NISTIR 7716

Recommended practice extensions for GD&T in STEP-AP210 in the context of electronic connectors

Jamie Stori
SFM Technology, Inc.
Urbana, Illinois

Kevin Brady
National Institute of Standards and Technology
Information Technology Laboratory

National Institute of Standards and Technology
U.S. Department of Commerce

NISTIR 7716

Recommended practice extensions for GD&T in STEP-AP210 in the context of electronic connectors

Jamie Stori
SFM Technology, Inc.
Urbana, Illinois

Kevin Brady
National Institute of Standards and Technology
Information Technology Laboratory

August 16, 2010

U.S. Department of Commerce
John Bryson, Secretary

National Institute of Standards and Technology
Patrick D. Gallagher,
Under Secretary of Commerce for Standards and Technology

Recommended practice extensions for GD&T in STEP-AP210 in the context of electronic connectors

Introduction

The purpose of this document is to extend the recommended practices for the representation of GD&T in STEP AP 210 (ISO-10303:210) in the context of packaged electronic connectors. Geometric dimensioning and tolerancing (GD&T) is critical to the design, manufacture, and assembly of electronic products. STEP AP 210 is unique in its ability to represent detailed usage views of electronic components spanning both the mechanical (MCAD) and electrical (ECAD) design domains. Within the STEP standards (ISO-10303), prior attention has been focused on the representation and presentation of GD&T in the context of three-dimensional (3D) boundary-representation (B-REP) models. In this context, GD&T annotations are typically associated with faces in the 3D model.

Vendors of electro-mechanical components such as connectors typically provide detailed specifications of the mechanical properties of their components, and often will include a recommended PCB layout for the component. Both the mechanical specification and the PCB layout recommendation will often include GD&T annotations. This is particularly relevant in the context of mounting and mating features, such as mounting and alignment holes and features such as bosses, slots, and studs. Typically, the GD&T annotations are provided as a technical drawing, most often in .pdf format.

An AP210 model of an electronic component such as a connector will ideally contain both a detailed three-dimensional geometric representation, suitable for use in mechanical design, as well as a detailed two-dimensional footprint definition that supports the application of the component in PCB layout and PCA assembly. Existing implementations have validated the ability of AP210 to represent the detailed two and three-dimensional geometric models and product structure of a connector. A prior recommended practice document[1], explored the population of GD&T annotations on a 3D geometric representation of a packaged electronic component represented in AP210. This document extends that prior work in the context of connectors by proposing a recommended practice for the representation of critical GD&T annotations within the footprint definition of a component, and updating the recommended representation of certain associations and relationships. The concepts discussed will support the GD&T representation requirements of a typical connector with a combination of both mounting features and through-hole terminals. Where relevant, elements of the prior recommended practice have been included in the current document. Many of the supporting details have not been repeated, however, and the prior document should be referenced as applicable.

[1] Stori, J., Brady, K., and Thurman, T, 2009 “Extensions to the recommended practices for GD&T in STEP-AP210 in the context of packaged electronic components,” NISTIR 7634, http://www.nist.gov/customcf/get_pdf.cfm?pub_id=903904

