VDA working group “CAD/CAM”

VDA KBL

Harness Description List (KBL)

Version *2.3 SR-1*
State *accepted by VDA AG Car Electric*
Autor(s) *Dr. Max Ungerer (PROSTEP AG), Olav Rabe (4Soft GmbH)*

Content

Diagrams

Package: ::KBL

This UML model defines how electrical/logical data coming from CAE systems (usages of components, connections, etc.) and topological data coming from 3D CAD systems (routing of cables and bundles) should be represented in an integrated way:
- Part identification including versioning, references to car projects or supplier information
- Harness, variants and options
- Modules to support modular design
- Harness components like connectors, wires, fixings, grommets or accessories
- Connectivity lists
- Part usage lists
- Topology and routing
- References to drawings or manufacturing documents

For various reasons, there may be some classes that cannot be completely populated in an implementation (e.g. XML file). Sometimes an application may not maintain all the information that is anticipated for the data exchange. Other times, the information may be maintained by a sending system but not included in the data exchange. Never the less, the interface must provide values for all mandatory attributes in an exchange file. When no data is provided by a sending system for a string value, the interface shall use '/NULL' or the empty string ''.
To further indicate the reason why no data is provided, the following convention shall be used:
- Empty string '' indicates user data managed by the sending system but not provided for data exchange;
- String '/NULL' indicates user data in a mandatory attribute that is not managed by the sending system or currently not known;

In general, the empty null string '' or the default string '/NULL' shall not be used as valid user data.

Package: ::KBL::1_Harness

This package specifies the information needed to describe the exchange context, i.e. the harness with its decomposition into modules. It also includes the information needed to build variants and control the individual components with respect to options and logistic control information.

Classes

Class: Connection_or_occurrence

A Connection_or_occurrence is an object that is controlled by a Module or a Harness.

Class: Harness

Kabelbaum
All wires in the car need to connect the ECU's, sensors, switches, and actuators. The harness depends on the car configuration.
A Harness is an assembly of insulated conductors formed to a predetermined pattern or configuration.

Attributes

public Content
The content specifies the information covered by the Harness.
The following values shall be used:
- 'Harness Complete Set'
- 'Harness Subset'

Associations

Components: Connection_or_occurrence [0..*]
The Components specifies the part occurrences and connectiosn which belong to the Harness.

Harness_configuration: Harness_configuration [0..*]
The harness_configuration specifies the configurations of the Harness.

Module: Module [0..*]
The module specifies the modules belonging to the harness.
Note: A Module belong to one harness, only. This is no restriction, as there can be only one harness object exchanged by one file.

Module_configuration: Module_configuration [0..*]
The module_configuration specifies the direct control of components outside of modules.

Module_families: Module_family [0..*]

Class: Harness_configuration

Kabelbaumvariante
A superset of all available functions (150%) is described in function diagram. The harness configuration describes the harness of useful car configurations.
A Harness_configuration is a variant of an harness.

Attributes

public Logistic_control_information [0..1]
The logistic_control_information specifies the calculated combination of the configuration codes reflecting customer, market or country requirements associated with a Harness_configuration.
Example: 'LOL/LOR+CFL'

Associations

Modules: Module [1..*]
The modules specifies the Module objects which build up the Harness_configuration.

Class: Harness_content

Attributes

public harness complete set

public harness subset

Class: KBL_container

The KBL_container is introduced to specify the information which can be exchanged by one file.

Associations

Accessory: Accessory [0..*]
The accessory specifies the Accessory objects contained in KBL_container.

Approval: Approval [0..*]
The approval specifies the Approval objects contained in KBL_container.

Assembly_part: Assembly_part [0..*]
The assembly_part specifies the Assembly_part objects contained in KBL_container.

Cartesian_point: Cartesian_point [0..*]

Cavity_plug: Cavity_plug [0..*]
The cavity_plug specifies the Cavity_plug objects contained in KBL_container.

Cavity_seal: Cavity_seal [0..*]
The cavity_seal specifies the Cavity_seal objects contained in KBL_container.

Co_pack_part: Co_pack_part [0..*]
The co_pack_part specifies the Co_pack_part objects contained in KBL_container.

Component: Component [0..*]
The component specifies the Component objects contained in KBL_container.

Connector_housing: Connector_housing [0..*]
The connector_housing specifies the Connector_housing objects contained in KBL_container.

Creation: Creation [0..*]
The creation specifies the Creation objects contained in KBL_container.

Dimension_specification: Dimension_specification [0..*]

External_reference: External_reference [0..*]
The external_reference specifies the External_reference objects contained in KBL_container.

Fixing: Fixing [0..*]
The fixing specifies the Fixing objects contained in KBL_container.

General_terminal: General_terminal [0..*]
The general_terminal specifies the General_terminal objects contained in KBL_container.

General_wire: General_wire [0..*]
The general_wire specifies the General_wire objects contained in KBL_container.

Harness: Harness [0..1]
The harness specifies the Harness objects contained in KBL_container.

Node: Node [0..*]
The node specifies the Node objects contained in KBL_container.

Routing: Routing [0..*]
The routing specifies the Routing objects contained in KBL_container.

Segment: Segment [0..*]
The segment specifies the Segment objects contained in KBL_container.

Unit: Unit [0..*]
The unit specifies the Unit objects contained in KBL_container.

Wire_protection: Wire_protection [0..*]
The wire_protection specifies the Wire_protection objects contained in KBL_container.

Class: Module

A Module is a physical part of harness electrically defined by one or more module groups including required harness furniture.

