that are connected via edges or vertices. If two or more groups are created for one **connected_face_set**, then it implies the existence of disconnection.

EXPRESS specification:

```
*)
 ENTITY disconnected_face_set
  SUBTYPE OF(erroneous_topology);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_logical_test;
  WHERE
   WR1 : validate_measured_data_type(SELF,
       'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
   WR2 : validate_inspected_elements_type(SELF,
       ['TOPOLOGY_SCHEMA.CONNECTED_FACE_SET']);
   WR3 : validate_locations_of_extreme_value_type(SELF,
       []);
   WR4 : validate_accuracy_types(SELF,
       []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests if the **face**s in the **connected_face_set** are not arcwise connected, which is the case ’TRUE’, or not.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **connected_face_set**.

WR3: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances** since identification of locations of extreme values is meaningless as an inspection result report.

WR4: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.19 erroneous_geometry

An **erroneous_geometry** is a type of **erroneous_data** that is a supertype of criteria for detecting geometrically invalid shape data.

EXPRESS specification:

```
*)
 ENTITY erroneous_geometry
  ABSTRACT SUPERTYPE OF (ONEOF(
    inconsistent_curve_transition_code,
    erroneous_b_spline_curve_definition,
    inconsistent_surface_transition_code,
    erroneous_b_spline_surface_definition))
  SUBTYPE OF(erroneous_data);
 END_ENTITY;
(*
```

7.4.20 erroneous_b_spline_curve_definition

An **erroneous_b_spline_curve_definition** asserts that a **b_spline_curve_with_knots** has inconsistent relationships between the degree, the number of knot values, the knot multiplicities and the number of control points. The function **constraints_param_b_spline** as defined in ISO 10303-42 checks these values. The assessment of this entity shall test if the function **constraints_param_b_spline** as defined in ISO 10303-42:2003, 4.6.22 returns FALSE or not.

NOTE This criterion corresponds to the violation of WR1 of **b_spline_curve_with_knots** as defined in ISO 10303-42.

EXPRESS specification:

```
*)
 ENTITY erroneous_b_spline_curve_definition
   SUBTYPE OF(erroneous_geometry);
    SELF\shape_data_quality_criterion.assessment_specification :
                        shape_data_quality_assessment_by_logical_test;
   WHERE
    WR1 : validate_measured_data_type(SELF,
         'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
    WR2 : validate_inspected_elements_type(SELF,
         ['GEOMETRY_SCHEMA.B_SPLINE_CURVE_WITH_KNOTS']);
    WR3 : validate_locations_of_extreme_value_type(SELF,
         []);
    WR4 : validate_accuracy_types(SELF,
         []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests if the function **constraints_param_b_spline** as defined in ISO 10303-42 returns FALSE or not.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **b_spline_curve_with_knots**.

WR3: The **shape_data_quality_inspection_instance_report_item** corresponding to this entity shall not be **instance_report_item_with_extreme_instances** since identification of locations of extreme values is meaningless as an inspection result report.

WR4: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.21 erroneous_b_spline_surface_definition

An **erroneous_b_spline_surface_definition** asserts that a **b_spline_surface_with_knots** has inconsistent relationships between the degree, the number of knot values, the knot multiplicities and the number of control points. The function **constraints_param_b_spline** as defined in ISO 10303-42 checks these values. The assessment of this entity shall test if the function **constraints_param_b_spline** as defined in ISO 10303-42:2003, 4.6.22 returns FALSE or not.

NOTE This criterion corresponds to the violation of WR1 or WR2 of **b_spline_surface_with_knots** as defined in ISO 10303-42.

<u>EXPRESS specification:</u>

```
*)
 ENTITY erroneous_b_spline_surface_definition
  SUBTYPE OF(erroneous_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
                     shape_data_quality_assessment_by_logical_test;
  WHERE
   WR1 : validate_measured_data_type(SELF,
        'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
   WR2 : validate_inspected_elements_type(SELF,
        ['GEOMETRY_SCHEMA.B_SPLINE_SURFACE_WITH_KNOTS']);
   WR3 : validate_locations_of_extreme_value_type(SELF,
        []);
   WR4 : validate_accuracy_types(SELF,
        []);
 END_ENTITY;
(*
```

<u>Attribute definitions:</u>

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_by_logical_test** that tests if the function **constraints_param_b_spline** as defined in ISO 10303-42 returns FALSE or not.

<u>Formal propositions:</u>

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_report_item** associated with this entity shall be of type **b_spline_surface_with_knots**.

WR3: The **shape_data_quality_inspection_instance_report_item** corresponding to this entity shall not be **instance_report_item_with_extreme_instances** since identification of locations of extreme values is meaningless as an inspection result report.

WR4: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.22 inconsistent_curve_transition_code

An **inconsistent_curve_transition_code** asserts that transition codes of **composite_curve_segment**s in a **composite_curve** are not consistent with the underlying geometry. The measurement requirement corresponding to this entity requires that the measurement shall check for each **composite_curve_-segment** in the **composite_curve** if geometric continuity evaluated at its end point and at the start point of the next segment is consistent with its transition code within given tolerances. When the **composite_curve** is closed, this entity requires that the measurement shall also check continuity at the end point of the last segment and at the start point of the first segment against the transition code of the last segment.

The angle between two tangent vectors, one at the end of current segment and the other at the start of the next segment shall be compared with **angle_tolerance** if the transition code is **cont_same_gradient**.

A check for for principal normal directions is also required if the transition code is **cont_same_-gradient_same_curvature**. Difference of two curvature radius values, one at the end of current segment and the other at the start of the next segment shall also be checked if the transition code is **cont_same_gradient_same_curvature**. To check the curvature radius continuity, let r1 and r2 be the two curvature radius values. Then change ratio of the curvature radii is determined as (2*|r1-r2|) / (|r1| + |r2|). If this change ratio is larger than the **curvature_ratio_tolerance**, then it is not curvature continuous. Distance between the end point of the current segment and the start point of the next segment shall be compared with **distance_tolerance** if the transition code is not **discontinuous**.

Geometry having continuity greater than the transition code shall not be detected by this criterion except for the case where the start point of the first segment and the end point of the last segment are coincident although the transition code of the last segment is discontinuous.

<u>EXPRESS specification:</u>

```
*)
 ENTITY inconsistent_curve_transition_code
  SUBTYPE OF(erroneous_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
                    shape_data_quality_assessment_by_logical_test;
   angle_tolerance              : plane_angle_measure;
   curvature_ratio_tolerance     : ratio_measure;
   distance_tolerance           : length_measure;
   flat_curvature_radius_tolerance : length_measure;
  WHERE
```

```
    WR1 : validate_measured_data_type(SELF,
        'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
    WR2 : validate_inspected_elements_type(SELF,
        ['GEOMETRY_SCHEMA.COMPOSITE_CURVE']);
    WR3 : validate_locations_of_extreme_value_type(SELF,
        ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'COMPOSITE_CURVE_TRANSITION_LOCATOR']);
    WR4 : validate_accuracy_types(SELF,
        []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical test** that tests if there exists any inconsistency between curve transition codes and the continuity of the underlying geometry.

angle_tolerance: The tolerance for checking if directions of two vectors are parallel.

curvature_ratio_tolerance: The tolerance for checking if two curvature values are equal.

distance_tolerance: The tolerance for checking if two points coincide.

flat_curvature_radius_tolerance: The tolerance value to check flat shape.

NOTE If the curvature radius value is larger than **flat_curvature_radius_tolerance**, that curvature value shall not be used to check curvature change ratio. If two curvature radii are larger than this value, then it shall be understood that two segments are G2 continuous. If one of the curvature radii is larger than this value and the other is smaller, then it shall be understood that two segments are G2 discontinuous.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **composite_curve**.

WR3: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **composite_-curve_transition_locator**. It implies that the transition code of the indicated segment is invalid with respect to the geometry of that segment and the next segment. If the indicated segment is the last segment of the **composite_curve**, then the transition code is invalid against geometry of the last segment and the first segment.

WR4: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.23 inconsistent_surface_transition_code

An **inconsistent_surface_transition_code** asserts that transition codes of **surface_patch**es in a **rectangular_composite_surface** are not consistent with its geometry. The measurement requirement corresponding to this entity requires that the measurement shall check for each **surface_patch** in the **rectangular_composite_surface** if geometric continuity to the next patch is consistent with corresponding transition code within given tolerances. This check shall be performed both in *u* and *v* directions. If the **surface_patch** under inspection is the last patch in *u* direction, only check for *v* direction is required but check with adjacent first patch in *u* direction is also required if the **rectangular_composite_surface** is closed in *u* direction. If the **surface_patch** under inspection is the last patch in *v* direction, only check for *u* direction is required but check with adjacent first patch in *v* direction is also required if the **rectangular_composite_surface** is closed in *v* direction. If the **surface_patch** under inspection is the last patch in both *u* and *v* directions, no check is required but check requirement in case the **rectangular_composite_surface** is closed described above also applies. At any point on the common patch boundary, distance of the two points, one evaluated on the current patch and the other evaluated on the next patch shall be compared with **distance_tolerance** if the transition code is not **discontinuous**.

The angle between two tangent vectors at any point of the common patch boundary, one at the end of current patch and the other at the start of the next patch shall be compared with **angle_tolerance** if the transition code is **cont_same_gradient**.

Check for continuity of curvature values and principal normal directions of the isoparametric curves in the cross-boundary direction at the same locations described above is also required if the transition code is **cont_same_gradient_same_curvature**. To check continuity of the curvature values, let r1 and r2 be the two curvature radius values. Then change ratio of the curvature radii is determined as (2*|r1-r2|) / (|r1| + |r2|). If this change ratio is larger than the **curvature_ratio_tolerance**, then it is not curvature continuous. The difference in principal normal directions shall be compared against **angle_tolerance**.

Geometry having continuity greater than the transition code shall not be detected by this criterion except for the case where the start patch and the end patch share a common patch boundary curve although the transition code of the last segment is discontinuous.

<u>EXPRESS specification:</u>

```
*)
 ENTITY inconsistent_surface_transition_code
  SUBTYPE OF(erroneous_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
                    shape_data_quality_assessment_by_logical_test;
   angle_tolerance              : plane_angle_measure;
   curvature_ratio_tolerance      : ratio_measure;
   distance_tolerance           : length_measure;
```

```
    flat_curvature_radius_tolerance  : length_measure;
   WHERE
    WR1 : validate_measured_data_type(SELF,
         'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
    WR2 : validate_inspected_elements_type(SELF,
         ['GEOMETRY_SCHEMA.RECTANGULAR_COMPOSITE_SURFACE']);
    WR3 : validate_locations_of_extreme_value_type(SELF,
         ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
          'RECTANGULAR_COMPOSITE_SURFACE_TRANSITION_LOCATOR']);
    WR4 : validate_accuracy_types(SELF,
         []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests if there exists any inconsistency between the transition codes and the continuity of the underlying geometry.

angle_tolerance: The tolerance for checking if directions of two vectors are parallel.

curvature_ratio_tolerance: The tolerance for checking if two curvature values are equal.

distance_tolerance: The tolerance for checking if two points coincide.

flat_curvature_radius_tolerance: The tolerance value to check flat shape.

NOTE If the curvature radius value is larger than **flat_curvature_radius_tolerance**, that curvature value shall not be used to check curvature change ratio. If two curvature radii are larger than this value, then it shall be understood that two segments are G2 continuous. If one of the curvature radii is larger than this value and the other is smaller, then it shall be understood that two segments are G2 discontinuous.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **rectangular_composite_ surface**.

WR3: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **rectangular_-composite_surface_transition_locator**. It implies that transition code of the indicated patch is invalid with respect to geometry of that patch and the next patch in the indicated direction. If the indicated patch is the last patch of the **rectangular_composite_surface** in the indicated direction, then the transition code is invalid against geometry of the last patch and the first patch in that direction.

WR4: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.24 erroneous_topology_and_geometry_relationship

An **erroneous_topology_and_geometry_relationship** is a type of **erroneous_data** and supertype of criteria for detecting mathematically invalid geometry and topology relationship.

EXPRESS specification:

```
*)
 ENTITY erroneous_topology_and_geometry_relationship
  ABSTRACT SUPERTYPE OF (ONEOF
   inconsistent_edge_and_curve_directions,
   inconsistent_face_and_surface_normals,
   inconsistent_face_and_closed_shell_normals,
   intersecting_loops_in_face,
   wrongly_placed_loop))
  SUBTYPE OF(erroneous_data);
 END_ENTITY;
(*
```

7.4.25 inconsistent_edge_and_curve_directions

For an **edge_curve,** an **inconsistent_edge_and_curve_directions** asserts that directions of **edge** and its curve geometry are not consistent with its sense information. The measurement requirement corresponding to this entity requires that the measurement shall check if sense information of **edge_curve** is correctly implemented. This criterion shall not be applied for an **edge_curve** defining a loop or for the case where **edge_start** and **edge_end** are within **distance_tolerance** of each other.

NOTE 1 To compare the direction of an edge and that of its underlying curve geometry, this entity requires that the measurement shall calculate the parameter values of the nearest point on the curve from its start vertex and also from its end vertex. The **edge_curve** will be detected by this criterion as having inconsistency if the parameter of the start point is larger than that of the end point when the **same_sense** is TRUE and the opposite when it is FALSE.

NOTE 2 Figure 6 shows that the topological direction of an edge and the direction of a curve geometry are opposite. If the **same_sense** of the **edge_curve** is TRUE, this edge shall be detected by this criterion.

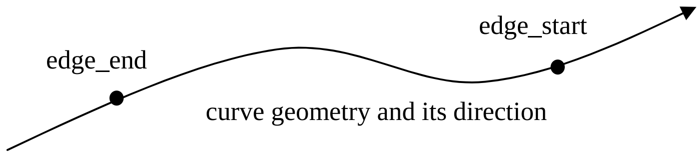

Figure 6 – Inconsistent edge and curve directions

EXPRESS specification:

```
*)
 ENTITY inconsistent_edge_and_curve_directions
  SUBTYPE OF(erroneous_topology_and_geometry_relationship);
   SELF\shape_data_quality_criterion.assessment_specification :
                     shape_data_quality_assessment_by_logical_test;
  WHERE
   WR1 : validate_measured_data_type(SELF,
       'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE')
       ;
   WR2 : validate_inspected_elements_type(SELF,
       ['TOPOLOGY_SCHEMA.EDGE_CURVE']);
   WR3 : validate_locations_of_extreme_value_type(SELF,
       []);
   WR4 : validate_accuracy_types(SELF,
       []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests if the sense information is inconsistent with the direction of the curve geometry or not.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **edge_curve**.

WR3: The **shape_data_quality_inspection_instance_report_item** corresponding to this entity shall not be **instance_report_item_with_extreme_instances** since identification of locations of extreme values is meaningless as an inspection result report.

WR4: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.26 inconsistent_face_and_surface_normals

For a **face_surface**, an **inconsistent_face_and_surface_normals** asserts that topological normal of a face and geometric normal of its underlying surface are not consistent with its sense information. The measurement requirement corresponding to this entity requires that the measurement shall check if face normal determined by outer **edge_loop** is consistent with surface normal. Face normal and surface normal shall be consistent at all points of its bounds. This criterion shall not be applied to the face consisting only of a **vertex_loop**.

NOTE In Figure 7, dashed lines represent the underlying surface of a **face_surface**, and arrowed solid lines on it represent the **edge_loop** of the **face_surface**. The geometric normal of the underlying surface and the topological normal of the **face_surface** are opposite in this case. This **face_surface** shall be detected by this criterion when the value of its **same_sense** attribute is TRUE.

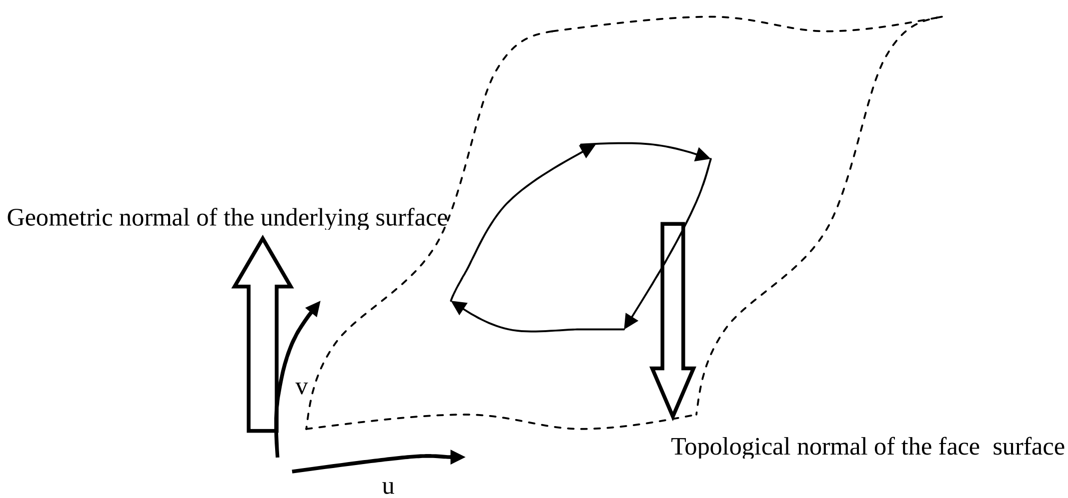

Figure 7 – Inconsistent face and surface notmals

<u>EXPRESS specification:</u>

```
*)
 ENTITY inconsistent_face_and_surface_normals
  SUBTYPE OF(erroneous_topology_and_geometry_relationship);
   SELF\shape_data_quality_criterion.assessment_specification :
                        shape_data_quality_assessment_by_logical_test;
```

```
   WHERE
     WR1 : validate_measured_data_type(SELF,
         'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
     WR2 : validate_inspected_elements_type(SELF,
         ['TOPOLOGY_SCHEMA.FACE_SURFACE']);
     WR3 : validate_locations_of_extreme_value_type(SELF,
         []);
     WR4 : validate_accuracy_types(SELF,
         []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests if topological normal of a face and geometric normal of its underlying surface are not consistent with its sense information or not.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **face_surface**.

WR3: The **shape_data_quality_inspection_instance_report_item** corresponding to this entity shall not be **instance_report_item_with_extreme_instances** since identification of locations of extreme values is meaningless as an inspection result report.

WR4: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.27 inconsistent_face_and_closed_shell_normals

An **inconsistent_face_and_closed_shell_normals** asserts that normal of a **closed_shell** is not consistent with those of faces in it. The measurement requirement corresponding to this entity requires that the measurement shall check if the normal direction of each constituent face is from finite region toward infinite region of the **closed_shell**.

NOTE If the face is **oriented_face**, its normal shall be determined considering its orientation.

EXPRESS specification:

```
*)
 ENTITY inconsistent_face_and_closed_shell_normals
  SUBTYPE OF(erroneous_topology_and_geometry_relationship);
   SELF\shape_data_quality_criterion.assessment_specification :
                      shape_data_quality_assessment_by_logical_test;
  WHERE
   WR1 : validate_measured_data_type(SELF,
       'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
   WR2 : validate_inspected_elements_type(SELF,
       ['TOPOLOGY_SCHEMA.FACE_SURFACE']);
   WR3 : validate_locations_of_extreme_value_type(SELF,
       []);
   WR4 : validate_accuracy_types(SELF,
       []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests if normal of a **closed_shell** is not consistent with those of faces in it or not.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **closed_shell**.

WR3: The **shape_data_quality_inspection_instance_report_item** corresponding to this entity shall not be **instance_report_item_with_extreme_instances** since identification of locations of extreme values is meaningless as an inspection result report.

WR4: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.28 intersecting_loops_in_face

An **intersecting_loops_in_face** asserts that bounds of a **face** have interference each other. The measurement requirement corresponding to this entity requires that the measurement shall detect the case where any combinations of the bounds in a **face** are intersecting each other. If the distance between a point on a bound and a point on another bound is smaller than the **interference_tolerance**, that case shall be detected as intersection.

NOTE Figure 8 shows the case when an inner loop intersects with the outer loop. Figure 9 shows the case when an inner loop touches with the outer loop within **interference_tolerance**. Both cases shall be detected by this criterion.

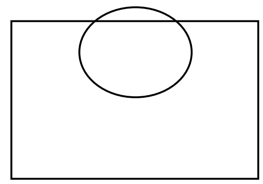

Figure 8 – An inner loop intersecting with the outer

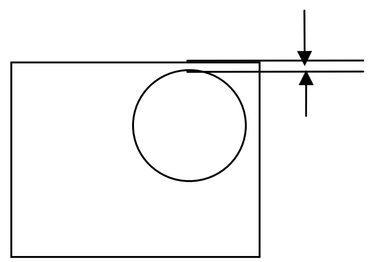

Figure 9 – An inner loop whose distance to the outer loop is smaller than the tolerance

<u>EXPRESS specification:</u>

```
*)
 ENTITY intersecting_loops_in_face
  SUBTYPE OF(erroneous_topology_and_geometry_relationship);
   SELF\shape_data_quality_criterion.assessment_specification :
                       shape_data_quality_assessment_by_logical_test;
   interference_tolerance  : length_measure;
  WHERE
   WR1 : validate_measured_data_type(SELF,
        'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
   WR2 : validate_inspected_elements_type(SELF,
        ['TOPOLOGY_SCHEMA.FACE']);
   WR3 : validate_locations_of_extreme_value_type(SELF,
        ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'POINT_ON_EDGE_CURVE',
         'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'POINT_ON_EDGE_CURVE']);
   WR4 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests if any intersection of bounds exists or not.

interference_tolerance: The tolerance used to check if two points are coincident.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **face**.

WR3: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of two **point_on_edge_-curve**s that are located within the **interference_tolerance** and whose **edge_curve**s are included in different **edge_loop**s in the **face**.

WR4: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.29 wrongly_placed_loop

A **wrongly_placed_loop** asserts that geometric location of **edge_loop**s in a face is illegal. The measurement requirement corresponding to this entity requires that the measurement shall detect two cases, one is the case where an inner loop is located outside of the outer loop, and the other is the case where an inner loop is located inside another inner loop.

NOTE Figure 10 and Figure 12 are typical cases to be detected by this criterion. Figure 11 is the case when an inner loop located outside the outer loop is touching with the outer loop. This case shall be detected by this criterion and also by **intersecting_loops_in_face**.

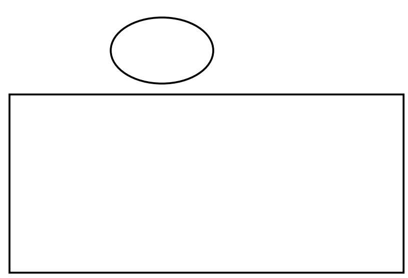

Figure 10 – An inner loop outside of the outer

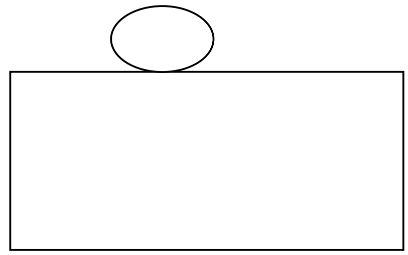

Figure 11 – An inner loop touching the outer from outside

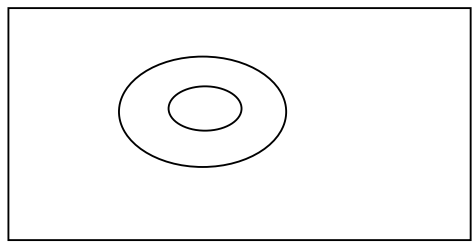

Figure 12 – An inner loop inside another inner loop

EXPRESS specification:

```
*)
 ENTITY wrongly_placed_loop
   SUBTYPE OF(erroneous_topology_and_geometry_relationship);
     SELF\shape_data_quality_criterion.assessment_specification :
                      shape_data_quality_assessment_by_logical_test;
   WHERE
     WR1 : validate_measured_data_type(SELF,
          'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
     WR2 : validate_inspected_elements_type(SELF,
          ['TOPOLOGY_SCHEMA.FACE']);
     WR3 : validate_locations_of_extreme_value_type(SELF,
          ['TOPOLOGY_SCHEMA.FACE_BOUND']);
     WR4 : validate_accuracy_types(SELF,
          ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests if there exists an inner loop located outside of the outer loop or inside of the other inner loop, which is the case ’TRUE’, or not.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **face**.

WR3: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be two **face_bound**s whose location is illegal.

WR4: If the accuracy is specified for this measurement, its **range** shall be of type **length_measure**.

7.4.30 erroneous_manifold_solid_brep

An **erroneous_manifold_solid_brep** is a type of **erroneous_data** that is a supertype of criteria for detecting mathematically invalid **manifold_solid_brep**.

EXPRESS specification:

```
*)
 ENTITY erroneous_manifold_solid_brep
  ABSTRACT SUPERTYPE OF (ONEOF(
   intersecting_shells_in_solid,
   wrongly_placed_void,
   wrongly_oriented_void,
   solid_with_wrong_number_of_voids))
  SUBTYPE OF(erroneous_data);
 END_ENTITY;
(*
```

7.4.31 intersecting_shells_in_solid

An **intersecting_shells_in_solid** asserts that there exist one or more intersections among **closed_-shell**s in a **brep_with_voids**. The measurement requirement corresponding to this entity requires that the measurement shall check intersection of two **closed_shell**s for any combination of **closed_shell**s in a **brep_with_voids**. If the distance between a point on a shell and a point on another shell is smaller than the **interference_tolerance**, that case shall be detected as intersection.

NOTE Figure 13 shows the case where a void is intersecting the outer closed shell. This is a typical case to be detected by this criterion.

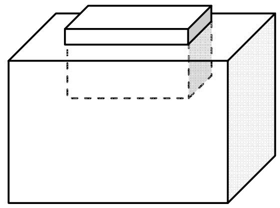

Figure 13 – A void intersecting the outer shell

EXPRESS specification:

```
*)
 ENTITY intersecting_shells_in_solid
   SUBTYPE OF(erroneous_manifold_solid_brep);
     SELF\shape_data_quality_criterion.assessment_specification :
            shape_data_quality_assessment_by_logical_test;
     interference_tolerance : length_measure;
   WHERE
     WR1 : validate_measured_data_type(SELF,
         'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE') ;
     WR2 : validate_inspected_elements_type(SELF,
         ['GEOMETRIC_MODEL_SCHEMA.BREP_WITH_VOIDS']);
     WR3 : validate_locations_of_extreme_value_type(SELF,
         ['GEOMETRY_SCHEMA.SURFACE_CURVE']);
     WR4 : validate_accuracy_types(SELF,
         []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests if there exists any intersection of shells, which is the case 'TRUE', or not.

interference_tolerance: The tolerance used to check if two points are coincident.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **brep_with_voids**.

WR3: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **surface_curve** which represents intersection between shells in the solid.

WR4: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.32 wrongly_placed_void

A **wrongly_placed_void** asserts that a void is wrongly placed with respect to other shells in a solid. The measurement requirement corresponding to this entity requires that the measurement shall check if a void is located outside of the outer shell of a solid or if it is located inside of another void shell.

NOTE 1 A void intersecting with the outer shell and intersecting voids shall be detected by the criteria **intersecting_shells_in_solid**.

NOTE 2 Figure 14 to **Figure 16** show the cases to be detected by this criterion. The case illustrated in Figure 15 shall be detected both by this criterion and by **intersecting_shells_in_solid**.

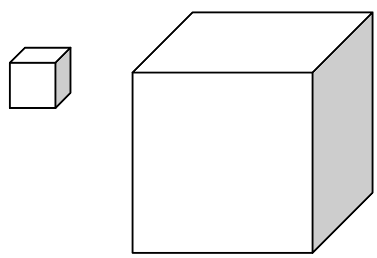

Figure 14 – A void located outside the outer shell

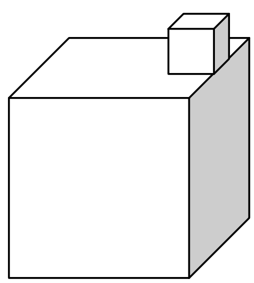

Figure 15 – A void located outside the outer shell and touching it

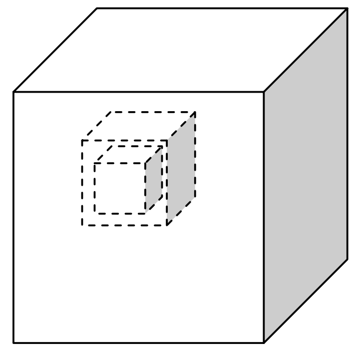

Figure 16 – A void inside of another void

EXPRESS specification:

```
*)
 ENTITY wrongly_placed_void
   SUBTYPE OF(erroneous_manifold_solid_brep);
     SELF\shape_data_quality_criterion.assessment_specification :
                          shape_data_quality_assessment_by_logical_test;
   WHERE
     WR1 : validate_measured_data_type(SELF,
          'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE') ;
     WR2 : validate_inspected_elements_type(SELF,
          ['GEOMETRIC_MODEL_SCHEMA.BREP_WITH_VOIDS']);
     WR3 : validate_locations_of_extreme_value_type(SELF,
          ['TOPOLOGY_SCHEMA.ORIENTED_CLOSED_SHELL']);
     WR4 : validate_accuracy_types(SELF,
          []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests if there exists void located outside of the outer shell or inside of the other void shell, which is the case ’TRUE’, or not.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **brep_with_voids**.

WR3: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **oriented_closed_-shell**. It implies that the **oriented_closed_shell** is wrongly placed.

WR4: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.33 wrongly_oriented_void

A **wrongly_oriented_void** asserts that the orientation of a void shell is illegal. The measurement requirement corresponding to this entity requires that the measurement shall check if the orientation of a void is toward outside of the void, which is the case ‘TRUE’, or not. For properly defined void, the judgement shall always be FALSE.

<u>EXPRESS specification:</u>

```
*)
 ENTITY wrongly_oriented_void
   SUBTYPE OF(erroneous_manifold_solid_brep);
     SELF\shape_data_quality_criterion.assessment_specification :
                      shape_data_quality_assessment_by_logical_test;
   WHERE
     WR1 : validate_measured_data_type(SELF,
         'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE') ;
     WR2 : validate_inspected_elements_type(SELF,
         ['GEOMETRIC_MODEL_SCHEMA.BREP_WITH_VOIDS']);
     WR3 : validate_locations_of_extreme_value_type(SELF,
         ['TOPOLOGY_SCHEMA.ORIENTED_CLOSED_SHELL']);
     WR4 : validate_accuracy_types(SELF,[]);
 END_ENTITY;
(*
```

<u>Attribute definitions:</u>

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests if the orientation of a void is wrong, which is the case ’TRUE’, or not.

<u>Formal propositions:</u>

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-**

value of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **brep_with_voids**.

WR3: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **oriented_closed_-shell**. It implies that the **oriented_closed_shell** is one of the voids of the inspected **brep_with_voids** and its orientation is TRUE.

WR4: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.34 solid_with_wrong_number_of_voids

A **solid_with_wrong_number_of_voids** asserts that number of voids which is implied by the entity instance of the **brep_with_voids** is inconsistent with the number of voids which actually present. Two cases shall be detected by this entity. One is the case where at least one void is not arcwise-connected. This case corresponds to the violation of IP8 of **closed_shell** as defined in ISO 10303-42. Another is the case where an actual void is represented as multiple **closed_shells**.

<u>EXPRESS specification:</u>

```
*)
 ENTITY solid_with_wrong_number_of_voids
  SUBTYPE OF(erroneous_manifold_solid_brep);
   SELF\shape_data_quality_criterion.assessment_specification :
                      shape_data_quality_assessment_by_logical_test;
  WHERE
   WR1 : validate_measured_data_type(SELF,
        'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE') ;
   WR2 : validate_inspected_elements_type(SELF,
        ['GEOMETRIC_MODEL_SCHEMA.BREP_WITH_VOIDS']);
   WR3 : validate_locations_of_extreme_value_type(SELF,[]);
   WR4 : validate_accuracy_types(SELF,[]);
 END_ENTITY;
(*
```

<u>Attribute definitions:</u>

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests if the number of voids which is implied by the entity instance of the **brep_with_voids** is inconsistent with the number of voids which actually present, which is the case 'TRUE', or not.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_report_item** associated with this entity shall be of type **brep_with_voids**.

WR3: The **shape_data_quality_inspection_instance_report_item** corresponding to this entity shall not be **instance_report_item_with_extreme_instances** since identification of locations of extreme values is meaningless as an inspection result report.

WR4: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_association**.

7.4.35 inapt_data

An **inapt_data** is a type of **shape_data_quality_criterion** that is a supertype of criteria for detecting potential problem data from application point of view. Its subtypes are further classified into **inapt_topology**, **inapt_geometry**, **inapt_topology_and_geometry_relationship** and **inapt_manifold_solid_brep**.

NOTE Inappropriate data is not mathematically incorrect data but data which may cause problems in its applications. In some situations they may be acceptable but in other cases they may cause serious troubles. Some inappropriate data are caused by the difference of numerical precision of shape representation or the difference of shape manipulation of CAD systems. Users shall select these criteria with appropriate threshold values according to their business requirement on shape data quality.

EXPRESS specification:

```
*)
 ENTITY inapt_data
  ABSTRACT SUPERTYPE OF (ONEOF(
   inapt_topology,
   inapt_geometry,
   inapt_topology_and_geometry_relationship,
   inapt_manifold_solid_brep))
  SUBTYPE OF(shape_data_quality_criterion);
 END_ENTITY;
(*
```

7.4.36 inapt_topology

An **inapt_topology** is a type of **inapt_data** that is a supertype of criteria for detecting topologically inappropriate shape data

EXPRESS specification:

```
*)
 ENTITY inapt_topology
  ABSTRACT SUPERTYPE OF (ONEOF(
   free_edge, non_manifold_at_edge,
   non_manifold_at_vertex,
   over_used_vertex))
  SUBTYPE OF(inapt_data);
 END_ENTITY;
(*
```

7.4.37 non_manifold_at_vertex

A **non_manifold_at_vertex** asserts that a **connected_face_set** under inspection contains one or more vertices where faces connected to it are in non-manifold situation. The measurement requirement corresponding to this entity requires that the measurement shall take the following processes. First, it shall create a list of the **face**s that connect to the **vertex** under inspection. Then it shall select any **face** from the list and mark **edge**s of that **face** using that **vertex**. The next selected **face** is any **face** sharing one of the already marked **edge**s connected to that **vertex**. After **edge**s connected to the test vertex, of the second **face** are similarly marked, the process continues by looking for another **face** sharing an already marked edge. If at the end of this process, there are **face**s in the list whose **edge**s are not yet been marked, then that **face**s are non-manifold at the **vertex**.

NOTE 1 This test does not detect **non_manifold_at_vertex** at vertices related to **non_manifold_at_edge** . Therefore, this criterion shall always be applied together with **non_manifold_at_edge**.

NOTE 2 Figure 17 shows a typical case to be detected by this criterion.

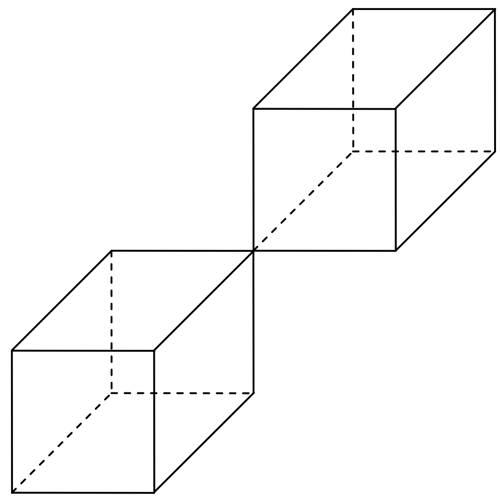

Figure 17 – Non-manifold vertex

EXPRESS specification:

```
*)
 ENTITY non_manifold_at_vertex
   SUBTYPE OF(inapt_topology);
     SELF\shape_data_quality_criterion.assessment_specification :
       shape_data_quality_assessment_by_logical_test;
   WHERE
     WR1 : validate_measured_data_type(SELF,
          'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
     WR2 : validate_inspected_elements_type(SELF,
          ['TOPOLOGY_SCHEMA.CONNECTED_FACE_SET']);
     WR3 : validate_locations_of_extreme_value_type(SELF,
          ['TOPOLOGY_SCHEMA.VERTEX_POINT']);
     WR4 : validate_accuracy_types(SELF,
          []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests if the **connected_face_set** under inspection contains one or more vertices where faces connected to it are in non-manifold situation, which is the case 'TRUE', or not.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-**

value of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **connected_face_set**.

WR3: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **vertex_point**. It implies that one or more **face**s connected to it are non-manifold.

WR4: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.38 non_manifold_at_edge

A **non_manifold_at_edge** asserts that a **connected_face_set** contains one or more edge with non-manifold situation.

NOTE 1 This case may or may not be erroneous according to the type of the inspected connected_face_set.

The measurement requirement corresponding to this entity requires that the measurement shall check for all the **edge**s related to the **face**s in the **connected_face_set** under inspection if that **edge** is referred from three or more **face**s.

NOTE 2 Figure 18 shows a typical case to be detected by this criterion.

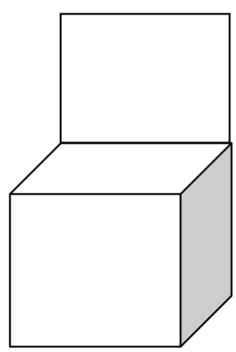

Figure 18 – Non-manifold edge

EXPRESS specification:

```
*)
 ENTITY non_manifold_at_edge
  SUBTYPE OF(inapt_topology);
   SELF\shape_data_quality_criterion.assessment_specification :
```

```
      shape_data_quality_assessment_by_logical_test;
   WHERE
     WR1 : validate_measured_data_type(SELF,
         'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
     WR2 : validate_inspected_elements_type(SELF,
         ['TOPOLOGY_SCHEMA.CONNECTED_FACE_SET']);
     WR3 : validate_locations_of_extreme_value_type(SELF,
         ['TOPOLOGY_SCHEMA.EDGE_CURVE']);
     WR4 : validate_accuracy_types(SELF,
         []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests if the **connected_face_set** contains one or more edges referred from three or more **face**s, which is the case ’TRUE’, or not.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **connected_face_set**.

WR3: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **edge_curve**. It implies that the **edge_curve** is referred from three or more **face**s.

WR4: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.39 over_used_vertex

An **over_used_vertex** asserts that too many **edge**s are connected to a **vertex**. The measurement requirement corresponding to this entity requires that the measurement shall count the number of **edge**s connected to a **vertex** for all vertices that shall be compared with the given threshold in assessment.

EXPRESS specification:

```
*)
 ENTITY over_used_vertex
```

```
   SUBTYPE OF(inapt_topology);
    SELF\shape_data_quality_criterion.assessment_specification :
      shape_data_quality_assessment_by_numerical_test;
   WHERE
    WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
        IN TYPEOF(SELF\shape_data_quality_criterion.
          assessment_specification.threshold))
        AND ('MEASURE_SCHEMA.COUNT_MEASURE'
            IN value_limit_type(SELF\shape_data_quality_criterion.
            assessment_specification.threshold));
    WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.COUNT_MEASURE');
    WR3 : validate_inspected_elements_type(SELF,
        ['TOPOLOGY_SCHEMA.VERTEX_POINT']);
    WR4 : validate_locations_of_extreme_value_type(SELF,
        []);
    WR5 : validate_accuracy_types(SELF,
        []);
  END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the maximum count with the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **count_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **count_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **vertex_point**.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances** since identification of locations of extreme values is meaningless as an inspection result report.

WR5: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.40 free_edge

A **free_edge** asserts that a **connected_face_set** contains one or more edges that have only one reference from a face.

NOTE 1 If the **connected_face_set** is a **closed_shell**, violation to this criterion means erroneous shape. Therefore, that case shall be detected by the criterion **open_closed_shell** that is a type of **erroneous_topology**.

NOTE 2 This criterion detects open shells which are intended to be closed solids. Those shells are often created as a result of translation failure of closed solids. **open_closed_shell** cannot detect them when the shell is not of type **closed_shell**.

NOTE 3 If this criterion is applied to an open shell which represents sheet without thickness, edges representing the outer boundary of the sheet and those representing internal holes are detected. Those edges are not inappropriate.

The measurement requirement corresponding to this entity requires that the measurement shall check for all the **edge**s related to the **face**s in the **connected_face_set**, if the **edge** is referred from some **face** only once.

EXPRESS specification:

```
*)
 ENTITY free_edge
  SUBTYPE OF(inapt_topology);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_logical_test;
  WHERE
   WR1 : validate_measured_data_type(SELF,
        'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
   WR2 : validate_inspected_elements_type(SELF,
        ['TOPOLOGY_SCHEMA.CONNECTED_FACE_SET']);
   WR3 : validate_locations_of_extreme_value_type(SELF,
        ['TOPOLOGY_SCHEMA.EDGE_CURVE']);
   WR4 : validate_accuracy_types(SELF,
        []);
   WR5 : SIZEOF(QUERY(sdqr <* bag_to_set(USEDIN(SELF,
         'PRODUCT_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'DATA_QUALITY_INSPECTION_RESULT.'+
         'CRITERION_INSPECTED'))|
         ('SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
          'SHAPE_DATA_QUALITY_INSPECTION_RESULT'
          IN TYPEOF(sdqr)) AND
          (SIZEOF(QUERY(sdqiir <* bag_to_set(USEDIN(sdqr,
            'PRODUCT_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
            'DATA_QUALITY_INSPECTION_REPORT.'+
            'INSPECTION_RESULT'))|
            ('SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
```

```
            'SHAPE_DATA_QUALITY_INSPECTION_INSTANCE_REPORT'
             IN TYPEOF(sdqiir)) AND
            (SIZEOF(QUERY(sdqiri <* sdqiir.inspected_instances|
                  ('TOPOLOGY_SCHEMA.CLOSED_SHELL'
             IN TYPEOF(sdqiri.inspected_elements[1])))) = 0)
          )) = 0)
        )) = 0;
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests if the **connected_face_set** contains one or more edges that has only one reference from a face, which is the case ’TRUE’, or not.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **connected_face_set**.

WR3: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **edge_curve**. It implies that that **edge_curve** is related to only one **face**.

WR4: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

WR5: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall not be of type **closed_shell**.

7.4.41 inapt_geometry

An **inapt_geometry** is a type of **inapt_data** that is supertype of criteria for detecting geometrically inappropriate shape data.

EXPRESS specification:

```
*)
 ENTITY inapt_geometry
  ABSTRACT SUPERTYPE OF (ONEOF(
```

```
    nearly_degenerate_geometry,
    discontinuous_geometry,
    self_intersecting_geometry,
    geometry_with_local_near_degeneracy,
    overlapping_geometry,
    multiply_defined_geometry,
    geometry_with_local_irregularity,
    overcomplex_geometry))
   SUBTYPE OF(inapt_data);
 END_ENTITY;
(*
```

7.4.42 discontinuous_geometry

A **discontinuous_geometry** is a type of **inapt_geometry** that is a supertype of criteria for detecting geometry data with discontinuity.

EXPRESS specification:

```
*)
 ENTITY discontinuous_geometry
   ABSTRACT SUPERTYPE OF (ONEOF(
    g1_discontinuous_curve, g2_discontinuous_curve,
    g1_discontinuous_surface, g2_discontinuous_surface))
   SUBTYPE OF(inapt_geometry);
 END_ENTITY;
(*
```

7.4.43 g1_discontinuous_curve

A **g1_discontinuous_curve** asserts that a **b_spline_curve** has some internal point where G1 continuity is violated. The measurement requirement corresponding to this entity requires that the measurement shall calculate the maximum angle of two tangent vectors at the selected internal knot points along the **b_spline_curve** one coming to the point under inspection and the other leaving from that point, that shall be compared with the given threshold in assessment. It is sufficient to perform this check at the points where knot multiplicities are greater than or equal to the degree of the **b_spline_curve**.

NOTE In Figure 19, a small black circle is a point on a curve where tangent vectors of the curve at that point are illustrated by a dashed line and a solid line. The curve shall be detected by this criterion if the angle between the dashed line and the solid line is larger than the threshold.

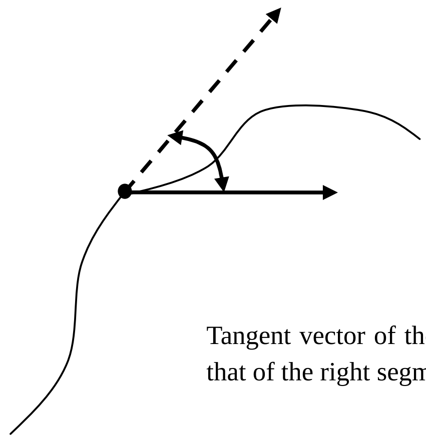

Figure 19 – G1 discontinuous curve

<u>EXPRESS specification:</u>

```
*)
 ENTITY g1_discontinuous_curve
   SUBTYPE OF(discontinuous_geometry);
     SELF\shape_data_quality_criterion.assessment_specification :
       shape_data_quality_assessment_by_numerical_test;
   WHERE
     WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
         'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
         IN TYPEOF(SELF\shape_data_quality_criterion.
           assessment_specification.threshold))
         AND ('MEASURE_SCHEMA.PLANE_ANGLE_MEASURE'
         IN value_limit_type(SELF\shape_data_quality_criterion.
           assessment_specification.threshold));
     WR2 : validate_measured_data_type(SELF,
         'MEASURE_SCHEMA.PLANE_ANGLE_MEASURE');
     WR3 : validate_inspected_elements_type(SELF,
         ['GEOMETRY_SCHEMA.B_SPLINE_CURVE']);
     WR4 : validate_locations_of_extreme_value_type(SELF,
         ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
          'B_SPLINE_CURVE_KNOT_LOCATOR']);
     WR5 : validate_accuracy_types(SELF,
         []);
 END_ENTITY;
(*
```

<u>Attribute definitions:</u>

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured angle against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **plane_angle_-measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **plane_angle_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **b_spline_curve**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **b_spline_curve-_knot_locator**. This indicates the location where G1 continuity is violated

WR5: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.44 g1_discontinuous_surface

A **g1_discontinuous_surface** asserts that a **b_spline_surface** has some internal point where G1 continuity is violated. The measurement requirement corresponding to this entity requires that the measurement shall calculate the maximum angle between two tangent vectors to an isoparametric curve at each selected internal point one coming to that point and the other leaving from that point, that shall be compared with the given threshold in assessment. This check shall be performed both in *u* and *v* directions. It is sufficient to perform this check at the points where knot multiplicities are greater than or equal to the degree of the surface.

EXPRESS specification:

```
*)
 ENTITY g1_discontinuous_surface
  SUBTYPE OF(discontinuous_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
        IN TYPEOF(SELF\shape_data_quality_criterion.
         assessment_specification.threshold))
```



```
        AND ('MEASURE_SCHEMA.PLANE_ANGLE_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
         assessment_specification.threshold)) ;
    WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.PLANE_ANGLE_MEASURE');
    WR3 : validate_inspected_elements_type(SELF,
        ['GEOMETRY_SCHEMA.B_SPLINE_SURFACE']);
    WR4 : validate_locations_of_extreme_value_type(SELF,
        ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'B_SPLINE_SURFACE_KNOT_LOCATOR']);
    WR5 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.PLANE_ANGLE_MEASURE']);
  END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_by_numerical_test** that tests the measured angle against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality_lower_value_limit** and the **value_component** of the **threshold** shall be of type **plane_angle_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **plane_angle_measure**.

WR3: The **inspected_elements for all instances** of **shape_data_quality_inspection_instance_report_item** associated with this entity shall be of type **b_spline_surface**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_report_item_with_extreme_instances** associated with this entity shall be of type **b_spline_curve_knot_locator**. This indicates the location where tangent vectors are not continuous.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **plane_angle_measure**.

7.4.45 g2_discontinuous_curve

A **g2_discontinuous_curve** asserts that a **b_spline_curve** has some internal point where G2 continuity is violated. The measurement requirement corresponding to this entity requires that the measurement shall calculate the maximum difference between the curvature radius values and the directions of the centre of curvature at the selected internal knot points along the curve. Curvature directions shall be compared with the plane angle threshold specified in the **assessment_specification**. Let r1 and r2 be curvature radius values to be compared. Curvature radius change ratio is defined as (2*|r1-r2|) / (|r1| + |r2|). Compare this value with the ratio threshold specified in **assessment_specification_2** to check continuity of curvature radius values. It is sufficient to perform this check at the points where knot multiplicities are greater than (degree - 2) of the curve.

NOTE Figure 20 shows the case where curvature directions are same, but the radii are different. In contrast, Figure 21 shows the case where curvature radii are same while curvature directions are different. Both cases shall be detected by this criterion.

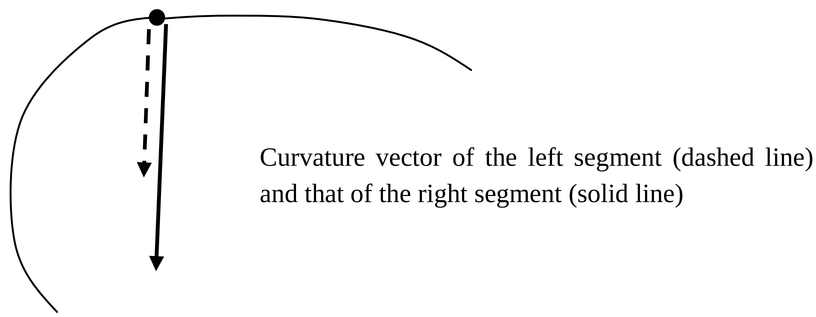

Figure 20 – G2 discontinuous because of the difference of curvature radius

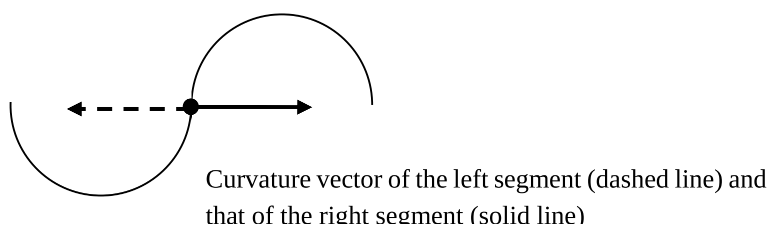

Figure 21 – G2 discontinuous because of the difference of curvature direction

<u>EXPRESS specification:</u>

```
*)
 ENTITY g2_discontinuous_curve
  SUBTYPE OF(discontinuous_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
     shape_data_quality_assessment_by_numerical_test;
   assessment_specification_2                       :
     shape_data_quality_assessment_by_numerical_test;
   flat_curvature_radius_tolerance                   : length_measure;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
```

```
        IN TYPEOF(SELF\shape_data_quality_criterion.
         assessment_specification.threshold))
        AND ('MEASURE_SCHEMA.PLANE_ANGLE_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
         assessment_specification.threshold));
    WR2 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
        IN TYPEOF(SELF.assessment_specification_2.threshold))
        AND ('MEASURE_SCHEMA.RATIO_MEASURE'
        IN value_limit_type(SELF.assessment_specification_2.threshold)) ;
    WR3 : validate_measured_data_type(SELF,
         'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'PLANE_ANGLE_AND_RATIO_PAIR');
    WR4 : validate_inspected_elements_type(SELF,
        ['GEOMETRY_SCHEMA.B_SPLINE_CURVE']);
    WR5 : validate_locations_of_extreme_value_type(SELF,
        ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'B_SPLINE_CURVE_KNOT_LOCATOR']);
    WR6 : validate_accuracy_types(SELF,
        []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification : **shape_data_quality_assessment_-by_numerical_test** that tests the measured angle against the given threshold.

assessment_specification_2 : **shape_data_quality_assessment_by_numerical_test** that tests the curvature change ratio against the given threshold.

flat_curvature_radius_tolerance: the tolerance value to check flat shape.

NOTE If the curvature radius value is larger than **flat_curvature_radius_tolerance**, that curvature value shall not be used to check curvature change ratio If two curvature radii are larger than this value, then it shall be understood that two segments are G2 continuous. If one of the curvature radii is larger than this value and the other is smaller, then it shall be understood that two segments are G2 discontinuous.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_-quality_lower_value_limit** and the **value_component** of the **threshold** shall be of type **plane_-angle_measure**.

WR2: The **threshold** of **assessment_specification_2** of this entity shall be of type **shape_data_-quality_lower_value_limit** and the **value_component** of the **threshold** shall be of type **ratio_-measure**.

WR3: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **plane_angle_and_ratio_pair**.

WR4: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **b_spline_curve**.

WR5: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **b_spline_curve_-knot_locator**. This indicates the point where the curvature values or principal curvature vectors are not continuous.

WR6: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.46 g2_discontinuous_surface

A **g2_discontinuous_surface** asserts that a **b_spline_surface** has some internal point where G2 continuity is violated. The measurement requirement corresponding to this entity requires that the measurement shall calculate the maximum difference between the curvature radius values and the directions of the principal curvatures of the isoparametric curves in the cross boundary directions at the selected internal knot points of the **b_spline_surface**. Principal curvature directions shall be compared with the plane angle threshold specified in the **assessment_specification**. Let r1 and r2 be principal curvature radius values to be compared. Curvature radius change ratio is defined as (2*|r1-r2|) / (|r1| + |r2|). Compare this value with the ratio threshold specified in **assessment_specification_2** to check continuity of curvature radius values. It is sufficient to perform this check at the points where knot multiplicities are greater than (degree **-** 2).

<u>EXPRESS specification:</u>

```
*)
 ENTITY g2_discontinuous_surface
  SUBTYPE OF(discontinuous_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
     shape_data_quality_assessment_by_numerical_test;
   assessment_specification_2   :
     shape_data_quality_assessment_by_numerical_test;
   flat_curvature_radius_tolerance : length_measure;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
        IN TYPEOF(SELF\shape_data_quality_criterion.
         assessment_specification.threshold))
        AND ('MEASURE_SCHEMA.PLANE_ANGLE_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
```

```
          assessment_specification.threshold));
    WR2 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
        IN TYPEOF(SELF.assessment_specification_2.threshold))
        AND ('MEASURE_SCHEMA.RATIO_MEASURE'
        IN value_limit_type(SELF.assessment_specification_2.threshold));
    WR3 : validate_measured_data_type(SELF,
         'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'PLANE_ANGLE_AND_RATIO_PAIR');
    WR4 : validate_inspected_elements_type(SELF,
        ['GEOMETRY_SCHEMA.B_SPLINE_SURFACE']);
    WR5 : validate_locations_of_extreme_value_type(SELF,
        ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'B_SPLINE_SURFACE_KNOT_LOCATOR']);
    WR6 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.RATIO_MEASURE'])
        ;
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured angle against the given threshold.

assessment_specification_2 : **shape_data_quality_assessment_by_numerical_test** that tests the measured curvature change ratio against the given threshold.

flat_curvature_radius_tolerance: The tolerance value to check flat shape.

NOTE If the curvature radius value is larger than **flat_curvature_radius_tolerance**, that curvature value shall not be used to check curvature change ratio If two curvature radii are larger than this value, then it shall be understood that two segments are G2 continuous. If one of the curvature radii is larger than this value and the other is smaller, then it shall be understood that two segments are G2 discontinuous.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **plane_angle_-measure**.

WR2: The **threshold** of **assessment_specification_2** of this entity shall be of type **shape_data_-quality_lower_value_limit** and the **value_component** of the **threshold** shall be of type **ratio_-measure**.

WR3: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-**

value of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **plane_angle_and_ratio_pair**.

WR4: The **inspected_elements for all instances** of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **b_spline_surface**.

WR5: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **b_spline_curve_-knot_locator**. This indicates the location where either curvature value or curvature vectors are not continuous, namely G2 continuity is violated.

WR6: If the accuracy is specified for this measurement, its **value_component** shall be of type **ratio_measure**.

7.4.47 self_intersecting_geometry

A **self_intersecting_geometry** is a type of **inapt_geometry** that is a supertype of criteria for detecting self-intersecting geometry data.

EXPRESS specification:

```
*)
 ENTITY self_intersecting_geometry
  ABSTRACT SUPERTYPE OF (ONEOF(
   self_intersecting_curve, self_intersecting_surface))
  SUBTYPE OF(inapt_geometry);
 END_ENTITY;
(*
```

7.4.48 self_intersecting_curve

A **self_intersecting_curve** asserts that a curve has self-intersection within its domain. The measurement requirement corresponding to this entity requires that the measurement shall detect points on the curve, which are within **interference_tolerance** each other, with arc length distance between them greater than the value determined by multiplying the **interference_tolerance** with **arc_length_distance_factor**. In the case of the closed curve, pairs of points whose arc length is within the **interference_tolerance** of the start point or the end point shall be excluded from the test. For the purpose of this test, a closed curve is one where the distance from the start point to the end point is less than **interference_tolerance.**

NOTE In Figure 22, *d* is the distance between two points on the curve and *l* is the arc length distance between the points.

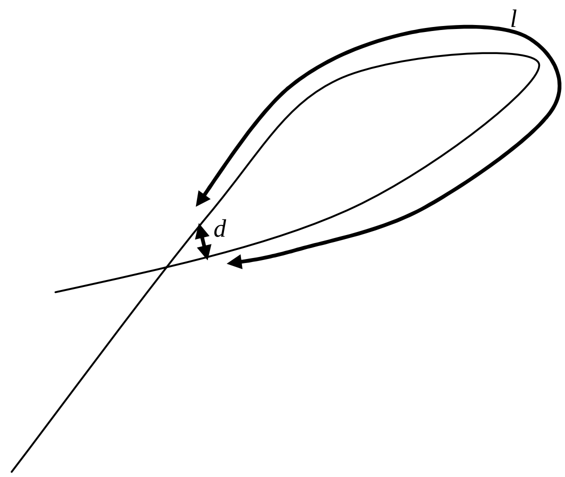

Figure 22 – Self-intersecting curve

EXPRESS specification:

```
*)
 ENTITY self_intersecting_curve
   SUBTYPE OF(self_intersecting_geometry);
     SELF\shape_data_quality_criterion.assessment_specification :
       shape_data_quality_assessment_by_logical_test;
     arc_length_separation_factor  : REAL;
     interference_tolerance : length_measure;
   WHERE
     WR1 : validate_measured_data_type(SELF,
           'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
     WR2 : validate_inspected_elements_type(SELF,
           ['GEOMETRY_SCHEMA.CURVE']);
     WR3 : validate_locations_of_extreme_value_type(SELF,
           ['GEOMETRY_SCHEMA.POINT_ON_CURVE',
           'GEOMETRY_SCHEMA.POINT_ON_CURVE']);
     WR4 : validate_accuracy_types(SELF,
           ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests the existence of intersection, which is the case ’TRUE’, or not.

arc_length_separation_factor: the factor for multiplying the **interference_tolerance** to determine if the arc length between two points is large enough.

interference_tolerance: the tolerance value to check if two points on the curve is coincident or not.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, corresponding to this entity shall be of type **boolean_value**.

WR2: The **inspected_elements for all instances** of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **curve**.

WR3: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be two **point_on_curve**s. It indicates the curve is self-intersecting within tolerance at those two points.

WR4: If the accuracy is specified for this measurement, its **range** shall be of type **length_measure**.

7.4.49 self_intersecting_surface

A **self_intersecting_surface** asserts that a surface has self-intersection within its domain. The measurement requirement corresponding to this entity requires that the measurement shall detect points on the surface, which are within **interference_tolerance** of each other, with a geodesic line distance between them greater than the value determined by multiplying the **interference_tolerance** with **arc_length_distance_factor**. In the case of the closed surface, pairs of points where the arc length distance following an isoparametric curve from the boundary is less than **interference_- tolerance** shall be excluded from the test. For the purpose of this test, a closed surface is one where the distance between the boundary curves at the start and the end parameters is less than **interference_tolerance** in either direction.

NOTE In Figure 23, *d* is the distance between two points on the surface and *l* is the geodesic line distance between them.

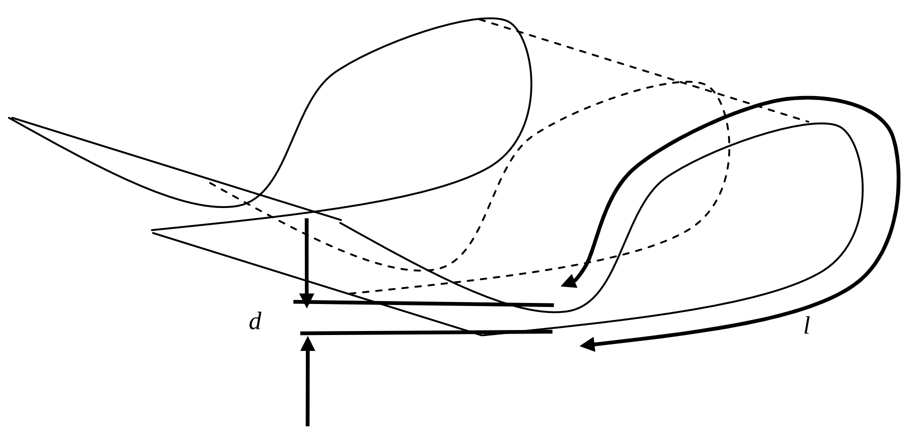

Figure 23 – Self-intersecting surface

EXPRESS specification:

```
*)
 ENTITY self_intersecting_surface
   SUBTYPE OF(self_intersecting_geometry);
     SELF\shape_data_quality_criterion.assessment_specification :
       shape_data_quality_assessment_by_logical_test;
     geodesic_separation_factor  : REAL;
     interference_tolerance : length_measure;
   WHERE
     WR1 : validate_measured_data_type(SELF,
          'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
     WR2 : validate_inspected_elements_type(SELF,
          ['GEOMETRY_SCHEMA.SURFACE']);
     WR3 : validate_locations_of_extreme_value_type(SELF,
          ['GEOMETRY_SCHEMA.POINT_ON_SURFACE',
           'GEOMETRY_SCHEMA.POINT_ON_SURFACE']);
     WR4 : validate_accuracy_types(SELF,
          ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests the existence of intersection, which is the case 'TRUE', or not.

geodesic_separation_factor: The factor for multiplying the **interference_tolerance** to determine if the arc length between two points is large enough.

interference_tolerance: The tolerance value to check if two points on the surface is coincident or not.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, corresponding to this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **surface**.