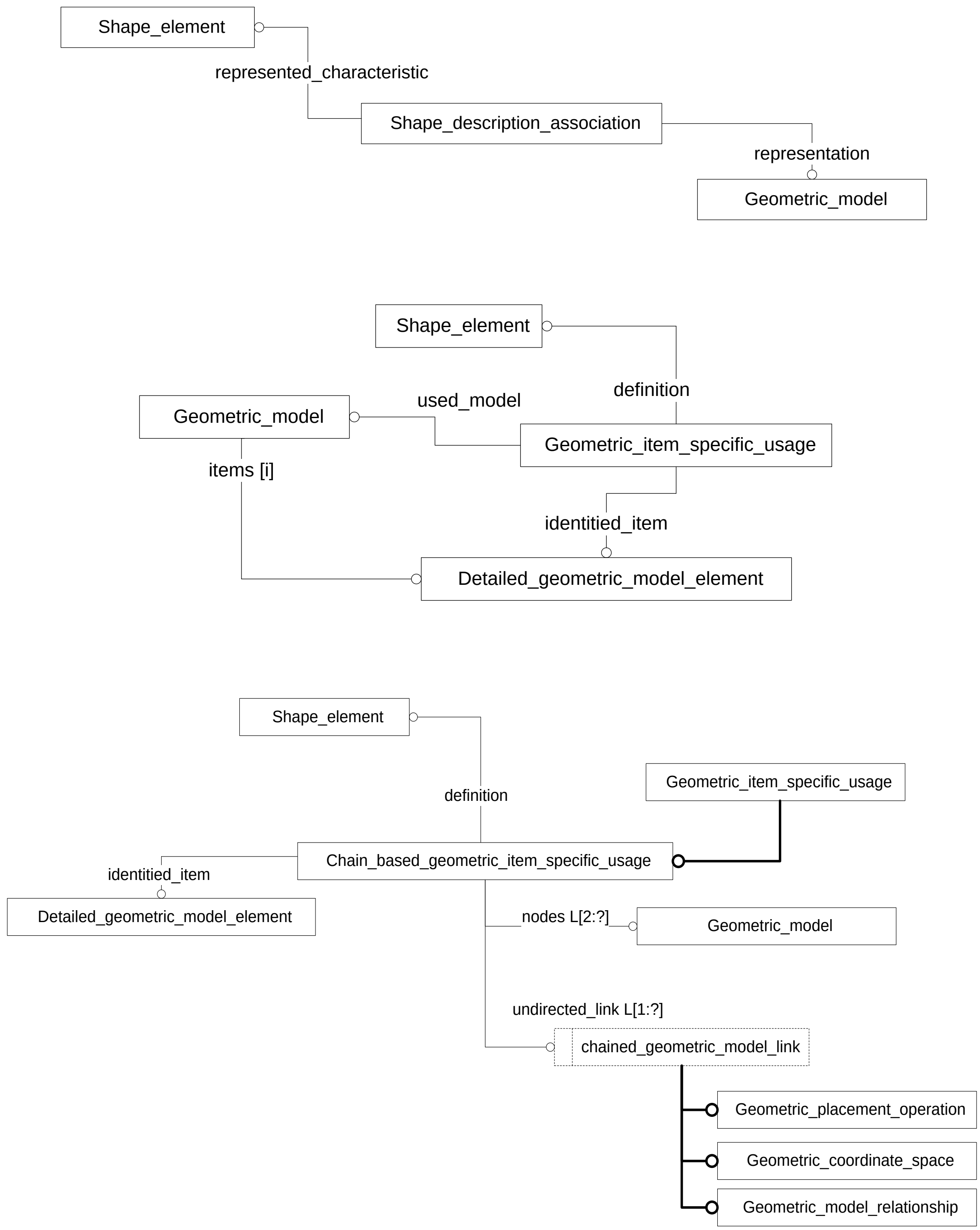


Figure 1. Three mechanisms for associating geometry with a Shape_element

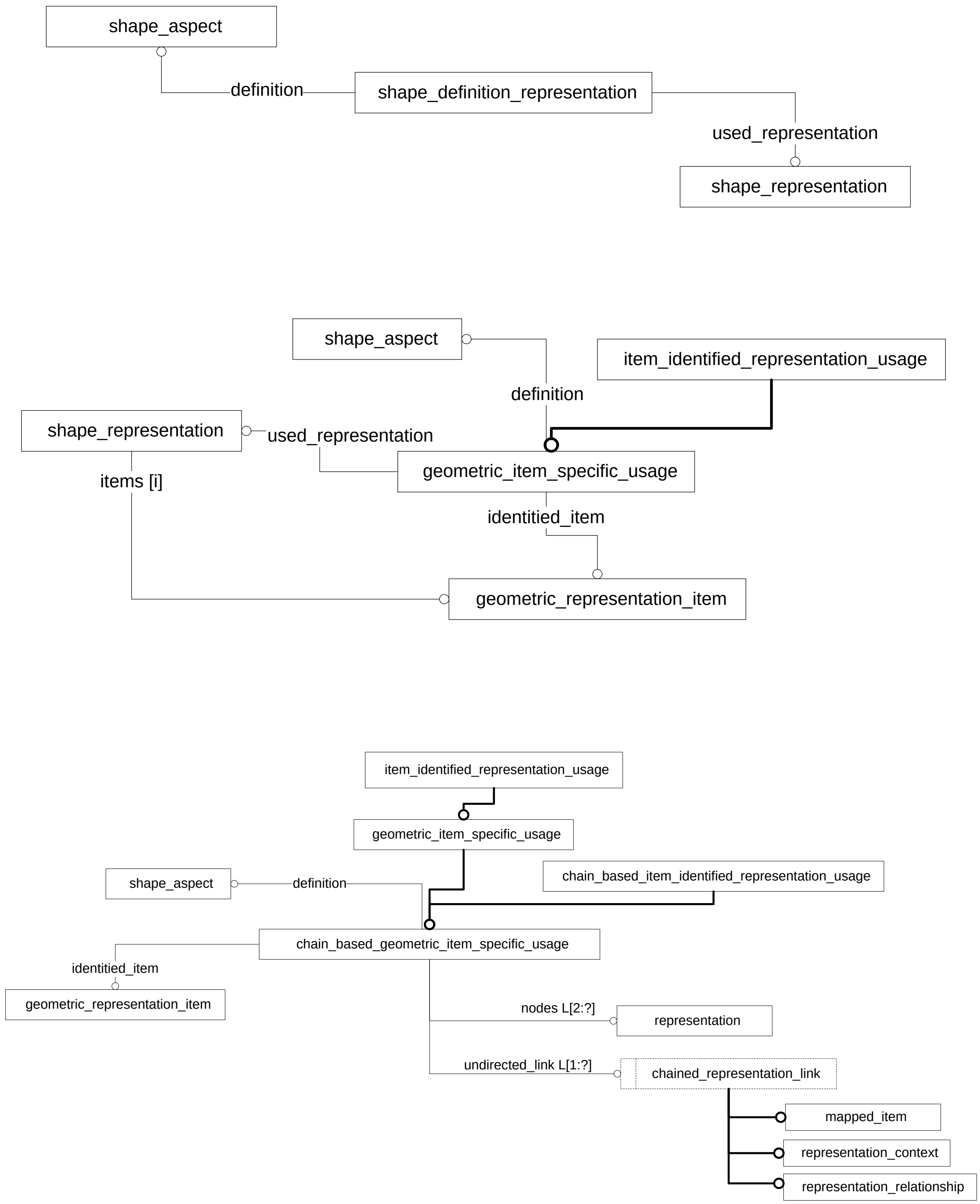


Figure 2. Three mechanisms for associating geometry with a Shape_element - MIM

Notation

ARM AOs will be denoted with a leading uppercase letter in Courier font (i.e. `Datum`) while a MIM entity will be displayed in all lowercase notation (i.e. `datum`).

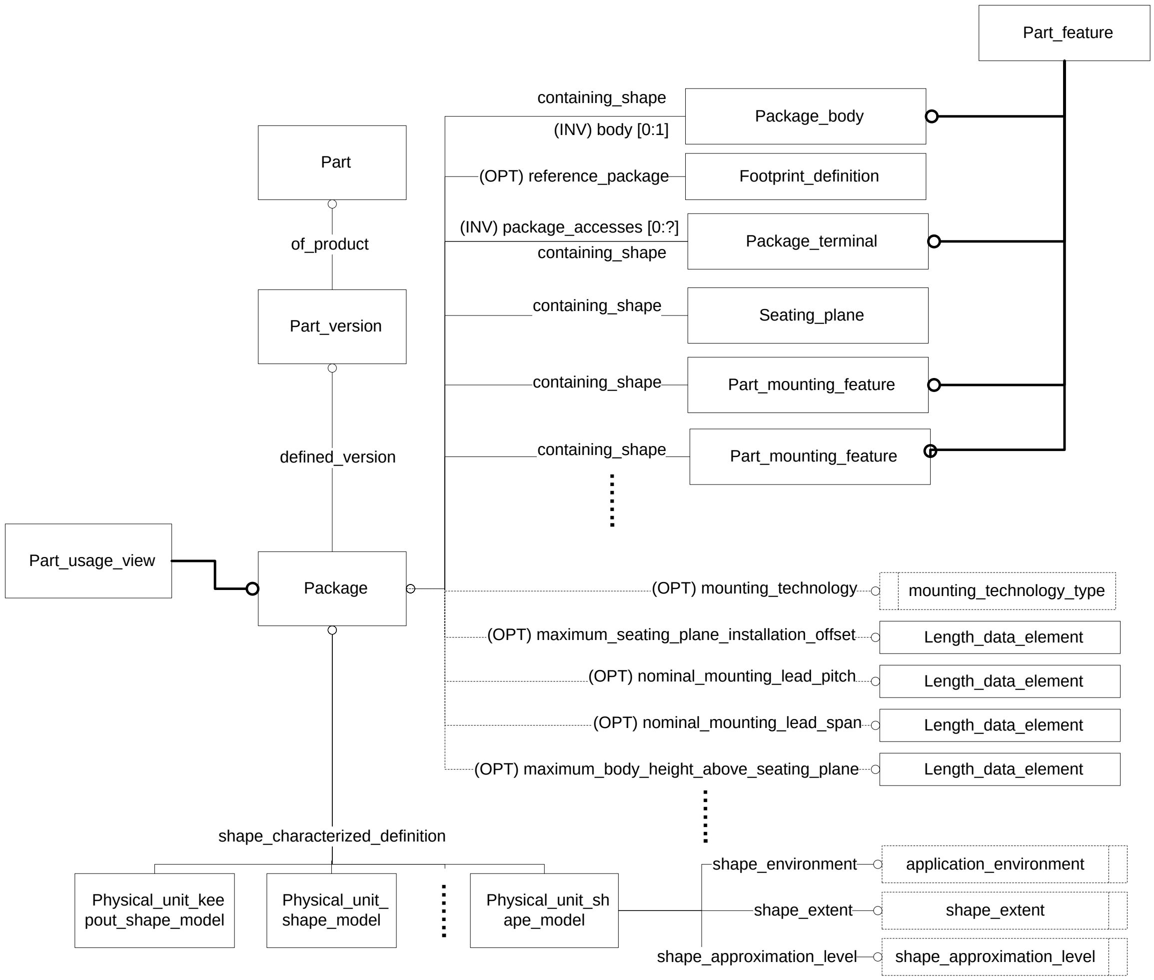


Figure 3. The top-level structure of a package model.

Shape_element

`Shape_element` is the ARM concept that enables a portion of a shape to be identified or called out for some particular purpose in model representation. Subtypes of `Shape_element` are used heavily in the package model representation, and are also the critical link between GD&T annotations, and the geometric model. Specific subtypes and their use will be discussed in further detail as relevant sections. There are several mechanisms for associated geometric the elements that compose a `Shape_element`, as illustrated in Figure 1, above. If there is an explicit `Geometric_model` that represents the `Shape_element`, a `Shape_description_association` may be employed. If the `Shape_element` is defined by a subset of a `Geometric_model`, either the `Geometric_item_specific_usage` or its subtype `Chain_based_geometric_item_specific_usage` may be used. Each of these creates a defining association between the `Shape_element` and a single `Detailed_geometric_model_element`. As will be discussed below, in the case of three-dimensional geometric models, the model element will often be a face of a boundary representation. In the case of a two-dimensional footprint definition, the model element may often be a templated shape. Often, multiple instances of the

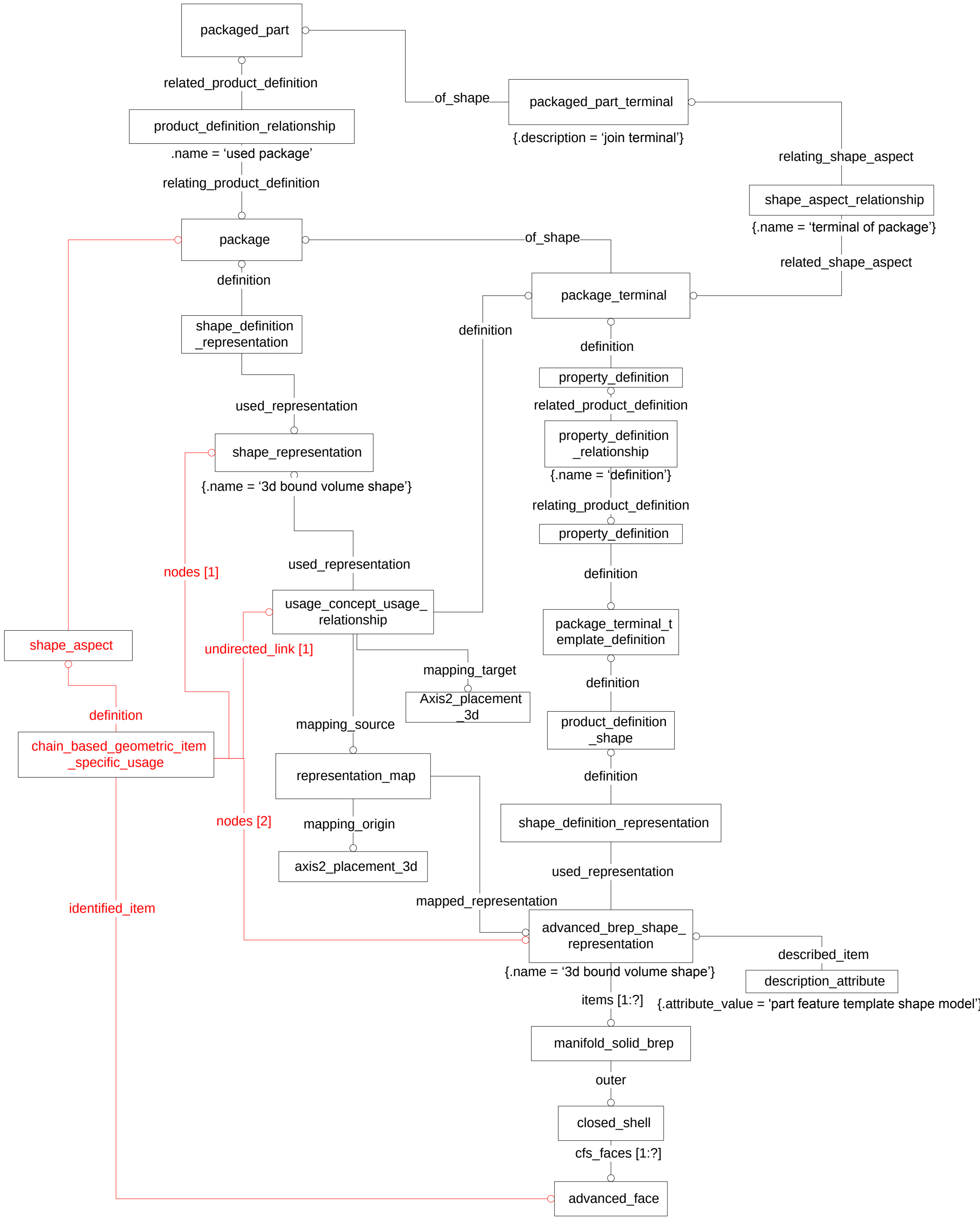


Figure 4. Defining a shape_aspect based on a surface of a terminal in a package model.

