INTERNATIONAL STANDARD

ISO 16757-2

First edition
2016-11-15

Data structures for electronic product catalogues for building services —

Part 2: Geometry

Structures de données pour catalogues électroniques de produits pour les services du bâtiment —

Partie 2: Géométrie

Reference number
ISO 16757-2:2016(E)

Contents

Page

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

The committee responsible for this document is ISO/TC 59, *Buildings and civil engineering works*, Subcommittee SC 13, *Organization of information about construction works*.

A list of all the parts in the ISO 16757 series can be found on the ISO website.

Introduction

There is a growing need for electronic, machine-readable, digital information about building services. The designers in building services have to execute detailed calculations and simulations to ensure saving of energy and to satisfy hygienic and comfort criteria in heating, ventilation, air conditioning, and sanitary plants. Designers must have access to more complete and more accurate documentation to address these needs. The resulting designs have to describe the complete building services system without internal interference to avoid collision with other systems and components and the building structure.

These requirements can only be achieved with modern building services applications such as computer-aided design (CAD) and computer-aided engineering (CAE) systems, calculation programmes, BIM tools, and management software. The software systems need exact data of the used plant components because each component contributes to the performance of the whole building.

Thus, an international standard is required to provide the models and definitions for product catalogue data exchange.

Such a standard eliminates the need to manage different data formats or to use different manufacturer-specific software systems to deal with products of different manufacturers. The standard will lead to a significant reduction of costs for manufacturers and users. Integrating this data into building information modelling (BIM) systems allows data interchange between information technology (IT) systems. In addition to the benefits of planning, there will be further advantages for other software solutions, such as facility management and life-cycle management.

This part of ISO 16757 offers for the first time an interface which allows the uniform handling of data about technical, maintenance and service, as well as geometry, images, video and text information.

The objectives of this part of ISO 16757 are to facilitate

— automatic integration of catalogue data of all manufacturers in engineering applications such as CAD, CAE, dimensioning and calculation systems,

— uniform product selection across manufacturers,

— dimensioning of products using manufacturers' algorithms,

— possibility to recalculate and re-simulate the whole system with data of all building services components as often as required, and

— standardized representation of technical data for data exchange and life-cycle management.

This part of ISO 16757 specifically provides definitions and specifications for modelling and exchanging geometric information of building services components.

ISO 16757-1 gives the overview about the standard and the rationale for its elements and organization. This document defines the geometric elements which are used to represent the products in ISO 16757 catalogues. ISO 16757-3 defines the script language used in ISO 16757 (all parts) for various purposes. ISO 16757-4 contains IDM descriptions for ISO 16757 (all parts), including process descriptions for those processes which are to be supported by the standard and it comprises the rules for mapping of product and the property descriptions to IFC and for defining properties semantically with IFD. ISO 16757-5 defines an exchange format in XML by which electronic catalogues can be exchanged according to the definitions of ISO 16757 (all parts). The exchange format will be specified as an XML Schema Definition (XSD). The content parts of ISO 16757 will define standardized properties for the product groups and the composition of the technical data model. Furthermore, the content parts of ISO 16757 determine the specific programming function-interfaces to layout, calculate and simulate the products.

Data structures for electronic product catalogues for building services —

Part 2: **Geometry**

1 Scope

This part of ISO 16757 describes the modelling of building services product geometry. The description is optimized for the interchange of product catalogue data and includes

— shapes for representing the product itself,

— symbolic shapes for the visualization of the product's function in schematic diagrams,

— spaces for functional requirements,

— surfaces for visualization, and

— ports to represent connectivity between different objects.

The shape and space geometry is expressed as Constructive Solid Geometry (CSG) based on geometric primitives concatenated to boundary representations by Boolean operations. This part of ISO 16757 uses the applicable primitives from ISO 10303-42 and from ISO 16739 and adds primitives which are required for the special geometry of building services products. For symbolic shapes, line elements are also used.

This part of ISO 16757 neither describes the inner structure and internal functionality of the product nor the manufacturing information because this is typically not published within a product catalogue.

Building services products can have millions of variant dimensions. To avoid the exchange of millions of geometries, a parametric model is introduced which allows the derivation of variant-specific geometries from the generic model. This is necessary to reduce the data to be exchanged in a catalogue to a manageable size. The parametric model will result in smaller data files, which can be easier transmitted during data exchanges.

The geometry model used does not contain any drawing information such as views, line styles or hatching.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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 16757-1, *Data structures for electronic product catalogues for building services — Part 1: Concepts, architecture and model*

ISO 6707-1,*Buildings and civil engineering works — Vocabulary — Part 1: General Terms*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 16757-1 and ISO 6707-1 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— IEC Electropedia: available at http://www.electropedia.org/

— ISO Online browsing platform: available at http://www.iso.org/obp

3.1
product shape
geometric representation of the space defined by the product's external boundaries

3.2
product surface
coloured and textured outer boundary of the product's shape whose rendered appearance responds to relative lighting and viewing angles

3.3
port
located, oriented and directed feature of the product's geometry model (1) for connecting the product with other ports to transfer media or (2) to fasten the product to other products, accessories, walls, ceilings, floors, etc. or (3) for executing control

3.4
solid model
complete representation of the nominal shape of a product such that all points in the interior are connected and that any point can be classified as being inside, outside or on the boundary of a solid.

[SOURCE: ISO 10303-42:2014, 6.4.1]

3.5
parametrizable primitive solid
model of a defined primitive solid, e.g. a block, cylinder, sphere or cone whose dimensions are represented by parameters to generate variants

3.6
constructive solid geometry
CSG
type of geometric modelling in which a solid is defined as the result of a sequence of regularised Boolean operations operating on solid models

[SOURCE: ISO 10303-42]

3.7
clipping
operation applied to a geometric model to remove parts of the model beyond a defined boundary

4 Catalogue structure and catalogue information

All kinds of product data in the scope of ISO 16757 can be transmitted in a product catalogue data file.

The catalogue structure which is explained in more depth in ISO 16757-1 is depicted in Figure 1.

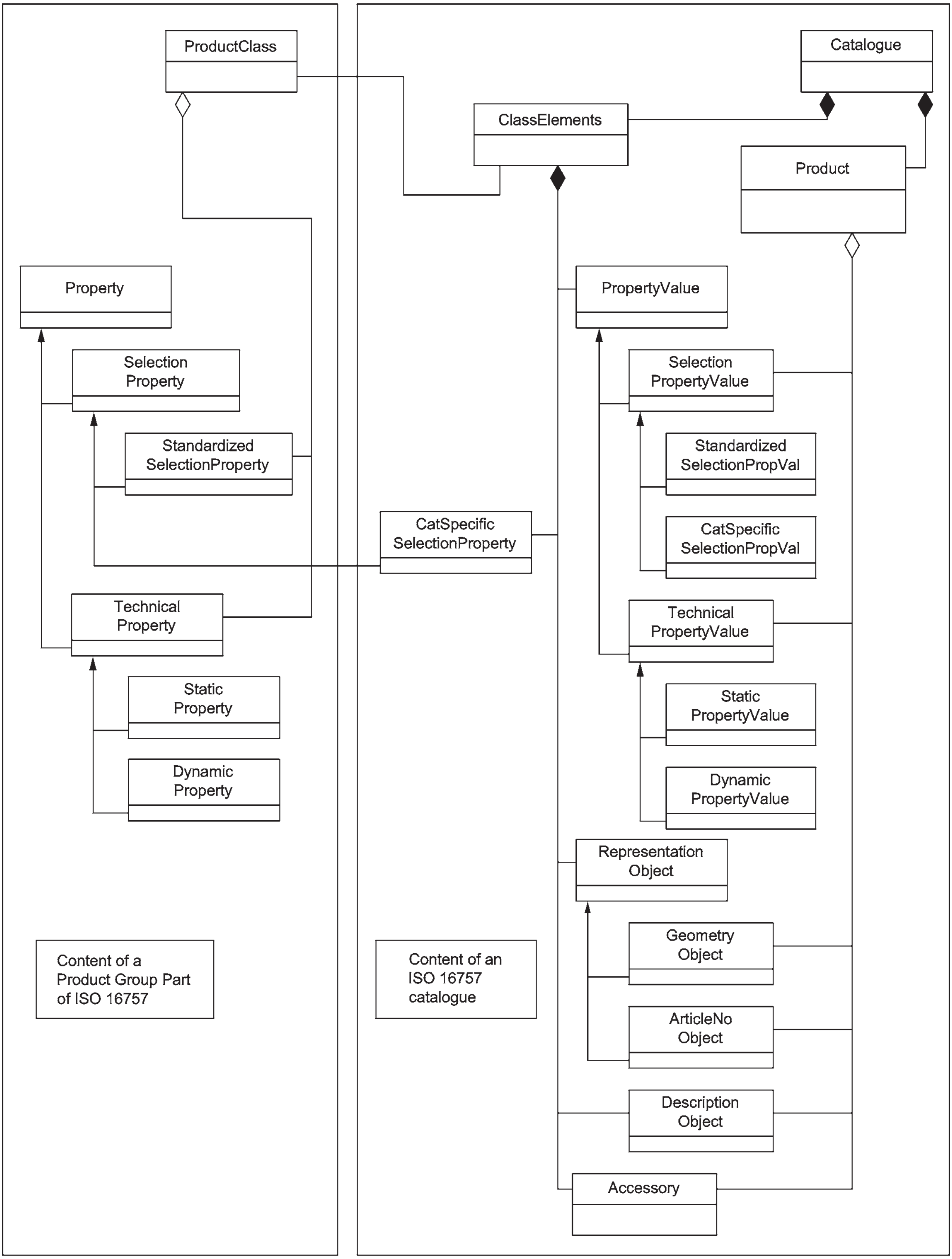

Figure 1 — Overview of the elements of a catalogue and the kinds of properties

5 Geometry

Geometry objects are representation objects in a catalogue. They can represent a product, an accessory or a part of one of them (see Figure 2).

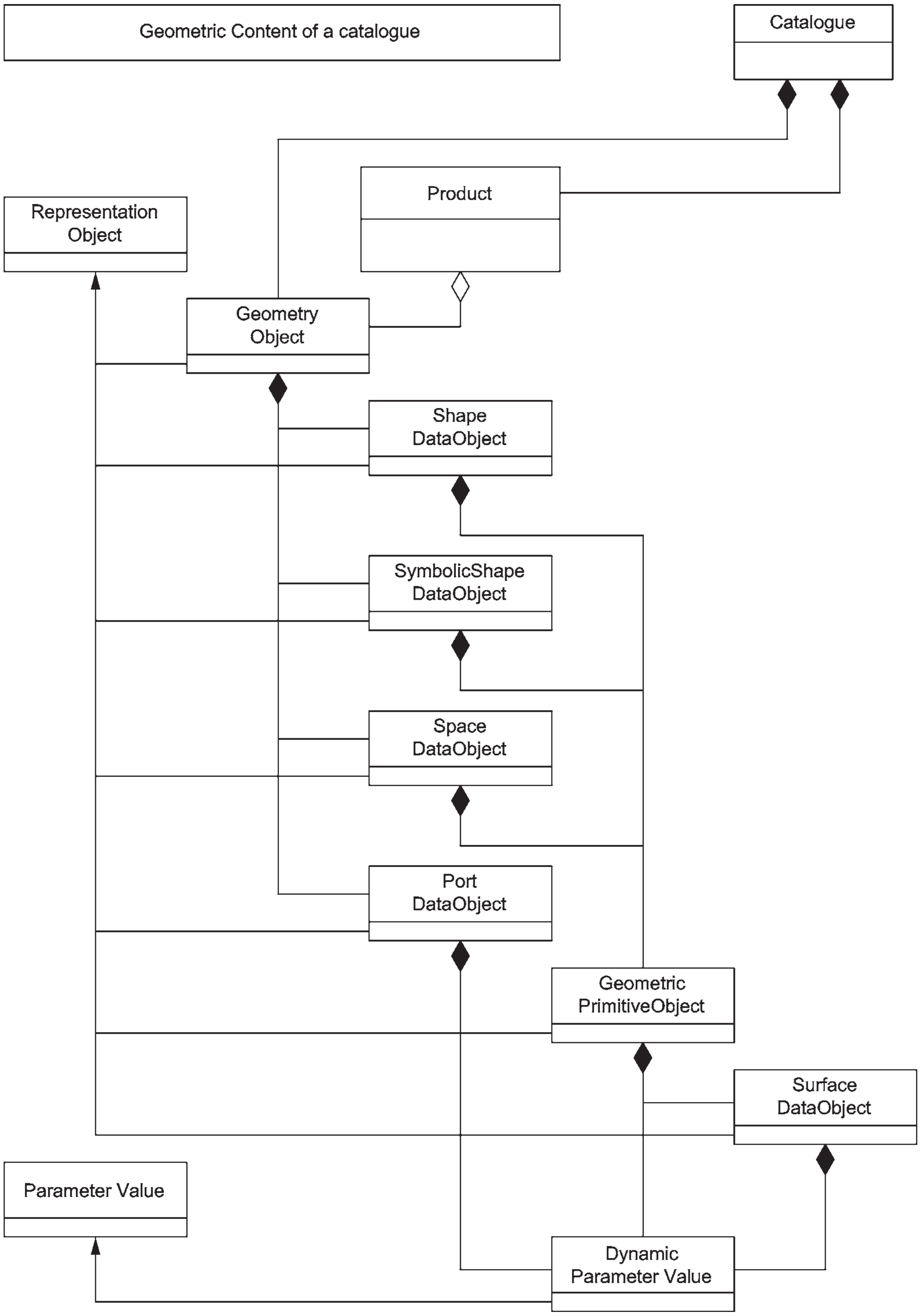

Figure 2 — Overview of the geometric elements of a catalogue and the kind of data objects

Geometry contains different kinds of geometric data:

— shapes;

— symbolic shapes;

— spaces;

— surfaces;

— ports.

The shapes, symbolic shapes and spaces are built by CSG trees. The leaves are geometric primitives (see Clause 7). For each primitive, a number of attributes have been defined which have to be supplied with concrete values to build the respective shapes. The inner nodes are CSG operators which also may have attributes that need to be fed by values. In the same way, attributes are specified for ports, and by filling in specific values for the attributes, specific ports are described.

To support the representation of a number of variants, each geometric representation is abstract, i.e. the attributes are not filled by fixed values for each product. Rather, the attribute values are described by formulas which use geometric properties as their parameters. These geometric properties are defined by the manufacturer, i.e. they are specific for the catalogue and may be different from catalogue to catalogue.

The geometric properties provide specific values for each product. They have to be computed for each product on the basis of the technical property values of that specific product variant. Thus, they are derived properties, and they are provided with a function which computes the actual value of the property for a given product variant (see ISO 16757-1). Some geometric properties may also be dynamic, i.e. they depend not only on the product properties, but also on conditions in the environment of the installed product.

A single product can consist of one or more components (see Figure 3). Each component of such a product shall be described as a separate geometric entity.

Figure 3 — Single product (heater with heat exchanger and water storage) as an assembly of components

5.1 Shapes

Shapes support the visualization of the product as a 3D geometry model (see Figure 4). In addition, they are required for interference checking with other shapes and spaces in the building model or the building services system model surrounding the building services product.

Figure 4 — Shape of a valve

5.2 Symbolic shapes

In addition to shapes, symbolic shapes are useful for the understanding of the model in visualizations and drawings. To illustrate, a 3D representation of a valve's shape cannot give information provided by a symbolic shape, e.g. a valve symbol will give additional information about the type of the valve, its function and form of activation.

The symbolic shape object also contains information about whether it is 2D or 3D.

The method for describing symbolic shapes is the same as that for shape data.

5.3 Space data

The description of the product's shape alone is not sufficient to check whether a product is correctly installed into the building services system. Many pieces of equipment need an operation space in front of their control or display panel, and additional space for installation and/or assembly (see Figure 5).

Spaces are categorized as follows.

5.3.1 Overall space

The space required for preliminary automatic interference checks by CAD systems, including all other spaces: the minimum operation space, the access space, the placement and transportation space and the installation space of the product.

5.3.2 Minimum operation space

The space needed by the product to function correctly, including spaces of opening doors, hatches, etc.

5.3.3 Access space

The space required by operators when maintaining and operating the product.

5.3.4 Placement and transportation space

The space needed by the largest single subassembly into which the product can be broken down to allow it to be moved in or out of the building to or from its place of installation.

5.3.5 Installation space

The space necessary for the onsite assembly and installation or de-installation of the product.

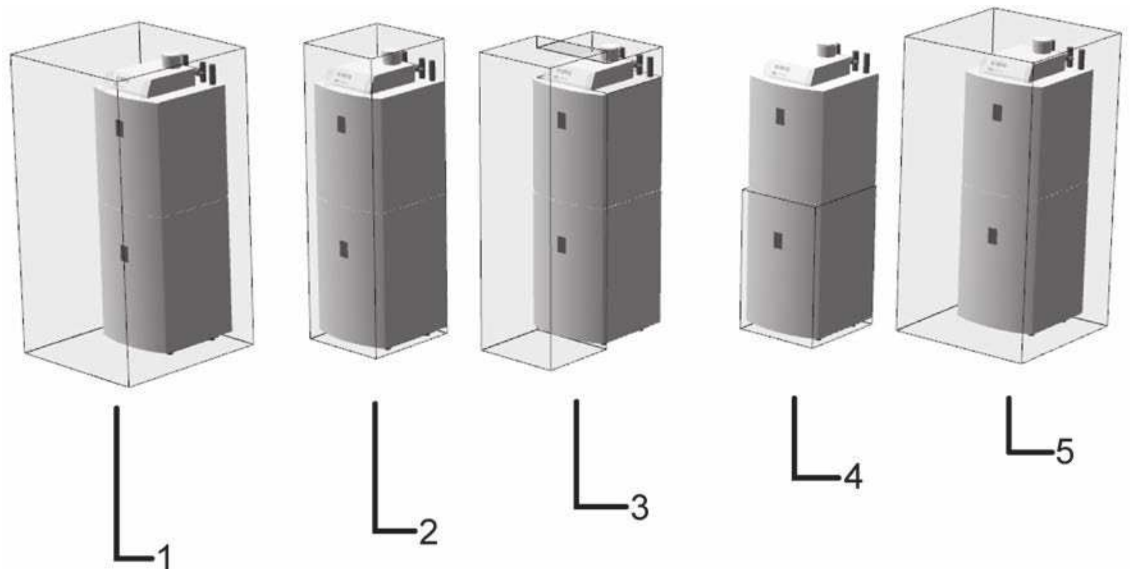

Figure 5 — Spaces of a product

The overall space is described by one single CSG primitive that represents an amalgamation of the product shape and all other spaces noted.

The other spaces can be assembled using one or more geometric primitives. They are configured in the same way as the shape itself.

5.4 Surfaces

Surfaces describe the colour and the texture of the product surface. Each different combination of colour and texture is listed once in the exchange data file and is referenced by geometric data.

5.5 Ports

Ports have to provide all the data which are necessary to identify product ports within a building services system model and to determine whether pairs of ports fit, or not.

Sufficiently described ports allow the automated installation of building services components in a system (e.g. automatic alignment) and for geometric checks to determine whether proper installation into a system is feasible.

The ports of a product can be categorized as

— media-carrying ports (carrying media (e.g. gas or liquid) to pipes, ducts, valves, fittings, etc.);

— fastening ports (means to fasten product to accessories, walls, ceilings, floors, etc.);

— control and monitoring signal ports.

If the software application analyses port information, it can automatically position the product relative to other system components, or offer alternative positions, e.g. placing a water heater on top, beside or behind a boiler. The same principle holds for fans and mounting frames, pumps and base frames, etc.

For this purpose, the ports have to be checked for functional and geometrical fitting.

6 Methodology of geometric description

6.1 Principle of geometric representation

The geometric representation of products in product catalogues comprises four main parts.

a) A combination of 3D solid primitives and the order to combine them by Boolean operations. This can be used to represent the product's shape, its symbolic geometry or the product's spaces.

   The positions and dimensions of the primitives can be constants, variables or mathematical rules using constants and variables. One combination of 3D solid primitives can be used for the geometric representation of a large range of product variants in a product series.

b) A definition of the product's surfaces to describe their visual appearance by allocation of a material definition.

c) A description of the product's ports to the building services system or other products, including their positions, directions and dimensions

d) A function ('get_geometry_values') or a set of functions which retrieve the property values of a product required for the calculation of its geometry. Together with the 3D-solid primitives, the surface values and the coordinate systems, these property values form the geometric representation of a single, identifiable product (see 6.5).

6.2 Level of detail

It is not unusual for a building to contain thousands of building services products. If they are all represented in great detail in a geometric building model, the data volume will increase dramatically.

Designers who use, for instance, thousands of radiators and radiator valves in one building model are not interested in a detailed view of the product. In drawings with large scales, a symbolic reference or less detailed visualization can be more informative than a detailed one.

A detailed visual impression of the product is often only required in certain instances. For example, when selecting a product, designers are usually very interested in its detailed geometry.

Levels of detail will be used in different documentations.

a) Schema drawing:

   1) Horizontal schema (e.g. for air condition flow plans):

      i) Pipes and ducts are represented by two parallel guided lines in 2D;

ii) Devices are represented by 2D symbols;

2) Vertical schema (e.g. for potable water plans, sewage plans and heating plans):

i) Pipes and ducts are represented by one guided line in 2D;

ii) Devices are represented by generic 2D symbols;

3) Isometric schema (e.g. for piping plans):

i) Pipes and ducts are represented by one guided line in 3D;

ii) Devices are represented by generic 3D symbols.

b) Spatial representation:

The spatial representation is very much dependent on the usage and the user.

EXAMPLE:

1) Building services system product manufacturers are interested in a nearly photo-realistic geometric representation which gives maximum information about the product.

2) Building services system designers are interested in a geometric representation which gives minimum information about the kind of the products for sizing, selecting, installing and operating.

3) Architects are interested in

— a generic representation of pipes and devices to realize room management, and

— a detailed representation of pipes and devices to get a visual impression of the visible parts of the building services systems (e.g. air outlets, radiators, visible pipes, visible ducts and other visible technical devices).

4) Owners, supervisors and general contractors are interested in a dynamic floating representation of pipes and devices, less detailed in an overview and more detailed while zoomed, for to check interoperability.

5) Users of receiving or interpreting applications are interested in a good performance of their software system.

To fulfil all these requirements, ISO 16757 provides for each building services product the following parallel levels of geometric details to design building services systems.

Level 1:

Less detailed symbolic shape geometry to design schemas as overview of building services systems. The symbol stands for the main function of the product. The symbol distinguishes, for example, a fire damper from a duct, a radiator from a heater, a bath tub from a sink and a valve from a pressure gauge. The geometry can contain four symbolic shapes to be used as a 2D top view, a 2D front view, a 2D side view or as 3D model.

Level 2:

High-detailed symbolic shape geometry to design schemas as overview of building services systems. This symbol stands for the explicit main function and sub functions of the product. The geometry can contain four symbolic shapes to be used as 2D top view, 2D front view, 2D side view or as 3D model.

Level 3:

Less detailed 3D shape geometry, elaborated so far, that the main classification into a product group is possible. It distinguishes, for example, a fire damper from a duct, a radiator from a heater, a bath tub from a sink and a valve from a pressure gauge.

The main target of this level is to provide maximum performance to 3D CAD systems. Therefore, the geometric shape of this level should be as simple as possible.

Level 4:

More detailed 3D shape geometry, elaborated so far, that a differentiation between products of different manufacturers is almost possible. The main geometric differences in the shape design may be displayed. The main target of this level is to divide different products while assuring acceptable performance of 3D CAD systems.

Level 5:

High-detailed 3D shape geometry, elaborated so far, that all main geometric properties of a product can be visible. This provides a nearly photorealistic view of the product without representing details of lower interest such as rivets or flat sheet metal seams. The main target of this level is to present particular products to end users or to create detailed visualizations in order to highlight particular instances of products in building services system models.

Ports will be defined differently in Levels 1 and 2 according to the specific requirements of the levels. The 3D spatial representation of ports in Levels 3, 4 and 5 are the same.

In contrast to the description of shapes, the 3D description of spaces is independent of levels of details.

Levels of details should not be confused with the level of development.[9]

Quote from BIMFORUM, *Level of Development Specification, Version 2013:*

“Level of Detail is essentially how much detail is included in the model element. Level of Development is the degree to which the element’s geometry and attached information has been thought through – the degree to which project team members may rely on the information when using the model. In essence, Level of Detail can be thought of as input to the element, while Level of Development is reliable output.”

6.3 Surfaces

The product’s surface is assembled of the surfaces of its components. In the same way, the component’s surface is composed of the surfaces of their primitives. Each primitive has at least one or more sub-area surfaces surrounded by edges.

In most cases, all sub-area surfaces of a primitive have the same colour and texture. So, the whole primitive derives its basic colour or surface texture from a single surface definition.

In some cases, different colours and textures have to be assigned to sub-areas of the primitive. It happens, that these sub-areas do not exist at the primitives but are later generated by the Boolean operations.

Therefore, in the following, a method for surface allocation is defined:

> By entering a 3D-coordinate point in an area to be coloured, a specific sub-area of the primitive’s surface is selected for individual colouring or texturing. The receiving or interpreting application compares this point with all surface-parts that have resulted from the block generation (see Figure 6). The point is mapped (projected) onto each area. The surface area to be coloured is the area with the shortest projection length where the projected point lies within the area boundary.

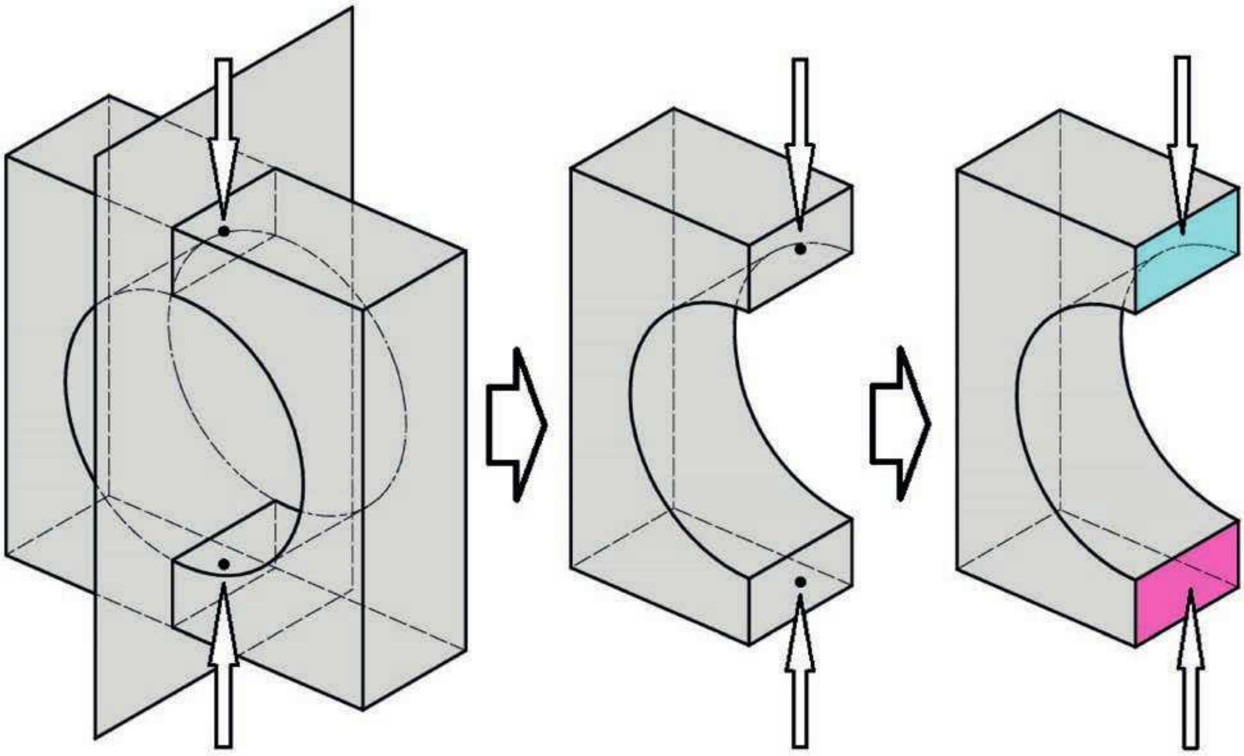

Figure 6 — Example — Defining the areas to be coloured or textured by entering 3D-coordinate points inside the prospective surfaces

The texture start point shall lie in the definition plane, definition cylinder, definition cone, definition sphere or definition toroid of the surface area to be coloured (see Figure 7). The u and v coordinate axes span the texture image to be applied, which is mapped onto the surface area in accordance with the scaling factors, i.e. either magnified or reduced. The texture pattern will be repeated in both axe directions as often as necessary. The whole texture will be clipped at the outer boundary line elements of the area.