WR3: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type two **point_on_-surface**s. This indicates that the surface is self-intersecting within tolerance at the two points.

WR4: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.50 nearly_degenerate_geometry

A **nearly_degenerate_geometry** is a type of **inapt_geometry** that is a supertype of criteria for detecting nearly degenerate geometry data.

EXPRESS specification:

```
*)
 ENTITY nearly_degenerate_geometry
  ABSTRACT SUPERTYPE OF (ONEOF(
   small_area_surface,
   short_length_curve,
   entirely_narrow_surface))
  SUBTYPE OF(inapt_geometry);
 END_ENTITY;
(*
```

7.4.51 short_length_curve

A **short_length_curve** asserts that arc length of a curve is smaller than the threshold. The measurement requirement corresponding to this entity requires that the measurement shall calculate the arc length of a curve, that shall be compared with the given threshold in assessment.

EXPRESS specification:

```
*)
 ENTITY short_length_curve
  SUBTYPE OF(nearly_degenerate_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT'
        IN TYPEOF(SELF\shape_data_quality_criterion.
         assessment_specification.threshold))
        AND ('MEASURE_SCHEMA.LENGTH_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
         assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.LENGTH_MEASURE');
   WR3 : validate_inspected_elements_type(SELF,
        ['GEOMETRY_SCHEMA.CURVE']);
   WR4 : validate_locations_of_extreme_value_type(SELF,
        []);
   WR5 : validate_accuracy_types(SELF,
```

```
        ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the arc length of a curve with the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_-quality_upper_value_limit** and the **value_component** of the **threshold** shall be of type **length_-measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item,** for all instances associated with this entity shall be of type **length_measure**.

WR3: The **inspected_elements for all instances** of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **curve**.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length-_measure**.

7.4.52 small_area_surface

A **small_area_surface** asserts that area of a surface is too small. The measurement requirement corresponding to this entity requires that the measurement shall calculate the area of a surface, that shall be compared with the given threshold in assessment. Surface with infinite area shall be ignored when calculating the area.

EXPRESS specification:

```
*)
 ENTITY small_area_surface
  SUBTYPE OF(nearly_degenerate_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT'
        IN TYPEOF(SELF\shape_data_quality_criterion.
         assessment_specification.threshold))
        AND ('MEASURE_SCHEMA.AREA_MEASURE'
```

```
        IN value_limit_type(SELF\shape_data_quality_criterion.
         assessment_specification.threshold));
    WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.AREA_MEASURE');
    WR3 : validate_inspected_elements_type(SELF,
        ['GEOMETRY_SCHEMA.SURFACE']);
    WR4 : validate_locations_of_extreme_value_type(SELF,
        []);
    WR5 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.AREA_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the area of the surface with the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_upper_value_limit** and the **value_component** of the **threshold** shall be of type **area_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **area_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **surface**.

WR4: The **shape_data_quality_inspection_instance_report** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **area_-measure**.

7.4.53 entirely_narrow_surface

An **entirely_narrow_surface** asserts that a surface is entirely narrow. A surface is narrow if for every point on the surface there is a point on the boundary within half the distance tolerance. The measurement requirement corresponding to this entity requires that the measurement shall calculate for selected points on the surface minimum distance to the boundary. If this exceeds half the distance tolerance, then the surface is not narrow.

NOTE Figure 24 illustrates the measurement of this entity. *d* is the distance from a point on the surface to its boundary.

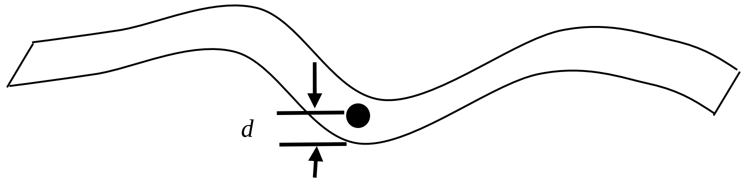

Figure 24 – Measurement of an entirely narrow face

EXPRESS specification:

```
*)
 ENTITY entirely_narrow_surface
   SUBTYPE OF(nearly_degenerate_geometry);
     SELF\shape_data_quality_criterion.assessment_specification :
       shape_data_quality_assessment_by_logical_test;
     width_tolerance  : length_measure;
   WHERE
     WR1 : validate_measured_data_type(SELF,
          'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
     WR2 : validate_inspected_elements_type(SELF,
          ['GEOMETRY_SCHEMA.BOUNDED_SURFACE']);
     WR3 : validate_locations_of_extreme_value_type(SELF,
          []);
     WR4 : validate_accuracy_types(SELF,
          ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests if the surface is entirely narrow, which is the case ’TRUE’, or not.

width_tolerance: the tolerance value to check if the surface is narrow or not.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all the instances associated with this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** of **shape_data_quality_inspection_instance_report_item** corresponding to this entity shall be of type **bounded_surface**.

WR3: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR4: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.54 geometry_with_local_near_degeneracy

A **geometry_with_local_near_degeneracy** is a type of **inapt_geometry** that is a supertype of criteria for detecting geometry including locally infinitesimal characteristics.

EXPRESS specification:

```
*)
 ENTITY geometry_with_local_near_degeneracy
  ABSTRACT SUPERTYPE OF (ONEOF(
   curve_with_small_curvature_radius,
   short_length_curve_segment,
   surface_with_small_curvature_radius,
   small_area_surface_patch,
   narrow_surface_patch,
   indistinct_surface_knots,
   indistinct_curve_knots,
   nearly_degenerate_surface_boundary,
   nearly_degenerate_surface_patch))
  SUBTYPE OF(inapt_geometry);
 END_ENTITY;
(*
```

7.4.55 short_length_curve_segment

A **short_length_curve_segment** asserts that a **b_spline_curve** has a knot interval whose arc length is too short. The measurement requirement corresponding to this entity requires that the measurement shall calculate the minimum arc length of a knot interval of **b_spline_curve** that shall be compared with the given threshold in assessment.

EXPRESS specification:

```
*)
 ENTITY short_length_curve_segment
  SUBTYPE OF(geometry_with_local_near_degeneracy);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT'
        IN TYPEOF(SELF\shape_data_quality_criterion.
         assessment_specification.threshold))
        AND ('MEASURE_SCHEMA.LENGTH_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
```

```
          assessment_specification.threshold));
    WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.LENGTH_MEASURE');
    WR3 : validate_inspected_elements_type(SELF,
        ['GEOMETRY_SCHEMA.B_SPLINE_CURVE']);
    WR4 : validate_locations_of_extreme_value_type(SELF,
        ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'B_SPLINE_CURVE_SEGMENT']);
    WR5 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured arc length against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_upper_value_limit** and the **value_component** of the **threshold** shall be of type **length_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **length_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **b_spline_curve**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **b_spline_curve_-segment**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.56 small_area_surface_patch

A **small_area_surface_patch** asserts that a **b_spline_surface** has too small area patch. The measurement requirement corresponding to this entity requires that the measurement shall calculate minimum area of any patch of the **b_spline_surface** that shall be compared with the given threshold in assessment.

EXPRESS specification:

```
*)
```

```
 ENTITY small_area_surface_patch
  SUBTYPE OF(geometry_with_local_near_degeneracy);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT'
        IN TYPEOF(SELF\shape_data_quality_criterion.
         assessment_specification.threshold))
        AND ('MEASURE_SCHEMA.AREA_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
         assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.AREA_MEASURE');
   WR3 : validate_inspected_elements_type(SELF,
        ['GEOMETRY_SCHEMA.B_SPLINE_SURFACE']);
   WR4 : validate_locations_of_extreme_value_type(SELF,
        ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'B_SPLINE_SURFACE_PATCH']);
   WR5 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.AREA_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured area against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_upper_value_limit** and the **value_component** of the **threshold** shall be of type **area_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **area_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **b_spline_surface**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **b_spline_-surface_patch**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **area_measure**.

7.4.57 narrow_surface_patch

A **narrow_surface_patch** asserts that a **b_spline_surface** has too narrow a dimension for one knot interval. A patch of a surface is narrow if at every point on the patch there is a point on the patch boundary within half the distance tolerance. The measurement requirement corresponding to this entity requires that the measurement shall calculate the distance described above at sufficient points on a patch so that the calculation satisfies the given accuracy.

EXPRESS specification:

```
*)
 ENTITY narrow_surface_patch
  SUBTYPE OF(geometry_with_local_near_degeneracy);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_logical_test;
   width_tolerance  : length_measure;
  WHERE
   WR1 : validate_measured_data_type(SELF,
        'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
   WR2 : validate_inspected_elements_type(SELF,
        ['GEOMETRY_SCHEMA.B_SPLINE_SURFACE']);
   WR3 : validate_locations_of_extreme_value_type(SELF,
        ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'B_SPLINE_SURFACE_PATCH']);
   WR4 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests if there exists any patch with the width smaller than the given tolerance, which is the case ’TRUE’, or not.

width_tolerance: the tolerance to judge if a patch is narrow or not.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, corresponding to this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be **b_spline_surface**.

WR3: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **b_spline_surface-_patch**.

WR4: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.58 indistinct_curve_knots

An instance of **indistinct_curve_knots** asserts that a **b_spline_curve_with_knots** has a portion where adjacent knots are too close. The measurement requirement corresponding to this entity requires that the measurement shall calculate the minimum difference of adjacent knot values of the **b_spline_-curve_with_knots,** that shall be compared with the given threshold in assessment.

EXPRESS specification:

```
*)
 ENTITY indistinct_curve_knots
  SUBTYPE OF(geometry_with_local_near_degeneracy);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
       'SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT'
       IN TYPEOF(SELF\shape_data_quality_criterion.
        assessment_specification.threshold))
       AND ('MEASURE_SCHEMA.PARAMETER_VALUE'
       IN value_limit_type(SELF\shape_data_quality_criterion.
        assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
       'MEASURE_SCHEMA.PARAMETER_VALUE');
   WR3 : validate_inspected_elements_type(SELF,
       ['GEOMETRY_SCHEMA.B_SPLINE_CURVE_WITH_KNOTS']);
   WR4 : validate_locations_of_extreme_value_type(SELF,
       ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
        'B_SPLINE_CURVE_KNOT_LOCATOR']);
   WR5 : validate_accuracy_types(SELF,
       []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured difference against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality_upper_value_limit** and the **value_component** of the **threshold** shall be of type **parameter_value**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **parameter_value**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_report_item** associated with this entity shall be of type **b_spline_curve_with_knots**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_report_item_with_extreme_instances** associated with this entity shall be of type **b_spline_curve_knot_locator**.

WR5: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_association**.

7.4.59 indistinct_surface_knots

An **indistinct_surface_knots** asserts that a **b_spline_surface** has a portion where adjacent knots are too close. The measurement requirement corresponding to this entity requires that the measurement shall calculate the minimum difference of adjacent knot values of the surface both in *u* and *v* directions that shall be compared with the given threshold in assessment.

<u>EXPRESS specification:</u>

```
*)
 ENTITY indistinct_surface_knots
  SUBTYPE OF(geometry_with_local_near_degeneracy);
   SELF\shape_data_quality_criterion.assessment_specification :
     shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
         'SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT'
         IN TYPEOF(SELF\shape_data_quality_criterion.
          assessment_specification.threshold))
         AND ('MEASURE_SCHEMA.PARAMETER_VALUE'
         IN value_limit_type(SELF\shape_data_quality_criterion.
          assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
         'MEASURE_SCHEMA.PARAMETER_VALUE');
   WR3 : validate_inspected_elements_type(SELF,
         ['GEOMETRY_SCHEMA.B_SPLINE_SURFACE_WITH_KNOTS']);
   WR4 : validate_locations_of_extreme_value_type(SELF,
         ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
```

```
          'B_SPLINE_SURFACE_KNOT_LOCATOR']);
    WR5 : validate_accuracy_types(SELF,
         []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured difference against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_upper_value_limit** and the **value_component** of the **threshold** shall be of type **parameter_value**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **parameter_value**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **b_spline_surface_with_knots**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **b_spline_-surface_knot_locator**.

WR5: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.60 curve_with_small_curvature_radius

A **curve_with_small_curvature_radius** asserts that a curve has a point where curvature radius is smaller than the threshold value. The measurement requirement corresponding to this entity requires that the measurement shall calculate the minimum value of curvature radii at the sufficient number of points on the curve.

NOTE Figure 25 illustrates a typical case to be detected by this criterion.

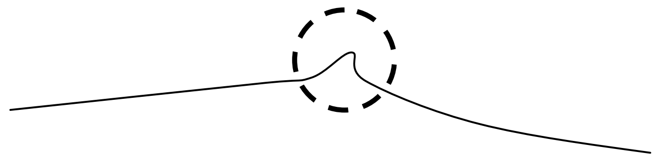

Figure 25 – Curve with small curvature radius

EXPRESS specification:

```
*)
 ENTITY curve_with_small_curvature_radius
  SUBTYPE OF(geometry_with_local_near_degeneracy);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT'
        IN TYPEOF(SELF\shape_data_quality_criterion.
         assessment_specification.threshold))
        AND ('MEASURE_SCHEMA.LENGTH_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
         assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.LENGTH_MEASURE');
   WR3 : validate_inspected_elements_type(SELF,
        ['GEOMETRY_SCHEMA.CURVE']);
   WR4 : validate_locations_of_extreme_value_type(SELF,
        ['GEOMETRY_SCHEMA.POINT_ON_CURVE']);
   WR5 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured curvature radius against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_upper_value_limit** and the **value_component** of the **threshold** shall be of type **length_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **length_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **curve**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **point_on_curve**. It indicates that the curvature radius at that point is smaller than the given tolerance.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.61 surface_with_small_curvature_radius

A **surface_with_small_curvature_radius** asserts that a surface has a point at which minimum principal curvature radius is smaller than the threshold value. The measurement requirement corresponding to this entity requires that the measurement shall calculate the minimum curvature radius at the sufficient number of points on the surface.

<u>EXPRESS specification:</u>

```
*)
 ENTITY surface_with_small_curvature_radius
  SUBTYPE OF(geometry_with_local_near_degeneracy);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
       'SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT'
       IN TYPEOF(SELF\shape_data_quality_criterion.
        assessment_specification.threshold))
       AND ('MEASURE_SCHEMA.LENGTH_MEASURE'
       IN value_limit_type(SELF\shape_data_quality_criterion.
        assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
       'MEASURE_SCHEMA.LENGTH_MEASURE');
   WR3 : validate_inspected_elements_type(SELF,
       ['GEOMETRY_SCHEMA.SURFACE']);
   WR4 : validate_locations_of_extreme_value_type(SELF,
       ['GEOMETRY_SCHEMA.POINT_ON_SURFACE']);
   WR5 : validate_accuracy_types(SELF,
       ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured curvature radius against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_upper_value_limit** and the **value_component** of the **threshold** shall be of type **length_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **length_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **surface**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **point_on_surface**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.62 nearly_degenerate_surface_boundary

A **nearly_degenerate_surface_boundary** asserts that a **b_spline_surface** or a **rectangular_-composite_surface** has one or more partially degenerate boundary curves.

NOTE 1 See IP2 of ISO 10303-42:2003,4.4.68 for reference.

The measurement requirement corresponding to this entity requires that the measurement shall calculate curve lengths of the all the boundary curves of the surface for comparing them with the given threshold in assessment.

NOTE 2 The surface in Figure 26 shall be detected by this criterion when the boundary curve length "*l*" is larger than the lower limit, and smaller than the upper limit of the threshold. Completely degenerated surface boundary is allowed in most CAD systems and is very often treated appropriately. The lower limit value is used for excluding completely degenerated surface boundary case.

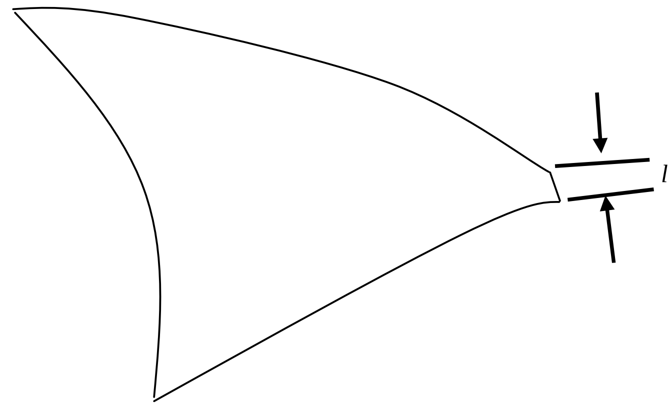

Figure 26 – Nearly degenerate surface boundary

EXPRESS specification:

```
*)
 ENTITY nearly_degenerate_surface_boundary
   SUBTYPE OF(geometry_with_local_near_degeneracy);
    SELF\shape_data_quality_criterion.assessment_specification :
      shape_data_quality_assessment_by_numerical_test;
   WHERE
    WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
          'SHAPE_DATA_QUALITY_VALUE_RANGE'
          IN TYPEOF(SELF\shape_data_quality_criterion.
            assessment_specification.threshold))
          AND ('MEASURE_SCHEMA.LENGTH_MEASURE'
          IN value_limit_type(SELF\shape_data_quality_criterion.
            assessment_specification.threshold));
    WR2 : validate_measured_data_type(SELF,
          'MEASURE_SCHEMA.LENGTH_MEASURE');
    WR3 : validate_inspected_elements_type(SELF,
          ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
           'B_SPLINE_OR_RECTANGULAR_COMPOSITE_SURFACE_SELECT']);
    WR4 : validate_locations_of_extreme_value_type(SELF,
          ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
           'BOUNDARY_CURVE_OF_B_SPLINE_OR_RECTANGULAR_COMPOSITE_SURFACE']);
    WR5 : validate_accuracy_types(SELF,
          ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_by_numerical_test** that tests the measured length against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality_value_range** and the **value_component** of the **threshold** shall be of type **length_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **length_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_report_item** associated with this entity shall be of type **b_spline_or_rectangular_composite_surface_select**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_report_item_with_extreme_instances** associated with this entity shall be of type **boundary_curve_of_b_spline_or_rectangular_composite_surface**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.63 nearly_degenerate_surface_patch

A **nearly_degenerate_surface_patch** asserts that a **b_spline_surface** has one or more patches which have one or more degenerate boundary curves. Degenerate boundary curves on a natural boundary whose length is less than the lower bound value of the threshold are not the target of this criterion. The measurement requirement corresponding to this entity requires that the measurement shall calculate curve lengths of all the boundary curves of each patch of a **b_spline_surface,** that will be compared with the given threshold in assessment**.**

NOTE In Figure 27, solid lines show natural boundary curves and dashed lines show patch boundary curves of a surface. If "*l*", a patch boundary curve length as shown in Figure 27, is smaller than the upper limit of the threshold, this surface shall be detected by this criterion as having quality defect. As described in NOTE 2 of 7.4.62, the lower limit of the threshold is used to compare only with lengths of the natural boundary curves for excluding completely degenerated natural boundary. A patch boundary curve shall be detected by this criterion even if its length is smaller than the lower limit when it is not on any natural boundary curve. If it is on any natural boundary curve then it shall be detected when its length is larger than the lower limit of the threshold.

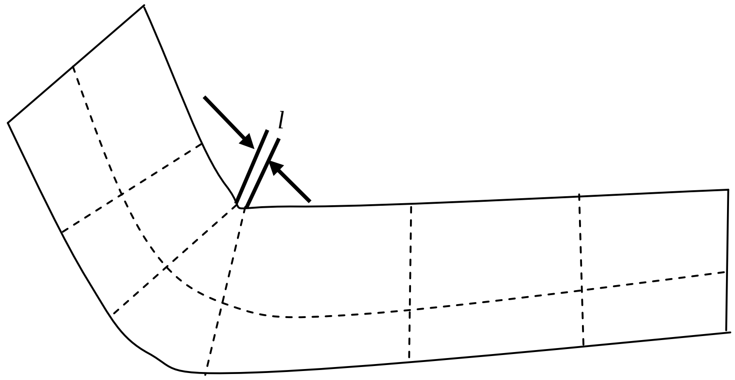

Figure 27 – Nearly degenerate surface patch

EXPRESS specification:

```
*)
 ENTITY nearly_degenerate_surface_patch
  SUBTYPE OF(geometry_with_local_near_degeneracy);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_VALUE_RANGE'
        IN TYPEOF(SELF\shape_data_quality_criterion.
         assessment_specification.threshold))
        AND ('MEASURE_SCHEMA.LENGTH_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
         assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.LENGTH_MEASURE');
   WR3 : validate_inspected_elements_type(SELF,
        ['GEOMETRY_SCHEMA.B_SPLINE_SURFACE']);
   WR4 : validate_locations_of_extreme_value_type(SELF,
        ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'B_SPLINE_SURFACE_PATCH']);
   WR5 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured length against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_value_range** and the **value_component** of the **threshold** shall be of type **length_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **length_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **b_spline_surface**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **b_spline_surface-_patch**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.64 geometry_with_local_irregularity

A **geometry_with_local_irregularity** is a type of **inapt_geometry** that is a supertype of criteria for detecting geometry data that have local inappropriate characteristics.

EXPRESS specification:

```
*)
 ENTITY geometry_with_local_irregularity
  ABSTRACT SUPERTYPE OF (ONEOF(
   extreme_patch_width_variation,
   zero_surface_normal,
   abrupt_change_of_surface_normal))
  SUBTYPE OF(inapt_geometry);
 END_ENTITY;
(*
```

7.4.65 zero_surface_normal

A **zero_surface_normal** asserts that a surface has one or more points on it where two tangent vectors in *u* and *v* directions are parallel within tolerance. This kind of surfaces may cause significant problems in various geometric calculations because magnitude of normal vector is infinitesimal at that point. The measurement requirement corresponding to this entity requires that the measurement shall calculate the angle of tangent vectors in *u* and *v* directions at any point on the surface and shall find out the case when it is close to zero within the given tolerance or it is close to 180 degree within the given tolerance.

NOTE Figure 28 is the case where angle between *u* and *v* vectors is close to 180 degree, and Figure 29 is the case where angle between *u* and *v* vectors is close to zero degree. Both cases shall be detected by this criterion.

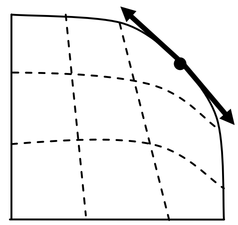

Figure 28 – Angle between *u* and *v* vectors is 180 degree

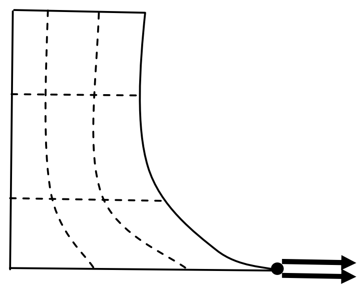

Figure 29 – Angle between *u* and *v* vectors is zero degree

EXPRESS specification:

```
*)
 ENTITY zero_surface_normal
  SUBTYPE OF(geometry_with_local_irregularity);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
       'SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT'
       IN TYPEOF(SELF\shape_data_quality_criterion.
        assessment_specification.threshold))
       AND ('MEASURE_SCHEMA.PLANE_ANGLE_MEASURE'
       IN value_limit_type(SELF\shape_data_quality_criterion.
        assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
       'MEASURE_SCHEMA.PLANE_ANGLE_MEASURE');
   WR3 : validate_inspected_elements_type(SELF,
       ['GEOMETRY_SCHEMA.SURFACE']);
   WR4 : validate_locations_of_extreme_value_type(SELF,
       ['GEOMETRY_SCHEMA.POINT_ON_SURFACE']);
   WR5 : validate_accuracy_types(SELF,
       ['MEASURE_SCHEMA.PLANE_ANGLE_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured angle against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_upper_value_limit** and the **value_component** of the **threshold** shall be of type **plane_angle_-measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **plane_angle_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **surface**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **point_on_surface**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **plane_angle_measure**.

7.4.66 abrupt_change_of_surface_normal

An **abrupt_change_of_surface_normal** asserts that a **surface** has a portion where normals at two nearby points on the **surface** abruptly change. The measurement requirement corresponding to this entity requires that the measurement shall calculate the scalar product of normal vectors at two nearby points on the surface. If a pair of points is found where the scalar product value is negative, then the surface shape is judged to abruptly change there. A normal vector shall be ignored if magnitude of either of the tangent vectors is smaller than the tolerance.

NOTE Figure 30 shows a typical case to be detected by this criterion.

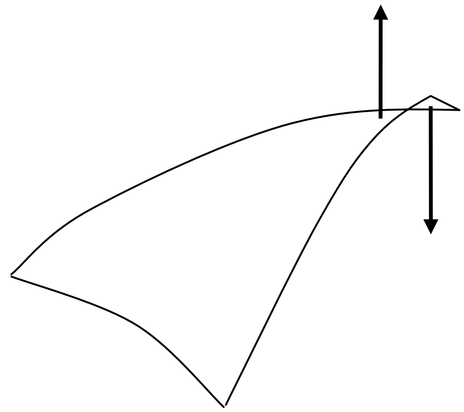

Figure 30 – Abrupt change of surface normals

EXPRESS specification:

```
*)
 ENTITY abrupt_change_of_surface_normal
  SUBTYPE OF(geometry_with_local_irregularity);
   SELF\shape_data_quality_criterion.assessment_specification :
     shape_data_quality_assessment_by_logical_test;
   small_vector_tolerance  : length_measure;
   test_point_distance_tolerance : length_measure;
  WHERE
   WR1 : validate_measured_data_type(SELF,
        'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
   WR2 : validate_inspected_elements_type(SELF,
        ['GEOMETRY_SCHEMA.SURFACE']);
   WR3 : validate_locations_of_extreme_value_type(SELF,
        ['GEOMETRY_SCHEMA.POINT_ON_SURFACE',
         'GEOMETRY_SCHEMA.POINT_ON_SURFACE']);
   WR4 : validate_accuracy_types(SELF,
        []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_by_logical_test** that judges satisfaction or dissatisfaction of this criterion by the sign of the scalar product of normal vectors at two nearby points on the surface.

small_vector_tolerance: **length_measure.** The normal vector shall be ignored if the magnitude of either of the surface derivatives with respect to *u* and *v* at the points under investigation is smaller than this value.

nearby_factor: Distance to determine nearby points. A pair of points is regarded as a target of the check if distance between them is smaller than this value.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, corresponding to this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** of **shape_data_quality_inspection_instance_report_item** corresponding to this entity shall be of type **surface**.

WR3: The **locations_of_extreme_value** of all the **extreme_instance**s in the **instance_report_item_-with_extreme_instances** corresponding to this entity shall be two **point_on_surface**s.

WR4: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.67 extreme_patch_width_variation

An **extreme_patch_width_variation** asserts that widths of the interval of knots of a **b_spline_- surface** are not balanced but significantly varies. This is an inappropriate case for surface design and may lead to errors in the processes such as modification of the surface, projection of a point or a curve onto the surface. The measurement requirement corresponding to this entity requires that the measurement shall calculate ratio of length of patch boundaries of two adjacent patches in all possible parametric direction combinations so that it can be compared with the given threshold in assessment

NOTE In Figure 31, dashed lines show patch boundary curves and solid lines show natural boundary curves of a surface. *l1* and *l2* are one pair of patch boundary curve length whose ratio shall be calculated in the measurement of this criterion.

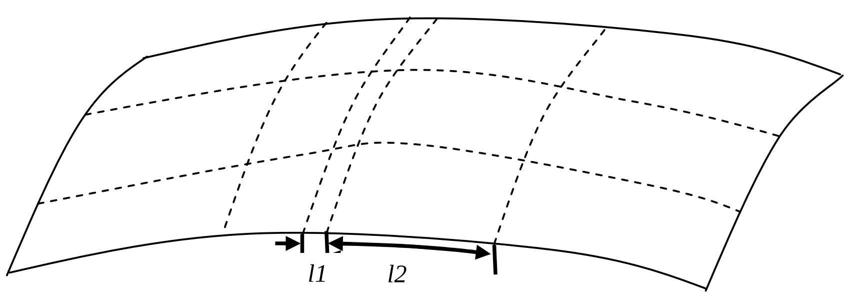

Figure 31 – Extreme patch width variation

EXPRESS specification:

```
*)
 ENTITY extreme_patch_width_variation
  SUBTYPE OF(geometry_with_local_irregularity);
   SELF\shape_data_quality_criterion.assessment_specification :
     shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
        IN TYPEOF(SELF\shape_data_quality_criterion.
         assessment_specification.threshold))
        AND ('MEASURE_SCHEMA.RATIO_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
         assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.RATIO_MEASURE');
   WR3 : validate_inspected_elements_type(SELF,
        ['GEOMETRY_SCHEMA.B_SPLINE_SURFACE']);
   WR4 : validate_locations_of_extreme_value_type(SELF,
        ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'B_SPLINE_SURFACE_KNOT_LOCATOR']);
   WR5 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured change ratio against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **ratio_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **ratio_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **b_spline_surface**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **b_spline_surface_-knot_locator**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.68 multiply_defined_geometry

A **multiply_defined_geometry** is a type of **inapt_geometry** that is a supertype of criteria for detecting multiply defined geometry data.

EXPRESS specification:

```
*)
 ENTITY multiply_defined_geometry
  ABSTRACT SUPERTYPE OF (ONEOF(
   multiply_defined_curves,
   multiply_defined_surfaces,
   multiply_defined_cartesian_points,
   multiply_defined_directions,
   multiply_defined_placements))
  SUBTYPE OF(inapt_geometry);
 END_ENTITY;
(*
```

7.4.69 multiply_defined_cartesian_points

A **multiply_defined_cartesian_points** asserts that two points are coincident. The measurement requirement corresponding to this entity requires that the measurement shall detect the case when the distance between two points under inspection is less than the threshold.

EXPRESS specification:

```
*)
 ENTITY multiply_defined_cartesian_points
  SUBTYPE OF(multiply_defined_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
       'SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT'
       IN TYPEOF(SELF\shape_data_quality_criterion.
        assessment_specification.threshold))
       AND ('MEASURE_SCHEMA.LENGTH_MEASURE'
       IN value_limit_type(SELF\shape_data_quality_criterion.
        assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
       'MEASURE_SCHEMA.LENGTH_MEASURE');
   WR3 : validate_inspected_elements_type(SELF,
       ['GEOMETRY_SCHEMA.CARTESIAN_POINT',
        'GEOMETRY_SCHEMA.CARTESIAN_POINT']);
```

```
    WR4 : validate_locations_of_extreme_value_type(SELF,
        []);
    WR5 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.LENGTH_MEASURE']);
  END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured distance against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_upper_value_limit** and the **value_component** of the **threshold** shall be of type **length_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **length_measure**.

WR3: The **inspected_elements** for instances of **shape_data_quality_inspection_instance_report_-item** associated with this entity shall be two **cartesian_point**s.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.70 multiply_defined_directions

A **multiply_defined_directions** asserts that two directions are identical. The measurement requirement corresponding to this entity requires that the measurement shall detect the case when the angle between two directions under inspection is less than the threshold.

EXPRESS specification:

```
*)
  ENTITY multiply_defined_directions
    SUBTYPE OF(multiply_defined_geometry);
      SELF\shape_data_quality_criterion.assessment_specification :
        shape_data_quality_assessment_by_numerical_test;
    WHERE
      WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
          'SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT'
          IN TYPEOF(SELF\shape_data_quality_criterion.
           assessment_specification.threshold))
```



```
        AND ('MEASURE_SCHEMA.PLANE_ANGLE_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
         assessment_specification.threshold));
    WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.PLANE_ANGLE_MEASURE');
    WR3 : validate_inspected_elements_type(SELF,
        ['GEOMETRY_SCHEMA.DIRECTION','GEOMETRY_SCHEMA.DIRECTION']);
    WR4 : validate_locations_of_extreme_value_type(SELF,
        []);
    WR5 : validate_accuracy_types(SELF,
        []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured angle against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_upper_value_limit** and the **value_component** of the **threshold** shall be of type **plane_angle_-measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **plane_angle_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be two **direction**s.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR5: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.71 multiply_defined_placements

A **multiply_defined_placements** asserts that two placements are identical. The measurement requirement corresponding to this entity requires that the measurement shall detect the case when the distance between locations of the two placements under inspection is less than the threshold given by **assessment_specification**, and the angle between two axes of the placements is less than the threshold given by **assessment_specification_2**.

NOTE If the inspected **placement**s are of type **axis2_placement_2d** or **axis2_placement_3d**, the angle between two **ref_direction**s of the **placement**s shall be also compared with the threshold given by **assessment_specification_2**.

EXPRESS specification:

```
*)
 ENTITY multiply_defined_placements
  SUBTYPE OF(multiply_defined_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
   assessment_specification_2  :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
         'SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT'
         IN TYPEOF(SELF\shape_data_quality_criterion.
          assessment_specification.threshold))
         AND ('MEASURE_SCHEMA.LENGTH_MEASURE'
         IN value_limit_type(SELF\shape_data_quality_criterion.
          assessment_specification.threshold));
   WR2 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
         'SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT'
         IN TYPEOF(SELF.assessment_specification_2.threshold))
         AND ('MEASURE_SCHEMA.PLANE_ANGLE_MEASURE'
         IN value_limit_type(SELF.assessment_specification_2.threshold));
   WR3 : validate_measured_data_type(SELF,
         'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'PLANE_ANGLE_AND_LENGTH_PAIR');
   WR4 : validate_inspected_elements_type(SELF,
         ['GEOMETRY_SCHEMA.PLACEMENT','GEOMETRY_SCHEMA.PLACEMENT']);
   WR5 : validate_locations_of_extreme_value_type(SELF,
         []);
   WR6 : validate_accuracy_types(SELF,
         []);
   WR7 : SIZEOF(QUERY(sdqr <* bag_to_set(USEDIN(SELF,
          'PRODUCT_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
          'DATA_QUALITY_INSPECTION_RESULT.'+
          'CRITERION_INSPECTED'))|
          ('SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
           'SHAPE_DATA_QUALITY_INSPECTION_RESULT'
           IN TYPEOF(sdqr)) AND
           (SIZEOF(QUERY(sdqiir <* bag_to_set(USEDIN(sdqr,
             'PRODUCT_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
             'DATA_QUALITY_INSPECTION_REPORT.'+
             'INSPECTION_RESULT'))|
             ('SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
              'SHAPE_DATA_QUALITY_INSPECTION_INSTANCE_REPORT'
```

```
             IN TYPEOF(sdqiir)) AND
            (SIZEOF(QUERY(sdqiri <* sdqiir.inspected_instances|
                  (TYPEOF(sdqiri.inspected_elements[1])
                   <>TYPEOF(sdqiri.inspected_elements[2])))) = 0)
          )) = 0)
        )) = 0;
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_by_numerical_test** that tests the measured distance against the given threshold.

assessment_specification_2: **shape_data_quality_assessment_by_numerical_test** that tests the measured angle against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality_upper_value_limit** and the **value_component** of the **threshold** shall be of type **length_measure**.

WR2: The **threshold** of **assessment_specification_2** of this entity shall be of type **shape_data_quality_upper_value_limit** and the **value_component** of the **threshold** shall be of type **plane_angle_measure**.

WR3: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **plane_angle_and_length_pair**.

WR4: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_report_item** associated with this entity shall be two **placement**s.

WR5: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR6: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_association**.

WR7: The two instances of **placement**s which are the **inspected_elements** for all instances of **shape_data_quality_inspection_instance_report_item** associated with this entity shall be of the same subtype of **placement**.

7.4.72 multiply_defined_curves

A **multiply_defined_curves** asserts that two curves are entirely overlapping. Two curves are entirely overlapping when the maximum distance between the curves is smaller than the given threshold value. The measurement requirement corresponding to this entity requires that the measurement shall calculate Hausdorff distance between two curves so that it can be checked if it is less than the given threshold value in assessment.

EXPRESS specification:

```
*)
 ENTITY multiply_defined_curves
  SUBTYPE OF(multiply_defined_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
       'SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT'
       IN TYPEOF(SELF\shape_data_quality_criterion.
        assessment_specification.threshold))
       AND ('MEASURE_SCHEMA.LENGTH_MEASURE'
       IN value_limit_type(SELF\shape_data_quality_criterion.
        assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
       'MEASURE_SCHEMA.LENGTH_MEASURE');
   WR3 : validate_inspected_elements_type(SELF,
       ['GEOMETRY_SCHEMA.CURVE','GEOMETRY_SCHEMA.CURVE']);
   WR4 : validate_locations_of_extreme_value_type(SELF,
       []);
   WR5 : validate_accuracy_types(SELF,
       ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured distance against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_upper_value_limit** and the **value_component** of the **threshold** shall be of type **length_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **length_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be two **curve**s.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.73 multiply_defined_surfaces

A **multiply_defined_surfaces** asserts that two surfaces are entirely overlapping. Two surfaces are entirely overlapping when the maximum distance between the surfaces is smaller than the threshold value. The measurement requirement corresponding to this entity requires that the measurement shall calculate Hausdorff distance between two surfaces so that it can be checked if it is less than the given threshold value in assessment.

<u>EXPRESS specification:</u>

```
*)
 ENTITY multiply_defined_surfaces
  SUBTYPE OF(multiply_defined_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT'
        IN TYPEOF(SELF\shape_data_quality_criterion.
         assessment_specification.threshold)) AND
        ('MEASURE_SCHEMA.LENGTH_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
         assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.LENGTH_MEASURE');
   WR3 : validate_inspected_elements_type(SELF,
        ['GEOMETRY_SCHEMA.SURFACE','GEOMETRY_SCHEMA.SURFACE']);
   WR4 : validate_locations_of_extreme_value_type(SELF,
        []);
   WR5 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

<u>Attribute definitions:</u>

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured distance against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_upper_value_limit** and the **value_component** of the **threshold** shall be of type **length_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item,** for all instances associated with this entity shall be of type **length_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be two **surface**s.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.74 overlapping_geometry

An **overlapping_geometry** is a type of **inapt_geometry** that is a supertype of criteria for detecting partly overlapping geometric entities.

EXPRESS specification:

```
*)
 ENTITY overlapping_geometry
  ABSTRACT SUPERTYPE OF (ONEOF(
   partly_overlapping_curves, partly_overlapping_surfaces))
  SUBTYPE OF(inapt_geometry);
 END_ENTITY;
(*
```

7.4.75 partly_overlapping_curves

A **partly_overlapping_curves** asserts that two curves are partly overlapping. Two curves overlap if each curve contains an arcwise connected set of points such that every point in the set for one curve is within **interference_tolerance** from the other curve. Arc length of the overlapping portion shall be evaluated for each curve and the longer one is required to be greater than or equal to the minimum of the specified threshold and the shorter arc length of the two curves under inspection.

NOTE In Figure 32, *l* shows the length of overlapping portion of two curves and *t* shows the threshold length. This case shall be detected by this criterion because *l* is greater than *t*. Figure 33 shows a case where a curve shorter than the threshold entirely lies on another curve within **interference_tolerance**. This case shall also be detected by this criterion because length of overlapping portion is equal to the arc length of the shorter curve.

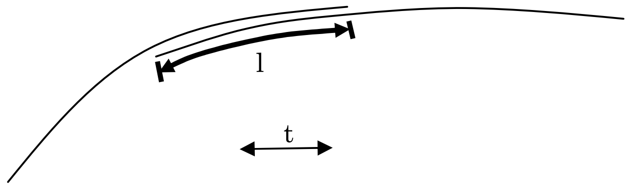

Figure 32 – Two curves whose length of overlapping portion is greater than the threshold

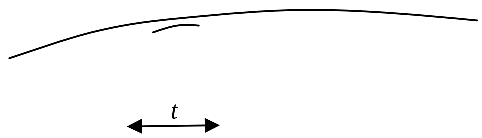

Figure 33 – A short curve lying on another curve

EXPRESS specification:

```
*)
 ENTITY partly_overlapping_curves
  SUBTYPE OF(overlapping_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
   interference_tolerance : length_measure;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
         'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
         IN TYPEOF(SELF\shape_data_quality_criterion.
          assessment_specification.threshold))
         AND ('MEASURE_SCHEMA.LENGTH_MEASURE'
         IN value_limit_type(SELF\shape_data_quality_criterion.
          assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
         'MEASURE_SCHEMA.LENGTH_MEASURE');
   WR3 : validate_inspected_elements_type(SELF,
         ['GEOMETRY_SCHEMA.CURVE','GEOMETRY_SCHEMA.CURVE']);
   WR4 : validate_locations_of_extreme_value_type(SELF,
         ['GEOMETRY_SCHEMA.TRIMMED_CURVE',
          'GEOMETRY_SCHEMA.TRIMMED_CURVE']);
   WR5 : validate_accuracy_types(SELF,
```

```
        ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured arc length against the given threshold.

interference_tolerance: tolerance to check if two points on the curves are coincident or not.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **length_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **length_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be two **curve**s.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be two **trimmed_curve**s. It implies that these are portions of the curves which are identical.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.76 partly_overlapping_surfaces

A **partly_overlapping_surfaces** asserts that two surfaces are partly overlapping. Two surfaces overlap if each surface contains an arcwise connected set of points such that every point in the set for one surface is within **interference_tolerance** from the other surface. Area of the overlapping portion shall be evaluated for each surface and the larger one is required to be greater than or equal to the minimum of the specified threshold and the smaller surface area of the two surfaces under inspection.

EXPRESS specification:

```
*)
 ENTITY partly_overlapping_surfaces
  SUBTYPE OF(overlapping_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
   interference_tolerance : length_measure;
  WHERE
```

```
    WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
        IN TYPEOF(SELF\shape_data_quality_criterion.
         assessment_specification.threshold))
        AND ('MEASURE_SCHEMA.AREA_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
         assessment_specification.threshold));
    WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.AREA_MEASURE');
    WR3 : validate_inspected_elements_type(SELF,
        ['GEOMETRY_SCHEMA.SURFACE','GEOMETRY_SCHEMA.SURFACE']);
    WR4 : validate_locations_of_extreme_value_type(SELF,
        ['GEOMETRY_SCHEMA.CURVE_BOUNDED_SURFACE',
         'GEOMETRY_SCHEMA.CURVE_BOUNDED_SURFACE']);
    WR5 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.AREA_MEASURE']);
 END_ENTITY;
(*
```

<u>Attribute definitions:</u>

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured area against the given threshold.

interference_tolerance: The tolerance to check if two points on the surfaces are coincident or not.

<u>Formal propositions:</u>

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **area_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **area_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be two **surface**s.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be two **curve_-bounded_surface**s. It implies that these are portions of the surfaces which are identical.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **area_measure**.

7.4.77 overcomplex_geometry

An **overcomplex_geometry** is a type of **inapt_geometry** and is a supertype of criteria for detecting overcomplex geometry data.

EXPRESS specification:

```
*)
 ENTITY overcomplex_geometry
   ABSTRACT SUPERTYPE OF (ONEOF(
     excessively_high_degree_curve,
     excessively_high_degree_surface,
     surface_with_excessive_patches_in_one_direction,
     curve_with_excessive_segments,
     high_degree_linear_curve,
     high_degree_planar_surface,
     high_degree_conic,
     high_degree_axi_symmetric_surface))
   SUBTYPE OF(inapt_geometry);
 END_ENTITY;
(*
```

7.4.78 excessively_high_degree_curve

An **excessively_high_degree_curve** asserts that a **b_spline_curve** has too high degree. The measurement requirement corresponding to this entity requires that the measurement shall check degree of the curve so that it can be compared with the given threshold in assessment.

EXPRESS specification:

```
*)
 ENTITY excessively_high_degree_curve
   SUBTYPE OF(overcomplex_geometry);
     SELF\shape_data_quality_criterion.assessment_specification :
       shape_data_quality_assessment_by_numerical_test;
   WHERE
     WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
         'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
         IN TYPEOF(SELF\shape_data_quality_criterion.
           assessment_specification.threshold))
         AND ('MEASURE_SCHEMA.COUNT_MEASURE'
         IN value_limit_type(SELF\shape_data_quality_criterion.
           assessment_specification.threshold));
     WR2 : validate_measured_data_type(SELF,
         'MEASURE_SCHEMA.COUNT_MEASURE');
     WR3 : validate_inspected_elements_type(SELF,
         ['GEOMETRY_SCHEMA.B_SPLINE_CURVE']);
```

```
    WR4 : validate_locations_of_extreme_value_type(SELF,
        []);
    WR5 : validate_accuracy_types(SELF,
        []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the degree obtained against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **count_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **count_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **b_spline_curve**.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR5: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.79 excessively_high_degree_surface

An **excessively_high_degree_surface** asserts that a **b_spline_surface** has too high degree. The measurement requirement corresponding to this entity requires that the measurement shall check degree of the surface so that it can be compared with the given threshold in assessment.

EXPRESS specification:

```
*)
 ENTITY excessively_high_degree_surface
  SUBTYPE OF(overcomplex_geometry);
    SELF\shape_data_quality_criterion.assessment_specification :
      shape_data_quality_assessment_by_numerical_test;
  WHERE
    WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
```

```
        IN TYPEOF(SELF\shape_data_quality_criterion.
         assessment_specification.threshold))
        AND ('MEASURE_SCHEMA.COUNT_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
         assessment_specification.threshold));
    WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.COUNT_MEASURE');
    WR3 : validate_inspected_elements_type(SELF,
        ['GEOMETRY_SCHEMA.B_SPLINE_SURFACE']);
    WR4 : validate_locations_of_extreme_value_type(SELF,
        []);
    WR5 : validate_accuracy_types(SELF,
        []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the degree obtained against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **count_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **count_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **b_spline_surface**.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR5: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.80 curve_with_excessive_segments

A **curve_with_excessive_segments** asserts that a **b_spline_curve** or a **composite_curve** has too many segments. The measurement requirement corresponding to this entity requires calculation of the number of segments. For a **b_spline_curve**, number of segments is calculated from the number of control points and the degree by the following formula:

(number of segments) = SIZEOF (**control_points_list**) **-** (**degree**).

For a **composite_curve**, the number of segments is the value of its **n_segments** attribute.

EXPRESS specification:

```
*)
 ENTITY curve_with_excessive_segments
  SUBTYPE OF(overcomplex_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
       'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
       IN TYPEOF(SELF\shape_data_quality_criterion.
        assessment_specification.threshold))
       AND ('MEASURE_SCHEMA.COUNT_MEASURE'
       IN value_limit_type(SELF\shape_data_quality_criterion.
        assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
       'MEASURE_SCHEMA.COUNT_MEASURE');
   WR3 : validate_inspected_elements_type(SELF,
       ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
        'B_SPLINE_OR_COMPOSITE_CURVE_SELECT']);
   WR4 : validate_locations_of_extreme_value_type(SELF,
       []);
   WR5 : validate_accuracy_types(SELF,
       []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the number of segments to the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **count_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **count_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance-_report_item** associated with this entity shall be of type **b_spline_or_composite_curve_select**.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR5: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.81 surface_with_excessive_patches_in_one_direction

A **surface_with_excessive_patches_in_one_direction** asserts that a **b_spline_surface** or a **rectangular_composite_surface** has too many patches in either *u* or *v* direction. The measurement requirement corresponding to this entity requires that the measurement shall calculate the maximum number of patches in either parametric direction. For a **b_spline_surface**, the maximum number of patches is calculated using the following formula:

(maximum number of patches in u or v direction)

= max((**u_upper - u_degree**), (**v_upper - v_degree**))

For a **rectangular_composite_surface**, the maximum number of patches is the larger value of its **n_u** and **n_v** attributes.

<u>EXPRESS specification:</u>

```
*)
 ENTITY surface_with_excessive_patches_in_one_direction
   SUBTYPE OF(overcomplex_geometry);
     SELF\shape_data_quality_criterion.assessment_specification :
       shape_data_quality_assessment_by_numerical_test;
   WHERE
     WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
          'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
          IN TYPEOF(SELF\shape_data_quality_criterion.
           assessment_specification.threshold))
          AND ('MEASURE_SCHEMA.COUNT_MEASURE'
          IN value_limit_type(SELF\shape_data_quality_criterion.
           assessment_specification.threshold));
     WR2 : validate_measured_data_type(SELF,
          'MEASURE_SCHEMA.COUNT_MEASURE');
     WR3 : validate_inspected_elements_type(SELF,
          ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
           'B_SPLINE_OR_RECTANGULAR_COMPOSITE_SURFACE_SELECT']);
     WR4 : validate_locations_of_extreme_value_type(SELF,
          []);
     WR5 : validate_accuracy_types(SELF,
          []);
 END_ENTITY;
(*
```

<u>Attribute definitions:</u>

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the number obtained against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **count_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **count_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **b_spline_or_rectangular_composite_-surface_select**.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR5: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.82 high_degree_linear_curve

A **high_degree_linear_curve** asserts that a **b_spline_curve** is a linear geometry but has high degree. The measurement requirement corresponding to this entity requires that the measurement shall check if the **b_spline_curve** is almost linear within **approximation_tolerance**, and if it has degree higher than the threshold.

EXPRESS specification:

```
*)
 ENTITY high_degree_linear_curve
  SUBTYPE OF(overcomplex_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
   approximation_tolerance : length_measure;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
         'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
         IN TYPEOF(SELF\shape_data_quality_criterion.
          assessment_specification.threshold))
         AND ('MEASURE_SCHEMA.COUNT_MEASURE'
         IN value_limit_type(SELF\shape_data_quality_criterion.
          assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
```

```
         'MEASURE_SCHEMA.COUNT_MEASURE');
    WR3 : validate_inspected_elements_type(SELF,
         ['GEOMETRY_SCHEMA.B_SPLINE_CURVE']);
    WR4 : validate_locations_of_extreme_value_type(SELF,
         []);
    WR5 : validate_accuracy_types(SELF,
         ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the degree obtained against the given threshold.

approximation_tolerance: the tolerance to check if the **b_spline_curve** geometry can be approximated to a line.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **count_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **count_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **b_spline_curve**.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.83 high_degree_conic

A **high_degree_conic** asserts that a **b_spline_curve** is a conic but has high degree. The measurement requirement corresponding to this entity requires that the measurement shall check if the **b_spline_-curve** can be approximated by a conic curve within **approximation_tolerance**, and if it has degree higher than the threshold.

EXPRESS specification:

```
*)
 ENTITY high_degree_conic
   SUBTYPE OF(overcomplex_geometry);
```

```
    SELF\shape_data_quality_criterion.assessment_specification :
      shape_data_quality_assessment_by_numerical_test;
    approximation_tolerance : length_measure;
  WHERE
    WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
        IN TYPEOF(SELF\shape_data_quality_criterion.
          assessment_specification.threshold))
        AND ('MEASURE_SCHEMA.COUNT_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
          assessment_specification.threshold));
    WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.COUNT_MEASURE');
    WR3 : validate_inspected_elements_type(SELF,
        ['GEOMETRY_SCHEMA.B_SPLINE_CURVE']);
    WR4 : validate_locations_of_extreme_value_type(SELF,
        []);
    WR5 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the degree obtained against the given threshold.

approximation_tolerance: The tolerance to check if the **b_spline_curve** geometry can be approximated as a conic.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **count_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **count_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **b_spline_curve**.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.84 high_degree_planar_surface

A **high_degree_planar_surface** asserts that a **b_spline_surface** is planar geometry but has high degree. The measurement requirement corresponding to this entity requires that the measurement shall check if the **b_spline_surface** is almost planar within **approximation_tolerance**, and if it has degree higher than the threshold.

EXPRESS specification:

```
*)
 ENTITY high_degree_planar_surface
  SUBTYPE OF(overcomplex_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
     shape_data_quality_assessment_by_numerical_test;
   approximation_tolerance : length_measure;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
        IN TYPEOF(SELF\shape_data_quality_criterion.
         assessment_specification.threshold))
        AND ('MEASURE_SCHEMA.COUNT_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
         assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.COUNT_MEASURE');
   WR3 : validate_inspected_elements_type(SELF,
        ['GEOMETRY_SCHEMA.B_SPLINE_SURFACE']);
   WR4 : validate_locations_of_extreme_value_type(SELF,
        []);
   WR5 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the degree obtained against the given threshold.

approximation_tolerance: the tolerance to check if the **b_spline_surface** geometry can be approximated as a plane.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **count_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **count_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **b_spline_surface**.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.85 high_degree_axi_symmetric_surface

A **high_degree_axi_symmetric_surface** asserts that a **b_spline_surface** is a type of axi-symmetric surface but has high degree. The measurement requirement corresponding to this entity requires that the measurement shall check if the **b_spline_surface** can be approximated as an axi-symmetric surface within **approximation_tolerance**, and if it has degree higher than the threshold.

<u>EXPRESS specification:</u>

```
*)
 ENTITY high_degree_axi_symmetric_surface
  SUBTYPE OF(overcomplex_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
   approximation_tolerance : length_measure;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
       'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
       IN TYPEOF(SELF\shape_data_quality_criterion.
        assessment_specification.threshold))
       AND ('MEASURE_SCHEMA.COUNT_MEASURE'
       IN value_limit_type(SELF\shape_data_quality_criterion.
        assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
       'MEASURE_SCHEMA.COUNT_MEASURE');
   WR3 : validate_inspected_elements_type(SELF,
       ['GEOMETRY_SCHEMA.B_SPLINE_SURFACE']);
   WR4 : validate_locations_of_extreme_value_type(SELF,
       []);
   WR5 : validate_accuracy_types(SELF,
       ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_by_numerical_test** that tests the degree obtained against the given threshold.

approximation_tolerance: The tolerance to check if the **b_spline_surface** geometry can be approximated as an axi-symmetric surface.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality_lower_value_limit** and the **value_component** of the **threshold** shall be of type **count_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **count_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_report_item** associated with this entity shall be of type **b_spline_surface**.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.86 inapt_topology_and_geometry_relationship

An **inapt_topology_and_geometry_relationship** is a type of **inapt_data** that is a supertype of criteria for detecting inappropriate relationship between topology and geometry.

EXPRESS specification:

```
*)
 ENTITY inapt_topology_and_geometry_relationship
  ABSTRACT SUPERTYPE OF (ONEOF(
   topology_related_to_nearly_degenerate_geometry,
   geometric_gap_in_topology,
   non_smooth_geometry_transition_across_edge,
   topology_related_to_self_intersecting_geometry,
   steep_geometry_transition_across_edge,
   topology_related_to_overlapping_geometry,
   topology_related_to_multiply_defined_geometry,
   overcomplex_topology_and_geometry_relationship))
  SUBTYPE OF(inapt_data);
 END_ENTITY;
(*
```

7.4.87 topology_related_to_nearly_degenerate_geometry

A **topology_related_to_nearly_degenerate_geometry** is a type of **inapt_topology_and_geometry_-relationship** that is a supertype of criteria for detecting topology related to nearly degenerate geometry of various types.

EXPRESS specification:

```
*)
 ENTITY topology_related_to_nearly_degenerate_geometry
  ABSTRACT SUPERTYPE OF (ONEOF(
   short_length_edge,
   small_area_face,
   entirely_narrow_face))
  SUBTYPE OF(inapt_topology_and_geometry_relationship);
 END_ENTITY;
(*
```

7.4.88 short_length_edge

A **short_length_edge** asserts that arc length of an **edge_curve** is too short. The measurement requirement corresponding to this entity requires the measurement shall calculate arc length of the curve geometry between two points corresponding to the start and the end vertices of the **edge_curve** under inspection.

EXPRESS specification:

```
*)
 ENTITY short_length_edge
  SUBTYPE OF(topology_related_to_nearly_degenerate_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
         'SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT'
         IN TYPEOF(SELF\shape_data_quality_criterion.
          assessment_specification.threshold))
         AND ('MEASURE_SCHEMA.LENGTH_MEASURE'
         IN value_limit_type(SELF\shape_data_quality_criterion.
          assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
         'MEASURE_SCHEMA.LENGTH_MEASURE');
   WR3 : validate_inspected_elements_type(SELF,
         ['TOPOLOGY_SCHEMA.EDGE_CURVE']);
   WR4 : validate_locations_of_extreme_value_type(SELF,
         []);
   WR5 : validate_accuracy_types(SELF,
```

```
        ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured arc length against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_upper_value_limit** and the **value_component** of the **threshold** shall be of type **length_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **length_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **edge_curve**.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.89 small_area_face

A **small_area_face** asserts that area of **face_surface** is too small. The measurement requirement corresponding to this entity requires the measurement shall calculate the area of the **face_surface**.

EXPRESS specification:

```
*)
 ENTITY small_area_face
  SUBTYPE OF(topology_related_to_nearly_degenerate_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
     shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT'
        IN TYPEOF(SELF\shape_data_quality_criterion.
         assessment_specification.threshold))
        AND ('MEASURE_SCHEMA.AREA_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
         assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
```

```
        'MEASURE_SCHEMA.AREA_MEASURE');
    WR3 : validate_inspected_elements_type(SELF,
        ['TOPOLOGY_SCHEMA.FACE_SURFACE']);
    WR4 : validate_locations_of_extreme_value_type(SELF,
        []);
    WR5 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.AREA_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured area against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_upper_value_limit** and the **value_component** of the **threshold** shall be of type **area_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **area_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **face_surface**.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **area_measure**.

7.4.90 entirely_narrow_face

An **entirely_narrow_face** asserts that a **face_surface** is too narrow. A **face_surface** is too narrow if for every point on the **face_surface** there is a point on the boundary within half the distance of the threshold value. The measurement requirement corresponding to this entity requires that the measurement shall calculate minimum distance from selected points on the **face_surface** to the boundary. If calculated value exceeds half the distance tolerance represented by **width_tolerance** attribute, then the **face_surface** is not narrow.

NOTE Figure 34 illustrates the measurement of this criterion. The measurement considers a sphere whose diameter is **width_tolerance** and whose centre is at any point on the **face_surface** and checks if the sphere does not intersect with the boundary of the **face_surface**. The test is complete when one such internal point is detected.

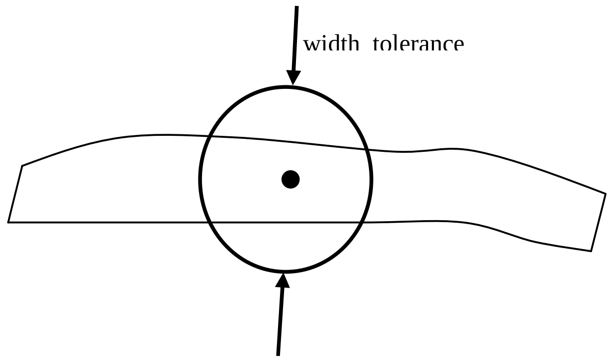

Figure 34 – Measurement of an entirely narrow face

EXPRESS specification:

```
*)
 ENTITY entirely_narrow_face
   SUBTYPE OF(topology_related_to_nearly_degenerate_geometry);
     SELF\shape_data_quality_criterion.assessment_specification :
       shape_data_quality_assessment_by_logical_test;
     width_tolerance : length_measure;
   WHERE
     WR1 : validate_measured_data_type(SELF,
          'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
     WR2 : validate_inspected_elements_type(SELF,
          ['TOPOLOGY_SCHEMA.FACE_SURFACE']);
     WR3 : validate_locations_of_extreme_value_type(SELF,
          []);
     WR4 : validate_accuracy_types(SELF,
          ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests if the face is narrow at all the inspected points, which is the case ’TRUE’, or not.

width_tolerance: the tolerance value used to check if the face is narrow or not.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item** corresponding to this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** of **shape_data_quality_inspection_instance_report_item** corresponding to this entity shall be of type **face_surface**.

WR3: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR4: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.91 topology_related_to_self_intersecting_geometry

A **topology_related_to_self_intersecting_geometry** is a type of **inapt_topology_and_geometry_-relationship** that is a supertype of several criteria for detecting topology related to self-intersecting geometry.

EXPRESS specification:

```
*)
 ENTITY topology_related_to_self_intersecting_geometry
  ABSTRACT SUPERTYPE OF (ONEOF(
   self_intersecting_loop,
   self_intersecting_shell,
   intersecting_connected_face_sets))
  SUBTYPE OF(inapt_topology_and_geometry_relationship);
 END_ENTITY;
(*
```

7.4.92 self_intersecting_loop

A **self_intersecting_loop** asserts that an **edge_loop** is self-intersecting. The measurement requirement corresponding to this entity requires that the measurement shall detect a point on the **edge_loop**, which is close to another point on the same **edge_loop** within **interference_tolerance** but has a sufficiently large arc length distance from that point.

NOTE Figure 35 is the case where **edge_curve**s in an **edge_loop** cross each other. Figure 36 is the case where there is no **edge_curve** that crosses other **edge_curve**s in an **edge_loop**, but there exist some **edge_curve**s that interfere with other **edge_curve**s in the **edge_loop** within **interference_tolernace**. This criterion shall detect both cases.

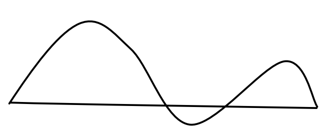

Figure 35 – Edge crosses another edge in an edge loop

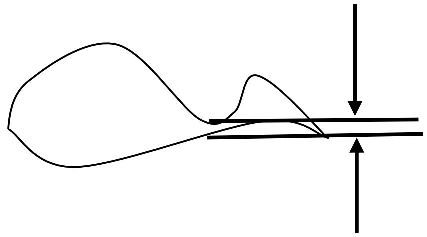

Figure 36 – Edge loop with a region narrower than the tolerance

EXPRESS specification:

```
*)
 ENTITY self_intersecting_loop
  SUBTYPE OF(topology_related_to_self_intersecting_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_logical_test;
   arc_length_distance_factor : REAL;
   interference_tolerance    : length_measure;
  WHERE
   WR1 : validate_measured_data_type(SELF,
        'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
   WR2 : validate_inspected_elements_type(SELF,
        ['TOPOLOGY_SCHEMA.EDGE_LOOP']);
   WR3 : validate_locations_of_extreme_value_type(SELF,
        ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'POINT_ON_EDGE_CURVE',
         'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'POINT_ON_EDGE_CURVE']);
   WR4 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests the existence of intersection, which is the case ’TRUE’, or not

arc_length_separation_factor: Real value for use in the judgement of self intersection. Two coincident points shall be regarded as self-intersecting points if arc length along the **edge_loop** between the points is larger than (**interference_tolerance**) * (**arc_length_separation_factor**).

interference_tolerance: The tolerance used to check if two points are coincident.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, corresponding to this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances **of shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **edge_loop**.