`Geometric_item_specific_usage` will be employed to enumerate the individual model elements that collectively define the `Shape_element`.
The "chain-based" subtype of Geometric_item_specific_usage recently incorporated into the standard to address a long-standing need to call out a specific instanced item within a representation. For example, in a typical mechanical assembly, there are often multiple

instances of the same part. The part may be represented with a single geometric model (i.e. a B-Rep model). The "chain-based" referencing mechanism, enables the identification of a specific face on the part model within a specific instance of the part in the assembly. Specific examples of this identification mechanism will be provided below in the context of both the package and the footprint. Figure 2 provides the MIM mapping of the `Shape_element` concepts of Figure 1.

Package Model

The key structure of a typical AP210 package model is summarized in Figure 3. Of particular relevance in the context of GD&T are the common subtypes of `Part_feature` critical to a package model representation – body, terminal, and mounting / mating features. Also of note is the `Seating_plane`, a `Non_feature_shape_element`. All of the preceding are types of `Shape_element`, and their geometric representation can vary significantly. For example, a `Package_terminal` may be a collection of faces called out from a B-Rep model of the entire package, or the package terminal may have its own defining geometric model associated with the template definition of the package terminal. Figure 4 illustrates such a case, in which an `advanced_brep_shape_representation` defines the shape of the `package_terminal_template_definition`. A specific `package_terminal` (a single terminal) is placed with respect to the shape model of the package using a `usage_concept_usage_relationship`. A `shape_aspect` is defined (in red) which is associated with a single face of this specific package terminal, despite the fact that the `shape_representation` of the terminal template is shared among many different terminals. The `chain_based_geometric_item_specific_usage` enables this identification to be made by qualifying the face in the template definition through the specific terminal placement of the `usage_concept_usage_relationship`. If it were desired to call out the entire geometric model of the template definition as a new `shape_aspect`, a direct relationship could be established with a `shape_definition_representation`. Alternatively, if it was necessary to call out a specific face of the terminal template definition as a `shape_aspect` that would apply to all uses of the template in the package (i.e. any and all package terminals that used the template), a `geometric_item_specific_usage` could be used.

Footprint_definition

A "footprint" in the context of PCB design and layout is the collections of features needed to support assembly and connectivity for a particular component. Footprint features include the copper pads or lands on the mounting surface of for surface mount terminals, mounting tabs, ground connections, and heat sinks, specification of compatible soldermask and solderpaste regions, and drill and via locations and sizes for through-hole features such as terminals, alignment studs, mounting bolts, etc.

In AP210, the definition of a footprint is represented with the ARM application object `Footprint_definition`. A footprint definition is a template that can be instanced when an individual component is placed into a design. Figure 5 outlines the most common ARM AOs and relationships composing the typical structure of a footprint definition. While a footprint definition can exist completely independently of a package

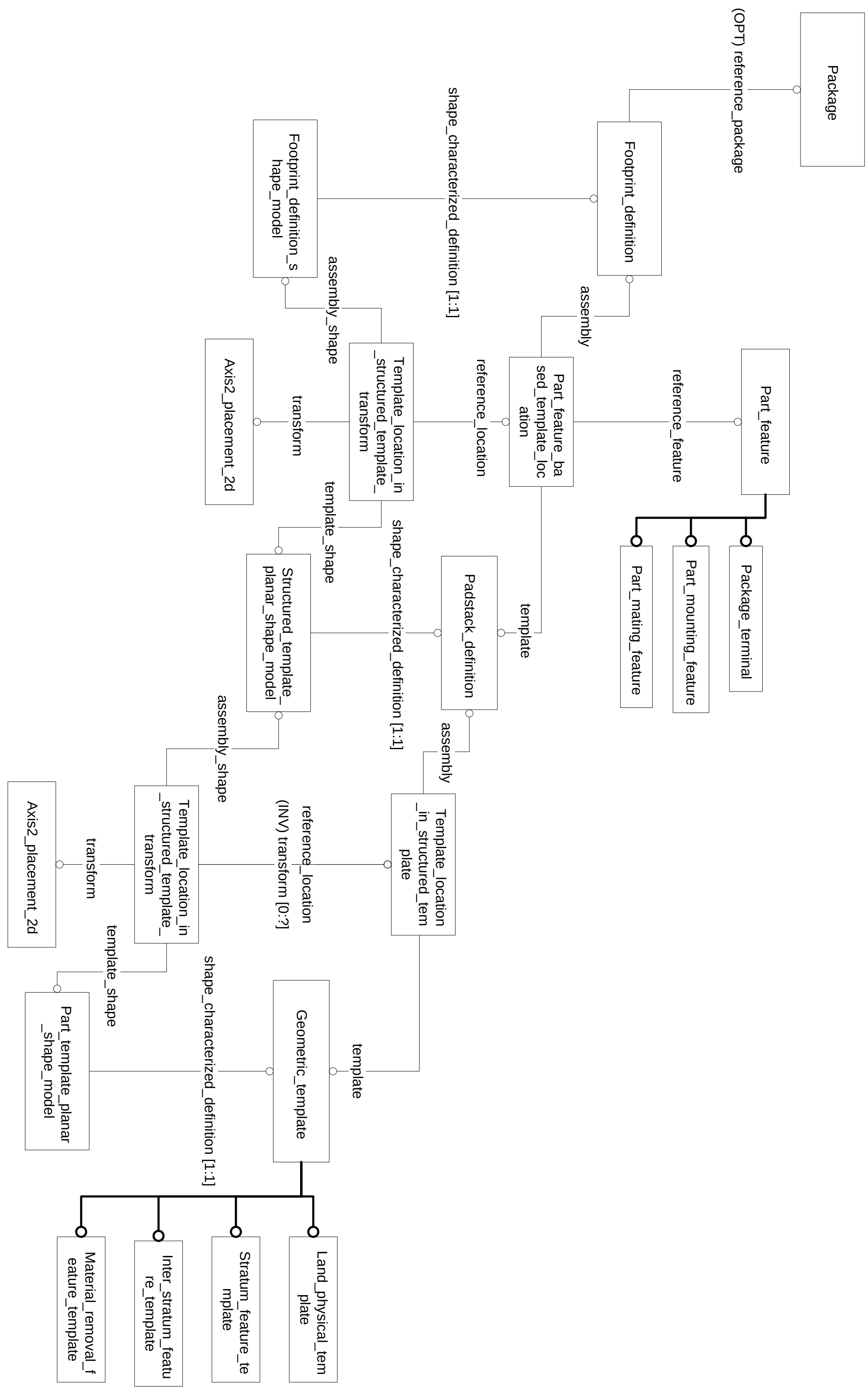
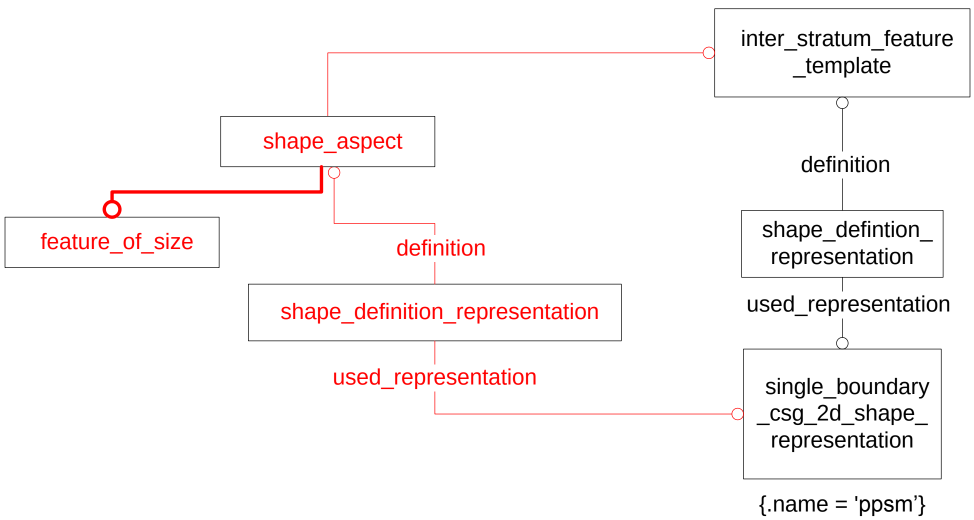