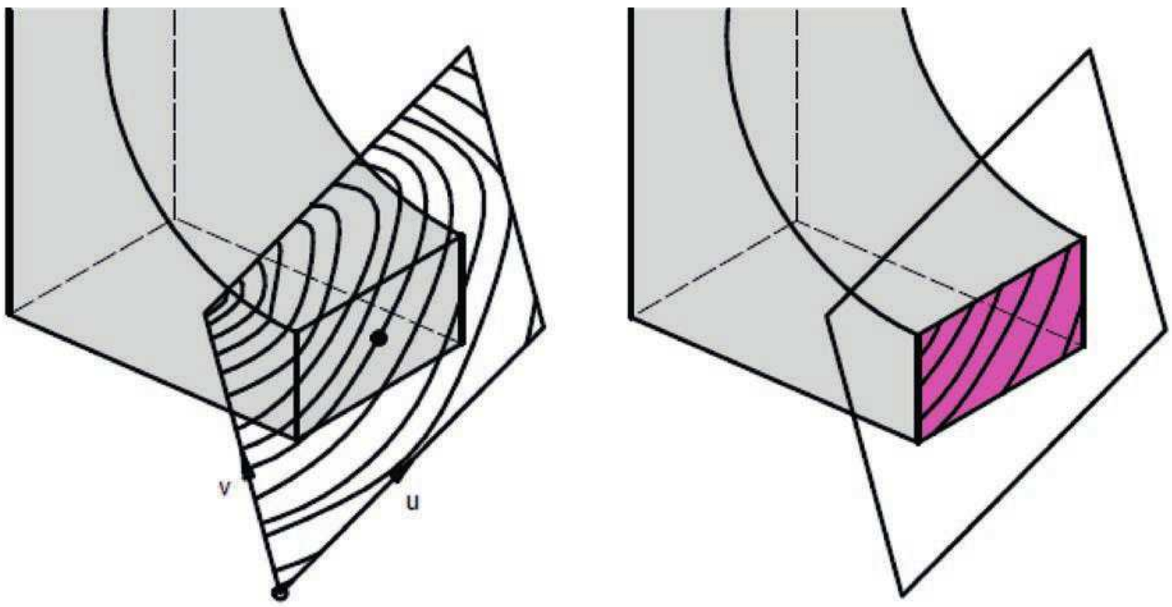

Figure 7 — Example — Defining the texture start point, direction and scaling

6.4 Ports

Ports are described by the following attributes:

— Port-ID

This numerical value identifies the port within the product.

EXAMPLE 1, 2, 3...

— Function

A list of alphanumerical codes identifies the use of the port for the product.

EXAMPLE "HFL" for heating flow and "HRT" for heating return

— Medium options

A list of alphanumerical codes defines the possible medium types at the port.

EXAMPLE "PWC" for cold potable water, "SEW" for sewage

— Medium flow direction

An alphanumerical value, sometimes dependent on carried medium, defines the right way to insert a product into a building services system. The values are

— "IN" for incoming medium,

— "OUT" for outgoing medium,

— "INOUT" for changing directions, and

— "NO" for no medium flow (fastening and control ports).

— Position

The port's position is given by a placement coordinate system.

EXAMPLE X = 123, Y = 456, Z = 789

— Direction

The x-vector of the placement coordinate system gives the port's direction.

EXAMPLE XXL = 0.86602540378, XYL = 0.5, XZL = 0 (for a direction with a 30° angle in the x-y-plane)

— Orientation

The y-vector of the placement coordinate system defines the orientation of the port. It defines the turn angle of the port geometry around the direction axis.

EXAMPLE YXL = -0.5, YYL = 0.86602540378, YZL = 0 (for an orientation with a 120° angle in the x-y-plane)

— Form

An alphanumerical code for the port itself and a list of forms of the possible counter-ports define the form to connect the port to others.

EXAMPLE "Nipple" and "Sleeve" or "Flange" and "Flange;" "Intermediate Flange"

— Method

An alphanumerical code which shall be equal at two fitting ports defines the method to connect the port to others. The code references lists of port methods in external definition dictionaries.

EXAMPLE ISO 128

— Dimensions

Alphanumerical values define the dimensions of the port itself and a list of alphanumerical values defines the possible port dimensions. Their number and values are dependent on the port's method.

EXAMPLE "DN 32" or "12 × 60"

The port function defines the role of the port either as a medium transfer point or as a fastening port. It can also determine the range of allowed media. Media flow direction, position, method and dimensions of the port serve the automated installation of building services components in a system (e.g. automatic alignment) and geometric checks to determine whether proper installation into a system is feasible.

The visible **port shape** (e.g. thread, hexagonals, flanges and frames) is **not** specified by the port itself. It shall be defined in the product shape and is dependant of Levels of Detail.

For ports, the "Function", "Medium options", "Form" and "Dimensions" are alphanumeric values. Unfortunately, there does not exist a standard with enumeration lists or other predefined sets of values for these kinds of attributes. Currently, they have to be negotiated between exchange partners or markets. The standardization of these value sets is out of the scope of this part of ISO 16757-2.

6.5 Generation of parameter values for the geometry

The parameterized positions and dimensions of ports and shape primitives shall be set up in a basic geometry model. Terms containing parameters and constant values define internal dependencies of dimensions and positions.

Figure 8 shows the dimensional dependencies in a small example. All dimensions are dependent on the parameters "rad" and "len." The mathematical terms define the rule that the circular hole will never touch or pass over the surrounding block. In the same way, it is ensured that the length of the cylindrical hole will fit the length of the block.

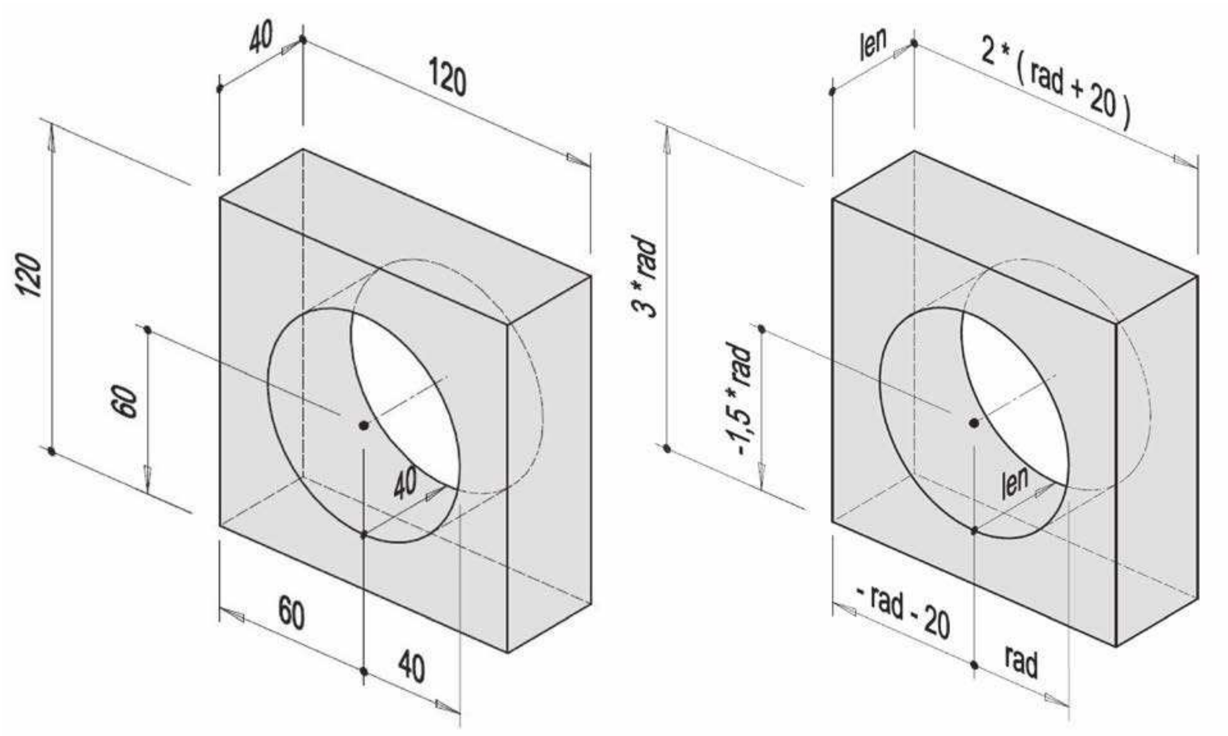

Figure 8 — Example — Parametrization of a geometric model

In this way, several similar variants of a geometric model (see Figure 9) can be built by changing the values of the parameters (“len” and “rad” in the example).

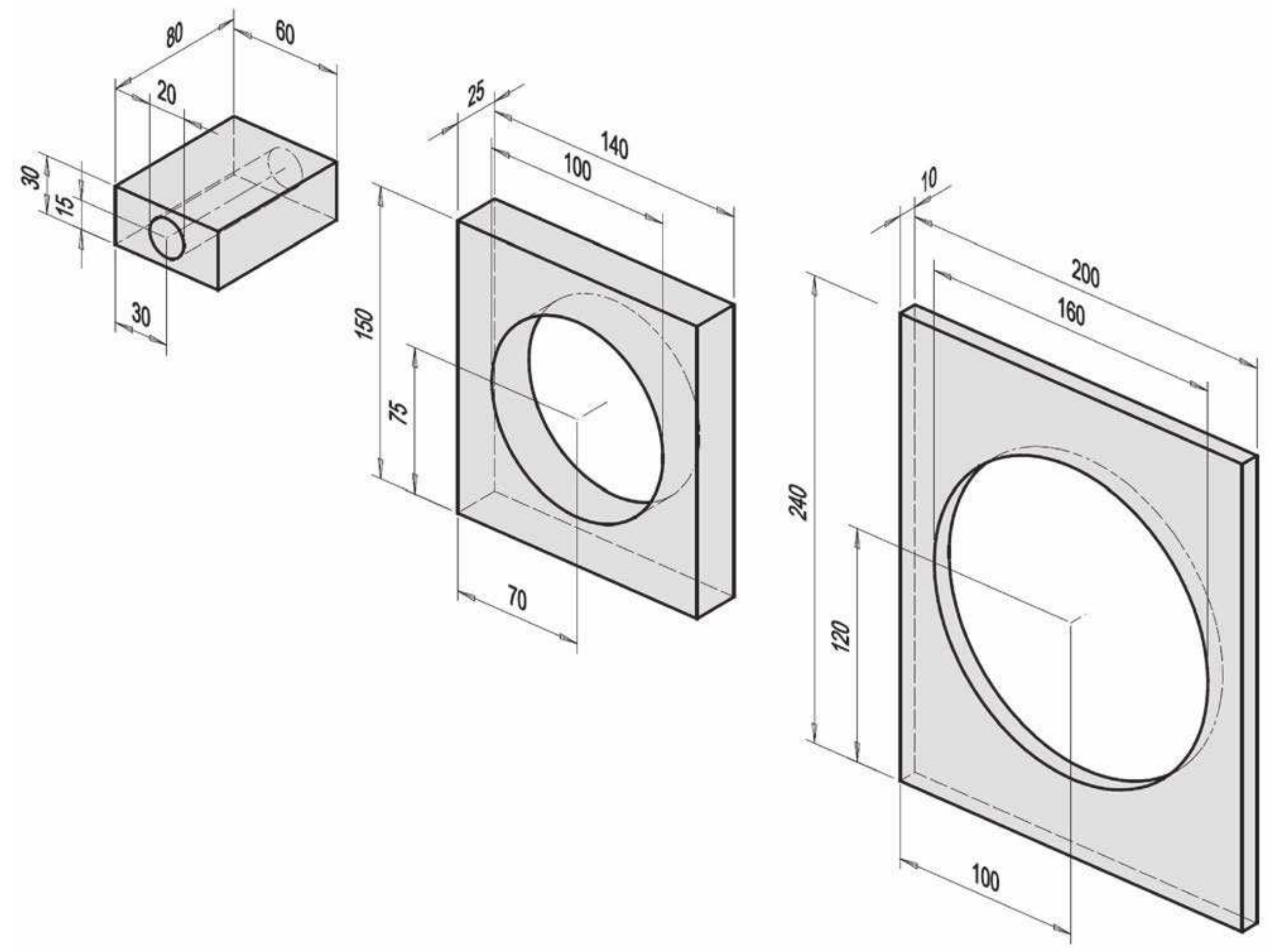

Figure 9 — Example — Variants of a geometric model

The values of the parameters are defined by the available product variants. Product catalogues normally provide charts of main dimension values to describe the shape and the positions of ports. The different values in a product series cannot necessarily be described by mathematical rules. Some dimensions are constant in parts of the series, some are dependent on main dimensions and some are dependent on accessories or the manufacturing process.

These dependencies shall be described in a function “get_geometry_values,” which delivers the values of all parameters used in the geometric model. Knowing the selected variant from the product catalogue, this function works in a similar way to the functions which deliver the values of computed properties, e.g. pressure loss.

7 Geometry elements

The parametric model used in ISO 16757 will lead to a small extension of the model of ISO 10303-42 which simply allows parametrization of the geometric object for data exchange. Some additional primitives are required to fulfil the special requirements for parametrization of geometric shapes and primitives in building services.

The additional primitives contain implicitly by their definition the description of their behaviour in parameterization. For example, a transition from rectangle to round follows certain rules while generating variants. Whatever the measures may be, it will continue to be a transition from rectangle to round which can be winded off from a metal sheet.

Using this implicit definition, it is much easier to parameterize such a primitive with a few measures than to redefine all these rules forming, for example, a boundary representation with B-spline surfaces.

Geometry elements are divided into:

— Line elements

Clause no:	Primitive identifier	STEP definition	IFC4 definition
1	Line	ISO 10303-42:2014, 4.4.24	IFC4 8.9.3.30
2	Circle	ISO 10303-42:2014, 4.4.26	IFC4 8.9.3.17
3	Circular arc (trimmed curve)	ISO 10303-42:2014, 4.4.40	IFC4 8.9.3.52
4	Spline (b_spline_curve)	ISO 10303-42:2014, 4.4.34	IFC4 8.9.3.7

Especially for symbols, line elements are required to display their shape. The measures and coordinates of the line elements are parametrizable.

— CSG form primitives

Normal CSG-primitives (Constructive Solid Geometry) such as blocks, cylinders, and sweep-bodies can be concatenated to produce shapes. Their definition is based on ISO 10303-42 definitions of positions, directions, curves and faces.

— CSG sheet metal primitives

Special sheet metal primitives for the building services, such as rectangle-round transitions, oval channels, tee-branches and wye-pieces, can be handled in the same way as normal CSG-primitives (see Figure 10).

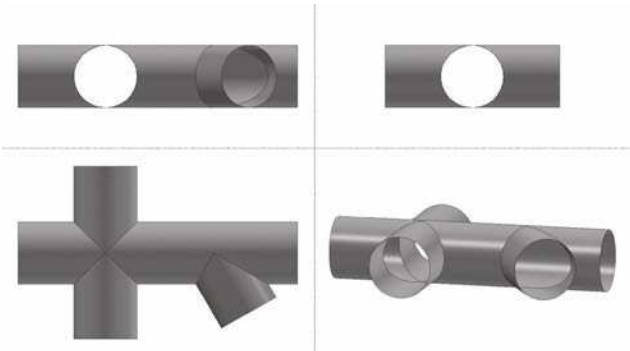

Figure 10 — Example — Sheet metal primitives can be concatenated by Boolean operations

Alternatively, a software application can display sheet metal primitives by three methods.

a) Ignoring the wall thickness and displaying the primitive as normal solid (see Figure 11).

b) Ignoring the wall thickness and displaying the base and end faces of the primitive as transparent (see Figure 12).

c) Displaying the wall thickness as result of an internal Boolean difference with a smaller solid (see Figure 13).

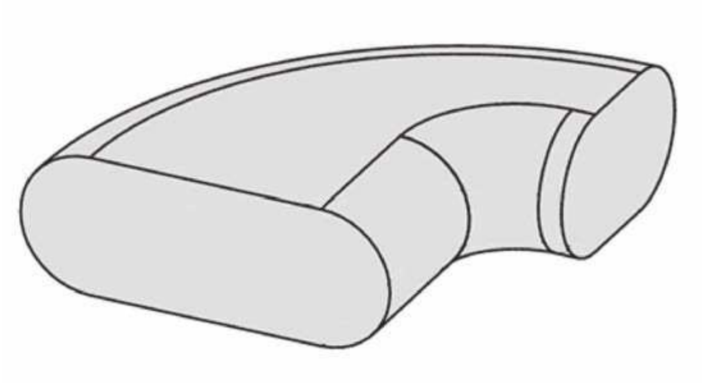

Figure 11 — Example — Wall thickness of CSG sheet metal primitives not applied

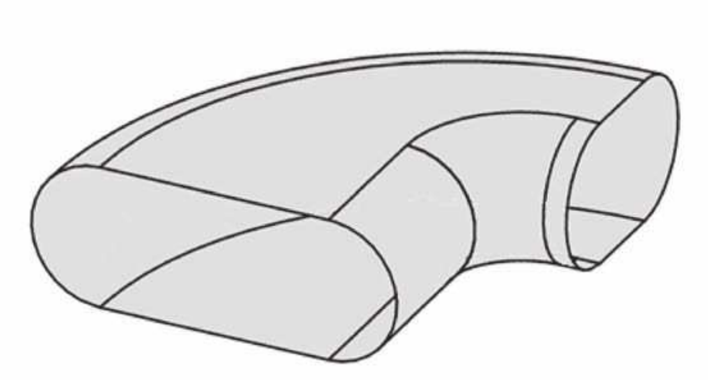

Figure 12 — Example — Thin wall thickness of CSG sheet metal primitives not applied but displayed by transparent base and end faces

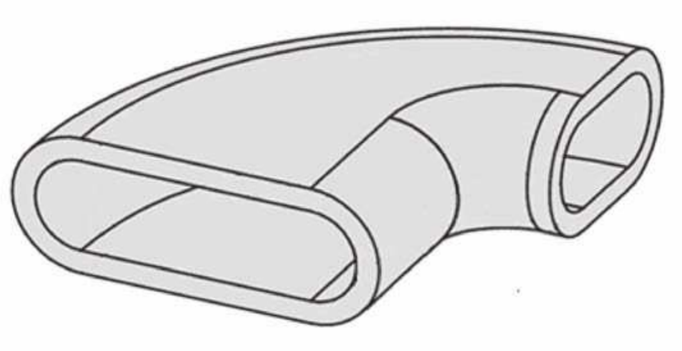

Figure 13 — Example — Wall thickness of CSG sheet metal primitives as Boolean difference to a similar smaller solid

Geometric elements inside the primitive (damper blades in fire dampers, measurement equipment) can be set as an independent CSG tree, not concatenated to the surrounding sheet-metal primitives (see Figure 14).

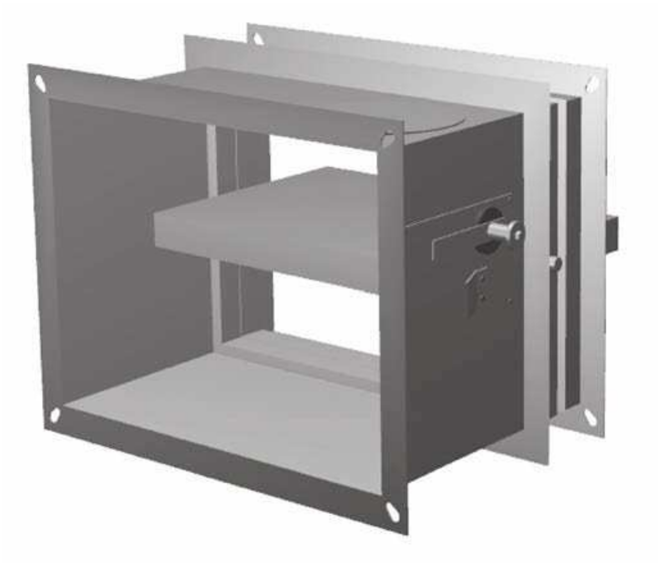

Figure 14 — Example — Air condition fire damper CSG-model made of CSG sheet metal primitives

7.1 CSG form primitives

The geometric model uses the following form primitives:

Clause no:	Primitive identifier	STEP definition	IFC4 definition
1	half_space_solid	ISO 10303-42:2014, 6.4.21	IFC4 8.8.3.20
2	block	ISO 10303-42:2014, 6.4.8	IFC4 8.8.3.3
3	right_angular_wedge	ISO 10303-42:2014, 6.4.9	—
5	sphere	ISO 10303-42:2014, 6.4.14	IFC4 8.8.3.31
6	right_circular_cone	ISO 10303-42:2014, 6.4.15	IFC4 8.8.3.26
7	right_circular_cylinder	ISO 10303-42:2014, 6.4.16	IFC4 8.8.3.27
8	excentric_cone	ISO 10303-42:2014, 6.4.17	—
10	extruded_face_solid	ISO 10303-42:2014, 6.4.33	IFC4 8.8.3.12
11	revolved_face_solid	ISO 10303-42:2014, 6.4.34	IFC4 8.8.3.24
12	faceted_b_rep	ISO 10303-42:2014, 6.4.4	IFC4 8.8.3.15

The following form primitives are defined in addition:

Clause no:	Primitive identifier	Definition	Image
1	uniform_polyhedral_prism	A.1	Figure A.1
2	toroidal_bend_transition	A.2	Figure A.2

7.2 CSG sheet metal primitives

The following sheet metal form primitives and respective identifiers are defined:

Clause no:	Primitive identifier	Definition	Image
1	rectangular_duct	A.3	Figure A.3
2	rectangular_duct_transition	A.4	Figure A.4
3	rectangular_duct_bend	A.5	Figure A.5
5	rectangular_duct_tee	A.6	Figure A.6

Clause no:	Primitive identifier	Definition	Image
6	rectangular_duct_wye	A.7	Figure A.7
7	rectangle_oval_transition	A.8	Figure A.8
8	rectangle_round_transition	A.9	Figure A.9
10	trapezoidal_duct	A.10	Figure A.10
11	trapezoidal_duct_transition	A.11	Figure A.11
12	oval_duct	A.12	Figure A.12
13	oval_duct_transition	A.13	Figure A.13
14	oval_duct_bend	A.14	Figure A.14
15	oval_round_transition	A.15	Figure A.15
16	round_pipe	A.16	Figure A.16
17	round_pipe_transition	A.17	Figure A.17
18	round_pipe_radius_transition	A.18	Figure A.18
19	round_pipe_bend	A.19	Figure A.19
20	round_pipe_wye	A.20	Figure A.20
21	extruded_face_solid	A.21	Figure A.21
22	revolved_face_solid	A.22	Figure A.22

7.3 Expanded CSG primitives

The following expanded primitives and respective identifiers are defined:

Clause no:	Primitive identifier	STEP definition	IFC4 definition
1	Text	ISO 10303-41	IFC4 8.11.2.88
2	Repeat	ISO 10303-11:2004, 13.9	—
3	Reference	ISO 10303-11:2004, 12.7	—

7.4 Mapping of ISO 16757 geometry to parametrizable STEP and IFC geometry

As far as possible, the primitive definitions of ISO 16757 are based on the definitions of ISO 10303-42 (STEP Geometry) and ISO 16739 (IFC).

ISO 16757 needs 36 parametrizable geometric primitives with most common forms in building services under avoidance of faceted boundary representations.

By building volume solids with these primitives, it is possible to represent the numerous product variants in a catalogue with a single parametrized geometric model.

The result of a selection within a product catalogue is a defined, generated product variant where all primitives have static measures. When the static variant is generated, all its primitives, defined in this standard additional to STEP and IFC, can be mapped to STEP or IFC primitives or boundary representations.

ISO 10303-42 (STEP) solids

ISO 10303-42 (STEP) offers only 11 volume solid primitives:

— block_volume

— wedge_volume

— spherical_volume,

— cylindrical_volume

— eccentric_conical_volume,

— toroidal_volume

— pyramid_volume

— b_spline_volume,

— ellipsoid_volume

— tetrahedron_volume

— hexahedron_volume

By restricting the geometric modeller to these 10 primitives, many building services components can only be modeled with B-spline volumes or faceted boundary representations.

Table 1 shows the mapping of ISO 16757 to ISO 10303-42 volume solid primitives.

Table 1 — Mapping solid volume primitives of ISO 16757 to ISO 10303-42 (STEP)

ISO 16757	ISO 10303-42 (STEP)
Section plan	Half_space_solid
Cuboid	Block_volume
Rectangular three-sided prism	Wedge_volume
Uniform polyhedral prism	--- B-spline volume
Cylinder	Cylindrical_volume
Truncated cone	Right_circular_cone with half_space_solid
Sphere	Spherical_volume
Toroidal bend	Toroidal_volume
Toroidal bend transition	--- B-spline volume
Solid of translation	Extruded_Area_Solid
Solid of revolution	Extruded_Area_Solid
Generally bounded solid	--- B-spline volume
Rectangular duct	Block_volume
Rectangular duct transition	--- B-spline volume
Rectangular duct bend	--- B-spline volume
Rectangular duct tee	--- B-spline volume
Rectangular duct Y-branch	--- B-spline volume
Rectangular/oval transition	--- B-spline volume
Rectangular/round transition	--- B-spline volume
Trapezoidal duct	Extruded_Area_Solid
Trapezoidal duct transition	--- B-spline volume
Oval duct	Extruded_Area_Solid
Oval duct transition	--- B-spline volume
Oval duct bend	--- B-spline volume
Oval/round transition	--- B-spline volume
Round pipe	Swept_Disk_Solid with Parametrized_Profile_Def
Round pipe transition	Right_Circular_Cone with half_Space_Solid
Round pipe radius transition	Revolved_Area_Solid with Parametrized_Profile_Def
Round pipe bend	--- B-spline volume
Round pipe Y-branch	--- B-spline volume

Table 1 *(continued)*

ISO 16757	ISO 10303-42 (STEP)
Duct of translation	Extruded_Area_Solid with Parametrized_Profile_Def
Duct of revolution	Revolved_Area_Solid with Parametrized_Profile_Def
Generally bounded duct	--- B-spline volume

ISO 16739 (IFC) solids

ISO 16739 (IFC) offers 13 volume primitives:

— IfcBlock

— IfcBoxedHalfSpace

— IfcExtrudedAreaSolid

— IfcHalfSpaceSolid

— IfcPolygonalBoundedHalfSpace

— IfcRectangularPyramid

— IfcRevolvedAreaSolid

— IfcRightCircularCone

— IfcRightCircularCylinder

— IfcSectionedSpine

— IfcSphere

— IfcSurfaceCurveSweptAreaSolid

— IfcSweptDiskSolid

By restricting the geometric modeller to these 13 primitives, many building services components can only be modelled with B-spline volumes or faceted boundary representations.