WR3: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be both of type **point_on_-edge_curve**.

NOTE These two **point_on_edge_curve**s are located within **interference_tolerance** but arc length between them on the **edge_loop** is large enough.

WR4: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.93 self_intersecting_shell

A **self_intersecting_shell** asserts that a **connected_face_set** in the form of open or closed shell is self-intersecting. The measurement requirement corresponding to this entity requires that the measurement shall detect points on the **connected_face_set** that are within **interference_tolerance** each other with the distance along the **connected_face_set** between them larger than the value determined by multiplying the tolerance with the specified factor.

EXPRESS specification:

```
*)
 ENTITY self_intersecting_shell
  SUBTYPE OF(topology_related_to_self_intersecting_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_logical_test;
   geodesic_separation_factor : REAL;
   interference_tolerance    : length_measure;
  WHERE
   WR1 : validate_measured_data_type(SELF,
        'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
   WR2 : validate_inspected_elements_type(SELF,
        ['TOPOLOGY_SCHEMA.CONNECTED_FACE_SET']);
   WR3 : validate_locations_of_extreme_value_type(SELF,
        ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'POINT_ON_FACE_SURFACE',
         'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'POINT_ON_FACE_SURFACE']);
   WR4 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.LENGTH_MEASURE']);
```

```
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_by_logical_test** that tests the existence of intersection, which is the case 'TRUE', or not

geodesic_separation_factor: Real value for use in the judgement of self intersection. Two coincident points shall be regarded as self-intersecting point if distance on the **connected_face_set** is larger than the value obtained by the following formulae:

(**interference_tolerance**) * (**geodesic_separation_factor**).

interference_tolerance: The tolerance used to check if two points are coincident.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, corresponding to this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances **of shape_data_quality_inspection_instance_report_item** associated with this entity shall be of type **connected_face_set**.

WR3: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_report_item_with_extreme_instances** associated with this entity shall be both of type **point_on_face_surface**.

NOTE These two **point_on_face_surface**s are located within **interference_tolerance** and their **face_surface**s are included in the **connected_face_set** but the distance along the **connected_face_set** between them is large enough.

WR4: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.94 intersecting_connected_face_sets

An **intersecting_connected_face_sets** asserts that two **connected_face_set**s are intersecting each other. The measurement requirement corresponding to this entity requires that the measurement shall detect the case where two **connected_face_set**s are intersecting each other. If distance between a point on a **connected_face_set** and a point on another **connected_face_set** is smaller than **interference_tolerance**, then those **connected_face_set**s shall be regarded as intersecting.

EXPRESS specification:

```
*)
 ENTITY intersecting_connected_face_sets
```

```
   SUBTYPE OF(topology_related_to_self_intersecting_geometry);
    SELF\shape_data_quality_criterion.assessment_specification :
      shape_data_quality_assessment_by_logical_test;
    interference_tolerance : length_measure;
   WHERE
    WR1 : validate_measured_data_type(SELF,
        'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
    WR2 : validate_inspected_elements_type(SELF,
        ['TOPOLOGY_SCHEMA.CONNECTED_FACE_SET',
         'TOPOLOGY_SCHEMA.CONNECTED_FACE_SET']);
    WR3 : validate_locations_of_extreme_value_type(SELF,
        ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'POINT_ON_FACE_SURFACE',
         'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'POINT_ON_FACE_SURFACE']);
    WR4 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.LENGTH_MEASURE']);
  END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests the existence of intersection, which is the case 'TRUE', or not.

interference_tolerance: The tolerance used to check if two points are coincident.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, corresponding to this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances **of shape_data_quality_inspection_instance_-report_item** associated with this entity shall be both of type **connected_face_set**.

WR3: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_report_item_with_extreme_instances** associated with this entity shall be both of type **point_on_face_surface.**

NOTE These two **point_on_face_surface**s are located within the **interference_tolerance** and their **face_surface**s are included in different **connected_face_set**s.

WR4: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.95 geometric_gap_in_topology

A **geometric_gap_in_topology** is a type of **inapt_topology_and_geometry_relationship** that is a supertype of criteria for detecting various types of gap between geometry and topology.

EXPRESS specification:

```
*)
 ENTITY geometric_gap_in_topology
  ABSTRACT SUPERTYPE OF (ONEOF(
   gap_between_adjacent_edges_in_loop,
   gap_between_vertex_and_base_surface,
   gap_between_vertex_and_edge,
   gap_between_edge_and_base_surface,
   gap_between_pcurves_related_to_an_edge,
   gap_between_faces_related_to_an_edge))
  SUBTYPE OF(inapt_topology_and_geometry_relationship);
 END_ENTITY;
(*
```

7.4.96 gap_between_vertex_and_edge

A **gap_between_vertex_and_edge** asserts that geometry of **vertex_point** does not lie on the related **edge_curve** within specified threshold. The measurement requirement corresponding to this entity requires that the measurement shall calculate distance between geometry of the **vertex_point** and geometry of the **edge_curve**.

EXPRESS specification:

```
*)
 ENTITY gap_between_vertex_and_edge
  SUBTYPE OF(geometric_gap_in_topology);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
        IN TYPEOF(SELF\shape_data_quality_criterion.
         assessment_specification.threshold))
        AND ('MEASURE_SCHEMA.LENGTH_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
         assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.LENGTH_MEASURE');
   WR3 : validate_inspected_elements_type(SELF,
        ['TOPOLOGY_SCHEMA.EDGE_CURVE']);
   WR4 : validate_locations_of_extreme_value_type(SELF,
```

```
        ['TOPOLOGY_SCHEMA.VERTEX_POINT',
         'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'POINT_ON_EDGE_CURVE']);
    WR5 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured gap between geometries of the **vertex_point** and the **edge_curve** against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **length_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **length_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **edge_curve**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_report_item_with_extreme_instances** associated with this entity shall be of types **vertex_point** and **point_on_edge_curve** respectively.

NOTE The **point_on_edge_curve** is the nearest point on the **edge_curve** from the **vertex_point**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.97 gap_between_vertex_and_base_surface

A **gap_between_vertex_and_base_surface** asserts that geometry of the **vertex_point** does not lie on the underlying surface of the related **face_surface** within specified threshold. The measurement requirement corresponding to this entity requires that the measurement shall calculate distance between geometry of the **vertex_point** and the underlying surface of the **face_surface**. The nearest point on the surface shall not be restricted to within the effective area bounded by the loops.

EXPRESS specification:

```
*)
 ENTITY gap_between_vertex_and_base_surface
```

```
   SUBTYPE OF(geometric_gap_in_topology);
    SELF\shape_data_quality_criterion.assessment_specification :
      shape_data_quality_assessment_by_numerical_test;
   WHERE
    WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
        IN TYPEOF(SELF\shape_data_quality_criterion.
         assessment_specification.threshold))
        AND ('MEASURE_SCHEMA.LENGTH_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
         assessment_specification.threshold));
    WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.LENGTH_MEASURE');
    WR3 : validate_inspected_elements_type(SELF,
        ['TOPOLOGY_SCHEMA.FACE_SURFACE']);
    WR4 : validate_locations_of_extreme_value_type(SELF,
        ['TOPOLOGY_SCHEMA.VERTEX_POINT',
         'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'POINT_ON_FACE_SURFACE']);
    WR5 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured gap between the **vertex_point** and the underlying surface against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **length_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **length_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **face_surface**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of types **vertex_point** and **point_on_face_surface** respectively.

NOTE The **point_on_face_surface** is the nearest point on the **face_surface** from the **vertex_point**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.98 gap_between_adjacent_edges_in_loop

A **gap_between_adjacent_edges_in_loop** asserts that geometric gap exceeding specified threshold exists between adjacent **oriented_edge**s in an **edge_loop**. The measurement requirement corresponding to this entity requires that the measurement shall calculate, for each **oriented_edge** in the **edge_loop**, the point on its underlying curve corresponding to its start vertex and the point on the underlying curve of the previous **oriented_edge** corresponding to the same vertex, then calculate the distance between those two points so that it can be compared with the specified threshold in assessment.

EXPRESS specification:

```
*)
 ENTITY gap_between_adjacent_edges_in_loop
  SUBTYPE OF(geometric_gap_in_topology);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
       'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
       IN TYPEOF(SELF\shape_data_quality_criterion.
        assessment_specification.threshold))
       AND ('MEASURE_SCHEMA.LENGTH_MEASURE'
       IN value_limit_type(SELF\shape_data_quality_criterion.
        assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
       'MEASURE_SCHEMA.LENGTH_MEASURE');
   WR3 : validate_inspected_elements_type(SELF,
       ['TOPOLOGY_SCHEMA.EDGE_LOOP']);
   WR4 : validate_locations_of_extreme_value_type(SELF,
       ['TOPOLOGY_SCHEMA.ORIENTED_EDGE']);
   WR5 : validate_accuracy_types(SELF,
       ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured gap between adjacent **oriented_edge**s against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **length_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **length_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **edge_loop**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **oriented_edge**. It implies that the start point of the **oriented_edge** has gap with the end point of the previous **oriented_edge**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.99 gap_between_edge_and_base_surface

A **gap_between_edge_and_base_surface** asserts that one or more **edge_curve**s which bound a **face_surface** does not lie on the **face_surface** within specified threshold. The measurement requirement corresponding to this entity requires that the measurement shall calculate the maximum directed Hausdorff distance from the **edge_curve**s to the underlying surface of the **face_surface** under inspection.

NOTE In Figure 37, dashed lines illustrate the underlying surface of a **face_surface**, and solid lines illustrate the **edge_loop** of the **face_surface**. The small circles at the terminal of the arrowed line show a point on an **edge_curve** and a point of the base surface whose distance is equal to the directed Hausdorff distance from the **edge_curve** to the underlying surface. These points will be **locations_of_extreme_value** when a defect is detected for this criterion.

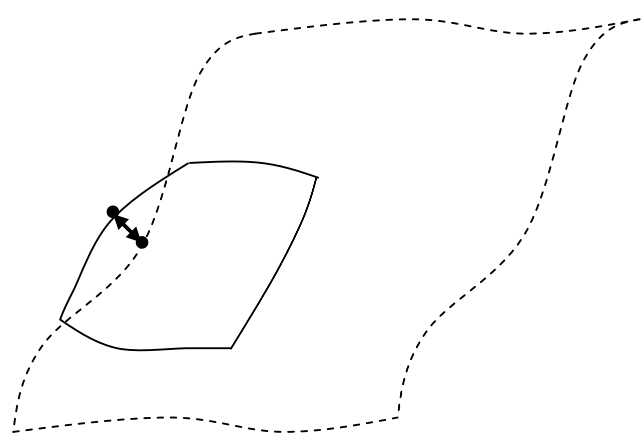

Figure 37 – Face surface with gap between its base surface and a bounding edge

EXPRESS specification:

```
*)
 ENTITY gap_between_edge_and_base_surface
   SUBTYPE OF(geometric_gap_in_topology);
    SELF\shape_data_quality_criterion.assessment_specification :
      shape_data_quality_assessment_by_numerical_test;
   WHERE
    WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
          'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
          IN TYPEOF(SELF\shape_data_quality_criterion.
            assessment_specification.threshold))
          AND ('MEASURE_SCHEMA.LENGTH_MEASURE'
          IN value_limit_type(SELF\shape_data_quality_criterion.
            assessment_specification.threshold));
    WR2 : validate_measured_data_type(SELF,
          'MEASURE_SCHEMA.LENGTH_MEASURE');
    WR3 : validate_inspected_elements_type(SELF,
          ['TOPOLOGY_SCHEMA.FACE_SURFACE']);
    WR4 : validate_locations_of_extreme_value_type(SELF,
          ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
           'POINT_ON_EDGE_CURVE',
           'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
           'POINT_ON_FACE_SURFACE']);
    WR5 : validate_accuracy_types(SELF,
          ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_by_numerical_test** that tests the gap between a bounding edge and the base surface of a **face_surface** against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality_lower_value_limit** and the **value_component** of the **threshold** shall be of type **length_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **length_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_report_item** associated with this entity shall be of type **face_surface**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of types **point_on_edge_-curve** and **point_on_face_surface** respectively.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.100 gap_between_faces_related_to_an_edge

A **gap_between_faces_related_to_an_edge** asserts that the results of projection from the geometry of an **edge_curve** onto geometry of related **face_surface**s are not identical within specified threshold. The measurement requirement corresponding to this entity requires that, for all the edges related to the faces in the **connected_face_set**, the measurement shall calculate two nearest points from any point on the **edge_curve** onto geometry of related **face_surface**s, and shall measure the distance between those two points. It shall obtain the maximum value of the distances calculated for each **edge_curve**.

NOTE In Figure 38 "**edge_curve** 1" is shared by "**face_surface** 1" and "**face_surface** 2." The figure at the bottom shows a cross section of the **edge_curve** shared by two **face_surface**s. The small white circle is a point on "**edge_curve** 1" and the dashed lines are underlying surfaces of "**face_surface** 1" and "**face_surface** 2." Two black small circles are the points projected from the point on "**edge_curve** 1" onto the underlying surfaces, and the distance between these points shall be evaluated. These points will be **locations_of_extreme_value** when a defect is detected for this criterion and the location gives the maximum distance.

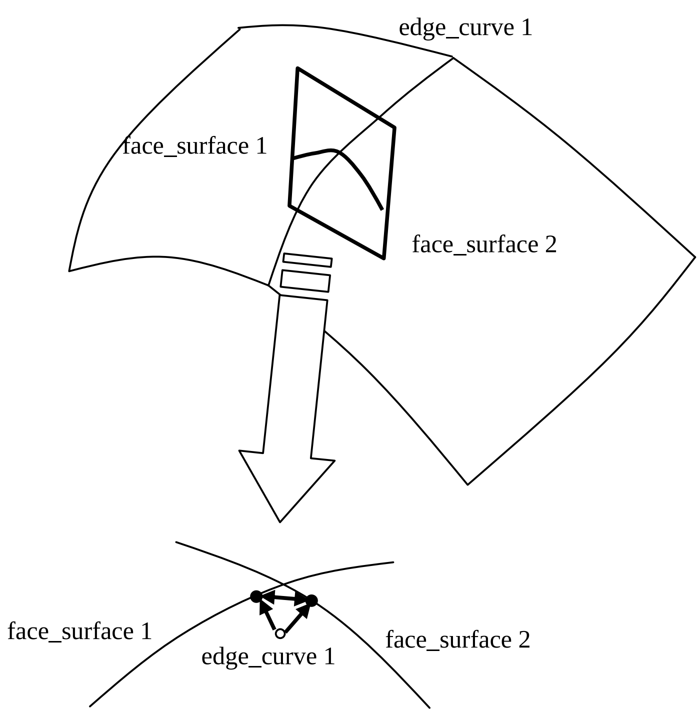

Figure 38 – Measurement of gap between faces related to an edge

EXPRESS specification:

```
*)
 ENTITY gap_between_faces_related_to_an_edge
   SUBTYPE OF(geometric_gap_in_topology);
    SELF\shape_data_quality_criterion.assessment_specification :
      shape_data_quality_assessment_by_numerical_test;
   WHERE
    WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
        IN TYPEOF(SELF\shape_data_quality_criterion.
         assessment_specification.threshold))
        AND ('MEASURE_SCHEMA.LENGTH_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
         assessment_specification.threshold));
    WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.LENGTH_MEASURE');
    WR3 : validate_inspected_elements_type(SELF,
        ['TOPOLOGY_SCHEMA.CONNECTED_FACE_SET']);
    WR4 : validate_locations_of_extreme_value_type(SELF,
        ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'POINT_ON_EDGE_CURVE']);
    WR5 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured gap between geometry of **face_surface**s against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **length_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **length_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **connected_face_set**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **point_on_edge_-curve**.

NOTE The **point_on_edge_curve** represents the position where maximum gap between related faces is detected.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.101 gap_between_pcurves_related_to_an_edge

A **gap_between_pcurves_related_to_an_edge** asserts that two **pcurve**s related to an **edge_curve** are not identical in 3D space within specified threshold. The measurement requirement corresponding to this entity requires that, for all the edges related to the faces in the **connected_face_set**, the measurement shall calculate Hausdorff distance between two curves which are 3D image of **pcurve**s associated to the **edge_curve**.

NOTE In Figure 39, "**edge_curve** 1" is shared by "**face_surface** 1" and "**face_surface** 2." Two dashed lines in the figure illustrate 3D space images of two **pcurve**s related to "**edge_curve** 1". The Hausdorff distance between these curves shall be measured for this criterion.

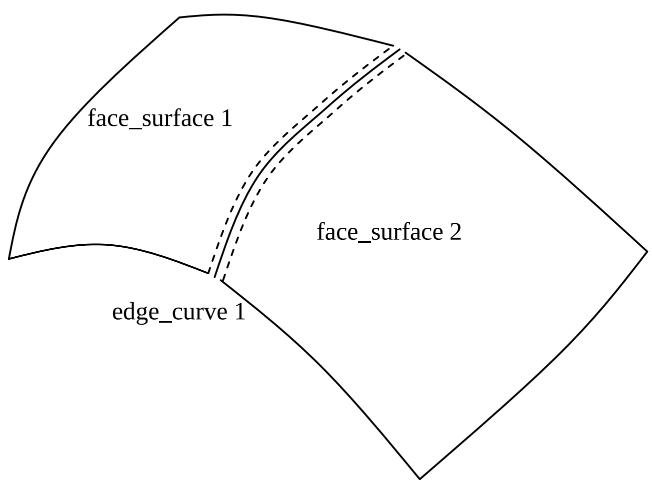

Figure 39 – Measurement of gap between pcurves related to an edge

EXPRESS specification:

```
*)
 ENTITY gap_between_pcurves_related_to_an_edge
  SUBTYPE OF(geometric_gap_in_topology);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
```

```
         'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
         IN TYPEOF(SELF\shape_data_quality_criterion.
          assessment_specification.threshold))
         AND ('MEASURE_SCHEMA.LENGTH_MEASURE'
         IN value_limit_type(SELF\shape_data_quality_criterion.
          assessment_specification.threshold));
    WR2 : validate_measured_data_type(SELF,
         'MEASURE_SCHEMA.LENGTH_MEASURE');
    WR3 : validate_inspected_elements_type(SELF,
         ['TOPOLOGY_SCHEMA.CONNECTED_FACE_SET']);
    WR4 : validate_locations_of_extreme_value_type(SELF,
         ['GEOMETRY_SCHEMA.POINT_ON_SURFACE',
          'GEOMETRY_SCHEMA.POINT_ON_SURFACE']);
    WR5 : validate_accuracy_types(SELF,
         ['MEASURE_SCHEMA.LENGTH_MEASURE']);
  END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_by_numerical_test** that tests the measured gap against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality_lower_value_limit** and the **value_component** of the **threshold** shall be of type **length_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **length_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_report_item** associated with this entity shall be of type **connected_face_set**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_report_item_with_extreme_instances** associated with this entity shall be both of type **point_on_surface**.

NOTE The **point_on_edge_curve** represents the position where the maximum gap between related **pcurve**s is detected.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.102 overcomplex_topology_and_geometry_relationship

An **overcomplex_topology_and_geometry_relationship** is a type of **inapt_topology_and_-geometry_relationship** and is a supertype of criteria for detecting topology related to overcomplex geometry data.

EXPRESS specification:

```
*)
 ENTITY overcomplex_topology_and_geometry_relationship
  ABSTRACT SUPERTYPE OF (ONEOF(
   unused_patches,
   edge_with_excessive_segments,
   face_surface_with_excessive_patches_in_one_direction))
  SUBTYPE OF(inapt_topology_and_geometry_relationship);
 END_ENTITY;
(*
```

7.4.103 edge_with_excessive_segments

An **edge_with_excessive_segments** asserts that geometry of an **edge_curve** contains excessive number of segments. This criterion is applicable only when the underlying curve is of type **b_spline_- curve**. The measurement requirement corresponding to this entity requires that the measurement shall calculate the bounded portion of its underlying curve trimmed by the start vertex and the end vertex. Then it shall count the number of the segments included in this portion.

EXPRESS specification:

```
*)
 ENTITY edge_with_excessive_segments
  SUBTYPE OF(overcomplex_topology_and_geometry_relationship);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
       'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
       IN TYPEOF(SELF\shape_data_quality_criterion.
        assessment_specification.threshold))
       AND ('MEASURE_SCHEMA.COUNT_MEASURE'
       IN value_limit_type(SELF\shape_data_quality_criterion.
        assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
       'MEASURE_SCHEMA.COUNT_MEASURE');
   WR3 : validate_inspected_elements_type(SELF,
       ['TOPOLOGY_SCHEMA.EDGE_CURVE']);
   WR4 : validate_locations_of_extreme_value_type(SELF,
       []);
```

```
    WR5 : validate_accuracy_types(SELF,
        []);
  END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the calculated number of segments against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **count_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **count_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **edge_curve**.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR5: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.104 face_surface_with_excessive_patches_in_one_direction

A **face_surface_with_excessive_patches_in_one_direction** asserts that geometry of a **face_surface** contains excessive number of patches within its effective area of the base surface. This criterion is applicable only when the base surface is of type **b_spline_surface**. The measurement requirement corresponding to this entity requires that the measurement shall obtain a part of the underlying surface by trimming it with the bounding box of the bounds of **face_surface** in parameter space, and then it shall calculate the number of patches of the trimmed surface both in u and v directions. The bigger one of the two shall be obtained as the maximum number of patches for comparison with the given threshold in assessment.

EXPRESS specification:

```
*)
  ENTITY face_surface_with_excessive_patches_in_one_direction
    SUBTYPE OF(overcomplex_topology_and_geometry_relationship);
      SELF\shape_data_quality_criterion.assessment_specification :
        shape_data_quality_assessment_by_numerical_test;
```

```
  WHERE
    WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
        IN TYPEOF(SELF\shape_data_quality_criterion.
         assessment_specification.threshold))
        AND ('MEASURE_SCHEMA.COUNT_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
         assessment_specification.threshold));
    WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.COUNT_MEASURE');
    WR3 : validate_inspected_elements_type(SELF,
        ['TOPOLOGY_SCHEMA.FACE_SURFACE']);
    WR4 : validate_locations_of_extreme_value_type(SELF,
        []);
    WR5 : validate_accuracy_types(SELF,
        []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the calculated number of patches against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **count_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **count_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **face_surface**.

NOTE Surface of the **face_surface** is required to be **b_spline_surface**.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR5: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.105 unused_patches

An **unused_patches** asserts that bounds of a **face_surface** uses only a small portion of its face geometry. The measurement requirement corresponding to this entity requires that the measurement shall calculate bounding box of the outer bound of the face in parameter space and shall compare it with the parameter range of surface geometry. And then it shall count rows and columns of the patches not included in the bounding box for comparison with the given threshold in assessment.

.

EXPRESS specification:

```
*)
 ENTITY unused_patches
   SUBTYPE OF(overcomplex_topology_and_geometry_relationship);
     SELF\shape_data_quality_criterion.assessment_specification :
       shape_data_quality_assessment_by_numerical_test;
   WHERE
     WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
         'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
         IN TYPEOF(SELF\shape_data_quality_criterion.
           assessment_specification.threshold))
          AND ('MEASURE_SCHEMA.COUNT_MEASURE'
         IN value_limit_type(SELF\shape_data_quality_criterion.
           assessment_specification.threshold));
     WR2 : validate_measured_data_type(SELF,
         'MEASURE_SCHEMA.COUNT_MEASURE');
     WR3 : validate_inspected_elements_type(SELF,
         ['TOPOLOGY_SCHEMA.FACE_SURFACE']);
     WR4 : validate_locations_of_extreme_value_type(SELF,
         ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
          'B_SPLINE_SURFACE_STRIP']);
     WR5 : validate_accuracy_types(SELF,
         []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the calculated number of rows or columns against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **count_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-**

value of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **count_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **face_surface**.

NOTE Surface of the **face_surface** is required to be **b_spline_surface**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **b_spline_surface-_strip**. It implies that this portion of the surface is not included in the effective region of the **face_surface**.

WR5: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.106 non_smooth_geometry_transition_across_edge

A **non_smooth_geometry_transition_across_edge** is a type of **inapt_topology_and_geometry_-relationship** that is a supertype of criteria for detecting two types of non smooth geometric transition across edge.

EXPRESS specification:

```
*)
 ENTITY non_smooth_geometry_transition_across_edge
  ABSTRACT SUPERTYPE OF (ONEOF(
   g1_discontinuity_between_adjacent_faces,
   g2_discontinuity_between_adjacent_faces))
  SUBTYPE OF(inapt_topology_and_geometry_relationship);
 END_ENTITY;
(*
```

7.4.107 g1_discontinuity_between_adjacent_faces

A **g1_discontinuity_between_adjacent_faces** asserts that underlying surfaces of adjacent faces sharing an **edge** are not G1 continuous.

NOTE 1 This is a selective criterion for selected set of faces in particular application.

The measurement requirement corresponding to this entity requires that the measurement shall take points on the **edge_curve**s where adjacent faces of the **connected_face_set** exist, and shall calculate the nearest points on the geometry of related **face_surface**s and shall get normal vectors on both surfaces. If the face is an **oriented_face**, its orientation shall be taken into account. It shall measure the maximum angle between the normal vectors among all the points on the **edge_curve** for comparison with the given threshold in assessment.

NOTE 2 Figure 40 illustrates measurement of angle between normal vectors of adjacent **face_surface**s.

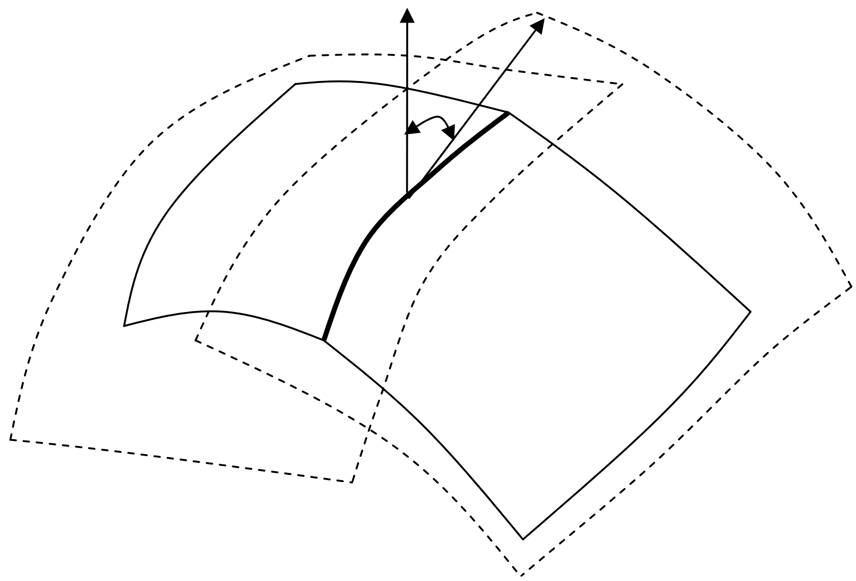

Figure 40 – Measurement of G1 discontinuity between faces

EXPRESS specification:

```
*)
 ENTITY g1_discontinuity_between_adjacent_faces
  SUBTYPE OF(non_smooth_geometry_transition_across_edge);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
        IN TYPEOF(SELF\shape_data_quality_criterion.
         assessment_specification.threshold))
        AND ('MEASURE_SCHEMA.PLANE_ANGLE_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
         assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.PLANE_ANGLE_MEASURE');
   WR3 : validate_inspected_elements_type(SELF,
        ['TOPOLOGY_SCHEMA.CONNECTED_FACE_SET']);
   WR4 : validate_locations_of_extreme_value_type(SELF,
        ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'POINT_ON_EDGE_CURVE']);
   WR5 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.PLANE_ANGLE_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_by_numerical_test** that tests the measured angle against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality_lower_value_limit** and the **value_component** of the **threshold** shall be of type **plane_angle_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **plane_angle_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_report_item** associated with this entity shall be of type **connected_face_set**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_report_item_with_extreme_instances** associated with this entity shall be of type **point_on_edge_curve**.

NOTE 2 The **point_on_edge_curve** represents the position where large angle between related faces is detected.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **plane_angle_measure**.

7.4.108 g2_discontinuity_between_adjacent_faces

A **g2_discontinuity_between_adjacent_faces** asserts that underlying surfaces of adjacent faces sharing an **edge** are not G2 continuous.

NOTE 1 This is a selective criterion for selected set of faces in particular application.

The measurement requirement corresponding to this entity requires that the measurement shall take points on the **edge_curve**s where adjacent faces of the **connected_face_set** exist, and shall get principal curvature value and principal curvature direction on both surfaces. It shall measure the maximum difference between the principal curvature and the maximum angle between principal curvature directions for all the points on the curve. Curvature directions shall be compared with the plane angle threshold given by the **assessment_specification**. Let r1 and r2 be principal curvature radius values to be compared. Curvature radius change ratio is defined as (2*|r1-r2|) / (|r1| + |r2|). Compare this value with the ratio threshold given by **assessment_specification_2** to check continuity of curvature radius values.

NOTE In Figure 41, the solid arrow line illustrates a principal curvature vector of the cylindrical surface of the left face and the dashed arrow line illustrates that of a surface of the right face. This is the case where the directions of curvature vectors are same, while the curvature radii are different. In contrast, Figure 42 shows a case where curvature radii are same and curvature vectors are different. This criterion shall detect both cases.

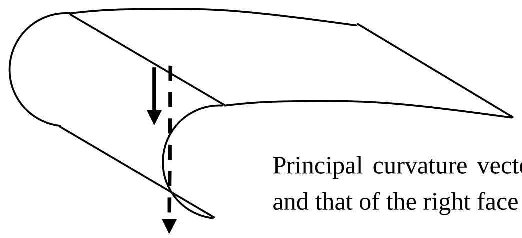

Figure 41 – G2 discontinuity between faces because of curvature radius difference

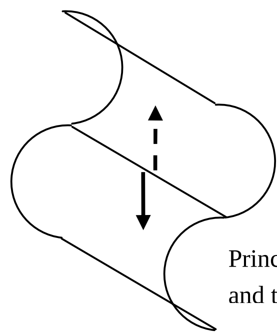

Figure 42 – G2 discontinuity between faces because of curvature direction difference

<u>EXPRESS specification:</u>

```
*)
 ENTITY g2_discontinuity_between_adjacent_faces
  SUBTYPE OF(non_smooth_geometry_transition_across_edge);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
   assessment_specification_2                             :
    shape_data_quality_assessment_by_numerical_test;
   flat_curvature_radius_tolerance : length_measure;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
         'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
         IN TYPEOF(SELF\shape_data_quality_criterion.
          assessment_specification.threshold))
         AND ('MEASURE_SCHEMA.PLANE_ANGLE_MEASURE'
         IN value_limit_type(SELF\shape_data_quality_criterion.
          assessment_specification.threshold));
   WR2 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
         'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
         IN TYPEOF(SELF.assessment_specification_2.threshold))
```

```
        AND ('MEASURE_SCHEMA.RATIO_MEASURE'
        IN value_limit_type(SELF.assessment_specification_2.threshold));
    WR3 : validate_measured_data_type(SELF,
         'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'PLANE_ANGLE_AND_RATIO_PAIR');
    WR4 : validate_inspected_elements_type(SELF,
        ['TOPOLOGY_SCHEMA.CONNECTED_FACE_SET']);
    WR5 : validate_locations_of_extreme_value_type(SELF,
        ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'POINT_ON_EDGE_CURVE']);
    WR6 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.RATIO_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_by_numerical_test** that tests the measured angle against the given threshold.

assessment_specification_2: **shape_data_quality_assessment_by_numerical_test** that tests the measured change ratio of the curvature radii against the given threshold.

flat_curvature_radius_tolerance: The tolerance value to check flat shape.

NOTE 2 If the curvature radius value is larger than **flat_curvature_radius_tolerance**, that curvature value shall not be used to check curvature change ratio. If two curvature radii are larger than this value, then it shall be understood that two segments are G2 continuous. If one of the curvature radii is larger than this value and the other is smaller, then it shall be understood that two segments are G2 discontinuous.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality_lower_value_limit** and the **value_component** of the **threshold** shall be of type **plane_angle_measure**.

WR2: The **threshold** of **assessment_specification_2** of this entity shall be of type **shape_data_quality_lower_value_limit** and the **value_component** of the **threshold** shall be of type **ratio_measure**.

WR3: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **plane_angle_and_ratio_pair**.

WR4: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_report_item** associated with this entity shall be of type **connected_face_set**.

WR5: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **point_on_edge_-curve**.

NOTE 3 The **point_on_edge_curve** represents the position where large curvature discontinuity between related faces is detected.

WR6: If the accuracy is specified for this measurement, its **value_component** shall be of type **ratio_-measure**.

7.4.109 steep_geometry_transition_across_edge

A **steep_geometry_transition_across_edge** is a type of **inapt_topology_and_geometry_-relationship** criterion that is a supertype of criteria for detecting a portion of shape data where adjacent topological elements are steeply connected.

<u>EXPRESS specification:</u>

```
*)
 ENTITY steep_geometry_transition_across_edge
  ABSTRACT SUPERTYPE OF (ONEOF(
   steep_angle_between_adjacent_edges,
   steep_angle_between_adjacent_faces))
  SUBTYPE OF(inapt_topology_and_geometry_relationship);
 END_ENTITY;
(*
```

7.4.110 steep_angle_between_adjacent_edges

A **steep_angle_between_adjacent_edges** asserts that adjacent **oriented_edge**s in an **edge_loop** form steep angle. The measurement requirement corresponding to this entity requires that, for each **oriented_edge** in the **edge_loop**, the measurement shall calculate the angle between tangent vectors of underlying curves of current **oriented_edge** and the next one in the **edge_loop** at the common vertex location for comparison with the given threshold in assessment.

<u>EXPRESS specification:</u>

```
*)
 ENTITY steep_angle_between_adjacent_edges
  SUBTYPE OF(steep_geometry_transition_across_edge);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
         'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
         IN TYPEOF(SELF\shape_data_quality_criterion.
          assessment_specification.threshold))
         AND ('MEASURE_SCHEMA.PLANE_ANGLE_MEASURE'
```

```
        IN value_limit_type(SELF\shape_data_quality_criterion.
         assessment_specification.threshold));
    WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.PLANE_ANGLE_MEASURE');
    WR3 : validate_inspected_elements_type(SELF,
        ['TOPOLOGY_SCHEMA.EDGE_LOOP']);
    WR4 : validate_locations_of_extreme_value_type(SELF,
        ['TOPOLOGY_SCHEMA.ORIENTED_EDGE']);
    WR5 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.PLANE_ANGLE_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured angle against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **plane_angle_-measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **plane_angle_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **edge_loop**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **oriented_edge**. It implies that the **oriented_edge** and the previous **oriented_edge** form a steep angle.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **plane_angle_measure**.

7.4.111 steep_angle_between_adjacent_faces

A **steep_angle_between_adjacent_faces** asserts that adjacent **faces** along an e**dge_curve** form steep angle. The measurement requirement corresponding to this entity requires that the measurement shall calculate for any point on the **edge_curve** the nearest point on underlying geometry of related **face_surface**s and shall get normal vectors on both surfaces. If the face is **oriented_face**, its orientation shall be taken into account. Then the measurement shall measure the maximum angle between the normal vectors for all of the points on the curve for comparison with the given threshold in assessment.

NOTE Figure 43 shows a typical case to be detected by this criterion. The solid arrow in the figure shows the normal vector on the lower **face_surface** and the dashed arrow shows the normal vector on the upper **face_surface**, both at a point on the **edge_curve** between two **face_surface**s.

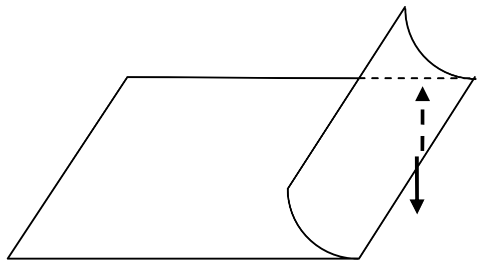

Figure 43 – Steep angle between faces

EXPRESS specification:

```
*)
 ENTITY steep_angle_between_adjacent_faces
   SUBTYPE OF(steep_geometry_transition_across_edge);
     SELF\shape_data_quality_criterion.assessment_specification :
       shape_data_quality_assessment_by_numerical_test;
   WHERE
     WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
         'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
         IN TYPEOF(SELF\shape_data_quality_criterion.
           assessment_specification.threshold))
         AND ('MEASURE_SCHEMA.PLANE_ANGLE_MEASURE'
         IN value_limit_type(SELF\shape_data_quality_criterion.
           assessment_specification.threshold));
     WR2 : validate_measured_data_type(SELF,
         'MEASURE_SCHEMA.PLANE_ANGLE_MEASURE');
     WR3 : validate_inspected_elements_type(SELF,
         ['TOPOLOGY_SCHEMA.CONNECTED_FACE_SET']);
     WR4 : validate_locations_of_extreme_value_type(SELF,
         ['SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
          'POINT_ON_EDGE_CURVE']);
     WR5 : validate_accuracy_types(SELF,
         ['MEASURE_SCHEMA.PLANE_ANGLE_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured angle against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **plane_angle_-measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **plane_angle_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **connected_face_set**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be of type **point_on_edge_-curve**. It indicates that the angle between two **face_surface**s related to the **edge_curve** form a steep angle at the point.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **plane_angle_measure**.

7.4.112 topology_related_to_multiply_defined_geometry

A **topology_related_to_multiply_defined_geometry** is a type of **inapt_topology_and_geometry_-relationship** and is a supertype of criteria for detecting topological entities whose underlying geometry completely overlaps.

EXPRESS specification:

```
*)
 ENTITY topology_related_to_multiply_defined_geometry
  ABSTRACT SUPERTYPE OF (ONEOF(
   multiply_defined_faces,
   multiply_defined_vertices,
   multiply_defined_edges))
  SUBTYPE OF(inapt_topology_and_geometry_relationship);
 END_ENTITY;
(*
```

7.4.113 multiply_defined_vertices

A **multiply_defined_vertices** asserts that two **vertex_point**s are overlapping. The measurement requirement corresponding to this entity requires that the measurement shall calculate distance between two **vertex_point**s.

EXPRESS specification:

```
*)
 ENTITY multiply_defined_vertices
   SUBTYPE OF(topology_related_to_multiply_defined_geometry);
     SELF\shape_data_quality_criterion.assessment_specification :
       shape_data_quality_assessment_by_numerical_test;
   WHERE
     WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
         'SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT'
         IN TYPEOF(SELF\shape_data_quality_criterion.
           assessment_specification.threshold))
         AND ('MEASURE_SCHEMA.LENGTH_MEASURE'
         IN value_limit_type(SELF\shape_data_quality_criterion.
           assessment_specification.threshold));
     WR2 : validate_measured_data_type(SELF,
         'MEASURE_SCHEMA.LENGTH_MEASURE');
     WR3 : validate_inspected_elements_type(SELF,
         ['TOPOLOGY_SCHEMA.VERTEX_POINT',
          'TOPOLOGY_SCHEMA.VERTEX_POINT']);
     WR4 : validate_locations_of_extreme_value_type(SELF,
         []);
     WR5 : validate_accuracy_types(SELF,
         []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured distance against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_upper_value_limit** and the **value_component** of the **threshold** shall be of type **length_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **length_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be both of type **vertex_point**.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR5: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.4.114 multiply_defined_edges

A **multiply_defined_edges** asserts that two **edge_curve**s are entirely overlapping. The measurement requirement corresponding to this entity requires that the measurement shall calculate Hausdorff distance between two **edge_curve**s for checking the result if it is less than the threshold value in assessment.

EXPRESS specification:

```
*)
 ENTITY multiply_defined_edges
   SUBTYPE OF(topology_related_to_multiply_defined_geometry);
     SELF\shape_data_quality_criterion.assessment_specification :
       shape_data_quality_assessment_by_numerical_test;
   WHERE
     WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
         'SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT'
         IN TYPEOF(SELF\shape_data_quality_criterion.
           assessment_specification.threshold))
         AND ('MEASURE_SCHEMA.LENGTH_MEASURE'
         IN value_limit_type(SELF\shape_data_quality_criterion.
           assessment_specification.threshold));
     WR2 : validate_measured_data_type(SELF,
         'MEASURE_SCHEMA.LENGTH_MEASURE');
     WR3 : validate_inspected_elements_type(SELF,
         ['TOPOLOGY_SCHEMA.EDGE_CURVE',
          'TOPOLOGY_SCHEMA.EDGE_CURVE']);
     WR4 : validate_locations_of_extreme_value_type(SELF,
         []);
     WR5 : validate_accuracy_types(SELF,
         ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured distance against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_upper_value_limit** and the **value_component** of the **threshold** shall be of type **length_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **length_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be both of type **edge_curve**.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.115 multiply_defined_faces

A **multiply_defined_faces** asserts that two **face_surface**s are entirely overlapping. The measurement requirement corresponding to this entity requires that the measurement shall calculate Hausdorff distance between two **face_surface**s for checking the result if it is less than the threshold value in assessment.

EXPRESS specification:

```
*)
 ENTITY multiply_defined_faces
  SUBTYPE OF(topology_related_to_multiply_defined_geometry);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT'
        IN TYPEOF(SELF\shape_data_quality_criterion.
         assessment_specification.threshold))
        AND ('MEASURE_SCHEMA.LENGTH_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
         assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.LENGTH_MEASURE');
   WR3 : validate_inspected_elements_type(SELF,
        ['TOPOLOGY_SCHEMA.FACE_SURFACE',
         'TOPOLOGY_SCHEMA.FACE_SURFACE']);
   WR4 : validate_locations_of_extreme_value_type(SELF,
        []);
   WR5 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured distance against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_upper_value_limit** and the **value_component** of the **threshold** shall be of type **length_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **length_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be both of type **face_surface**.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.116 topology_related_to_overlapping_geometry

A **topology_related_to_overlapping_geometry** is a type of **inapt_topology_and_geometry_-relationship** that is a supertype of criteria for detecting topological entities whose underlying geometry partially overlaps.

EXPRESS specification:

```
*)
 ENTITY topology_related_to_overlapping_geometry
  ABSTRACT
  SUPERTYPE OF (ONEOF(
   partly_overlapping_faces,partly_overlapping_edges))
  SUBTYPE OF(inapt_topology_and_geometry_relationship);
 END_ENTITY;
(*
```

7.4.117 partly_overlapping_edges

A **partly_overlapping_edges** asserts that two **edge_curve**s are partly overlapping. Two **edge_curve**s overlap if each **edge_curve** contains arcwise connected set of points such that every point in the set for one **edge_curve** is within **interference_tolerance** from the other **edge_curve**. Arc length of the overlapping portion shall be evaluated for each **edge_curve** and the larger one is required to be greater than or equal to the minimum of the specified threshold value and the shorter arc length.

EXPRESS specification:

```
*)
 ENTITY partly_overlapping_edges
   SUBTYPE OF(topology_related_to_overlapping_geometry);
     SELF\shape_data_quality_criterion.assessment_specification :
       shape_data_quality_assessment_by_numerical_test;
     interference_tolerance : length_measure;
   WHERE
     WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
           'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
           IN TYPEOF(SELF\shape_data_quality_criterion.
             assessment_specification.threshold))
           AND ('MEASURE_SCHEMA.LENGTH_MEASURE'
           IN value_limit_type(SELF\shape_data_quality_criterion.
             assessment_specification.threshold));
     WR2 : validate_measured_data_type(SELF,
           'MEASURE_SCHEMA.LENGTH_MEASURE');
     WR3 : validate_inspected_elements_type(SELF,
           ['TOPOLOGY_SCHEMA.EDGE_CURVE',
            'TOPOLOGY_SCHEMA.EDGE_CURVE']);
     WR4 : validate_locations_of_extreme_value_type(SELF,
           ['TOPOLOGY_SCHEMA.SUBEDGE','TOPOLOGY_SCHEMA.SUBEDGE']);
     WR5 : validate_accuracy_types(SELF,
           ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured arc length against the given threshold.

interference_tolerance: tolerance to check if two points on the edges are coincident or not.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **length_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **length_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be both of type **edge_curve**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be both of type **subedge**.

NOTE These two **subedge**s are part of the inspected **edge_curve**s where distance from all points on one **edge_curve** to the other **edge_curve** is smaller than the **interference_tolerance**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.118 partly_overlapping_faces

A **partly_overlapping_faces** asserts that two **face_surface**s are partly overlapping. Two **face_surface**s overlap if each **face_surface** contains an arcwise connected set of points such that every point in the set for one **face_surface** is within **interference_tolerance** from the other **face_surface**. Area of the overlapping portion shall be evaluated for each **face_surface** and the larger one is required to be greater than or equal to the minimum of the specified threshold value and the smaller surface area.

EXPRESS specification:

```
*)
 ENTITY partly_overlapping_faces
   SUBTYPE OF(topology_related_to_overlapping_geometry);
    SELF\shape_data_quality_criterion.assessment_specification :
      shape_data_quality_assessment_by_numerical_test;
    interference_tolerance : length_measure;
   WHERE
    WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
         'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
         IN TYPEOF(SELF\shape_data_quality_criterion.
          assessment_specification.threshold))
         AND ('MEASURE_SCHEMA.AREA_MEASURE'
         IN value_limit_type(SELF\shape_data_quality_criterion.
          assessment_specification.threshold));
    WR2 : validate_measured_data_type(SELF,
         'MEASURE_SCHEMA.AREA_MEASURE');
    WR3 : validate_inspected_elements_type(SELF,
         ['TOPOLOGY_SCHEMA.FACE_SURFACE',
          'TOPOLOGY_SCHEMA.FACE_SURFACE']);
    WR4 : validate_locations_of_extreme_value_type(SELF,
         ['TOPOLOGY_SCHEMA.SUBFACE','TOPOLOGY_SCHEMA.SUBFACE']);
    WR5 : validate_accuracy_types(SELF,
         ['MEASURE_SCHEMA.AREA_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured area against the given threshold.

interference_tolerance: The tolerance used to check if two points are coincident.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **area_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **area_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be both of type **face_surface**.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be both of type **subface**.

NOTE These **subface**s are part of the inspected **face_surface**s where distance from all points on one **face_surface** to the other **face_surface** is smaller than the **interference_tolerance**.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **area_measure**.

7.4.119 inapt_manifold_solid_brep

An **inapt_manifold_solid_brep** is a type of **inapt_data** and is a supertype of criteria for detecting some types of inappropriate geometric model.

EXPRESS specification:

```
*)
 ENTITY inapt_manifold_solid_brep
  ABSTRACT SUPERTYPE OF (ONEOF(
   partly_overlapping_solids,
   small_volume_solid,
   multiply_defined_solids,
   entirely_narrow_solid,
   solid_with_excessive_number_of_voids))
  SUBTYPE OF(inapt_data);
 END_ENTITY;
(*
```

7.4.120 small_volume_solid

A **small_volume_solid** asserts that volume of a **manifold_solid_brep** is too small. The measurement requirement corresponding to this entity requires that the measurement shall calculate volume of the **manifold_solid_brep** that shall be compared with the given threshold in assessment. Volume of voids shall be subtracted in case of **brep_with_voids**.

EXPRESS specification:

```
*)
 ENTITY small_volume_solid
  SUBTYPE OF(inapt_manifold_solid_brep);
   SELF\shape_data_quality_criterion.assessment_specification :
    shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
       'SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT'
       IN TYPEOF(SELF\shape_data_quality_criterion.
         assessment_specification.threshold))
       AND ('MEASURE_SCHEMA.VOLUME_MEASURE'
           IN value_limit_type(SELF\shape_data_quality_criterion.
           assessment_specification.threshold));
   WR2 : validate_measured_data_type(SELF,
       'MEASURE_SCHEMA.VOLUME_MEASURE');
   WR3 : validate_inspected_elements_type(SELF,
       ['GEOMETRIC_MODEL_SCHEMA.MANIFOLD_SOLID_BREP']);
   WR4 : validate_locations_of_extreme_value_type(SELF,
       []);
   WR5 : validate_accuracy_types(SELF,
       ['MEASURE_SCHEMA.VOLUME_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured volume against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_upper_value_limit** and the **value_component** of the **threshold** shall be of type **volume_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **volume_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **manifold_solid_brep**.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances** since identification of locations of extreme values is meaningless as an inspection result report.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **volume_measure**.

7.4.121 entirely_narrow_solid

An **entirely_narrow_solid** asserts that the width of a **manifold_solid_brep** is entirely infinitesimal. The measurement requirement corresponding to this entity requires that the measurement shall calculate for sufficient number of selected points within the **manifold_solid_brep** minimum distance to the bounding shells. If this exceeds half the distance tolerance, then **manifold_solid_brep** is not narrow.

NOTE Figure 44 illustrates the measurement of this criterion. The measurement considers a sphere whose diameter is **width_tolerance** and whose centre is at any point inside the **manifold_solid_brep** and checks if the sphere does not intersect with the boundary of the **manifold_solid_brep**. The test is complete when one such internal point is detected.

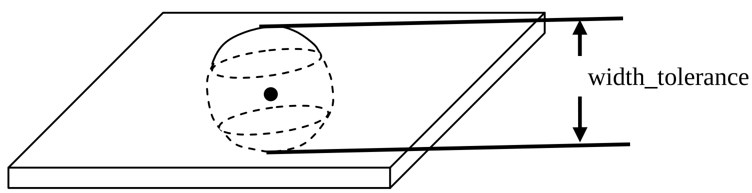

Figure 44 – Measurement of entirely narrow solid

<u>EXPRESS specification:</u>

```
*)
 ENTITY entirely_narrow_solid
  SUBTYPE OF(inapt_manifold_solid_brep);
   SELF\shape_data_quality_criterion.assessment_specification :
     shape_data_quality_assessment_by_logical_test;
   width_tolerance    : length_measure;
  WHERE
   WR1 : validate_measured_data_type(SELF,
        'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.BOOLEAN_VALUE');
   WR2 : validate_inspected_elements_type(SELF,
        ['GEOMETRIC_MODEL_SCHEMA.MANIFOLD_SOLID_BREP']);
   WR3 : validate_locations_of_extreme_value_type(SELF,
```

```
        []);
    WR4 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_logical_test** that tests if the solid is narrow for all the inspected points, which is the case 'TRUE', or not.

width_tolerance: The tolerance value used to check if the solid is narrow or not.

Formal propositions:

WR1: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **boolean_value**.

WR2: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **manifold_solid_brep**.

WR3: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances** since identification of locations of extreme values is meaningless as an inspection result report.

WR4: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.122 multiply_defined_solids

A **multiply_defined_solids** asserts that two **manifold_solid_brep**s are entirely overlapping. The measurement requirement corresponding to this entity requires that the measurement shall calculate Hausdorff distance between two **manifold_solid_brep**s for the check if it is less than the threshold value in assessment.

EXPRESS specification:

```
*)
 ENTITY multiply_defined_solids
  SUBTYPE OF(inapt_manifold_solid_brep);
   SELF\shape_data_quality_criterion.assessment_specification :
     shape_data_quality_assessment_by_numerical_test;
  WHERE
   WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT'
        IN TYPEOF(SELF\shape_data_quality_criterion.
```

```
          assessment_specification.threshold))
        AND ('MEASURE_SCHEMA.LENGTH_MEASURE'
            IN value_limit_type(SELF\shape_data_quality_criterion.
             assessment_specification.threshold));
    WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.LENGTH_MEASURE');
    WR3 : validate_inspected_elements_type(SELF,
        ['GEOMETRIC_MODEL_SCHEMA.MANIFOLD_SOLID_BREP',
         'GEOMETRIC_MODEL_SCHEMA.MANIFOLD_SOLID_BREP']);
    WR4 : validate_locations_of_extreme_value_type(SELF,
        []);
    WR5 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.LENGTH_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured distance against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_upper_value_limit** and the **value_component** of the **threshold** shall be of type **length_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **length_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type two **manifold_solid_brep**s.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances** since identification of locations of extreme values is meaningless as an inspection result report.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **length_measure**.

7.4.123 partly_overlapping_solids

A **partly_overlapping_solids** asserts that two **manifold_solid_brep**s are partly overlapping. Two solids overlap if each solid contains arcwise connected set of points on the boundary faces such that every point in the set for one solid is within **interference_tolerance** from the boundary of the other solid. The area of the overlapping portion shall be evaluated for each solid and the larger one is required to be greater than or equal to the minimum of the given threshold value and the surface area of the smaller solid.

EXPRESS specification:

```
*)
 ENTITY partly_overlapping_solids
   SUBTYPE OF(inapt_manifold_solid_brep);
    SELF\shape_data_quality_criterion.assessment_specification :
      shape_data_quality_assessment_by_numerical_test;
    interference_tolerance : length_measure;
   WHERE
    WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
        'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
        IN TYPEOF(SELF\shape_data_quality_criterion.
          assessment_specification.threshold))
         AND ('MEASURE_SCHEMA.AREA_MEASURE'
        IN value_limit_type(SELF\shape_data_quality_criterion.
         assessment_specification.threshold));
    WR2 : validate_measured_data_type(SELF,
        'MEASURE_SCHEMA.AREA_MEASURE');
    WR3 : validate_inspected_elements_type(SELF,
        ['GEOMETRIC_MODEL_SCHEMA.MANIFOLD_SOLID_BREP',
         'GEOMETRIC_MODEL_SCHEMA.MANIFOLD_SOLID_BREP']);
    WR4 : validate_locations_of_extreme_value_type(SELF,
        ['TOPOLOGY_SCHEMA.SUBFACE','TOPOLOGY_SCHEMA.SUBFACE']);
    WR5 : validate_accuracy_types(SELF,
        ['MEASURE_SCHEMA.AREA_MEASURE']);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the measured area against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **area_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report**, the **measured_value** of **shape_data_quality_inspection_instance_report_item**, and the **measured_-value** of all the **extreme_instance** in the **instance_report_item_with_extreme_instances**, for all instances associated with this entity shall be of type **area_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be two **manifold_solid_brep**s.

WR4: The **locations_of_extreme_value** for all instances of the **extreme_instance**s in the **instance_-report_item_with_extreme_instances** associated with this entity shall be two **subface**s.

WR5: If the accuracy is specified for this measurement, its **value_component** shall be of type **area_measure**.

7.4.124 solid_with_excessive_number_of_voids

A **solid_with_excessive_number_of_voids** asserts that a solid has one or more voids. The measurement requirement corresponding to this entity requires that the measurement shall count the number of voids in the **manifold_solid_brep** for comparison with the given threshold in assessment.

NOTE This criterion is intended to detect voids that are unintentionally generated in the design process.

EXPRESS specification:

```
*)
 ENTITY solid_with_excessive_number_of_voids
   SUBTYPE OF(inapt_manifold_solid_brep);
     SELF\shape_data_quality_criterion.assessment_specification :
       shape_data_quality_assessment_by_numerical_test;
   WHERE
     WR1 : ('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.' +
          'SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT'
          IN TYPEOF(SELF\shape_data_quality_criterion.
             assessment_specification.threshold))
          AND ('MEASURE_SCHEMA.COUNT_MEASURE'
              IN value_limit_type(SELF\shape_data_quality_criterion.
              assessment_specification.threshold));
     WR2 : validate_measured_data_type(SELF,
          'MEASURE_SCHEMA.COUNT_MEASURE');
     WR3 : validate_inspected_elements_type(SELF,
          ['GEOMETRIC_MODEL_SCHEMA.MANIFOLD_SOLID_BREP']);
     WR4 : validate_locations_of_extreme_value_type(SELF,
          []);
     WR5 : validate_accuracy_types(SELF,
          []);
 END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_data_quality_criterion.assessment_specification: **shape_data_quality_assessment_-by_numerical_test** that tests the number of voids against the given threshold.

Formal propositions:

WR1: The **threshold** of **assessment_specification** of this entity shall be of type **shape_data_quality-_lower_value_limit** and the **value_component** of the **threshold** shall be of type **count_measure**.

WR2: The **representative_measured_value** of **shape_data_quality_inspection_criterion_report** and the **measured_value** of **shape_data_quality_inspection_instance_report_item**, for all instances associated with this entity shall be of type **count_measure**.

WR3: The **inspected_elements** for all instances of **shape_data_quality_inspection_instance_-report_item** associated with this entity shall be of type **manifold_solid_brep**.

WR4: The **shape_data_quality_inspection_instance_report_item** associated with this entity shall not be of type **instance_report_item_with_extreme_instances** since identification of locations of extreme values is meaningless as an inspection result report.

WR5: This entity shall not be related with any accuracy types via **shape_data_quality_criterion_-and_accuracy_association.** The entity **shape_data_quality_inspection_result** corresponding to this entity shall also not be related with any accuracy types via **shape_inspection_result_accuracy_-association**.

7.5 Shape data quality criteria function definitions

7.5.1 validate_inspected_elements_type

A **validate_inspected_elements_type** function checks whether types of the **inspected_elements** specified in the **shape_data_quality_inspection_instance_report_item** of the **shape_data_quality_-inspection_instance_report** for **shape_data_quality_criterion** given as an input argument are contained in the type list given as an input argument of this function. If no instance of **shape_data_-quality_inspection_instance_report** for this criterion exists, this function returns true. The operation of the function is:

1. collect all **shape_data_quality_inspection_instance_report_item** instances associated with the **shape_data_quality_criterion** given as an argument of this function via **shape_data_quality_-inspection_result** and **shape_data_quality_inspection_instance_report**.
2. check if the types of **inspected_elements** of the collected instances are exactly coincident with the type names in the type list given as an input argument of this function.

<u>EXPRESS specification:</u>

```
*)
 FUNCTION validate_inspected_elements_type
  (ei:shape_data_quality_criterion;
   type_list:BAG OF STRING):LOGICAL;
   LOCAL
     sdqir: SET OF
      SHAPE_DATA_QUALITY_INSPECTION_RESULT:=[];
     sdir: SET OF
      SHAPE_DATA_QUALITY_INSPECTION_INSTANCE_REPORT:=[];
     sdiri: SET OF
      SHAPE_DATA_QUALITY_INSPECTION_INSTANCE_REPORT_ITEM:=[];
     t_type_list:BAG OF STRING:=[];
```

```
END_LOCAL;

-- STEP-1
-- collect all shape_data_quality_inspection_instance_report_item
-- instances associated with the given shape_data_quality_criterion
-- via shape_data_quality_inspection_result and
-- shape_data_quality_inspection_instance_report

sdqir := QUERY(q <* bag_to_set(USEDIN(ei,
         'PRODUCT_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'DATA_QUALITY_INSPECTION_RESULT.'+
         'CRITERION_INSPECTED'))|
    'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
    'SHAPE_DATA_QUALITY_INSPECTION_RESULT'
     IN TYPEOF(q));
IF(SIZEOF(sdqir) = 0) THEN
 RETURN(TRUE);
END_IF;
REPEAT i:= 1 TO SIZEOF(sdqir);
 sdir := sdir + QUERY(q <* bag_to_set(USEDIN(sdqir[i],
         'PRODUCT_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
         'DATA_QUALITY_INSPECTION_REPORT.'+
         'INSPECTION_RESULT'))|
    'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
    'SHAPE_DATA_QUALITY_INSPECTION_INSTANCE_REPORT'
     IN TYPEOF(q));
END_REPEAT;
IF(SIZEOF(sdir) = 0) THEN
 RETURN(TRUE);
END_IF;

REPEAT i:= 1 TO SIZEOF(sdir);
 sdiri := sdiri + QUERY(q <* sdir[i].inspected_instances|
    'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
    'SHAPE_DATA_QUALITY_INSPECTION_INSTANCE_REPORT_ITEM'
     IN TYPEOF(q));
END_REPEAT;

-- STEP-2
-- check if the types of inspected_elements of the collected
-- instances are exactly coincident with the type names in the
-- type list given as an input argument of this function.