Figure 5. A Footprint_definition is a structured template that is composed of lower-level template elements, including Padstack_definition.

model, AP210 provides important mechanisms for explicitly linking the features of the



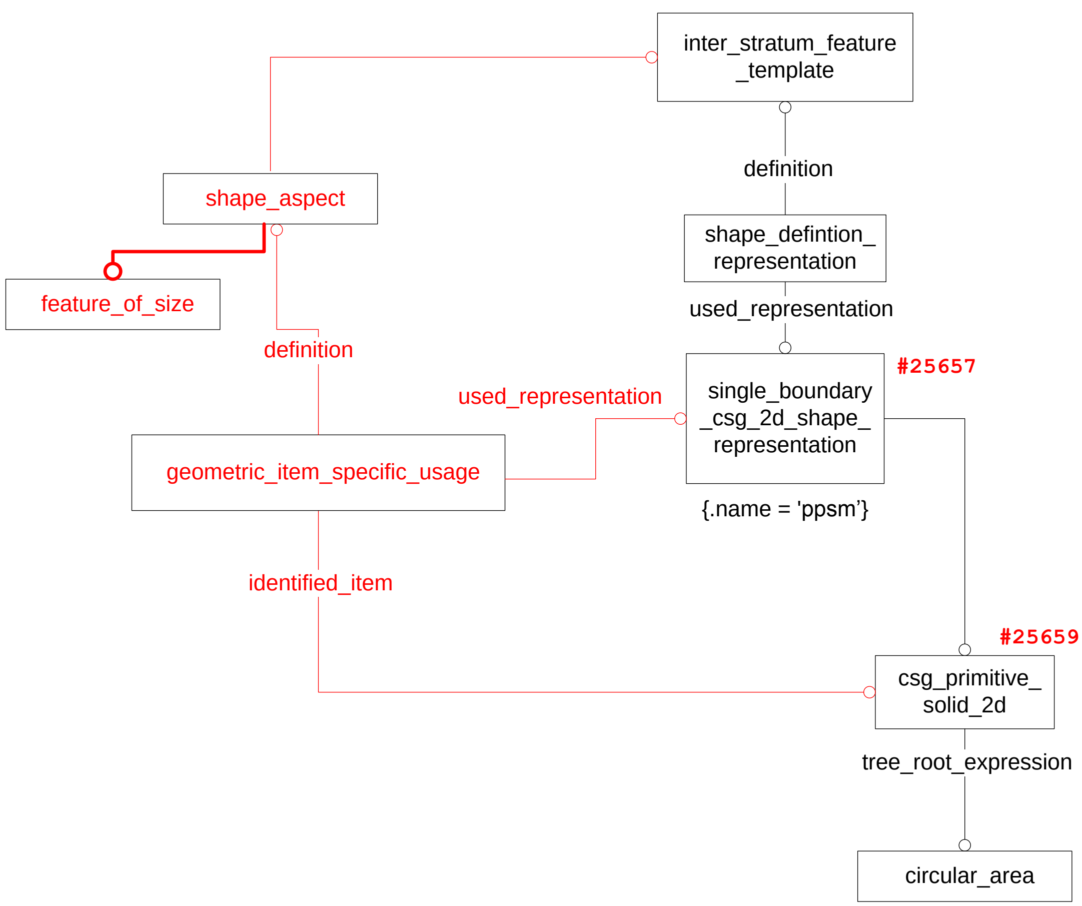


Figure 6. Alternate means of associating a shape_aspect with a csg_solid_2d in a geometric_template.

footprint with those of the package. The

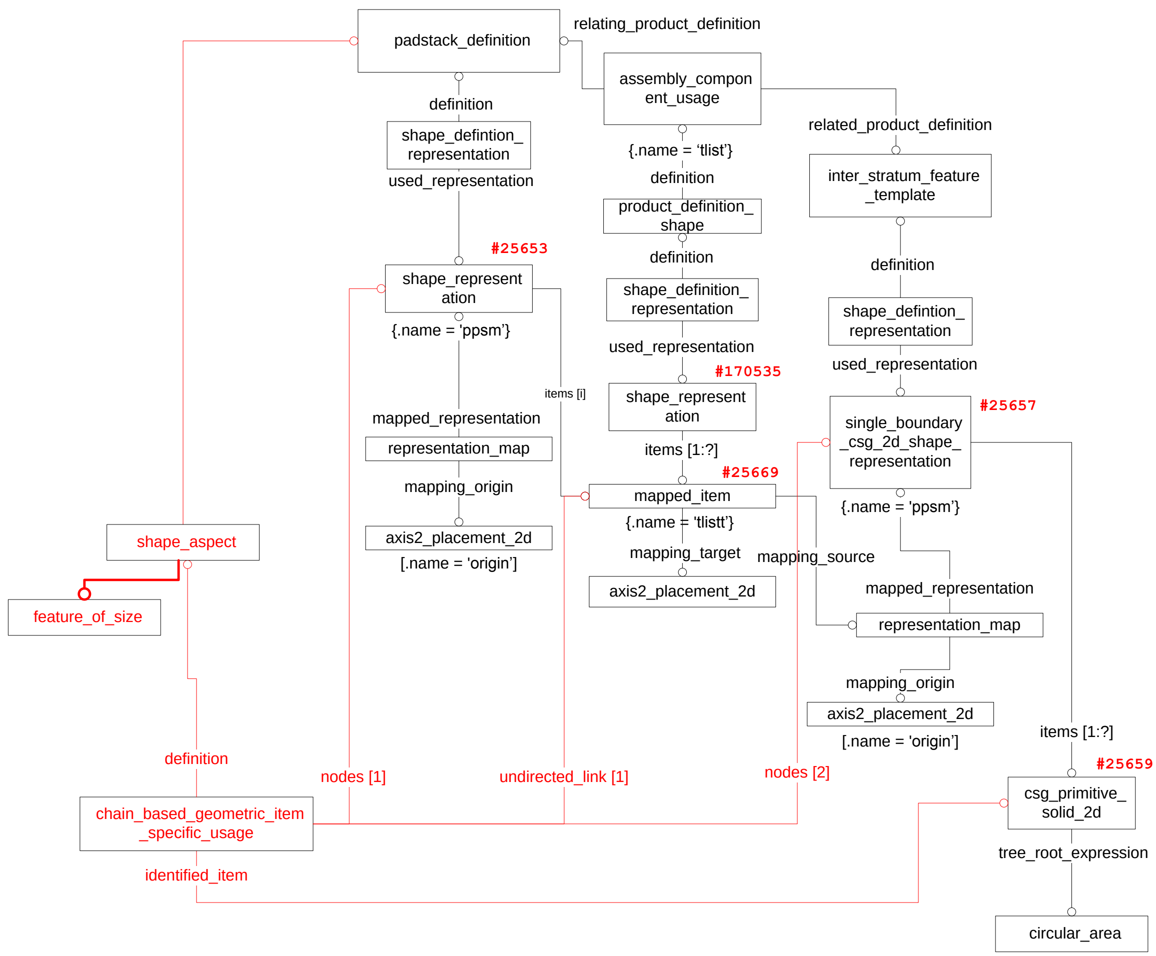


Figure 7. Associating a shape_aspect with an individual placement of a csg_solid_2d within a padstack_definition

`Part_feature_based_template_location` enables an explicit relationship to be established between a placement of a template (for example, a padstack) and the corresponding `Part_feature` of the package (for example, a terminal). A `Padstack_definition` is typically used to represent the collection of PCB elements needed to support the application of a particular part feature. For example, a surface mount terminal may be supported by a padstack containing a surface mount land, a soldermask cutout, and possible a solder paste mask. A through hole terminal may be supported by a padstack consisting of a plated passage (a through hole plated for electrical connectivity), and round pads and soldermask cutouts on both the top and bottom pcb surfaces. The elements of a padstack definition can be decomposed into single-stratum features (i.e. lands and soldermask cutouts) and inter-stratum features (holes, vias, etc.). Consider a common critical feature in the application of a connector – a mounting hole or plated through hole to accommodate a connector terminal. A drilled hole would be represented in a footprint as an `inter_stratum_feature_template` or a specialized subtype such as a `component_termination_passage_template`. It will commonly be desirable to annotate such a feature with either a dimensional tolerance