Attributes

public Content
The content specifies the information covered by the Module.
The following values shall be used:
- 'variant'
- 'module'

Associations

Module_configuration: Module_configuration [1]
The module_configuration specifies the Module_configuration the Module represents.

Of_family: Module_family [0..1]
The of_family specifies the Module_family to which the Module belongs to.

Class: Module_configuration

A Module_configuration is a grouping of wires and components fulfilling a specific functionality of a harness.

Example:
Option: Audio System with CD Changer

Attributes

public Configuration_type [0..1]
The configuration_type specifies further information on the type of the logistic_control_information.
Note: To control completion parts which are only used if a specific combination of modules occur, a Module_configuration can be used with a logistic_control_information containing the Boolean expression for the combination and a configuration_type 'module list'.
Where applicable the following values shall be used:
- 'option code': the logistic_control_information contains configuration codes reflecting customer, market or country requirements
- 'module list': the logistic_control_information contains a list of modules, to which a completion part controlled by the Modul_configuration belongs to

public Logistic_control_information
The logistic_control_information specifies the calculated combination of the configuration codes reflecting customer, market or country requirements, or a list of modules associated with a Module_configuration. The meaning of the string is further described by the configuration_type.
Example: 'LOL/LOR+CFL'

Associations

Controlled_components: Connection_or_occurrence [0..*]
The controlled_components specifies the components which are grouped by the Module_configuration.

Class: Module_configuration_type

Attributes

public module list

public option code

Class: Module_content

Attributes

public module

public variant

Class: Module_family

Funktionsfamilie
Description of a set of modules.
A Module_family is a mechanism to group mutually exclusive modules.
EXAMPLE "audio equipment"

Attributes

public Description [0..1]
The description specifies additional information about the Module_family.

public Id
The id specifies the identifier of the Module_family.

Package: ::KBL::2_Parts

This package specifies the part master information of the harness components. It defines the attributes

necessary to provide further information on parts like wires or accessories. It further provides the capability to describe the information, which is common to all kind of parts. This includes basic attributes like part number or mass and also more complex descriptions like material or change information.

Classes

Class: Accessory

Zubehör
Components which may be used with connectors to facilitate their use in a wide range of applications. Accessories perform no electrical function and include such items as caps, covers, gaskets washers and boots.
An Accessory is any supplementary portion of a connector with the obligation to help a Harness to perform its function.
Example: An Accessory may be a sleeve, a cap, a cable strap or comparable parts, which are installed to a plug.

Attributes

public Accessory_type [0..1]
The accessory_type specifies the type of an Accessory.
Note: There are no values pre-defined. Special values have to be negotiated between exchange partners.

Class: Assembly_part

Teilmontiertes Bauteil
An assembly of multiple parts identified by a common part number.
An Assembly_part is a component that contains other subordinate objects. An Assembly_part is ordered as a part to be bought under a part number, due to limited functionalities of the CAD systems these must be described however with several individual parts.

Attributes

public Part_type [0..1]
The part_type specifies the type of an Assembly_part.
Note: There are no values pre-defined. Special values have to be negotiated between exchange partners.

Associations

Components: Connection_or_occurrence [0..*]
The components specifies the Parts of which the Assembly_part conists.

Class: Cavity

Kontaktkammer
A cavity is a defined space in a housing for location of an electrical terminal or cavity plug/seal (can be empty).

Attributes

public Cavity_number
The cavity_number specifies the identifier of the Cavity.
Note: The uniqueness of a cavity id within a harness is fulfilled by the concatenation of the connector id, the slot id, and the cavity id.

Class: Cavity_plug

Blindstopfen
Connector accessory to fill and seal empty cavities

A Cavity_plug is a water tight non-electrical object to fill an empty cavity.

Attributes

public Colour [0..1]
The colour specifies the colour of the Cavity_plug.

public Plug_type [0..1]
The plug_type specifies the type of a Cavity_plug.
Note: There are no values pre-defined. Special values have to be negotiated between exchange partners.

Class: Cavity_seal

Kontaktkammerversiegelung
Connector accessory to fill and seal a populated cavity.
A Cavity_seal is a water tight non-electrical object to fill a populated Cavity.

Attributes

public Colour [0..1]
The colour specifies the colour of the Cavity_seal.

public Seal_type [0..1]
The seal_type specifies the type of a Cavity_seal.
Note: There are no values pre-defined. Special values have to be negotiated between exchange partners.

public Wire_size [0..1]
The wire_size specifies the size range of the wires the seal fits.

Class: Co_pack_part

Kabelbaumzubehör
A Co_pack_part is a Part which is supplied and installed with the wiring harness, but without any electrical connection.

Attributes

public Part_type [0..1]
The part_type specifies the type of a Co_pack_part.
Note: There are no values pre-defined. Special values have to be negotiated between exchange partners.

Class: Component

A Component is an object which belongs to the harness.
Example: Fuse, fusebox, relay sockets

Class: Connector_housing

Steckergehäuse
Body of the connector with 1-n cavities.
A Connector_housing is a non populated connector, i.e. without addressed/populated cavities.
A Connector_housing without any slot allows for the representation of a connector shell (Mehrfachstecker)

Attributes

public Housing_code [0..1]
The housing_code specifies a coding for the type of the Connector_housing.

public Housing_colour [0..1]
The housing_colour specifies the colour of the Connector_housing.

public Housing_type [0..1]
The housing_type specifies the type of a Connector_housing.

Example: family series type like 'MQS2.8'
Note: There are no values pre-defined. Special values have to be negotiated between exchange partners.

Associations

Slots: Slot [0..*]
The slots specifies the Slots which belong to the Connector_housing.