-- If there is no instance_report_item, then return true.
IF(SIZEOF(sdiri) = 0) THEN
 RETURN(TRUE);
END_IF;
```

```
   -- When reports exist, then perform the following checking process.
   REPEAT i:= 1 TO SIZEOF(sdiri);
    -- If the size of inspected_elements is not equal to the size of
    -- type_list, then return false.
    IF(SIZEOF(sdiri[i].inspected_elements) <>
       SIZEOF(type_list)) THEN
      RETURN(FALSE);
    END_IF;
    -- check all elements in the type_list
    -- for each element of inspected_elements.
    t_type_list := type_list;
    REPEAT j:= 1 TO SIZEOF(sdiri[i].inspected_elements);
      REPEAT k:= 1 TO SIZEOF(t_type_list);
        IF(t_type_list[k] IN
           TYPEOF(sdiri[i].inspected_elements[j])) THEN
         t_type_list := t_type_list - t_type_list[k];
         ESCAPE;
        END_IF;
      END_REPEAT;
    END_REPEAT;
    IF(SIZEOF(t_type_list) > 0) THEN
      RETURN(FALSE);
    END_IF;
   END_REPEAT;
  RETURN(TRUE);
 END_FUNCTION;
(*
```

<u>Argument definitions:</u>

ei: An instance of **shape_data_quality_criterion** to be checked.

type_list: A bag of string giving acceptable types in the **inspected_shape_element_select**.

7.5.2 validate_locations_of_extreme_value_type

A **validate_locations_of_extreme_value_type** function validates the types of the **locations_of_-extreme_values** of **instance_report_item_with_extreme_instances** associated with a **shape_data_-quality_criterion** instance. The operation of the function is:

1. collect all **instance_report_item_with_extreme_instances** instances associated with the **shape_data_quality_criterion** given as an argument of this function via **shape_data_quality_-inspection_result** and **shape_data_quality_inspection_instance_report**.
2. check if the types of **locations_of_extreme_value** attributes of all **extreme_instances** attributes of the collected instances are exactly coincident with the type names in the type list given as an input argument of this function.

NOTE If the type list is empty and **instance_report_item_with_extreme_instances** relating to the **shape_data_quality_criterion** exists, this function returns false.

EXPRESS specification:

```
*)
 FUNCTION validate_locations_of_extreme_value_type
  (ei:shape_data_quality_criterion;
   type_list:BAG OF STRING):LOGICAL;
   LOCAL
     sdqir: SET OF
      SHAPE_DATA_QUALITY_INSPECTION_RESULT:=[];
     sdir: SET OF
      SHAPE_DATA_QUALITY_INSPECTION_INSTANCE_REPORT:=[];
     sdei: SET OF
      INSTANCE_REPORT_ITEM_WITH_EXTREME_INSTANCES:=[];
     t_type_list:BAG OF STRING:=[];
   END_LOCAL;

   -- STEP-1
   -- collect all instance_report_item_with_extreme_instances instances
   -- associated with the given shape_data_quality_criterion
   -- via shape_data_quality_inspection_result and
   -- shape_data_quality_inspection_instance_report.

   sdqir := QUERY(q <* bag_to_set(USEDIN(ei,
              'PRODUCT_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
              'DATA_QUALITY_INSPECTION_RESULT.'+
              'CRITERION_INSPECTED'))|
        'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
        'SHAPE_DATA_QUALITY_INSPECTION_RESULT'
         IN TYPEOF(q));
   IF(SIZEOF(sdqir) = 0) THEN
     RETURN(TRUE);
   END_IF;
   REPEAT i:= 1 TO SIZEOF(sdqir);
     sdir := sdir + QUERY(q <* bag_to_set(USEDIN(sdqir[i],
              'PRODUCT_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
              'DATA_QUALITY_INSPECTION_REPORT.'+
              'INSPECTION_RESULT'))|
        'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
        'SHAPE_DATA_QUALITY_INSPECTION_INSTANCE_REPORT'
         IN TYPEOF(q));
   END_REPEAT;
   REPEAT i:= 1 TO SIZEOF(sdir);
     sdei := sdei + QUERY(q <* sdir[i].inspected_instances|
        'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
        'INSTANCE_REPORT_ITEM_WITH_EXTREME_INSTANCES'
         IN TYPEOF(q));
   END_REPEAT;
```

```
-- STEP-2
-- check if the types of locations_of_extreme_value attributes
-- of all extreme_instances attributes of the collected instances are
-- exactly coincident with the type names in the type list given as an
-- input argument of this function.

-- If there is no instance report, then return true.
IF(SIZEOF(sdei) = 0) THEN
 RETURN(TRUE);
END_IF;

-- If the type_list is empty and
-- instance_report_item_with_extreme_instances exists, then return false.
IF((SIZEOF(type_list) = 0) AND (SIZEOF(sdei) <> 0)) THEN
 RETURN(FALSE);
END_IF;

-- When report_items exist, then perform the following checking process.
REPEAT i:= 1 TO SIZEOF(sdei);
 REPEAT j:= 1 TO SIZEOF(sdei[i].extreme_instances);
  -- If the size of locations_of_extreme_value is not equal to
  -- the size of type_list, then return false.
  IF(SIZEOF(sdei[i].extreme_instances[j].
             locations_of_extreme_value) <>
    SIZEOF(type_list)) THEN
    RETURN(FALSE);
  END_IF;
  t_type_list := type_list;
  -- check all elements of the type_list
  --  for each element of locations_of_extreme_value.
  REPEAT k:= 1 TO SIZEOF(sdei[i].extreme_instances[j]
                 .locations_of_extreme_value);
   REPEAT l:= 1 TO SIZEOF(t_type_list);
    IF(t_type_list[l] IN
     TYPEOF(sdei[i].extreme_instances[j].
          locations_of_extreme_value[k])) THEN
     t_type_list := t_type_list - t_type_list[l];
     ESCAPE;
    END_IF;
   END_REPEAT;
  END_REPEAT;
  IF(SIZEOF(t_type_list) > 0) THEN
   RETURN(FALSE);
  END_IF;
 END_REPEAT;
END_REPEAT;
```

```
   RETURN(TRUE);
 END_FUNCTION;
(*
```

Argument definitions:

ei: An instance of **shape_data_quality_criterion** to be checked.

type_list: A bag of string giving acceptable types for the **location_of_extreme_value_select**.

7.5.3 validate_measured_data_type

A **validate_measured_data_type** function validates the type of **representative_measured_value** of the **shape_data_quality_inspection_criterion_report**, the **measured_value** of the **shape_data_-quality_inspection_instance_report_item** and the **measured_value** of **extreme_instance**s referred from **instance_report_item_with_extreme_instances** associated with a **shape_data_quality_-criterion** instance.

NOTE If any instance of **shape_data_quality_inspection_criterion_report** and **shape_data_quality_-inspection_instance_report** for the criterion do not exist, this function returns true.

EXPRESS specification:

```
*)
 FUNCTION validate_measured_data_type
  (ei:shape_data_quality_criterion;
   type_string: STRING):LOGICAL;
   LOCAL
     sdqir: SET OF
      SHAPE_DATA_QUALITY_INSPECTION_RESULT:=[];
     sdcr: SET OF
      SHAPE_DATA_QUALITY_INSPECTION_CRITERION_REPORT:=[];
     sdir: SET OF
      SHAPE_DATA_QUALITY_INSPECTION_INSTANCE_REPORT:=[];
     sdii: SET OF
      SHAPE_DATA_QUALITY_INSPECTION_INSTANCE_REPORT_ITEM:=[];
     sdei: BAG OF
      INSTANCE_REPORT_ITEM_WITH_EXTREME_INSTANCES:=[];
   END_LOCAL;

   -- STEP-1
   -- collect all shape_data_quality_inspection_criterion_report instances,
   -- all shape_data_quality_inspection_instance_report_item instances, and
   -- instance_report_item_with_extreme_instances instances
   -- associated with the given shape_data_quality_criterion
   -- via shape_data_quality_inspection_result.

   sdqir := QUERY(q <* bag_to_set(USEDIN(ei,
```

```
          'PRODUCT_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
          'DATA_QUALITY_INSPECTION_RESULT.'+
          'CRITERION_INSPECTED'))|
      'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
      'SHAPE_DATA_QUALITY_INSPECTION_RESULT'
       IN TYPEOF(q));
  IF(SIZEOF(sdqir) = 0) THEN
   RETURN(TRUE);
  END_IF;
  REPEAT i:= 1 TO SIZEOF(sdqir);
   sdcr := sdcr + QUERY(q <* bag_to_set(USEDIN(sdqir[i],
          'PRODUCT_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
          'DATA_QUALITY_INSPECTION_REPORT.'+
          'INSPECTION_RESULT'))|
      'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
      'SHAPE_DATA_QUALITY_INSPECTION_CRITERION_REPORT'
       IN TYPEOF(q));
  END_REPEAT;
  REPEAT i:= 1 TO SIZEOF(sdqir);
   sdir := sdir + QUERY(q <* bag_to_set(USEDIN(sdqir[i],
          'PRODUCT_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
          'DATA_QUALITY_INSPECTION_REPORT.'+
          'INSPECTION_RESULT'))|
      'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
      'SHAPE_DATA_QUALITY_INSPECTION_INSTANCE_REPORT'
       IN TYPEOF(q));
  END_REPEAT;

  -- If there is no report, then return true.
  IF((SIZEOF(sdcr) = 0) AND (SIZEOF(sdir) = 0)) THEN
   RETURN(TRUE);
  END_IF;

  REPEAT i:= 1 TO SIZEOF(sdir);
   sdii := sdii + QUERY(q <* sdir[i].inspected_instances|
      'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
      'SHAPE_DATA_QUALITY_INSPECTION_INSTANCE_REPORT_ITEM'
       IN TYPEOF(q));
  END_REPEAT;
  REPEAT i:= 1 TO SIZEOF(sdir);
   sdei := sdei + QUERY(q <* sdir[i].inspected_instances|
      'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
      'INSTANCE_REPORT_ITEM_WITH_EXTREME_INSTANCES'
       IN TYPEOF(q));
  END_REPEAT;

  -- When reports exist, then perform the following checking process.
  -- Check representative_measured_value of
```

```
   -- shape_data_quality_inspection_criterion_report
   REPEAT i:= 1 TO SIZEOF(sdcr);
    IF(NOT(type_string IN
      TYPEOF(sdcr[i].representative_measured_value))) THEN
      RETURN(FALSE);
    END_IF;
   END_REPEAT;

   -- Check measured_value of shape_data_quality_inspection_
   -- instance_report_item
   REPEAT i:= 1 TO SIZEOF(sdii);
    IF(NOT(type_string IN
      TYPEOF(sdii[i].measured_value))) THEN
      RETURN(FALSE);
    END_IF;
   END_REPEAT;

   -- Check measured_value of extreme_instances of
   -- instance_report_item_with_extreme_instance
   REPEAT i:= 1 TO SIZEOF(sdei);
    REPEAT j:= 1 TO SIZEOF(sdei[i].extreme_instances);
      IF(NOT(type_string IN
       TYPEOF(sdei[i].extreme_instances[j].measured_value)))
      THEN
       RETURN(FALSE);
      END_IF;
    END_REPEAT;
   END_REPEAT;

   RETURN(TRUE);

  END_FUNCTION;
(*
```

<u>Argument definitions:</u>

ei: An instance of **shape_data_quality_criterion** to be checked.

type_string: A string giving acceptable type for the **measured_value_select**.

7.5.4 validate_accuracy_types

A **validate_accuracy_types** function validates the accuracy types associated with a **shape_data_-quality_criterion** instance. The operation of the function is:
1. collect types of **range** attribute of **shape_measurement_accuracy** referred from **shape_data_-quality_criterion_and_accuracy_association** that refers to **shape_data_quality_criterion** given as an argument of this function.

2. check if the collected types are exactly coincident with the type names in the type list given as an input argument of this function.
3. collect types of **range** attribute of **shape_measurement_accuracy** referred from **shape_-inspection_result_accuracy_association** that refers to **shape_data_quality_inspection_result** for the **shape_data_quality_criterion.**
4. check if the collected types are exactly coincident with the type names in the type list given as an input argument of this function.

NOTE If the type list is empty and **shape_measurement_accuracy** relating to the **shape_data_quality_-criterion** exists, this function returns false.

EXPRESS specification:

```
*)
 FUNCTION validate_accuracy_types
   (ei:shape_data_quality_criterion;
    type_list:BAG OF STRING):LOGICAL;
    LOCAL
      sdqca: BAG OF
       SHAPE_DATA_QUALITY_CRITERION_AND_ACCURACY_ASSOCIATION:=[];
      sma: BAG OF SHAPE_MEASUREMENT_ACCURACY:=[];
      sdqir: SET OF
       SHAPE_DATA_QUALITY_INSPECTION_RESULT:=[];
      scraa: BAG OF
       SHAPE_INSPECTION_RESULT_ACCURACY_ASSOCIATION:=[];
      t_logic:BOOLEAN;
    END_LOCAL;
    -- STEP-1
    -- collect types of range attribute of shape_measurement_accuracy
    -- referred from shape_data_criterion_and_accuracy_association
    -- that refers to given shape_data_quality_criterion.
    sdqca := USEDIN(ei,
           'SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.'+
           'SHAPE_DATA_QUALITY_CRITERION_AND_ACCURACY_ASSOCIATION.'+
           'SHAPE_DATA_QUALITY_CRITERION');
    IF(SIZEOF(sdqca) <> 0) THEN
      REPEAT i:= 1 TO SIZEOF(sdqca);
        sma := sma + sdqca[i].required_specific_accuracy;
      END_REPEAT;

      -- STEP-2
      -- check whether the collected types include all the type names
      -- in the type list given as an input argument of this function
      -- and they do not include any other type names.
      IF(SIZEOF(type_list) < SIZEOF(sma)) THEN
       RETURN(FALSE);
      END_IF;

      REPEAT i:= 1 TO SIZEOF(sma);
```

```
      t_logic := FALSE;
      REPEAT j:= 1 TO SIZEOF(type_list);
        IF(type_list[j] IN value_limit_type(sma[i].range)) THEN
          t_logic := TRUE;
          ESCAPE;
        END_IF;
      END_REPEAT;
      IF(NOT(t_logic)) THEN
        RETURN(FALSE);
      END_IF;
    END_REPEAT;
  END_IF;

  -- STEP-3
  -- collect types of range attribute of shape_measurement_accuracy
  -- referred from shape_inspection_result_accuracy_association
  -- that refers to shape_data_quality_inspection_result
  -- for the given shape_data_quality_criterion.
  sdqir := QUERY(q <* bag_to_set(USEDIN(ei,
             'PRODUCT_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
             'DATA_QUALITY_INSPECTION_RESULT.'+
             'CRITERION_INSPECTED'))|
        'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
        'SHAPE_DATA_QUALITY_INSPECTION_RESULT'
         IN TYPEOF(q));
  IF(SIZEOF(sdqir) = 0) THEN
    RETURN(TRUE);
  END_IF;

  REPEAT i:= 1 TO SIZEOF(sdqir);
    scraa := USEDIN(sdqir[i],
          'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
          'SHAPE_INSPECTION_RESULT_ACCURACY_ASSOCIATION.'+
          'SHAPE_DATA_QUALITY_RESULT');
    IF(SIZEOF(scraa) <> 0) THEN
      sma := [];
      REPEAT j:= 1 TO SIZEOF(scraa);
        sma := sma + scraa[j].applied_specific_accuracy;
      END_REPEAT;

      -- STEP-4
      -- check whether the collected types include all the type
      -- names in the type list given as an input argument of
      -- this function and they do not include any other type names.
      IF(SIZEOF(type_list) < SIZEOF(sma)) THEN
        RETURN(FALSE);
      END_IF;
```

```
      REPEAT j:= 1 TO SIZEOF(sma);
        t_logic := FALSE;
        REPEAT k:= 1 TO SIZEOF(type_list);
          IF(type_list[k] IN value_limit_type(sma[j].range)) THEN
            t_logic := TRUE;
            ESCAPE;
          END_IF;
        END_REPEAT;
        IF(NOT(t_logic)) THEN
          RETURN(FALSE);
        END_IF;
      END_REPEAT;
    END_IF;
  END_REPEAT;

  RETURN(TRUE);
 END_FUNCTION;
(*
```

Argument definitions:

ei: An instance of **shape_data_quality_criterion** to be checked.

type_list: A bag of string giving acceptable types.

7.5.5 value_limit_type

A **value_limit_type** function returns the names of the types of **value_component**s appearing within **item_element** of given **shape_data_quality_value_limit_type_select** if it is of type **shape_data_-quality_value_range**, and returns the names of the types of **value_component**s of **value_-representation_item** if it is of type **shape_data_quality_value_limit**.

EXPRESS specification:

```
*)
 FUNCTION value_limit_type
  (iv:shape_data_quality_value_limit_type_select):SET OF STRING;
  LOCAL
   retval : SET OF STRING := [];
  END_LOCAL;
  IF(('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.'
    +'SHAPE_DATA_QUALITY_VALUE_RANGE') IN TYPEOF(iv)) THEN
   retval := TYPEOF(iv\compound_representation_item.
                item_element[1].value_component);
  END_IF;
  IF(('SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.'
    +'SHAPE_DATA_QUALITY_VALUE_LIMIT') IN TYPEOF(iv)) THEN
   retval := TYPEOF(iv\measure_representation_item.value_component);
```

```
    END_IF;
    RETURN(retval);
  END_FUNCTION;
(*
```

Argument definitions:

iv: An instance of **shape_data_quality_value_limit_type_select**

retval(Return): A SET of string containing all the names of the types of **value_component**s appearing within **item_element** of **iv** if it is of type **shape_data_quality_value_range**, and all the names of the types of **value_component**s of **iv** if it is of type **shape_data_quality_value_limit**.

EXPRESS specification:

```
*)
END_SCHEMA; -- end shape_data_quality_criteria_schema
(*
```

8 Shape data quality inspection result

8.1 Introduction

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

EXPRESS specification:

```
*)
SCHEMA SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA;
 REFERENCE FROM GEOMETRIC_MODEL_SCHEMA; -- ISO 10303-42
 REFERENCE FROM GEOMETRY_SCHEMA; -- ISO 10303-42
 REFERENCE FROM MEASURE_SCHEMA;                -- ISO 10303-41
 REFERENCE FROM PRODUCT_DATA_QUALITY_CRITERIA_SCHEMA; -- ISO 10303-59
 REFERENCE FROM PRODUCT_DATA_QUALITY_INSPECTION_RESULT_SCHEMA;
 -- ISO 10303-59
 REFERENCE FROM PRODUCT_DEFINITION_SCHEMA   -- ISO 10303-41
  (product_definition);
 REFERENCE FROM PRODUCT_PROPERTY_DEFINITION_SCHEMA
  (product_definition_shape ,
   characterized_product_definition);           -- ISO 10303-41
 REFERENCE FROM PRODUCT_PROPERTY_REPRESENTATION_SCHEMA
  (shape_representation,                -- ISO 10303-41
   shape_representation_relationship,
   shape_definition_representation);
 REFERENCE FROM QUALIFIED_MEASURE_SCHEMA         -- ISO 10303-45
  (qualified_representation_item,
   value_qualifier,
   type_qualifier ,
   measure_representation_item);
 REFERENCE FROM REPRESENTATION_SCHEMA -- ISO 10303-43
  (founded_item_select,
   founded_item,
   representation_item,
   using_representations,
   representation_relationship,
   compound_representation_item,
   value_representation_item,
   set_representation_item);
 REFERENCE FROM SHAPE_DATA_QUALITY_CRITERIA_SCHEMA; -- ISO 10303-59
 REFERENCE FROM SUPPORT_RESOURCE_SCHEMA    -- ISO 10303-41
  (bag_to_set);
 REFERENCE FROM TOPOLOGY_SCHEMA; -- ISO 10303-42
(*
```

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

geometric_model_schema	ISO 10303-42
geometry_schema	ISO 10303-42
measure_schema	ISO 10303-41
product_data_quality_criteria_schema	clause 5 of this part of ISO 10303
product_data_quality_inspection_result_schema	clause 6 of this part of ISO 10303
product_definition_schema	ISO 10303-41
product_property_definition_schema	ISO 10303-41
product_property_representation_schema	ISO 10303-41
qualified_measure_schema	ISO 10303-45
representation_schema	ISO 10303-43
shape_data_quality_criteria_schema	clause 7 of this part of ISO 10303
support_resource_schema	ISO 10303-41
topology_schema	ISO 10303-42

NOTE 2 See annex D, Figure D.17 – D.23, for a graphical presentation of this schema.

8.2 Fundamental concepts and assumptions

This schema provides representation of inspection results of three dimensional product shape data quality. This schema is a specialization of **product_data_quality_inspection_result_schema** defined in clause 6 to the quality of three dimensional product shape. Therefore, all the fundamental concepts and assumptions described in 6.2 also apply.

As this schema deals with inspection result of the quality of product shape data, a relationship is established between **shape_data_quality_inspection_result_representation** that is a type of **data_-quality_inspection_result_representation** defined in **product_data_quality_inspection_result_-schema** and **shape_representation** representing the inspected product shape data by **shape_data_-quality_inspected_shape_and_result_relationship**.

Geometric data newly created for representing inspection results, such as points where quality defect was detected, shall be created in the context of the inspected product shape data.

A **shape_data_quality_inspection_criterion_report** is a type of **data_quality_inspection_-criterion_report** (see clause 6.4.5) that represents a summary inspection result based on a specific shape data quality criterion. It specifies how many entities were inspected, and how many among them were judged to have a quality defect as judged by this criterion. It also specifies a **representative_-measured_value** that is a typical **measured_value** resulting from the inspection, which may be useful to judge how serious the overall quality defect is.

A **shape_data_quality_inspection_instance_report** is a type of **data_quality_inspection_-instance_report** (see clause 6.4.6) that represents the most detailed inspection results for a specific shape data quality criterion. Two or more instances will in general be inspected or reported as having a quality defect. The concept corresponding to one inspected instance or pair of instances is **shape_data_quality_inspection_instance_report_item**, a type of **data_quality_inspection_-instance_report_item** (see clause 6.4.7). An instance of this entity is created for each instance inspected or pair of instances inspected, and it includes information concerning the shape element types inspected and the corresponding measured values. If it is required to capture precisely where an extreme measured value was detected, **instance_report_item_with_extreme_instances** shall be used. This is a type of **shape_data_quality_inspection_instance_report_item** that provides the required location information.

NOTE The type of shape element to be measured and the type of shape element to represent the location of quality defect are controlled by each subtype of **shape_data_quality_criterion** defined in **shape_data_quality-_criteria_schema**. The number of **shape_data_quality_inspection_instance_report_item** instances to be generated can be controlled by the attributes of **data_quality_report_request** defined in **product_data_-quality_criteria_schema**.

Users will specify thresholds to judge if a criterion is satisfied or not, which depends on business situation. They may or may not specify the accuracy for controlling precision of numerical calculation used in the measurement. But there are many cases that the information on the accuracy that is actually applied by the measurement is useful to interpret an inspection result. For this purpose, representation of a common accuracy applied to every measurement as well as the accuracy applied to the measurement of a specific criterion is provided in this schema.

8.2.1 Representation of location information for existing quality defect

This schema allows representation of precise location information for existing quality defects.

EXAMPLE A **point_on_edge_curve** can be used to represent the location of a quality defect regarding the test on '**gap_between_edge_and_base_surface**' criterion.

NOTE In the example above, the detected **edge_curve** should be reliably identified by all checking systems but the point information should be treated as a reference to approximate location.

8.3 Shape data quality inspection result type definitions

8.3.1 measured_value_select

A **measured_value_select** allows selection of the type of measured value that shall be calculated by PDQ-checking software based on the measurement requirement.

EXPRESS specification:

```
*)
 TYPE measured_value_select = SELECT
   (boolean_value,
    measure_value,
    plane_angle_and_ratio_pair,
    plane_angle_and_length_pair);
 END_TYPE;
(*
```

8.3.2 boolean_value

A **boolean _value** is used for selecting Boolean measured value.

EXPRESS specification:

```
*)
 TYPE boolean_value = BOOLEAN;
 END_TYPE;
(*
```

8.3.3 location_of_extreme_value_select

A **location_of_extreme_value_select** allows selection of the type of location representing where the extreme value is measured.

NOTE The extreme value may be the measured value that causes quality defect, or the measured value that is close to the threshold though not within the specified range. The extreme value is represented by the attribute **measured_value** of an entity **extreme_instance** referred from the **instance_report_item_with_extreme_-instances**.

EXPRESS specification:

```
*)
 TYPE location_of_extreme_value_select = SELECT
   (inspected_shape_element_select,
    b_spline_surface_knot_locator,
    b_spline_curve_knot_locator,
    b_spline_curve_segment,
    b_spline_surface_strip,
    b_spline_surface_patch,
    composite_curve_transition_locator,
    rectangular_composite_surface_transition_locator,
    boundary_curve_of_b_spline_or_rectangular_composite_surface,
    oriented_edge,
    face_bound,
    point_on_edge_curve,
    point_on_face_surface,
    subedge,
    subface);
 END_TYPE;
(*
```

8.3.4 inspected_shape_element_select

An **inspected_shape_element_select** allows selection of the shape element type to be inspected that is clarified in each measurement requirement.

EXPRESS specification:

```
*)
 TYPE inspected_shape_element_select = SELECT
   (edge_loop,
    edge_curve,
    face_surface,
    curve,
    surface,
    vertex_point,
    closed_or_open_shell_select,
    b_spline_or_composite_curve_select,
    b_spline_or_rectangular_composite_surface_select,
    point,
    connected_face_set,
    manifold_solid_brep,
    direction,
    placement);
 END_TYPE;
(*
```

8.3.5 closed_or_open_shell_select

A **closed_or_open_shell_select** allows selection of the type of topological dimensionality-2 shell either **closed_shell** or **open_shell**.

NOTE **closed_shell** and **open_shell** are defined in **topology_schema** of ISO 10303-42.

EXPRESS specification:

```
*)
 TYPE closed_or_open_shell_select = SELECT
   (closed_shell,
    open_shell);
 END_TYPE;
(*
```

8.3.6 b_spline_or_composite_curve_select

A **b_spline_or_composite_curve_select** allows selection of either **b_spline_curve** or **composite_-curve**.

NOTE **b_spline_curve** and **composite_curve** are defined in **geometry_schema** of ISO 10303-42.

EXPRESS specification:

```
*)
```

```
 TYPE b_spline_or_composite_curve_select = SELECT
   (b_spline_curve,
    composite_curve);
 END_TYPE;
(*
```

8.3.7 b_spline_or_rectangular_composite_surface_select

A **b_spline_or_rectangular_composite_surface_select** allows selection of either **b_spline_surface** or **rectangular_composite_surface**.

NOTE **b_spline_surface** and **rectangular_composite_surface** are defined in **geometry_schema** of ISO 10303-42.

EXPRESS specification:

```
*)
 TYPE b_spline_or_rectangular_composite_surface_select = SELECT
   (b_spline_surface,
    rectangular_composite_surface);
 END_TYPE;
(*
```

8.3.8 u_or_v_parameter

An **u_or_v_parameter** allows selection of a parameter representing a surface.

EXPRESS specification:

```
*)
 TYPE u_or_v_parameter = ENUMERATION OF
   (U_PARAMETER,
    V_PARAMETER);
 END_TYPE;
(*
```

8.3.9 surface_boundary_type

A **surface_boundary_type** allows selection of a surface parameter terminal.

EXPRESS specification:

```
*)
 TYPE surface_boundary_type = ENUMERATION OF
   (U_PARAMETER_MIN,
    V_PARAMETER_MIN,
    U_PARAMETER_MAX,
    V_PARAMETER_MAX);
```

```
 END_TYPE;
(*
```

8.4 shape data quality inspection result entity definitions

8.4.1 shape_data_quality_inspection_result_representation

A **shape_data_quality_inspection_result_representation** is a type of **data_quality_inspection_-result_representation** defined in **product_data_quality_inspection_result_schema**, and represents all the inspection results for an inspected product data against a set of shape data quality criteria.

EXPRESS specification:

```
*)
 ENTITY shape_data_quality_inspection_result_representation
  SUBTYPE OF(data_quality_inspection_result_representation);
  WHERE
   WR1 : SIZEOF( QUERY( q <* SELF\representation.items|
                    'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.' +
                    'SHAPE_DATA_QUALITY_INSPECTION_RESULT'
                    IN TYPEOF(q))) > 0;
   WR2 : 'GEOMETRY_SCHEMA.GEOMETRIC_REPRESENTATION_CONTEXT' IN
         TYPEOF(SELF\representation.context_of_items);
   WR3 : SIZEOF(USEDIN(SELF,
          'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.'+
          'SHAPE_DATA_QUALITY_INSPECTED_SHAPE_AND_RESULT_RELATIONSHIP.'+
          'REP_2'))>0;
 END_ENTITY;
(*
```

Formal propositions:

WR1: At least one instance of **representation_item** that belongs to a **shape_data_quality_-inspection_result_representation** shall be of type **shape_data_quality_inspection_result**.

WR2: The context of this entity shall be of type **geometric_representation_context** as defined in **geometry_schema** of ISO 10303-42.

WR3: At least one instance of **shape_data_quality_inspected_shape_and_result_relationship** associated with this entity shall exist.

8.4.2 shape_inspection_result_representation_with_accuracy

A **shape_inspection_result_representation_with_accuracy** is a type of **shape_data_quality_-inspection_result_representation** that has the information of common accuracy applied to all the measurements.

EXPRESS specification:

```
*)
 ENTITY shape_inspection_result_representation_with_accuracy
  SUBTYPE OF(shape_data_quality_inspection_result_representation);
   applied_general_accuracy : SET [1:?] OF shape_measurement_accuracy;
 END_ENTITY;
(*
```

Attribute definitions:

applied_general_accuracy: Common accuracy applied to all the measurements for inspecting specified product data against specified set of criteria.

NOTE For representing the accuracy applied to the measurement against a particular criterion, the attribute **applied_specific_accuracy** associated by **shape_inspection_criterion_report_accuracy_association** is provided and it supersedes the accuracy defined by this entity if present.

8.4.3 shape_data_quality_inspected_shape_and_result_relationship

A **shape_data_quality_inspected_shape_and_result_relationship** is a type of **shape_-representation_relationship** as defined in ISO 10303-43 and it represents the relationship between **shape_data_quality_inspection_result_representation** representing the inspection results and **shape_representation** representing the inspection target product shape data.

EXPRESS specification:

```
*)
 ENTITY shape_data_quality_inspected_shape_and_result_relationship
  SUBTYPE OF(shape_representation_relationship);
   SELF\representation_relationship.rep_1 : shape_representation;
   SELF\representation_relationship.rep_2 :
shape_data_quality_inspection_result_representation;
  WHERE
   WR1 : SIZEOF(
       using_product_definition_of_shape_representation(SELF.rep_1)
       *
       bag_to_set(inspected_product_definition(SELF.rep_2))
       ) > 0;
   WR2 : SELF.rep_1\representation.context_of_items :=:
       SELF.rep_2\representation.context_of_items;
 END_ENTITY;
(*
```

Attribute definitions:

SELF\representation_relationship.rep_1: **shape_representation** representing the product shape data inspected.

SELF\representation_relationship.rep_2: **shape_data_quality_inspection_result_representation** representing the result of data quality inspection for the product shape data represented by **rep_1** attribute.

Formal propositions:

WR1: There shall be at least one common **product_definition** among those retrieved starting from the attribute **rep_2** and those where the **shape_representation** specified by the attribute **rep_1** belongs.

NOTE The relationship between **shape_data_quality_inspection_result_representation** and inspected **product_definition** is captured by the two relationship entities defined in **product_data_quality_-definition_schema**, namely, **product_data_and_data_quality_relationship** which relates **product_definition** and **data_quality_definition** and **data_quality_definition_representation** which relates **data_quality_-definition** and **shape_data_quality_inspection_result_representation**.

WR2: The context of **shape_representation** specified by the attribute **rep_1** and the context of **shape_data_quality_inspection_result_representation** specified by the attribute **rep_2** shall be identical.

8.4.4 shape_data_quality_inspection_result

A **shape_data_quality_inspection_result** is a type of **data_quality_inspection_result** defined in **product_data_quality_inspection_result_schema** and represents the inspection result for a specific criterion on shape data quality.

EXPRESS specification:

```
*)
 ENTITY shape_data_quality_inspection_result
  SUBTYPE OF(data_quality_inspection_result);
  WHERE
   WR1 : 'SHAPE_DATA_QUALITY_CRITERIA_SCHEMA.'+
        'SHAPE_DATA_QUALITY_CRITERION' IN TYPEOF
        (SELF\data_quality_inspection_result.criterion_inspected);
 END_ENTITY;
(*
```

Formal propositions:

WR1: The criterion specified by the attribute **criterion_inspected** against which the inspection is performed shall be of type **shape_data_quality_criterion**.

8.4.5 shape_data_quality_inspection_criterion_report

A **shape_data_quality_inspection_criterion_report** is a type of **data_quality_inspection_-criterion_report** and represents summarized information on the result of inspection for a specific shape data quality criterion.

EXPRESS specification:

```
*)
 ENTITY shape_data_quality_inspection_criterion_report
   SUBTYPE OF(data_quality_inspection_criterion_report);
     representative_measured_value : measured_value_select;
 END_ENTITY;
(*
```

Attribute definitions:

representative_measured_value: **measured_value_select** for selecting a representative extreme value from among the measured values for the criterion.

8.4.6 shape_inspection_result_accuracy_association

A **shape_inspection_result_accuracy_association** represents the relationship between **shape_data-_quality_inspection_result** and the accuracy applied by the inspection.

EXPRESS specification:

```
*)
 ENTITY shape_inspection_result_accuracy_association;
     applied_specific_accuracy : shape_measurement_accuracy;
     shape_data_quality_result : shape_data_quality_inspection_result;
 END_ENTITY;
(*
```

Attribute definitions:

applied_specific_accuracy: **shape_measurement_accuracy** applied by the inspection specifically for the criterion.

NOTE 1 Applied accuracy that is not specific to the criterion but common to all the criteria inspected is provided by the attribute **applied_general_accuracy** of the entity **shape_inspection_result_representation_-with_accuracy**.

NOTE 2 If **applied_general_accuracy** is specified by the entity **shape_inspection_result_representation_-with_accuracy** and **applied_specific_accuracy** is also specified by the instantiation of this entity, then the latter supersedes the former for the criterion.

shape_data_quality_result: **shape_data_quality_inspection_result** that represents the result of inspection for the criterion.

8.4.7 shape_data_quality_inspection_instance_report

A **shape_data_quality_inspection_instance_report** is a type of **data_quality_inspection_instance-_report** and represents detailed inspection report of the product shape data for a specific criterion. It contains a list of **shape_data_quality_inspection_instance_report_item**s, each representing inspection result of an inspected shape element or a pair of shape elements.

EXPRESS specification:

```
*)
 ENTITY shape_data_quality_inspection_instance_report
  SUBTYPE OF(data_quality_inspection_instance_report);
   SELF\data_quality_inspection_instance_report.inspected_instances :
        LIST [1:?] OF shape_data_quality_inspection_instance_report_item;
 END_ENTITY;
(*
```

Attribute definitions:

SELF\data_quality_inspection_instance_report.inspected_instances: A list of **shape_data_-quality_inspection_instance_report_item**s.

NOTE Expected quantity and order of **shape_data_quality_inspection_instance_report_item**s in the list of this attribute can be controlled by **data_quality_report_request** as defined in **product_data_quality_criteria_-schema**.

8.4.8 shape_data_quality_inspection_instance_report_item

A **shape_data_quality_inspection_instance_report_item** is a type of **data_quality_inspection_-instance_report_item** and represents the detailed information obtained for each inspected element or a pair of elements of product shape data.

EXPRESS specification:

```
*)
 ENTITY shape_data_quality_inspection_instance_report_item
  SUBTYPE OF(data_quality_inspection_instance_report_item);
   inspected_elements : SET [1:?] OF inspected_shape_element_select;
   measured_value    : measured_value_select;
 END_ENTITY;
 (*
```

Attribute definitions:

inspected_elements: One or more **inspected_shape_element_select** that represent data types of the shape elements inspected.

measured_value: **measured_value_select** that represents the measured value obtained by the inspection.

NOTE The types of **measured_value** and **inspected_elements** are specified in each **shape_data_quality_-criterion** entity defined in **shape_data_quality_criteria_schema**.

8.4.9 instance_report_item_with_extreme_instances

An **instance_report_item_with_extreme_instances** is a type of **shape_data_quality_inspection-_instance_report_item** and has the information on pairs of extreme value and its location.

EXPRESS specification:

```
*)
 ENTITY instance_report_item_with_extreme_instances
   SUBTYPE OF(shape_data_quality_inspection_instance_report_item);
     extreme_instances : LIST [1:?] OF extreme_instance;
 END_ENTITY;
(*
```

Attribute definitions:

extreme_instances: A list of **extreme_instance**s that is a pair of extreme value and its location detected by the measurement. The extreme value is the measured value that causes quality defect, namely satisfaction of the criterion concerned, or the measured value that is close to the threshold though the criterion is not satisfied.

8.4.10 extreme_instance

An **extreme_instance** represents a pair of extreme measured value and related location information where it was detected.

EXPRESS specification:

```
*)
 ENTITY extreme_instance;
    locations_of_extreme_value :
      SET [1:?] OF location_of_extreme_value_select;
    measured_value         : measured_value_select;
 END_ENTITY;
(*
```

Attribute definitions:

locations_of_extreme_value: One or more **location_of_extreme_value_select** that represents the location where the extreme value was detected.

measured_value: **measured_value_select** that represents the extreme value obtained by the measurement.

EXAMPLE When a gap between edge and base surface is measured, **locations_of_extreme_value** includes two points, one is a point on the edge and the other is a point on the base surface. The extreme value represented by the **measured_value** is the distance between these two points.

NOTE Each **shape_data_quality_criterion** entity defined in **shape_data_quality_criteria_schema** has a rule to specify what type of **locations_of_extreme_value** to require.

8.4.11 point_on_edge_curve

A **point_on_edge_curve** is a type of **point_on_curve** as defined in ISO 10303-42 and represents the location on the identified **edge_curve** where the extreme value was measured.

EXPRESS specification:

```
*)
 ENTITY point_on_edge_curve
  SUBTYPE OF(point_on_curve);
   basis_edge_curve : edge_curve;
  DERIVE
   SELF\point_on_curve.basis_curve :
     curve := SELF.basis_edge_curve.edge_geometry;
  WHERE
   WR1 : SIZEOF(
        using_representations(SELF.basis_edge_curve) *
        relating_shape_representations_of_extreme_values(SELF))>0;
 END_ENTITY;
(*
```

Attribute definitions:

basis_edge_curve: **edge_curve** on which the measurement was performed.

SELF\point_on_curve.basis_curve: **curve** that is the basis curve to which the parameter value of the location relates. This is derived from the **edge_geometry** of **basis_edge_curve**.

Formal propositions:

WR1: **shape_representation** that uses the **edge_curve** specified in **basis_edge_curve** of this entity shall be the **shape_representation** of the **product_definition** that is related to **shape_data_quality_-inspection_result_representation** that has **instance_report_item_with_extreme_instances** containing an instance of this entity as its location of extreme value.

8.4.12 point_on_face_surface

A **point_on_face_surface** is a type of **point_on_surface** as defined in ISO 10303-42 and represents the location on the identified **face_surface** where the extreme value was measured.

EXPRESS specification:

```
*)
 ENTITY point_on_face_surface
  SUBTYPE OF(point_on_surface);
   basis_face_surface : face_surface;
  DERIVE
   SELF\point_on_surface.basis_surface :
     surface := SELF.basis_face_surface.face_geometry;
  WHERE
   WR1 : SIZEOF(
       using_representations(SELF.basis_face_surface) *
       relating_shape_representations_of_extreme_values(SELF))>0;
 END_ENTITY;
(*
```

Attribute definitions:

basis_face_surface: **face_surface** on which the measurement was performed.

SELF\point_on_surface.basis_surface: **surface** that is the basis surface to which the parameter value of the location relates. This is derived from the **face_geometry** of **basis_face_surface**.

Formal propositions:

WR1: **shape_representation** that uses the **face_surface** specified in **basis_face_- surface** shall be the **shape_representation** of the **product_definition** that is related to **shape_data_quality_inspection_-result_representation** that has **instance_report_item_with_extreme_instances** containing an instance of this entity as its location of extreme value.

8.4.13 plane_angle_and_ratio_pair

A **plane_angle_and_ratio_pair** represents the measured value that consists of a **plane_angle** and a ratio of curvature radius change. This entity is used to represent the difference of curvature which requires the difference between directions of principal vector as well as the difference of curvature radius values. Let r1 and r2 be curvature radii to be compared, ratio of curvature radius change is given by (2|r1-r2|)/(|r1|+|r2|).

EXPRESS specification:

```
*)
 ENTITY plane_angle_and_ratio_pair
```

```
  SUBTYPE OF(founded_item);
    plane_angle_value : plane_angle_measure;
    ratio_value : ratio_measure;
 END_ENTITY;
(*
```

Attribute definitions:

plane_angle_value: **plane_angle_measure**.

ratio_value: **ratio_measure,** ratio of curvature radius change.

8.4.14 plane_angle_and_length_pair

A **plane_angle_and_length_pair** represents the measured value that consists of a **plane_angle** and a distance. This entity is used to represent the difference of two **placement**s.

EXPRESS specification:

```
*)
 ENTITY plane_angle_and_length_pair
  SUBTYPE OF(founded_item);
    plane_angle_value : plane_angle_measure;
    length_value    : length_measure;
 END_ENTITY;
(*
```

Attribute definitions:

plane_angle_value: **plane_angle_measure.**

length_value: **length_measure**.

8.4.15 b_spline_surface_knot_locator

A **b_spline_surface_knot_locator** represents a location by specifying an index of the knot array and a direction in the parametric space of the identified **b_spline_surface**. This entity is used to represent the location where the extreme value was measured.

NOTE The **b_spline_surface** and knot array are defined in **geometry_schema** of ISO 10303-42.

EXPRESS specification:

```
*)
 ENTITY b_spline_surface_knot_locator
  SUBTYPE OF(founded_item);
    basis_surface   : b_spline_surface;
    knot_index     : INTEGER;
```

```
    surface_parameter : u_or_v_parameter;
   WHERE
    WR1 : SIZEOF(
        using_representations(SELF.basis_surface) *
        relating_shape_representations_of_extreme_values(SELF))>0;
 END_ENTITY;
(*
```

Attribute definitions:

basis_surface: **b_spline_surface** on which the measurement was performed.

knot_index: Index of the knot array of the **basis_surface** where the extreme value was measured.

surface_parameter: **u_or_v_parameter** where the **knot_index** corresponds.

Formal propositions:

WR1: **shape_representation** that uses the **b_spline_surface** specified in **basis_surface** shall be the **shape_representation** of the **product_definition** that is related to **shape_data_quality_inspection_-result_representation** that has **instance_report_item_with_extreme_instances** containing an instance of this entity as its location of extreme value.

8.4.16 b_spline_surface_patch

A **b_spline_surface_patch** represents a location by specifying a patch with *u*, *v* knot indices on the identified **b_spline_surface**. This entity is used to represent where the extreme value was measured.

NOTE The **b_spline_surface** and knot array are defined in **geometry_schema** of ISO 10303-42.

EXPRESS specification:

```
*)
 ENTITY b_spline_surface_patch
   SUBTYPE OF(founded_item);
    basis_surface    : b_spline_surface;
    u_start_knot_index : INTEGER;
    v_start_knot_index : INTEGER;
   WHERE
    WR1 : SIZEOF(
        using_representations(SELF.basis_surface) *
        relating_shape_representations_of_extreme_values(SELF))>0;
 END_ENTITY;
(*
```

Attribute definitions:

basis_surface: **b_spline_surface** on which the measurement was performed.

u_start_knot_index: start index of the surface patch in the knot array in *u* parameter of **basis_surface** where the extreme value was measured.

v_start_knot_index: start index of the surface patch in the knot array in *v* parameter of **basis_surface** where the extreme value was measured.

NOTE If the **u_start_knot_index** is *i* and the **v_start_knot_index** is *j*, then it shall be understood that the portion of the surface is represented where *u* parameter is larger than or equal to i-th knot and is smaller than or equal to i+1-th knot, and *v* parameter is larger than or equal to j-th knot and is smaller than or equal to j+1-th knot.

Formal propositions:

WR1: **shape_representation** that uses the **b_spline_surface** specified in **basis_surface** shall be the **shape_representation** of the **product_definition** that is related to **shape_data_quality_inspection_-result_representation** that has **instance_report_item_with_extreme_instances** containing an instance of this entity as its location of extreme value.

8.4.17 b_spline_curve_knot_locator

A **b_spline_curve_knot_locator** represents a location by specifying an index of the knot array of the identified **b_spline_curve**. This entity is used to represent the location where the extreme value was measured.

NOTE The **b_spline_curve** and knot array are defined in **geometry_schema** of ISO 10303-42.

EXPRESS specification:

```
*)
 ENTITY b_spline_curve_knot_locator
  SUBTYPE OF(founded_item);
   basis_curve : b_spline_curve;
   knot_index  : INTEGER;
  WHERE
   WR1 : SIZEOF(
        using_representations(SELF.basis_curve) *
        relating_shape_representations_of_extreme_values(SELF))>0;
 END_ENTITY;
(*
```

Attribute definitions:

basis_curve: **b_spline_curve** on which the measurement was performed.

knot_index: start index of the segment in the knot array of the **basis_curve** where the extreme value was measured.

Formal propositions:

WR1: **shape_representation** that uses the **b_spline_curve** specified in **basis_curve** of shall be the **shape_representation** of the **product_definition** that is related to **shape_data_quality_inspection_-result_representation** that has **instance_report_item_with_- extreme_instances** containing an instance of this entity as its location of extreme value.

8.4.18 b_spline_curve_segment

A **b_spline_curve_segment** represents a location by specifying a knot span of the identified **b_spline_curve** where parameter is larger than or equal to (**start_knot_index**)th knot and less or equal to (**start_knot_index**+1)th knot. This entity is used to represent the location where the extreme value was measured.

NOTE The **b_spline_curve** and knot array are defined in **geometry_schema** of ISO 10303-42.

EXPRESS specification:

```
*)
 ENTITY b_spline_curve_segment
  SUBTYPE OF(founded_item);
   basis_curve     : b_spline_curve;
   start_knot_index : INTEGER;
  WHERE
   WR1 : SIZEOF(
        using_representations(SELF.basis_curve) *
        relating_shape_representations_of_extreme_values(SELF))>0;
 END_ENTITY;
(*
```

Attribute definitions:

basis_curve: **b_spline_curve** on which the measurement was performed.

start_knot_index: the start index of the knot array of the **basis_curve** where the extreme value was measured. If the **start_knot_index** is *i*, then it shall be understood that portion of the curve is represented where parameter is larger than *i*-th knot and smaller than *i*+1-th knot of the **basis_curve**.

Formal propositions:

WR1: **shape_representation** that uses the **b_spline_curve** specified in **basis_curve** shall be the **shape_representation** of the **product_definition** that is related to **shape_data_quality_inspection_-result_representation** that has **instance_report_item_with_extreme_instances** containing an instance of this entity as its location of extreme value.

8.4.19 b_spline_surface_strip

A **b_spline_surface_strip** represents a location by specifying a surface strip of the identified **b_spline_surface**. This entity is used to represent the location where the extreme value was measured.

EXPRESS specification:

```
*)
 ENTITY b_spline_surface_strip
   SUBTYPE OF(founded_item);
    basis_surface    : b_spline_surface;
    start_knot_index : INTEGER;
    surface_parameter : u_or_v_parameter;
   WHERE
    WR1 : SIZEOF(
        using_representations(SELF.basis_surface) *
        relating_shape_representations_of_extreme_values(SELF))>0;
 END_ENTITY;
(*
```

Attribute definitions:

basis_surface: **b_spline_surface** on which the measurement was performed.

start_knot_index: start index of the surface strip in the knot array in *u* or *v* parameter of **basis_surface** where the extreme value was measured. If the **start_knot_index** is *i*, then it shall be understood that portion of the surface is represented where the *u* or *v* parameter specified by **surface_parameter** is larger than *i*-th knot and smaller than *i*+1-th knot of the **basis_surface**.

surface_parameter: **u_or_v_parameter** where the **start_knot_index** corresponds.

NOTE The **b_spline_surface** and knot array are defined in **geometry_schema** of ISO 10303-42.

Formal propositions:

WR1: **shape_representation** that uses the **b_spline_surface** specified in **basis_surface** shall be the **shape_representation** of the **product_definition** that is related to **shape_data_quality_inspection_-result_representation** that has **instance_report_item_with_extreme_instances** containing an instance of this entity as its location of extreme value.

8.4.20 composite_curve_transition_locator

A **composite_curve_transition_locator** represents a location by specifying an index of connecting point between two segments in the identified **composite_curve**. This entity is used to represent the location where the extreme value was measured.

NOTE The **composite_curve** is defined in **geometry_schema** of ISO 10303-42.

EXPRESS specification:

```
*)
 ENTITY composite_curve_transition_locator
```

```
   SUBTYPE OF(founded_item);
     basis_curve    : composite_curve;
     transition_index : INTEGER;
   WHERE
     WR1 : SIZEOF(
         using_representations(SELF.basis_curve) *
         relating_shape_representations_of_extreme_values(SELF))>0;
 END_ENTITY;
(*
```

Attribute definitions:

basis_curve: **composite_curve** on which the measurement was performed.

transition_index: index of the connecting point of **basis_curve** where the extreme value was measured. If **transition_index** is j, then it shall be understood that the connecting point between j-th segment and j+1-th segment is represented.

Formal propositions:

WR1: **shape_representation** that uses the **composite_curve** specified in **basis_curve** shall be the **shape_representation** of the **product_definition** that is related to **shape_data_quality_inspection_-result_representation** that has **instance_report_item_with_extreme_instances** containing an instance of this entity as its location of extreme value.

8.4.21 rectangular_composite_surface_transition_locator

A **rectangular_composite_surface_transition_locator** represents a location by specifying an index of connecting segments between two patches in the identified **rectangular_composite_surface**. This entity is used to represent the location where the extreme value was measured.

NOTE The **rectangular_composite_curve** is defined in **geometry_schema** of ISO 10303-42.

EXPRESS specification:

```
*)
 ENTITY rectangular_composite_surface_transition_locator
   SUBTYPE OF(founded_item);
     basis_surface    : rectangular_composite_surface;
     surface_parameter  : u_or_v_parameter;
     u_transition_index : INTEGER;
     v_transition_index : INTEGER;
   WHERE
     WR1 : SIZEOF(
         using_representations(SELF.basis_surface) *
         relating_shape_representations_of_extreme_values(SELF))>0;
 END_ENTITY;
(*
```

Attribute definitions:

basis_surface: **rectangular_composite_surface** on which the measurement was performed.

surface_parameter: **u_or_v_parameter** that indicates the boundary segment of the specified patch.

u_transition_index: index of the patches in *u*-direction of **basis_surface** where the extreme value was measured.

v_transition_index: index of the patches in *v*-direction of **basis_surface** where the extreme value was measured.

NOTE If the **u_transition_index** is *i* and the **v_transition_index** is *j* and the **surface_parameter** is **U_PARAMETER**, then it shall be understood that the transition between **segments**[*i*][*j*] and **segments**[*i*+1][*j*] is represented. If the **u_transition_index** is *i* and the **v_transition_index** is *j* and the **surface_parameter** is **V_PARAMETER**, then it shall be understood that the transition between **segments**[*i*][*j*] and **segments**[*i*][*j*+1] is represented.

Formal propositions:

WR1: **shape_representation** that uses the **rectangular_composite_surface** specified in **basis_surface** shall be the **shape_representation** of the **product_definition** that is related to **shape_data_quality_inspection_result_representation** that has **instance_report_item_with_extreme_instances** containing an instance of this entity as its location of extreme value.

8.4.22 boundary_curve_of_b_spline_or_rectangular_composite_surface

A **boundary_curve_of_b_spline_or_rectangular_composite_surface** represents a location by specifying a boundary of the parametric space of the identified surface. The entity is used to represent the location where the extreme value is measured.

EXPRESS specification:

```
*)
 ENTITY boundary_curve_of_b_spline_or_rectangular_composite_surface
  SUBTYPE OF(founded_item);
   basis_surface : b_spline_or_rectangular_composite_surface_select;
   boundary_type : surface_boundary_type;
  WHERE
   WR1 : SIZEOF(
        using_representations(SELF.basis_surface) *
        relating_shape_representations_of_extreme_values(SELF))>0;
 END_ENTITY;
(*
```

Attribute definitions:

basis_surface: **b_spline_surface** or **rectangular_composite_surface** on which the measurement was performed.

boundary_type: type of the boundary of the parametric space of **basis_surface** where the extreme value was measured.

Formal propositions:

WR1: The **shape_representation** that uses the **b_spline_surface** or **rectangular_composite_surface** specified in **basis_surface** shall be the **shape_representation** of the **product_definition** that is related to **shape_data_quality_inspection_result_representation** that has **instance_report_item_-with_extreme_instances** containing an instance of this entity as its location of extreme value.

8.5 Shape data quality inspection result function definitions

8.5.1 relating_shape_representations_of_extreme_values

The function **relating_shape_representations_of_extreme_values** finds all **shape_representation**s on which inspection was performed and **instance_report_item_with_extreme_instances** was created. An instance of **locations_of_extreme_value_select** is given as an input argument of this function.

EXPRESS specification:

```
*)
 FUNCTION relating_shape_representations_of_extreme_values
 (item:location_of_extreme_value_select)
  : SET OF shape_representation;

 LOCAL
  results  : SET OF shape_representation:=[];
  exti    : SET OF extreme_instance:=[];
  ii_ei   : SET OF
      instance_report_item_with_extreme_instances:=[];
  sdiir     : SET OF
      shape_data_quality_inspection_instance_report:=[];
  sdqr    : SET OF
      shape_data_quality_inspection_result_representation:=[];
  sdisr   : SET OF
      shape_data_quality_inspected_shape_and_result_relationship:=[];
 END_LOCAL;
 -- find extreme_instances
 exti := exti +
   bag_to_set( USEDIN(item,
     'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.' +
     'EXTREME_INSTANCE.'+
     'LOCATIONS_OF_EXTREME_VALUE'));

 -- find instance_report_item_with_extreme_instances
```

```
 REPEAT i := 1 TO HIINDEX (exti);
  ii_ei := ii_ei +
   bag_to_set( USEDIN(exti[i],
     'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.' +
     'INSTANCE_REPORT_ITEM_WITH_EXTREME_INSTANCES.'+
     'EXTREME_INSTANCES'));
 END_REPEAT;

 -- find shape_data_quality_inspection_instance_report
 REPEAT i := 1 TO HIINDEX (ii_ei);
  sdiir := sdiir +
   bag_to_set( USEDIN(ii_ei[i],
     'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.' +
     'SHAPE_DATA_QUALITY_INSPECTION_INSTANCE_REPORT.'+
     'INSPECTED_INSTANCES'));
 END_REPEAT;

 -- find shape_data_quality_inspection_result_representation
 REPEAT i := 1 TO HIINDEX (sdiir);
  sdqr := sdqr +
   QUERY(i1<* bag_to_set( USEDIN(sdiir[i].inspection_result,
     'REPRESENTATION_SCHEMA.REPRESENTATION.ITEMS'))|
   'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.' +
   'SHAPE_DATA_QUALITY_INSPECTION_RESULT_REPRESENTATION'
    IN TYPEOF(i1));
 END_REPEAT;

 -- find shape_data_quality_inspected_shape_and_result_relationship
 REPEAT i := 1 TO HIINDEX (sdqr);
  sdisr := sdisr + bag_to_set(USEDIN(sdqr[i],
    'SHAPE_DATA_QUALITY_INSPECTION_RESULT_SCHEMA.' +
    'SHAPE_DATA_QUALITY_INSPECTED_SHAPE_AND_RESULT_RELATIONSHIP.'+
    'REP_2'));
 END_REPEAT;

 -- find shape_representation
 REPEAT i := 1 TO HIINDEX (sdisr);
  results := results + sdisr[i].rep_1;
 END_REPEAT;

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

<u>Argument definitions:</u>

item: An instance of **location_of_extreme_value_select** for which related **shape_representation**s to be found.

results(Return): A SET containing all instances of **shape_representation** that are referenced by **item** via **extreme_instance**, **instance_report_item_with_extreme_instances**, **shape_data_quality_-inspection_instance_report**, **shape_data_quality_inspection_result_representation** and **shape_-data_quality_inspected_shape_and_result_relationship.**

8.5.2 using_product_definition_of_shape_representation

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

<u>EXPRESS specification:</u>

```
*)
 FUNCTION using_product_definition_of_shape_representation
 (item: shape_representation) : SET OF product_definition;
 LOCAL
   local_s_d_r: SET OF shape_definition_representation := [];
   local_p_d: SET OF product_definition := [];
   i : INTEGER;
 END_LOCAL;

 -- find shape_definition_representations
   local_s_d_r  := QUERY(i1<* bag_to_set (USEDIN (item,
  'PRODUCT_PROPERTY_REPRESENTATION_SCHEMA.' +
  'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION'))|
  'PRODUCT_PROPERTY_REPRESENTATION_SCHEMA.'+
  'SHAPE_DEFINITION_REPRESENTATION'
  IN TYPEOF(i1));

 -- find product_definition_shape and product_definitions
 REPEAT i := 1 TO HIINDEX (local_s_d_r);
    IF (('PRODUCT_PROPERTY_DEFINITION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF (local_s_d_r[i].definition))
      AND
      ('PRODUCT_DEFINITION_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF
        (local_s_d_r[i].definition.definition)))
    THEN local_p_d := local_p_d + local_s_d_r[i].definition.definition;
    ELSE IF (('PRODUCT_PROPERTY_DEFINITION_SCHEMA.PROPERTY_DEFINITION' IN
      TYPEOF (local_s_d_r[i].definition))
      AND
      ('PRODUCT_DEFINITION_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF
        (local_s_d_r[i].definition.definition)))
    THEN local_p_d := local_p_d + local_s_d_r[i].definition.definition;
    END_IF;
    END_IF;
 END_REPEAT;
 RETURN (local_p_d);
```

```
 END_FUNCTION;
(*
```

Argument definitions:

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

local_p_d (Return): A SET containing all instances of **product_definition** that use **item**.