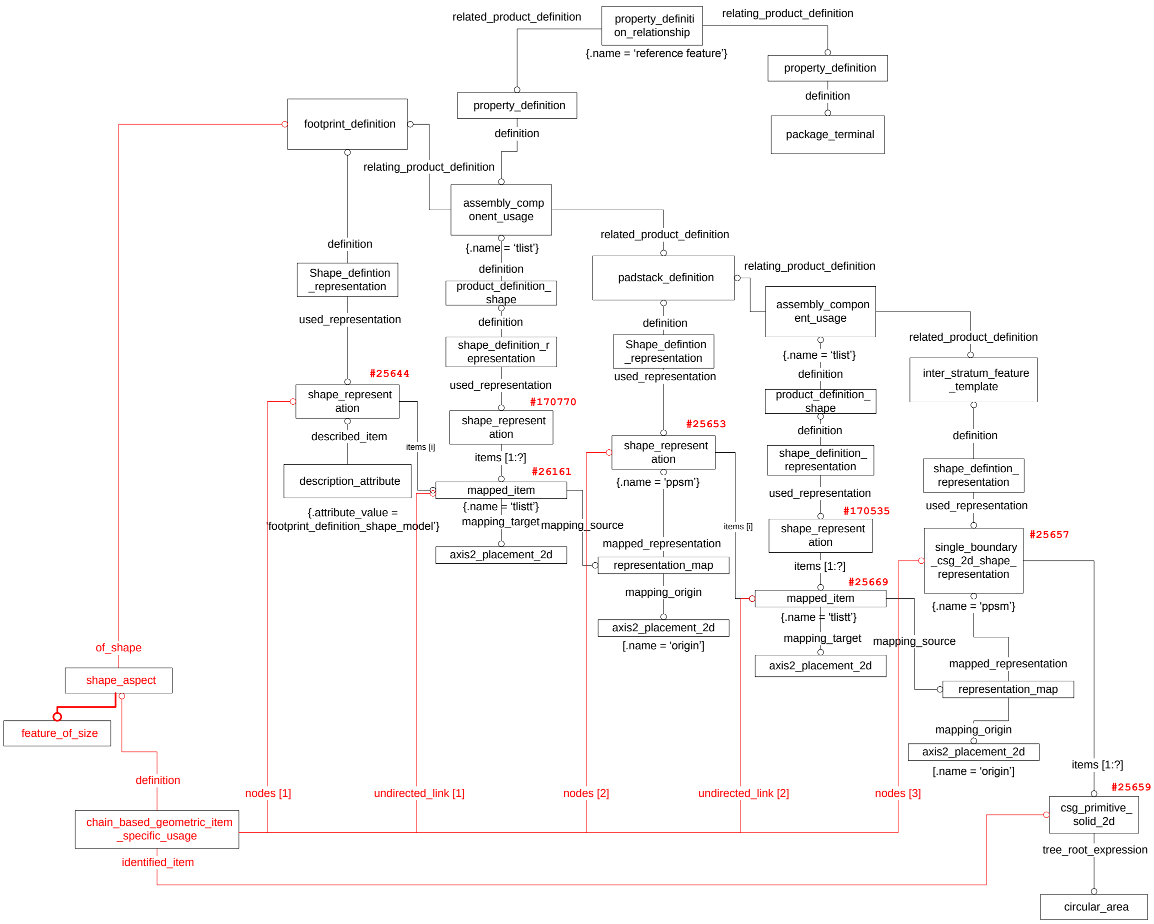


Figure 8. Associating a shape_aspect with an individual placement of a csg_solid_2d within a footprint_definition

(i.e. diameter) or a positional tolerance. A `shape_aspect` must be defined to support the GD&T annotation. Figure 6, Figure 7and Figure 8 illustrates various scenarios for the identification of a `shape_aspect` associated with a drilled hole within a `footprint_definition`.

The two scenarios illustrated in Figure 6 are functionally equivalent in this case, as the `shape_representation` of the template contains only the geometric specification of the hole's cross section (i.e. a circle). Either representation implies that a dimension or tolerance is associated with all occurrences of the template. This is a common and desirable scenario – often a series of equivalent holes are to be created to support a series of equivalent terminals. A common dimensional and/or locational tolerance would apply to each instance of the hole individually.

Figure 7 illustrates a significantly less common scenario. In this instance, the `shape_aspect` has been qualified by a specific hole placement within a `padstack_definition`. The associated GD&T annotation would apply to all occurrences of the padstack.

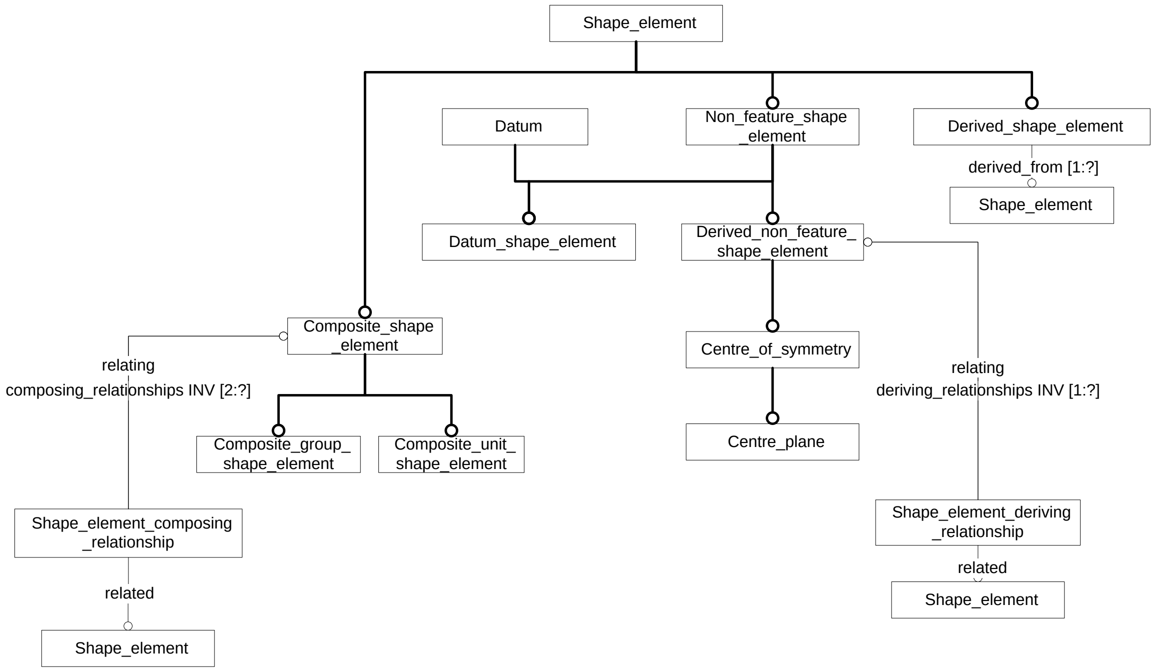


Figure 9. Several relevant subtypes of the Shape_element AO for GD&T

If it is necessary to call out a specific instance of the hole within a `footprint_definition` as a feature for GD&T association, a `chain_based_geometric_item_specific_usage` could be employed to qualify the individual inter-stratum feature within a specific placement of the padstack definition. The provides the ability to attach annotation that are specific to an individual hole, and would often be employed in the definition of a datum, for example (the datum feature is one particular hole).

Shape_elements for GD&T

Features of the model relevant for GD&T are represented by the `Shape_element` AO. Figure 9 shows several of the important subtypes of `Shape_element`. When it is desired to treat multiple disjoint regions as a single feature, a `Composite_shape_element` may be employed. A `Composite_shape_element` has two important subtypes – the `Composite_group_shape_element` and the `Composite_unit_shape_element`. A `Composite_unit_shape_element` is used to aggregate multiple `Shape_elements` that are to be treated as a unit. For example, multiple surfaces of a single feature-of-size. A `Composite_group_shape_element` is used when it is desired to apply a property to each of the constituent elements individually.

It is often common to require a reference to derived geometry, such as a center plane of a feature. Such a feature would be modeled through the appropriate subtype of `Derived_non_feature_shape_element` (a `Shape_element` that is not on the physical boundary of the part). Several examples for the representation of shape_elements in the context of a package model were previously documented in NISTIR 7634.

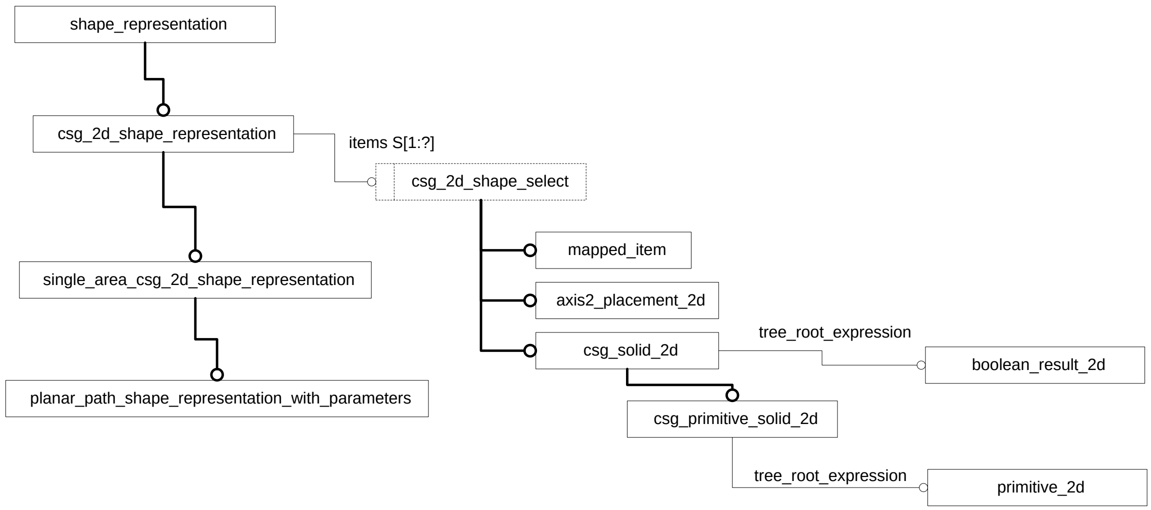


Figure 10. The 2D geometric elements of an ECAD footprint are typically represented with a csg_2d_shape_representation.