Class: Core

Kabelader
A core is part of a cable. A cable consists of 1-n cores
A Core is a single conductor of a multi-core wire including its isolation.

Attributes

public Bend_radius [0..1]
The bend_radius specifies the bend radius of a core.

public Cable_designator [0..1]
The cable_designator specifies additional information to refer to a Core.

public Cross_section_area
The cross_section_area specifies the electrical cross section of the Core.

public Id
The id specifies the identifier of the Core.

public Outside_diameter [0..1]
The outside_diameter specifies the outer width of the Core.

public Wire_type [0..1]
The wire_type specifies the type of a Core.
Note: There are no values pre-defined. Special values have to be negotiated between exchange partners.
Example: 'protected wire', 'flat band'.

Associations

Core_colour: Wire_colour [1..*]
The Core_colour specifies the Wire_colour objects of the core.

Class: Fixing

Kabelbefestigung
A Fixing is an accessory part of the harness, used to fix the position. Fixing elements cover all parts, that are used in cable section.
Example: clips, sleeves, cable ducts, grommet, etc.

Attributes

public Fixing_type [0..1]
The fixing_type specifies the type of a Fixing.
Note: There are no values pre-defined. Special values have to be negotiated between exchange partners.
Example: Examples are 'fastening part', 'channel', 'sleeve', 'shaft', 'conduct', 'pine-tree'

Class: General_terminal

Kontakt
A General_terminal is a device designed to terminate a conductor to be affixed usually to a post, stud, chassis, or other conductor or the like in order to establish electrical connection.
Note: A General_terminal describes the active part of the connector, which connects electrical, equal to "contact".

Attributes

public Cross_section_area [0..1]
The cross_section_area specifies the electrical cross section, which can be accommodated by the General_terminal.

public Outside_diameter [0..1]
The outside_diameter specifies the outer width of the isolation, which can be accommodated by the General_terminal.

public Plating_material [0..1]
The plating_material specifies the overlaying of a thin coating of metal on components to improve conductivity, provide for easy soldering or prevent rusting or corrosion.

public Terminal_type [0..1]
The terminal_type specifies the type of a General_terminal.
Note: There are no values pre-defined. Special values have to be negotiated between exchange partners.
Example: 'pin', 'Blade', 'male', 'female', 'grease', 'FL - flat contact', 'KK - box contact', 'SK - ring wire', 'RK - ring lug', 'KS - lug', 'BK - battery clamp', 'OL - open line end of frame', 'SP - splice', 'MK - multiple contact'

Class: General_wire

A General_wire is a physical wire, performing electrical connection. A General_wire can either be used to define a single wire or a multi-core wire.

Attributes

public Bend_radius [0..1]
The bend_radius specifies the bend radius of a wire.

public Cable_designator [0..1]
The cable_designator specifies additional information to refer to a General_wire.

public Cross_section_area [0..1]
The cross_section_area specifies the electrical cross section of the General_wire.

public Outside_diameter [0..1]
The outside_diameter specifies the outer width of the General_wire.

public Wire_type [0..1]
The wire_type specifies the type of a General_wire.
Note: There are no values pre-defined. Special values have to be negotiated between exchange partners.
Example: 'individual wire', 'multi-core wire', 'protected wire', 'flat band'.

Associations

Core: Core [0..*]
The Core specifies the constituents of the multi-core wire.

Cover_colour: Wire_colour [1..*]
The cover_colour specifies the Wire_colour objects of a General_wire. In the case of a single wire it specifies the colour of the conductor. In case of a multi-core wire it specifies the colour of the cover.

Class: Part

Bauteil, Komponente
A part is an element of a product relevant for a bill-of material.

Attributes

public Abbreviation
The abbreviation specifies a short name for a Part.

public Alias_id [0..*]
The alias_id specifies an additional part_number that is used to identify the Part in another organizational context (e.g. company).

public Company_name

The company_name specifies the name of the organizational context in which the Part_number is defined.

public Copyright_note [0..1]
The copyright_note specifies copyright information for a Part.

public Degree_of_maturity [0..1]
The degree_of_maturity specifies the degree of maturity of a Part.
Where applicable the following values shall be used:
- 'draft'
- 'planning'
- 'equipment order'
- 'disposition'

public Description
The description specifies additional information about the Part.

public Mass_information [0..1]
The mass_information specifies the mass of a Part.
Example: Valid values for the unit of the Numerical_value specifying the mass are 'gram', 'kilogram', or also 'kg/piece', 'gram/meter'

public Part_number
The part_number specifies the identifier of the Part. The format of the part number is user defined (OEM specific).

public Predecessor_part_number [0..1]
The predecessor_part_number specifies the part number of the predecessor of the Part.

public Version
The version specifies the version identifier of the Part. A version cumulates and consolidates one or more single changes.

Associations

Change: Change [0..*]
The change specifies the change history of the Part .

External_references: External_reference [0..*]
The external_references specifies references to documents.

Material_information: Material [0..1]
The material_information specifies the material of a Part.

Processing_information: Processing_instruction [0..*]
The processing_information provides additional information for the processing of the Part.

Class: Part_with_title_block

A Part_with_title_block is a mechanism to assign additional information to a Harness, a Harness_configuration or a Module.