Table 2 shows the mapping of ISO 16757 to ISO 16739 (IFC 4) volume solid primitives:

Table 2 — Mapping of solid volume primitives of ISO 16757 to ISO 16739 (IFC 4)

ISO 16757	ISO 16739 **(IFC)**
Section plan	IfcHalfSpaceSolid
Cuboid	IfcBlock
Rectangular three-sided prism	IfcExtrudedAreaSolid with IfcParametrizedProfileDef
Uniform polyhedral prism	--- *Polyhedron is not a defined profile in IfcParametrizedProfileDef for IfCExtrudedAreaSolid*
Cylinder	IfcRightCircularCylinder
Truncated cone	IfcRightCircularCone with IfcHalfSpaceSolid
Sphere	IfcSphere
Toroidal bend	IfcRevolvedAreaSolid with IfcParametrizedProfileDef
Toroidal bend transition	--- *Faceted boundary representation*
Solid of translation	IfcExtrudedAreaSolid with IfcParametrizedProfileDef
Solid of revolution	IfcExtrudedAreaSolid with IfcParametrizedProfileDef
Generally bounded solid	--- *Faceted boundary representation*

Table 2 *(continued)*

ISO 16757	ISO 16739 **(IFC)**
Rectangular duct	–-- *Faceted boundary representation*
Rectangular duct transition	*IfcBlock*
Rectangular duct bend	–-- *Faceted boundary representation*
Rectangular duct tee	–-- *Faceted boundary representation*
Rectangular duct Y-branch	–-- *Faceted boundary representation*
Rectangular/oval transition	–-- *Faceted boundary representation*
Rectangular/round transition	–-- *Faceted boundary representation*
Trapezoidal duct	*IfcRevolvedAreaSolid with IfcParametrizedProfileDef*
Trapezoidal duct transition	–-- *Faceted boundary representation*
Oval duct	*IfcExtrudedAreaSolid with IfcParametrizedProfileDef*
Oval duct transition	–-- *Faceted boundary representation*
Oval duct bend	–-- *Faceted boundary representation*
Oval/round transition	–-- *Faceted boundary representation*
Round pipe	*IfcSweptDiskSolid with IfcParametrizedProfileDef*
Round pipe transition	*IfcRightCircularCone with IfcHalfSpaceSolid*
Round pipe radius transition	*IfcRevolvedAreaSolid with IfcParametrizedProfileDef*
Round pipe bend	–-- *Faceted boundary representation*
Round pipe Y-branch	–-- *Faceted boundary representation*
Duct of translation	*IfcExtrudedAreaSolid with IfcParametrizedProfileDef*
Duct of revolution	*IfcRevolvedAreaSolid with IfcParametrizedProfileDef*
Generally bounded duct	–-- *Faceted boundary representation*

Annex A
(normative)

Additional geometry elements

A.1 General

This Annex describes the parametrizable geometry elements that are needed to represent products in building services but which are not standardized in ISO 10303-42 (STEP geometry) and/or ISO 16739 (IFC).

The definitions of these extension are listed as EXPRESS specification in the form of ISO 10303-42 and in the form of ISO 16739 as EXPRESS (ISO 10303-11) and XSD (ISO 10303-28) specification.

A.2 Uniform polyhedral prism

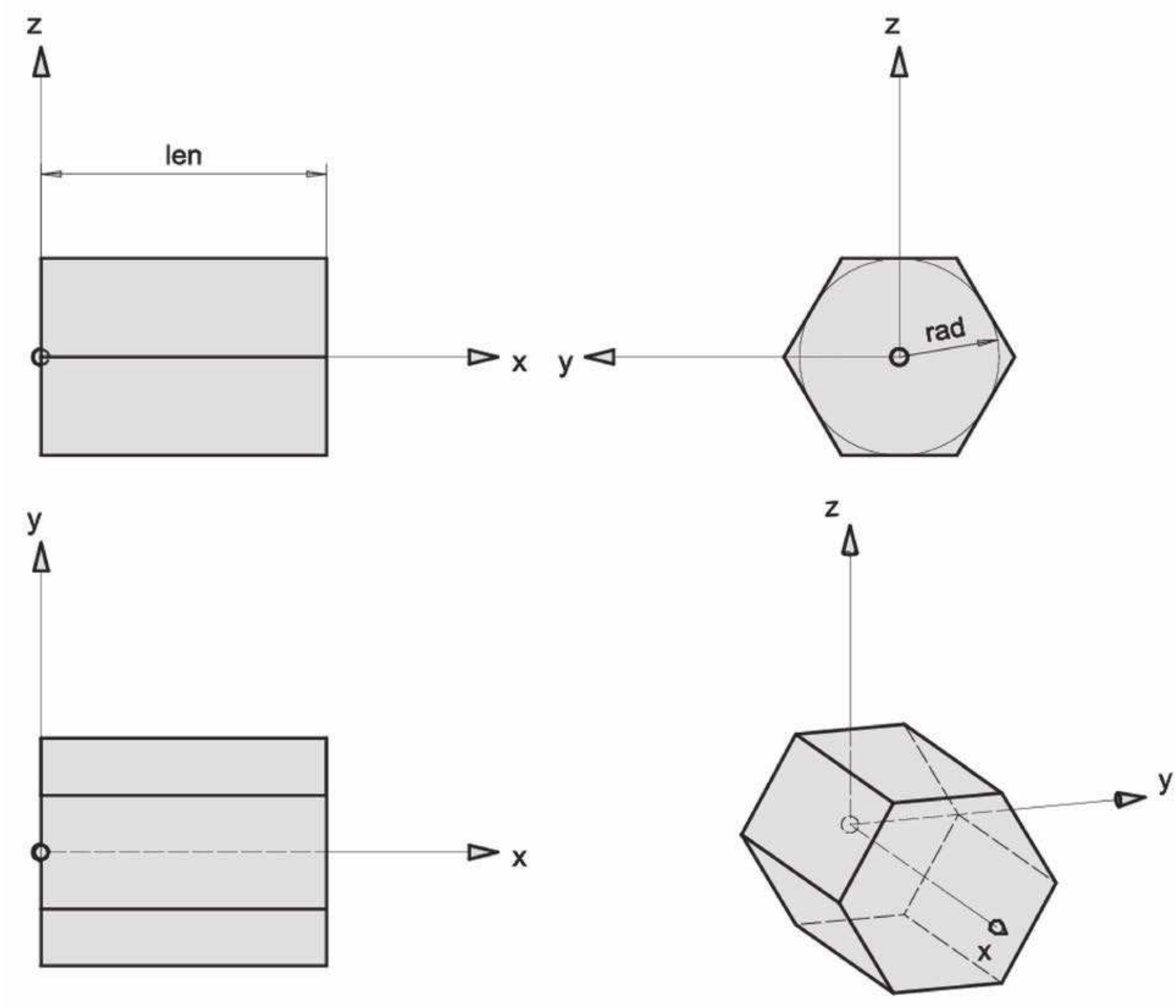

Figure A.1 — Uniform polyhedral prism

Primitive identifier: 'uniform_polyhedral_prism'

A uniform polyhedral prism is a type of regular prismatic CSG primitive with a uniform polyhedral base. All side faces are rectangles with the same length and width.

EXPRESS Specification:

```
*)
ENTITY uniform_polyhedral_prism
   SUBTYPE OF (geometric_representation_item);
     position: axis2_placement_3d;
     len:      positive_length_measure;
     rad:      positive_length_measure;
     num:      integer;
     WHERE
       WR1:    len > 0;
       WR2:    rad > 0;
       WR3:    num >= 3;
END_ENTITY;
(*
```

Attribute definitions:

position: The location and orientation of the placement axis system for the uniform polyhedral prism.

The base of the **uniform_polyhedral_prism** has its centre at **position.location** and the bottom side (minimum Z) of the polyhedron is parallel to the placement x-y plane.

len: The length of the base measured along the placement x-axis (position.p[1]).

rad: The radius of the circle tangential to all vertices of the base face (identical with the radius of the circle tangential to all vertices of the end face).

num: The number of edges and corresponding vertices of the base face (identical with the number of edges and corresponding vertices of the end face).

Formal propositions:

WR1: The length **len** shall be greater than 0.

WR2: The radius **rad** shall be greater than 0.

WR3: The number of edges **num** shall be greater than 2.

Specifications in IFC:

The uniform polyhedral prism is defined by using the existing IFC primitive: **IfcExtrudedAreaSolid.**

The base face is defined as parametrizable 2D profile.

XSD Specification of the 2D profile:

```
<xs:element name="IfcUniformPolyhedralProfileDef" type="ifc:IfcRectangleProfileDef"
substitutionGroup="ifc:IfcParametrizedProfileDef" nillable="true"/>
 <xs:complexType name="IfcUniformPolyhedralProfileDef">
  <xs:complexContent>
   <xs:extension base="ifc:IfcParametrizedProfileDef">
   <xs:attribute name="rad" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
   <xs:attribute name="num" type="ifc:IfcInteger" use="optional"/>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
```


<u>EXPRESS Specification of the 2D profile (IFC):</u>

```
ENTITY IfcUniformPolyhedralProfileDef

SUBTYPE OF IfcParametrizedProfileDef;
  <xs:attribute name="rad" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
  <xs:attribute name="num" type="ifc:IfcInteger" use="optional"/>
END_ENTITY;
```

A.3 Toroidal bend transition

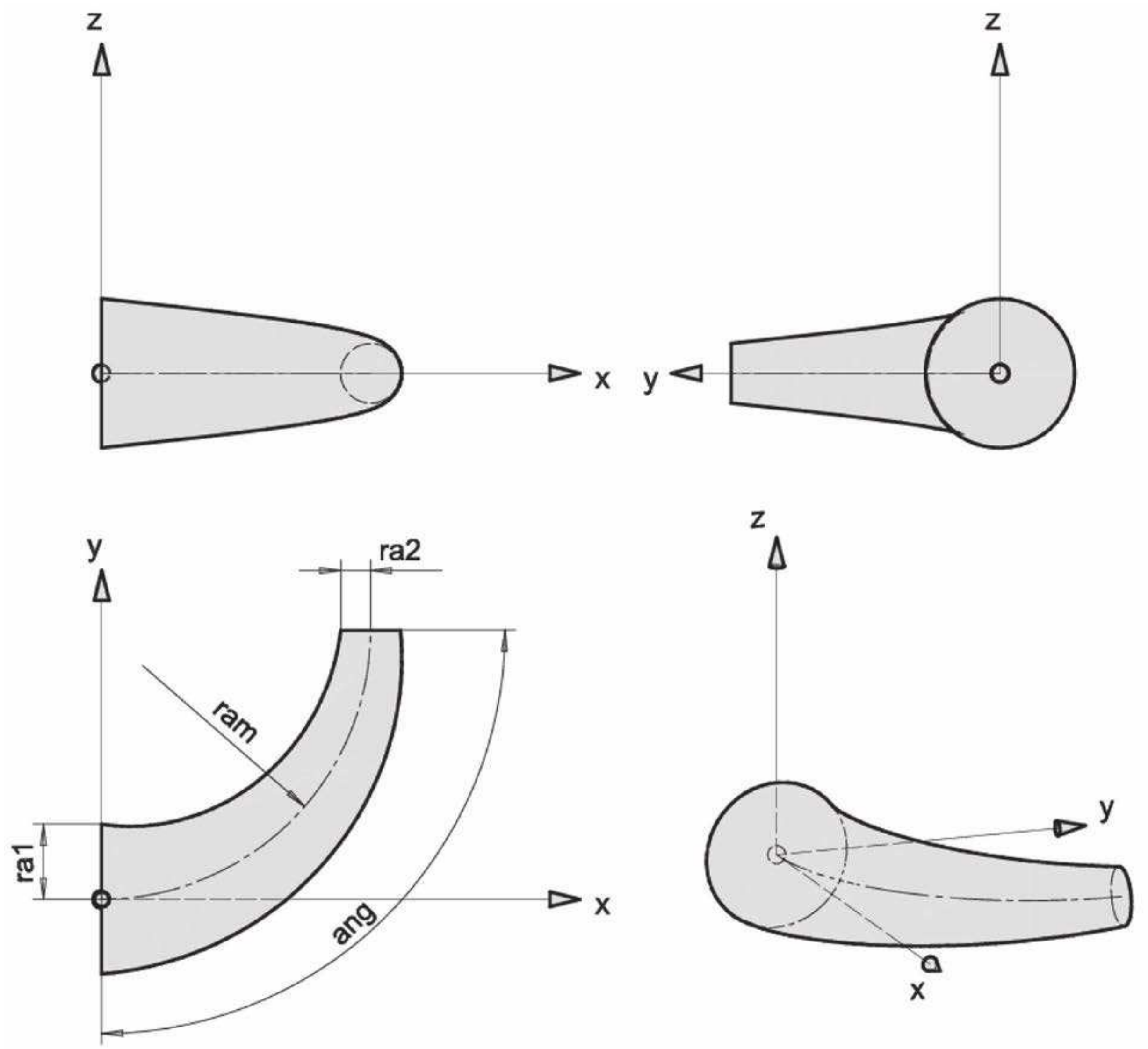

Figure A.2 — Toroidal bend transition

<u>Primitive identifier: 'toroidal_bend_transition'</u>

A toroidal bend transition is a type of CSG primitive with a circular base face and a circular end face. The directrix is a circle segment lying in the x-y plane with its centre point on the positive y-axis.

The cross-section of a plane normal to the directrix plane and the centre point of the directrix is always a circle. The radii of all these circles are that of a cone on the developed length of the directrix.

EXPRESS Specification:

```
*)
ENTITY toroidal_bend_transition
   SUBTYPE OF (geometric_representation_item);
     position: axis2_placement_3d;
     ram:      positive_length_measure;
     ra1:      positive_length_measure;
     ra2:      positive_length_measure;
     ang:      positive_plane_angle_measure;
     WHERE
       WR1:    ram >= MAX(ra1,ra2);
       WR2:    ra1 >= 0;
       WR3:    ra2 >= 0;
       WR4:    – ra2 = 0 ==> ra1 > 0;
               NOT (ra2 = 0) OR (ra1 > 0);
       WR5:    – ra1 = 0 ==> ra2 > 0;
               NOT (ra1 = 0) OR (ra2 > 0);
       WR6:    ang > 0;
       WR7:    ang <= 360°;
END_ENTITY;
(*
```

Attribute definitions:

position: The location and orientation of the placement axis system for the toroidal bend transition.

position defines a placement coordinate system for the **toroidal_bend_transition**. The base circular face of the toroidal_bend_transition has its centre at **position.location** and the directrix has its centre on the placement y-axis.

ram: The radius (mayor radius) of the directrix.

ra1: The radius at the base face which is parallel to the placement y-z plane.

ra2: The radius at the end face.

ang: The turn angle between the planes of the two circular faces of the solid, measured in the sector containing the solid.

Formal propositions:

WR1: The radius **ram** shall be greater or equal to the maximum of **ra1** and **ra2.**

WR2: The radius **ra1** shall be greater or equal to 0.

WR3: The radius **ra2** shall be greater or equal to 0.

WR4: The radius **ra1** shall be greater than 0 if **ra2** is equal to 0.

WR5: The radius **ra2** shall be greater than 0 if **ra1** is equal to 0.

WR6: The angle **ang** shall be greater than 0.

WR7: The angle **ang** shall be less or equal to 360°.

Informal propositions:

IP1: The toroidal_bend_transition can become a torus bend, if the initial and the end radius are equal.

IP2: A torus bend with a turn angle of 360° is a closed torus.

Specifications in IFC:

Toroidal bend transitions are defined with existing IFC definitions.

The toroidal bend transition is defined as:	**IfcRevolvedAreaSolid**
The base face is the parametrizable 2D profile:	**IfcCircleProfileDef**
The base face will be transformed by:	**IfcDerivedProfileDef**
	IfcCartesianTransformationOperator2D[a]

a http://www.buildingsmart-tech.org/ifc/IFC4/final/html/schema/ifcgeometryresource/lexical/ifc-cartesiantransformationoperator2d.htm

A.4 Rectangular duct

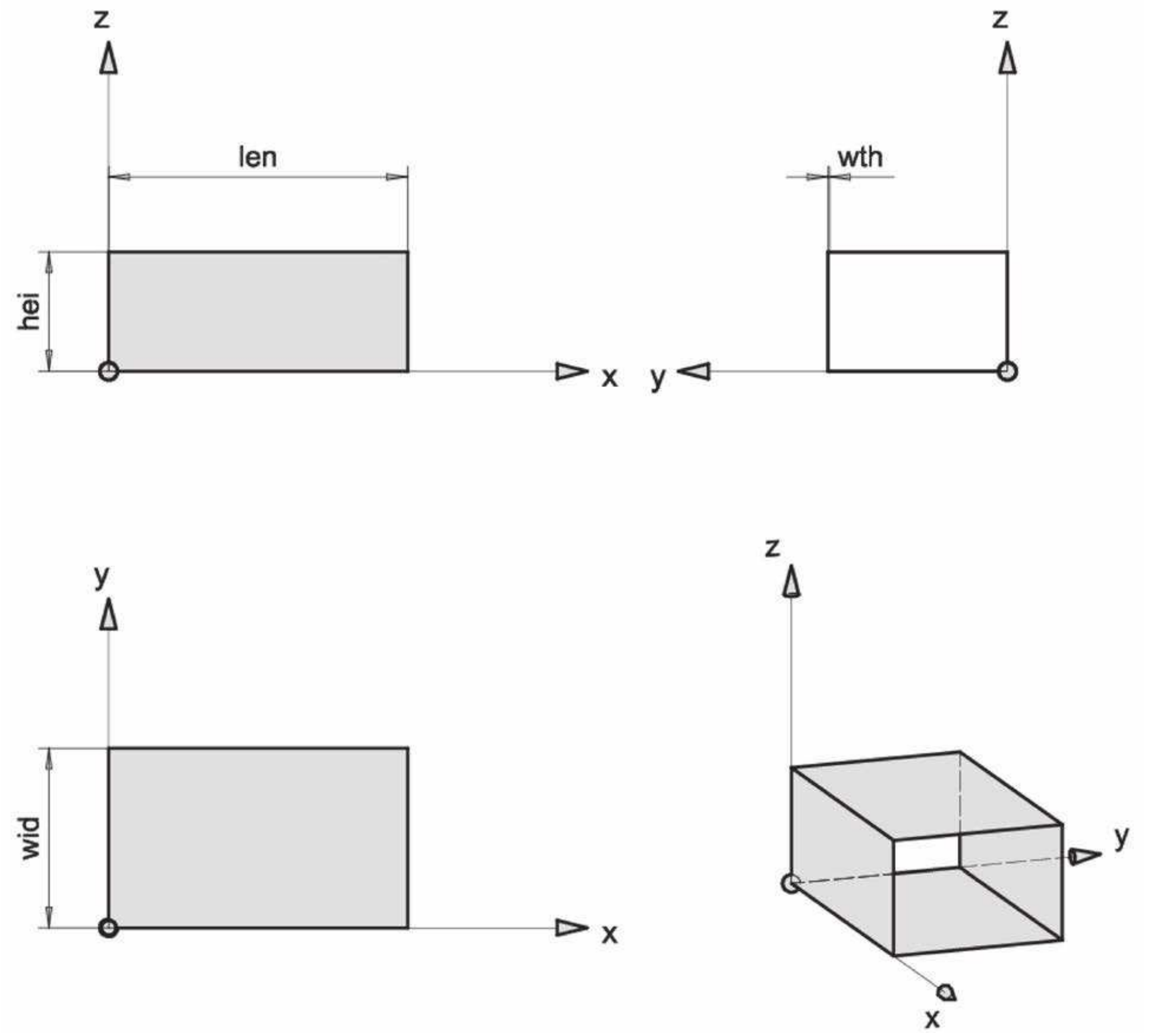

Figure A.3 — Rectangular duct

Primitive identifier: 'rectangular_duct'

A rectangular duct is a type of regular prismatic CSG primitive with rectangular base. Its behaviour in Boolean operations is the same as that of a block. The base and end surface are declared as open, so they can be displayed as invisible to represent a duct of sheet metal.

EXPRESS Specification:

```
*)
ENTITY rectangular_duct
   SUBTYPE OF (geometric_representation_item);
     position: axis2_placement_3d;
     wth:      positive_length_measure;
     len:      positive_length_measure;
     wid:      positive_length_measure;
     hei:      positive_length_measure;
     WHERE
       WR1:    wth > 0
       WR2:    len > 0
       WR3:    wid > 2*wth
       WR4:    hei > 2*wth
END_ENTITY;
(*
```

Attribute definitions:

position: The location and orientation of the placement axis system for the rectangular duct.

The **rectangular_duct** has one vertex at **position.location** and its edges are aligned with the placement axes in the positive direction.

wth: The wall thickness size of the thin sheet metal forming the duct (position.p[1]).

len: The length size of the rectangular duct along the placement x-axis (position.p[1]).

wid: The width size of the rectangular duct along the placement y-axis (position.p[2]).

hei: The height size of the rectangular duct along the placement z-axis (position.p[3]).

Formal propositions:

WR1: The wall thickness **wth** shall be greater than 0.

WR2: The length **len** shall be greater than 0.

WR3: The width **wid** shall be greater than 2 * **wth**

WR4: The height **hei** shall be greater than 2 * **wth**

Specifications in IFC:

The rectangular duct is defined with existing IFC definitions.

The rectangular_duct is defined as: **IfcExtrudedAreaSolid**

The base face is the parametrizable 2D profile: **IfcRectangleHollowProfileDef**

A.5 Rectangular duct transition

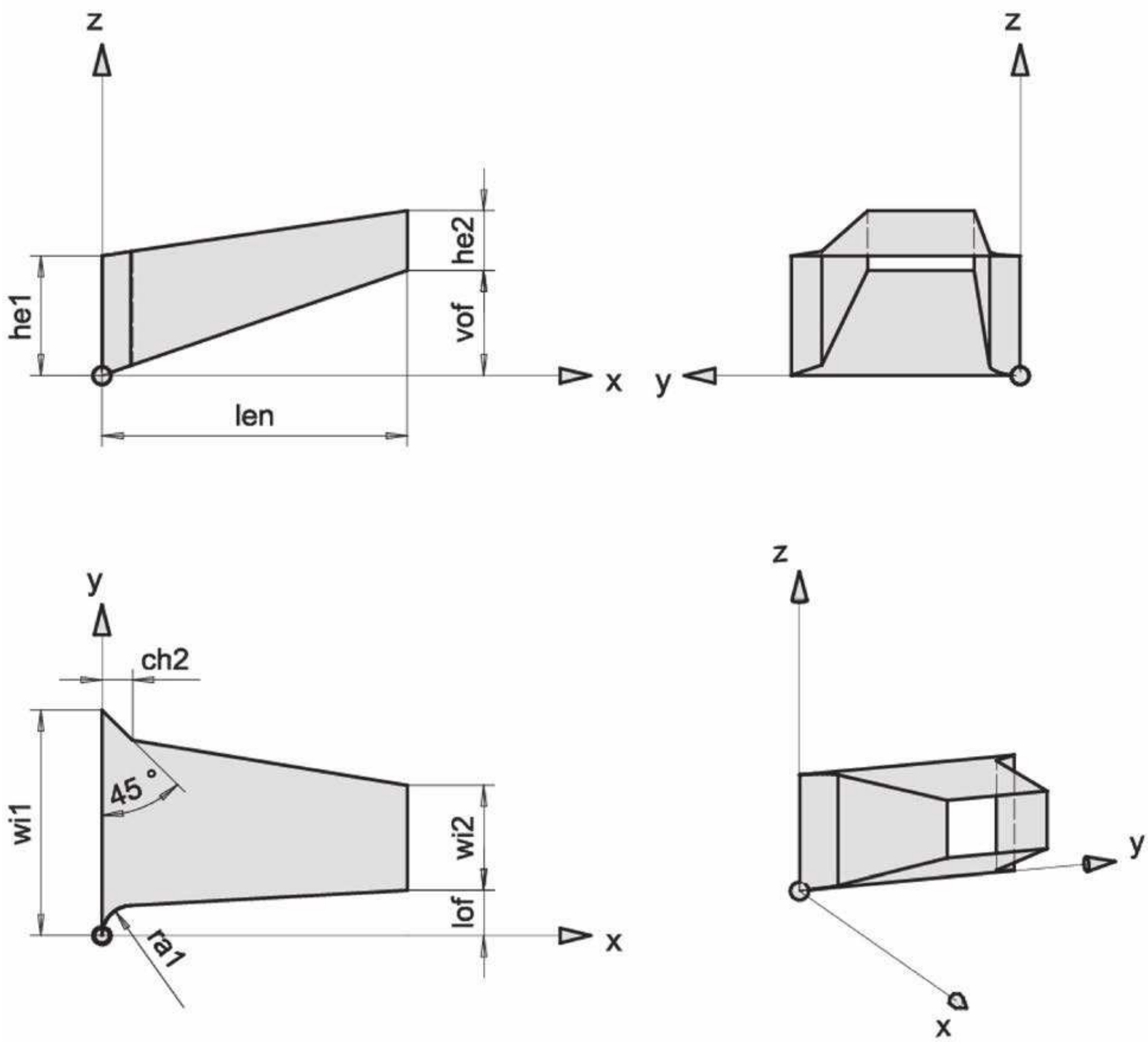

Figure A.4 — Rectangular duct transition

<u>**Primitive identifier: ‘rectangular_duct_transition’**</u>

A rectangular duct transition is a type of regular CSG primitive with rectangular base and end. Its behaviour in Boolean operations is the same as that of a solid primitive. The base and end surfaces can be displayed as invisible to perform a duct transition of thin sheet metal.