Feature-of-size

The representation of a "feature of size" is critical to the successful application and interpretation of many common dimensions and tolerances. Certain common GD&T annotations, such as a positional tolerance must be applied to a feature of size. A feature of size must have opposed points and contain a reproducible median point, axis, or plane. When a positional or perpendicularity tolerance is applied to a feature of size, it is controlling the feature's center point, axis, or plane. The most common feature of size is referred to as a "Regular Feature of Size," defined in the 2009 edition of the ASME Y14.5 standard as follows:

> *"one cylindrical or spherical surface, a circular element, and a set of two opposed parallel elements or opposed parallel surfaces, each of which is associated with a directly toleranced dimension."*

In the context of a three-dimensional B-Rep model of a solid, a feature of size would typically reference either a cylindrical face, or two parallel opposed planar faces. In a 3D B-Rep, there is typically no functional or feature-based decomposition of the geometric model available, and individual faces of a complex model would be called out through a `geometric_item_specific_usage` or its chain-based subtype.

In the context of two-dimensional ECAD geometry, developing a clear representation of a feature-of-size is critical. Within a footprint, there is a parallel structure between the footprint design elements and their geometric representation. The shape representations in a footprint definition are most commonly either a `csg_2d_shape_representation` or a subtype. The majority of individual geometric elements will be represented as a subtype of `primitive_2d`. Figure 11 details the common subtypes of `primitive_2d` and their key attributes.

The representation of the individual geometric elements are particularly relevant in the context of the feature of size. A circular feature of size can be unambiguously defined by

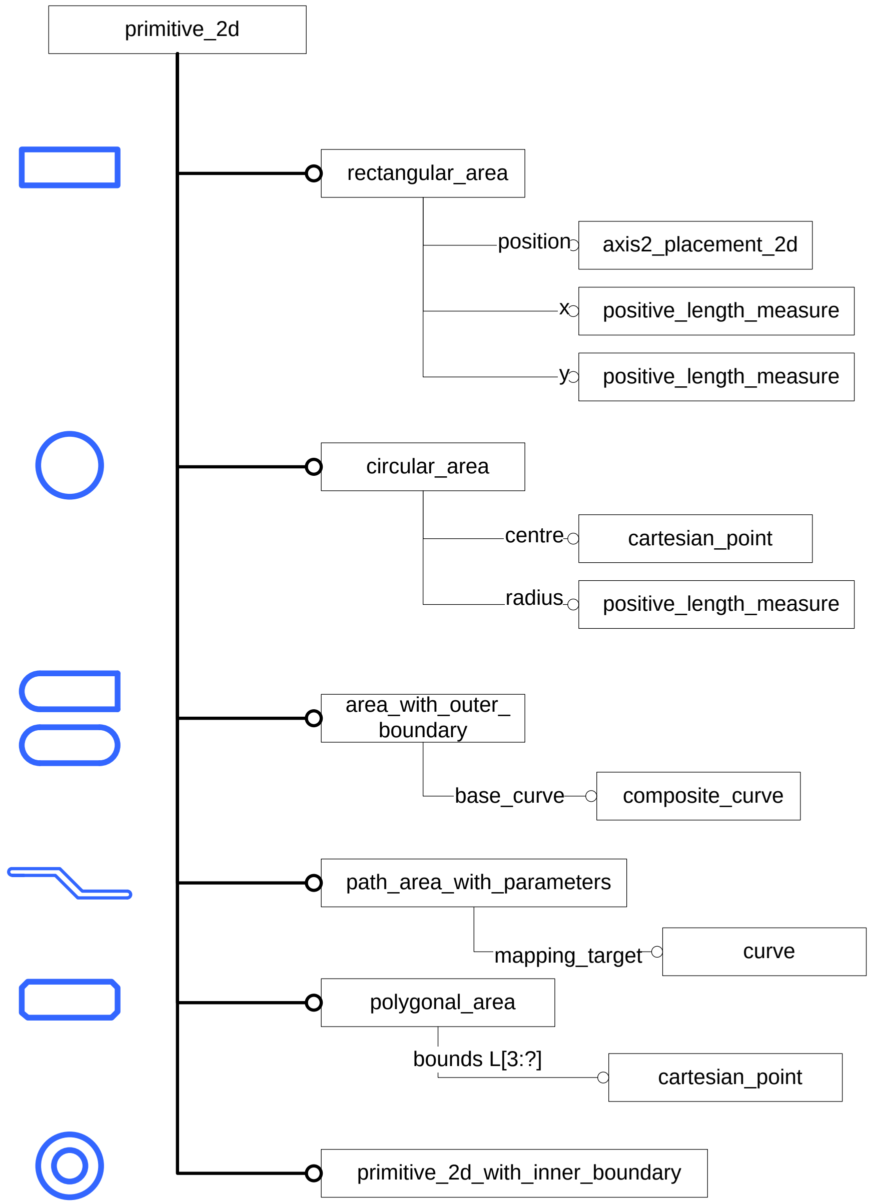


Figure 11. Common 2D geometric elements used in an ECAD footprint representation.

associating a `shape_aspect` with a `shape_representation` containing the `circular_area` entity. However, the `primitive_2d` representations do not support the unambiguous definition of a feature of size corresponding to two opposed parallel surfaces. Nevertheless, positional tolerances and length / width dimensional tolerances on the variety of common pad geometries shown in Figure 11 are very common.

To support this need, the addition of a subtype of `Shape_element` is proposed for consideration, that would enable the optional specification of an orientation for the feature of size.

```
ENTITY Feature_of_size
  SUBTYPE OF (Shape_element);
  direction_ratios : Optional LIST[2:3] OF length_measure;
END_ENTITY;
```

In the above, the optional direction_ratios enable specification of the orientation. If an orientation is provided, it is assumed that two opposed parallel surfaces are contained within the associated geometry normal to the specified direction.

Representation of geometric dimensions, tolerances, and datums.

The top-level ARM application objects and attributes used in the representation of a geometric dimension and tolerance are outlined in Figure 12 and Figure 14. For additional detail in the context of packaged components regarding the mapping of

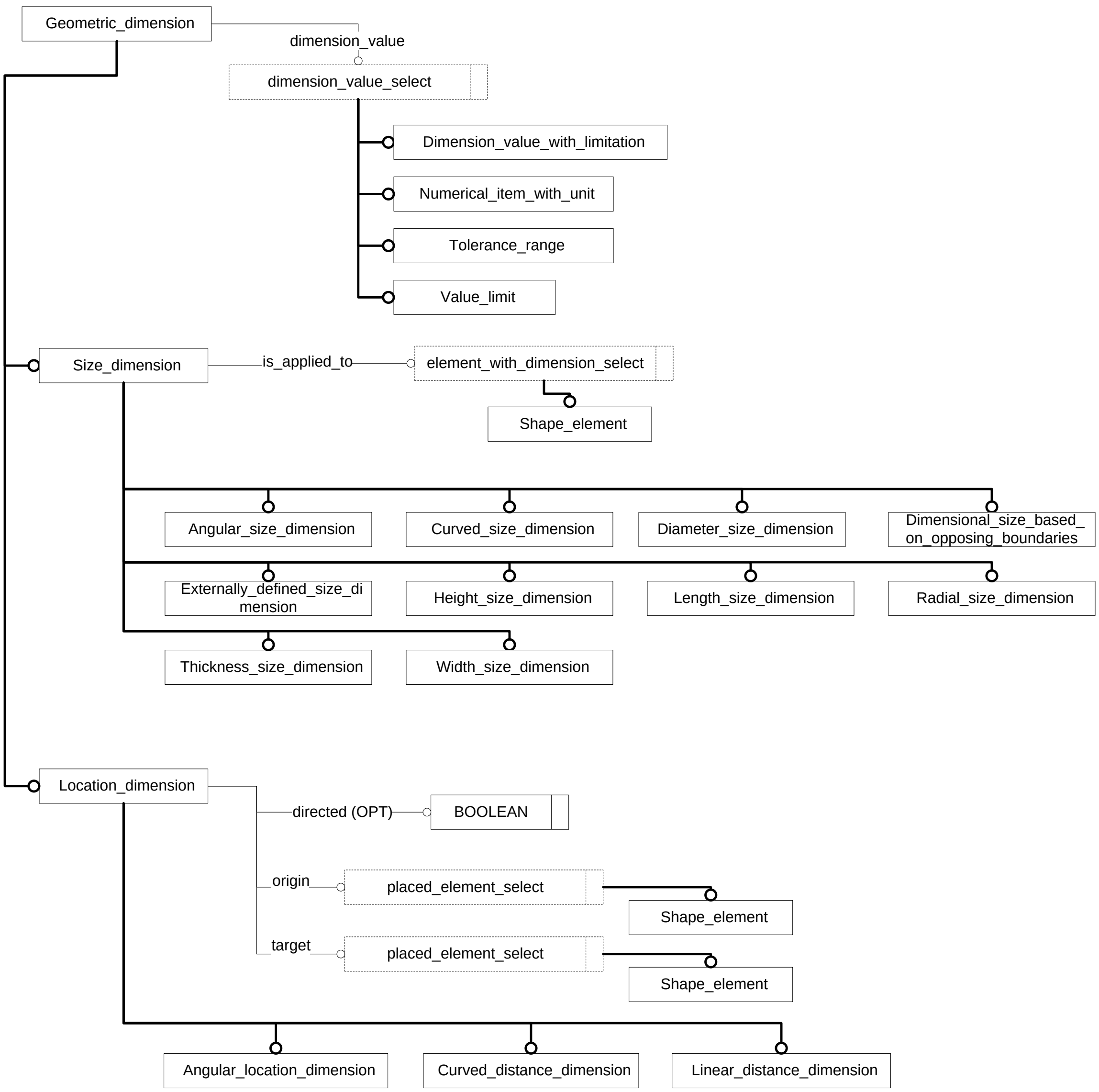


Figure 12. The top-level subtypes and key entities of the Geometric_dimension AO.