Attributes

public Car_classification_level_2
The car_classification_level_2 provides a classification according to "CC8 Recommended Practices Specification and Configuration, Product Structures". Car classification is the identification of a set of similar cars to be offered to the market. Level 2 stands for "Technical information / platform" and reflects the level of a product class in a BoM system which represents a main technical product base (e.g. project, platform, engineering series etc.). In some cases this level carries a complete BoM ("Maximum BoM") for a project, platform, engineering series etc. This level is in some cases called technical documentation.

public Car_classification_level_3 [0..1]
The car_classification_level_3 provides a classification according to "CC8 Recommended Practices Specification and Configuration, Product Structures". Car classification is the identification of a set of similar cars to be offered to the market. Level 3 stands for "Configuration information / product family" where all

variant control mechanism are attached.

public Car_classification_level_4 [0..1]
The car_classification_level_4 provides a classification according to "CC8 Recommended Practices Specification and Configuration, Product Structures". Car classification is the identification of a set of similar cars to be offered to the market. Level 4 stands for "Furthest pre-configured abstract product class" and represents the furthest configured class of a product, which is not yet a real product. E.g. this could be a complete vehicle, engine, gear-box etc. which has not been evaluated against customer special choices or a abstract vehicle, engine, gear-box etc. which could become a real one after the associated BoM is evaluated. The purpose of this level of a product class instance is in any case to reflect that level of product class of a BoM system which leads to the individual BoM for a single product.

public Model_year
The model_year specifies the year of the car model.

public Project_number [0..1]
The project_number specifies the development order number (car or engine project)

Class: Slot

A Slot is a mechanism to group the Cavity objects of a Connector_housing.

Attributes

public Id [0..1]
The id specifies the identifier of the Slot.

public Number_of_cavities
The number_of_cavities specifies the number of cavities associated with the Slot.

Associations

Cavities: Cavity [1..*]
The cavities specifies the Cavity objects belonging to the Slot.

Class: Wire_protection

Kabelschutzmantel
A Wire_protection is a mechanism to describe harness wrappings, a shield to prevent the wire from damaging.. It covers all kinds of wrappings modeled by CAD systems.

Attributes

public Protection_type [0..1]
The protection_type specifies the type of the Wire_protection.
Note: There are no values pre-defined. Special values have to be negotiated between exchange partners.
Example: 'coil', 'corrugated pipe', 'foam rubber strip'

public Type_dependent_parameter [0..1]
The type_dependent_parameter specifies further information dependent on the type of the Wire_protection.
Note: There are no values pre-defined. Special values have to be negotiated between exchange partners.
Example: for corrugated pipe: wave shape, for pull-push rule: width.

Package: ::KBL::3_Part_structure

This package provides the information typically contained in bill of material lists. It describes how a harness is assembled.

Classes

Class: Accessory_occurrence

An Accessory_occurrence is the occurrence of an Accessory in a module part list.

Attributes

public Alias_id [0..*]
The alias_id specifies additional identifiers for the Accessory _occurrence.

public Description [0..1]
The description specifies additional information about the object.

public Id
The id specifies the identifier of the Accessory_occurrence.

public Placement [0..1]
The placement specifies the transformation information, which is used to locate and orient the occurrence in the car coordinate system. For further information see [CESAK].

Associations

Part: Accessory [1]
The part specifies the Accessory that serves as a definition for this particular occurrence.

Reference_element: Connector_occurrence [0..*]
The reference_element specifies the Connector_occurrence objects supplemented by the Accessory_occurrence.

Installation_information: Installation_instruction [0..*]
The installation_information provides additional information for the installation of the Accessory_occurrence.

Class: Assembly_part_occurrence

An Assembly_part_occurrence is the occurrence of an Assembly_part in a module part list.

Attributes

public Alias_id [0..*]
The alias_id specifies additional identifiers for the Assembly_part _occurrence.

public Description [0..1]
The description specifies additional information about the ob ject.

public Id
The id specifies the identifier of the Assembly_part_occurrence.

public Placement [0..1]
The placement specifies the transformation information, which is used to locate and orient the occurrence in the car coordinate system. For further information see [CESAK].

Associations

Part: Assembly_part [1]
The part specifies the Assembly_part that serves as a definition for this particular occurrence.

Installation_information: Installation_instruction [0..*]
The installation_information provides additional information for the installation of the Connection.

Class: Cavity_plug_occurrence

A Cavity_plug_occurrence is the occurrence of a Cavity_plug in a module part list.
Note: Cavity_plugs do not show up in a module bill of material.

Associations

Part: Cavity_plug [1]

The part specifies the Cavity_plug that serves as a definition for this particular occurrence.

Class: Cavity_seal_occurrence

A Cavity_seal_occurrence is the occurrence of a Cavity_seal in a module.
Note: The number of occurrences used in a specific module or harness can be calculated by the individual occurrences.
Note: The usage of a Cavity_seal_occurrence for a particular Cavity is specified by the Part_usage object.

Associations

Part: Cavity_seal [1]
The part specifies the Cavity_seal that serves as a definition for this particular occurrence.

Class: Co_pack_occurrence

A Co_pack_occurrence is the occurrence of a Co_pack_part in a module part list.

Attributes

public Alias_id [0..*]
The alias_id specifies additional identifiers for the Co_pack _occurrence.

public Description [0..1]
The description specifies additional information about the ob ject.

public Id
The id specifies the identifier of the Co_pack_occurrence.

Associations

Part: Co_pack_part [1]
The part specifies the Co_pack_part that serves as a definition for this particular occurrence.

Installation_information: Installation_instruction [0..*]
The installation_information provides additional information for the installation of the Co_pack_occurrence.

Class: Component_occurrence

A Component_occurrence is the occurrence of a Component in a module part list.

Attributes

public Alias_id [0..*]
The alias_id specifies additional identifiers for the Component.

public Description [0..1]
The description specifies additional information about the object.

public Id
The id specifies the identifier of the Component.