<u>EXPRESS Specification:</u>

```
*)
ENTITY rectangular_duct_transition
   SUBTYPE OF (geometric_representation_item);
     position: axis2_placement_3d;
     wth:      positive_length_measure;
     len:      positive_length_measure;
     wi1:      positive_length_measure;
     wi2:      positive_length_measure;
     he1:      positive_length_measure;
     he2:      positive_length_measure;
     lof:      length_measure;
     vof:      length_measure;
```



```
   ra1: positive_length_measure;
   ra2: positive_length_measure;
   ch1: positive_length_measure;
   ch2: positive_length_measure;
WHERE
   WR1:  wth >= 0
   WR2:  len > 0
   WR3:  wi1 > 2*wth
   WR4:  wi2 > 2*wth
   WR5:  he1 > 2*wth
   WR6:  he2 > 2*wth
   WR7:  ra1 >= 0
   WR8:  ra2 >= 0
   WR9:  ch1 >= 0
   WR10: ch2 >= 0
   WR11: - ch1 > 0 ==> ra1 = 0
         NOT (ch1 > 0) OR (ra1 = 0);
   WR12: - ch2 > 0 ==> ra2 = 0
         NOT (ch2 > 0) OR (ra2 = 0);
   WR13: - ra1 > 0 ==> ch1 = 0
         NOT (ra1 > 0) OR (ch1 = 0);
   WR14: - ra2 > 0 ==> ch2 = 0
         NOT (ra2 > 0) OR (ch2 = 0);
END_ENTITY;
(*
```

Attribute definitions:

position: The location and orientation of the placement axis system for the rectangular duct transition.

The **rectangular_duct_transition** has one vertex at **position.location** and its edges are aligned with the placement axes in the positive sense.

wth: The wall thickness size of the thin sheet metal forming the duct transition (position.p[1]).

len: The length size of the rectangular duct transition along the placement x-axis (position.p[1]).

wi1: The width size of the rectangular duct transition base along the placement y-axis (position.p[2]).

wi2: The width size of the rectangular duct transition end along the placement y-axis (position.p[2]).

he1: The height size of the rectangular duct transition base along the placement z-axis (position.p[3]).

he2: The height size of the rectangular duct transition end along the placement z-axis (position.p[3]).

lof: The lateral offset of the rectangular duct transition end along the placement y-axis (position.p[2]).

vof: The vertical offset of the rectangular duct transition end along the placement z-axis (position.p[3]).

ra1: The rounding off radius on the lower y-side of the base.

ra2: The rounding off radius on the higher y-side of the base.

ch1: The 45° chamfer depth on the lower y-side of the base.

ch2: The 45° chamfer depth on the higher y-side of the base.

Formal propositions:

WR1: The wall thickness **wth** shall be greater than 0.

WR2: The length **len** shall be greater than 0.

WR3: The width **wi1** shall be greater than 2 * **wth.**

WR4: The width **wi2** shall be greater than 2 * **wth.**

WR5: The height **he1** shall be greater than 2 * **wth.**

WR6: The height **he2** shall be greater than 2 * **wth.**

WR7: The radius **ra1** shall be greater or equal to 0.

WR8: The radius **ra2** shall be greater or equal to 0.

WR9: The chamfer depth **ch1** shall be greater or equal to 0.

WR10: The chamfer depth **ch2** shall be greater or equal to 0.

WR11: The radius **ra1** shall be 0 if the chamfer depth **ch1** is greater than 0.

WR12: The radius **ra2** shall be 0 if the chamfer depth **ch2** is greater than 0.

WR13: The chamfer depth **ch1** shall be 0 if the radius **ra1** is greater than 0.

WR14: The chamfer depth **ch2** shall be 0 if the radius **ra2** is greater than 0.

Informal propositions:

IP1: Vertical and lateral offsets can be entered as positive or negative values referred to the placement y-axis or z-axis.

IP2: The inlet radii or chamfers can be given for one or both sides. The values of inlet chamfers determine the depth of 45° chamfers. On the same side values may not be given for both an inlet radius and a chamfer— either there is a radius or a chamfer.

Specifications in IFC:

Duct transitions without radii and chamfers are defined with existing IFC definitions.

The rectangular_duct_transition is defined as: **IfcExtrudedAreaSolid**

The base face is the parametrizable 2D profile: **IfcRectangleHollowProfileDef**

Included is the transformation of the start face by: **IfcDerivedProfileDef**

IfcCartesianTransformationOperator2D

Duct transitions with radii and chamfers are defined with the following IFC definitions.

XSD Specification (IFC):

```
<xs:element name="IfcRectangularDuctTransition" type="ifc:IfcRectangularDuctTransition"
substitutionGroup="ifc:IfcCsgPrimitive3D" nillable="true"/>
 <xs:complexType name="IfcRectangularDuctTransition">
  <xs:complexContent>
   <xs:extension base="ifc:IfcCsgPrimitive3D">
    <xs:attribute name="wth" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="len" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="wi1" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="wi2" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
```

```
    <xs:attribute name="he1" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="he2" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="lof" type="ifc:IfcLengthMeasure" use="optional"/>
    <xs:attribute name="vof" type="ifc:IfcLengthMeasure" use="optional"/>
    <xs:attribute name="ra1" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="ra2" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="ch1" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="ch2" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
```

EXPRESS Specification (IFC):

```
ENTITY IfcRectangularDuctTransition1)
  SUBTYPE OF IfcCsgPrimitive3D2);
     wth: IfcPositiveLengthMeasure;
     len: IfcPositiveLengthMeasure;
     wi1: IfcPositiveLengthMeasure;
     wi2: IfcPositiveLengthMeasure;
     he1: IfcPositiveLengthMeasure;
     he2: IfcPositiveLengthMeasure;
     lof: IfcLengthMeasure;
     vof: IfcLengthMeasure;
     ra1: IfcPositiveLengthMeasure;
     ra2: IfcPositiveLengthMeasure;
     ch1: IfcPositiveLengthMeasure;
     ch2: IfcPositiveLengthMeasure;
END_ENTITY;
```

1) http://www.buildingsmart-tech.org/ifc/IFC4/final/html/schema/ifcgeometricmodelresource/lexical/ifcblock.htm

2) http://www.buildingsmart-tech.org/ifc/IFC4/final/html/schema/ifcgeometricmodelresource/lexical/ifccsgprimitive3d.htm

A.6 Rectangular duct bend

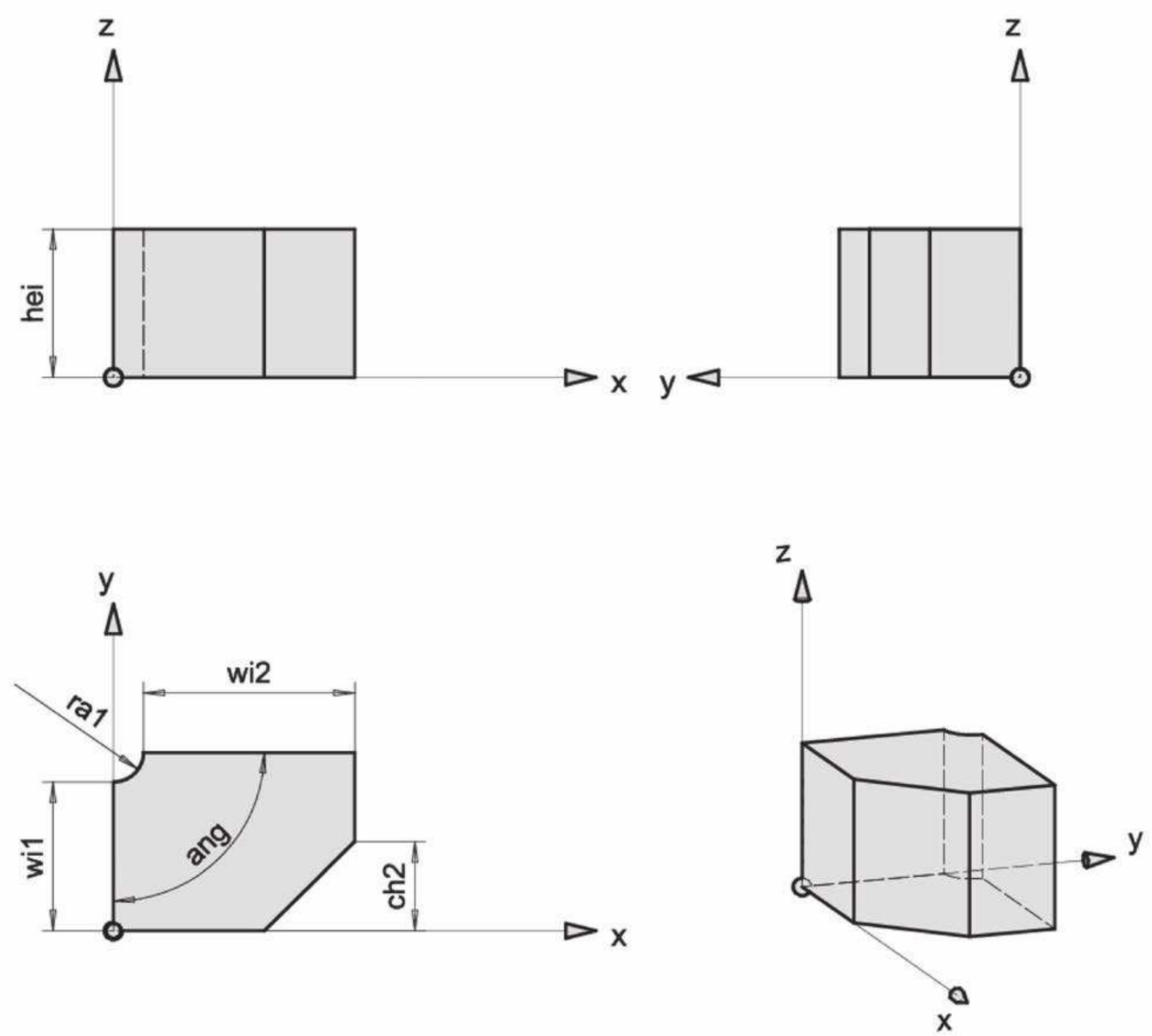

Figure A.5 — Rectangular duct bend

Primitive identifier: 'rectangular_duct_bend'

A rectangular duct bend is a type of regular CSG primitive with rectangular base and end. Its behaviour in Boolean operations is the same as that of a solid primitive. The base and end surface can be displayed as invisible to perform a duct bend of thin sheet metal.

EXPRESS Specification:

```
*)
ENTITY rectangular_duct_bend
  SUBTYPE OF (geometric_representation_item);
    position: axis2_placement_3d;
    wth:      positive_length_measure;
    ang:      positive_plane_angle_measure;
    wi1:      positive_length_measure;
    wi2:      positive_length_measure;
    hei:      positive_length_measure;
    ra1:      positive_length_measure;
    ra2:      positive_length_measure;
    ch1:      positive_length_measure;
```



```
    ch2:       positive_length_measure;
    WHERE
      WR1:     wth >= 0
      WR2:     ang > 0
      WR3:     ang < 360
      WR4:     wi1 > 2*wth
      WR5:     wi2 > 2*wth
      WR6:     hei > 2*wth
      WR7:     ra1 >= 0
      WR8:     ra2 >= 0
      WR9:     ch1 >= 0
      WR10:    ch2 >= 0
      WR11:    - ch1 = 0 ==> ra1 > 0
               NOT (ch1 = 0) OR ra1 > 0;
      WR12:    - ch2 = 0 ==> ra2 > 0
               NOT (ch2 = 0) OR ra2 > 0;
      WR13:    - ra1 = 0 ==> ch1 > 0
               NOT (ra1 = 0) OR ch1 > 0;
      WR14:    - ra2 = 0 ==> ch2 > 0
               NOT (ra2 = 0) OR ch2 > 0;
END_ENTITY;
(*
```

Attribute definitions:

position: The location and orientation of the placement axis system for the rectangular duct bend.

The **rectangular_duct_bend** has one vertex at **position.location** and its edges are aligned with the placement axes in the positive sense.

wth: The wall thickness size of the thin sheet metal forming the duct bend (position.p[1]).

len: The length size of the rectangular duct bend along the placement x-axis (position.p[1]).

ang: The size of the rectangular duct bend along the placement x-axis (position.p[1])

wi1: The size of the rectangular duct bend base along the placement y-axis (position.p[2]).

wi2: The size of the rectangular duct bend end in the x-y-plane in the direction of TA.

hei: The size of the rectangular duct bend base along the placement z-axis (position.p[3]).

ra1: The rounding off radius on the inner side of the duct bend.

ra2: The rounding off radius on the outer side of the duct bend.

ch1: The depth of symmetrical chamfer on the inner side of the duct bend.

ch2: The depth of symmetrical chamfer on the outer side of the duct bend.

Formal propositions:

WR1: The wall thickness **wth** shall be greater than 0.

WR2: The angle **ang** shall be greater than 0.

WR3: The angle **ang** shall be less or equal to 360.

WR4: The width **wi1** shall be greater than 2 * **wth.**

WR5: The width **wi2** shall be greater than 2 * **wth.**

WR6: The height **hei** shall be greater than 2 * **wth.**

WR7: The radius **ra1** shall be greater or equal to 0.

WR8: The radius **ra2** shall be greater or equal to 0.

WR9: The chamfer depth **ch1** shall be greater or equal to 0.

WR10: The chamfer depth **ch2** shall be greater or equal to 0.

WR11: The radius **ra1** shall be 0 if the chamfer depth **ch1** is greater than 0.

WR12: The radius **ra2** shall be 0 if the chamfer depth **ch2** is greater than 0.

WR13: The chamfer depth **ch1** shall be 0 if the radius **ra1** is greater than 0.

WR14: The chamfer depth **ch2** shall be 0 if the radius **ra2** is greater than 0.

Informal propositions:

IP1: All common wall forms of bends and knees are described by entering positive values (bend), negative values (chamfer) or 0 (corner) for the radii. The radii or chamfers can be given for one or both sides.

IP2: The chamfers are symmetrical. Hence, for an overall angle of, e.g. 90°, it is 45°. The cross-section **wi2 * hei** is at the end of the bend having the radius **ra1** or **ra2** or at the end of the respective chamfers **ch1** or **ch2**.

IP3: No concurrent values may be given for radii and chamfers on the same side — either there is a radius or a chamfer. In Figure A.5, an inner radius **ra1** and an outer chamfer **ch2** is given (**ra2** and **ch1** are not given).

IP4: No chamfers are possible with aperture angles between 180° and 360°. Any value given for **ra2** is then ignored. **ra2** is then calculated as the smallest possible tangent circle on the outer walls of the duct bend.

Specifications in IFC:

The rectangular duct bend is defined with the following IFC definitions.

XSD Specification (IFC):

```
<xs:element name="IfcRectangularDuctBend" type="ifc:IfcRectangularDuctBend"
substitutionGroup="ifc:IfcCsgPrimitive3D" nillable="true"/>
 <xs:complexType name="IfcRectangularDuctBend">
  <xs:complexContent>
   <xs:extension base="ifc:IfcCsgPrimitive3D">
    <xs:attribute name="wth" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="ang" type="ifc:IfcPlaneAngleMeasure" use="optional"/>
    <xs:attribute name="wi1" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="wi2" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="hei" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="ra1" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="ra2" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="ch1" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="ch2" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
```

EXPRESS Specification (IFC):

```
ENTITY IfcRectangularDuctBend3)
 SUBTYPE OF IfcCsgPrimitive3D;
    wth: IfcPositiveLengthMeasure;
    ang: IfcPlaneAngleMeasure;
    wi1: IfcPositiveLengthMeasure;
    wi2: IfcPositiveLengthMeasure;
    hei: IfcPositiveLengthMeasure;
    ra1: IfcPositiveLengthMeasure;
    ra2: IfcPositiveLengthMeasure;
    ch1: IfcPositiveLengthMeasure;
    ch2: IfcPositiveLengthMeasure;
END_ENTITY;
```

A.7 Rectangular duct tee

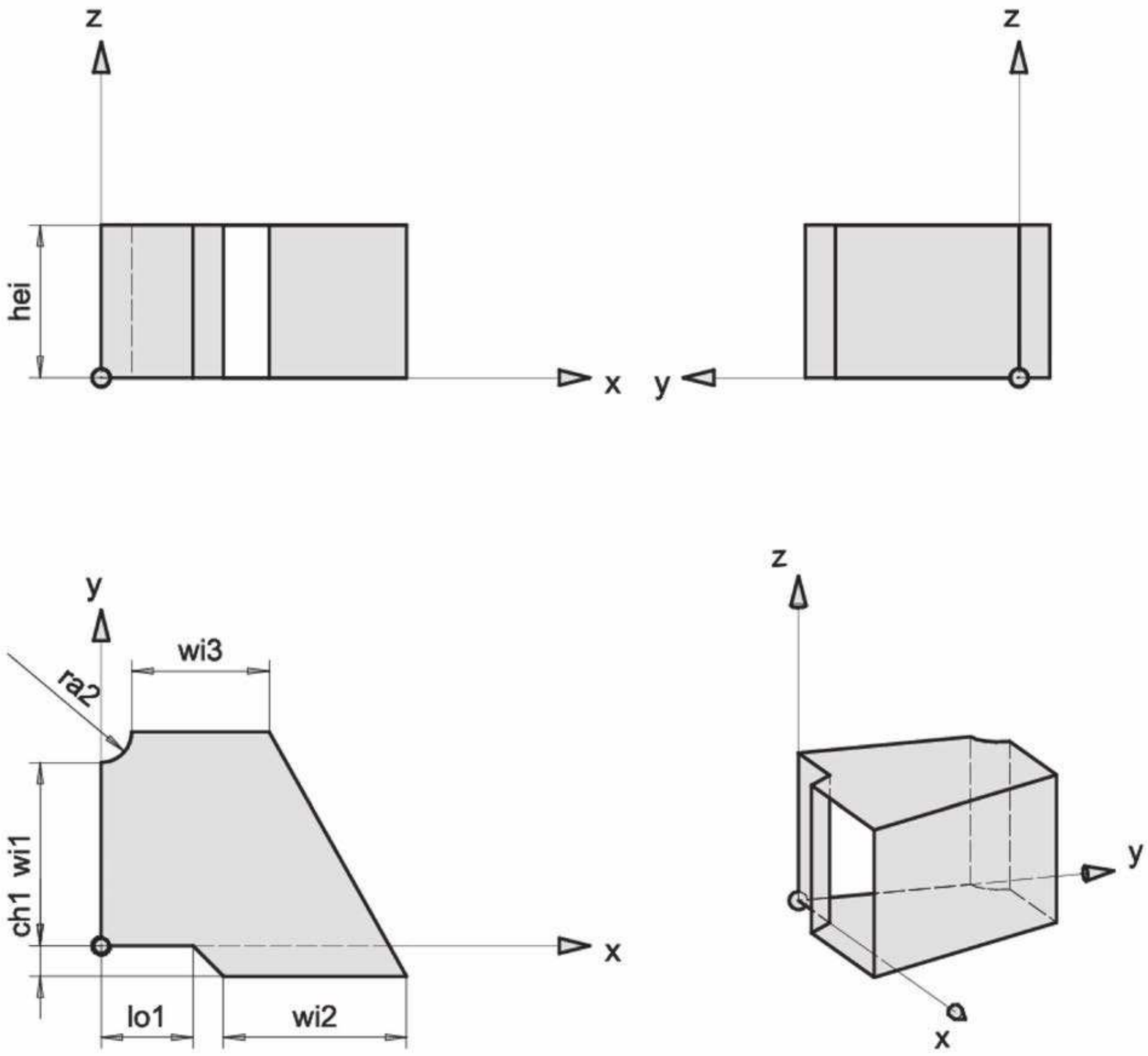

Figure A.6 — Rectangular duct tee

3) http://www.buildingsmart-tech.org/ifc/IFC4/final/html/schema/ifcgeometricmodelresource/lexical/ifcblock.htm

Primitive identifier: 'rectangular_duct_tee'

A rectangular duct tee is a type of regular CSG primitive with one rectangular base and two rectangular ends. Its behaviour in Boolean operations is the same as that of a solid primitive. The base and end surfaces can be displayed as invisible to perform a duct tee of thin sheet metal.

EXPRESS Specification:

```
*)
ENTITY rectangular_duct_tee
  SUBTYPE OF (geometric_representation_item);
    position: axis2_placement_3d;
    wth:      positive_length_measure;
    wi1:      positive_length_measure;
    wi2:      positive_length_measure;
    wi3:      positive_length_measure;
    hei:      positive_length_measure;
    ra1:      positive_length_measure;
    ra2:      positive_length_measure;
    ch1:      positive_length_measure;
    ch2:      positive_length_measure;
    lo1:      positive_length_measure;
    lo2:      positive_length_measure;
    WHERE
      WR1:    wth > 0
      WR2:    wi1 > 2*wth
      WR3:    wi2 > 2*wth
      WR4:    wi3 > 2*wth
      WR5:    hei > 2*wth
      WR6:    ra1 >= 0
      WR7:    ra2 >= 0
      WR8:    ch1 >= 0
      WR9:    ch2 >= 0
      WR10:   - ch1 = 0 ==> ra1 > 0
              NOT (ch1 = 0) OR ra1 > 0;
      WR11:   - ch2 = 0 ==> ra2 > 0
              NOT (ch2 = 0) OR ra2 > 0;
      WR12:   - ra1 = 0 ==> ch1 > 0
              NOT (ra1 = 0) OR ch1 > 0;
      WR13:   - ra2 = 0 ==> ch2 > 0
              NOT (ra2 = 0) OR ch2 > 0;
      WR14:   lo1 >= 0
      WR15:   lo2 >= 0
      WR16:   - lo2 = 0 ==> lo1 > 0
              NOT (lo2 = 0) OR lo1 > 0;
      WR17:   - lo1 = 0 ==> lo2 > 0
              NOT (lo1 = 0) OR lo2 > 0;
END_ENTITY;
(*
```

Attribute definitions:

position: The location and orientation of the placement axis system for the rectangular duct tee.

The **rectangular_duct_tee** has one vertex at **position.location** and its edges are aligned with the placement axes in the positive sense.

wth: The wall thickness size of the thin sheet metal forming the duct tee (position.p[1]).

wi1: The size of the rectangular duct tee base along the placement y-axis (position.p[2]).

wi2: The size of the rectangular duct tee end along the placement x (position.p[1]) at the lower y position.

wi3: The size of the rectangular duct tee end along the placement x (position.p[1]) at the upper y position.

hei: The size of the rectangular duct tee base along the placement z-axis (position.p[3]).

ra1: The rounding off radius at the lower y position.

ra2: The rounding off radius at the upper y position.

ch1: The depth of symmetrical chamfer at the lower y position.

ch2: The depth of symmetrical chamfer at the upper y position.

lo1: The lateral offset of the radius or chamfer at the lower y position along the placement x (position.p[1]).

lo2: The lateral offset of the radius or chamfer at the upper y position along the placement X (position.p[1]).

Formal propositions:

WR1: The wall thickness **wth** shall be greater than 0.

WR2: The width **wi1** shall be greater than 2 * **wth.**

WR3: The width **wi2** shall be greater than 2 * **wth.**

WR4: The width **wi3** shall be greater than 2 * **wth.**

WR5: The height **hei** shall be greater than 2 * **wth.**

WR6: The radius **ra1** shall be greater or equal to 0.

WR7: The radius **ra2** shall be greater or equal to 0.

WR8: The chamfer depth **ch1** shall be greater or equal to 0.

WR9: The chamfer depth **ch2** shall be greater or equal to 0.

WR10: The radius **ra1** shall be 0 if the chamfer depth **ch1** is greater than 0.

WR11: The radius **ra2** shall be 0 if the chamfer depth **ch2** is greater than 0.

WR12: The chamfer depth **ch1** shall be 0 if the radius **ra1** is greater than 0.

WR13: The chamfer depth **ch2** shall be 0 if the radius **ra2** is greater than 0.

WR14: The lateral offset **lo1** shall be greater or equal to 0.

WR15: The lateral offset **lo2** shall be greater or equal to 0.

WR16: The lateral offset **lo1** shall be 0 if the lateral offset **lo2** is greater than 0.

WR17: The lateral offset **lo2** shall be 0 if the lateral offset **lo1** is greater than 0.

Informal propositions:

IP1: The radii or chamfers can be given for one or both sides. The values of inlet chamfers determine the depth of 45° chamfers. No concurrent values may be given for radii and chamfers on the same side — either there is a radius or a chamfer.

IP2: There are no straight duct extensions: either **lo1** = 0 or **lo2** = 0. The cross-sections **wi2 x hei** and **wi3 x hei** lie at the end of the radii or 45° chamfers.

Specifications in IFC:

Duct tees are defined with the following IFC definitions.

XSD Specification (IFC):

```
<xs:element name="IfcRectangularDuctTee" type="ifc:IfcRectangularDuctTee"
substitutionGroup="ifc:IfcCsgPrimitive3D" nillable="true"/>
 <xs:complexType name="IfcRectangularDuctTee">
  <xs:complexContent>
   <xs:extension base="ifc:IfcCsgPrimitive3D">
    <xs:attribute name="wth" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="wi1" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="wi2" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="wi3" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="hei" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="ra1" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="ra2" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="ch1" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="ch2" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="lo1" type="ifc:IfcLengthMeasure" use="optional"/>
    <xs:attribute name="lo2" type="ifc:IfcLengthMeasure" use="optional"/>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
```

EXPRESS Specification (IFC):

```
ENTITY IfcRectangularDuctTee[4)]
 SUBTYPE OF IfcCsgPrimitive3D;
    wth: IfcPositiveLengthMeasure;
    wi1: IfcPositiveLengthMeasure;
    wi2: IfcPositiveLengthMeasure;
    wi3: IfcPositiveLengthMeasure;
    hei: IfcPositiveLengthMeasure;
    ra1: IfcPositiveLengthMeasure;
    ra2: IfcPositiveLengthMeasure;
    ch1: IfcPositiveLengthMeasure;
    ch2: IfcPositiveLengthMeasure;
    lo1: IfcPositiveLengthMeasure;[5)]
    lo2: IfcPositiveLengthMeasure;
END_ENTITY;
```

4) http://www.buildingsmart-tech.org/ifc/IFC4/final/html/schema/ifcgeometricmodelresource/lexical/ifcblock.htm

5) http://www.buildingsmart-tech.org/ifc/IFC4/final/html/schema/ifcmeasureresource/lexical/ifcpositivelengthmeasure.htm

A.8 Rectangular duct wye

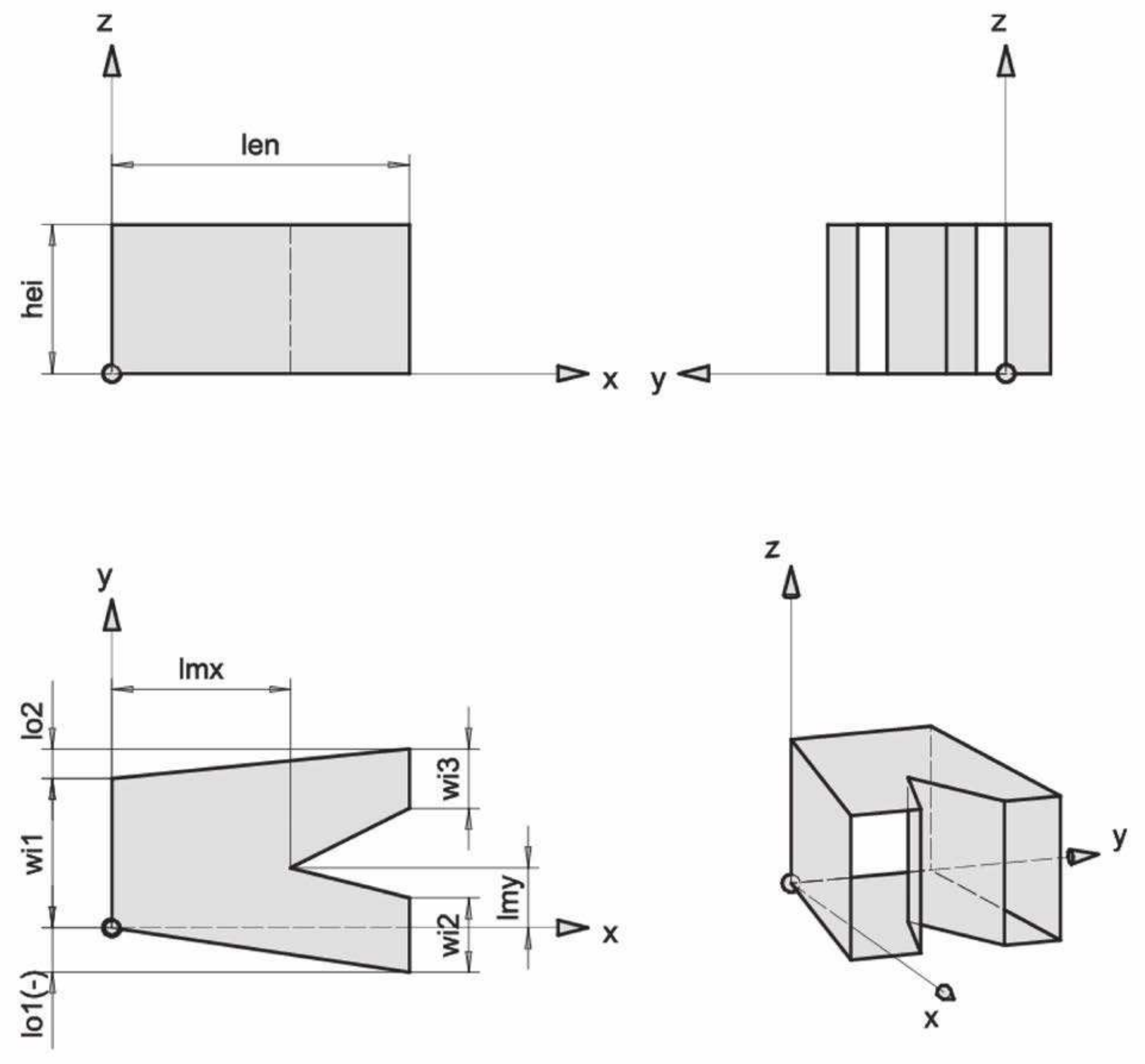

Figure A.7 — Rectangular duct wye

Primitive identifier: 'rectangular_duct_wye'

A rectangular duct wye is a type of regular CSG primitive with one rectangular base and two rectangular ends. Its behaviour in Boolean operations is the same as that of a solid primitive. The base and end surfaces can be displayed as invisible to perform a duct wye of thin sheet metal.