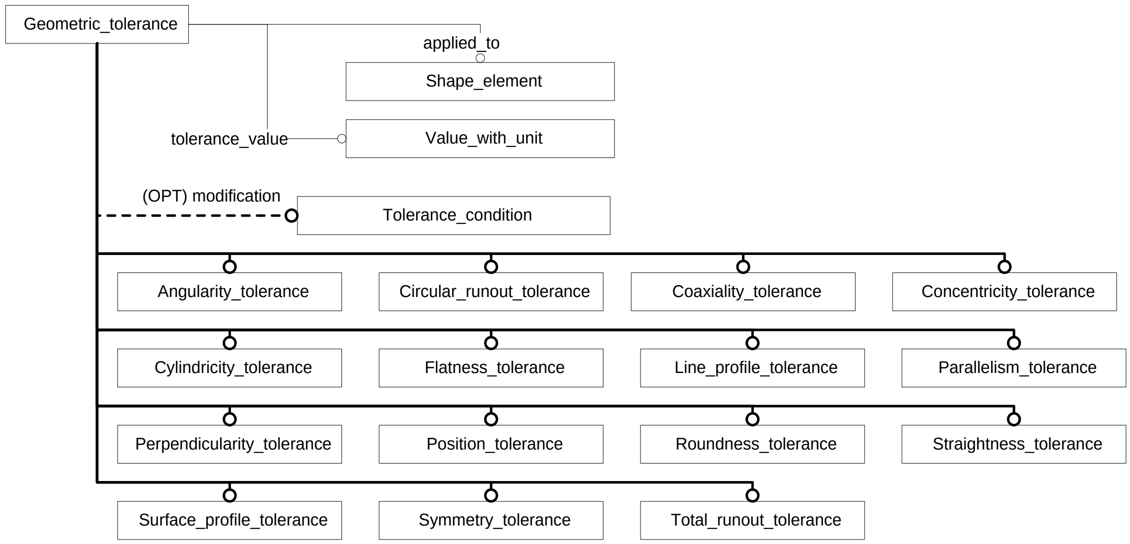


Figure 14. The top-level subtypes of the Geometric_tolerance AO.

common representations, refer to NISTIR 7634.

Per ASME Y14.5 (2009), a datum is a theoretically exact point, line or plane derived from geometric counterpart(s) on the physical model. The geometric counterpart is known as a datum feature. The AP210 ARM supports a family of application objects (AOs) that are subtypes of `Datum`. Several of the commonly used subtypes that are most relevant for the domain of packaged electronic components are documented in Figure 13.

In many cases, a datum feature will correspond to either a surface feature of a part or a "feature of size" of a part. When it is required that a specific region of a surface be used to determine the datum, a datum target may be used. Many times, a feature of size is used as a datum feature in defining a datum.

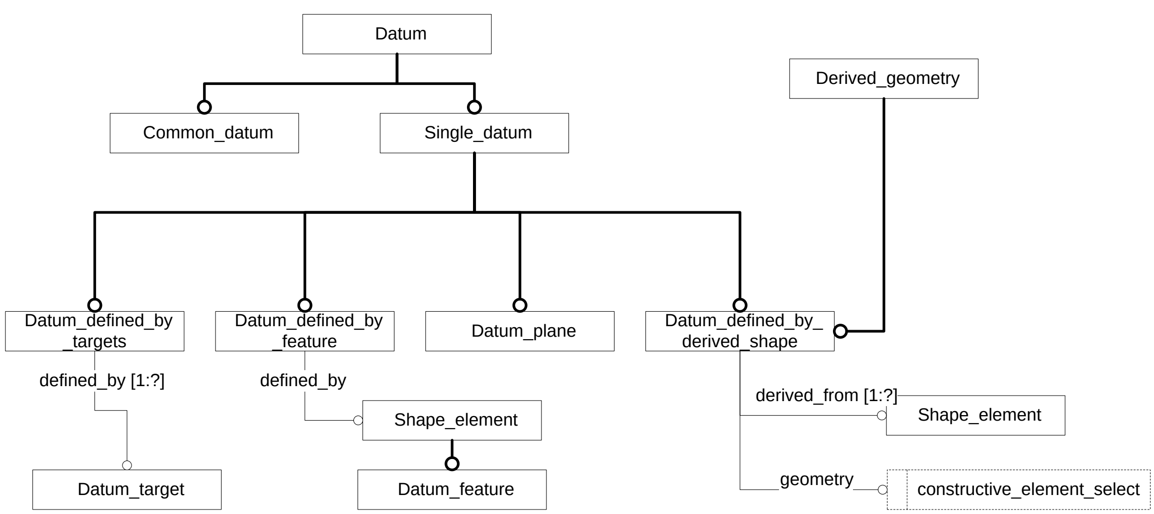


Figure 13. Selected subtypes of the Datum application object

Application to a common connector configuration

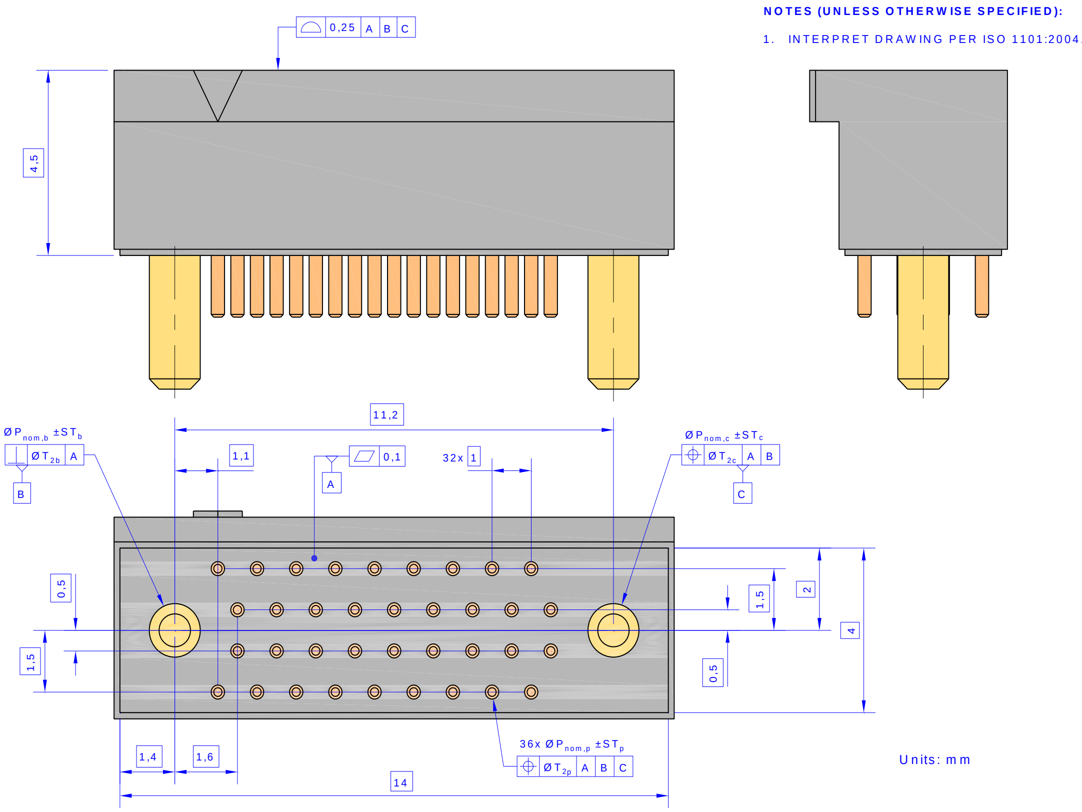
Figure illustrates a recommended GD&T based dimensioning and tolerancing scheme for a common connector and the corresponding PCB mating features. A connector vendor will often provide both tolerance specifications for the connector as well as the corresponding PCB layout. In the absence of a recommended PCB layout, the dimensions and tolerances for compatible mating features on the PCB can often be derived from the provided connector specifications. The assumptions, variables, and relationships needed for this example, based on simple tolerance stack-up analysis are provided in the Appendix.