Associations

Part: Component [1]
The part specifies the Component that serves as a definition for this particular occurrence.

Mounting: Mounting_element [1..*]
The mounting specifies the Cavity_occurrence, Slot_occurrence, or Connector_occurrence, which is associated with the Component_occurrence.

Class: Connector_occurrence

Steckverbinder

A Connector_occurrence is the occurrence of a Connector_housing in a module part list.

Attributes

public Alias_id [0..*]
The alias_id specifies additional identifiers for the Connector_occurrence.

public Description [0..1]
The description specifies additional information about the ob ject.

public Id
The id specifies the identifier of the Connector_occurrence.

public Placement [0..1]
The placement specifies the transformation information, which is used to locate and orient the occurrence in the car coordinate system. For further information see [CESAK].

public Usage [0..1]
The usage specifies the way how a Connector_occurrence is used in a connection.
Where applicable the following values shall be used:
- 'no end': end of wire without any connector ("blunt cut")
- 'ring terminal':
- 'splice':
- 'dangler': terminal without any connector housing.

Associations

Part: Connector_housing [1]
The part specifies the Connector_housing that serves as a definition for this particular occurrence.

Contact_points: Contact_point [0..*]
The contact_points specify the Contact_points, which belong to the connector.
Note: All Contact_points of a Connector_occurrence shall reference Cavity_occurrence objects of this Connector_occurrence.

Installation_information: Installation_instruction [0..*]
The installation_information provides additional information for the installation of the Connect-or_occurrence.

Slots: Slot_occurrence [0..*]
The slots specifies the Slot_occurrence objects which belong to the Connector_occurrence.

Class: Connector_usage

Attributes

public dangler

public no end

public ring terminal

public splice

Class: Core_occurrence

A Core_occurrence is the occurrence of a Core within a Special_wire_occurrence.

Attributes

public Wire_number
The wire_number specifies the identification of a wire. This number is unique within a vehicle and usually automatically generated (dependent on "from- to" information).
Note: Each interior conductor of a multi-core wire produces an own wire number.

Associations

Part: Core [1]
The part specifies the Core that serves as a definition for this particular occurrence.

Length_information: Wire_length [1..*]
The length_information specifies the length of a Core_occurrence. For a core, as many Wire_length objects as required may be defined. There are two pre-defined types:
- DMU: By CAD model calculated wire length. Usually it is the Sum of the lengths of neutral fibers of the bundle sections, normally too short.
- Manufacturing: determination by means of correction algorithms or measurement of the prototype.

Class: Fixing_occurrence

A Fixing_occurrence is the occurrence of a Fixing in a module part list.

Attributes

public Alias_id [0..*]
The alias_id specifies additional identifiers for the Fixing _occurrence.

public Description [0..1]
The description specifies additional information about the ob ject.

public Id
The id specifies the identifier of the Fixing_occurrence.

public Placement [0..1]
The placement specifies the transformation information, which is used to locate and orient the occurrence in the car coordinate system. For further information see [CESAK].

Associations

Part: Fixing [1]
The part specifies the Fixing that serves as a definition for this particular occurrence.

Installation_information: Installation_instruction [0..*]
The installation_information provides additional information for the installation of the Fixing_occurrence.

Class: General_wire_occurrence

A General_wire_occurrence is the occurrence of a Wire_occurrence or a Special_wire_occurrence in a module part list.

Associations

Part: General_wire [1]
The part specifies the General_wire that serves as a definition for this particular occurrence.

Installation_information: Installation_instruction [0..*]
The installation_information provides additional information for the installation of the Connection.

Length_information: Wire_length [1..*]
The length_information specifies the length of a General_wire_occurrence. For a wire, as many Wire_length objects as required may be defined. There are two pre-defined types:
- DMU: By CAD model calculated wire length. Usually it is the Sum of the lengths of neutral fibers of the bundle sections, normally too short.
- Manufacturing: determination by means of correction algorithms or measurement of the prototype.
There shall be at least one Wire_length of type DMU be defined for each wire.

Class: Mounting_element

A Mounting_element is an object which is associated with a Component_occurrence.

Class: Part_substitution

Austauschbauteil
A Part_substitution is a mechanism that describes the replacement of a sealing plug with a terminal. Typically usage: an optional module is added to a harness and a common connector will be used.

Associations

Replaced: Cavity_plug_occurrence [1]
The replaced specifies the Cavity_plug_occurrence, that is replaced.

Class: Part_usage_select

A Part_usage is an object which replaces a Cavity_occurrence.

Associations

Replacing: Part_substitution [0..*]

Class: Special_terminal_occurrence

A Special_terminal_occurrence is the occurrence of a General_terminal with an identifier in a module part list.

Attributes

public Alias_id [0..*]
The alias_id specifies additional identifiers for the Special_terminal _occurrence

public Description [0..1]
The description specifies additional information about the ob ject.

public Id
The id specifies the identifier of the Special_terminal_occurrence.

public Placement [0..1]
The placement specifies the transformation information, which is used to locate and orient the occurrence in the car coordinate system. For further information see [CESAK].

Associations

Part: General_terminal [1]
The part specifies the General_terminal that serves as a definition for this particular occurrence.

Installation_information: Installation_instruction [0..*]
The installation_information provides additional information for the installation of the Special_terminal_occurrence.

Class: Special_wire_occurrence

Mehrdrahtleitung
A Special_wire_occurrence is the occurrence of a multi-core wire in a module part list.

Attributes

public Special_wire_id
A special_wire_id specifies the identifier of the individual occurrence of multi-core wire. In contrast to the occurrence of a single wire which do not need to be identified in the module part list, each multi-core wire occurrence must be identified.