EXPRESS specification:

```
*)
END_SCHEMA; -- end shape_data_quality_inspection_result_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 short names are found in the implementation methods included in ISO 10303.

Table A.1 – Short names of entities

Entity data type names	Short names
abrupt_change_of_surface_normal	ACOSN
b_spline_curve_knot_locator	BSCKL
b_spline_curve_segment	BSCS
b_spline_surface_knot_locator	BSSKL
b_spline_surface_patch	BSSP
b_spline_surface_strip	BSSS
boundary_curve_of_b_spline_or_rectangular_composite_surface	BCOBSO
composite_curve_transition_locator	CCTL
curve_with_excessive_segments	CWES
curve_with_small_curvature_radius	CWSCR
data_quality_assessment_measurement_association	DQAMA
data_quality_assessment_specification	DQAS
data_quality_criteria_representation	DQCR
data_quality_criterion	DTQLCR
data_quality_criterion_assessment_association	DQCAA
data_quality_criterion_measurement_association	DQCMA
data_quality_definition	DTQLDF
data_quality_definition_relationship	DQDR
data_quality_definition_representation_relationship	DQDRR
data_quality_inspection_criterion_report	DQICR
data_quality_inspection_criterion_report_item	DQICRI
data_quality_inspection_instance_report	DQIIR
data_quality_inspection_instance_report_item	DQIIRI
data_quality_inspection_report	DQI0
data_quality_inspection_result	DQIR
data_quality_inspection_result_representation	DQIRR
data_quality_inspection_result_with_judgement	DQIRWJ
data_quality_measurement_requirement	DQMR
data_quality_report_measurement_association	DQRMA
data_quality_report_request	DQRR
detailed_report_request	DTRPRQ
detailed_report_request_with_number_of_data	DRRWNO
disconnected_face_set	DSFCST
discontinuous_geometry	DSCGMT

Table A.1 – (continued)

Entity data type names	Short names
edge_with_excessive_segments	EWES
entirely_narrow_face	ENNRFC
entirely_narrow_solid	ENNRSL
entirely_narrow_surface	ENNRSR
erroneous_b_spline_curve_definition	EBSCD
erroneous_b_spline_surface_definition	EBSSD
erroneous_data	ERRDT
erroneous_geometry	ERRGMT
erroneous_manifold_solid_brep	EMSB
erroneous_topology	ERRTPL
erroneous_topology_and_geometry_relationship	ETAGR
excessively_high_degree_curve	EHDC
excessively_high_degree_surface	EHDS
extreme_instance	EXTINS
extreme_patch_width_variation	EPWV
face_surface_with_excessive_patches_in_one_direction	FSWEPI
free_edge	FREDG
g1_discontinuity_between_adjacent_faces	GDBAF
g1_discontinuous_curve	G1DSCR
g1_discontinuous_surface	G1DSSR
g2_discontinuity_between_adjacent_faces	GDB0
g2_discontinuous_curve	G2DSCR
g2_discontinuous_surface	G2DSSR
gap_between_adjacent_edges_in_loop	GBAEIL
gap_between_edge_and_base_surface	GBEABS
gap_between_faces_related_to_an_edge	GBFRTA
gap_between_pcurves_related_to_an_edge	GBPRTA
gap_between_vertex_and_base_surface	GBVABS
gap_between_vertex_and_edge	GBVAE
geometric_gap_in_topology	GGIT
geometry_with_local_irregularity	GWLI
geometry_with_local_near_degeneracy	GWLND
high_degree_axi_symmetric_surface	HDASS
high_degree_conic	HGDGCN
high_degree_linear_curve	HDLC
high_degree_planar_surface	HDPS
inapt_data	INPDT
inapt_geometry	INPGMT
inapt_manifold_solid_brep	IMSB
inapt_topology	INPTPL

Table A.1 – (continued)

Entity data type names	Short names
inapt_topology_and_geometry_relationship	ITAGR
inconsistent_adjacent_face_normals	IAFN
inconsistent_curve_transition_code	ICTC
inconsistent_edge_and_curve_directions	IEACD
inconsistent_face_and_closed_shell_normals	IFACSN
inconsistent_face_and_surface_normals	IFASN
inconsistent_surface_transition_code	ISTC
indistinct_curve_knots	INCRKN
indistinct_surface_knots	INSRKN
instance_report_item_with_extreme_instances	IRIWEI
intersecting_connected_face_sets	ICFS
intersecting_loops_in_face	ILIF
intersecting_shells_in_solid	ISIS
multiply_defined_cartesian_points	MDCP
multiply_defined_curves	MLDFCR
multiply_defined_directions	MLDFDR
multiply_defined_edges	MLDFED
multiply_defined_faces	MLDFFC
multiply_defined_geometry	MLDFGM
multiply_defined_placements	MLDFPL
multiply_defined_solids	MLDFSL
multiply_defined_surfaces	MLDFSR
multiply_defined_vertices	MLDFVR
narrow_surface_patch	NRSRPT
nearly_degenerate_geometry	NRDGGM
nearly_degenerate_surface_boundary	NDSB
nearly_degenerate_surface_patch	NDSP
non_manifold_at_edge	NMAE
non_manifold_at_vertex	NMAV
non_smooth_geometry_transition_across_edge	NSGTAE
open_closed_shell	OPCLSH
open_edge_loop	OPEDLP
over_used_vertex	OVUSVR
overcomplex_geometry	OVR0
overcomplex_topology_and_geometry_relationship	OTAGR
overlapping_geometry	OVRGMT
partly_overlapping_curves	PROVCR
partly_overlapping_edges	PROVED
partly_overlapping_faces	PROVFC
partly_overlapping_solids	PROVSL
partly_overlapping_surfaces	PROVSR

Table A.1 – (continued)

Entity data type names	Short names
plane_angle_and_length_pair	PAALP
plane_angle_and_ratio_pair	PAARP
point_on_edge_curve	POEC
point_on_face_surface	POFS
product_data_and_data_quality_relationship	PDADQR
rectangular_composite_surface_transition_locator	RCSTL
self_intersecting_curve	SLINCR
self_intersecting_geometry	SLINGM
self_intersecting_loop	SLINLP
self_intersecting_shell	SLINSH
self_intersecting_surface	SLINSR
shape_criteria_representation_with_accuracy	SCRWA
shape_data_quality_assessment_by_logical_test	SDQABL
shape_data_quality_assessment_by_numerical_test	SDQABN
shape_data_quality_criteria_representation	SDQCR
shape_data_quality_criterion	SDQC
shape_data_quality_criterion_and_accuracy_association	SDQCAA
shape_data_quality_inspected_shape_and_result_relationship	SDQISA
shape_data_quality_inspection_criterion_report	SDQICR
shape_data_quality_inspection_instance_report	SDQIIR
shape_data_quality_inspection_instance_report_item	SDQ0
shape_data_quality_inspection_result	SDQIR
shape_data_quality_inspection_result_representation	SDQIRR
shape_data_quality_lower_value_limit	SDQLVL
shape_data_quality_upper_value_limit	SDQUVL
shape_data_quality_value_limit	SDQVL
shape_data_quality_value_range	SDQVR
shape_inspection_result_accuracy_association	SIRAA
shape_inspection_result_representation_with_accuracy	SIRRWA
shape_measurement_accuracy	SHMSAC
shape_summary_request_with_representative_value	SSRWRV
short_length_curve	SHLNCR
short_length_curve_segment	SLCS
short_length_edge	SHLNED
small_area_face	SMARFC
small_area_surface	SMARSR
small_area_surface_patch	SASP
small_volume_solid	SMVLSL
software_for_data_quality_check	SFDQC
solid_with_excessive_number_of_voids	SWENOV
solid_with_wrong_number_of_voids	SWWNOV
steep_angle_between_adjacent_edges	SABAE

Table A.1 – (continued)

Entity data type names	Short names
steep_angle_between_adjacent_faces	SABAF
steep_geometry_transition_across_edge	SGTAE
summary_report_request	SMRPRQ
surface_with_excessive_patches_in_one_direction	SWEPIO
surface_with_small_curvature_radius	SWSCR
topology_related_to_multiply_defined_geometry	TRTMDG
topology_related_to_nearly_degenerate_geometry	TRTNDG
topology_related_to_overlapping_geometry	TRTOG
topology_related_to_self_intersecting_geometry	TRTSIG
unused_patches	UNSPTC
wrongly_oriented_void	WRORVD
wrongly_placed_loop	WRPLLP
wrongly_placed_void	WRPLVD
zero_surface_normal	ZRSRNR

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(59) version(1)}
```

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

B.2.1 Identification of the schema product_data_quality_definition

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

```
{iso standard 10303 part(59) version(1) schema(1)
  product-data-quality-definition-schema(1)}
```

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

B.2.2 Identification of the schema product_data_quality_criteria

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

```
{iso standard 10303 part(59) version(1) schema(1)
product-data-quality-criteria-schema(2)}
```

is assigned to the **product_data_quality_criteria** schema (see clause 5). The meaning of this value is defined in ISO / IEC 8824-1, and is described in ISO 10303-1.

B.2.3 Identification of the schema product_data_quality_inspection_result

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

```
{iso standard 10303 part(59) version(1) schema(1)
product-data-quality-inspection-result-schema(3)}
```

is assigned to the **product_data_quality_inspection_result** schema (see clause 6). The meaning of this value is defined in ISO / IEC 8824-1, and is described in ISO 10303-1.

B.2.4 Identification of the schema shape_data_quality_criteria

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

```
{iso standard 10303 part(59) version(1) schema(1)
  shape-data-quality-criteria-schema(4)}
```

is assigned to the **shape_data_quality_criteria** schema (see clause 7). The meaning of this value is defined in ISO / IEC 8824-1, and is described in ISO 10303-1.

B.2.5 Identification of the schema shape_data_quality_inspection_result

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

```
{iso standard 10303 part(59) version(1) schema(1)
 shape-data-quality-inspection-result-schema(5)}
```

is assigned to the **shape_data_quality_inspection_result** schema (see clause 8). The meaning of this value is defined in ISO / IEC 8824-1, and is described in ISO 10303-1.

Annex C
(informative)

Computer interpretable listings

This annex provides a listing of the EXPRESS entity names and corresponding short names as specified in this Part of ISO 10303 without comments or other explanatory text. This annex is available in computer-interpretable form and can be found at the following URLs:

Short names: <http://www.tc184-sc4.org/Short_Names/>
EXPRESS: <http://www.tc184-sc4.org/EXPRESS/>

If there is difficulty accessing these sites contact ISO Central Secretariat or contact the ISO TC 184/SC4 Secretariat directly at: sc4sec@tc184-sc4.org.

NOTE The information provided in computer-interpretable form at the above URLs is informative. The information that is contained in the body of this part of ISO 10303 is normative.

Annex D
(informative)

EXPRESS-G diagrams

The diagrams in this annex correspond to the EXPRESS schemas specified in this part of ISO 10303. The diagrams use the EXPRESS-G graphical notation for the EXPRESS language. EXPRESS-G is defined in ISO 10303-11:2004,Annex D.

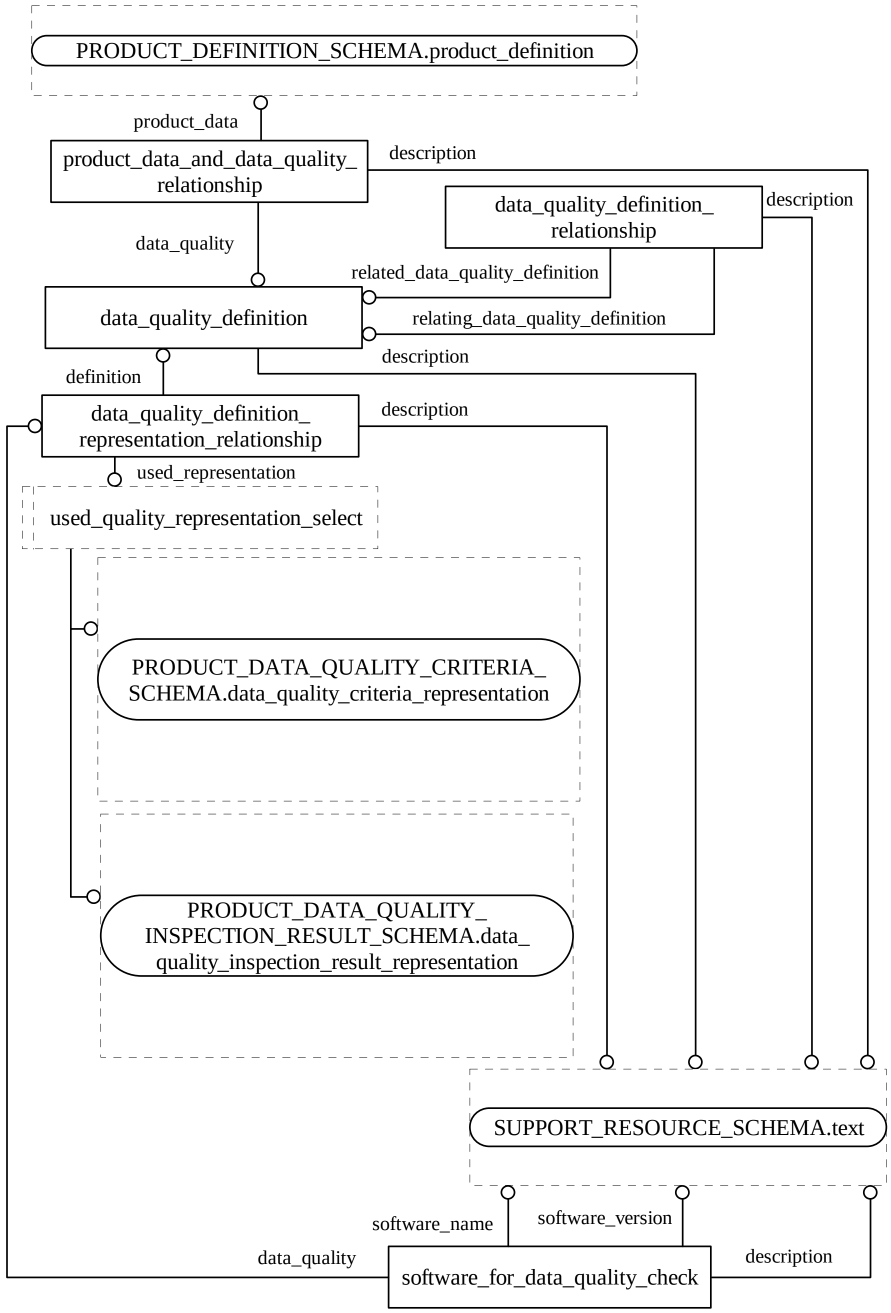

Figure D.1 - EXPRESS diagram of the product_data_quality_definition_schema (1 of 1)

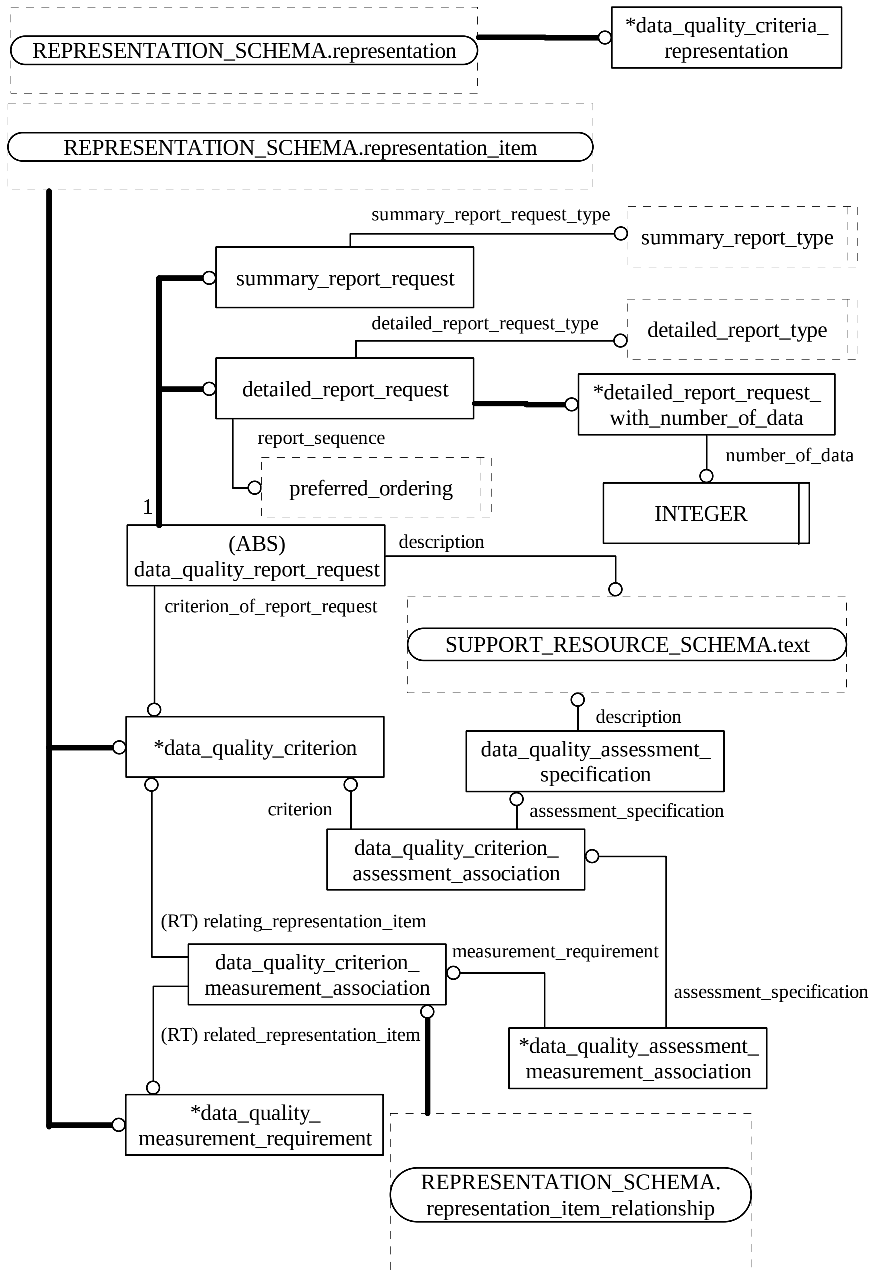

Figure D.2 - EXPRESS diagram of the product_data_quality_criteria_schema (1 of 1)

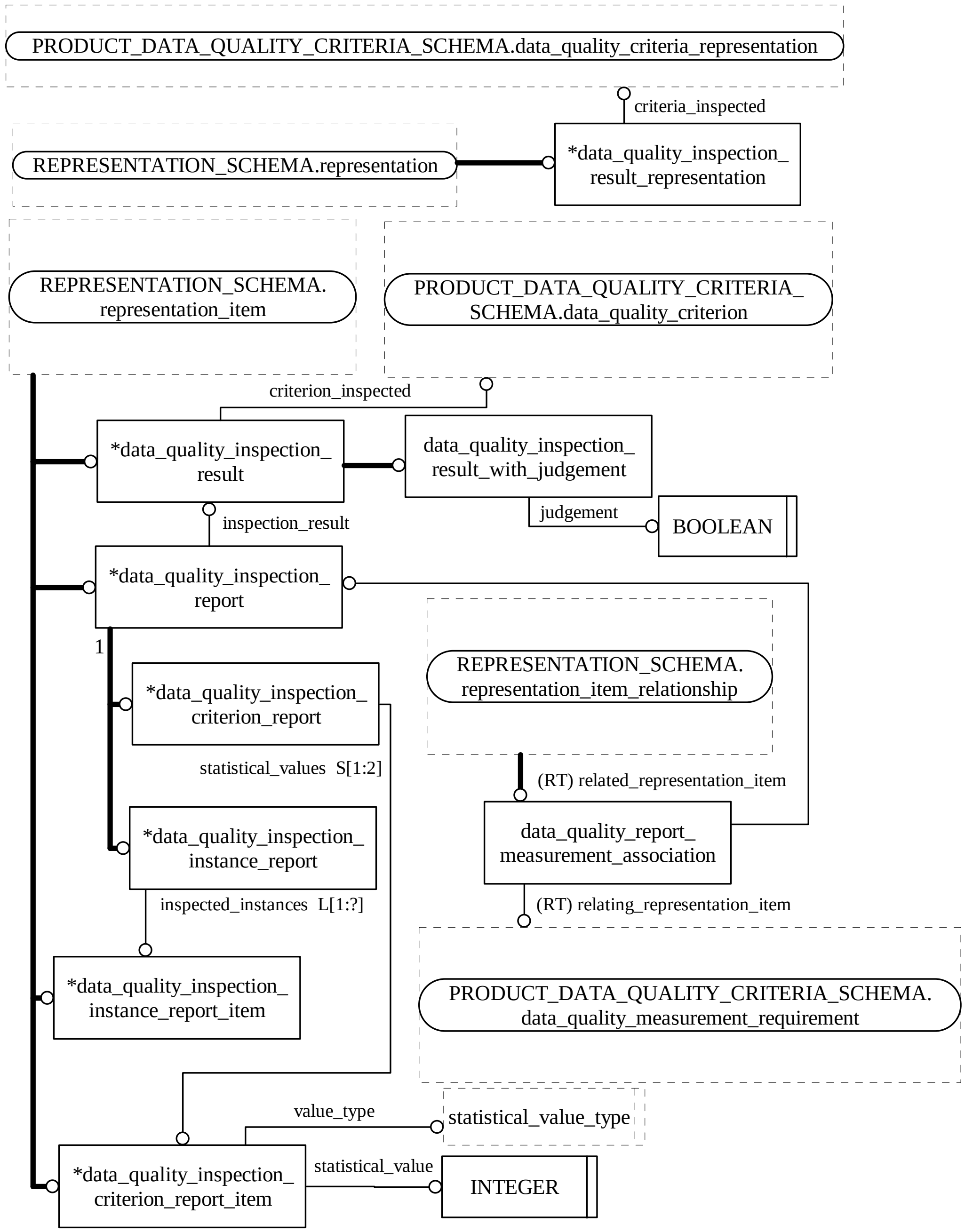

Figure D.3 - EXPRESS diagram of the product_data_quality_inspection_result_schema (1 of 1)

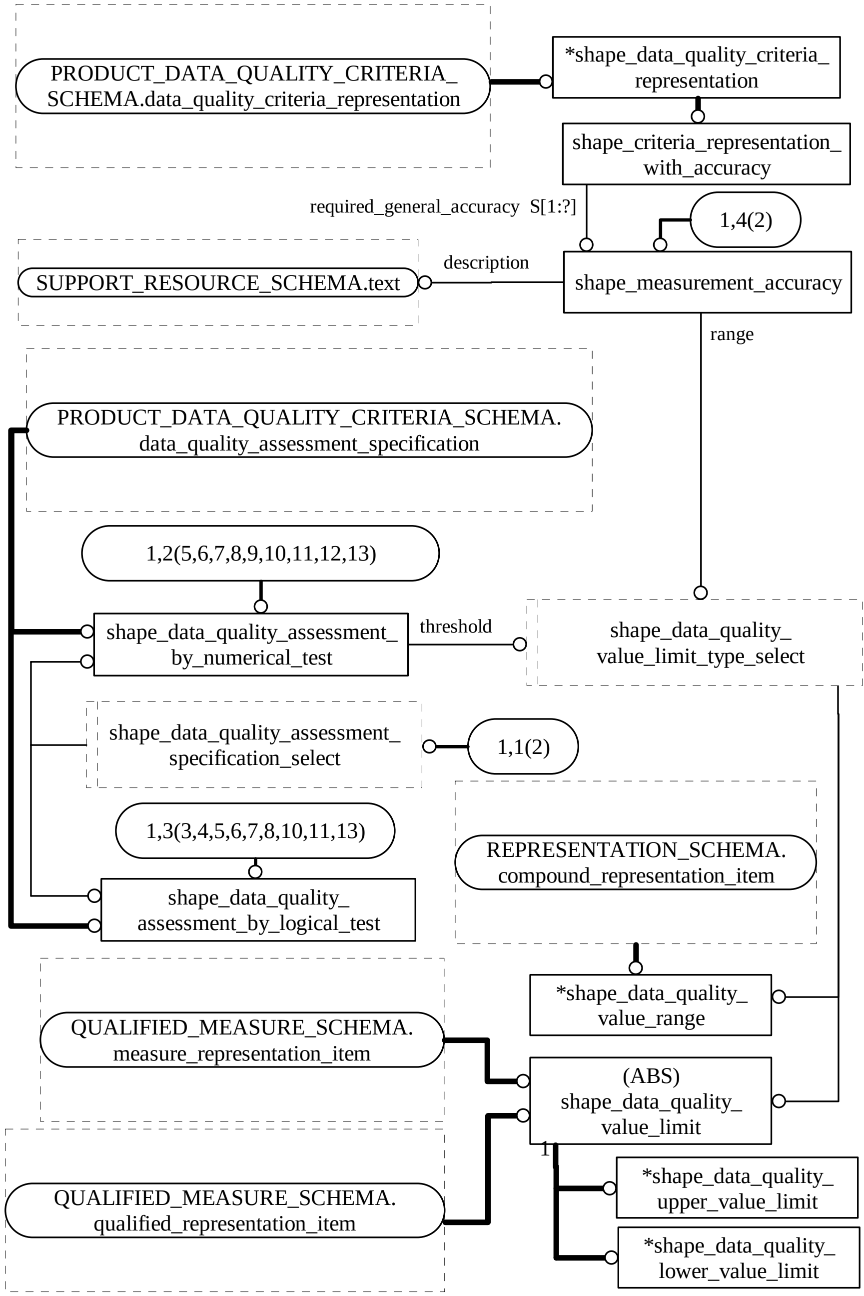

Figure D.4 - EXPRESS diagram of the shape_data_quality_criteria_schema (1 of 13)

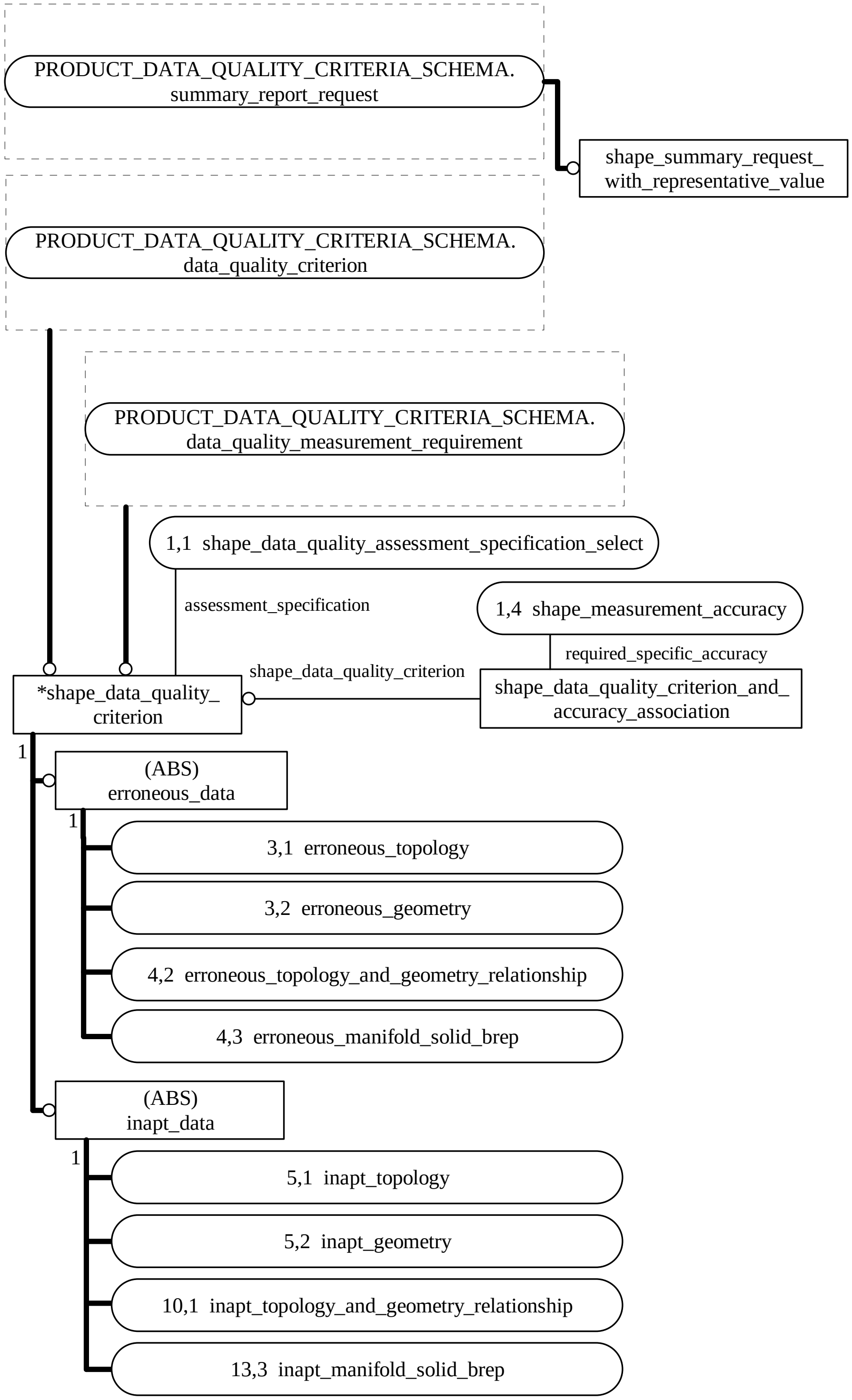

Figure D.5 - EXPRESS diagram of the shape_data_quality_criteria_schema (2 of 13)

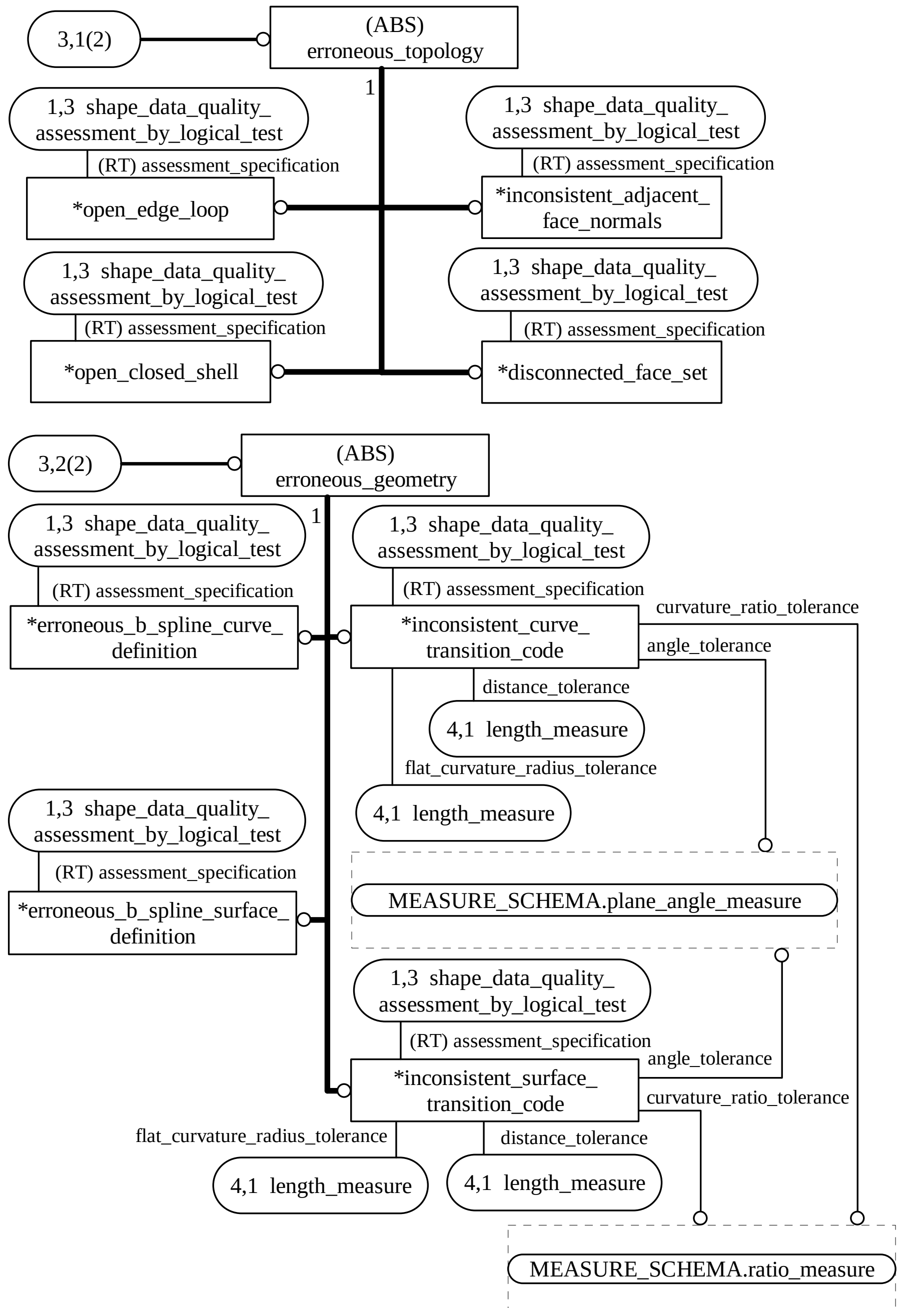

Figure D.6 - EXPRESS diagram of the shape_data_quality_criteria_schema (3 of 13)

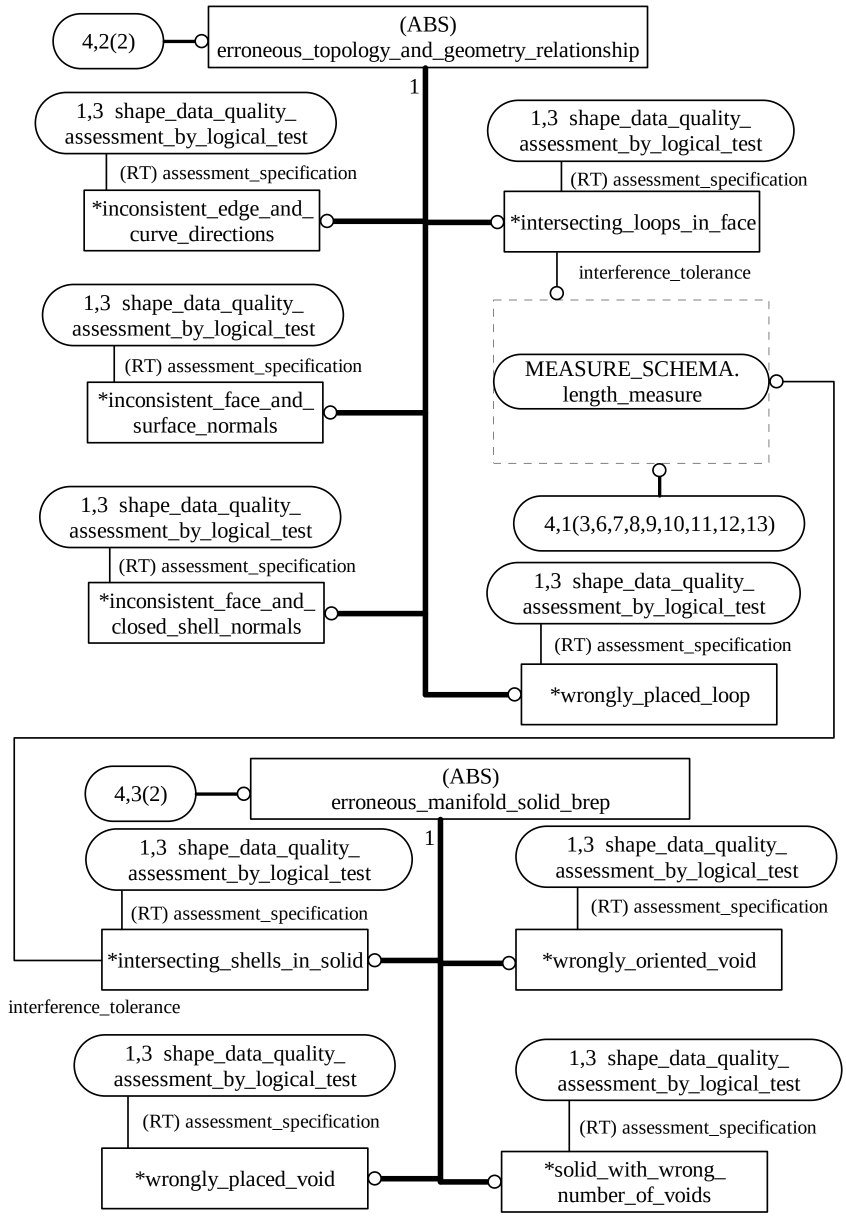

Figure D.7 - EXPRESS diagram of the shape_data_quality_criteria_schema (4 of 13)

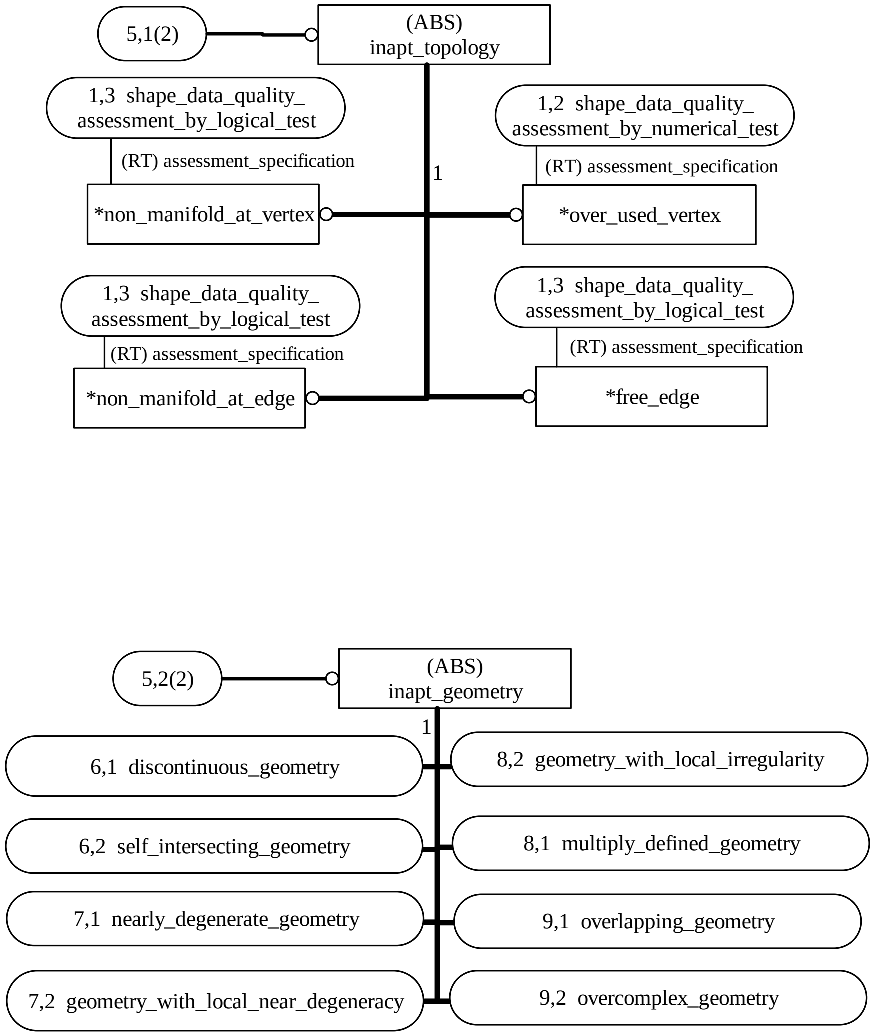

Figure D.8 - EXPRESS diagram of the shape_data_quality_criteria_schema (5 of 13)

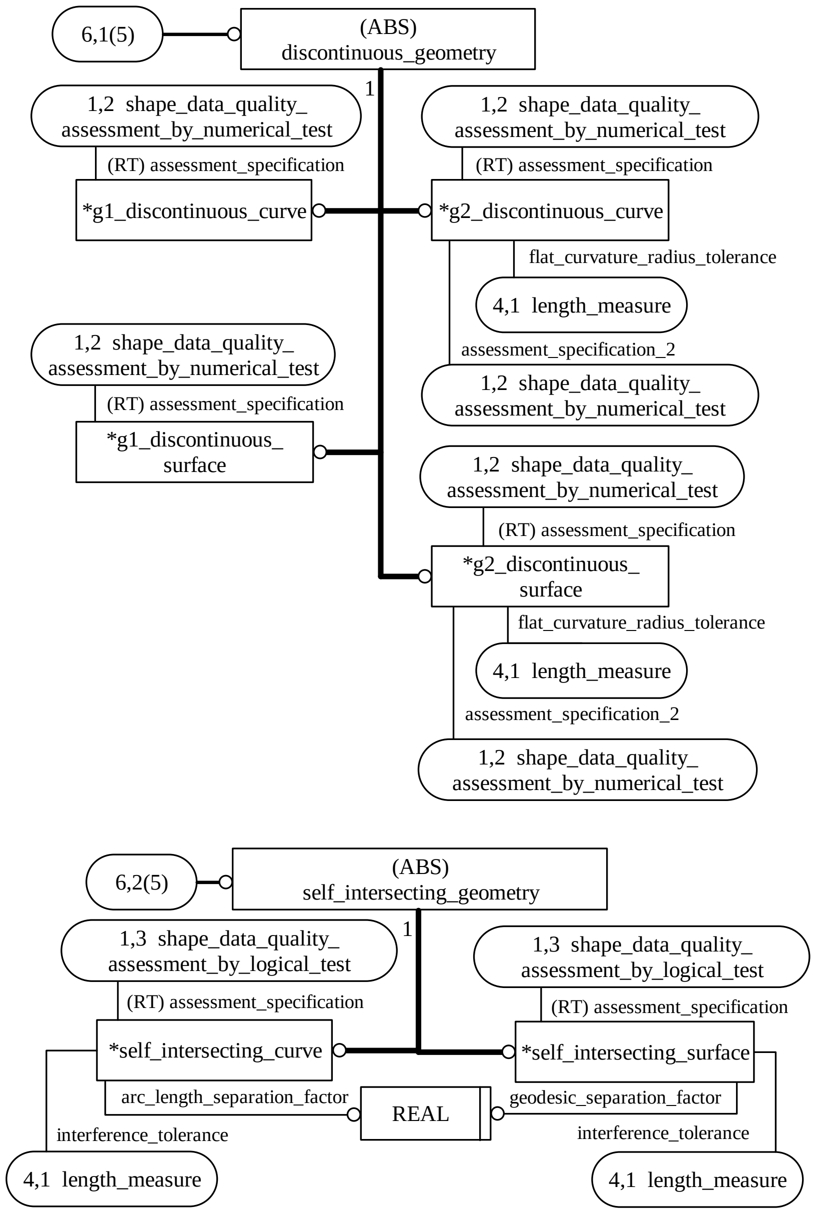

Figure D.9 - EXPRESS diagram of the shape_data_quality_criteria_schema (6 of 13)

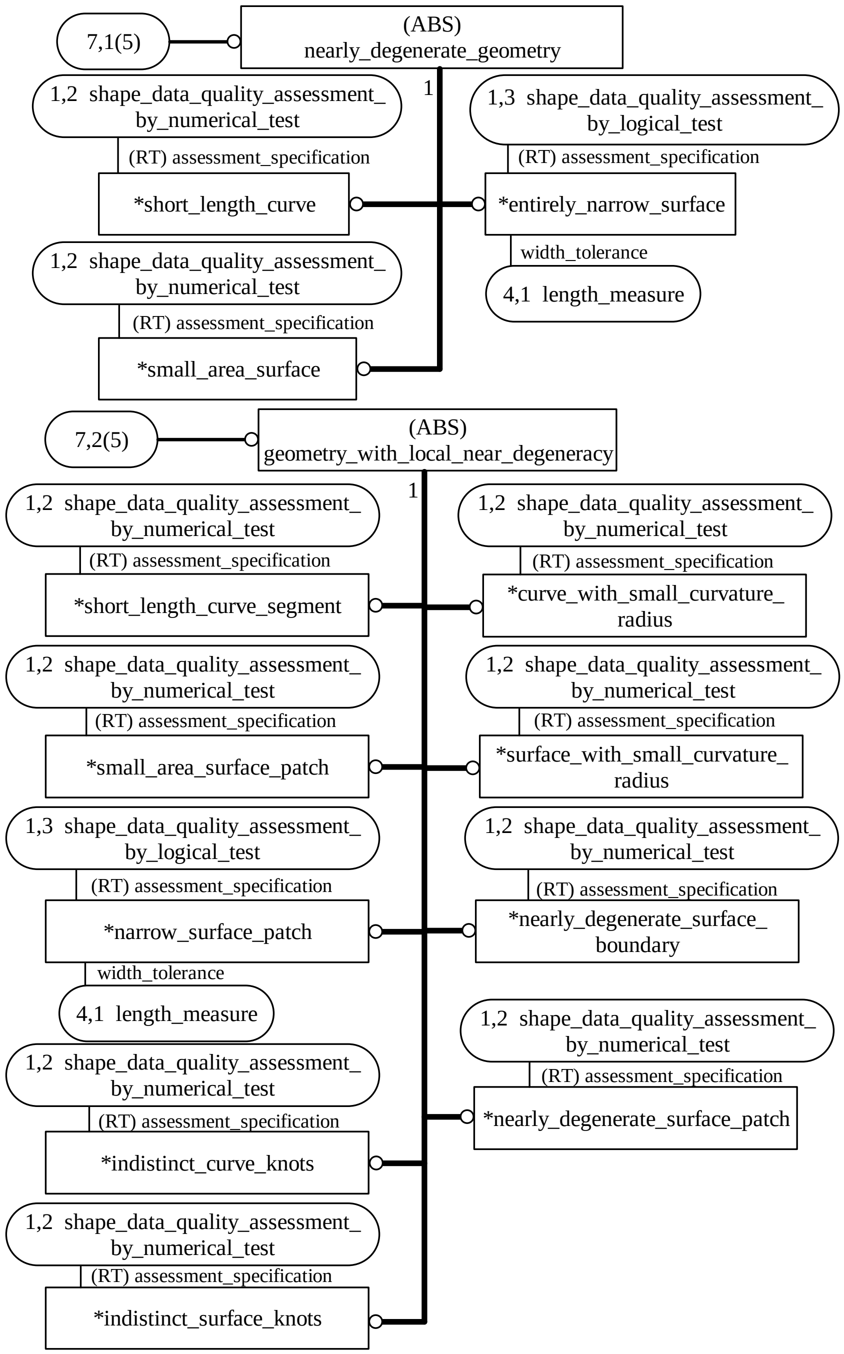

Figure D.10 - EXPRESS diagram of the shape_data_quality_criteria_schema (7 of 13)

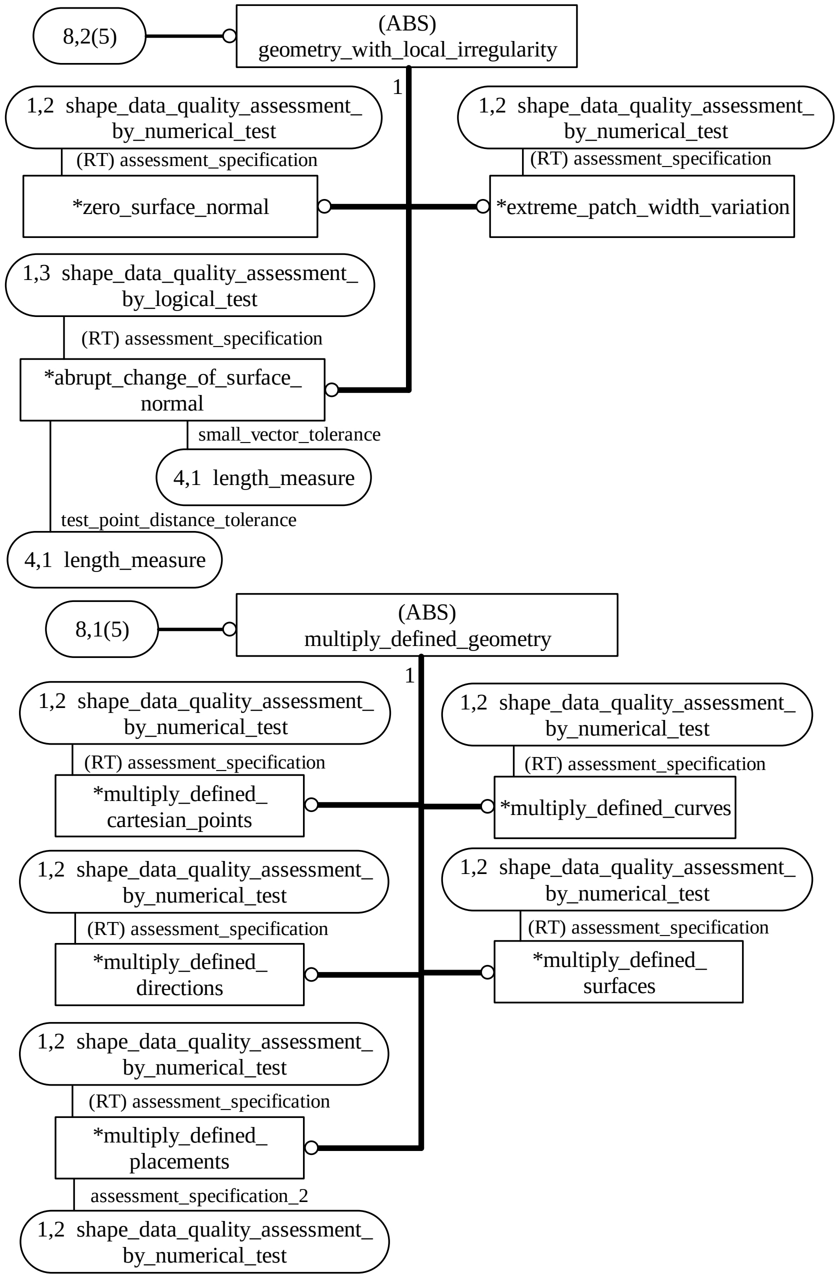

Figure D.11 - EXPRESS diagram of the shape_data_quality_criteria_schema (8 of 13)

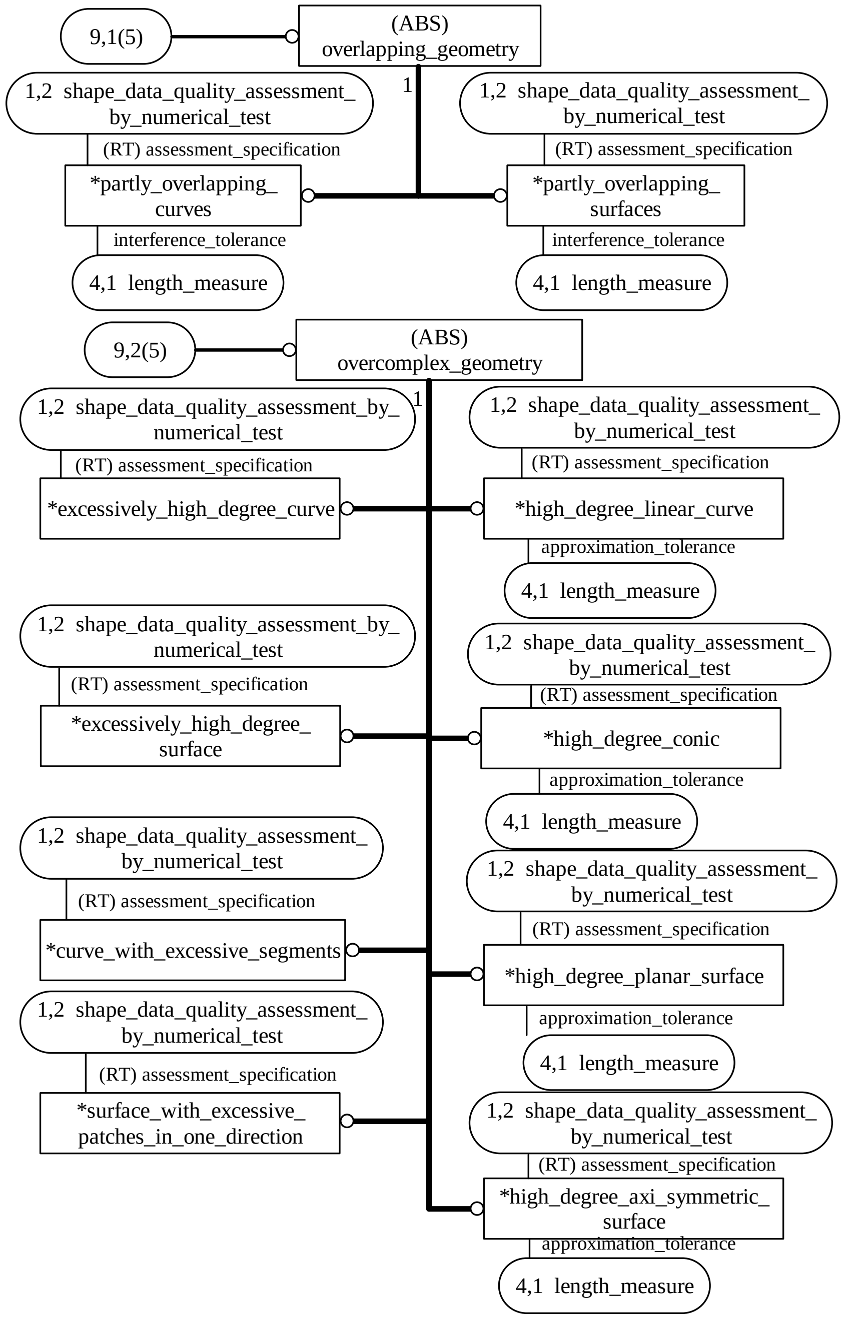

Figure D.12 - EXPRESS diagram of the shape_data_quality_criteria_schema (9 of 13)

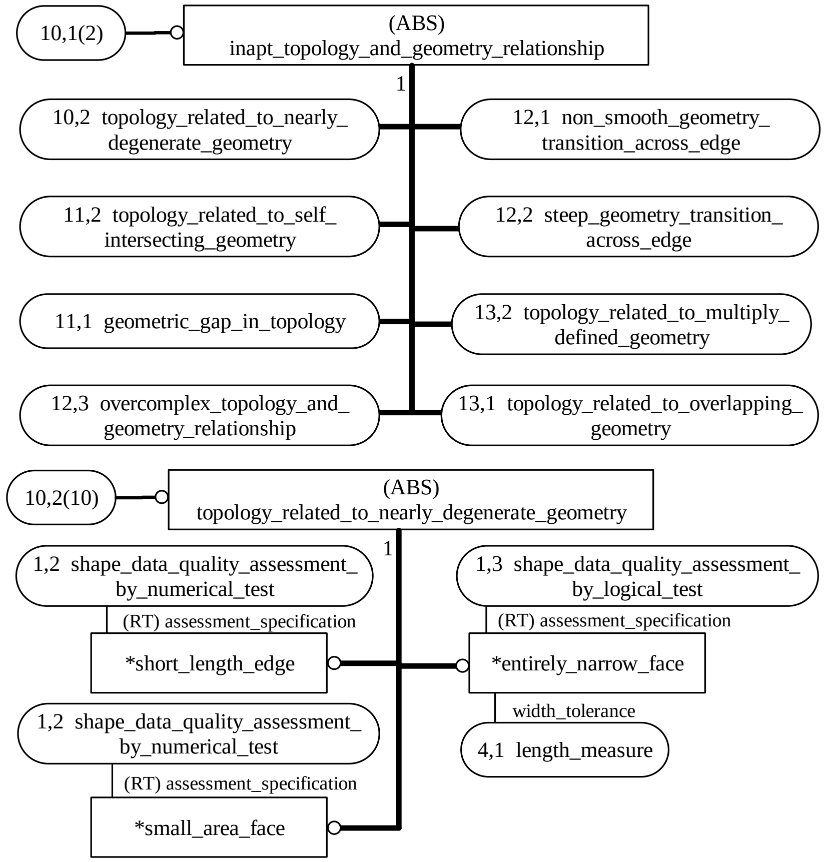

Figure D.13 - EXPRESS diagram of the shape_data_quality_criteria_schema (10 of 13)

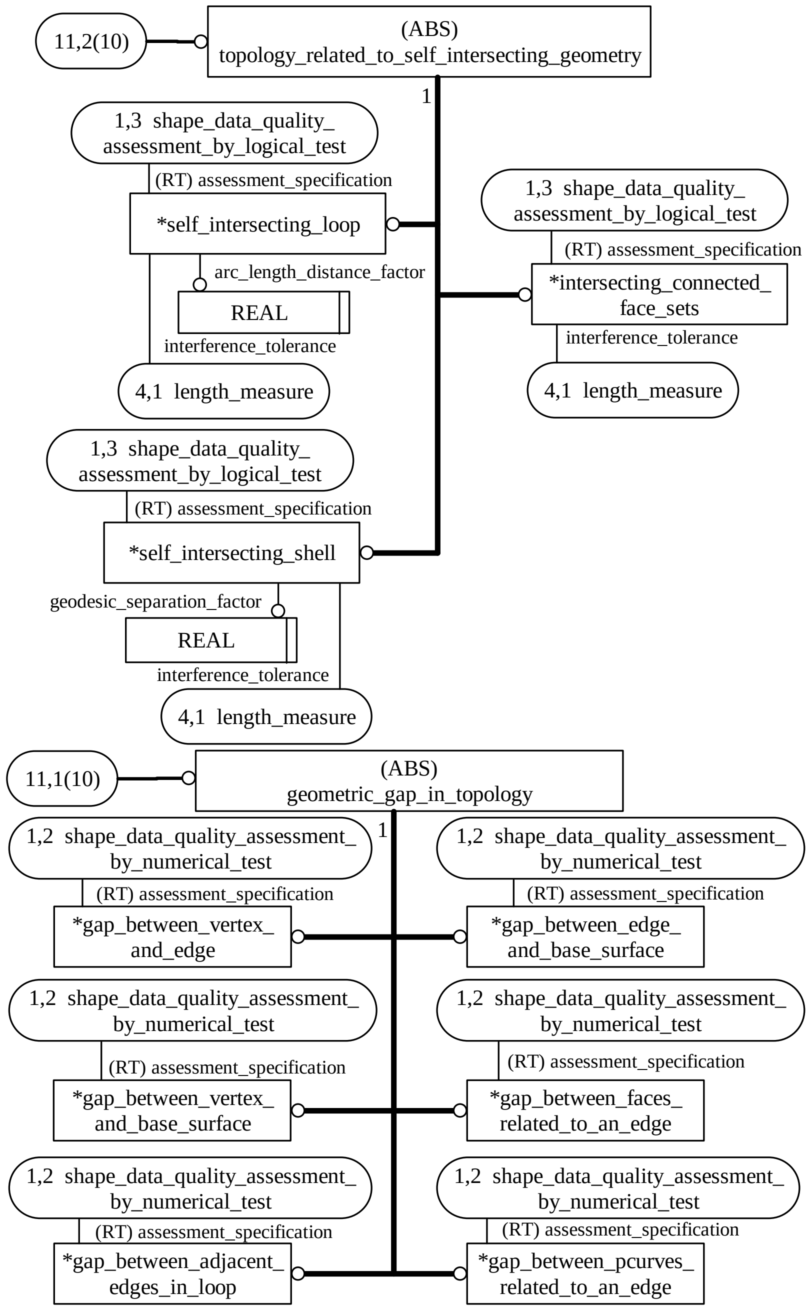

Figure D.14 - EXPRESS diagram of the shape_data_quality_criteria_schema (11 of 13)

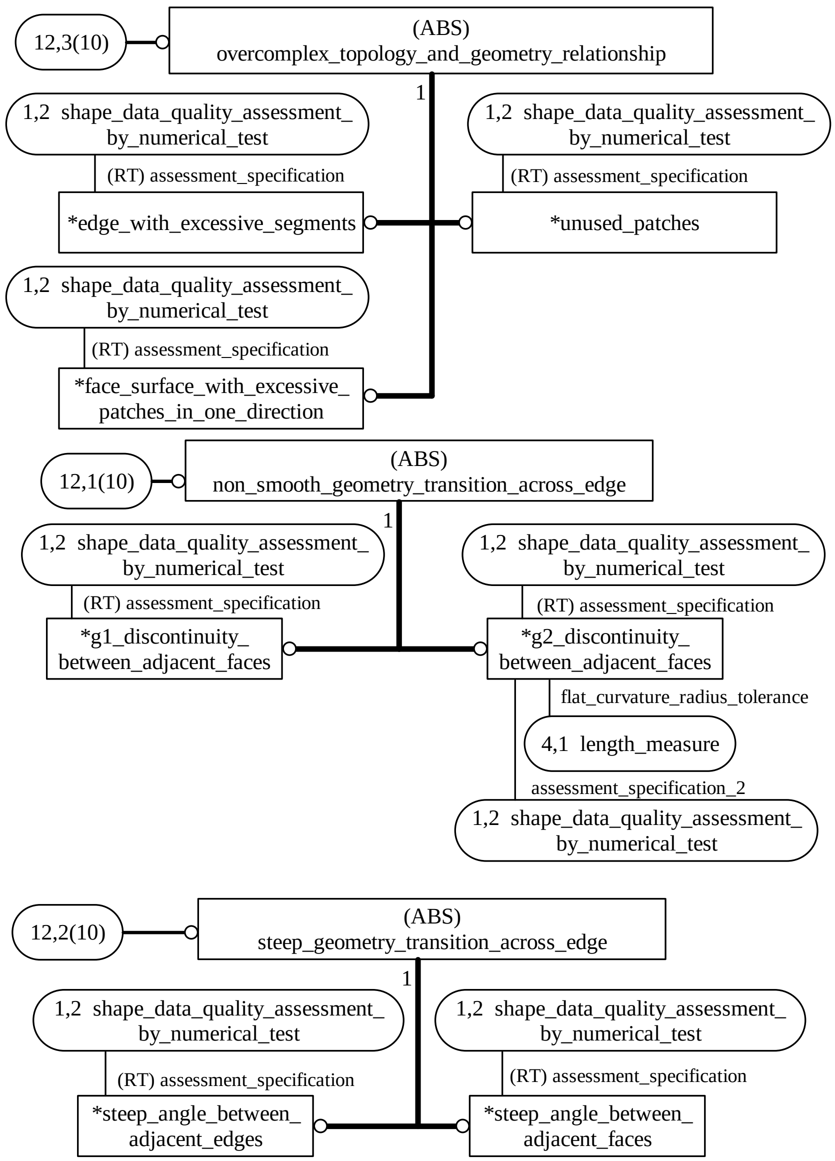

Figure D.15 - EXPRESS diagram of the shape_data_quality_criteria_schema (12 of 13)