EXPRESS Specification:

```
*)
ENTITY rectangular_duct_wye
  SUBTYPE OF (geometric_representation_item);
    position: axis2_placement_3d;
    wth:      positive_length_measure;
    len:      positive_length_measure;
    wi1:      positive_length_measure;
    wi2:      positive_length_measure;
    wi3:      positive_length_measure;
    hei:      positive_length_measure;
    lmx:      positive_length_measure;
    lmy:      length_measure;
```

```
    lo1:      length_measure;
    lo2:      length_measure;
    WHERE
      WR1:    wth >= 0
      WR2:    len > 0
      WR3:    lmx >= 0
      WR4:    lmx <= len
      WR5:    lmy <= wi1 + lmx / len * lo2 - wth
      WR6:    lmy >= lmx / len * lo1 + wth
      WR7:    wi1 > 2 * wth
      WR8:    wi2 > 2 * wth
      WR9:    wi3 > 2 * wth
      WR10:   hei > 2 * wth
END_ENTITY;
(*
```

Attribute definitions:

position: The location and orientation of the placement axis system for the rectangular duct wye.

The **rectangular_duct_wye** has one vertex at **position.location** and its edges are aligned with the placement axes in the positive sense.

wth: The wall thickness size of the thin sheet metal forming the duct wye (position.p[1]).

wi1: The size of the rectangular duct wye base along the placement y-axis (position.p[2]).

wi2: The size of the rectangular duct wye end along the placement x-axis (position.p[1]) at the lower y position.

wi3: The size of the rectangular duct wye end along the placement x-axis (position.p[1]) at the upper y position.

hei: The size of the rectangular duct wye base along the placement z-axis (position.p[3]).

len The length size of the rectangular duct wye along the placement x-axis (position.p[1]).

lmx: The lateral offset of the wye branch edge along the placement x-axis (position.p[1]).

lmy: The lateral offset of the wye branch edge along the placement y-axis (position.p[2]).

lo1: The lateral offset of the wye branch at the lower y position along the placement y-axis (position.p[2]).

lo2: The lateral offset of the wye branch at the upper y position along the placement y-axis (position.p[2]).

Formal propositions:

WR1: The wall thickness **wth** shall be greater than 0.

WR2: The length **len** shall be greater than 0.

WR3: The length **lmx** shall be less or equal to **len.**

WR4: The length **lmy** shall be less or equal to **wi1 + lmx / len * lo2 - wth.**

WR5: The length **lmy** shall be greater or equal to **lmx / len * lo1 + wth.**

WR6: The width **wi1** shall be greater than 2 * **wth.**

WR7: The width **wi2** shall be greater than 2 * **wth.**

WR8: The width **wi3** shall be greater than 2 * **wth.**

WR9: The height **hei** shall be greater than 2 * **wth.**

Informal propositions:

IP1: The offsets may have positive or negative values referred to the placement y-axis.

IP2: Wyes with legs of different lengths can be created by attaching a **rectangle_duct_transition** primitive.

Specifications in IFC:

The rectangular duct wye is defined with the following IFC definitions.

XSD Specification (IFC):

```
<xs:element name="IfcRectangularDuctWye" type="ifc:IfcRectangularDuctWye"
substitutionGroup="ifc:IfcCsgPrimitive3D" nillable="true"/>
 <xs:complexType name="IfcRectangularDuctWye">
  <xs:complexContent>
   <xs:extension base="ifc:IfcCsgPrimitive3D">
    <xs:attribute name="wth" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="len" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="wi1" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="wi2" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="wi3" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="hei" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="lmx" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="lmy" type="ifc:IfcLengthMeasure" use="optional"/>
    <xs:attribute name="lo1" type="ifc:IfcLengthMeasure" use="optional"/>
    <xs:attribute name="lo2" type="ifc:IfcLengthMeasure" use="optional"/>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
```

EXPRESS Specification (IFC):

```
ENTITY IfcRectangularDuctWye[6]
 SUBTYPE OF IfcCsgPrimitive3D;
    wth: IfcPositiveLengthMeasure;
    len: IfcPositiveLengthMeasure;
    wi1: IfcPositiveLengthMeasure;
    wi2: IfcPositiveLengthMeasure;
    wi3: IfcPositiveLengthMeasure;
    hei: IfcPositiveLengthMeasure;
    lmx: IfcPositiveLengthMeasure;
    lmy: IfcLengthMeasure;
    lo1: IfcLengthMeasure;
    lo2: IfcLengthMeasure;
END_ENTITY;
```

6) http://www.buildingsmart-tech.org/ifc/IFC4/final/html/schema/ifcgeometricmodelresource/lexical/ifcblock.htm

A.9 Rectangle/oval transition

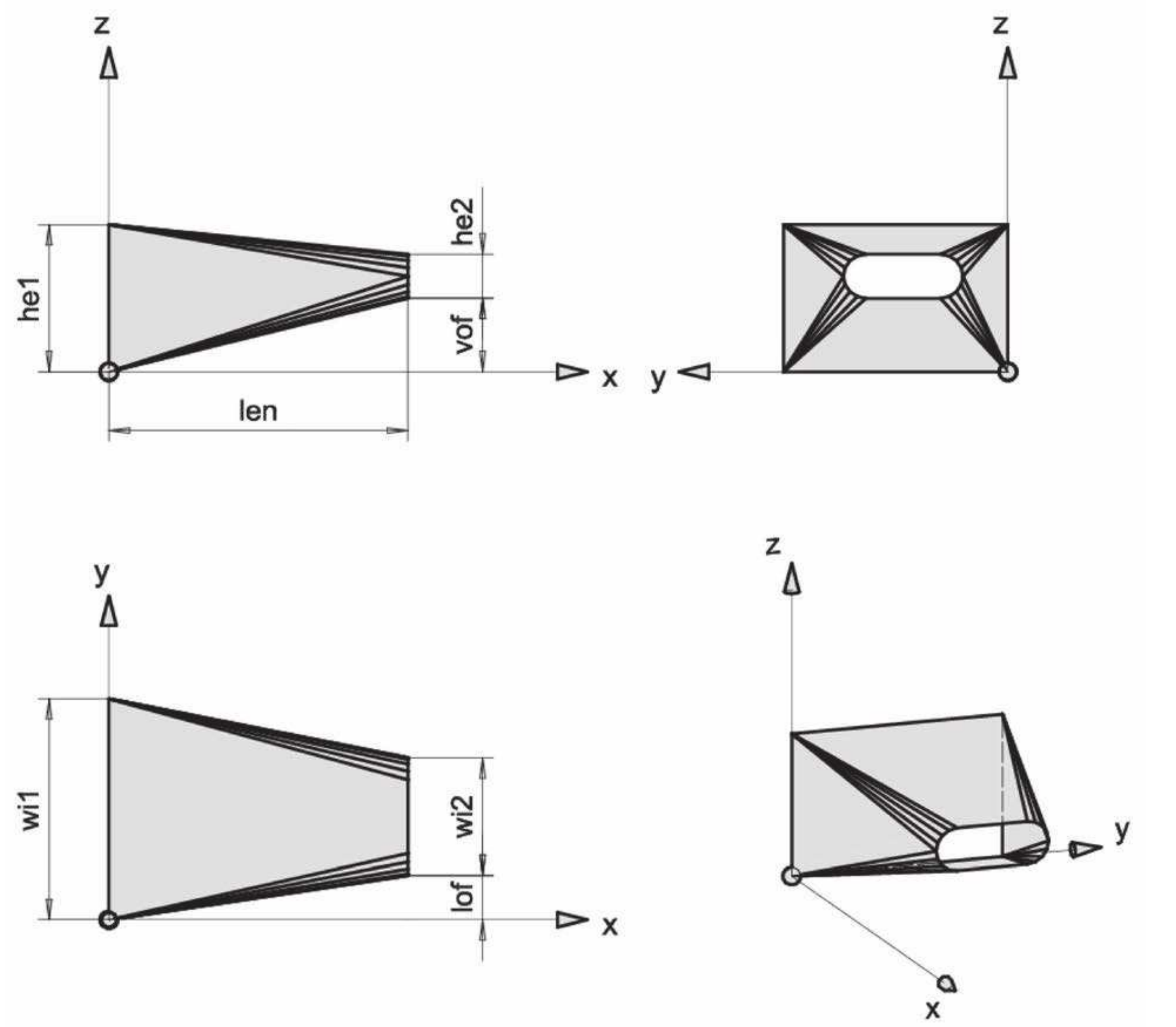

Figure A.8 — Rectangle/oval transition

Primitive identifier: 'rectangle_oval_transition'

A rectangle oval transition is a type of regular CSG primitive with rectangular base and oval end. Its behaviour in Boolean operations is the same as that of a solid primitive. The base and end surface can be displayed as invisible to perform a rectangle/oval transition of thin sheet metal.

EXPRESS Specification:

```
*)
ENTITY rectangle_oval_transition
  SUBTYPE OF (geometric_representation_item);
    position: axis2_placement_3d;
    wth:      positive_length_measure;
    len:      positive_length_measure;
    wi1:      positive_length_measure;
    wi2:      positive_length_measure;
    he1:      positive_length_measure;
    he2:      positive_length_measure;
    lof:      length_measure;
    vof:      length_measure;
```

```
    WHERE
      WR1:    wth >= 0
      WR2:    len > 0
      WR3:    wi1 > 2*wth
      WR4:    wi2 > 2*wth
      WR5:    he1 > 2*wth
      WR6:    he2 > 2*wth
END_ENTITY;
(*
```

Attribute definitions:

position: The location and orientation of the placement axis system for the rectangle oval transition.

The rectangle_oval_transition has one vertex at **position.location** and its edges are aligned with the placement axes in the positive sense.

wth: The wall thickness size of the thin sheet metal forming the rectangle/oval transition (position.p[1]).

len: The length size of the rectangular oval transition along the placement x-axis (position.p[1]).

wi1: The width size of the rectangular oval transition base along the placement y-axis (position.p[2]).

wi2: The width size of the rectangle oval transition end along the placement y-axis (position.p[2]).

he1: The height size of the rectangle oval transition base along the placement z-axis (position.p[3]).

he2: The height size of the rectangle oval transition end along the placement z-axis (position.p[3]).

lof: The lateral offset of the rectangle oval transition end along the placement y-axis (position.p[2]).

vof: The vertical offset of the rectangle oval transition end along the placement z-axis (position.p[3]).

Formal propositions:

WR1: The wall thickness **wth** shall be greater than 0.

WR2: The length **len** shall be greater than 0.

WR3: The width **wi1** shall be greater than 2 * **wth.**

WR4: The width **wi2** shall be greater than 2 * **wth.**

WR5: The height **he1** shall be greater than 2 * **wth.**

WR6: The height **he2** shall be greater than 2 * **wth.**

Informal propositions:

IP1: Vertical and lateral offsets can be entered as positive or negative values referred to the placement y-axis or z-axis.

IP2: The round walls of the oval cross-section lie in whichever is the smaller of the two cross-sectional dimensions (**hei2** or **wi2**).

Specifications in IFC:

The rectangle oval transition is defined with the following IFC definitions.

XSD Specification (IFC):

```
<xs:element name="IfcRectangleOvalTransition" type="ifc:IfcRectangleOvalTransition"
substitutionGroup="ifc:IfcCsgPrimitive3D" nillable="true"/>
 <xs:complexType name="IfcRectangleOvalTransition">
  <xs:complexContent>
   <xs:extension base="ifc:IfcCsgPrimitive3D">
    <xs:attribute name="wth" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="len" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="wi1" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="wi2" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="he1" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="he2" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="lof" type="ifc:IfcLengthMeasure" use="optional"/>
    <xs:attribute name="vof" type="ifc:IfcLengthMeasure" use="optional"/>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
```

EXPRESS Specification (IFC):

```
ENTITY IfcRectangleOvalTransition7)
 SUBTYPE OF IfcCsgPrimitive3D;
    wth: IfcPositiveLengthMeasure;
    len: IfcPositiveLengthMeasure;
    wi1: IfcPositiveLengthMeasure;
    wi2: IfcPositiveLengthMeasure;
    he1: IfcPositiveLengthMeasure;
    he2: IfcPositiveLengthMeasure;
    lof: IfcLengthMeasure;
    vof: IfcLengthMeasure;
END_ENTITY;
```

7) http://www.buildingsmart-tech.org/ifc/IFC4/final/html/schema/ifcgeometricmodelresource/lexical/ifcblock.htm

A.10 Rectangle/round transition

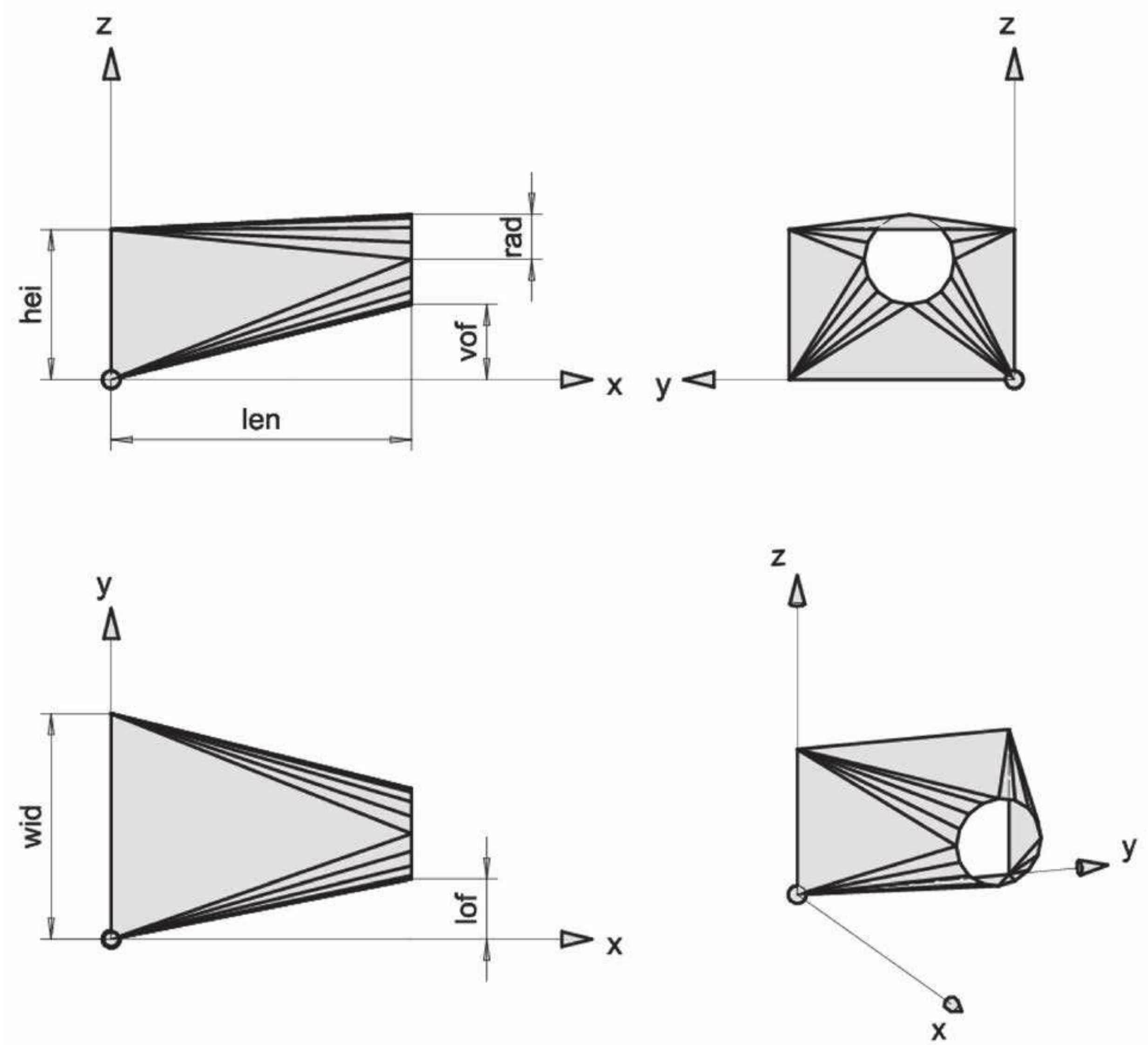

Figure A.9 — Rectangle/round transition

Primitive identifier: 'rectangle_round_transition'

A rectangular/round transition is a type of regular CSG primitive with rectangular base and round end. Its behaviour in Boolean operations is the same as that of a solid primitive. The base and end surface can be displayed as invisible to perform a rectangle/round transition of thin sheet metal.

EXPRESS Specification:

```
*)
ENTITY rectangle_round_transition
  SUBTYPE OF (geometric_representation_item);
    position: axis2_placement_3d;
    wth:      positive_length_measure;
    len:      positive_length_measure;
    wid:      positive_length_measure;
    hei:      positive_length_measure;
    rad:      positive_length_measure;
    lof:      length_measure;
    vof:      length_measure;
    WHERE
```



```
      WR1:    wth >= 0
      WR2:    len > 0
      WR3:    wid > 2*wth
      WR4:    hei > 2*wth
      WR5:    rad > wth
END_ENTITY;
(*
```

Attribute definitions:

position: The location and orientation of the placement axis system for the rectangle round transition.

The **rectangle_round_transition** has one vertex at **position.location** and its edges are aligned with the placement axes in the positive sense.

wth: The wall thickness size of the thin sheet metal forming the rectangle round transition (position.p[1]).

len: The length size of the rectangle_round_transition along the placement x-axis (position.p[1]).

wid: The width size of the rectangular transition base along the placement y-axis (position.p[2]).

hei: The height size of the rectangular transition base along the placement z-axis (position.p[3]).

rad: The radius of the round transition end along the placement y-axis (position.p[2]).

lof: The lateral offset of the round transition end along the placement y-axis (position.p[2]).

vof: The vertical offset of the round transition end along the placement z-axis (position.p[3]).

Formal propositions:

WR1: The wall thickness **wth** shall be greater than 0.

WR2: The length **len** shall be greater than 0.

WR3: The width **wid** shall be greater than 2 * **wth.**

WR4: The height **hei** shall be greater than 2 * **wth.**

WR5: The radius **rad** shall be greater than **wth.**

Informal propositions:

IP1: Vertical and lateral offsets can be entered as positive or negative values referred to the placement y axis or z axis.

Specifications in IFC:

The rectangle round transition is defined with the following IFC definitions.

XSD Specification (IFC):

```
<xs:element name="IfcRectangleRoundTransition" type="ifc:IfcRectangleRoundTransition"
substitutionGroup="ifc:IfcCsgPrimitive3D" nillable="true"/>
 <xs:complexType name="IfcRectangleRoundTransition">
  <xs:complexContent>
   <xs:extension base="ifc:IfcCsgPrimitive3D">
    <xs:attribute name="wth" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="len" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
```

```
    <xs:attribute name="wid" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="hei" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="rad" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="lof" type="ifc:IfcLengthMeasure" use="optional"/>
    <xs:attribute name="vof" type="ifc:IfcLengthMeasure" use="optional"/>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
```

EXPRESS Specification (IFC):

```
ENTITY IfcRectangleRoundTransition8)
 SUBTYPE OF IfcCsgPrimitive3D;
    wth: IfcPositiveLengthMeasure;
    len: IfcPositiveLengthMeasure;
    wid: IfcPositiveLengthMeasure;
    hei: IfcPositiveLengthMeasure;
    rad: IfcPositiveLengthMeasure;
    lof: IfcLengthMeasure;
    vof: IfcLengthMeasure;
END_ENTITY;
```

8) http://www.buildingsmart-tech.org/ifc/IFC4/final/html/schema/ifcgeometricmodelresource/lexical/ifcblock.htm

A.11 Trapezoidal duct

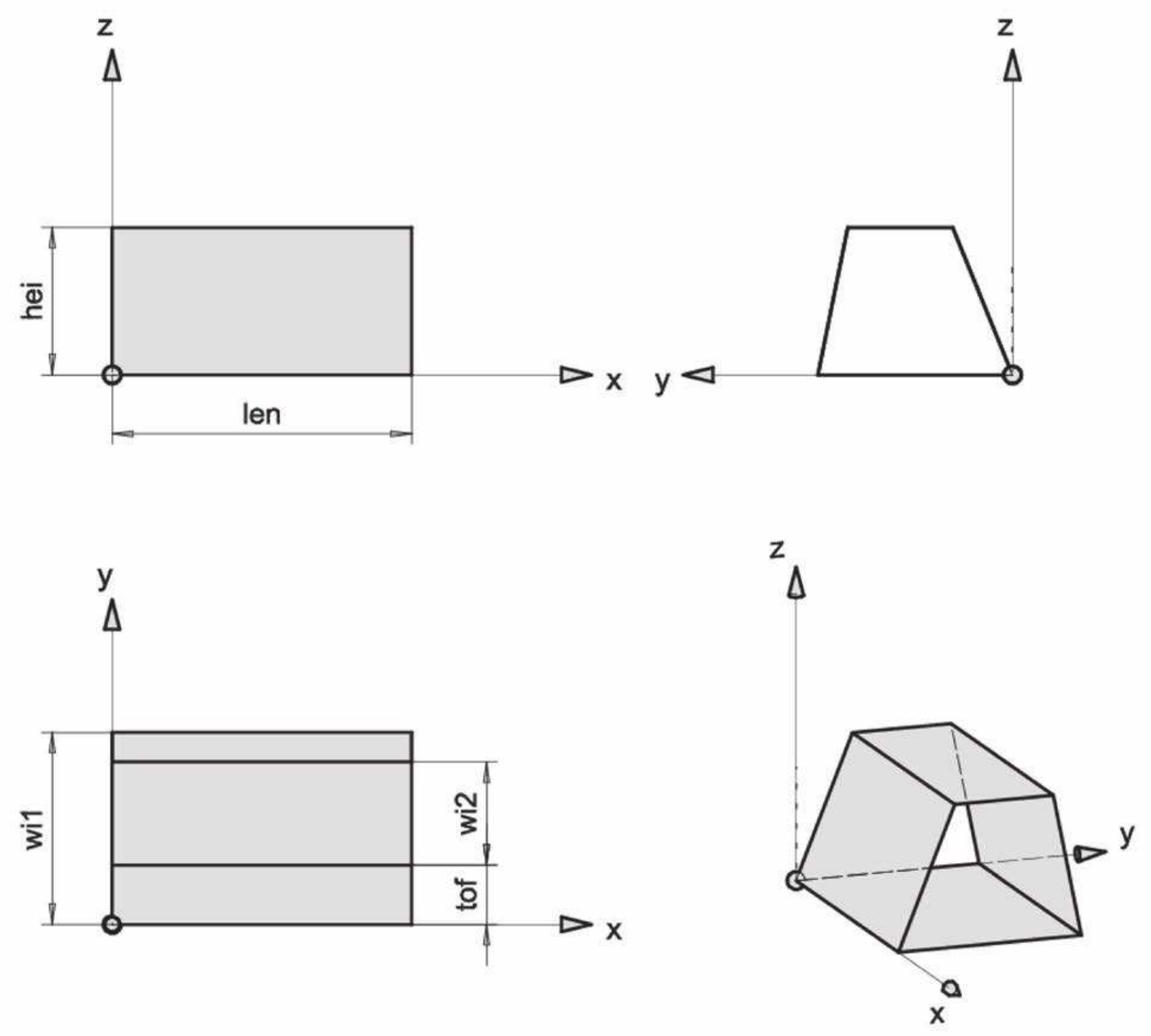

Figure A.10 — Trapezoidal duct

Primitive identifier: 'trapezoidal_duct'

A trapezoidal duct is a type of regular prismatic CSG primitive with trapezoidal base and end. Its behaviour in Boolean operations is the same as that of a solid primitive. The base and end surface can be displayed as invisible to perform a duct of thin sheet metal.

EXPRESS Specification:

```
*)
ENTITY trapezoidal_duct
  SUBTYPE OF (geometric_representation_item);
    position: axis2_placement_3d;
    wth:      positive_length_measure;
    len:      positive_length_measure;
    wi1:      positive_length_measure;
    wi2:      positive_length_measure;
    hei:      positive_length_measure;
    tof:      length_measure;
    WHERE
      WR1:    wth >= 0
      WR2:    len > 0
      WR3:    wi1 > wth*2
      WR4:    wi2 >= 0
      WR5:    hei > wth*2
END_ENTITY;
(*
```

Attribute definitions:

position: The location and orientation of the placement axis system for the trapezoidal duct.