Associations

Core_occurrence: Core_occurrence [0..*]
The core_occurrence specifies the Core_occurrence cantained in the Special_wire_occurrence.

Class: Specified_accessory_occurrence

A Specified_ accessory _occurrence is the occurrence of a accessory_occurrence within an Assembly_part_occurrence.

Associations

Related_occurrence: Accessory_occurrence [1]
The related_occurrence specifies the Accessory_occurrence it is an occurrence of.

Related_assembly: Assembly_part_occurrence [1]
The related_ assembly specifies the Assembly_part_occurrence in which the occurrence is used.

Class: Specified_cavity_plug_occurrence

A Specified_ cavity _plug _occurrence is the occurrence of Cavity_plug_occurrence within an Assembly_part_occurrence.

Associations

Related_assembly: Assembly_part_occurrence [1]
The related_ assembly specifies the Assembly_part_occurrence in which the occurrence is used.

Related_occurrence: Cavity_plug_occurrence [1]
The related_occurrence specifies the Cavity_plug_occurrence it is an occurrence of.

Class: Specified_cavity_seal_occurrence

A Specified_ cavity_seal _occurrence is the occurrence of a Cavity_seal_occurrence within an Assembly_part_occurrence.

Associations

Related_assembly: Assembly_part_occurrence [1]
The related_ assembly specifies the Assembly_part_occurrence in which the occurrence is used.

Related_occurrence: Cavity_seal_occurrence [1]
The related_occurrence specifies the Cavity_seal_occurrence it is an occurrence of.

Class: Specified_co_pack_occurrence

A Specified_ co_pack _occurrence is the occurrence of a Co_pack_occurrence within an Assembly_part_occurrence.

Associations

Related_assembly: Assembly_part_occurrence [1]
The related_ assembly specifies the Assembly_part_occurrence in which the occurrence is used.

Related_occurrence: Co_pack_occurrence [1]
The related_occurrence specifies the Co_pack_occurrence it is an occurrence of.

Class: Specified_component_occurrence

A Specified_component_occurrence is the occurrence of a Component within an Assembly_part_occurrence.

Associations

Related_assembly: Assembly_part_occurrence [1]

Related_occurrence: Component_occurrence [1]

Class: Specified_connector_occurrence

A Specified_connector_occurrence is the occurrence of a Connector_occurrence within an Assembly_part_occurrence.

Associations

Related_assembly: Assembly_part_occurrence [1]
The related_ assembly specifies the Assembly_part_occurrence in which the occurrence is used.

Related_occurrence: Connector_occurrence [1]
The related_occurrence specifies the Connector_occurrence it is an occurrence of.

Class: Specified_fixing_occurrence

A Specified_ fixing_occurrence is the occurrence of a Fixing_occurrence within an Assembly_part_occurrence.

Associations

Related_assembly: Assembly_part_occurrence [1]
The related_ assembly specifies the Assembly_part_occurrence in which the occurrence is used.

Related_occurrence: Fixing_occurrence [1]
The related_occurrence specifies the Fixing_occurrence it is an occurrence of.

Class: Specified_special_terminal_occurrence

A Specified_ special_terminal _occurrence is the occurrence of a Special_terminal_occurrence within an Assembly_part_occurrence.

Associations

Related_assembly: Assembly_part_occurrence [1]
The related_ assembly specifies the Assembly_part_occurrence in which the occurrence is used.

Related_occurrence: Special_terminal_occurrence [1]
The related_occurrence specifies the Special_terminal_occurrence it is an occurrence of.

Class: Specified_special_wire_occurrence

A Specified_ special_wire _occurrence is the occurrence of a Special_wire_occurrence within an Assembly_part_occurrence.

Associations

Related_assembly: Assembly_part_occurrence [1]
The related_ assembly specifies the Assembly_part_occurrence in which the occurrence is used.

Related_occurrence: Special_wire_occurrence [1]
The related_occurrence specifies the Special_wire_occurrence it is an occurrence of.

Class: Specified_terminal_occurrence

A Specified_ terminal _occurrence is the occurrence of a Terminal_occurrence within an Assembly_part_occurrence.

Associations

Related_assembly: Assembly_part_occurrence [1]
The related_ assembly specifies the Assembly_part_occurrence in which the occurrence is used.

Related_occurrence: Terminal_occurrence [1]
The related_occurrence specifies the Terminal_occurrence it is an occurrence of.

Class: Specified_wire_occurrence

A Specified_wire_occurrence is the occurrence of a Wire_occurrence within an Assembly_part_occurrence.

Associations

Related_assembly: Assembly_part_occurrence [1]
The related_ assembly specifies the Assembly_part_occurrence in which the occurrence is used.

Related_occurrence: Wire_occurrence [1]
The related_occurrence specifies the Wire_occurrence it is an occurrence of.

Class: Specified_wire_protection_occurrence

A Specified_wire_protection_occurrence is the occurrence of a Wire_protection_occurrence within an Assembly_part_occurrence.

Associations

Related_assembly: Assembly_part_occurrence [1]
The related_ assembly specifies the Assembly_part_occurrence in which the occurrence is used.

Related_occurrence: Wire_protection_occurrence [1]
The related_occurrence specifies the Wire_protection_occurrence it is an occurrence of.

Class: Terminal_occurrence

A Terminal_occurrence is the occurrence of a General_terminal in a module part list.
Note: The number of occurrences used in a specific module or harness can be calculated by the individual occurrences.
Note: The usage of a Terminal_occurrence for a particular Cavity is specified by the Part_usage object.