Whenever feasible, it is recommended that the seating plane be treated as a primary datum plane. In this example, the connector locating pins serve as the secondary and tertiary datum features in the feature control frame for the connector, and the corresponding mating holes in the PCB are the secondary and tertiary datum features in the feature control frame for the PCB.

The detailed population of the GD&T annotations will be dependent on the geometric representation of the package and footprint. Assuming a single B-Rep model of the connector is provided by the vendor, datum A (coincident with seating plane) could be established with a `Datum_defined_by_feature` based on datum feature A – a `Shape_element` associated with the appropriate face of the geometric model through a `Geometric_item_specific_usage`. (a `datum` related to a `datum_feature` through a `shape_aspect_relationship`) Datum B and C could each be established through a `Datum_defined_by_derived_shape` (in this case, a `centre_of_symmetry`) with reference to the cylindrical face of the mounting pin. In the event that the two mounting pins shared a common geometric model, it would be necessary to employ a `chain_based_geometric_item_specific_usage` to (see Figure 4) discriminate between the two occurrences of the face.

In terms of the footprint definition, the two mounting holes would likely share a common padstack definition. A `shape_aspect` similar to that of Figure 8 would be employed to establish the secondary and tertiary datum features. There is no explicit geometric representation of the seating plane within the footprint definition. The primary datum feature should be established through reference to the seating plane.

It is desirable that the tolerance on the diameter of the two mounting holes in the PCB be symmetric. In this case, this tolerance could be associated directly with the `inter_stratum_feature_template` for the mounting holes (see Figure 6). A single padstack definition that incorporates this inter-stratum feature template could be instanced twice in the footprint definition for the two mounting holes. The perpedicularity and positional tolerance would rely on the same `shape_aspect` used to define the secondary and tertiary datum features. Finally, the positional tolerance of the 36 terminal mating holes could be associated directly with the `inter_stratum_feature_template` for the terminal mating holes.



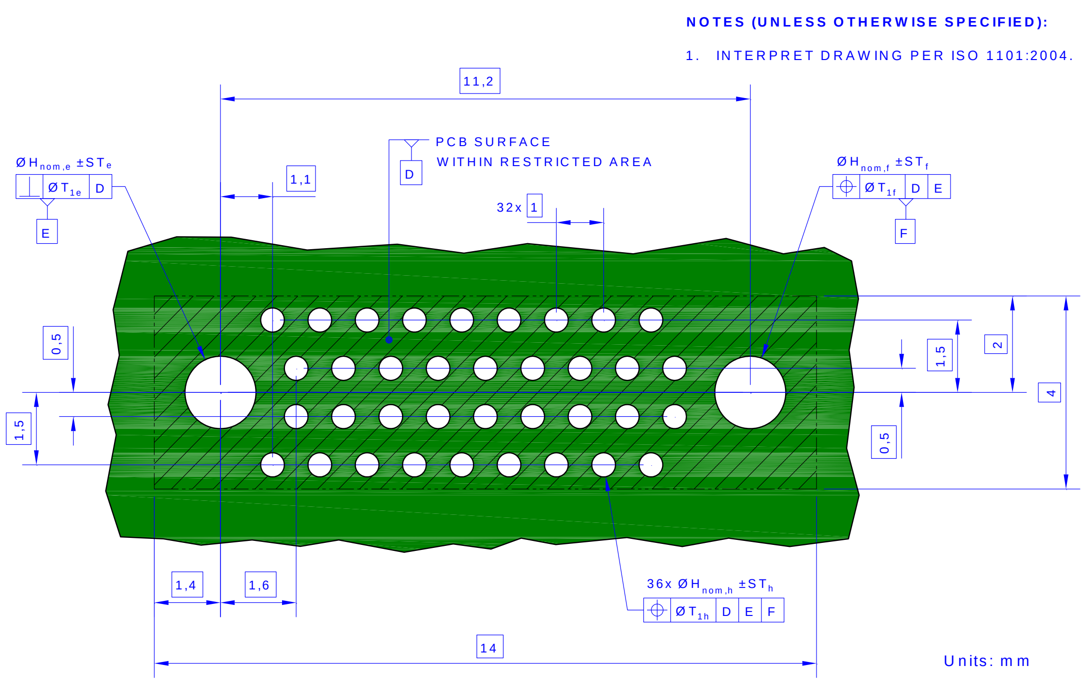


Figure Error!**. Representative connector with dimension and tolerance variables, and corresponding PCB layout.**
[Source: Bryan Fischer, Advanced Dimensional Management LLC]

Appendix – Assumptions, variables, and formulae related to the connector example of Figure .
[Source: Bryan Fischer, Advanced Dimensional Management LLC]

Assumptions:

- The connector is located by locator pins in mating holes in the PCB.
- The connector locating pins are referenced as the secondary and tertiary datum features in the positional tolerance feature control frame that controls the other (numbered) connector pins.
- A local, functional datum reference frame is established at each location a connector mates with the PCB.
- The holes in the PCB that mate with the connector's locating pins are referenced as the secondary and tertiary datum features in the positional tolerance feature control frame that controls the holes that correspond to the numbered pins on the connector.
- The secondary and tertiary datum features are the same size and have the same form, size, orientation, and location tolerances applied as applicable.
- The secondary and tertiary datum features on the connector engage the corresponding locating holes in the PCB at the same time and play an equal role in locating the connector.
- Maximum assembly shift is possible between the locating pins on the connector and the corresponding holes in the PCB.
- The assembly shift manifests itself at assembly as translational variation – rotational assembly shift is not addressed in the calculations. In this example, because the locating pins and holes are farther apart than the working pins and holes, the effect of rotational assembly shift will be less than the effect of translational assembly shift.
- The dimensioning and tolerancing schemes and calculation methods proposed here apply to other applications.
- Equal-bilateral size tolerances applied to all holes and pins.
- The pins on the connector and the holes in the PCB are cylindrical features.

Variables:

PCB

$H_{e,f}$	Hole (MMC, smallest) – Datum Features E and F on PCB
H_h	Hole (MMC, smallest) – PCB Holes that interface with numbered connector pins
$H_{nom,e,f}$	Hole (nominal, stated size) – Datum Features E and F on PCB
$H_{nom,h}$	Hole (nominal, stated size) – PCB Holes that interface with numbered connector pins
$H_{lmc,e,f}$	Hole (LMC, largest size) – Datum Features E and F on PCB
$ST_{e,f}$	Size Tolerance for Datum Features E and F on PCB
ST_h	Size Tolerance for PCB Holes that interface with numbered connector pins
$T_{1,e,f}$	Perpendicularity or Positional Tolerance applied to Datum Features E and F on PCB
$T_{1,h}$	Positional Tolerance applied to the PCB Holes that interface with numbered connector pins

Connector

$P_{b,c}$	Pin (MMC, largest) – Datum Features B and C on PCB
P_p	Pin (MMC, largest) – Numbered connector pins
$P_{nom,b,c}$	Pin (nominal, stated size) – Datum Features B and C on PCB
$P_{nom,p}$	Pin (nominal, stated size) – Numbered connector pins
$P_{lmc,b,c}$	Pin (LMC, smallest size) – Datum Features B and C on PCB
$ST_{b,c}$	Size Tolerance for Datum Features B and C on PCB
ST_p	Size Tolerance for numbered connector pins
$T_{2b,c}$	Perpendicularity or Positional Tolerance applied to Datum Features B and C on connector
$T_{2,p}$	Positional Tolerance applied to numbered connector pins

Assembly

AS	Assembly Shift (Maximum Clearance between $H_{lmc,e,f}$ and $P_{lmc,b,c}$)

Formulas:

Calculate and Verify Fit, Size, Size Tolerance, or Geometric Tolerances for Alignment Features: Corresponding Datum Features on Connector and PCB: Use Fixed Fastener Formula

$$H_{e,f} = P_{b,c} + T_{1,e,f} + T_{2,b,c}$$

Calculate and Verify Positional Tolerance for Holes and Numbered Pins: Working Features:
Use Modified Fixed Fastener Formula

$$H_h = P_p + T_{1,h} + T_{2,p} + AS$$

Calculate $H_{nom,e,f}$: Nominal Hole Size for the Datum Feature Holes on the PCB

$$H_{nom,e,f} = H_{e,f} + ST_{e,f}$$

Calculate $H_{lmc,e,f}$: Largest Size for Datum Feature Holes on the PCB

$$H_{lmc,e,f} = H_{nom,e,f} + ST_{e,f}$$

Calculate $P_{nom,b,c}$: Nominal Pin Size for the Datum Feature Pins on the Connector

$$P_{nom,b,c} = P_{b,c} - ST_{b,c}$$

Calculate $P_{lmc,b,c}$: Smallest Datum Feature Pin Size for Connector

$$P_{lmc,b,c} = P_{nom,b,c} - ST_{b,c}$$

Calculate AS: Assembly Shift: Worst-Case Variation between LMC Datum Features on PCB and Connector

$$AS = H_{lmc,e,f} - P_{lmc,b,c}$$

Calculate $H_{nom,h}$: Nominal Size for Working Holes in PCB (Correspond to Numbered Pins on Connector)

$$H_{nom,h} = H_h + ST_h$$

Calculate $P_{nom,p}$: Nominal Size for Numbered Pins on Connector

$$P_{nom,p} = P_p - ST_p$$