The **trapezoidal_duct** has one vertex at **position.location** and its edges are aligned with the placement axes in the positive sense.

wth: The wall thickness size of the thin sheet metal forming the duct (position.p[1]).

len: The length size of the trapezoidal duct along the placement x-axis (position.p[1]).

wi1: The width size on the lower z side of the trapezoidal duct along the placement y-axis (position.p[2]).

wi2: The width size on the upper z side of the trapezoidal duct along the placement y-axis (position.p[2]).

hei: The height size of the trapezoidal duct along the placement z-axis (position.p[3]).

tof: The lateral offset of the upper z side of the trapezoidal duct along the placement y-axis (position.p[2]).

Formal propositions:

WR1: The wall thickness **wth** shall be greater than 0.

WR2: The length **len** shall be greater than 0.

WR3: The width **wi1** shall be greater than 2 * **wth.**

WR4: The width **wi2** shall be greater or equal to 0.

WR5: The height **hei** shall be greater than 2 * **wth.**

Informal propositions:

IP1: The offset **tof** can be entered as positive or negative value referred to the placement y-axis.

IP2: The value of the width wi2 can also become zero, in which case the trapezoidal duct is triangular.

Specifications in IFC:

The trapezoidal duct is defined with the existing definition: **IfcExtrudedAreaSolid.**

The base face is defined as parametrizable 2D profile.

XSD Specification of the 2D profile (IFC):

XSD Specification:

```
<xs:element name="IfcTrapeziumHollowProfileDef" type="ifc:IfcTrapeziumHollowProfileDef"
substitutionGroup="ifc:IfcParametrizedProfileDef" nillable="true"/>
 <xs:complexType name="IfcTrapeziumHollowProfileDef">
  <xs:complexContent>
   <xs:extension base="ifc:IfcParametrizedProfileDef">
    <xs:attribute name="wth" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="wi1" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="wi2" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="hei" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="tof" type="ifc:IfcLengthMeasure" use="optional"/>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
```

EXPRESS Specification:

```
ENTITY IfcTrapeziumProfileDef
 SUBTYPE OF IfcParametrizedProfileDef;
    wth: IfcPositiveLengthMeasure;
    wi1: IfcPositiveLengthMeasure;
    wi2: IfcPositiveLengthMeasure;
    hei: IfcPositiveLengthMeasure;
    tof: IfcLengthMeasure;
END_ENTITY;
```


A.12 Trapezoidal duct transition

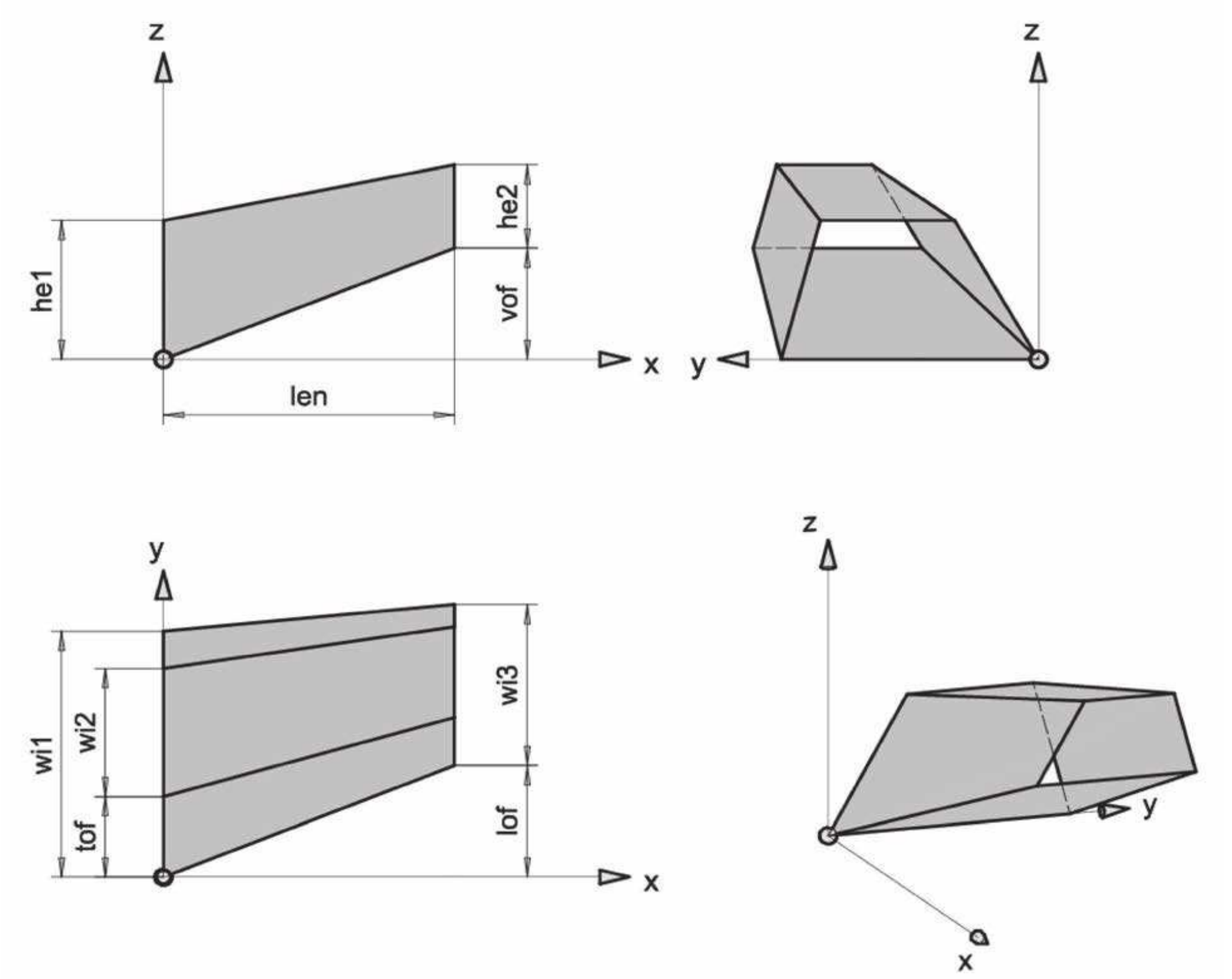

Figure A.11 — Trapezoidal duct transition

Primitive identifier: 'trapezoidal_duct_transition'

A trapezoidal duct transition is a type of regular CSG primitive with trapezoidal base and end. Its behaviour in Boolean operations is the same as that of a solid primitive. The base and end surface can be displayed as invisible to perform a duct transition of thin sheet metal.

EXPRESS Specification:

```
*)
ENTITY trapezoidal_duct_transition
  SUBTYPE OF (geometric_representation_item);
    position: axis2_placement_3d;
    wth:      positive_length_measure;
    len:      positive_length_measure;
    wi1:      positive_length_measure;
    wi2:      positive_length_measure;
    wi3:      positive_length_measure;
    he1:      positive_length_measure;
    he2:      positive_length_measure;
    lof:      length_measure;
    vof:      length_measure;
    WHERE
      WR1:    wth >= 0
      WR2:    len > 0
      WR3:    wi1 > 2*wth
```

```
      WR4:    wi2 >= 0
      WR5:    wi3 > 2*wth
      WR6:    wi1 <= wi2 + wi3 * he1 / he2
      WR7:    wi2 >= wi1 - wi3 * he1 / he2
      WR8:    wi3 >= (wi1 – wi2) * he2 / he1
      WR9:    he1 >= he2 * (wi1 - wi2) / wi3
      WR10:   he2 <= he1 * wi3 / (wi1 - wi2)
      WR11:   he1 > 2*wth
      WR12:   he2 > 2*wth
END_ENTITY;
(*
```

Attribute definitions:

position: The location and orientation of the placement axis system for the trapezoidal duct transition.

The trapezoidal_duct_transition has one vertex at **position.location** and its edges are aligned with the placement axes in the positive sense.

wth: The wall thickness size of the thin sheet metal forming the trapezoidal duct transition (position.p[1]).

len: The length size of the trapezoidal duct transition along the placement x-axis (position.p[1]).

wi1: The width size on the lower z side of the trapezoidal duct transition base along the placement y-axis (position.p[2]).

wi2: The width size on the upper z side of the trapezoidal duct transition base along the placement y-axis (position.p[2]).

wi3: The width size on the lower z side of the trapezoidal duct transition end along the placement y-axis (position.p[2]).

he1: The height size of the trapezoidal duct transition base along the placement z-axis (position.p[3]).

he2: The height size of the trapezoidal duct transition end along the placement z-axis (position.p[3]).

tof: The lateral offset of the upper z side of the trapezoidal transition along the placement y-axis (position.p[2]).

lof: The lateral offset of the trapezoidal duct transition end along the placement y axis (position.p[2]).

vof: The vertical offset of the trapezoidal duct transition end along the placement z-axis (position.p[3]).

Formal propositions:

WR1: The wall thickness **wth** shall be greater than 0.

WR2: The length **len** shall be greater than 0.

WR3: The width **wi1** shall be greater than 2 * **wth.**

WR4: The width **wi2** shall be greater or equal to 0.

WR5: The width **wi3** shall be greater than 2 * **wth.**

WR6-10: The inequalities in WR6–WR10 are all transformations of one single inequality which ensures, that the two lateral surfaces do not intersect.

WR11: The height **he1** shall be greater than 2 * **wth.**

WR12: The height **he2** shall be greater than 2 * **wth**

Informal propositions:

IP1: Vertical and lateral offsets can be entered as positive or negative values referred to the placement y-axis or z-axis.

Specifications in IFC:

The trapezoidal duct transition is defined with existing IFC definitions.

The trapezoidal_duct_transition is defined as:	**IfcExtrudedAreaSolid**
The base face is the parametrizable 2D profile:	**IfcTrapeziumHollowProfileDef (see A.10)**
Included is the transformation of the start face by:	**IfcDerivedProfileDef**
	IfcCartesianTransformationOperator2D

A.13 Oval duct

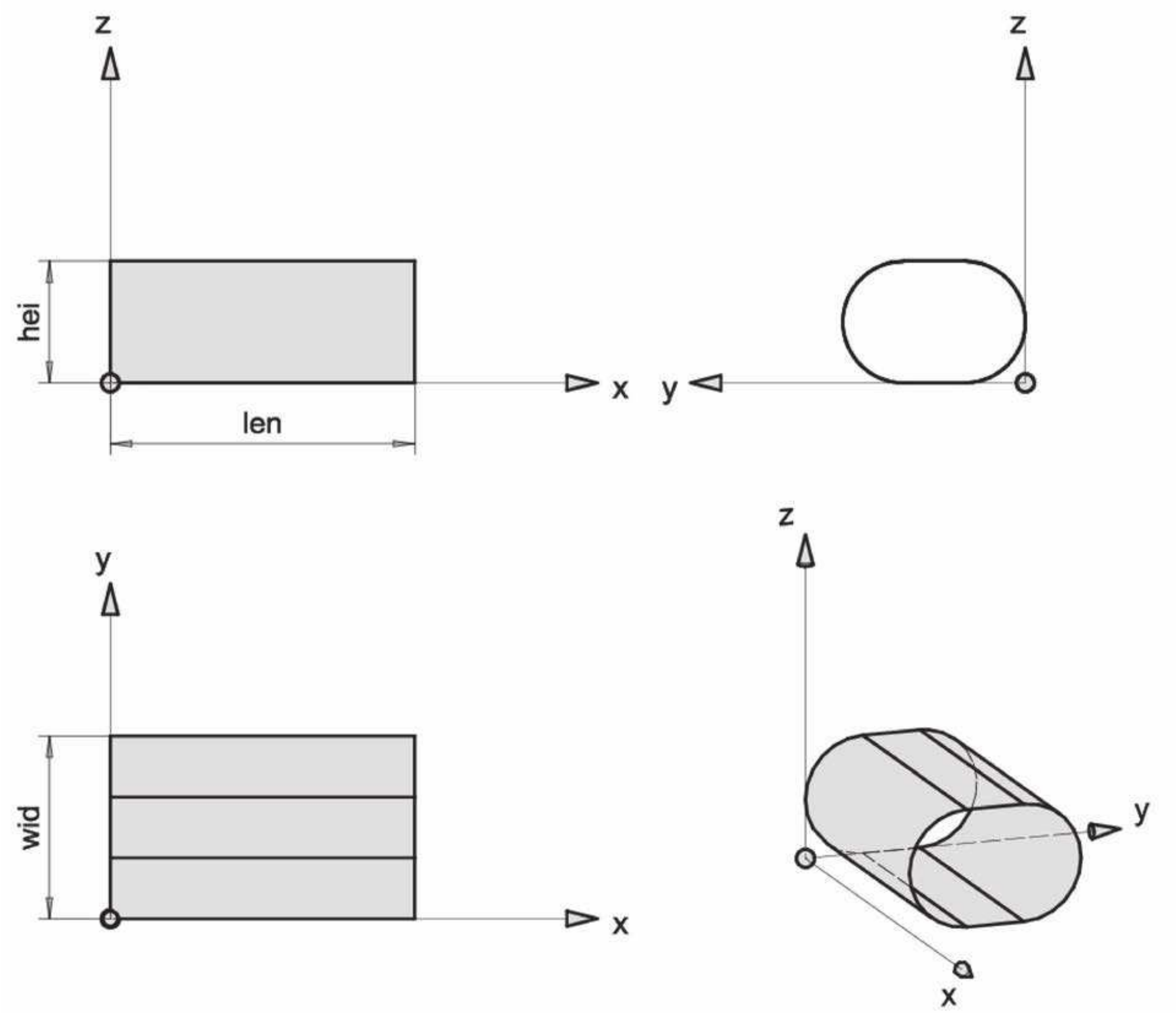

Figure A.12 — Oval duct

Primitive identifier: ‘oval_duct’

An oval duct is a type of regular prismatic CSG primitive with oval base. Its behaviour in Boolean operations is the same as that of a solid primitive. The base and end surface can be displayed as invisible to perform a duct of thin sheet metal.

EXPRESS Specification:

```
*)
ENTITY oval_duct
  SUBTYPE OF (geometric_representation_item);
    position: axis2_placement_3d;
    wth:      positive_length_measure;
    len:      positive_length_measure;
    wid:      positive_length_measure;
    hei:      positive_length_measure;
    WHERE
      WR1:    wth > 0
      WR2:    len > 0
      WR3:    wid > 2*wth
      WR4:    hei > 2*wth
END_ENTITY;
(*
```

Attribute definitions:

position: The location and orientation of the placement axis system for the oval duct.

The **oval_duct** has one vertex at **position.location** and its edges are aligned with the placement axes in the positive sense.

wth: The wall thickness size of the thin sheet metal forming the oval duct (position.p[1]).

len: The length size of the oval duct along the placement x-axis (position.p[1]).

wid: The width size of the oval duct along the placement y-axis (position.p[2]).

hei: The height size of the oval duct along the placement z-axis (position.p[3]).

Formal propositions:

WR1: The wall thickness **wth** shall be greater than 0.

WR2: The length **len** shall be greater than 0.

WR3: The width **wid** shall be greater than 2 * **wth.**

WR4: The height **hei** shall be greater than 2 * **wth.**

Informal propositions:

IP1: The orientation of the oval is determined by dimension **hei** or **wid**, whichever has the greater absolute value.

|**hei**| > |**wid**| → vertical oval

|**hei**| < |**wid**| → lateral oval

Specifications in IFC:

The oval duct is defined with the existing definition: **IfcExtrudedAreaSolid.**

The base face is defined as parametrizable 2D profile.

XSD Specification of the 2D profile (IFC):

XSD Specification:

```
<xs:element name="IfcOvalHollowProfileDef" type="ifc:IfcOvalHollowProfileDef"
substitutionGroup="ifc:IfcParametrizedProfileDef" nillable="true"/>
 <xs:complexType name="IfcOvalHollowProfileDef">
  <xs:complexContent>
   <xs:extension base="ifc:IfcParametrizedProfileDef">
    <xs:attribute name="wth" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="wid" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="hei" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
```

EXPRESS Specification:

```
ENTITY IfcTrapeziumProfileDef
 SUBTYPE OF IfcParametrizedProfileDef;
    wth: IfcPositiveLengthMeasure;
    wid: IfcPositiveLengthMeasure;
    hei: IfcPositiveLengthMeasure;
END_ENTITY;
```

A.14 Oval duct transition

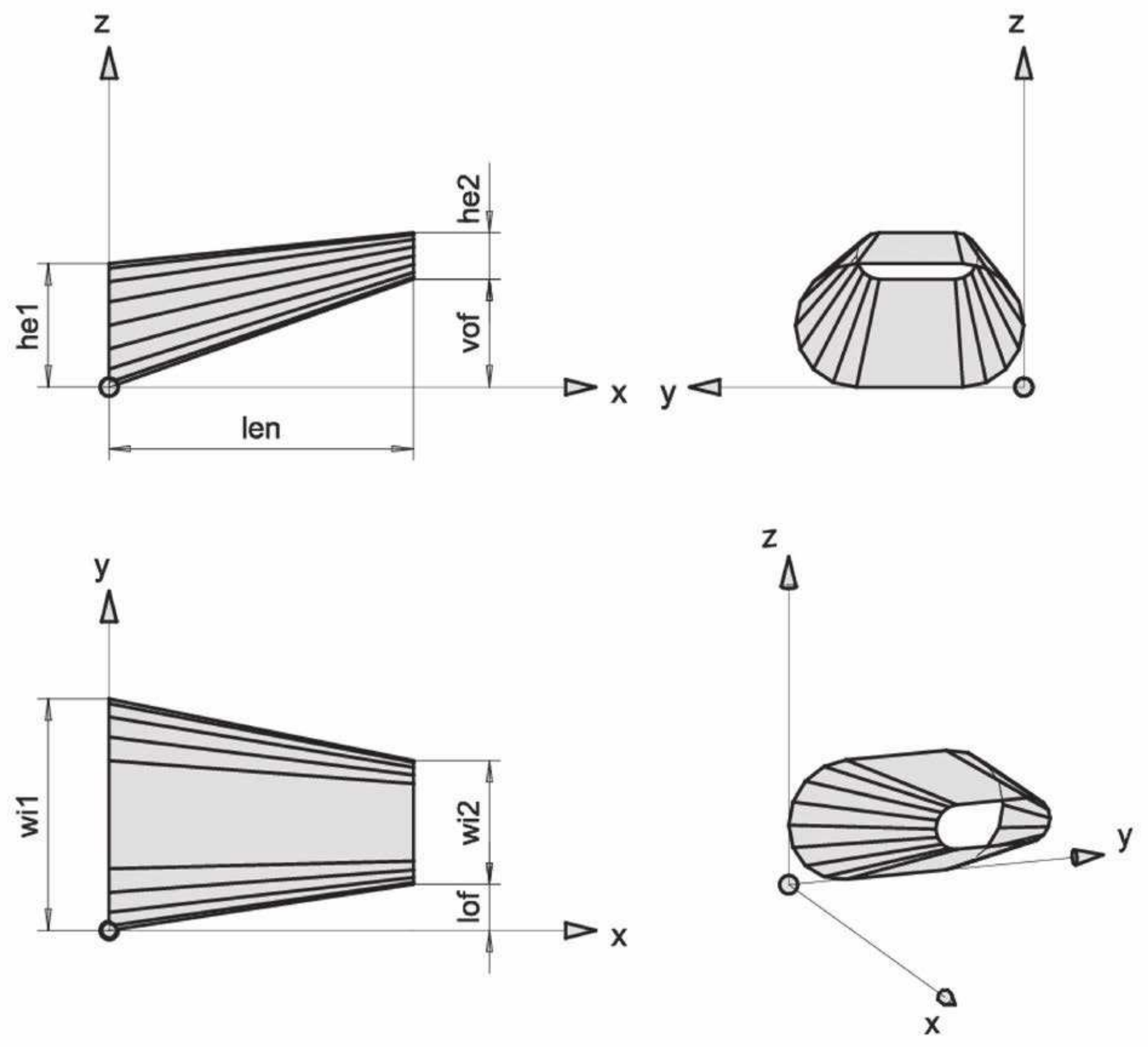

Figure A.13 — Oval duct transition

Primitive identifier: 'oval_duct_transition'

An oval duct transition is a type of regular CSG primitive with oval base and end. Its behaviour in Boolean operations is the same as that of a solid primitive. The base and end surface can be displayed as invisible to perform a duct transition of thin sheet metal.

EXPRESS Specification:

```
*)
ENTITY oval_duct_transition
  SUBTYPE OF (geometric_representation_item);
    position: axis2_placement_3d;
    wth:      positive_length_measure;
    len:      positive_length_measure;
    wi1:      positive_length_measure;
    wi2:      positive_length_measure;
    he1:      positive_length_measure;
    he2:      positive_length_measure;
    lof:      length_measure;
    vof:      length_measure;
    WHERE
      WR1:    wth >= 0
      WR2:    len > 0
      WR3:    wi1 > 2*wth
      WR4:    wi2 > 2*wth
      WR5:    he1 > 2*wth
      WR6:    he2 > 2*wth
END_ENTITY;
(*
```

Attribute definitions:

position: The location and orientation of the placement axis system for the oval duct transition.

The **oval_duct_transition** has one vertex at **position.location** and its edges are aligned with the placement axes in the positive sense.

wth: The wall thickness size of the thin sheet metal forming the duct transition (position.p[1]).

len: The length size of the oval duct transition along the placement x-axis (position.p[1]).

wi1: The width size of the oval duct transition base along the placement y-axis (position.p[2]).

wi2: The width size of the oval duct transition end along the placement y-axis (position.p[2]).

he1: The height size of the oval duct transition base along the placement z-axis (position.p[3]).

he2: The height size of the oval duct transition end along the placement z-axis (position.p[3]).

lof: The lateral offset of the oval duct transition end along the placement y-axis (position.p[2]).

vof: The vertical offset of the oval duct transition end along the placement z-axis (position.p[3]).

Formal propositions:

WR1: The wall thickness **wth** shall be greater than 0.

WR2: The length **len** shall be greater than 0.

WR3: The width **wi1** shall be greater than 2 * **wth.**

WR4: The width **wi2** shall be greater than 2 * **wth.**

WR5: The height **he1** shall be greater than 2 * **wth.**

WR6: The height **he2** shall be greater than 2 * **wth.**

Informal propositions:

IP1: Vertical and lateral offsets can be entered as positive or negative values referred to the placement y-axis or z-axis.

IP2: The orientations of the ovals are determined by whichever has the greater absolute value of the dimensions **he1** or **wi1** and **he2** or **wi2**, respectively.

Specifications in IFC:

The trapezoidal duct transition is defined with existing IFC definitions.

The trapezoidal_duct_transition is defined as:	**IfcExtrudedAreaSolid**
The base face is the parametrizable 2D profile:	**IfcOvalHollowProfileDef (see A.12)**
Included is the transformation of the start face by:	**IfcDerivedProfileDef**
	IfcCartesianTransformationOperator2D

A.15 Oval duct bend

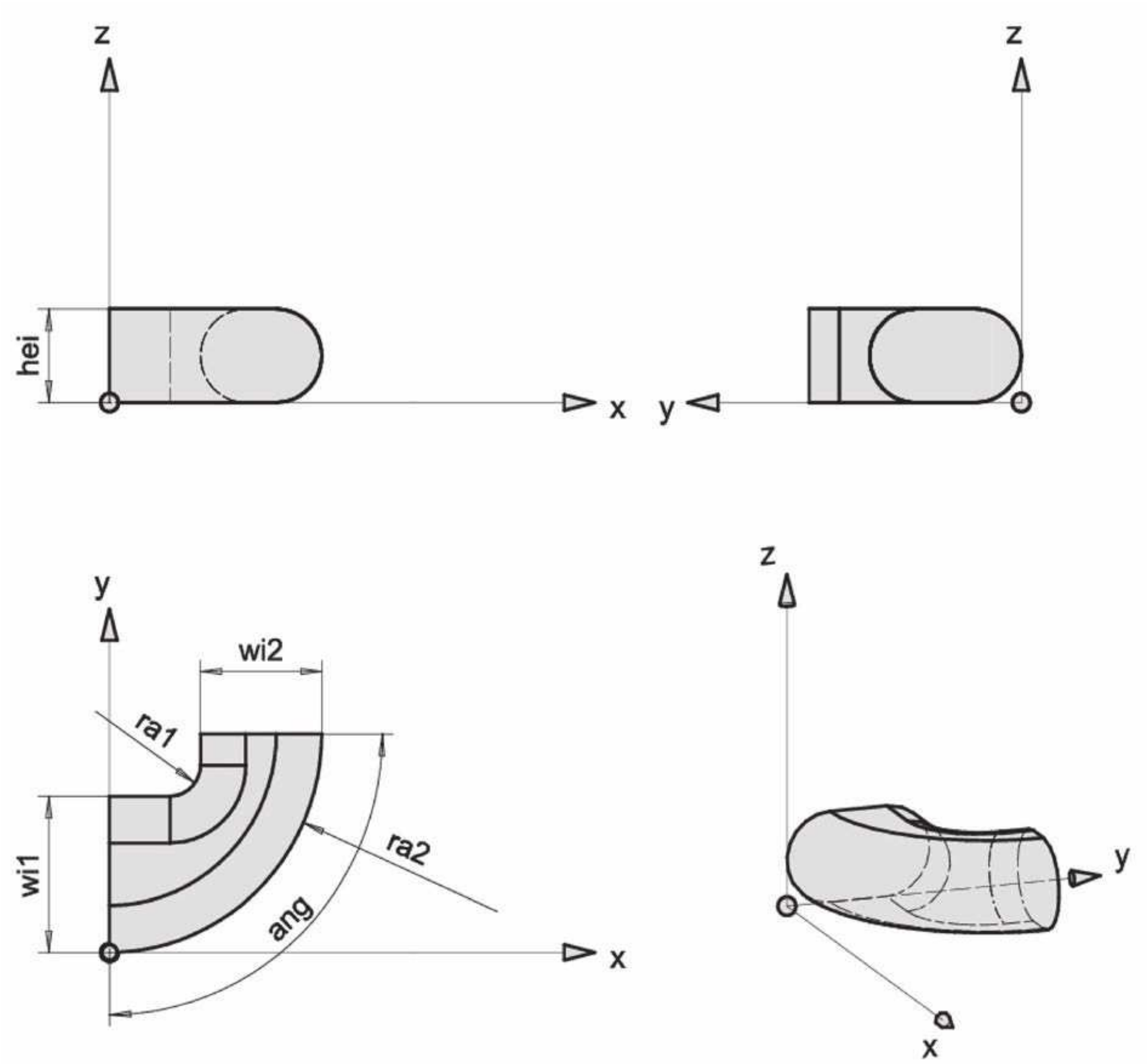

Figure A.14 — Oval duct bend

Primitive identifier: 'oval_duct_bend'

An oval duct bend is a type of regular CSG primitive with oval base and end. Its behaviour in Boolean operations is the same as that of a solid primitive. The base and end surface can be displayed as invisible to perform a duct bend of thin sheet metal.