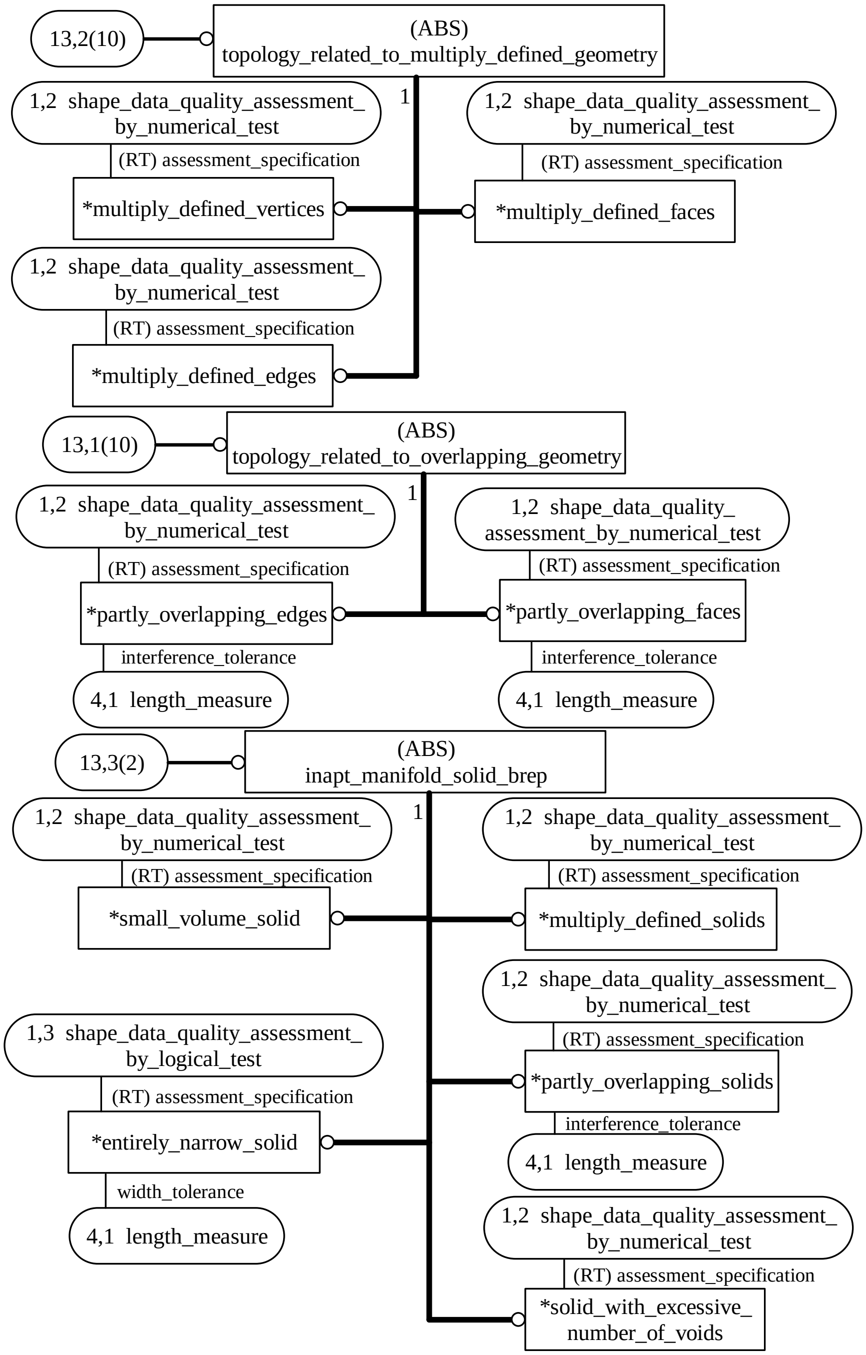

Figure D.16 - EXPRESS diagram of the shape_data_quality_criteria_schema (13 of 13)

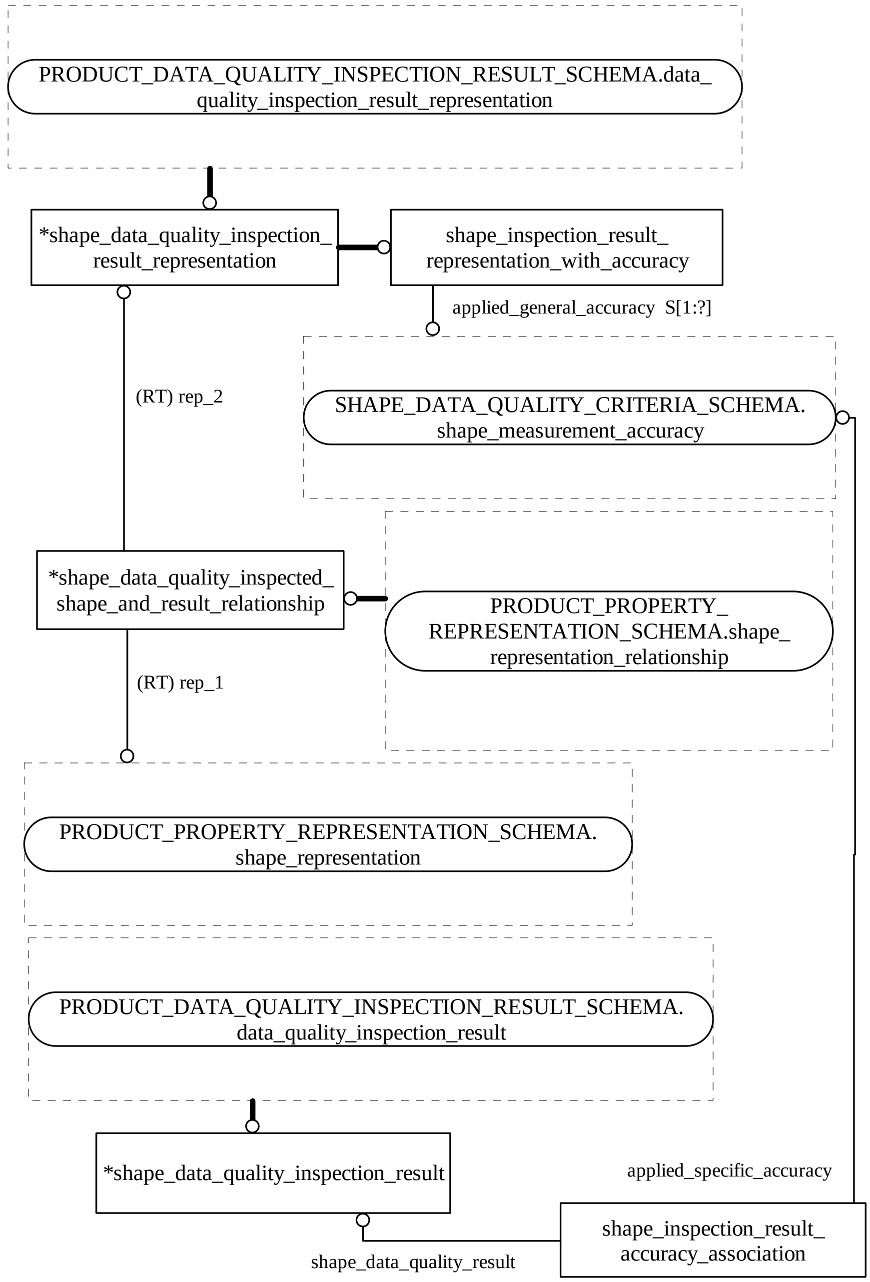

Figure D.17 - EXPRESS diagram of the shape_data_quality_inspection_result_schema (1 of 7)

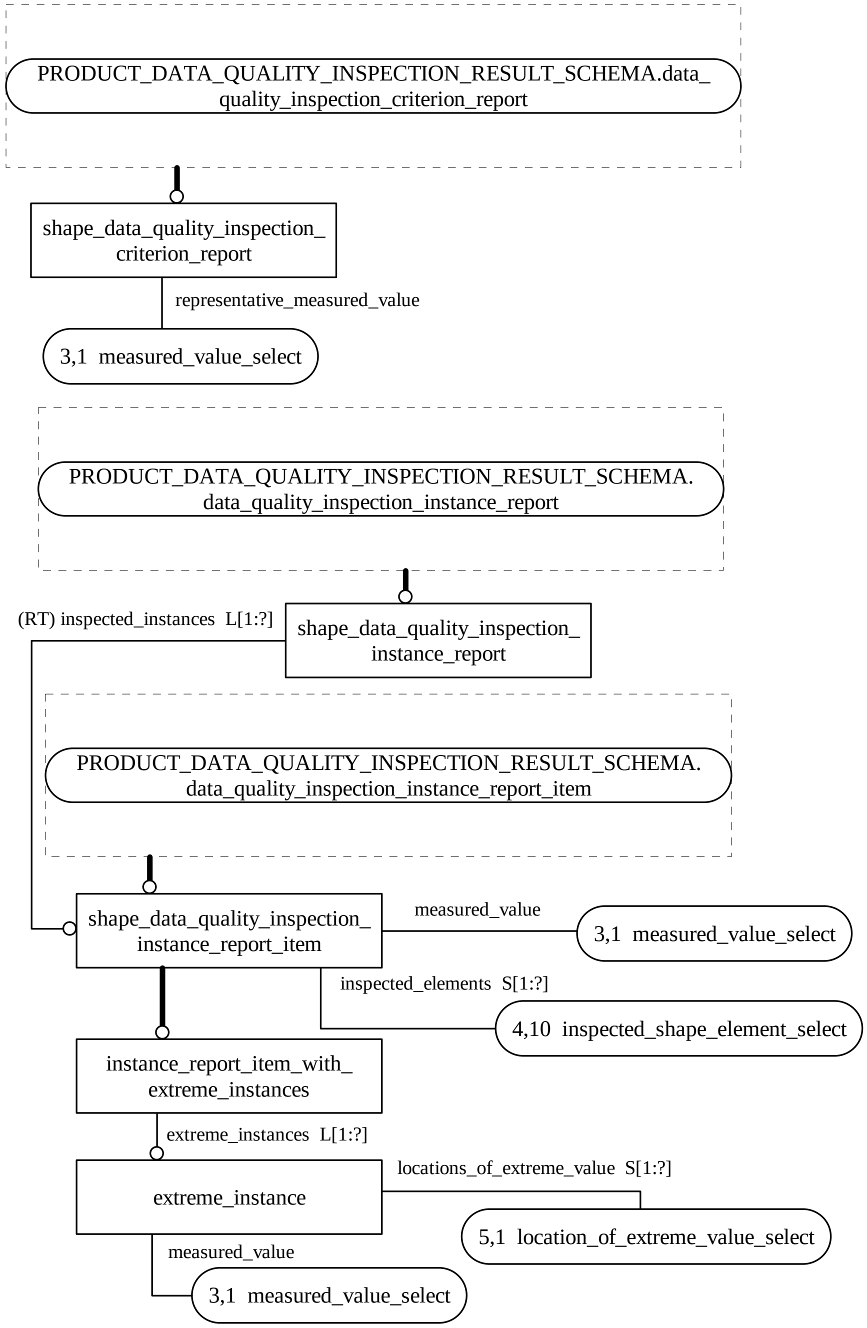

Figure D.18 - EXPRESS diagram of the shape_data_quality_inspection_result_schema (2 of 7)

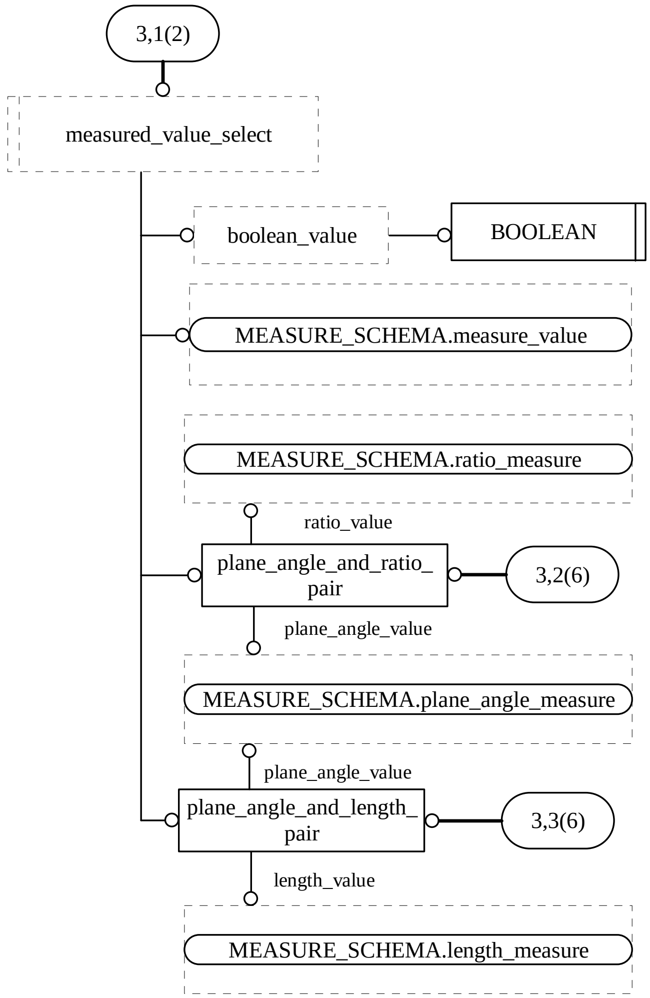

Figure D.19 - EXPRESS diagram of the shape_data_quality_inspection_result_schema (3 of 7)

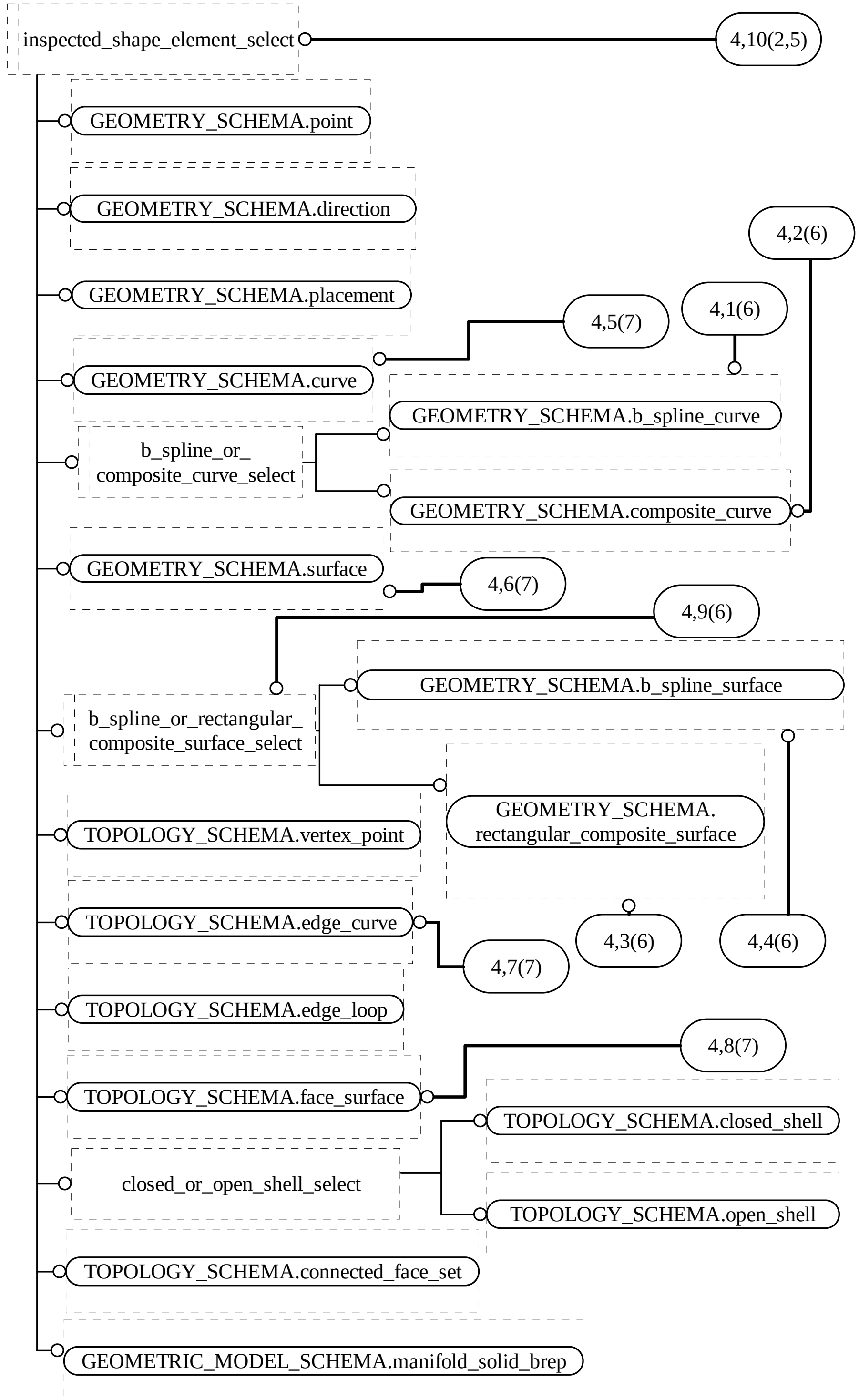

Figure D.20 - EXPRESS diagram of the shape_data_quality_inspection_result_schema (4 of 7)

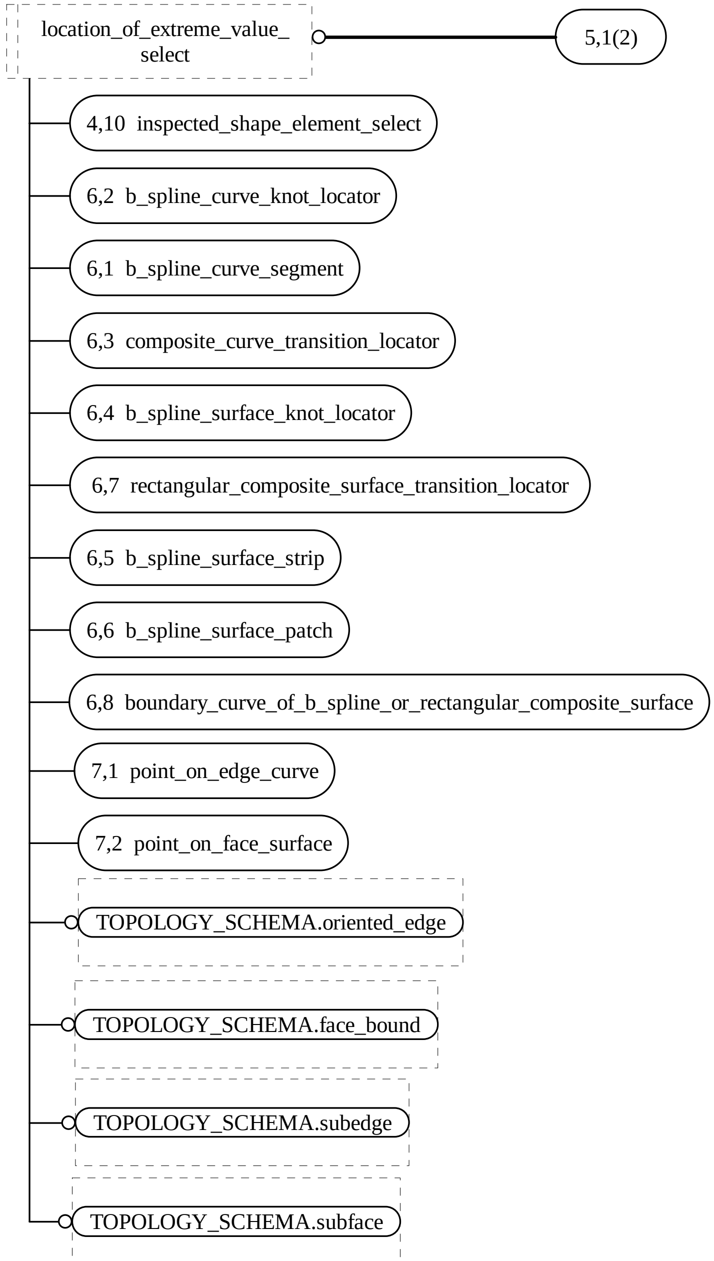

Figure D.21 - EXPRESS diagram of the shape_data_quality_inspection_result_schema (5 of 7)

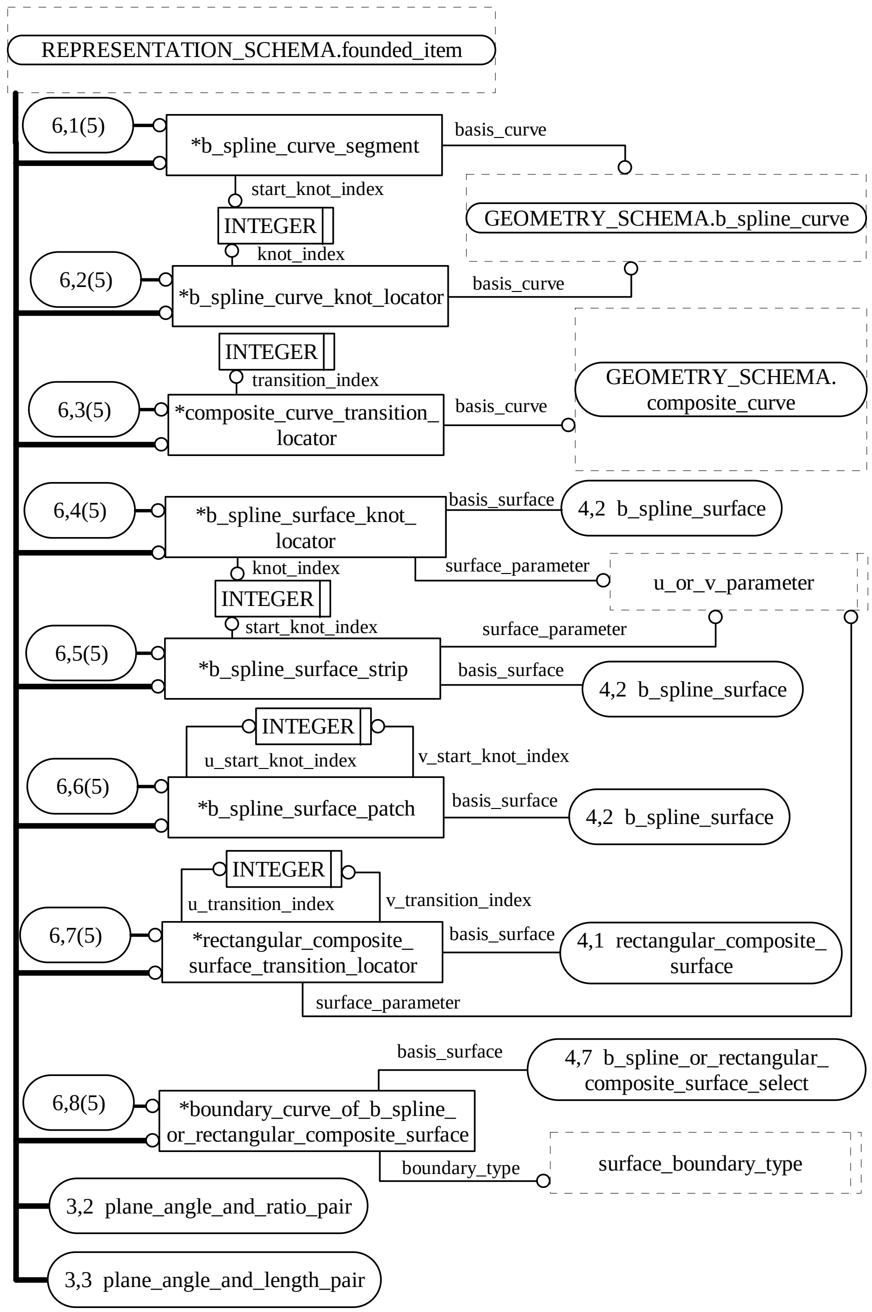

Figure D.22 - EXPRESS diagram of the shape_data_quality_inspection_result_schema (6 of 7)

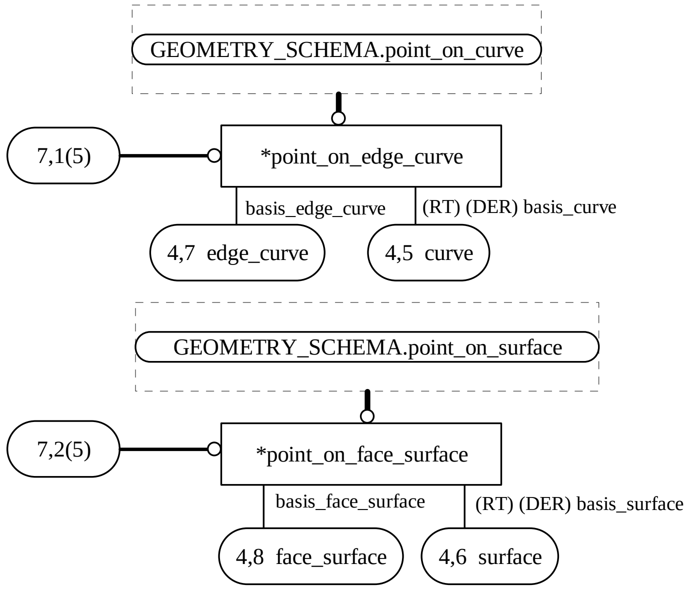

Figure D.23 - EXPRESS diagram of the shape_data_quality_inspection_result_schema (7 of 7)

Annex E
(informative)

Technical discussion

E.1 Background

Increase of the use of engineering systems (CAD/CAM/CAE/CG systems) in product development has revealed the significance of product data quality (PDQ) as a key issue for realizing successful collaborative product development based on product data.

A NIST report titled “Economic Impact Assessment of the International Standard for the Exchange of Product Model Data (STEP) in Transportation Industries”[4] estimates that one billion dollars per year saving will be realized by full use of STEP in US transportation industries (Aerospace, Automotive, Shipbuilding). Improvement of PDQ that eliminates rework could realize further cost savings.

Namely, establishment of widely acceptable methodology for guaranteeing appropriate quality of product model data is critically important for drastically raising effectiveness of the use of 3D engineering systems and for decreasing economic loss.

SASIG’s initiative to develop the automotive PDQ guidelines is a major response to this critical issue. However, PDQ issues are not specific to the automotive industry, but are common to all manufacturing industries. PDQ criteria and acceptable values may differ from one industry to another, and this part of ISO 10303 is intended to allow for such differences.

Additionally, standardization of PDQ methodology is expected to encourage IT vendors to develop better systems in view of product data quality.

E.2 Definition of product data quality

Although importance of product data quality has been widely recognized, it is not yet clearly defined what “product data quality” is. In the discussion of “product data quality”, we have to distinguish quality of “product data” from that of “product” itself and “product model”, because these three types of qualities are often mixed up in conventional context.

“Product” is a manufactured physical body, and “product model” is a mathematical model to represent “product” on computers. “Product data” is numerical data that represent product information in accordance with “product model.”

“Product quality” is defined as the degree of satisfaction of users’ requirements on its function, performance, appearance and so on. There are a number of existing researches and achievements on “product quality.” We also have the standard “ISO 9000” for the management of “product quality.”

On the other hand, we don’t even have a basic definition of “product model quality.”

Since “product data” is based on “product model”, it is difficult to give rigorous or theoretical definition of “product data quality” today.

This is the reason why we adopt in this standard a practical approach described below for modelling "product data quality."

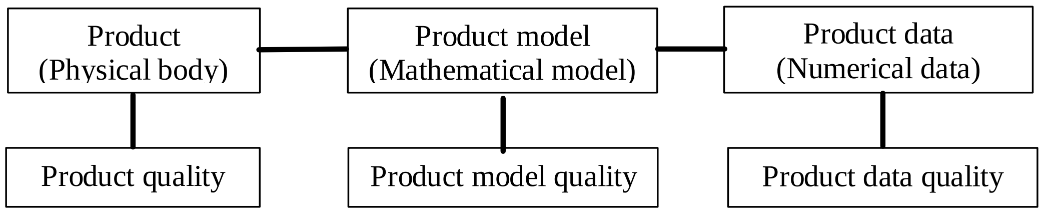

Figure E.1 - Quality of product, product model and product data

Issues that hinder successful exchange or sharing of product data have been categorized as either erroneous data or inapt data. Each of these classes has been further categorized into the subclasses of geometry-specific issues, topology-specific issues, combined geometry and topology issues and geometric modelling issues. These are further discussed in clause E.4.

E.3 Target data type of PDQ-S

Among variety of information consisting of product data, we concentrate on three dimensional shape data in this standard though schemas for allowing future extension to cope with non-shape data quality are also included. The reason why we concentrate on three dimensional shape data is that it is the core information and most repair work is done to improve shape data today. Reflecting our focus, this project is called "PDQ-S." (Product Data Quality – Shape)

There are various ways to represent three dimensional product shape, B-rep, CSG and parametric feature based representation. Among them, we concentrate on parametric curves and surfaces, and B-rep data, because these data are majority of 3D shape data transferred and shared today.

Although use of parametrically represented 3D shape data in explicit form with geometric constraints, implicit form with construction history, or with their mixture are increasing especially in solid modelling environment, it will take some time to enable parametric data exchange since some of the related standards in ISO 10303 have reached to IS level very recently. Therefore, we will leave quality of shape data with parametric information as a future issue.

E.4 Quality criteria of three-dimensional shape data

Quality criteria of three-dimensional shape data are classified into the following classes:

Erroneous data;

Inapt data;

Each of these classes is further categorized into the following sub-classes:

Geometry specific issue;

Topology specific issue;

Combined geometry and topology issue;

Geometric model issues.

'Erroneous data' implies mathematically invalid product shape data. Amongst examples are erroneous b_spline surface definition, open edge loop and inconsistent face and underlying surface normals.

'Inapt data' implies data inappropriate for applications though it may not be mathematically incorrect. Amongst examples are self-intersecting curve, non-manifold at edge, narrow width surface patch and gap between bounding edge and underlying surface.

Even when shape data have acceptable quality in the senses described above, there exists a number of cases where engineers in downstream processes have to rework on the data.

For example, mould design engineers have to modify a product shape data if appropriate draft angle is not taken into account in the data. Just knowing the fact whether various manufacturing requirements are incorporated into the data or not may save the cost of rework. Therefore, these issues are within the scope of this project because they hinder circulation of product data from up stream to down stream.

Unfortunately we cannot focus on them in this initial version because the information to be incorporated is not yet well formalized. But it's among one of the most important open issues for this project

E.5 Thresholds and Accuracies

For the assessment of shape data quality by numerical test, thresholds which are user definable from AP (Application Protocol) play a key role. An example of a typical threshold is a distance threshold for evaluating a gap between a base surface and bounding curves for trimming the effective portion of the surface. That distance threshold implies that maximum distance between the surface and the curves shall be smaller than the threshold. Appropriate thresholds depend on many factors such as size of a product, design requirements, robustness of engineering systems on numerical precision, etc. Therefore, what threshold value to use shall be carefully determined in each business situation based on agreement among business partners.

In most cases, especially when free form geometry is involved, the measurement algorithm calculates an approximate solution not an exact solution. In the above example, surface and curves consist of an infinite number of points. Since calculation on an infinite number of points is impossible, any algorithm tries to calculate the solution using a sufficient number of finite points. In order to require the difference of the approximate solution and the exact solution, even when it is unknown, smaller than the expected value, accuracy specification is provided. The accuracy shall be understood that an approximate solution is acceptable if the difference of the approximate solution and other approximate solution obtained by calculation with any other finer sampling points is smaller than the given accuracy. There are two types of accuracies, one is general accuracy applied to all the measurements and the other is specific accuracy applied only to specified measurement.

E.6 Assumptions on representation of product data

The following two assumptions are rather reluctantly adopted in this standard.
(1) Inspection target product data is limited to those conformant with ISO 10303.
(2) Quality data is assumed to be stored in the same STEP file as the inspection target product data

It is clearly understood that product data quality is a common issue for all the product data representations. The reason why assumption-(1) is adopted is that consistency of PDQ data and inspection target product data , such as consistency of representation context, is only guaranteed if the inspection target product data is conformant with ISO 10303. If this consistency issue is resolved, then the usability of this standard will be easily extended to other product data representations.

As already described, it is understood that PDQ data is not a constituent of product data. It implies that PDQ data may or may not be stored together with the inspection target product data. The reason why assumption-(2) is adopted is that no methodology currently exists within SC4 to refer to entity instance in externally defined product data file. If entity instance reference mechanism to external file is provided then this limitation will be immediately removed and PDQ data and the inspection target product data could become represented in separate files with necessary relationship on condition that the entity instance reference mechanism to external file does not require the change of quality model information structure.

Annex F
(informative)

Similarity/Difference of this standard with the SASIG PDQ Guidelines

F.1 Similarity

F.1.1 Enumeration of concrete measures to eliminate low quality product data

Based on the understanding that low quality numerical data representing three dimensional shape, which necessitates considerable time and cost in receiver side for data repair, is the major reason of enormous economic loss and considerable delay of product development, both the SASIG PDQ Guidelines[5] and the PDQ-S enumerate concrete measures to eliminate low quality product shape data.

F.1.2 Product data quality inspection report

Both the SASIG PDQ Guidelines and the PDQ-S provide product data quality inspection report though the latter provides more detailed information in entity instance level. The concept of Quality Stamp of the former corresponds to **shape_data_quality_inspection_criterion_report** of the latter.

F.2 Difference

F.2.1 Representation method

The SASIG PDQ Guidelines is written by a natural language (UK English) but core entities and functions of this standard are written by formal specification description language: EXPRESS similar to other SC4 standards. Formal description has advantages such as unambiguous understanding of the specification and ease of implementation by computer programs.

F.2.2 Target industry

The SASIG PDQ Guidelines has been developed for resolving today's critical quality related problems in automotive industry. It naturally targets automotive industry but this standard targets all manufacturing industries including automotive, aerospace, commercial electric/electronic, precision mechanical industry, etc.

But, amount of industry type specific quality problems is far small than expected. Therefore, difference of contents of this standard and those of the SASIG PDQ Guidelines is not significant.

F.2.3 Target data type

The SASIG PDQ Guidelines mainly targets three dimensional shape data. But, it additionally deals with some types of non-geometry attributes, CAE data and management data. This standard deals with three dimensional shape data in detail and provides schemas to extend to cope with non-shape data in the future.

Annex G
(informative)

Scenarios for the use of this standard

The following subsections define data elements from Figure 1 (see Introduction) to be instantiated for different usage scenarios involving quality information concerning product data. For each case, the associated figure in this annex is a modified version of Figure 1 retaining only required data elements.

G.1 Requirement and Declaration of data quality

Ordering company may require ordered company to create a product data so that it satisfies prescribed quality requirements. Examples are not to include infinitesimal geometry smaller than the given tolerance, exclusion of redundant geometry not contributing to the representation of product shape, etc. Very limited information, namely relevant criteria together with required thresholds from those in Figure 1, is necessary in this scenario. The information may be transferred together with the ordering sheet.

The creator of a product data may use quality information for explicitly declaring the quality level satisfied by his/her model. Depending on the design method and the CAD system used, the quality of the product data may be unambiguously declared without any inspection. Selective criteria and thresholds for which the model is judged to be quality defect free are required information in this scenario. The quality information may be transferred together with the corresponding product model data.

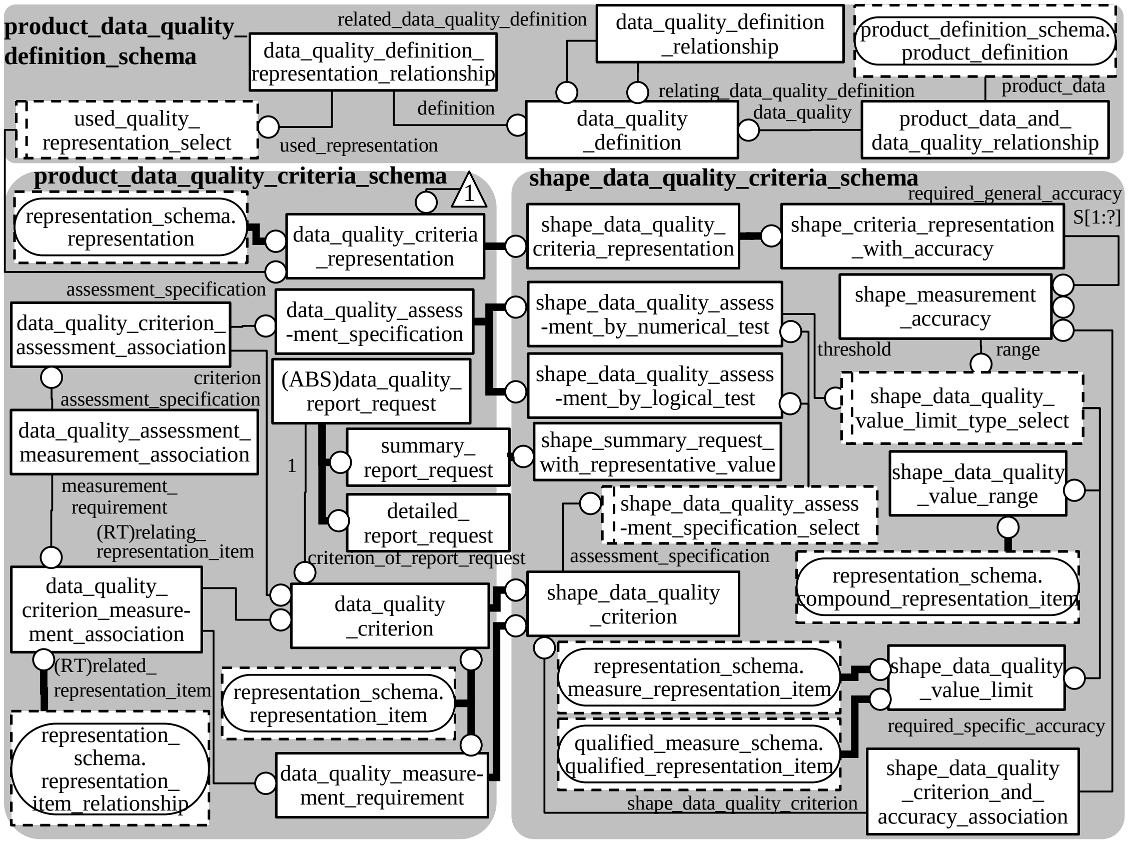

Figure G.1 - Quality information for use in requirement/declaration of data quality

G.2 Assurance of data quality, and long term data archiving

A quality assurance organization may use quality information for representing the results of quality inspection for a particular product model data. This scenario will require inspected quality criteria together with thresholds used, measurement requirements deployed and inspection results obtained. The accuracies used may also be included. The information is transferred together with the corresponding product model data.

It is desirable that a detailed record of product model data quality is archived with product data. The data requirement for this purpose is similar to that needed for assurance of quality.

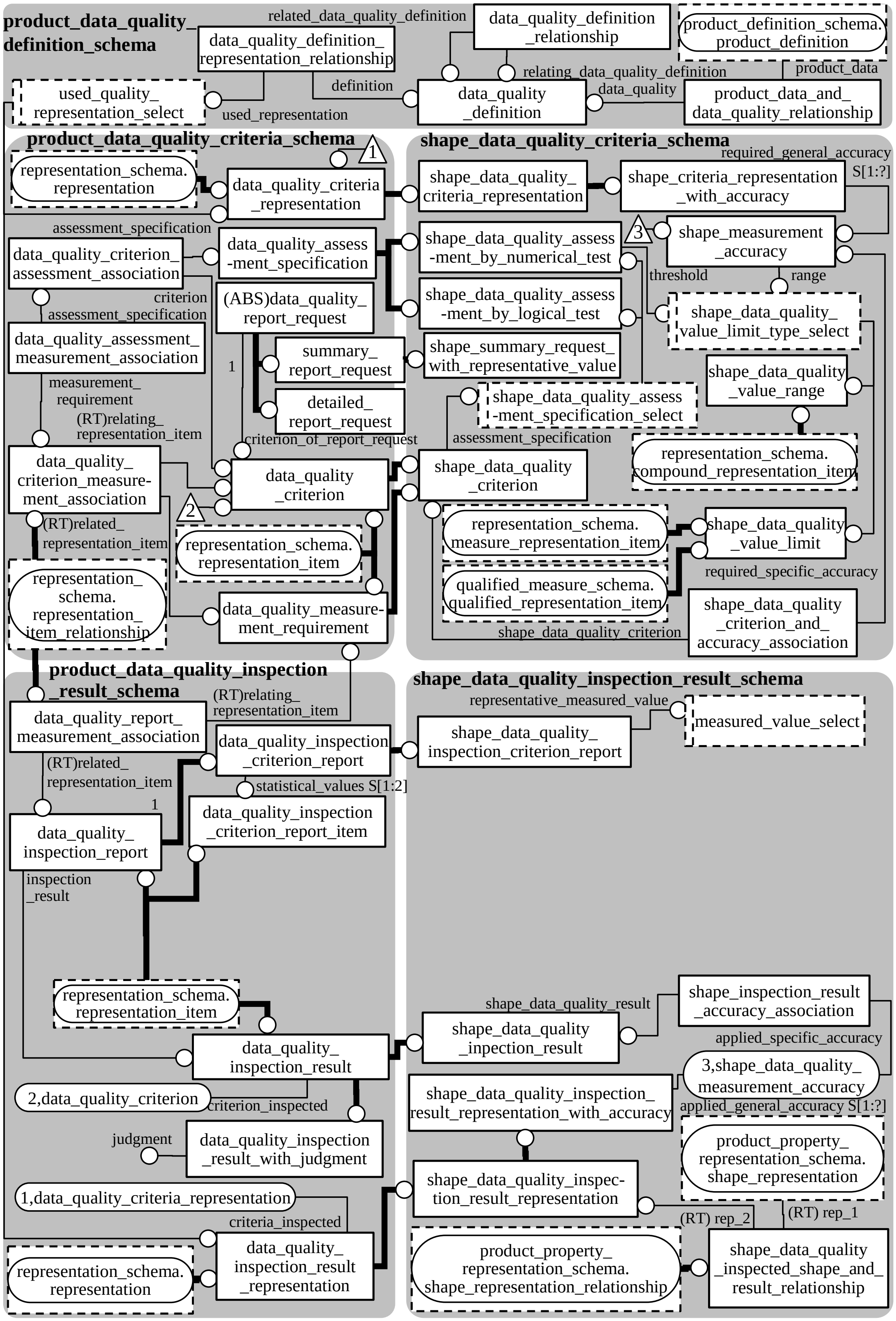

Figure G.2 - Quality information for use in assurance of data quality

G.3 Data quality information for use in quality improvement

If a quality defect is detected by quality inspection, necessary actions for improving critical data will be required. For that purpose, information on the nature and severity of any quality defects should be provided. Therefore, this scenario will require a detailed inspection result report at the geometric entity instance level. This information is transferred together with the corresponding product model data.

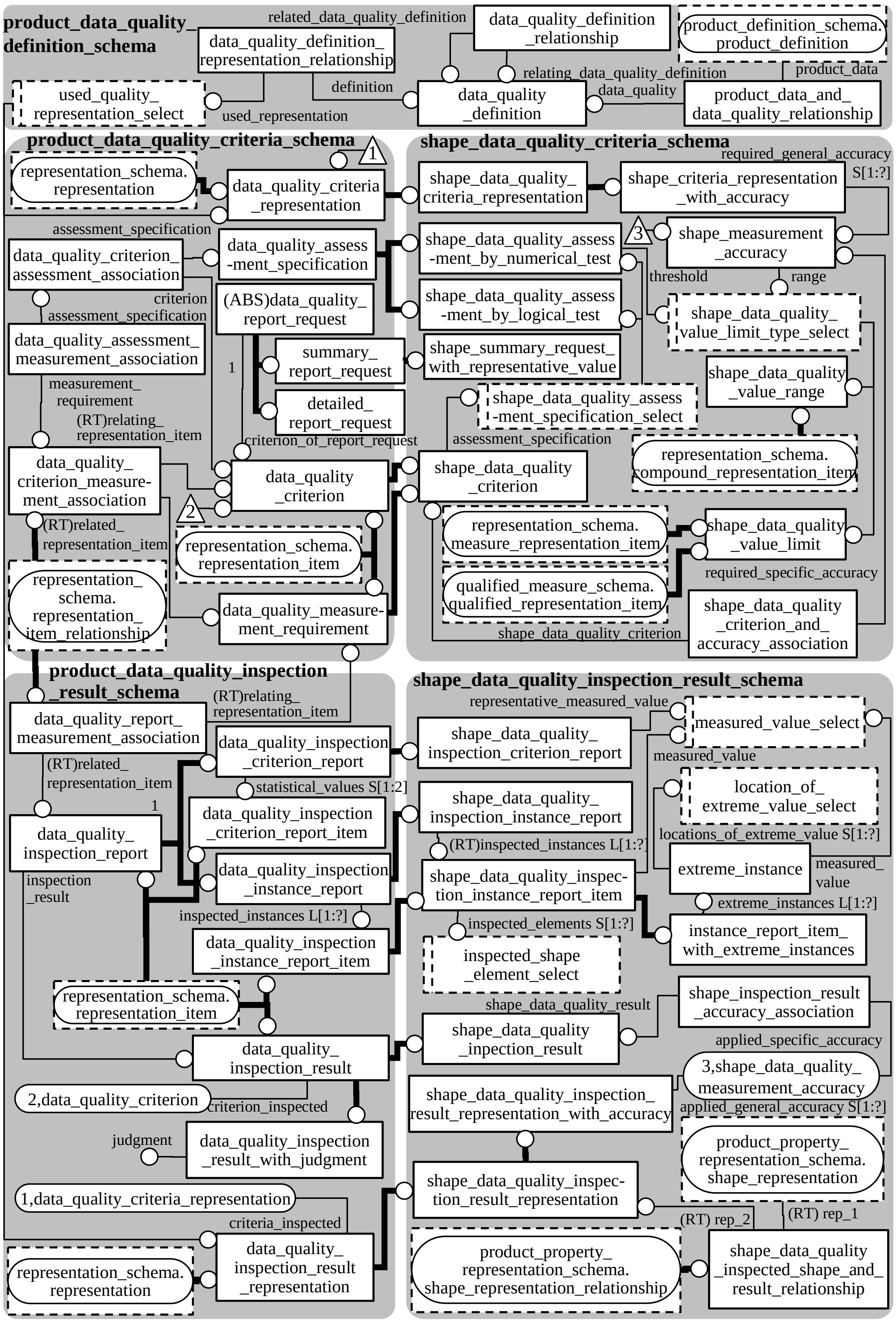

Figure G.3 - Quality information for use in quality improvement

Annex H
(informative)

Instantiation Examples

This annex provides some example instantiations of ISO 10303-59. First, a graphical notation for the presentation of instances is explained that is a customization of EXPRESS-G for ease of understanding. Two practically important criteria, **short_length_edge** and **gap_between_- edge_and_base_surface**, are selected as examples. Example instances are generated for the three scenarios described in Annex G, i.e., requirements/declaration of data quality, assurance of data quality and data quality information for use in quality improvement.

H.1 Graphical notation of instances

The graphical notation for the presentation of instances used in this clause includes some extensions to conventional EXPRESS-G:
- Multiple instances can be presented for a model entity.
- Actual values can be presented using call-outs.
- An instance for a group of entities with inheritance relationships is shown by circumscribed broken lines.

Figure H.1 illustrates an example usage of the notation. Three instances created for the EXPRESS-G model on the left, are shown on the right. The diagram shows three instances of **person**, namely John, Mary and their son Mike.

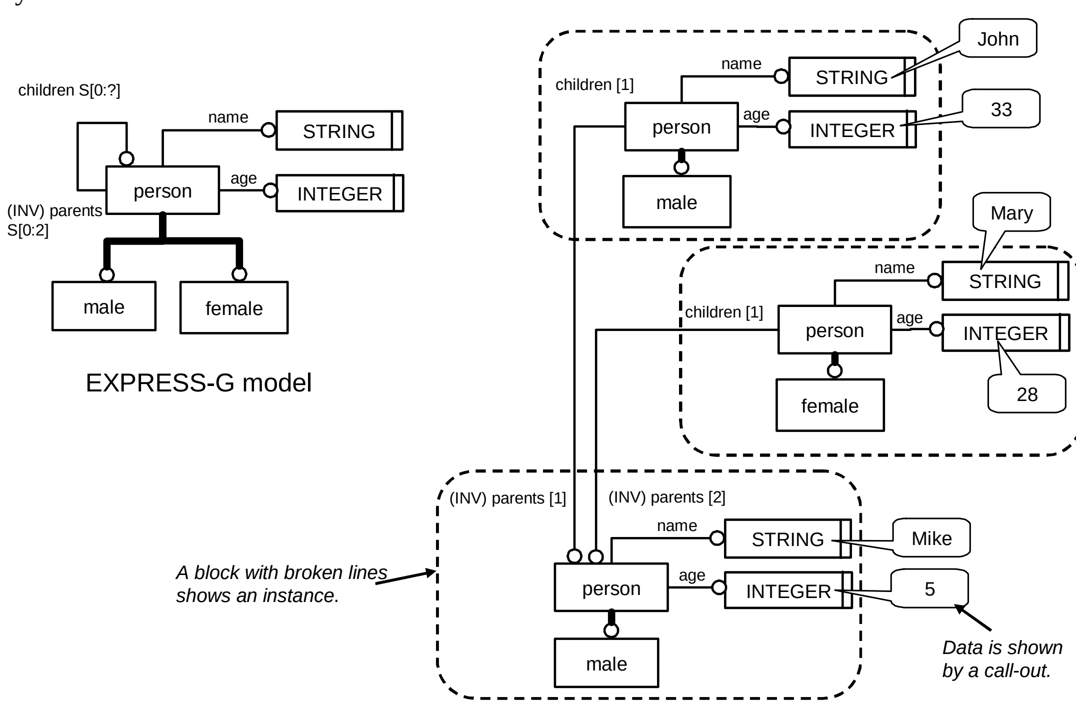

Figure H.1 - Example usage of the notation

H.2 Instances for short_length_edge

As described in Annex G, ISO 10303-59 assumes the following scenarios. Examples are expanded to each of these scenarios.

1) Representation of requirements of data quality
2) Representation of declaration of data quality
3) Representation of assurance of data quality
4) Representation of data quality information for use in quality improvement

Since the same data are required for scenario 1) and 2), three kinds of examples are shown below.

The first example criterion is **short_length_edge**. If this criterion is assigned as a concern, the PDQ checking system shall inspect the target shape model and shall detect any edge whose length is shorter than the given threshold. The definition of the entity **short_length_edge** is shown in 7.4.88. A typical example of short length edge is shown in Figure H.2.

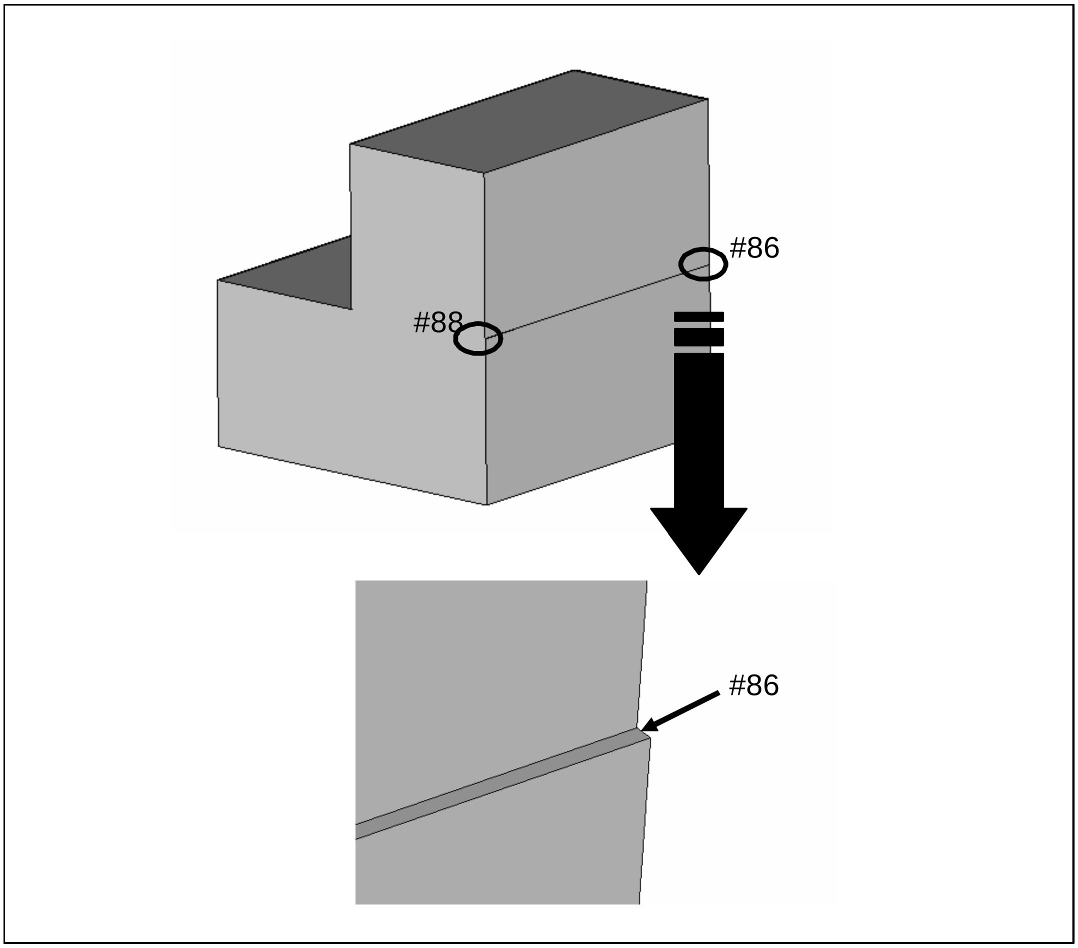

Figure H.2 - A typical example of short length edge

(1) Requirement/declaration of quality for **short_length_edge**.

The first two scenarios are requirement of data quality and declaration of data quality. As described in Annex G, if you are in a position of ordering a design job to some other company, you can use this type of quality information to require the company receiving the order to create a product data that satisfies your prescribed quality requirements. If you are a creator of a product data, you can use this type of quality information for explicitly declaring the quality level satisfied by your model. To create this type of quality information by the use of ISO 10303-59, you can select a set of criteria that composes your requirements. Provide the value of threshold for each criterion and the accuracy of the corresponding measurement if necessary. No reference is made to individual product data in 'requirement' case, though you may or may not attach the product data you created in 'declaration' case. You can assign at most two **data_quality_report_request**s to represent the kind of report you need. See Figure G.1 in EXPRESS-G for entities required for these two scenarios.

As an example of quality information for this scenario, the following data is assumed for the criterion **short_length_edge**.

- The threshold for the detection of **short_length_edge** is smaller than or equal to 0.01mm.

- Two types of report is required; summary report that shows number of inspected instances and number of instances detected as having quality defect as well as the report on the edge instances that are shorter than the given threshold.

The instances to be created for this scenario are shown in Figures H.3 and H.4. 13 instances are created.

NOTE 1 Instances whose name is shown in italics differ depending upon the criterion. In this case, criterion **short_length_edge** requires **shape_data_quality_assessment_by_numerical_test** as its assessment requirement, which in turn requires **shape_data_quality_upper_value_limit** as the threshold.

NOTE 2 4 instances shown in Figure H.4 depict units used in general. If no other unit is specified for specific criterion or measurement requirement, these units are applied.

NOTE 3 Management information such as creator of the criterion and the creation date and time can be attached to **data_quality_definition**. Since representation of this kind of management information depends on each application protocol using ISO 10303-59, these instances are not shown in this example.

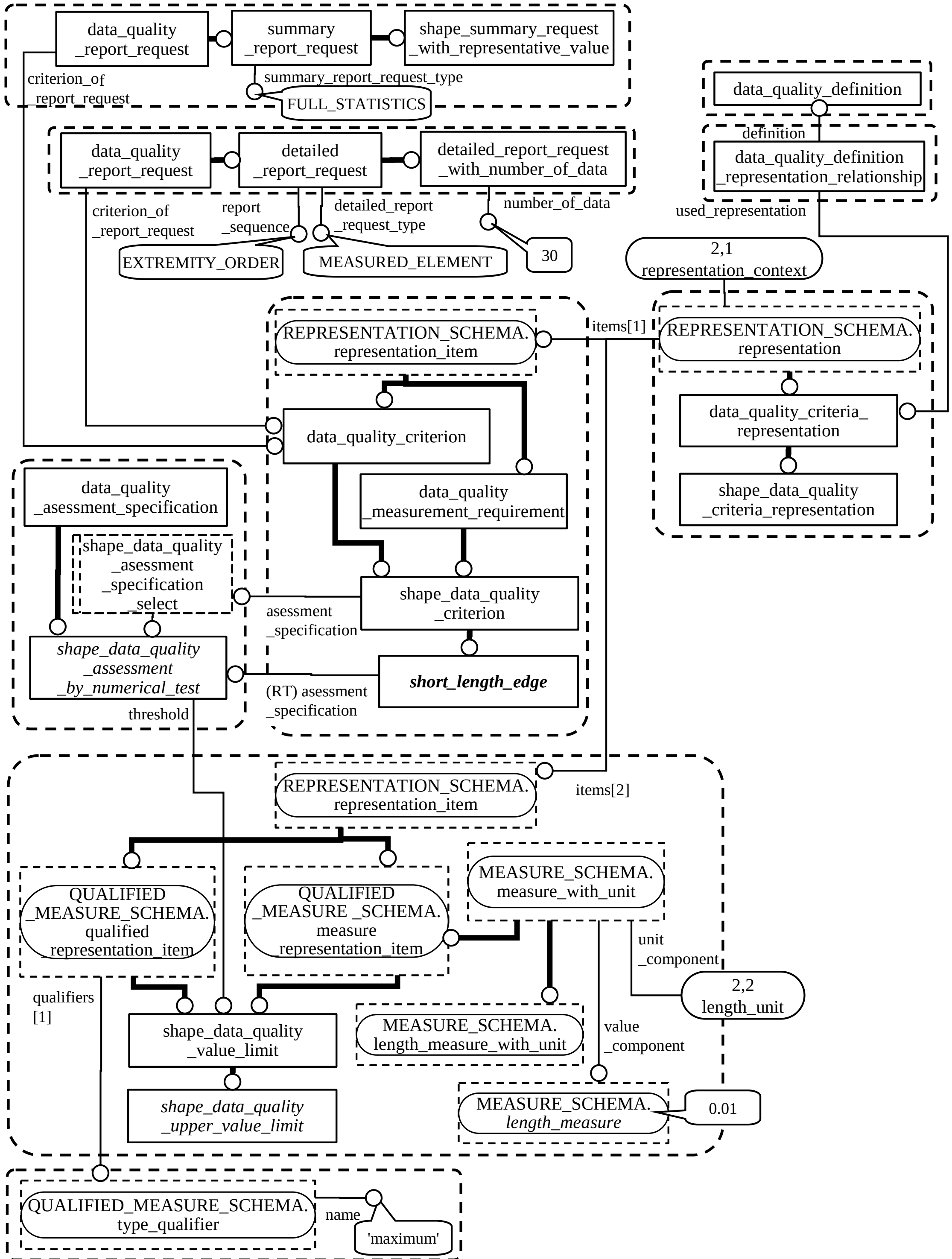

Figure H.3 - Example instances of quality information for use in requirement/declaration of short_length_edge without the accuracy specification (1 of 2)

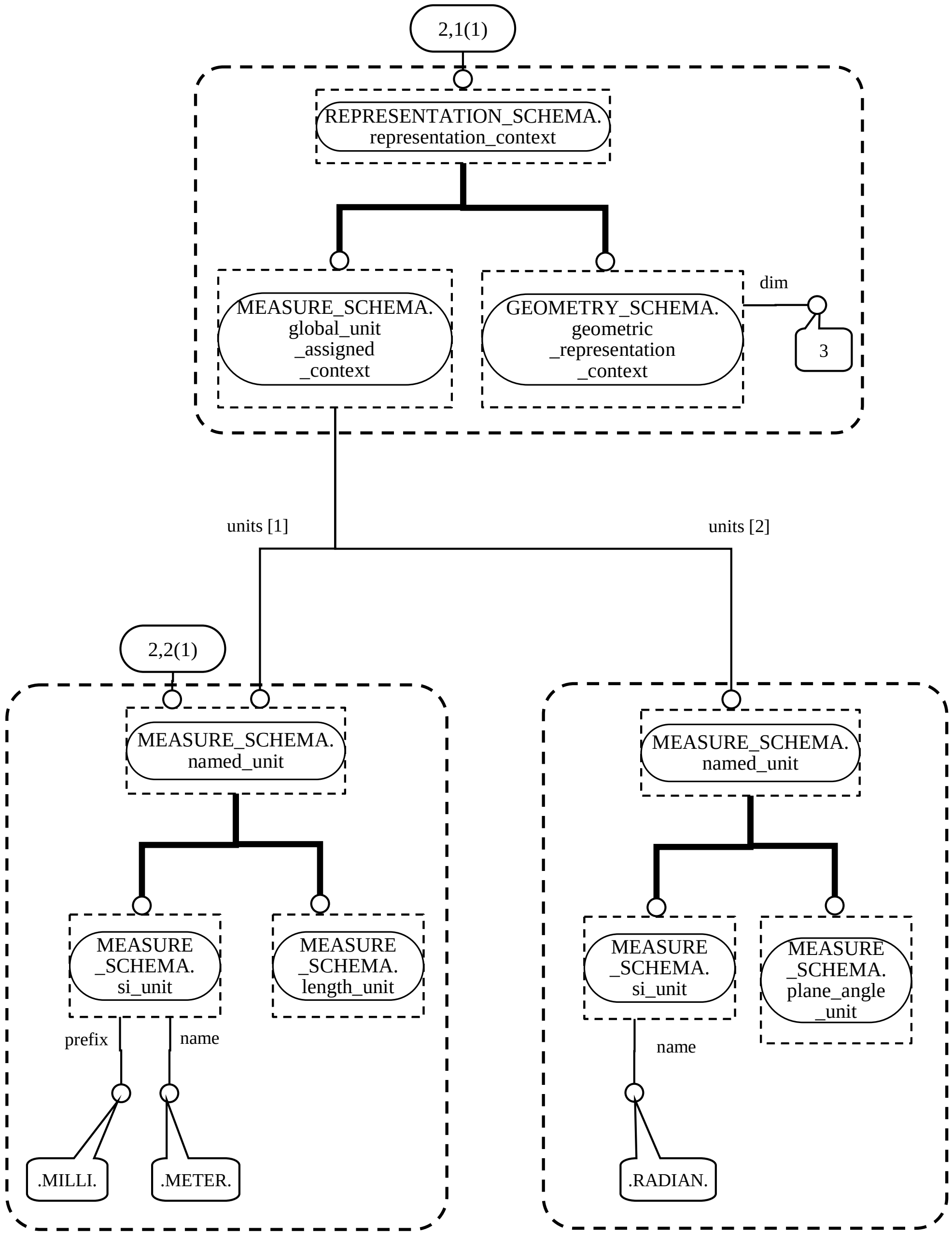

Figure H.4 - Example instances of quality information for use in requirement/declaration of short_length_edge without the accuracy specification (2 of 2)

The data for this example described with the format defined in ISO 10303-21 [6] is as follows.

```
DATA;
#1= DATA_QUALITY_DEFINITION('No short edge is required.');
#2= DATA_QUALITY_DEFINITION_REPRESENTATION_RELATIONSHIP('',#1,#3);
#3= SHAPE_DATA_QUALITY_CRITERIA_REPRESENTATION('sdqc-check1',(#12,#16),#8);
#4= ID_ATTRIBUTE('',#3);
#5= DESCRIPTION_ATTRIBUTE('',#3);
#6= (LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT(.MILLI.,.METRE.));
#7= (NAMED_UNIT(*)PLANE_ANGLE_UNIT()SI_UNIT($,.RADIAN.));
#8= (GEOMETRIC_REPRESENTATION_CONTEXT(3)

GLOBAL_UNIT_ASSIGNED_CONTEXT((#6,#7))REPRESENTATION_CONTEXT('',''));
#12= SHORT_LENGTH_EDGE('',#15);
#13= SHAPE_SUMMARY_REQUEST_WITH_REPRESENTATIVE_VALUE('',#12,
        .FULL_STATISTICS.);
#14= DETAILED_REPORT_REQUEST_WITH_NUMBER_OF_DATA('',#12,.MEASURED_ELEMENT.,
        .EXTREMITY_ORDER.,30);
#15= SHAPE_DATA_QUALITY_ASSESSMENT_BY_NUMERICAL_TEST('threshold:0.01mm',
        #16);
#16= (LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
          MEASURE_WITH_UNIT(LENGTH_MEASURE(0.01),#6)
          QUALIFIED_REPRESENTATION_ITEM((#17))REPRESENTATION_ITEM('upper
limit')
          SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT()
          SHAPE_DATA_QUALITY_VALUE_LIMIT());
#17= TYPE_QUALIFIER('maximum');
ENDSEC;
```

Additionally if you need to specify accuracy for measurement, instances for that purpose should be created as shown in Figures H.5, H.6 and H.7. In this example, the accuracy in general is 0.001mm and the accuracy specific to this criterion is 10^{-5}mm.