Associations

Part: General_terminal [1]
The part specifies the General_terminal that serves as a definition for this particular occurrence.

Class: Wire_occurrence

A Wire_occurrence is the occurrence of a single wire in a module part list.

Attributes

public Wire_number
The wire_number specifies the identification of a wire. This number is unique within a vehicle and usually automatically generated (dependent on "from-to" information).

Class: Wire_protection_occurrence

A Wire_protection_occurrence is the occurrence of a Wire_protection in a module part list.

Attributes

public Alias_id [0..*]
The alias_id specifies additional identifiers for the Wire_protection _occurrence.

public Description [0..1]
The description specifies additional information about the ob ject.

public Id
The id specifies the identifier of the Wire_protection.

public Protection_length [0..1]
The protection_length specifies the length of the Wire_protection_occurrence with respect to the wrapping.
Note: The protection_length may differ from the real length of the material. E.g. for a tube, the protection_length and the real length are same, whereas for a tape they are different.

Associations

Installation_information: Installation_instruction [0..*]
The installation_information provides additional information for the installation of the Wire_protection_occurrence.

Part: Wire_protection [1]
The part specifies the Wire_protection that serves as a definition for this particular occurrence.

Package: ::KBL::4_Connectivity

This concept specifies the concepts required to specify the connectivity among the connectors within a harness. The specification of contact points and wiring groups are included.

Classes

Class: Cavity_occurrence

A Cavity_occurrence is the occurrence of a Cavity in the context of a connector_occurrence.

Associations

Part: Cavity [1]
The part specifies the Cavity that serves as a definition for this particular occurrence.

Associated_plug : Cavity_plug_occurrence [0..1]
The associated_plug specifies the plug the cavity is equipped with.

Class: Connection

Verbindung
A Connection is a mechanism to specify the electrical connectivity between two or more contact points.

Attributes

public Description [0..1]
The description specifies additional information about the object.

public Id [0..1]
The id specifies the identifier of the Connectivity. The values are company specific. They can depend also on wire parameters.
Example: electrical potential, start->destination

public Signal_name [0..1]
The signal_name specifies logical information on a Connection.
Example: packet on a bus, analogue voltage(high/low, waved) on a wire.

Associations

External_references: External_reference [0..*]
The external_references specifies references to documents.
Example: Schematic diagram for the connection.

Extremities: Extremity [2..*]
The extremities specify Extremity objects, which reference the Contact_points connected by this Connection. There shall be at least two objects specified.

Installation_information: Installation_instruction [0..*]
The installation_information provides additional information for the installation of the Connection.

Processing_information: Processing_instruction [0..*]

The processing_information provides additional information for the processing of the Connection.

Wire: Wire_or_core_occurrence [1]
The wire specifies the Wire_occurrence or Core_occurrence which realizes the Connection.

Class: Contact_point

A Contact_point defines the positions where electrical connectivity takes place.

Attributes

public Id
The id specifies the identifier of the Contact_point.

Associations

Contacted_cavity: Cavity_occurrence [1..*]
The contacted_cavity specifies the Cavity_objects related with the Contact_point.

Associated_parts: Part_usage_select [0..*]
The associated_parts specifies the Cavity_seal_occurrences, Special_terminal_occurrences, or Terminal_occurrences associated with the Contact_point.

Class: Extremity

The Extremity specifies the Contact_point which is connected by a spcific Connection.

Attributes

public Position_on_wire
The Position_on_wire describes the position where the contacting takes place. That is important for contacts like IDC, where the contacting is not at the beginning or end of the wire.
Note: A value of 0.0 designates the beginning of the wire, a value of 1.0 the end. Intermediate contact_points are defined relative to the first one by a a value between 0.0 and 1.0.
The position_on_wire defines the ordering of the Extremities within a Connection.

Associations

Contact_point: Contact_point [1]
The contact_point specifies the Contact_point, which is associated with the Extremity, and by this with the connection.

Class: Slot_occurrence

A Slot_occurrence is the occurrence of a Slot in a module part list.

Associations

Cavities: Cavity_occurrence [1..*]
The cavities specifies the Cavity_occurrence objects belonging to the Slot_occurrence.

Part: Slot [1]
The part specifies the Slot that serves as a definition for this particular occurrence.

Class: Wire_or_core_occurrence

A Wire_or_core_occurrence is an occurrence of a single wire, a multi-core wire, or a individual core of a multi-core wire.

Class: Wiring_group

A Wiring_goup is a mechanism to group wire or core occurrences together.
Example: Twisted pairs.

Attributes

public Id
The id specifies the identifier of the Wiring_group.

public Type [0..1]
The type specifies the type of the grouping.
Example: twisted-pair.

Associations

Processing_information: Processing_instruction [0..*]
The processing_information provides additional information for the processing of the Wiring_group.

Assigned_wire: Wire_or_core_occurrence [2..*]
The assigned_wire specifies the Wire_occurrence and Core_occurrence objects which are grouped together.

Package: ::KBL::5_Topology

This package specifies the information contained in harness layouts and drawings. This includes the geometric description of bundles and the information how a specific wire is routed.

Classes

Class: Dimension_specification

A Dimension_specification is the definition of the distance between two elements.

Associations

target: Placed_element [1]
The target specifies the Node or the located component that is the target for the Dimension_specification.

origin: Placed_element [1]
The origin specifies the Node or the located component that is the origin for the Dimension_specification.

Tolerance_indication: Tolerance [0..1]
The Tolerance_indication specifies the value of the distance between the two elements.

Class: Fixed_component

A Fixed_component is an object that can be assigned by a fixing.