EXPRESS Specification:

```
*)
ENTITY oval_duct_bend
  SUBTYPE OF (geometric_representation_item);
    position: axis2_placement_3d;
    wth:      positive_length_measure;
    ang:      positive_plane_angle_measure;
    wi1:      positive_length_measure;
    wi2:      positive_length_measure;
    hei:      positive_length_measure;
    ra1:      positive_length_measure;
    ra2:      positive_length_measure;
    WHERE
```

```
      WR1:    wth >= 0
      WR2:    len > 0
      WR3:    ang > 0
      WR4:    ang < 360
      WR5:    wi1 > 2*wth
      WR6:    wi2 > 2*wth
      WR7:    hei > 2*wth
      WR8:    ra1 >= 0
      WR9:    ra2 > ra1+2*wth
END_ENTITY;
(*
```

Attribute definitions:

position: The location and orientation of the placement axis system for the oval duct bend.

The **oval_duct_bend** has one vertex at **position.location** and its edges are aligned with the placement axes in the positive sense.

wth: The wall thickness size of the thin sheet metal forming the duct bend (position.p[1]).

ang: The turn angle between the planes of the two oval faces of the solid, measured in the sector containing the solid.

wi1: The size of the oval duct bend base along the placement y-axis (position.p[2]).

wi2: The size of the oval duct bend end in the x-y plane in the direction of TA.

hei: The size of the oval duct bend base along the placement z-axis (position.p[3]).

ra1: The rounding off radius on the inner side of the duct bend.

ra2: The rounding off radius on the outer side of the duct bend.

Formal propositions:

WR1: The wall thickness **wth** shall be greater than 0.

WR2: The angle **ang** shall be greater than 0.

WR3: The angle **ang** shall be less or equal to 360.

WR4: The width **wi1** shall be greater than 2 * **wth.**

WR5: The width **wi2** shall be greater than 2 * **wth.**

WR6: The height **hei** shall be greater than 2 * **wth.**

WR7: The radius **ra1** shall be greater than 0.

WR8: The radius **ra2** shall be greater than **ra1** + 2***wth.**

Informal propositions:

IP1: With aperture angles between 180° and 360°, any value given for **ra2** will be ignored. **ra2** is calculated as the smallest tangent circle on the outer walls of the duct bend.

IP2: The cross-section **wi2 × hei** is located at the end of the circle bend having the radius **ra2** or **ra1.**

IP3: The orientations of the ovals are determined by whichever has the greater absolute value of the dimensions **hei** or **wi1** and **hei** or **wi2**, respectively.

Specifications in IFC:

The oval duct bend is defined with the following IFC definitions.

XSD Specification (IFC):

```
<xs:element name="IfcOvalDuctBend" type="ifc:OvalDuctBend"
substitutionGroup="ifc:IfcCsgPrimitive3D" nillable="true"/>
 <xs:complexType name="IfcOvalDuctBend">
  <xs:complexContent>
   <xs:extension base="ifc:IfcCsgPrimitive3D">
    <xs:attribute name="wth" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="ang" type="ifc:IfcPlaneAngleMeasure" use="optional"/>
    <xs:attribute name="wi1" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="wi2" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="hei" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="ra1" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="ra2" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
```

EXPRESS Specification (IFC):

```
ENTITY IfcOvalDuctBend
 SUBTYPE OF IfcCsgPrimitive3D;
    wth: IfcPositiveLengthMeasure;
    ang: IfcIfcPlaneAngleMeasure;
    wi1: IfcPositiveLengthMeasure;
    wi2: IfcPositiveLengthMeasure;
    hei: IfcPositiveLengthMeasure;
    ra1: IfcPositiveLengthMeasure;
    ra2: IfcPositiveLengthMeasure;
END_ENTITY;
```

A.16 Oval/round transition

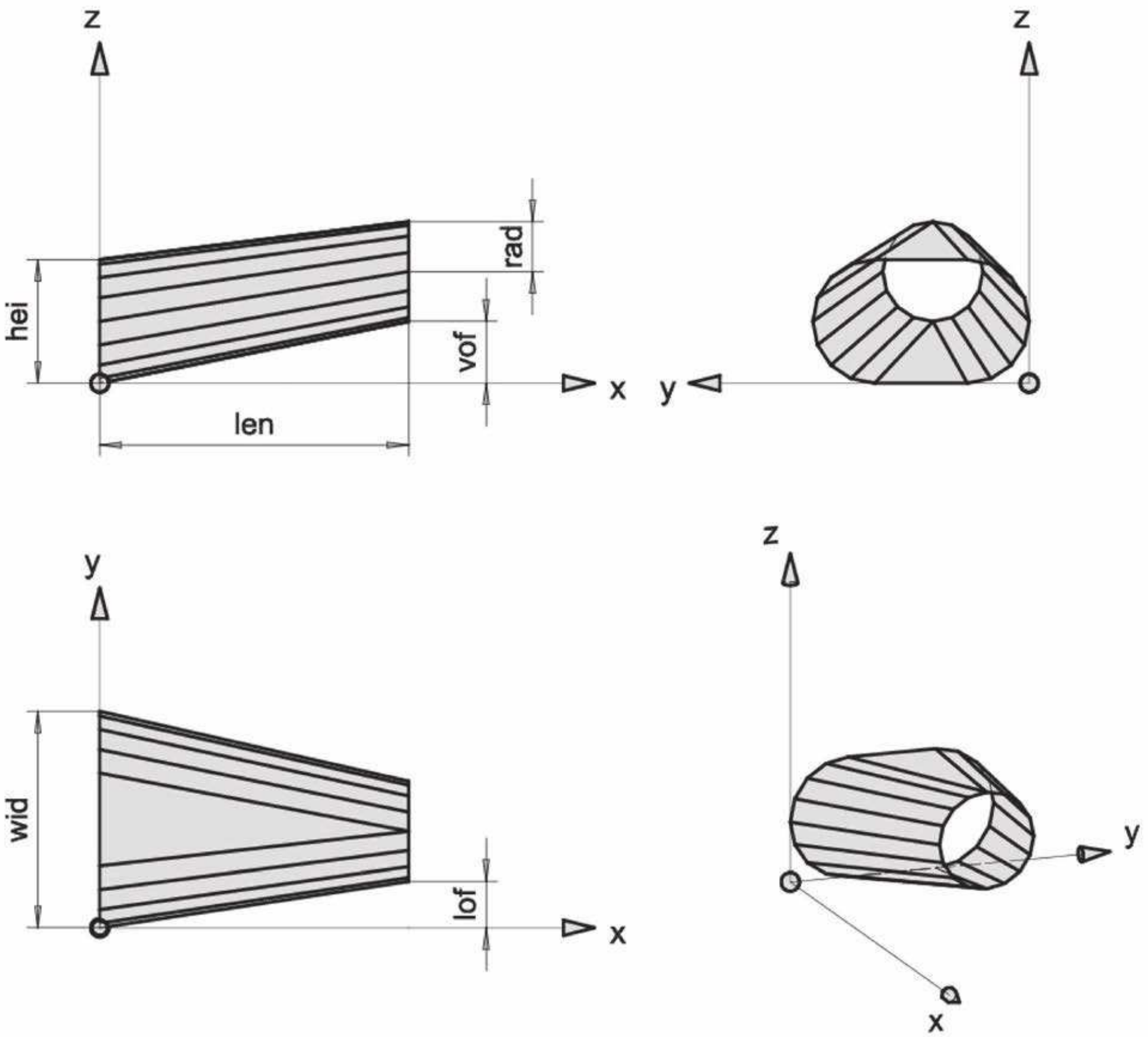

Figure A.15 — Oval/round transition

Primitive identifier: 'oval_round_transition'

An oval/round transition is a type of regular CSG primitive with rectangular base and round end. Its behaviour in Boolean operations is the same as that of a solid primitive. The base and end surface can be displayed as invisible to perform a rectangle/round transition of thin sheet metal.

EXPRESS Specification:

```
*)
ENTITY oval_round_transition
 SUBTYPE OF (geometric_representation_item);
    position: axis2_placement_3d;
    wth:      positive_length_measure;
    len:      positive_length_measure;
    wid:      positive_length_measure;
    hei:      positive_length_measure;
    rad:      positive_length_measure;
    lof:      length_measure;
    vof:      length_measure;
    WHERE
      WR1:    wth >= 0
      WR2:    len > 0
      WR3:    wid > 2*wth
      WR4:    hei > 2*wth
      WR5:    rad > wth
END_ENTITY;
(*
```

Attribute definitions:

position: The location and orientation of the axis system for the oval/round transition.

The **oval_round_transition** has one vertex at **position.location** and its edges are aligned with the placement axes in the positive sense.

wth: The wall thickness size of the thin sheet metal forming the oval/round transition (position.p[1]).

len: The length size of the oval/round transition along the placement x-axis (position.p[1]).

wid: The width size of the oval/round transition base along the placement y-axis (position.p[2]).

hei: The height size of the oval/round transition base along the placement z-axis (position.p[3]).

rad: The radius of the round transition end along the placement y-axis (position.p[2]).

lof: The lateral offset of the round transition end along the placement y-axis (position.p[2]).

vof: The vertical offset of the round transition end along the placement z-axis (position.p[3]).

Formal propositions:

WR1: The wall thickness **wth** shall be greater or equal than 0.

WR2: The length **len** shall be greater than 0.

WR3: The width **wid** shall be greater than 2 * **wth.**

WR4: The height **hei** shall be greater than 2 * **wth.**

WR5: The radius **rad** shall be greater than **wth.**

Informal propositions:

IP1: Vertical and lateral offsets can be entered as positive or negative values referred to the placement y-axis or z-axis.

IP2: The orientations of the ovals are determined by whichever has the greater absolute value of the dimensions **hei** or **wid**.

Specifications in IFC:

The oval round transition is defined with the following IFC definitions.

XSD Specification (IFC):

```
<xs:element name="IfcOvalRoundTransition" type="ifc:IfcOvalRoundTransition"
substitutionGroup="ifc:IfcCsgPrimitive3D" nillable="true"/>
 <xs:complexType name="IfcOvalRoundTransition">
  <xs:complexContent>
   <xs:extension base="ifc:IfcCsgPrimitive3D">
    <xs:attribute name="wth" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="len" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="wid" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="hei" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="rad" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="lof" type="ifc:IfcLengthMeasure" use="optional"/>
    <xs:attribute name="vof" type="ifc:IfcLengthMeasure" use="optional"/>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
```

EXPRESS Specification (IFC):

```
ENTITY IfcOvalRoundTransition9)
 SUBTYPE OF IfcCsgPrimitive3D;
    wth: IfcPositiveLengthMeasure;
    len: IfcPositiveLengthMeasure;
    wid: IfcPositiveLengthMeasure;
    hei: IfcPositiveLengthMeasure;
    rad: IfcPositiveLengthMeasure;
    lof: IfcLengthMeasure;
    vof: IfcLengthMeasure;
END_ENTITY;
```

9) http://www.buildingsmart-tech.org/ifc/IFC4/final/html/schema/ifcgeometricmodelresource/lexical/ifcblock.htm

A.17 Round pipe

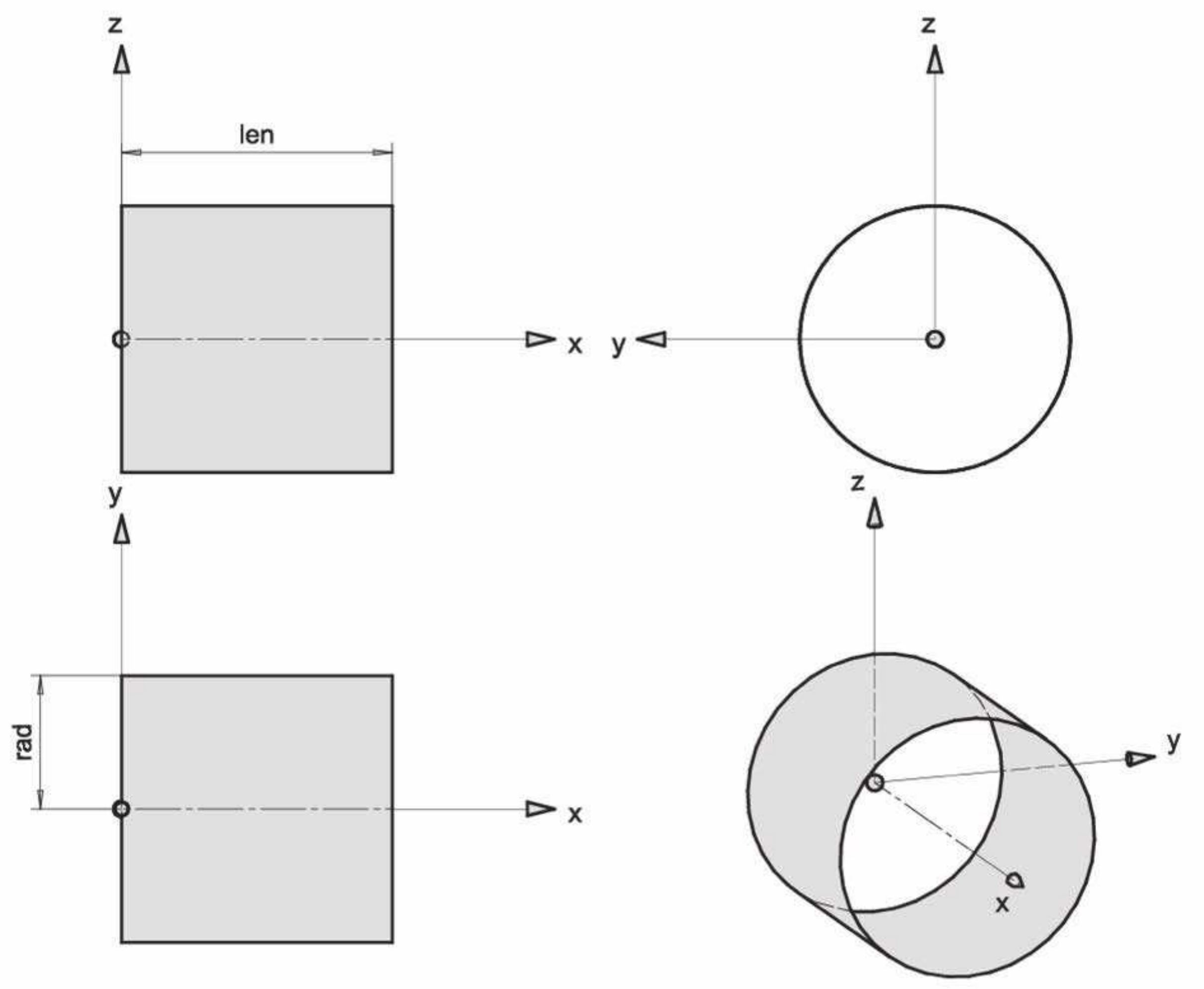

Figure A.16 — Round pipe

Primitive identifier: 'round_pipe'

A round pipe is a type of regular prismatic CSG primitive with round base. Its behaviour in Boolean operations is the same as that of a right_circular_cylinder. The base and end surface can be displayed as invisible to perform a pipe of thin sheet metal.

EXPRESS Specification:

```
*)
ENTITY round_pipe
 SUBTYPE OF (geometric_representation_item);
    position: axis2_placement_3d;
    wth:      positive_length_measure;
    len:      positive_length_measure;
    rad:      positive_length_measure;
    WHERE
      WR1:    wth >= 0
      WR2:    len > 0
      WR3:    rad > wth
END_ENTITY;
(*
```

Attribute definitions:

position: The location and orientation of the axis system for the round pipe.

The **round_pipe** has one vertex at **position.location** and its edges are aligned with the placement axes in the positive sense. The central axis of the pipe is the placement x-axis.

wth: The wall thickness size of the thin sheet metal forming the round pipe (position.p[1]).

len: The length size of the round pipe along the placement x-axis (position.p[1]).

rad: The radius of the round pipe along the placement y-axis (position.p[2]).

Formal propositions:

WR1: The wall thickness **wth** shall be greater than 0.

WR2: The length **len** shall be greater than 0.

WR3: The radius **rad** shall be greater than **wth.**

Specifications in IFC:

The round pipe is defined with existing IFC definitions.

The rectangular_duct is defined as:	**IfcExtrudedAreaSolid**
The base face is the parametrizable 2D profile:	**IfcCircleHollowProfileDef**

A.18 Round pipe transition

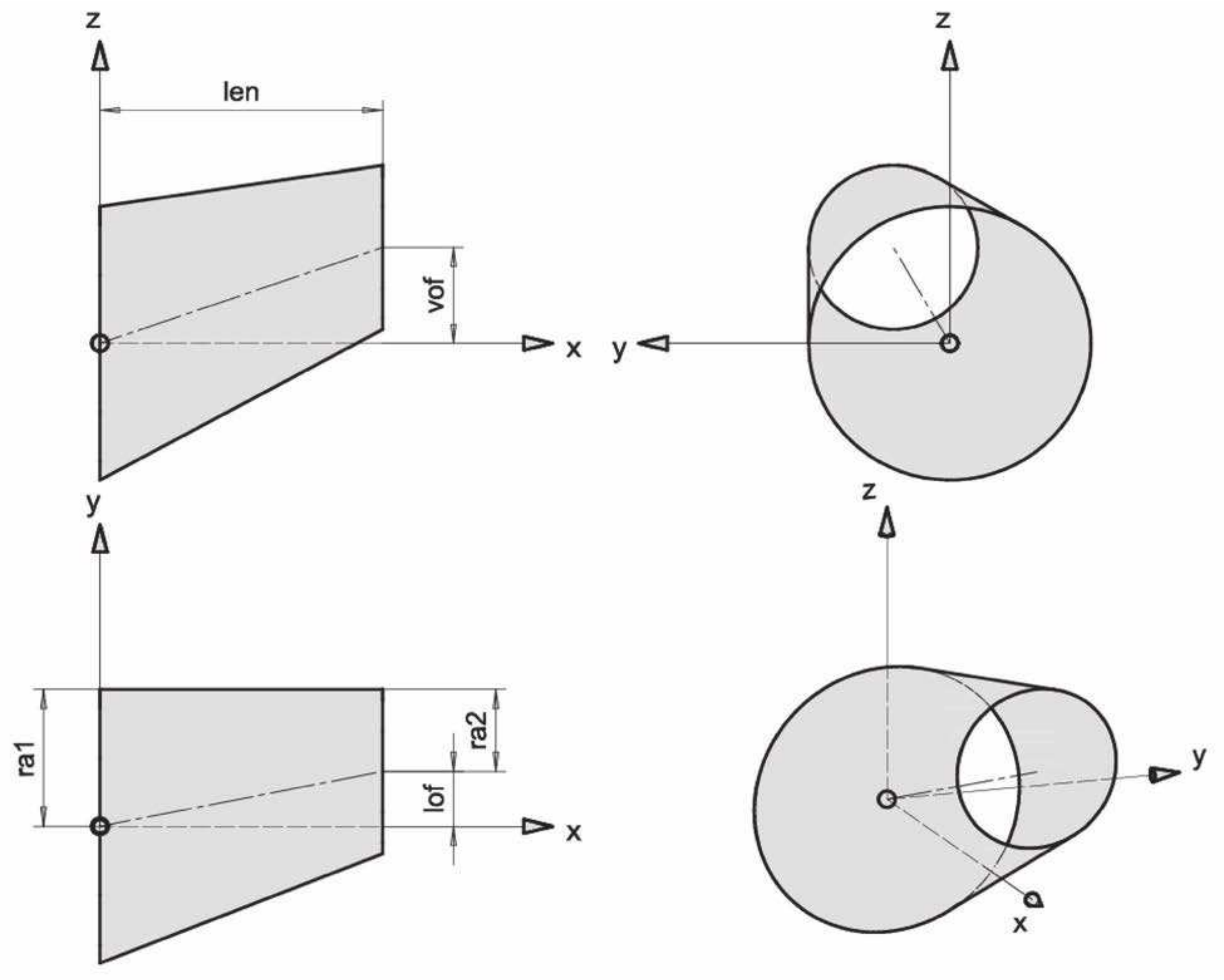

Figure A.17 — Round pipe transition

Primitive identifier: 'round_pipe_transition'

A round_pipe_transition is a type of regular CSG primitive with round base and round end. Its behaviour in Boolean operations is the same as that of an excentric_cone. The base and end surface can be displayed as invisible to perform a round pipe transition of thin sheet metal.

EXPRESS Specification:

```
*)
ENTITY round_pipe_transition
 SUBTYPE OF (geometric_representation_item);
    position: axis2_placement_3d;
    wth:      positive_length_measure;
    len:      positive_length_measure;
    ra1:      positive_length_measure;
    ra2:      positive_length_measure;
    lof:      length_measure;
    vof:      length_measure;
    WHERE
      WR1:    wth >= 0
      WR2:    len > 0
      WR3:    ra1 > wth
      WR4:    ra2 > wth
END_ENTITY;
(*
```

Attribute definitions:

position: The location and orientation of the axis system for the **round_pipe_transition.**

The **round_pipe_transition** has one vertex at **position.location** and its edges are aligned with the placement axes in the positive sense.

wth: The wall thickness size of the thin sheet metal forming the round pipe transition (position.p[1]).

len: The length size of the round pipe transition along the placement x-axis (position.p[1]).

ra1: The height size of the round pipe transition base along the placement y-axis (position.p[2]).

ra2: The radius of the round transition end along the placement y-axis (position.p[2]).

lof: The lateral offset of the round transition end along the placement y-axis (position.p[2]).

vof: The vertical offset of the round transition end along the placement z-axis (position.p[3]).

Formal propositions:

WR1: The wall thickness **wth** shall be greater than 0.

WR2: The length **len** shall be greater than 0.

WR3: The radius **ra1** shall be greater than **wth.**

WR4: The radius **ra2** shall be greater than **wth.**

Informal propositions:

IP1: Vertical and lateral offsets can be entered as positive or negative values referred to the placement y-axis or z-axis.

Specifications in IFC:

The round pipe transition is defined with existing IFC definitions.

The round pipe transition is defined as:	**IfcExtrudedAreaSolid**
The base face is the parametrizable 2D profile:	**IfcCircleHollowProfileDef**
Included is the transformation of the start face by:	**IfcDerivedProfileDef**
	IfcCartesianTransformationOperator2D

A.19 Round pipe radius transition

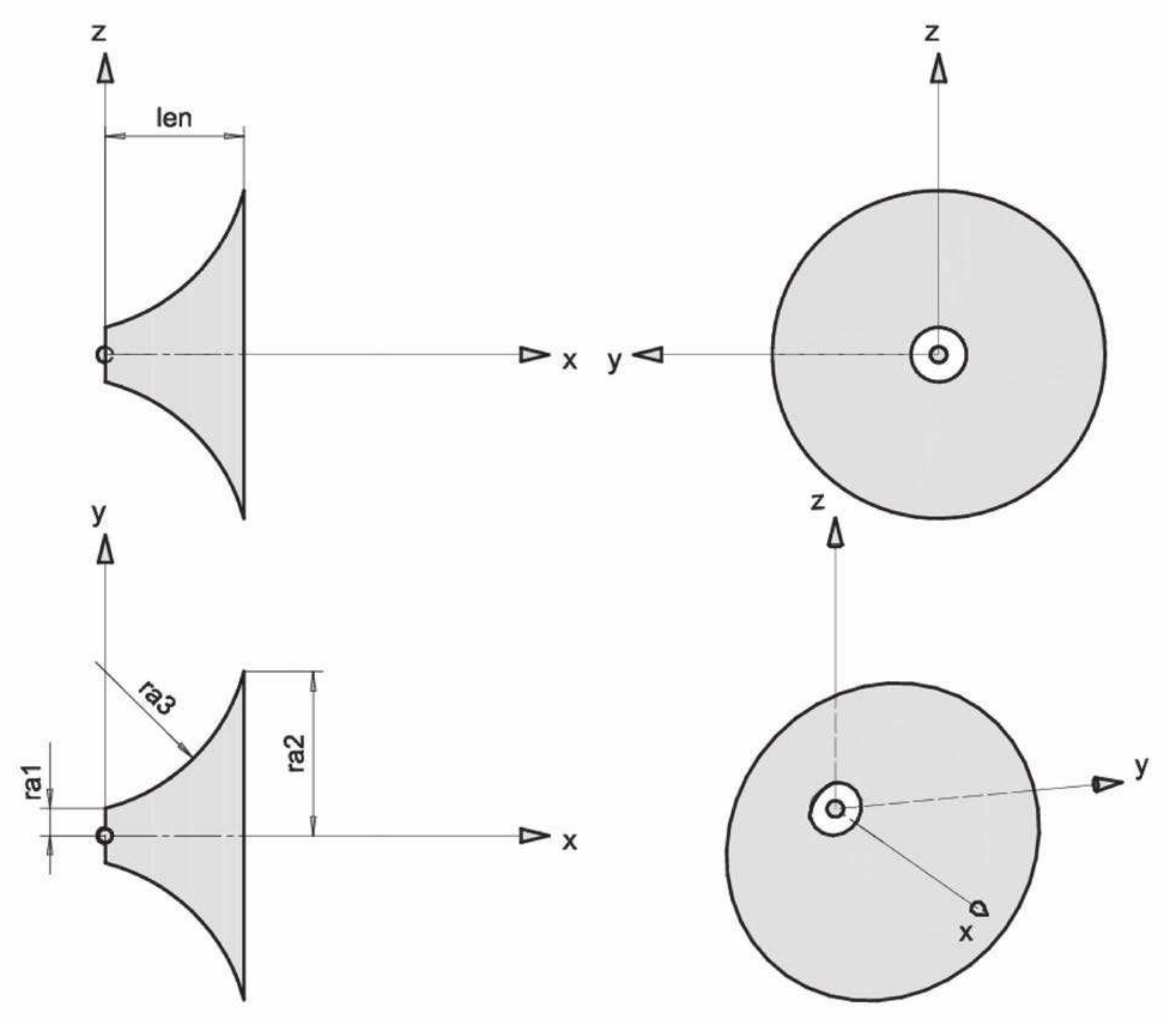

Figure A.18 — Round pipe radius transition

Primitive identifier: 'round_pipe_radius_transition'

A round_pipe_radius_transition is a type of regular CSG primitive with round base and round end. Its behaviour in Boolean operations is the same as that of a solid primitive. The base and end surface can be displayed as invisible to perform a round pipe transition of thin sheet metal.

EXPRESS Specification:

```
*)
ENTITY round_pipe_radius_transition
 SUBTYPE OF (geometric_representation_item);
    position: axis2_placement_3d;
    wth:      positive_length_measure;
    len:      positive_length_measure;
    ra1:      positive_length_measure;
    ra2:      positive_length_measure;
    ra3:      length_measure;
    WHERE
      WR1:    wth >= 0
      WR2:    len > 0
      WR3:    ra1 > wth
      WR4:    ra2 > wth
      WR5:    ABS(ra3) > 0.5 * ((ra2 – ra1) ^2 + len ^2) ^0.5
END_ENTITY;
(*
```

Attribute definitions:

position: The location and orientation of the axis system for the round pipe transition.

The **round_pipe_radius_transition** has one vertex at **position.location** and its edges are aligned with the placement axes in the positive sense.

wth: The wall thickness size of the thin sheet metal forming the round pipe radius transition (position.p[1]).

len: The length size of the round pipe radius transition along the placement x-axis (position.p[1]).

ra1: The height size of the round pipe radius transition base along the placement z-axis (position.p[3]).

ra2: The radius of the round pipe radius transition end along the placement y-axis (position.p[2]).

ra3: The radius between the round pipe radius transition base and end along the placement y-axis (position.p[2]).

Formal propositions:

WR1: The wall thickness **wth** shall be greater than 0.