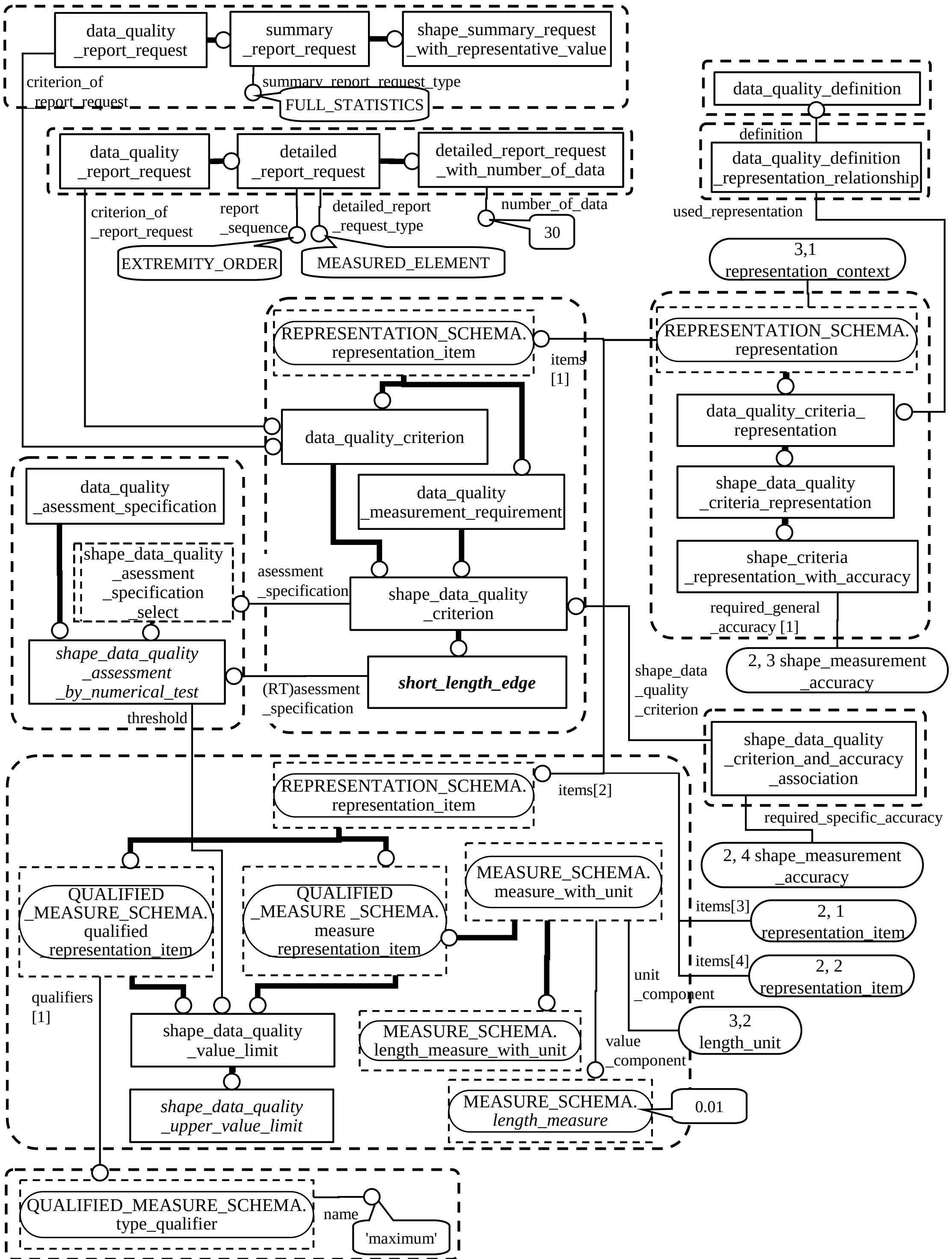

Figure H.5 - Example instances of quality information for use in requirement/declaration of short_length_edge with the accuracy specification (1 of 3)

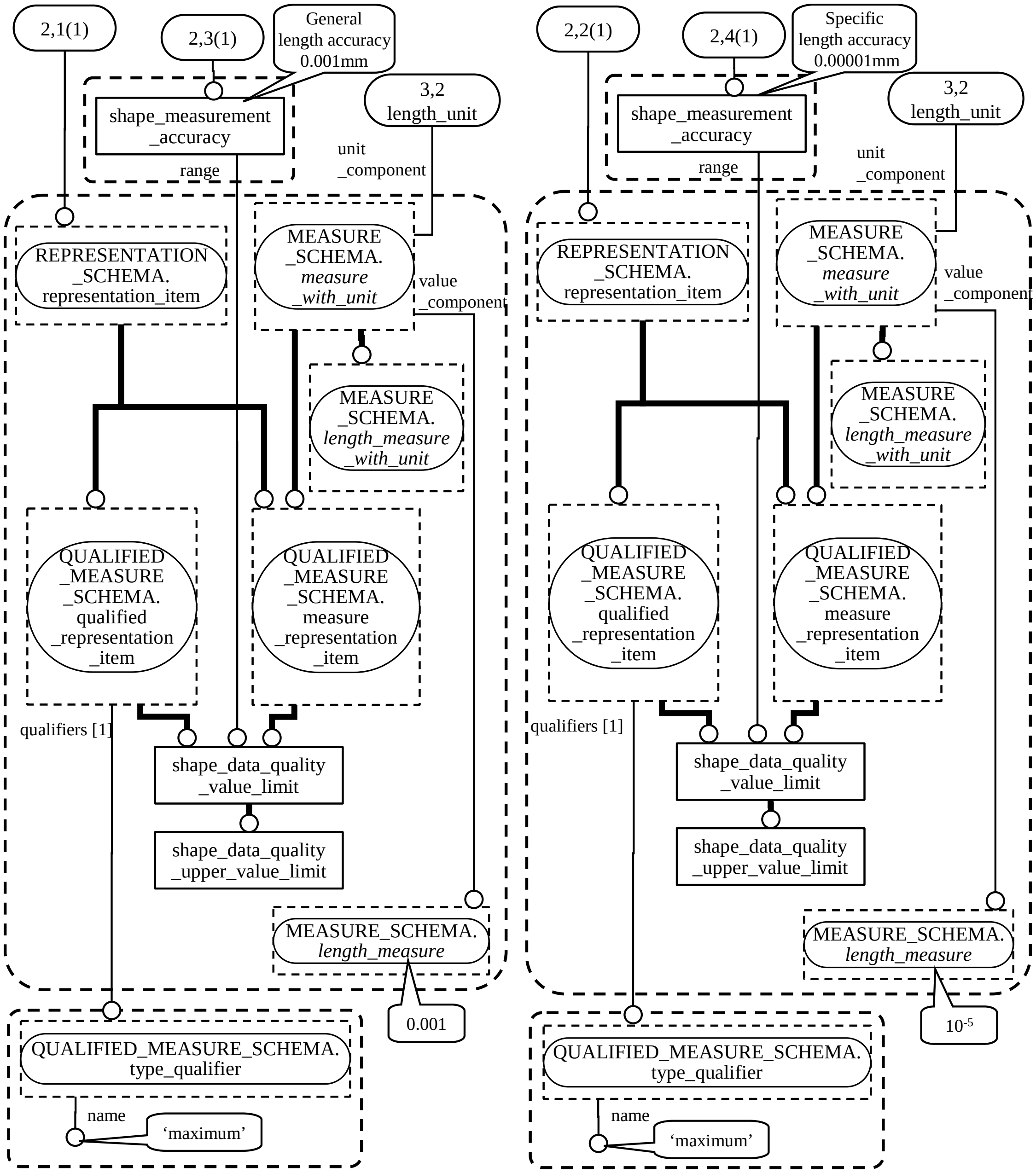

Figure H.6 - Example instances of quality information for use in requirement/declaration of short_length_edge with the accuracy specification (2 of 3)

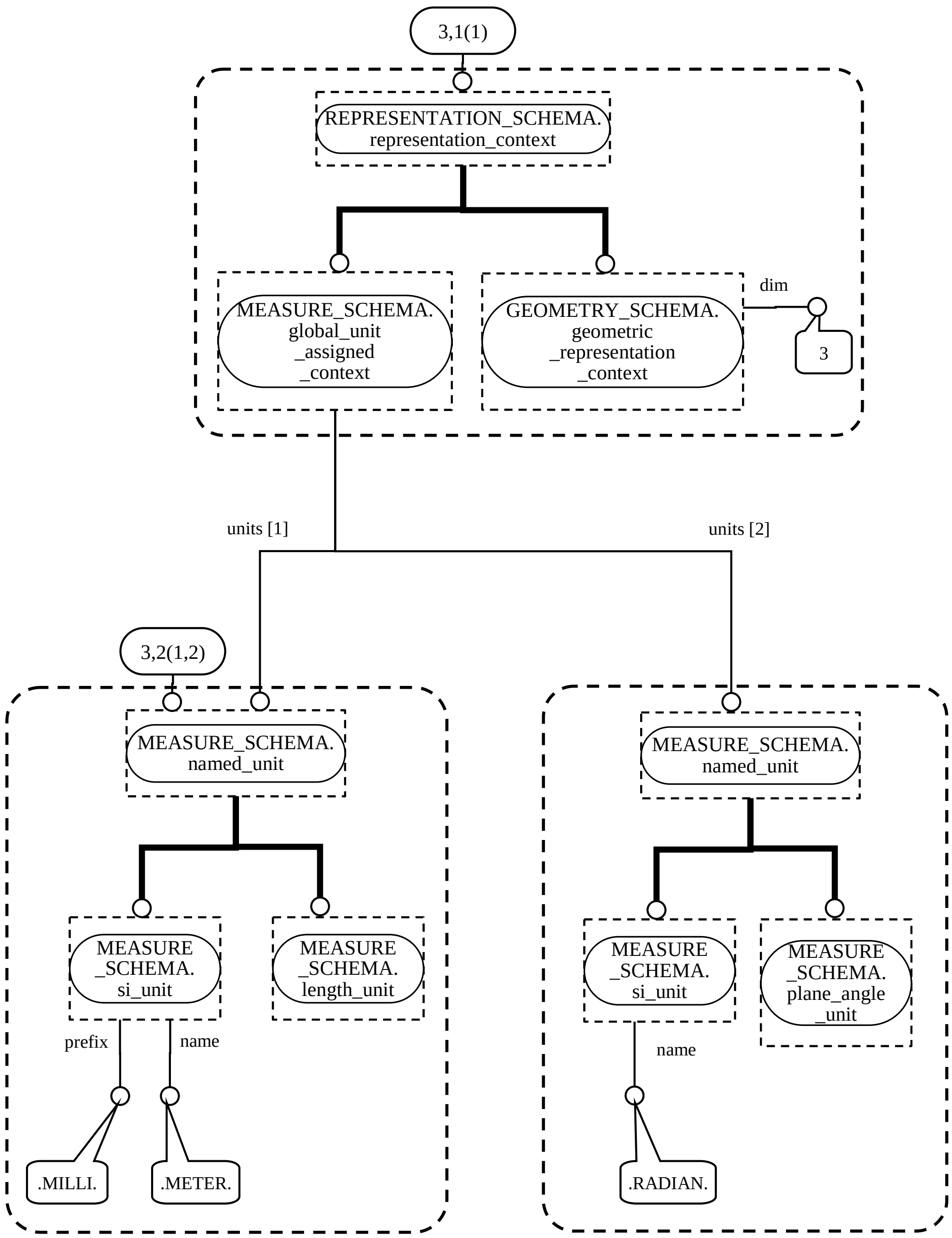

Figure H.7 - Example instances of quality information for use in requirement/declaration of short_length_edge with the accuracy specification (3 of 3)

The data for this example described with the format defined in ISO 10303-21 is as follows.

```
DATA;
#1= DATA_QUALITY_DEFINITION(
  'No short edge is required with the accuracy 0.00001.');
#2= DATA_QUALITY_DEFINITION_REPRESENTATION_RELATIONSHIP('',#1,#3);
#3= SHAPE_CRITERIA_REPRESENTATION_WITH_ACCURACY('sdqc-check1',
       (#12,#16,#21,#26),#8,(#19));
#4= ID_ATTRIBUTE('',#3);
#5= DESCRIPTION_ATTRIBUTE('',#3);
#6= (LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT(.MILLI.,.METRE.));
#7= (NAMED_UNIT(*)PLANE_ANGLE_UNIT()SI_UNIT($,.RADIAN.));
#8= (GEOMETRIC_REPRESENTATION_CONTEXT(3)

GLOBAL_UNIT_ASSIGNED_CONTEXT((#6,#7))REPRESENTATION_CONTEXT('',''));
#12= SHORT_LENGTH_EDGE('',#15);
#13= SHAPE_SUMMARY_REQUEST_WITH_REPRESENTATIVE_VALUE('',#12,
        .FULL_STATISTICS.);
#14= DETAILED_REPORT_REQUEST_WITH_NUMBER_OF_DATA('',#12,
        .MEASURED_ELEMENT.,.EXTREMITY_ORDER.,30);
#15= SHAPE_DATA_QUALITY_ASSESSMENT_BY_NUMERICAL_TEST('threshold:0.01mm',
        #16);
#16= (LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
          MEASURE_WITH_UNIT(LENGTH_MEASURE(0.01),#6)
          QUALIFIED_REPRESENTATION_ITEM((#17))REPRESENTATION_ITEM('upper
limit')
          SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT()
          SHAPE_DATA_QUALITY_VALUE_LIMIT());
#17= TYPE_QUALIFIER('maximum');
#19= SHAPE_MEASUREMENT_ACCURACY('general length accuracy 0.001mm',#21);
#21= (LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
          MEASURE_WITH_UNIT(LENGTH_MEASURE(0.001),#6)
          QUALIFIED_REPRESENTATION_ITEM((#22))REPRESENTATION_ITEM('upper
limit')
          SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT()
          SHAPE_DATA_QUALITY_VALUE_LIMIT());
#22= TYPE_QUALIFIER('maximum');
#24= SHAPE_DATA_QUALITY_CRITERION_AND_ACCURACY_ASSOCIATION(#25,#12);
#25= SHAPE_MEASUREMENT_ACCURACY('specific length accuracy 0.00001mm',#26);
#26= (LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
          MEASURE_WITH_UNIT(LENGTH_MEASURE(1.0000000E-5),#6)
          QUALIFIED_REPRESENTATION_ITEM((#27))REPRESENTATION_ITEM('upper
limit')
          SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT()
          SHAPE_DATA_QUALITY_VALUE_LIMIT());
#27= TYPE_QUALIFIER('maximum');
ENDSEC;
```

(2) Assurance of quality for **short_length_edge**.

The second scenario is assurance of data quality. When you are responsible for quality assurance, you may use this type of quality information for representing the results of quality inspection against particular product model data. To create this type of quality information by the use of ISO 10303-59, you should create inspection results with references to both requirements and inspected product data. Inspection results shall be created so that each inspection result corresponds to exactly one particular criterion specified. You shall create each inspection result with the type of report specified by **data_quality_report_request**. The entities related to this scenario are as shown in Figure G.2 of Annex G.

An example for this scenario presented below is based on the following assumptions;
- The requirement is to detect edges shorter than or equal to 0. 01mm with the accuracy of 10^{-5}mm and to output summary report of the inspection.
- The inspection is performed against the shape data of a product model. The ID of the product model is P#319 and the ID of the shape representation is P#304.
- As the result of inspection, 24 edges are inspected and 2 edges are found to be within the given threshold. The length of the shortest edge detected is 0.009mm. The accuracy actually applied to this inspection is 10^{-5}mm for this criterion. The general accuracy applied is 10^{-3}mm.

The instances to be created for this scenario are as shown in Figures H.8, H.9, H.10, H.11, H.12 and H.13.

NOTE Instances without circumscribed broken lines are not created with entities from this part of ISO 10303: examples are **product_definition** and **shape_representation** of inspected product in Figure H.11. Identifiers for these instances are shown with prefix ‘P’ such as ‘product_definition P#319’.

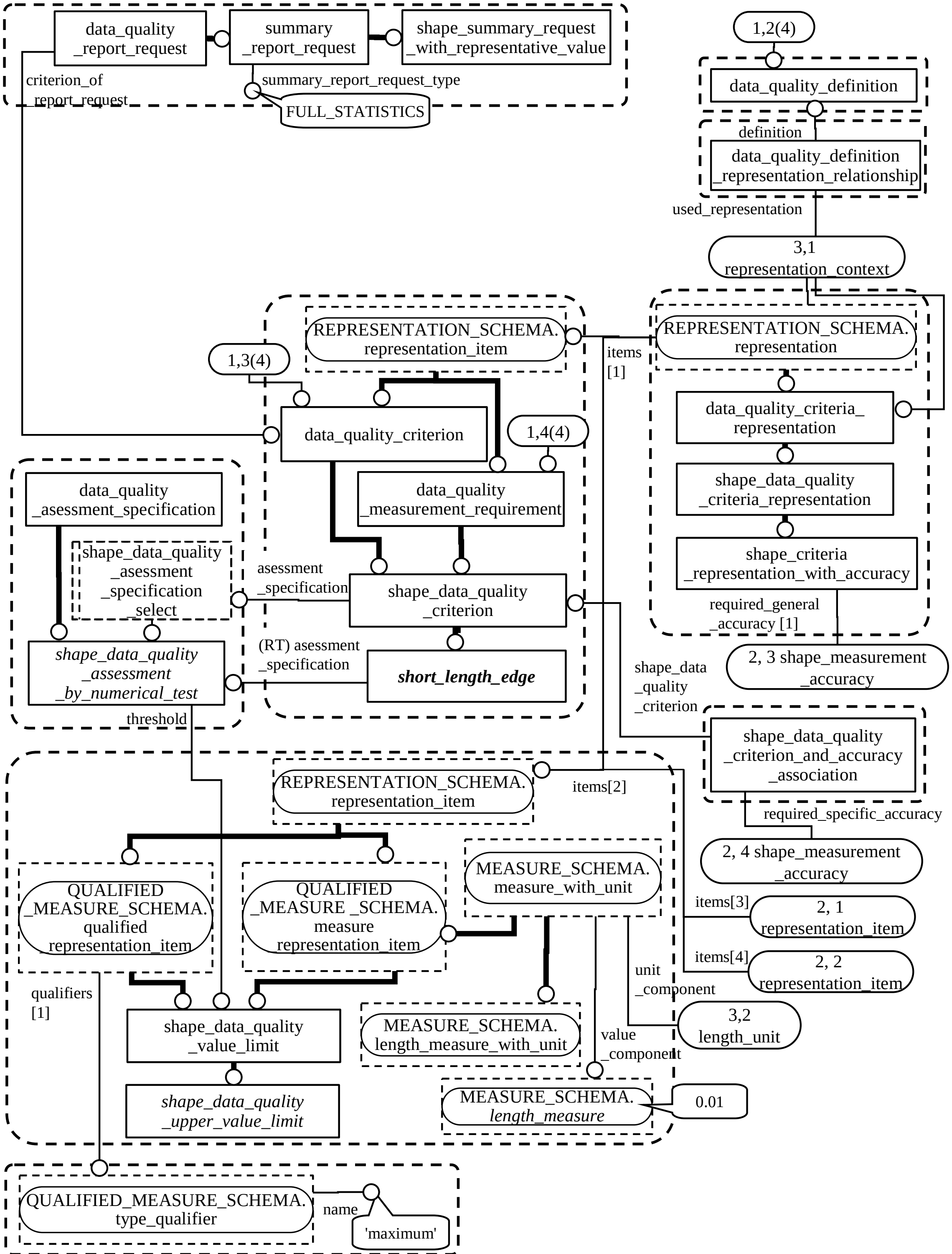

Figure H.8- Example instances of quality information for use in assurance of short_length_edge (1 of 6)

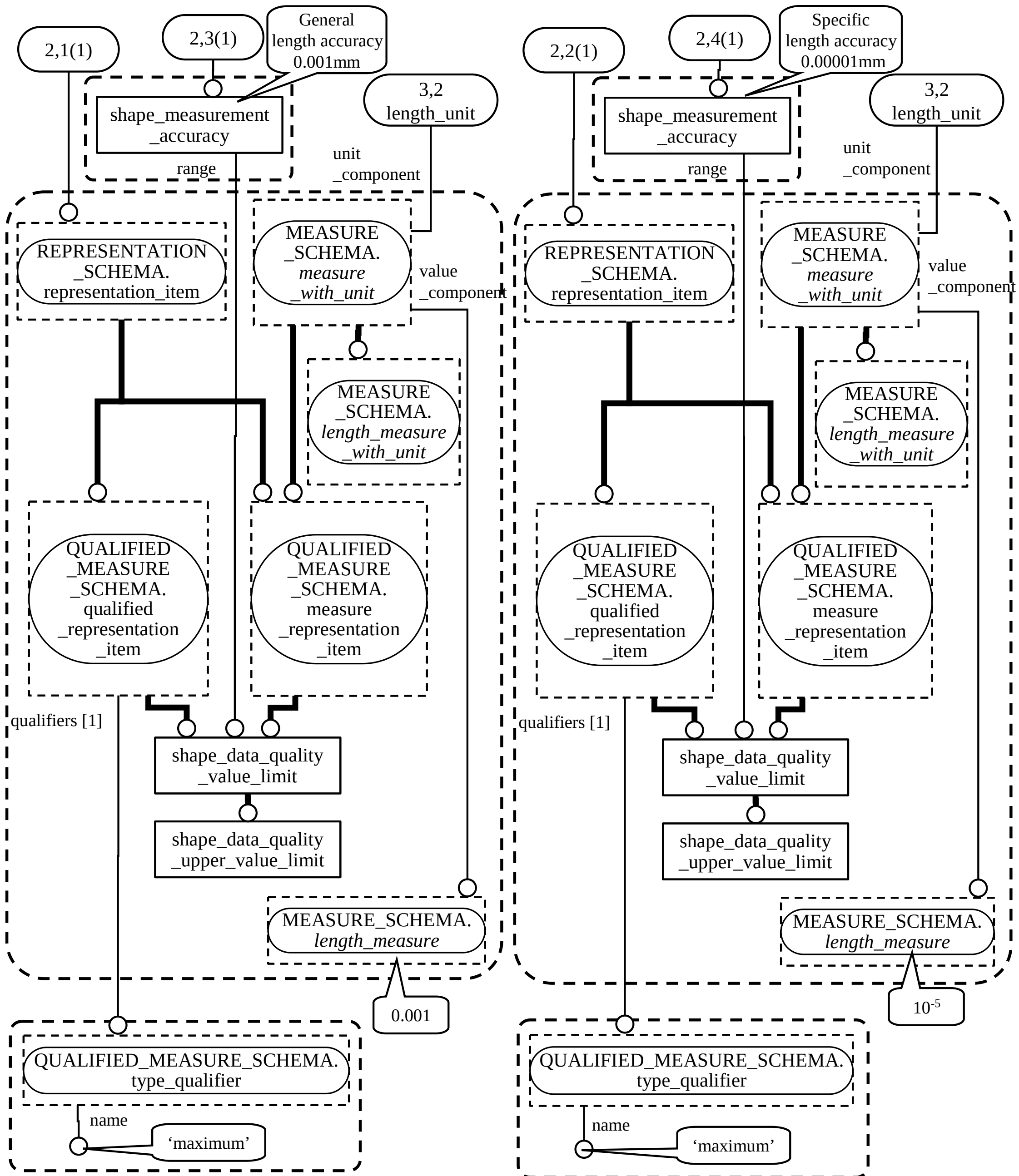

Figure H.9 - Example instances of quality information for use in assurance of short_length_edge (2 of 6)

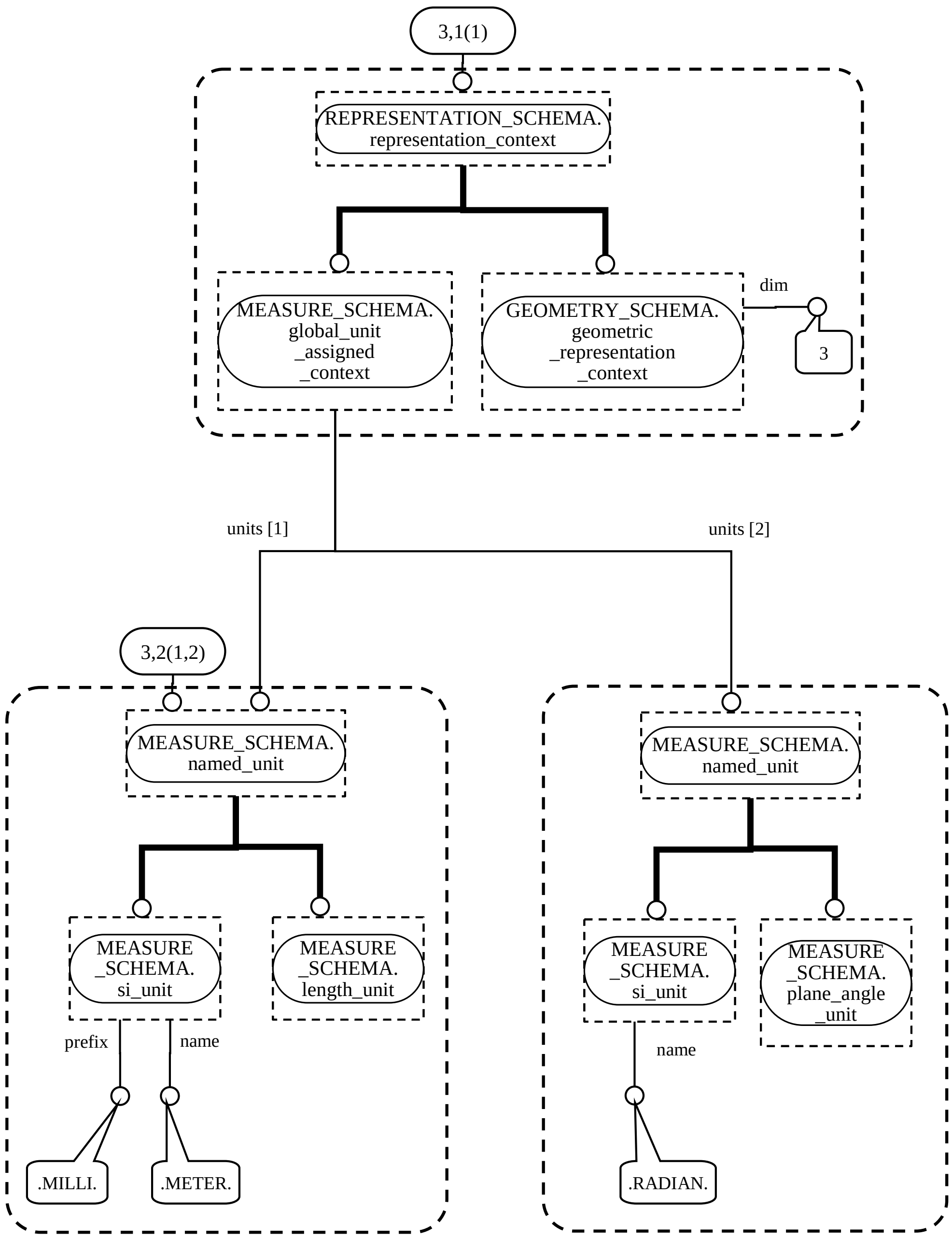

Figure H.10 - Example instances of quality information for use in assurance of short_length_edge (3 of 6)

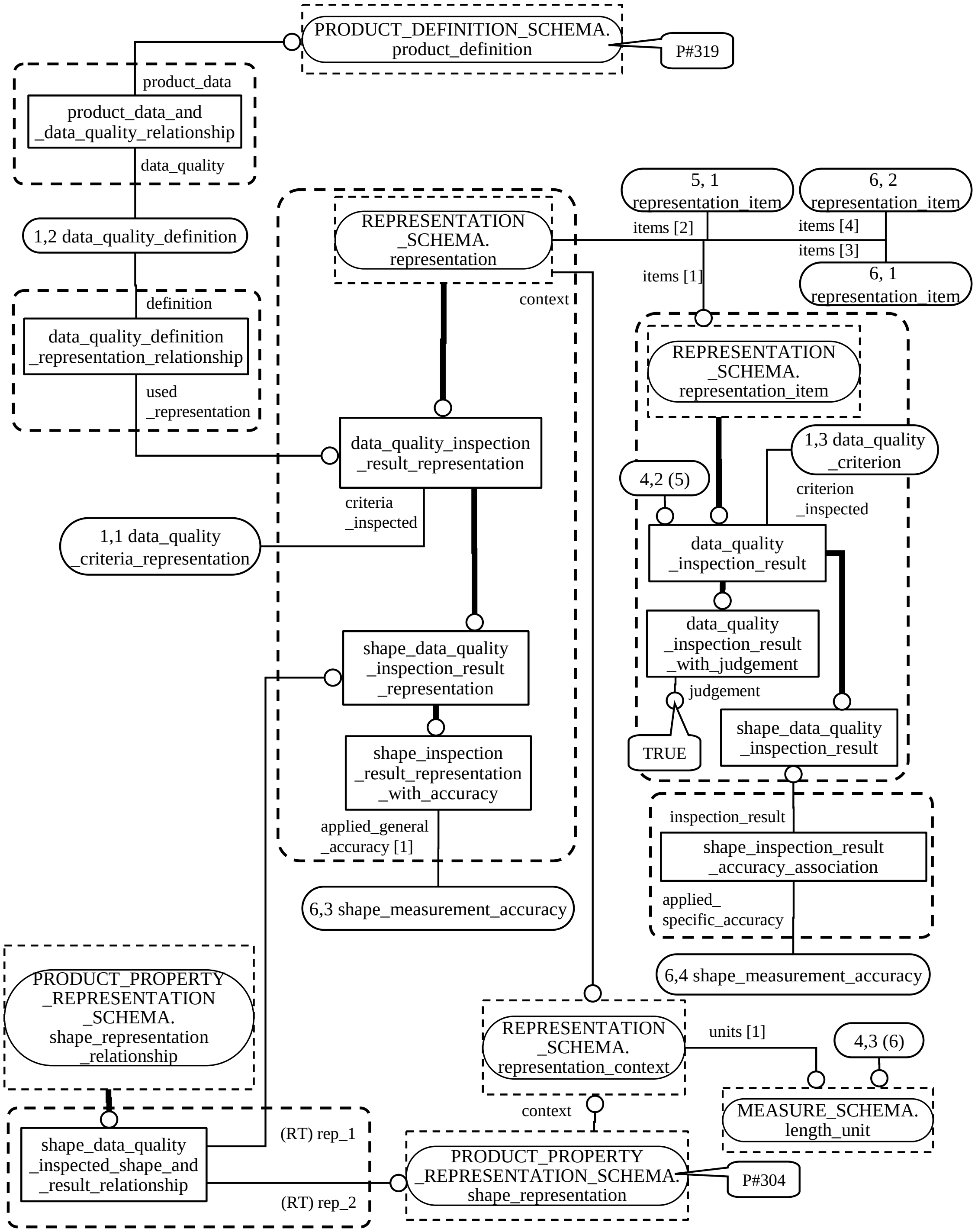

Figure H.11 - Example instances of quality information for use in assurance of short_length_edge (4 of 6)

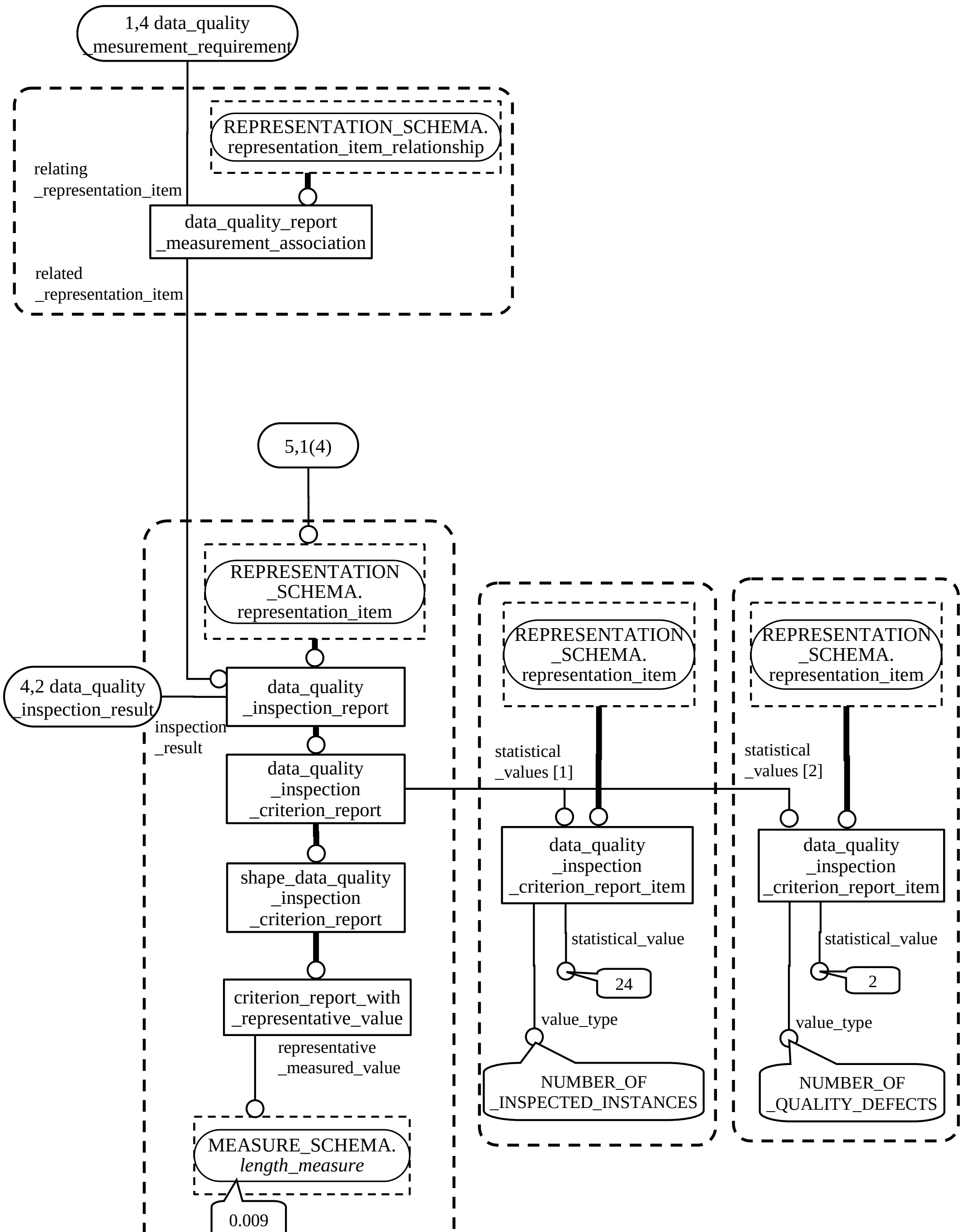

Figure H.12 - Example instances of quality information for use in assurance of short_length_edge (5 of 6)

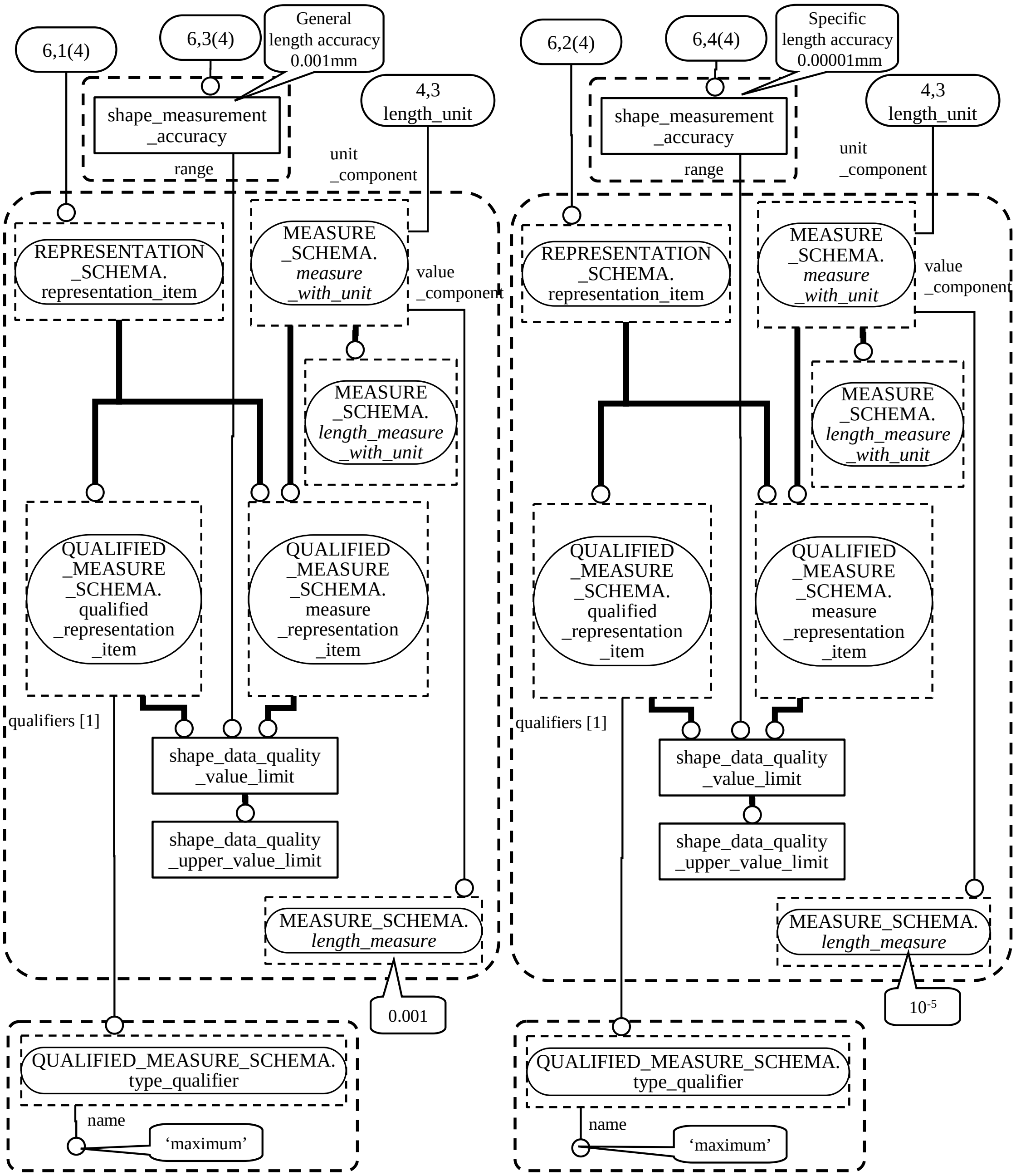

Figure H.13 - Example instances of quality information for use in assurance of short_length_edge (6 of 6)

The data for this example described with the format defined in ISO 10303-21 is as follows.

```
DATA;
#1= PRODUCT_DEFINITION('target product data',$,#3,#11);
#2= NAME_ATTRIBUTE('P#319',#1);
#3= PRODUCT_DEFINITION_FORMATION('target data1',$,#4);
#4= PRODUCT('','',$,(#6));
#6= PRODUCT_CONTEXT('',#7,'mechanical');
#7= APPLICATION_CONTEXT(
       'configuration  controlled  3D  designs  of  mechanical  parts  and
assemblies');
#9= ID_ATTRIBUTE('',#7);
#10= DESCRIPTION_ATTRIBUTE('',#7);
#11= PRODUCT_DEFINITION_CONTEXT('',#7,'design');
#12= SHAPE_REPRESENTATION('target shape_representation',(#86,#88),#17);
#13= ID_ATTRIBUTE('P#304',#12);
#14= DESCRIPTION_ATTRIBUTE('',#12);
#15= (LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT(.MILLI.,.METRE.));
#16= (NAMED_UNIT(*)PLANE_ANGLE_UNIT()SI_UNIT($,.RADIAN.));
#17= (GEOMETRIC_REPRESENTATION_CONTEXT(3)
          GLOBAL_UNIT_ASSIGNED_CONTEXT((#15,#16))
          REPRESENTATION_CONTEXT('',''));
#20= PRODUCT_DEFINITION_SHAPE('','',#1);
#21= ID_ATTRIBUTE('',#20);
#22= SHAPE_DEFINITION_REPRESENTATION(#20,#12);
#23= DESCRIPTION_ATTRIBUTE('',#22);
#24= NAME_ATTRIBUTE('',#22);
#30= CURVE('');
#31= CURVE('');
#86= EDGE_CURVE('P#86',#30,.T.);
#88= EDGE_CURVE('P#88',#31,.T.);
#101= DATA_QUALITY_DEFINITION(
          'Short edge is detected with the accuracy 0.00001.');
#102= PRODUCT_DATA_AND_DATA_QUALITY_RELATIONSHIP('',#1,#101);
#103= DATA_QUALITY_DEFINITION_REPRESENTATION_RELATIONSHIP('',#101,#107);
#104= (LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT(.MILLI.,.METRE.));
#105= (NAMED_UNIT(*)PLANE_ANGLE_UNIT()SI_UNIT($,.RADIAN.));
#106= ID_ATTRIBUTE('',#107);
#107= SHAPE_CRITERIA_REPRESENTATION_WITH_ACCURACY('',(#114,#117,#122,#127),
          #109,(#120));
#108= DESCRIPTION_ATTRIBUTE('',#107);
#109= (GEOMETRIC_REPRESENTATION_CONTEXT(3)
           GLOBAL_UNIT_ASSIGNED_CONTEXT((#104,#105))
           REPRESENTATION_CONTEXT('',''));
#114= SHORT_LENGTH_EDGE('',#116);
#115= SHAPE_SUMMARY_REQUEST_WITH_REPRESENTATIVE_VALUE('',#114,
          .FULL_STATISTICS.);
#116= SHAPE_DATA_QUALITY_ASSESSMENT_BY_NUMERICAL_TEST('threshold:0.01mm',
```

```
          #117);
#117= (LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
            MEASURE_WITH_UNIT(LENGTH_MEASURE(0.01),#104)
            QUALIFIED_REPRESENTATION_ITEM((#118))
            REPRESENTATION_ITEM('upper limit')
            SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT()
            SHAPE_DATA_QUALITY_VALUE_LIMIT());
#118= TYPE_QUALIFIER('maximum');
#120= SHAPE_MEASUREMENT_ACCURACY('General length accuracy 0.001mm',#122);
#122= (LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
            MEASURE_WITH_UNIT(LENGTH_MEASURE(0.001),#104)
            QUALIFIED_REPRESENTATION_ITEM((#123))
            REPRESENTATION_ITEM('upper limit')
            SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT()
            SHAPE_DATA_QUALITY_VALUE_LIMIT());
#123= TYPE_QUALIFIER('maximum');
#125= SHAPE_DATA_QUALITY_CRITERION_AND_ACCURACY_ASSOCIATION(#126,#114);
#126= SHAPE_MEASUREMENT_ACCURACY('Specific length accuracy 0.00001mm',#127);
#127= (LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
            MEASURE_WITH_UNIT(LENGTH_MEASURE(1.0000000E-5),#104)
            QUALIFIED_REPRESENTATION_ITEM((#128))
            REPRESENTATION_ITEM('upper limit')
            SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT()
            SHAPE_DATA_QUALITY_VALUE_LIMIT());
#128= TYPE_QUALIFIER('maximum');
#130= SHAPE_INSPECTION_RESULT_REPRESENTATION_WITH_ACCURACY('sdqir-check1',
         (#135,#137,#144,#149),#17,#107,(#142));
#131= DATA_QUALITY_DEFINITION_REPRESENTATION_RELATIONSHIP('',#101,#130);
#132= SHAPE_DATA_QUALITY_INSPECTED_SHAPE_AND_RESULT_RELATIONSHIP('','',#12,
         #130);
#133= ID_ATTRIBUTE('',#130);
#134= DESCRIPTION_ATTRIBUTE('',#130);
#135= (DATA_QUALITY_INSPECTION_RESULT(#114)
           DATA_QUALITY_INSPECTION_RESULT_WITH_JUDGEMENT(.T.)
           REPRESENTATION_ITEM('')SHAPE_DATA_QUALITY_INSPECTION_RESULT());
#137= SHAPE_DATA_QUALITY_INSPECTION_CRITERION_REPORT('',#135,(#139,#140),
         LENGTH_MEASURE(0.009));
#139= DATA_QUALITY_INSPECTION_CRITERION_REPORT_ITEM('',24,
         .NUMBER_OF_INSPECTED_INSTANCES.);
#140= DATA_QUALITY_INSPECTION_CRITERION_REPORT_ITEM('',2,
         .NUMBER_OF_QUALITY_DEFECTS_DETECTED.);
#141= DATA_QUALITY_REPORT_MEASUREMENT_ASSOCIATION('','',#114,#137);
#142= SHAPE_MEASUREMENT_ACCURACY('General length accuracy 0.001mm',#144);
#144= (LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
           MEASURE_WITH_UNIT(LENGTH_MEASURE(0.001),#15)
           QUALIFIED_REPRESENTATION_ITEM((#145))
           REPRESENTATION_ITEM('upper limit')
           SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT()
```

```
          SHAPE_DATA_QUALITY_VALUE_LIMIT());
#145= TYPE_QUALIFIER('maximum');
#147= SHAPE_INSPECTION_RESULT_ACCURACY_ASSOCIATION(#148,#135);
#148= SHAPE_MEASUREMENT_ACCURACY('Specific length accuracy 0.00001mm',#149);
#149= (LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
          MEASURE_WITH_UNIT(LENGTH_MEASURE(1.0000000E-5),#15)
          QUALIFIED_REPRESENTATION_ITEM((#150))
          REPRESENTATION_ITEM('upper limit')
          SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT()
          SHAPE_DATA_QUALITY_VALUE_LIMIT());
#150= TYPE_QUALIFIER('maximum');
ENDSEC;
```

(3) Quality information for use in the improvement of quality for **short_length_edge**.

The last scenario is data quality information to be used in quality improvement. This scenario assumes that the detailed information obtained for detecting **short_length_edge** by a PDQ checking system will be used in the later stage for healing problem data by the use of some PDQ healing system, or by manual work. PDQ checking system is required to generate the information on what quality defect is detected to what extent and provide it as a detailed inspection result report at the geometric entity instance level. Inspection result shall be created for each measurement identifying quality defects detected.

An example for this scenario presented below is based on the following assumptions;
- The requirement is to detect edges shorter than or equal to 0.01mm with the accuracy of 10^{-5}mm.
- For the purpose of this scenario, it is necessary to generate the report showing the edges that are shorter in length than or equal to 0.01mm and their measured length. The reports should be created in extremity order.
- The summary report is the same as the example in (2).
- The detailed information on the two edges that are detected as **short_length_edge** is as follows.
  - edge_curve #86 has the length 0.009mm.
  - edge_curve #88 has the length 0.009mm.

The instances to be created for this scenario are shown in Figures H.14, H.15, H.16, H.17, H.18, H.19 and H.20.

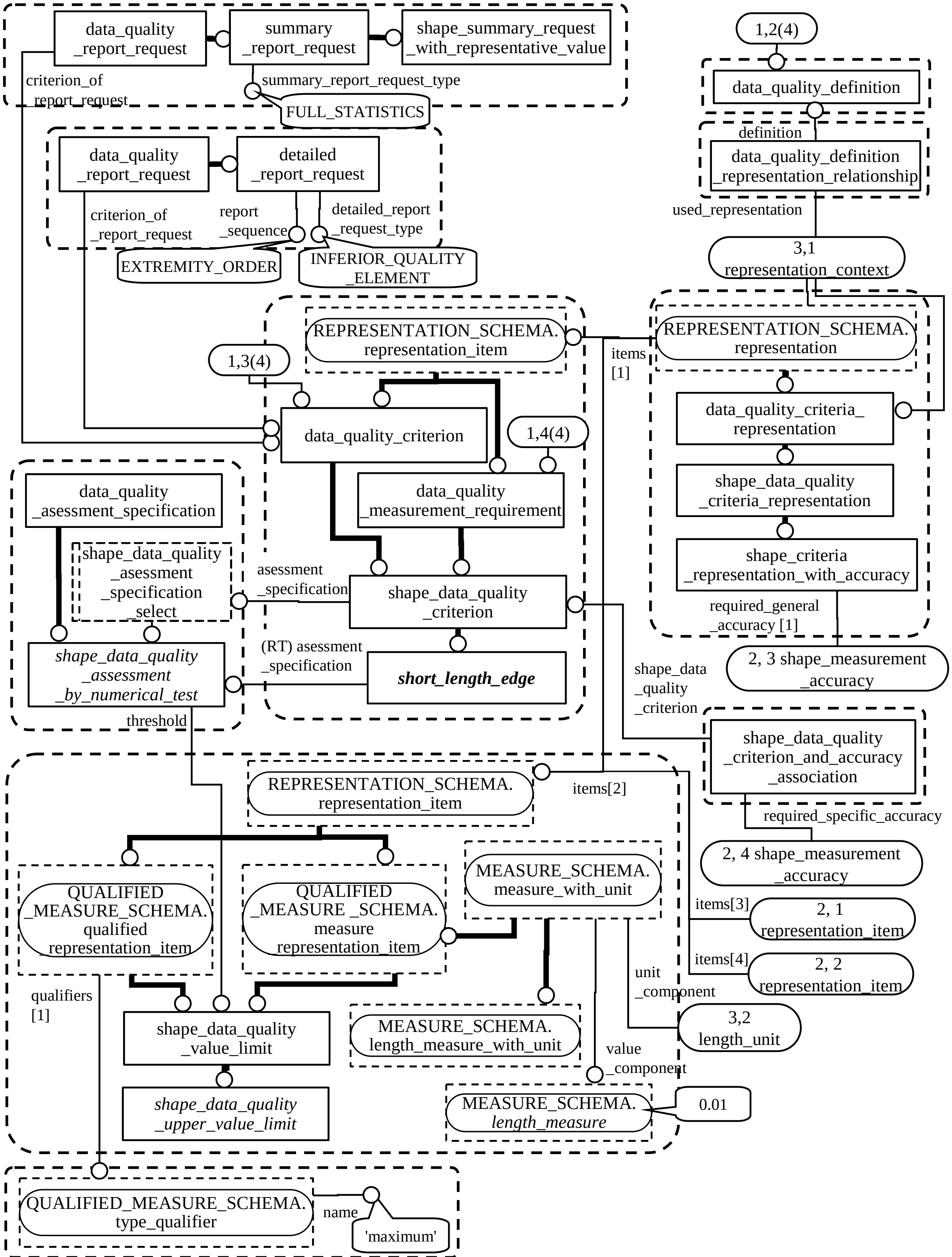

Figure H.14 - Example instances of quality information for use in the improvement of quality short_length_edge (1 of 7)

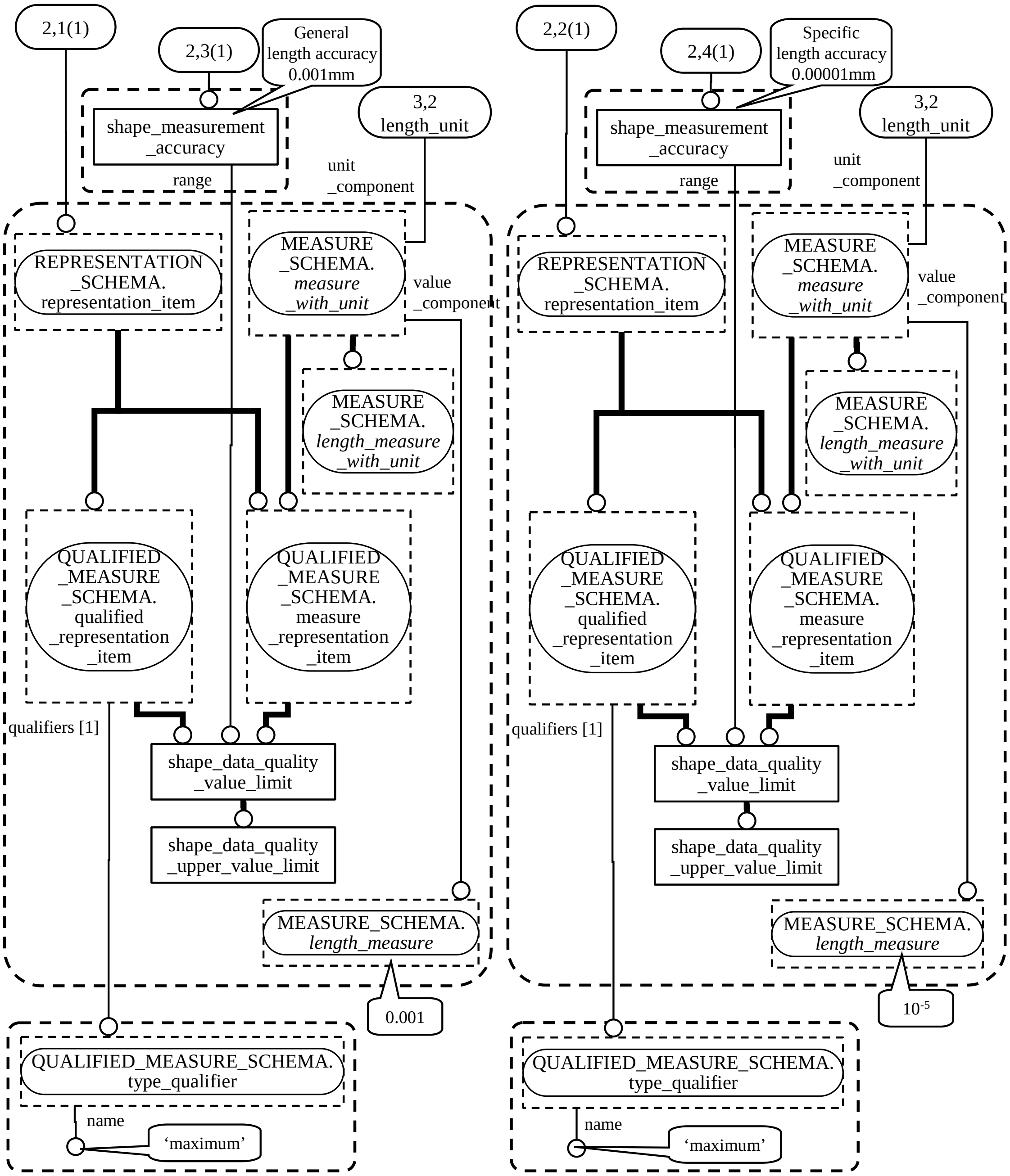

Figure H.15 - Example instances of quality information for use in the improvement of quality short_length_edge (2 of 7)

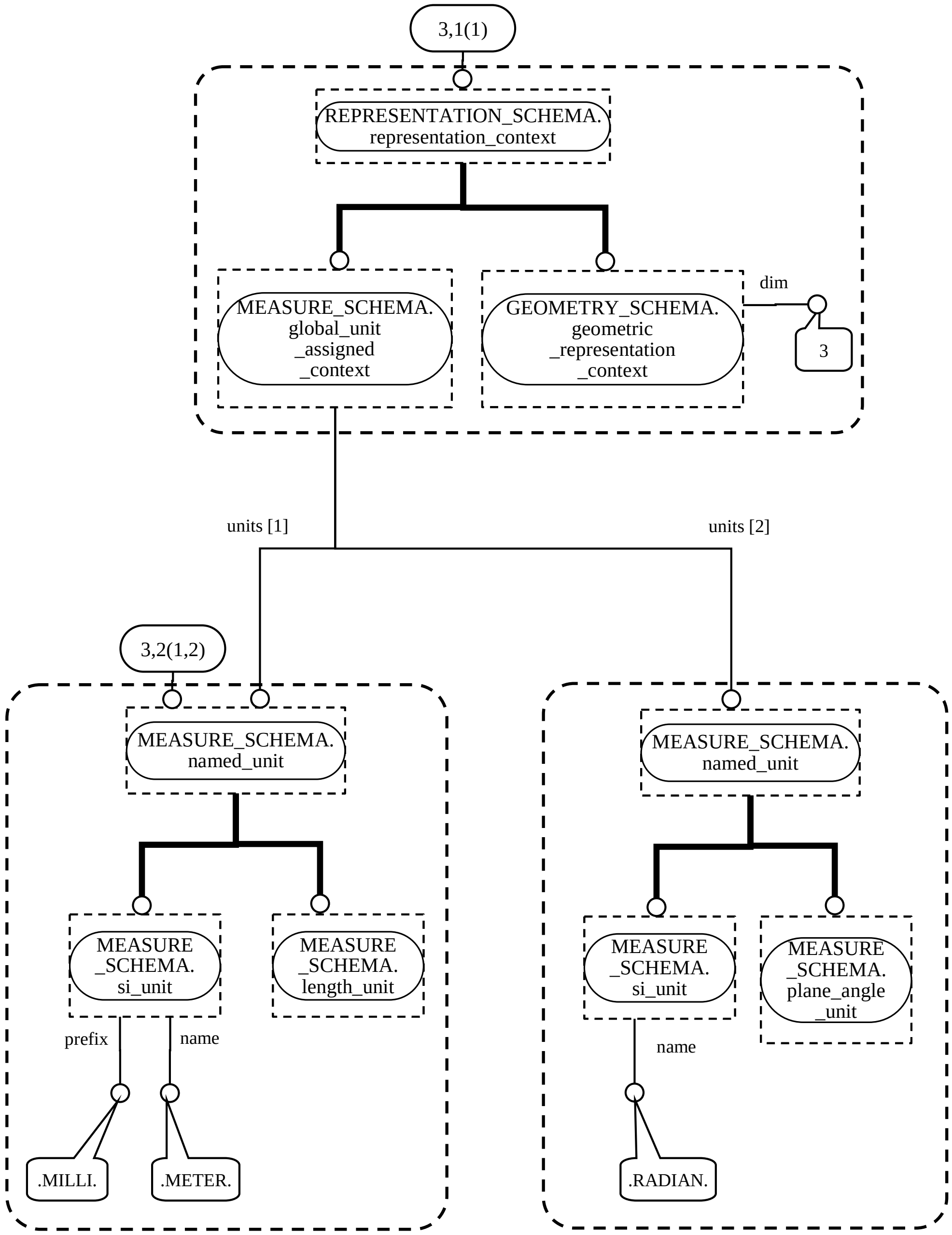

Figure H.16 - Example instances of quality information for use in the improvement of quality short_length_edge (3 of 7)

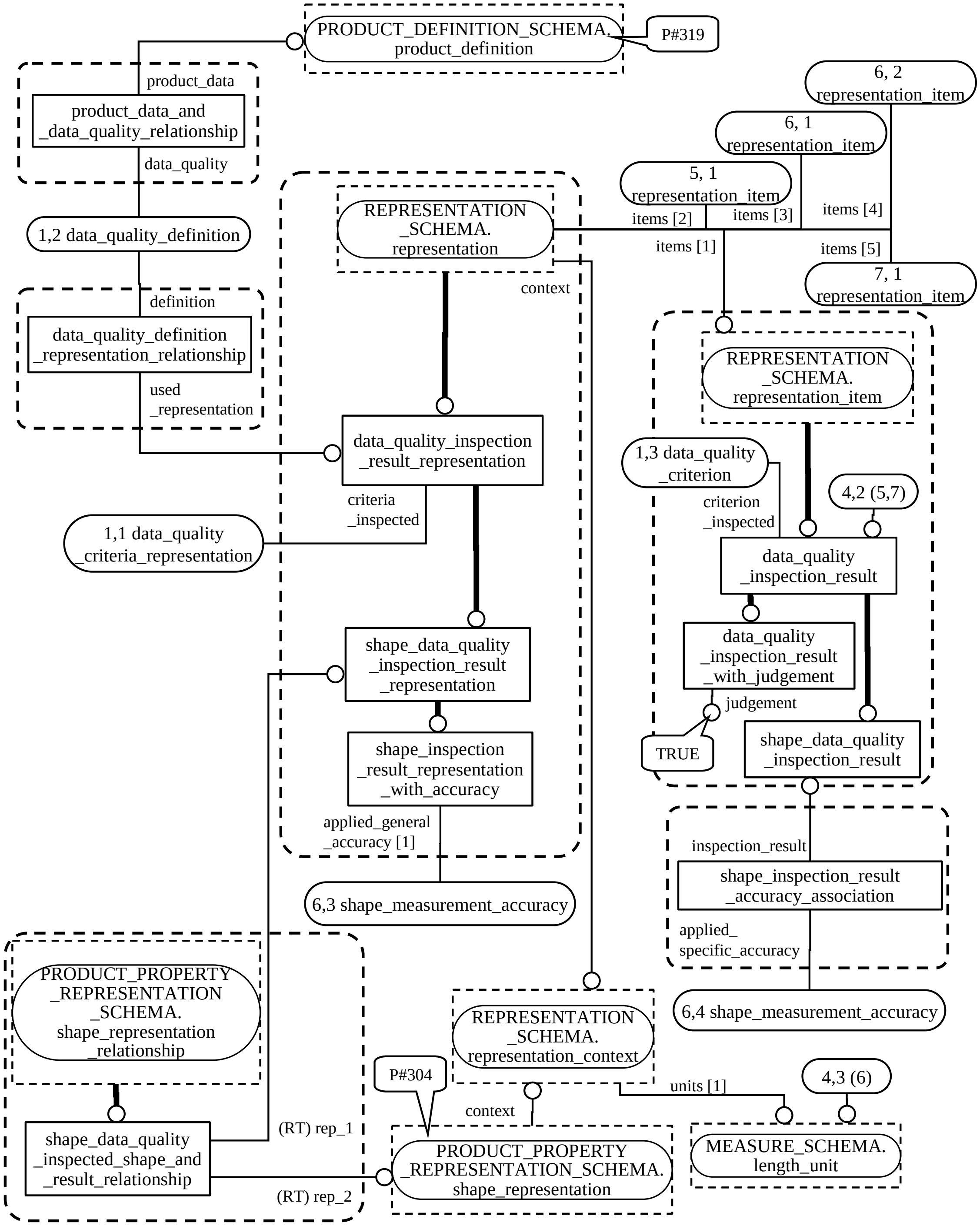

Figure H.17 - Example instances of quality information for use in the improvement of quality short_length_edge (4 of 7)

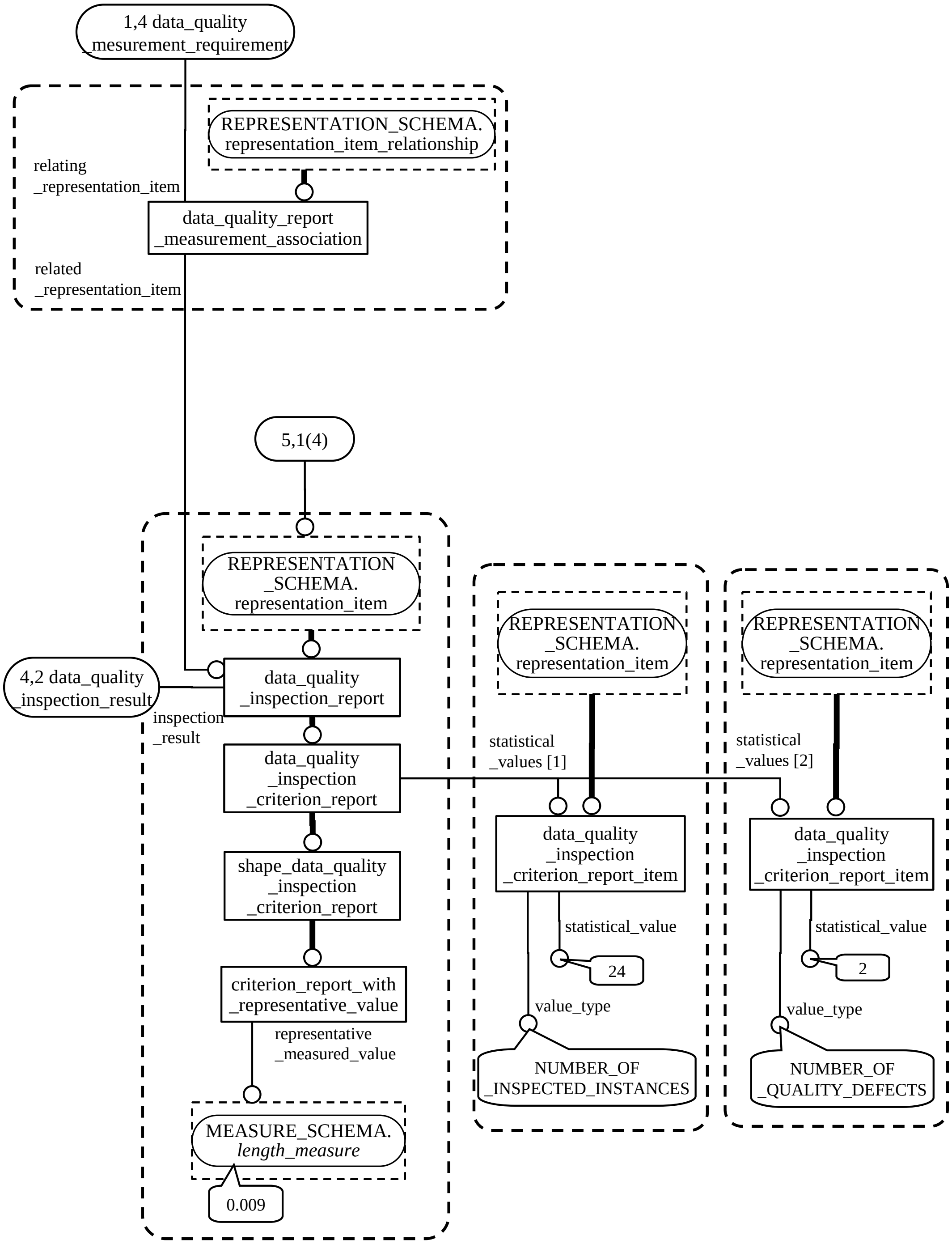

Figure H.18 - Example instances of quality information for use in the improvement of quality short_length_edge (5 of 7)

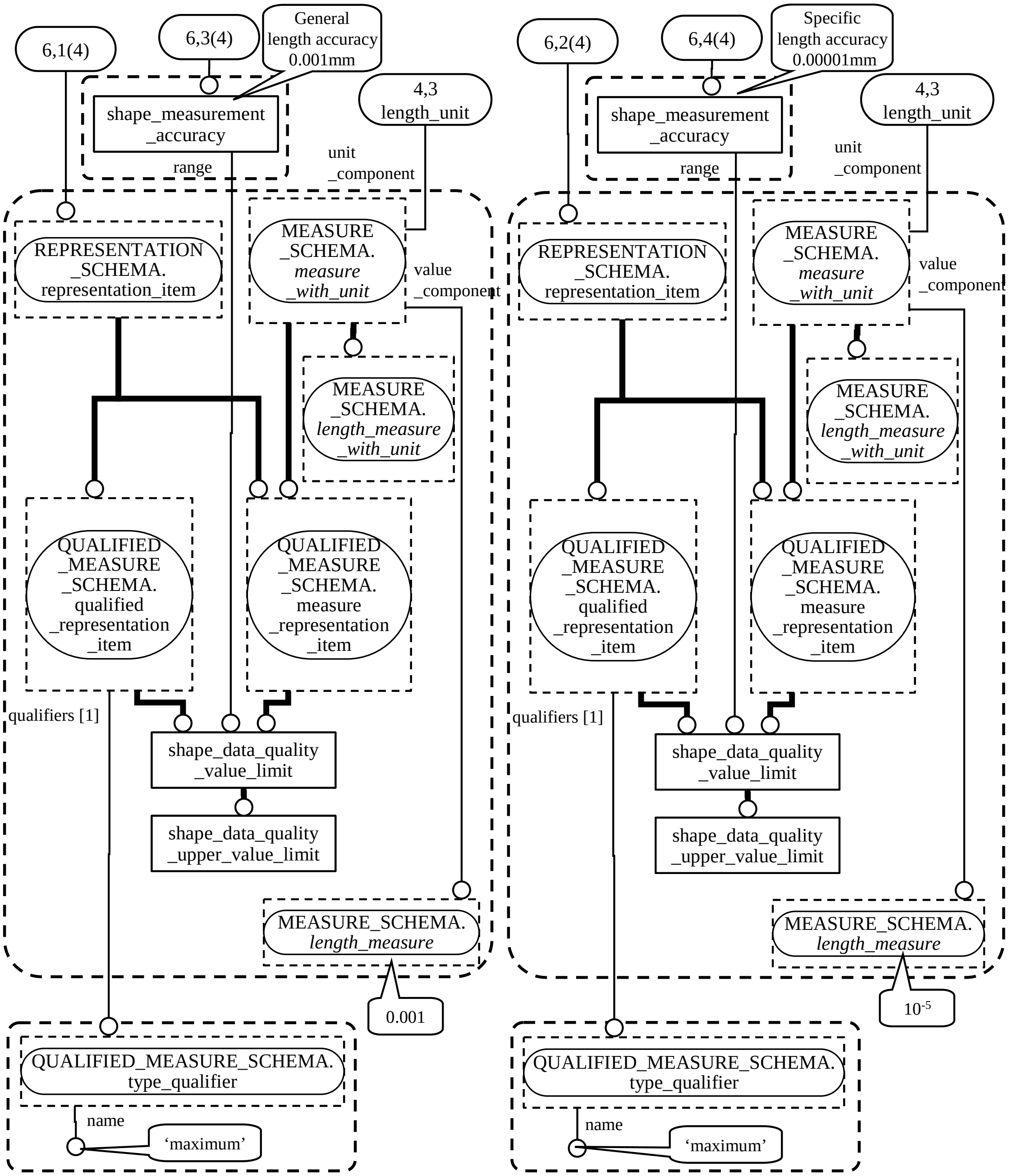

Figure H.19 - Example instances of quality information for use in the improvement of quality short_length_edge (6 of 7)

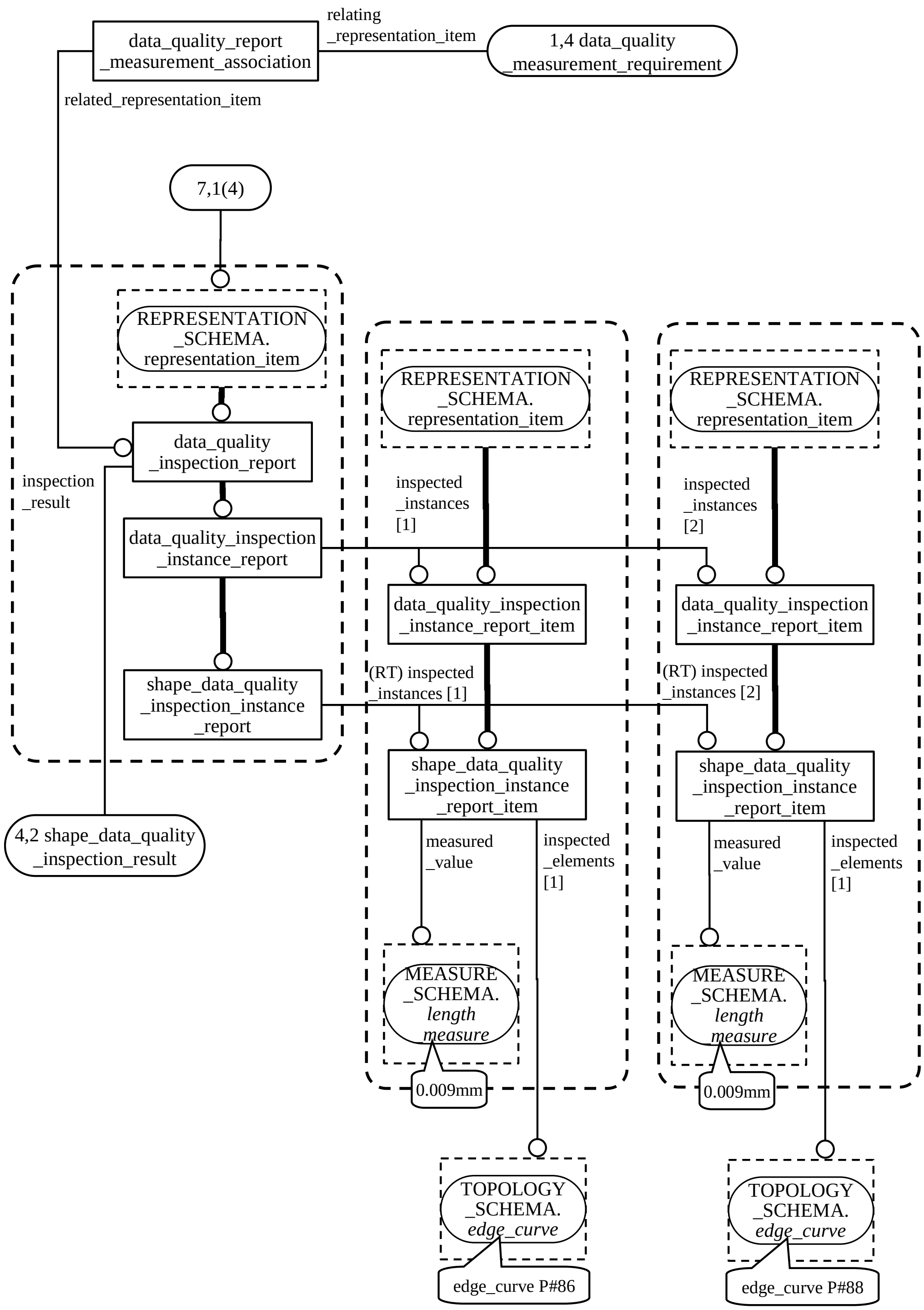

Figure H.20 - Example instances of quality information for use in the improvement of quality short_length_edge (7 of 7)

The data for this example described with the format defined in ISO 10303-21 is as follows.

```
DATA;
#1= PRODUCT_DEFINITION('target product data',$,#3,#11);
#2= NAME_ATTRIBUTE('P#319',#1);
#3= PRODUCT_DEFINITION_FORMATION('target data1',$,#4);
#4= PRODUCT('','',$,(#6));
#6= PRODUCT_CONTEXT('',#7,'mechanical');
#7= APPLICATION_CONTEXT(
      'configuration  controlled  3D  designs  of  mechanical  parts  and
assemblies');
#9= ID_ATTRIBUTE('',#7);
#10= DESCRIPTION_ATTRIBUTE('',#7);
#11= PRODUCT_DEFINITION_CONTEXT('',#7,'design');
#12= SHAPE_REPRESENTATION('target shape_representation',(#86,#88),#17);
#13= ID_ATTRIBUTE('P#304',#12);
#14= DESCRIPTION_ATTRIBUTE('',#12);
#15= (LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT(.MILLI.,.METRE.));
#16= (NAMED_UNIT(*)PLANE_ANGLE_UNIT()SI_UNIT($,.RADIAN.));
#17= (GEOMETRIC_REPRESENTATION_CONTEXT(3)
         GLOBAL_UNIT_ASSIGNED_CONTEXT((#15,#16))
         REPRESENTATION_CONTEXT('',''));
#20= PRODUCT_DEFINITION_SHAPE('',$,#1);
#21= ID_ATTRIBUTE('',#20);
#22= SHAPE_DEFINITION_REPRESENTATION(#20,#12);
#23= DESCRIPTION_ATTRIBUTE('',#22);
#24= NAME_ATTRIBUTE('',#22);
#30= CURVE('');
#31= CURVE('');
#86= EDGE_CURVE('P#86',#30,.T.);
#88= EDGE_CURVE('P#88',#31,.T.);
#101= DATA_QUALITY_DEFINITION(
         'Short edge is detected with the accuracy 0.00001.');
#102= PRODUCT_DATA_AND_DATA_QUALITY_RELATIONSHIP('',#1,#101);
#103= DATA_QUALITY_DEFINITION_REPRESENTATION_RELATIONSHIP('',#101,#108);
#104= (LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT(.MILLI.,.METRE.));
#105= (NAMED_UNIT(*)PLANE_ANGLE_UNIT()SI_UNIT($,.RADIAN.));
#107= ID_ATTRIBUTE('',#108);
#108= SHAPE_CRITERIA_REPRESENTATION_WITH_ACCURACY('',(#114,#118,#123,#128),
         #110,(#121));
#109= DESCRIPTION_ATTRIBUTE('',#108);
#110= (GEOMETRIC_REPRESENTATION_CONTEXT(3)
          GLOBAL_UNIT_ASSIGNED_CONTEXT((#104,#105))
          REPRESENTATION_CONTEXT('',''));
#114= SHORT_LENGTH_EDGE('',#117);
#115= SHAPE_SUMMARY_REQUEST_WITH_REPRESENTATIVE_VALUE('',#114,
         .FULL_STATISTICS.);
#116= DETAILED_REPORT_REQUEST('',#114,.INFERIOR_QUALITY_ELEMENT.,
```

```
        .EXTREMITY_ORDER.);
#117= SHAPE_DATA_QUALITY_ASSESSMENT_BY_NUMERICAL_TEST('threshold:0.01mm',
        #118);
#118= (LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
          MEASURE_WITH_UNIT(LENGTH_MEASURE(0.01),#104)
          QUALIFIED_REPRESENTATION_ITEM((#119))
          REPRESENTATION_ITEM('upper limit')
          SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT()
          SHAPE_DATA_QUALITY_VALUE_LIMIT());
#119= TYPE_QUALIFIER('maximum');
#121= SHAPE_MEASUREMENT_ACCURACY('General length accuracy 0.001mm',#123);
#123= (LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
          MEASURE_WITH_UNIT(LENGTH_MEASURE(0.001),#104)
          QUALIFIED_REPRESENTATION_ITEM((#124))
          REPRESENTATION_ITEM('upper limit')
          SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT()
          SHAPE_DATA_QUALITY_VALUE_LIMIT());
#124= TYPE_QUALIFIER('maximum');
#126= SHAPE_DATA_QUALITY_CRITERION_AND_ACCURACY_ASSOCIATION(#127,#114);
#127= SHAPE_MEASUREMENT_ACCURACY('Specific length accuracy 0.00001mm',#128);
#128= (LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
          MEASURE_WITH_UNIT(LENGTH_MEASURE(1.0000000E-5),#104)
          QUALIFIED_REPRESENTATION_ITEM((#129))
          REPRESENTATION_ITEM('upper limit')
          SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT()
          SHAPE_DATA_QUALITY_VALUE_LIMIT());
#129= TYPE_QUALIFIER('maximum');
#131= SHAPE_INSPECTION_RESULT_REPRESENTATION_WITH_ACCURACY('sdqir-check1',
        (#136,#138,#145,#150,#153),#17,#108,(#143));
#132= DATA_QUALITY_DEFINITION_REPRESENTATION_RELATIONSHIP('',#101,#131);
#133= SHAPE_DATA_QUALITY_INSPECTED_SHAPE_AND_RESULT_RELATIONSHIP('',$,#12,
        #131);
#134= ID_ATTRIBUTE('',#131);
#135= DESCRIPTION_ATTRIBUTE('',#131);
#136= (DATA_QUALITY_INSPECTION_RESULT(#114)
         DATA_QUALITY_INSPECTION_RESULT_WITH_JUDGEMENT(.T.)
         REPRESENTATION_ITEM('')SHAPE_DATA_QUALITY_INSPECTION_RESULT());
#138= SHAPE_DATA_QUALITY_INSPECTION_CRITERION_REPORT('',#136,(#140,#141),
        LENGTH_MEASURE(0.009));
#140= DATA_QUALITY_INSPECTION_CRITERION_REPORT_ITEM('',24,
        .NUMBER_OF_INSPECTED_INSTANCES.);
#141= DATA_QUALITY_INSPECTION_CRITERION_REPORT_ITEM('',2,
        .NUMBER_OF_QUALITY_DEFECTS_DETECTED.);
#142= DATA_QUALITY_REPORT_MEASUREMENT_ASSOCIATION('','',#114,#138);
#143= SHAPE_MEASUREMENT_ACCURACY('General length accuracy 0.001mm',#145);
#145= (LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
          MEASURE_WITH_UNIT(LENGTH_MEASURE(0.001),#15)
          QUALIFIED_REPRESENTATION_ITEM((#146))
```

```
            REPRESENTATION_ITEM('upper limit')
            SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT()
            SHAPE_DATA_QUALITY_VALUE_LIMIT());
#146= TYPE_QUALIFIER('maximum');
#148= SHAPE_INSPECTION_RESULT_ACCURACY_ASSOCIATION(#149,#136);
#149= SHAPE_MEASUREMENT_ACCURACY('Specific length accuracy 0.00001mm',#150);
#150= (LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
            MEASURE_WITH_UNIT(LENGTH_MEASURE(1.0000000E-5),#15)
            QUALIFIED_REPRESENTATION_ITEM((#151))
            REPRESENTATION_ITEM('upper limit')
            SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT()
            SHAPE_DATA_QUALITY_VALUE_LIMIT());
#151= TYPE_QUALIFIER('maximum');
#153= SHAPE_DATA_QUALITY_INSPECTION_INSTANCE_REPORT('',#136,(#155,#158));
#154= DATA_QUALITY_REPORT_MEASUREMENT_ASSOCIATION('',$,#114,#153);
#155= SHAPE_DATA_QUALITY_INSPECTION_INSTANCE_REPORT_ITEM('',(#86),
          LENGTH_MEASURE(0.009));
#158= SHAPE_DATA_QUALITY_INSPECTION_INSTANCE_REPORT_ITEM('',(#88),
          LENGTH_MEASURE(0.009));
ENDSEC;
```

H.3 Instances for gap_between_edge_and_base_surface

The next example concerns with the **gap_between_edge_and_base_surface** criterion. If this criterion is assigned as a concern, PDQ checking system shall inspect every **edge_curve**s which bounds a **face_surface** of the target shape model, calculate the maximum value of the minimum distance from any point on the **edge_curve**s under inspection to the underlying surface, and detect all the **edge_curve**s with the calculated value larger than the specified threshold. The definition of the entity **gap_between_edge_and_base_surface** is shown in 7.4.99.

For requirement/declaration of quality and for assurance of quality, examples are omitted, as they are similar to the previous example for the criterion **short_length_edge**. Example instances are shown only for the scenario of data quality information to be used in quality improvement.

An example for this scenario presented below is based on the following assumptions;
- The requirement is to detect gaps larger than or equal to 0.01mm with the accuracy of 10^{-5}mm.
- The requirement on report is: 1) to identify faces that has a gap larger than or equal to 0.01mm between its bounding edges and the underlying surface, 2) to identify pairs of the point on the edge and the point on the base surface where distance between them is within the given threshold, and 3) reports should be created in extremity order.
- The inspection is performed against the shape data of a product model. The ID of the product model is P#319 and the ID of the shape representation is P#304.
- The summary report has the following contents as the result of inspection: 66 face_surfaces are inspected and one face_surface is found to have a gap within the given threshold. The length of the gap is 0.013mm.
- The detailed information on the face_surface that is detected to have a large gap between its bounding edge_curves is: The gap is detected on the face_surface P#43 between point_on_edge_curve Q# 157 and point_on_face_surface Q#155 and its value is 0.013mm.

NOTE Instances circumscribed with broken lines are created with this part of standard. ID for these instances is shown with prefix ‘Q’ such as ‘point_on_face_surface Q#21’.

The instances to be created for this scenario are shown in Figures H.21, H.22, H.23, H.24, H.25, H.26 and H.27.

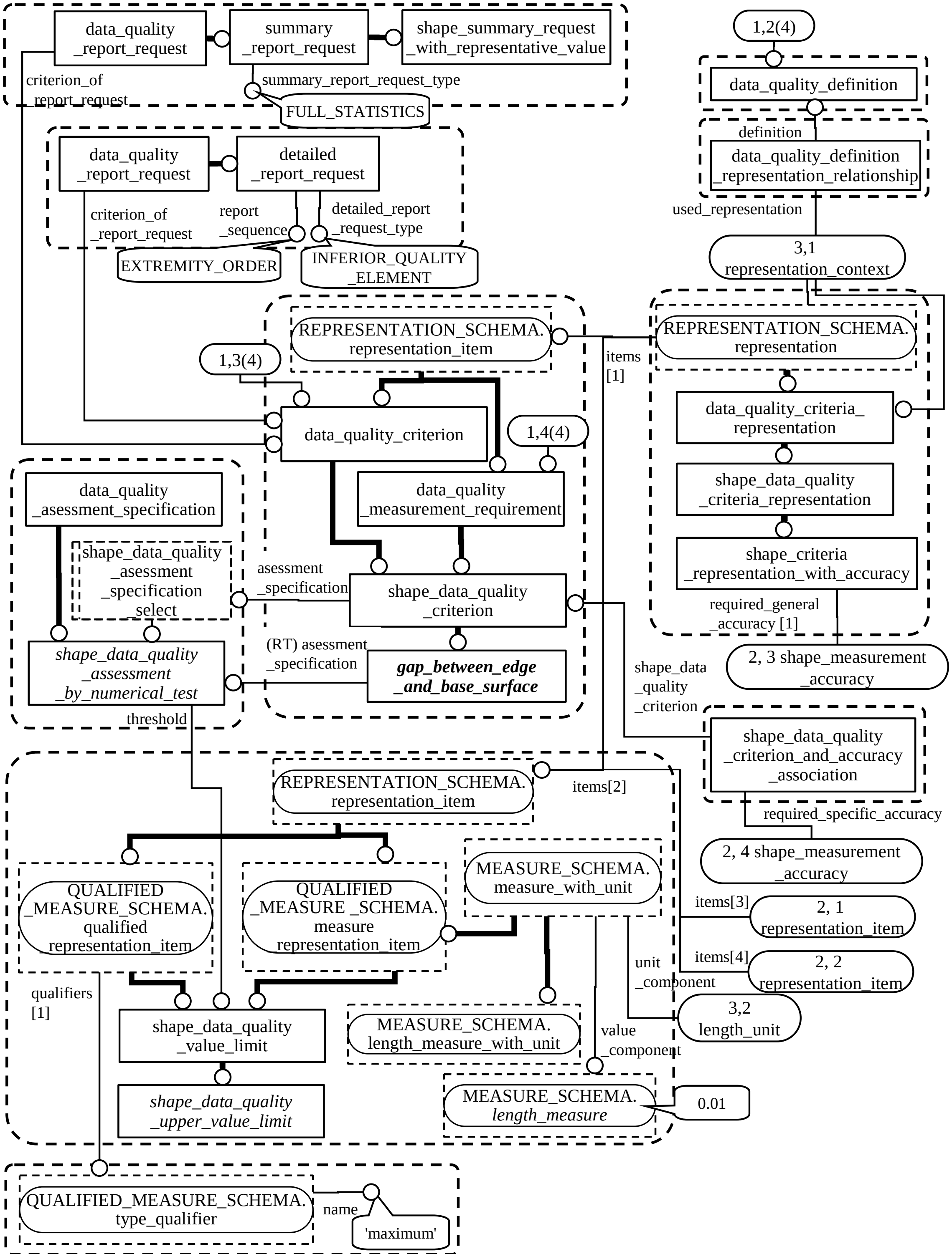

Figure H.21 - Example instances of quality information for use in the improvement of quality gap_between_edge_and_base_surface (1 of 7)

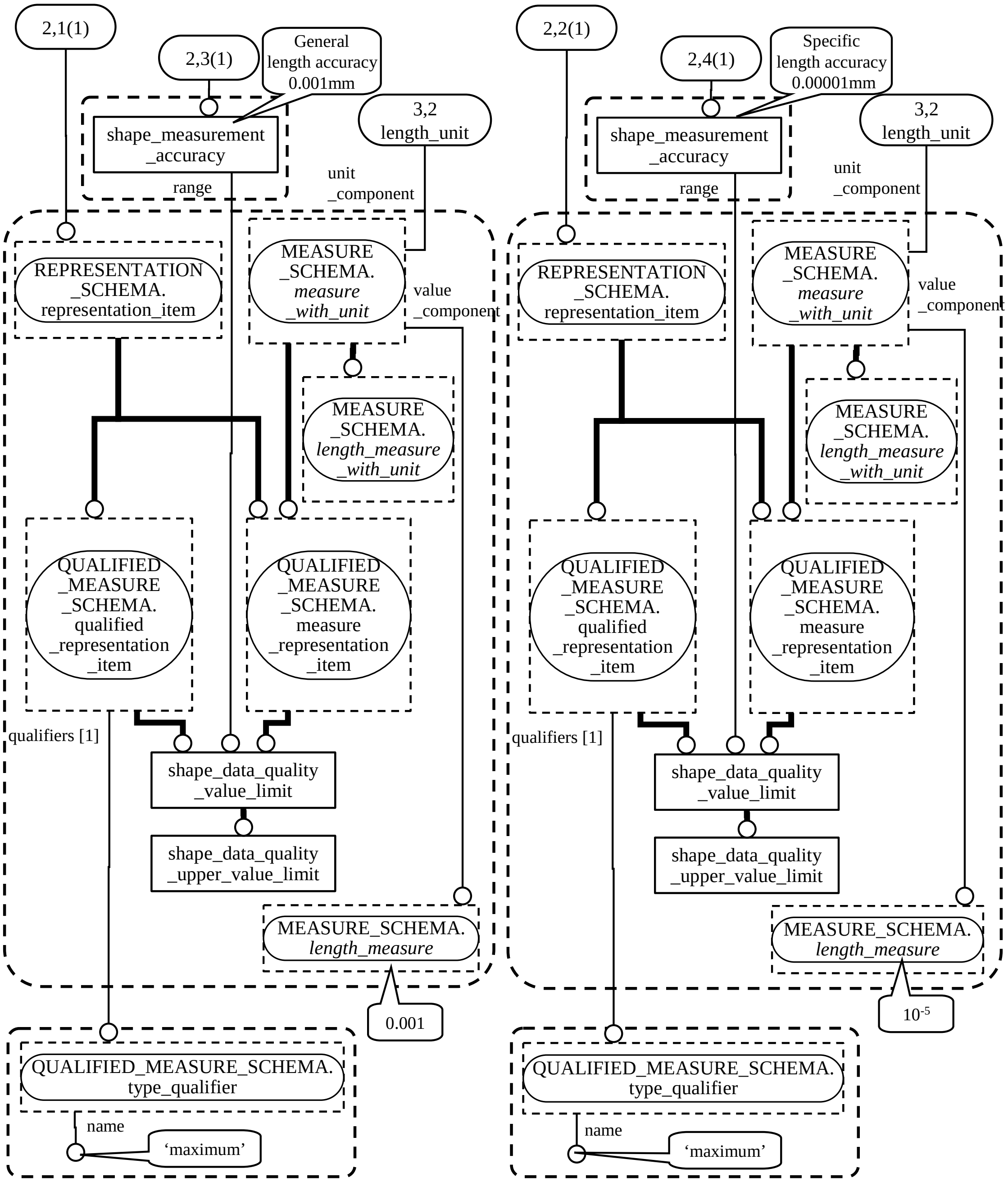

Figure H.22 - Example instances of quality information for use in the improvement of quality gap_between_edge_and_base_surface (2 of 7)

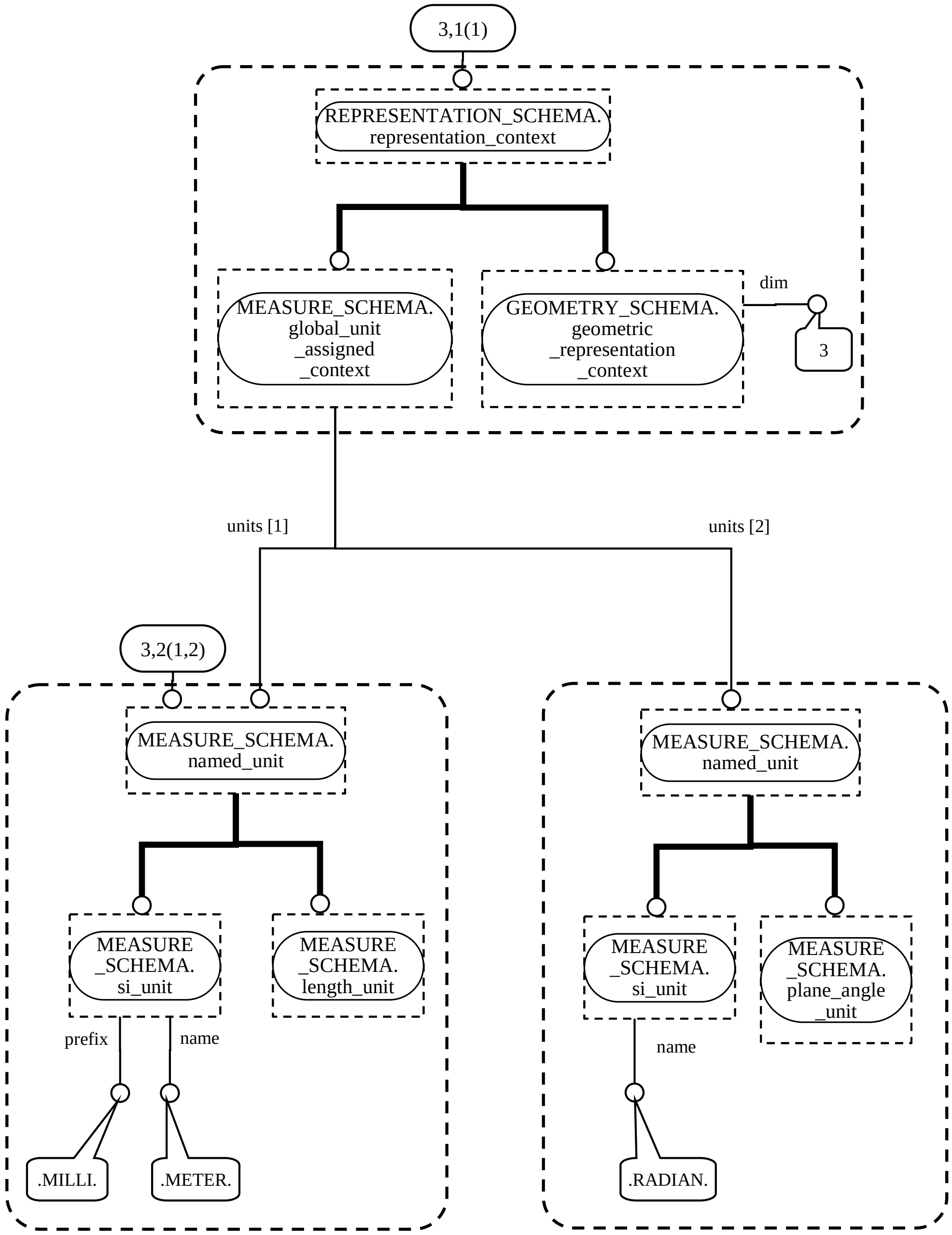

Figure H.23 - Example instances of quality information for use in the improvement of quality gap_between_edge_and_base_surface (3 of 7)

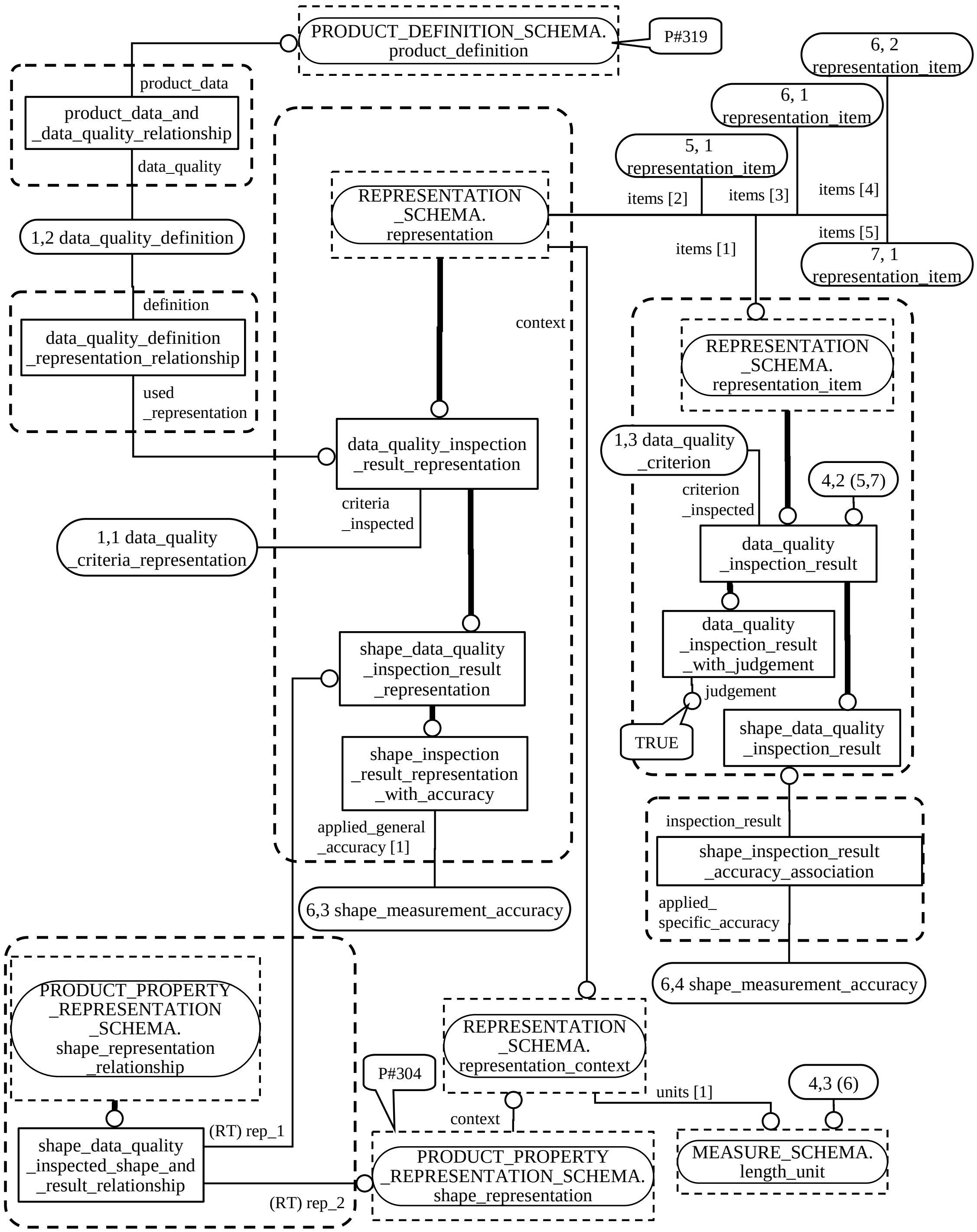

Figure H.24 - Example instances of quality information for use in the improvement of quality gap_between_edge_and_base_surface (4 of 7)

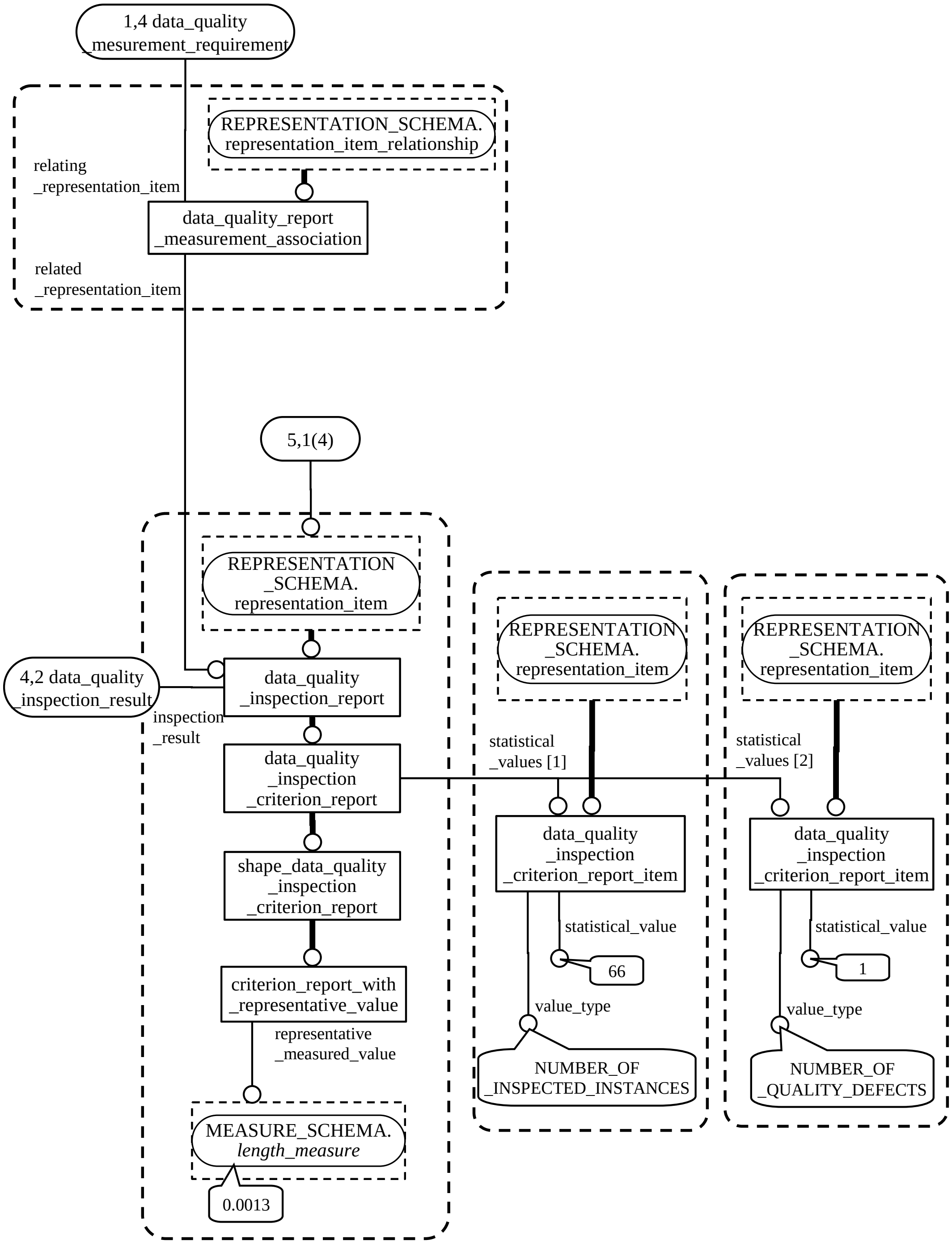

Figure H.25 - Example instances of quality information for use in the improvement of quality gap_between_edge_and_base_surface (5 of 7)

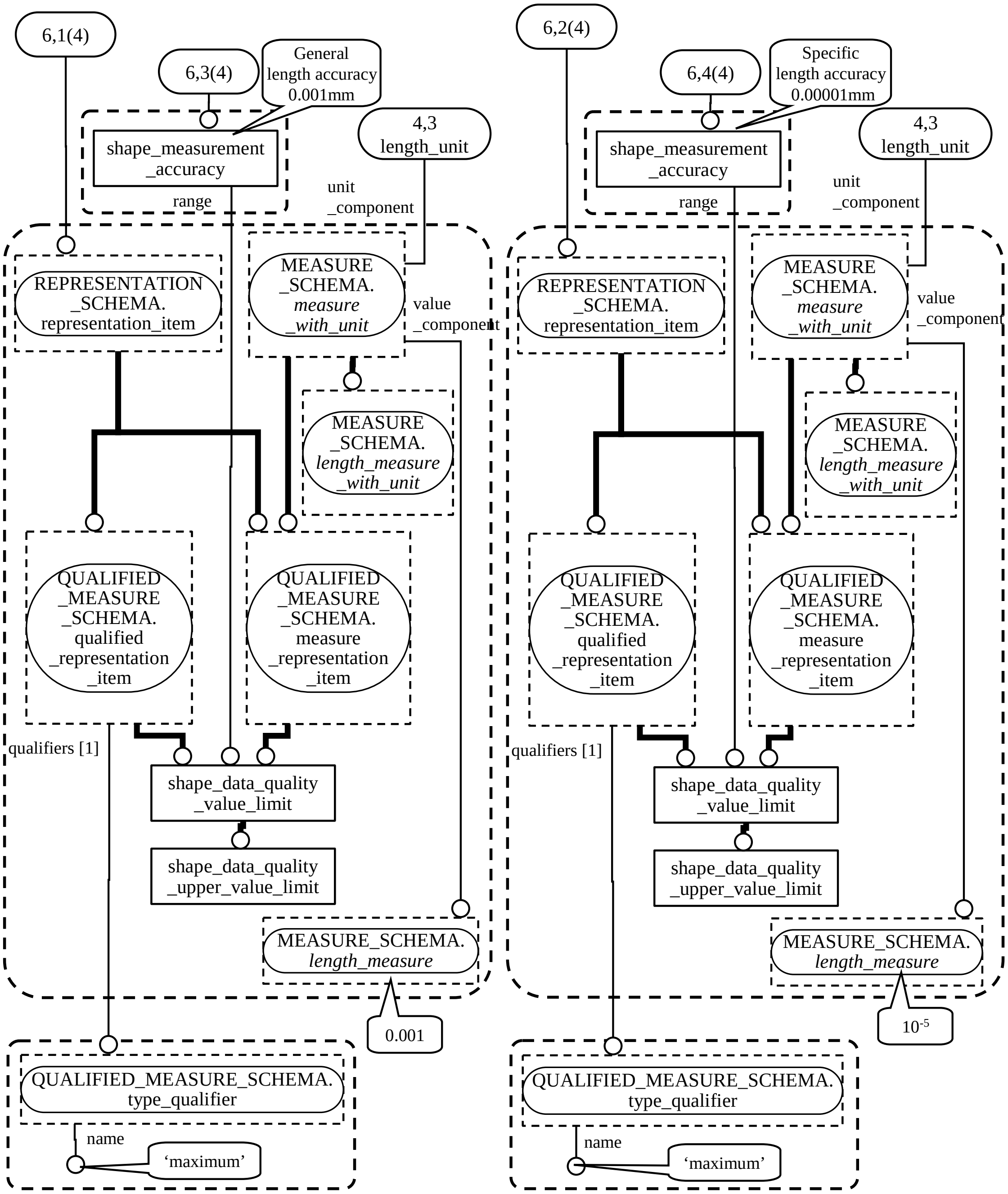

Figure H.26 - Example instances of quality information for use in the improvement of quality gap_between_edge_and_base_surface (6 of 7)

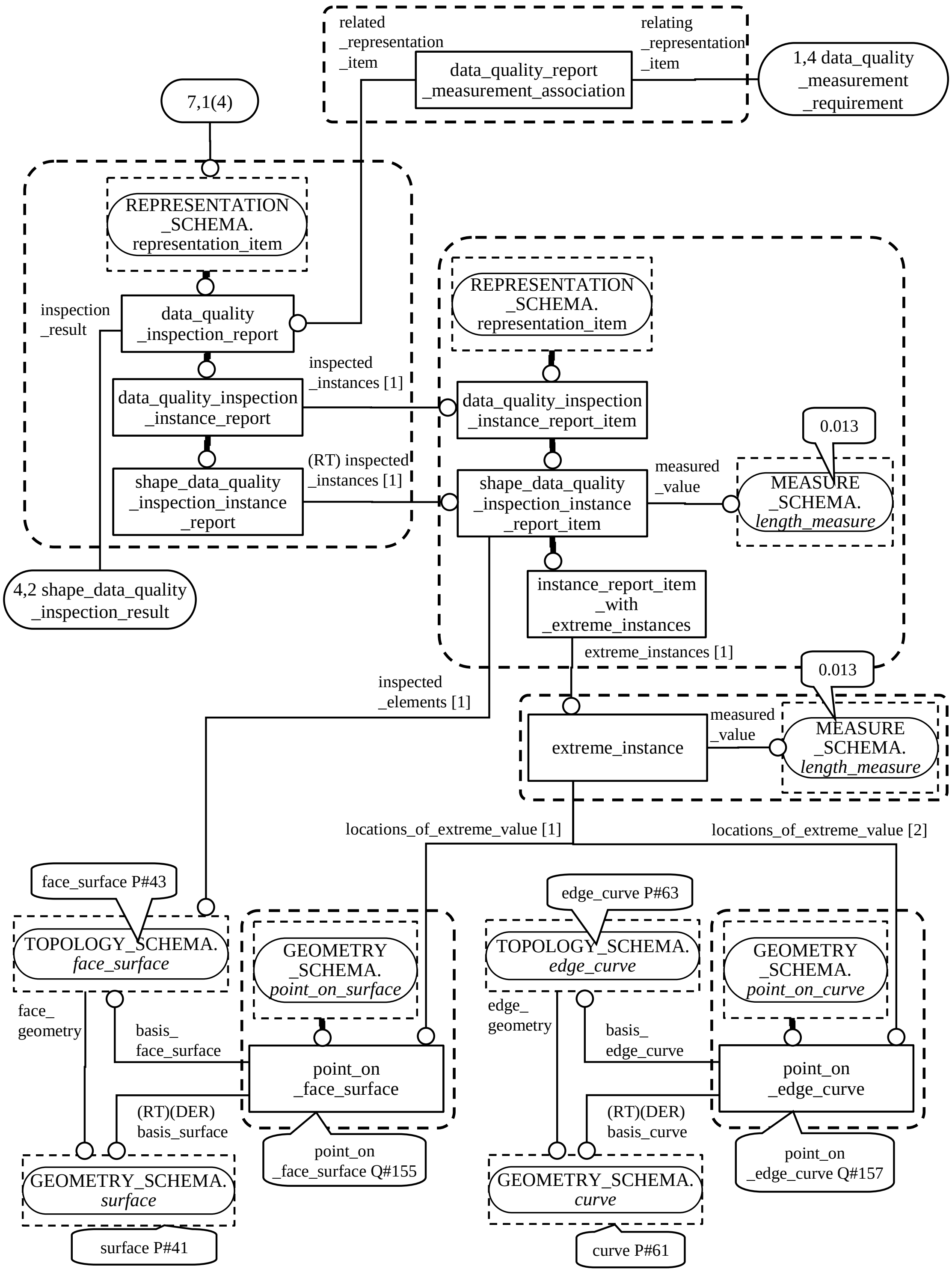

Figure H.27 - Example instances of quality information for use in the improvement of quality gap_between_edge_and_base_surface (7 of 7)

The data for this example described with the format defined in ISO 10303-21 is as follows.

```
DATA;
#1= PRODUCT_DEFINITION('target product data',$,#3,#11);
#2= NAME_ATTRIBUTE('P#319',#1);
#3= PRODUCT_DEFINITION_FORMATION('target data1',$,#4);
#4= PRODUCT('','',$,(#6));
#6= PRODUCT_CONTEXT('',#7,'mechanical');
#7= APPLICATION_CONTEXT(
      'configuration  controlled  3D  designs  of  mechanical  parts  and
assemblies');
#9= ID_ATTRIBUTE('',#7);
#10= DESCRIPTION_ATTRIBUTE('',#7);
#11= PRODUCT_DEFINITION_CONTEXT('',#7,'design');
#12= SHAPE_REPRESENTATION('target shape_representation',(#43,#63),#17);
#13= ID_ATTRIBUTE('P#304',#12);
#14= DESCRIPTION_ATTRIBUTE('',#12);
#15= (LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT(.MILLI.,.METRE.));
#16= (NAMED_UNIT(*)PLANE_ANGLE_UNIT()SI_UNIT($,.RADIAN.));
#17= (GEOMETRIC_REPRESENTATION_CONTEXT(3)
         GLOBAL_UNIT_ASSIGNED_CONTEXT((#15,#16))
         REPRESENTATION_CONTEXT('',''));
#20= PRODUCT_DEFINITION_SHAPE('',$,#1);
#21= ID_ATTRIBUTE('',#20);
#22= SHAPE_DEFINITION_REPRESENTATION(#20,#12);
#23= DESCRIPTION_ATTRIBUTE('',#22);
#24= NAME_ATTRIBUTE('',#22);
#41= SURFACE('P#41');
#43= FACE_SURFACE('P#43',#41,.T.);
#61= CURVE('P#61');
#63= EDGE_CURVE('P#63',#61,.T.);
#101= DATA_QUALITY_DEFINITION('');
#102= PRODUCT_DATA_AND_DATA_QUALITY_RELATIONSHIP('',#1,#101);
#103= DATA_QUALITY_DEFINITION_REPRESENTATION_RELATIONSHIP('',#101,#108);
#104= (LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT(.MILLI.,.METRE.));
#105= (NAMED_UNIT(*)PLANE_ANGLE_UNIT()SI_UNIT($,.RADIAN.));
#107= ID_ATTRIBUTE('',#108);
#108= SHAPE_CRITERIA_REPRESENTATION_WITH_ACCURACY('sdqc-check1',
         (#113,#118,#123,#128),#110,(#121));
#109= DESCRIPTION_ATTRIBUTE('',#108);
#110= (GEOMETRIC_REPRESENTATION_CONTEXT(3)
           GLOBAL_UNIT_ASSIGNED_CONTEXT((#104,#105))
           REPRESENTATION_CONTEXT('',''));
#113= GAP_BETWEEN_EDGE_AND_BASE_SURFACE('',#117);
#115= SUMMARY_REPORT_REQUEST('',#113,.FULL_STATISTICS.);
#116= DETAILED_REPORT_REQUEST('',#113,.INFERIOR_QUALITY_ELEMENT.,
         .EXTREMITY_ORDER.);
#117= SHAPE_DATA_QUALITY_ASSESSMENT_BY_NUMERICAL_TEST('threshold:0.01mm',
```

```
         #118);
#118= (LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
           MEASURE_WITH_UNIT(LENGTH_MEASURE(0.01),#104)
           QUALIFIED_REPRESENTATION_ITEM((#119))
           REPRESENTATION_ITEM('lower limit')
           SHAPE_DATA_QUALITY_LOWER_VALUE_LIMIT()
           SHAPE_DATA_QUALITY_VALUE_LIMIT());
#119= TYPE_QUALIFIER('minimum');
#121= SHAPE_MEASUREMENT_ACCURACY('General length accuracy 0.001mm',#123);
#123= (LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
           MEASURE_WITH_UNIT(LENGTH_MEASURE(0.001),#104)
           QUALIFIED_REPRESENTATION_ITEM((#124))
           REPRESENTATION_ITEM('upper limit')
           SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT()
           SHAPE_DATA_QUALITY_VALUE_LIMIT());
#124= TYPE_QUALIFIER('maximum');
#126= SHAPE_DATA_QUALITY_CRITERION_AND_ACCURACY_ASSOCIATION(#127,#113);
#127= SHAPE_MEASUREMENT_ACCURACY('Specific length accuracy 0.00001mm',#128);
#128= (LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
           MEASURE_WITH_UNIT(LENGTH_MEASURE(1.0000000E-5),#104)
           QUALIFIED_REPRESENTATION_ITEM((#129))
           REPRESENTATION_ITEM('upper limit')
           SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT()
           SHAPE_DATA_QUALITY_VALUE_LIMIT());
#129= TYPE_QUALIFIER('maximum');
#131= SHAPE_INSPECTION_RESULT_REPRESENTATION_WITH_ACCURACY('sdqir-check1',
         (#136,#138,#145,#150,#153,#160,#162),#17,#108,(#143));
#132= DATA_QUALITY_DEFINITION_REPRESENTATION_RELATIONSHIP('',#101,#131);
#133= SHAPE_DATA_QUALITY_INSPECTED_SHAPE_AND_RESULT_RELATIONSHIP('',$,#12,
         #131);
#134= ID_ATTRIBUTE('',#131);
#135= DESCRIPTION_ATTRIBUTE('',#131);
#136= (DATA_QUALITY_INSPECTION_RESULT(#113)
           DATA_QUALITY_INSPECTION_RESULT_WITH_JUDGEMENT(.T.)
           REPRESENTATION_ITEM('')SHAPE_DATA_QUALITY_INSPECTION_RESULT());
#138= SHAPE_DATA_QUALITY_INSPECTION_CRITERION_REPORT('',#136,(#140,#141),
         LENGTH_MEASURE(0.013));
#140= DATA_QUALITY_INSPECTION_CRITERION_REPORT_ITEM('',66,
         .NUMBER_OF_INSPECTED_INSTANCES.);
#141= DATA_QUALITY_INSPECTION_CRITERION_REPORT_ITEM('',1,
         .NUMBER_OF_QUALITY_DEFECTS_DETECTED.);
#142= DATA_QUALITY_REPORT_MEASUREMENT_ASSOCIATION('',$,#113,#138);
#143= SHAPE_MEASUREMENT_ACCURACY('General length accuracy 0.001mm',#145);
#145= (LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
          MEASURE_WITH_UNIT(LENGTH_MEASURE(0.001),#15)
          QUALIFIED_REPRESENTATION_ITEM((#146))
          REPRESENTATION_ITEM('upper limit')
          SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT()
```

```
          SHAPE_DATA_QUALITY_VALUE_LIMIT());
#146= TYPE_QUALIFIER('maximum');
#148= SHAPE_INSPECTION_RESULT_ACCURACY_ASSOCIATION(#149,#136);
#149= SHAPE_MEASUREMENT_ACCURACY('Specific length accuracy 0.00001mm',#150);
#150= (LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
          MEASURE_WITH_UNIT(LENGTH_MEASURE(1.0000000E-5),#15)
          QUALIFIED_REPRESENTATION_ITEM((#151))
          REPRESENTATION_ITEM('upper limit')
          SHAPE_DATA_QUALITY_UPPER_VALUE_LIMIT()
          SHAPE_DATA_QUALITY_VALUE_LIMIT());
#151= TYPE_QUALIFIER('maximum');
#153= SHAPE_DATA_QUALITY_INSPECTION_INSTANCE_REPORT('',#136,(#155));
#154= DATA_QUALITY_REPORT_MEASUREMENT_ASSOCIATION('',$,#113,#153);
#155= INSTANCE_REPORT_ITEM_WITH_EXTREME_INSTANCES('',(#43),
         LENGTH_MEASURE(0.013),(#158));
#158= EXTREME_INSTANCE((#160,#162),LENGTH_MEASURE(0.013));
#160= POINT_ON_FACE_SURFACE('Q#155',*,0.1,0.1,#43);
#162= POINT_ON_EDGE_CURVE('Q#157',*,0.1,#63);
ENDSEC;
```

Bibliography

[1] ISO 10303-203:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 203: Application protocol: Configuration controlled 3D designs of mechanical parts and assemblies*

[2] ISO 10303-214:2003, *Industrial automation systems and integration — Product data representation and exchange — Part 214: Application protocol: Core data for automotive mechanical design processes*

[3] ISO TC184/SC4 WG12 N5098, *Towards Data Quality Guaranteed Data Exchange and Sharing*

[4] A NIST report: 2002, *Economic Impact Assessment of the International Standard for the Exchange of Product Model Data (STEP) in Transportation Equipment Industries*

[5] ISO/PAS 26183:2006, *SASIG Product data quality guidelines for the global automotive industry*

[6] ISO 10303-21:2002, *Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure*

[7] ISO 31 (all parts), *Quantities and units*

Index

.

ICS 25.040.40

Price based on 306 pages