WR2: The length **len** shall be greater than 0.

WR3: The radius **ra1** shall be greater than **wth.**

WR4: The radius **ra2** shall be greater than **wth**

WR5: The radius **ra3** is constructed as either a positive (concave) or negative (convex) circular arc between the points defined by **ra1**, **ra2** and **len** (basic geometric problem: given two points and the radius of a circle).

Hence, the absolute value of **ra3** shall **not** be smaller than half the point spacing:

$$\left|ra3\right| > \frac{1}{2} * \sqrt{[(ra2 - ra1)^2 + len^2]}$$

Specifications in IFC:

The round pipe radius transition is defined with the following IFC definitions.

XSD Specification (IFC):

```
<xs:element name="IfcRoundPipeRadiusTransition" type="ifc:IfcRoundPipeRadiusTransition"
substitutionGroup="ifc:IfcCsgPrimitive3D" nillable="true"/>
 <xs:complexType name="IfcRoundPipeRadiusTransition">
  <xs:complexContent>
   <xs:extension base="ifc:IfcCsgPrimitive3D">
    <xs:attribute name="wth" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="len" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="wid" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="ra1" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="ra2" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="ra3" type="ifc:IfcLengthMeasure" use="optional"/>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
```

EXPRESS Specification (IFC):

```
ENTITY IfcOvalRoundTransition
 SUBTYPE OF IfcCsgPrimitive3D;
    wth: IfcPositiveLengthMeasure;
    len: IfcPositiveLengthMeasure;
    ra1: IfcPositiveLengthMeasure;
    ra2: IfcPositiveLengthMeasure;
    ra3: IfcLengthMeasure;
END_ENTITY;
```

A.20 Round pipe bend

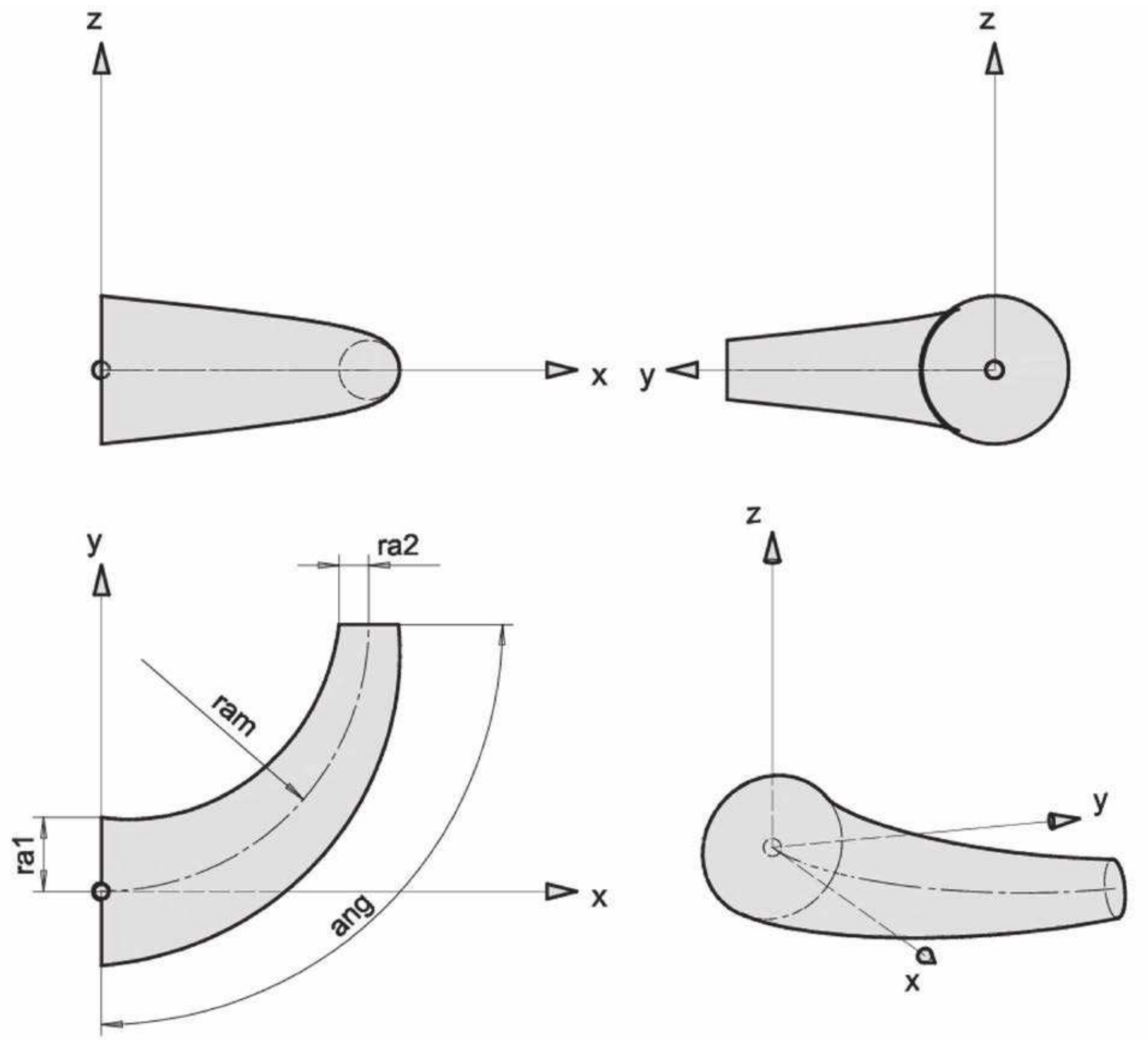

Figure A.19 — Round pipe bend transition

Primitive identifier: 'round_pipe_bend_transition'

A round pipe bend transition is a type of CSG primitive with a circular base face and a circular end face. directrix is a circle segment lying in the x-y plane with its centre point on the positive y-axis.

The cross-section of a plane normal to the directrix plane and the centre point of the directrix is always a circle. The radii of all these circles are that of a cone on the developed length of the directrix.

EXPRESS Specification:

```
*)
ENTITY round_pipe_bend_transition
  SUBTYPE OF (geometric_representation_item);
    position: axis2_placement_3d;
    wth:      positive_length_measure
    ram:      positive_length_measure;
    ra1:      positive_length_measure;
    ra2:      positive_length_measure;
    ang:      plane_angle_measure;
    WHERE
      WR1:    wth > 0
```

```
      WR2:    ram >= MAX(ra1,ra2)
      WR3:    ra1 >= wth
      WR4:    ra2 >= wth
      WR5:    ang > 0
      WR6:    ang <= 360
END_ENTITY;
(*
```

Attribute definitions:

position: The location and orientation of the placement axis system for the round pipe bend transition.

The base circular face of the **round_pipe_bend_transition** has its centre at **position. location** and the directrix has its centre on the placement y-axis

wth: The wall thickness size of the thin sheet metal forming the round pipe bend transition (position.p[1]).

ram: The radius (mayor radius) of the directrix.

ra1: The radius at the start face which is parallel to the placement y-z plane.

ra2: The radius at the end face.

ang: The turn angle between the planes of the two circular faces of the solid, measured in the sector containing the solid.

Formal propositions:

WR1: The wall thickness **wth** shall be greater than 0.

WR2: The radius **ram** shall be greater or equal to the maximum of **ra1** and **ra2.**

WR3: The radius **ra1** shall be greater or equal to **wth.**

WR4: The radius **ra2** shall be greater or equal to **wth.**

WR5: The angle **ang** shall be greater than 0.

WR6: The angle **ang** shall be less or equal to 360°.

Informal propositions:

IP1: The round pipe bend transition can become a torus bend, if the base and the end radius are equal.

IP2: A torus bend with a turn angle of 360° is a closed torus.

Specifications in IFC:

Round pipe bend transitions are defined with existing IFC definitions.

The toroidal bend transition is defined as:	**IfcRevolvedAreaSolid**
The base face is the parametrizable 2D profile:	**IfcHollowCircleProfileDef**
The base face will be transformed by:	**IfcDerivedProfileDef**
	IfcCartesianTransformationOperator2D

A.21 Round pipe wye

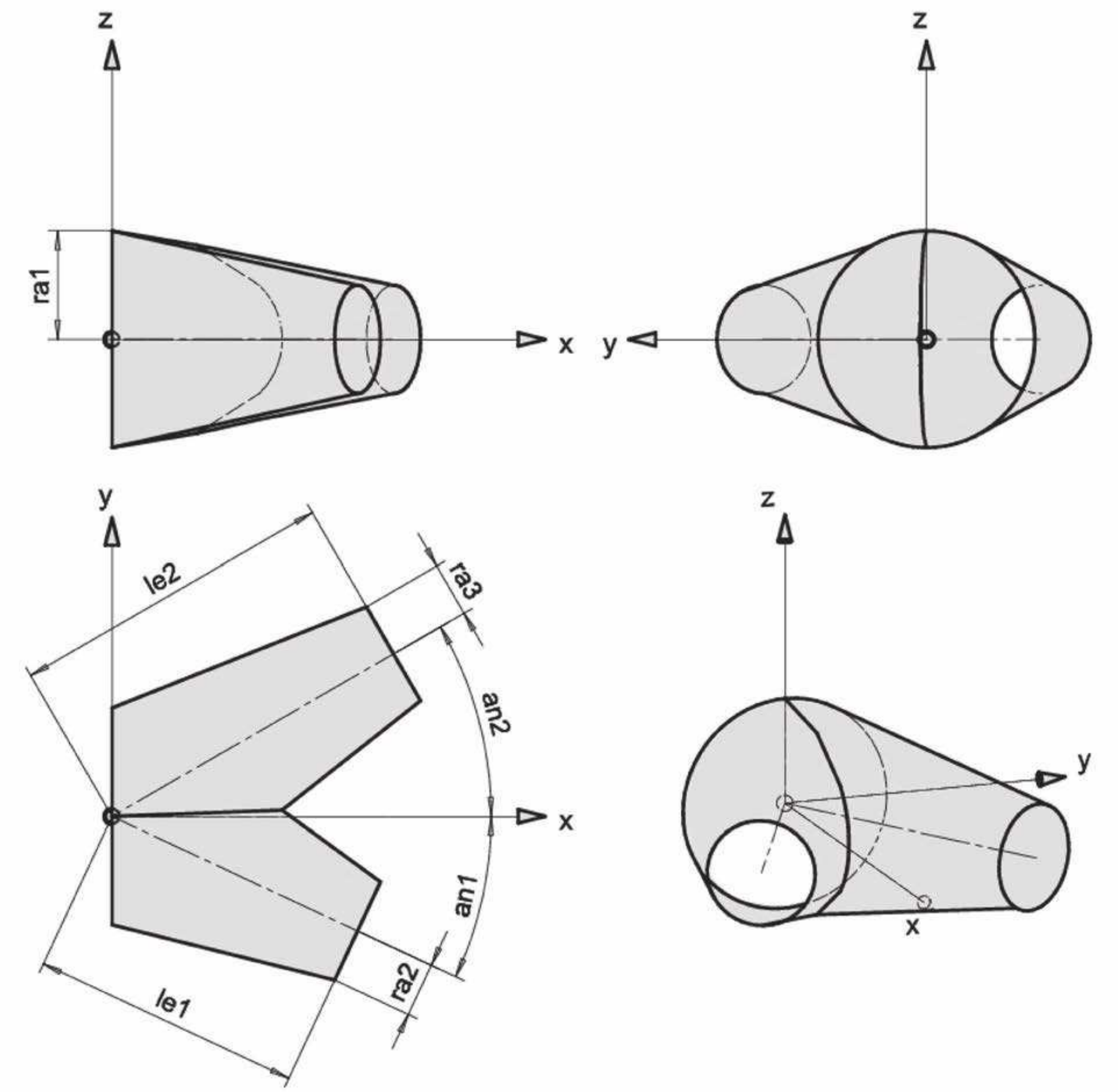

Figure A.20 — Round pipe wye

Primitive identifier: 'round_pipe_wye'

A round pipe wye is a type of regular CSG primitive with one round base and two round ends. Its behaviour in Boolean operations is the same as that of a solid primitive. The base and end surfaces can be displayed as invisible to perform a round pipe wye of thin sheet metal.

EXPRESS Specification:

```
*)
ENTITY round_pipe_wye
  SUBTYPE OF (geometric_representation_item);
    position: axis2_placement_3d;
    wth:      positive_length_measure;
    ra1:      positive_length_measure;
    ra2:      positive_length_measure;
    ra3:      positive_length_measure;
    le1:      positive_length_measure;
```



```
    le2:        positive_length_measure;
    an1:        plane_angle_measure;
    an2:        plane_angle_measure;
    WHERE
      WR1:      wth > 0
      WR2:      ra1 >= wth
      WR3:      ra2 >= wth
      WR4:      ra3 >= wth
      WR5:      le1 > 0
      WR6:      le2 > 0
      WR7:      an1 > -90
      WR8:      an1 < 90
      WR9:      an2 > -90
      WR10:     an2 < 90
END_ENTITY;
(*
```

Attribute definitions:

position: The location and orientation of the axis system for the round pipe wye.

The **round_pipe_wye** has one vertex at **position.location** and its edges are aligned with the placement axes in the positive sense.

wth: The wall thickness size of the thin sheet metal forming the round pipe wye (position.p[1]).

ra1: The radius of the round pipe wye base along the placement y-axis (position.p[2]).

ra2: The radius of the round pipe wye end to the x-axis (position.p[1]) at the lower y position.

ra3: The radius of the round pipe wye end to the x-axis (position.p[1]) at the upper y position.

le1: The length of the round pipe wye end to the x-axis (position.p[1]) at the lower y position.

le2: The length of the round pipe wye end to the x-axis (position.p[1]) at the upper y position.

an1: The angle of the round pipe wye branch axe at the lower y position to the x-axis (position.p[1]).

an2: The angle of the round pipe wye branch axe at the upper y position to the x-axis (position.p[1]).

Formal propositions:

WR1: The wall thickness **wth** shall be greater than 0.

WR2: The radius **ra1** shall be greater or equal to 0.

WR3: The radius **ra2** shall be greater or equal to 0.

WR4: The radius **ra3** shall be greater or equal to 0.

WR5: The length **le1** shall be greater than 0.

WR6: The length **le2** shall be greater than 0.

WR7: The angle **an1** shall be greater than −90.

WR8: The angle **an1** shall be less than 90.

WR9: The angle **an2** shall be greater than −90.

WR10: The angle **an2** shall be less than 90.

Informal propositions:

IP1: For the offsets, positive or negative values referred to the placement y-axis can be entered. wyes with legs differing in length can be created by attaching a round_pipe_transition primitive.

IP2: The angles to the axes of the conical branches are constructed starting from the x-axis to the right and left, respectively. They can also be entered as negative values.

Specifications in IFC:

The round pipe wye is defined with the following IFC definitions.

XSD Specification (IFC):

```
<xs:element name="IfcRoundPipeWye" type="ifc: IfcRoundPipeWye"
substitutionGroup="ifc:IfcCsgPrimitive3D" nillable="true"/>
 <xs:complexType name=" IfcRoundPipeWye">
  <xs:complexContent>
   <xs:extension base="ifc:IfcCsgPrimitive3D">
    <xs:attribute name="wth" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="ra1" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="ra2" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="ra3" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="le1" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="le2" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="an1" type="ifc:IfcPlaneAngleMeasure" use="optional"/>
    <xs:attribute name="an2" type="ifc:IfcPlaneAngleMeasure" use="optional"/>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
```

EXPRESS Specification (IFC):

```
ENTITY IfcRoundPipeWye10)
 SUBTYPE OF IfcCsgPrimitive3D;
    wth: IfcPositiveLengthMeasure;
    ra1: IfcPositiveLengthMeasure;
    ra2: IfcPositiveLengthMeasure;
    ra3: IfcPositiveLengthMeasure;
    le1: IfcPositiveLengthMeasure;
    le2: IfcPositiveLengthMeasure;
    an1: IfcPlaneAngleMeasure;
    an2: IfcPlaneAngleMeasure;
END_ENTITY;
```

10) http://www.buildingsmart-tech.org/ifc/IFC4/final/html/schema/ifcgeometricmodelresource/lexical/ifcblock.htm

A.22 Extruded face duct

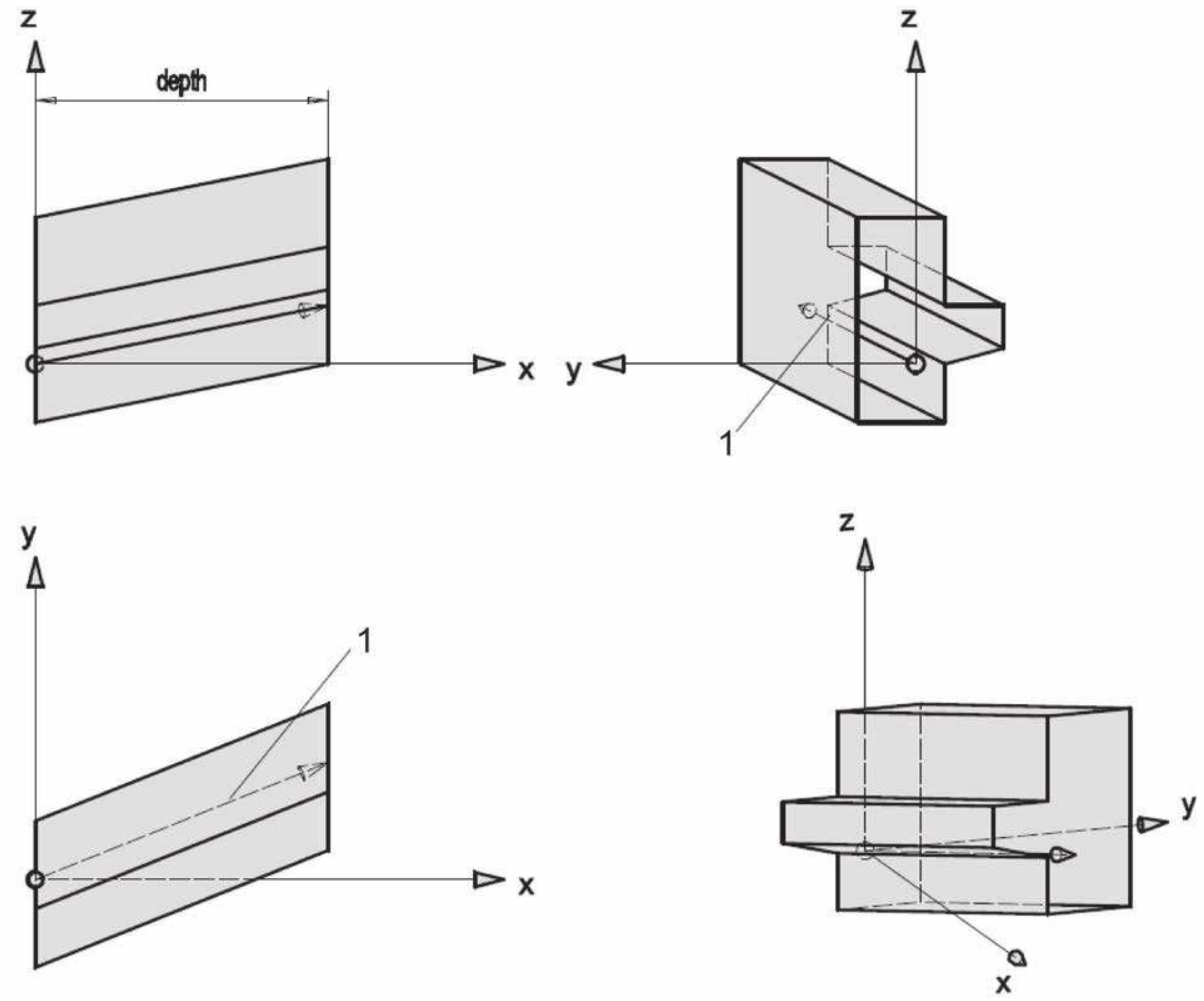

Figure A.21 — Extruded face duct

Key

1 extruded direction

Primitive identifier: 'extruded_face_duct'

An **extruded_face_duct** is a solid defined by sweeping a planar **face**. The direction of translation isdefined by a **direction** vector, and the length of the translation is defined by a distance **depth**. The planar face may **not** have any holes.

EXPRESS Specification:

```
*)
ENTITY extruded_face_duct
  SUBTYPE OF (swept_face_solid);
    wth: positive_length_measure;
    DERIVE
    extruded_direction: direction;
    depth: positive_length_measure;
    WHERE
      WR1:
        dot_product((SELF\swept_face_solid.swept_face.face_geometry\elementary_surface.
position.p[3]), extruded_direction) <> 0.0;
END_ENTITY;
(*
```

Attribute definitions:

SELF\swept_face_solid.swept_face: The face to be extruded to produce the solid.

wth: The wall thickness size of the thin sheet metal forming the extruded face duct (position.p[1]).

extruded_direction: The direction in which the face is to be swept.

depth: The distance the face is to be swept.

Informal propositions:

IP1: extruded_direction shall not be perpendicular to the normal to the plane of the swept_face.

A.23 Revolved face duct

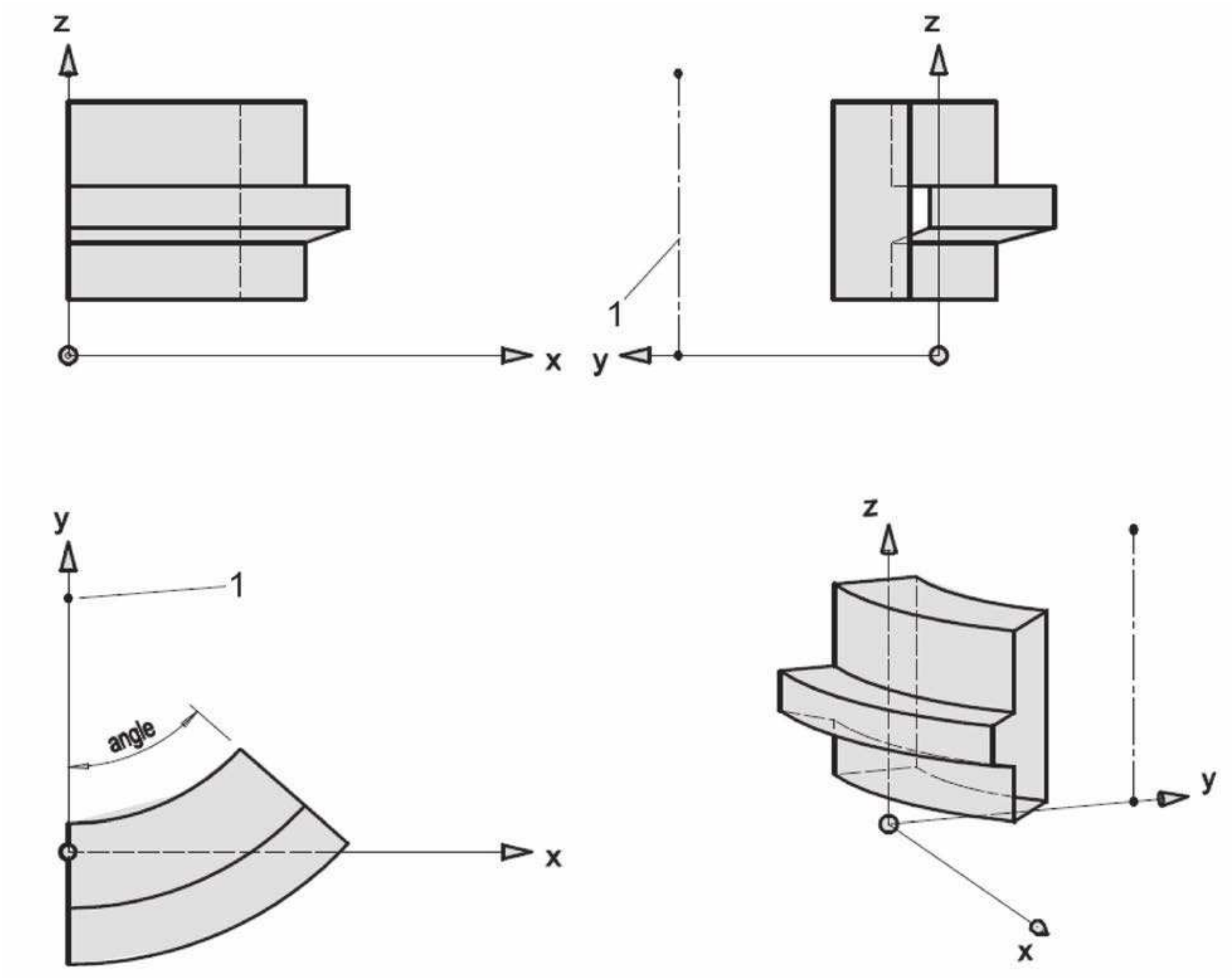

Figure A.22 — Revolved face duct

Key

1 axis line

Primitive identifier: 'revolved_face_duct'

A **revolved_face_duct** is a solid of revolution formed by revolving a planar **face** about an axis. The axis shall be in the plane of the face and the axis shall not intersect the interior of the face. The planar face may **not** have any holes. The direction of revolution is clockwise when viewed along the axis in the positive direction. More precisely, if **A** is the axis location and **d** is the axis direction and **C** is an arc on the surface of revolution generated by an arbitrary point **p** on the boundary of the face, then **C** leaves **p** in direction **d** x (**p** - **A**) as the face is revolved.

EXPRESS Specification:

```
*)
ENTITY revolved_face_solid
  SUBTYPE OF (swept_face_solid);
       axis: axis1_placement;
       wth: positive_length_measure;
       angle: plane_angle_measure;
       DERIVE
axis_line: line: = representation_item('')||
geometric_representation_item()|| curve()||
line(axis.location, representation_item('')||
geometric_representation_item()||
vector(axis.z, 1.0));
END_ENTITY;
(*
```

Attribute definitions:

SELF\swept_face_solid.swept_face:	The face to be revolved to produce the solid.
wth:	The wall thickness size of the thin sheet metal forming the duct wye (position.p[1]).
axis_line:	Axis about which revolution will take place.
angle:	Angle through which the sweep will be made. This angle is measured from the plane of the swept face.
axis_line:	The line of the axis of revolution.

Informal propositions:

IP1: The axis_line shall lie in plane of swept_face attribute of the swept_face_solid supertype.

IP2: The axis_line shall not intersect the interior of the swept_face.

IP3: Angle shall lie in the range 0° < angle ≤ 360°.

IP4: If the rotation angle is 0°, the revolved face duct is regarded as geometrically inexistent.

Bibliography

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

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

[3] ISO 10303-28, *Industrial automation systems and integration — Product data representation and exchange — Part 28: Implementation methods: XML representations of EXPRESS schemas and data, using XML schemas*

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

[5] ISO 10303-42:2014, *Industrial automation systems and integration — Product data representation and exchange — Part 42: Integrated generic resource: Geometric and topological representation*

[6] ISO 12006-3, *Building construction — Organization of information about construction works — Part 3: Framework for object-oriented information*

[7] ISO 16739:2013, *Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries*

[8] ISO 13584-31, *Industrial automation systems and integration — Parts library — Part 31: Implementation resources: Geometric programming interface*

[9] BIMFORUM, Level of Development Specification — For Building Information Models — Version 2013

ICS 91.010.01
Price based on 82 pages