Class: Fixing_assignment

A Fixing_assignment assigns a Fixing_occurrence or an Accessory_occurrence to a Segment.

Attributes

public Location
The location specifies the position of the fixing on the Segment. The value is given in curve parameters running from 0.0 to 1.0.

public Orientation [2..3]
The orientation specifies the direction of the z-axis and is given by 2 or 3 values. The y-axis is defined by the tangent of the center curve at the location.

Associations

Fixing: Fixed_component [1]

The fixing specifies the Fixing_occurrence or the Accessory_occurrence, which is assigned.

Class: Located_component

Positionierbare Komponenten der KBL: z.B. Connector_occurrence, Accessory_occurrence, Fixing_occurrence, ...
A Located_component is a component which can get a position in the topology by a node.

Class: Node

A node forms begin and end of a Segment.

Attributes

public Alias_id [0..*]
The alias_id specifies additional identifiers for the Node.
Example: Node Ids may vary from one CAD system export to another.
Therefore the CAD system Id is improper for Id attribute. Id shall have a value which is unique within the harness. Alias_id may be used for the CAD system Id.

public Bend_radius [0..1]
The bend_radius specifies the maximum bend radius at the Node.
Example: May be used for the routing of fibre-optics.

public Id
The id specifies the identifier of the Node.

Associations

Cartesian_point: Cartesian_point [1]
The cartesian_point specifies the position of the Node with respect to the absolute car coordinate system.

referenced_components: Located_component [0..*]
The referenced_components specifies the Accessory_occurrences, Assembly_part_occurrences, Connector_occurrences, Fixing_occurrences, Special_terminal_occurrences, and Wire_protection_occurrences located at the node.
Note: Different elements can lead to a node, e.g. plug, Splice, perhaps clip.

Processing_information: Processing_instruction [0..*]
The processing_information provides additional information for the processing of the Node.

Class: Placed_element

A Placed_element is an element where a Dimension_specification can apply.

Class: Protection_area

Schutzzone
A Protection_area is a a mechanism to describe the area on a Segment covered by a Wire_protection_occurrence. Some segments of a harness are crossing areas with critical conditions like higher temperatures, higher humidity, etc. To prevent the damage of the cable, protect shields are mounted for that segment.
Note: For each Segment covered by a particular Wire_protection_occurrence a separate Protection_area has to be instantiated.

Attributes

public End_location
The end_location specifies the end position of the Wire_protection_occurrence on the Segment. The value is given in curve parameters running from 0.0 to 1.0.

public Gradient [0..1]
The gradient specifies the gradient of the protection.

Note: The gradient is defined in mm. Gradient 15 mms means that the winding screws within a turn 15 mms along the bundle.

public Start_location
The start_location specifies the start position of the Wire_protection_occurrence on the Segment. The value is given in curve parameters running from 0.0 to 1.0.

public Taping_direction [0..1]
The taping_direction specifies the direction of the taping (left/right) regarding to the Segment direction.

Associations

Processing_information: Processing_instruction [0..*]
The processing_information provides additional information for the processing of the Protection_area.

Associated_protection: Wire_protection_occurrence [1]
The associated_protection specifies the Wire_protection_occurrence associated Segment objects.

Class: Routing

A Routing is a course taken to get from a starting point to a destination.

Associations

Routed_wire: Connection [1]
The routed_wire specifies the Connection for which the Routing is defined.

Segments: Segment [0..*]
The segments specifies the Segment objects the course follows.

Class: Segment

Verbindungsabschnitt
A Segment is a section of a Connection where no intermediate electrical contacts appear. At the beginning and at the end the same wires go in and out. Cables and wires are divided in segments. Every segment has its conditions like length, temperature range, etc.

Attributes

public Alias_id [0..*]
The alias_id specifies additional identifiers for the Segment.
Example: Segement Ids may vary from one CAD system export to another.
Therefore the CAD system Id is improper for Id attribute. Id shall have a value which is unique within the harness. Alias_id may be used for the CAD system Id.

public End_vector [0..3]
The end_vector specifies the tangent of the center curve at the end position.
Note: The value can be derived from the center_curve. To avoid that the receiving system has to calculate the value, it can be explicitly specified.

public Form [0..1]
The form specifies information on the geometric shape of the Segment.
The following values shall be used:
- 'circular'
- 'noncircular'

public Id
The id specifies the identifier of the Segment.

public Physical_length [0..1]
The physical_length specifies the arc length of the neutral phase of the segment in 3d.

public Start_vector [0..3]
The start_vector specifies the tangent of the center curve at the start position.
Note: The value can be derived from the center_curve. To avoid that the receiving system has to calculate the value, it can be explicitly specified.

public Virtual_length [0..1]
The virtual_length specifies the represented length of the neutral phase of the segment in 3d.

Associations

Center_curve: B_spline_curve [0..*]
The center_curve specifies the mathematical definition of the set of center curves of the Segment.

Cross_section_area_information: Cross_section_area [0..*]
The cross_section_area_information specifies the cross-sectional area of the route.
Note: This information is needed for noncircular Segments.

Fixing_assignment : Fixing_assignment [0..*]
The fixing_assignment specifies the Fixing_assignment objects associated with the Segement.

End_node: Node [1]
The end_node specifies the Node defining the end of the Segment.

Start_node: Node [1]
The start_node specifies the Node defining the start of the Segment.

Protection_area: Protection_area [0..*]
The Protection_area specifies the Protection_area objects associated with the Segment.

Class: Segment_form

Attributes

public circular

public noncircular

Class: Tolerance

A Tolerance is the specification of the allowable range for a dimension.

Attributes

public Lower_limit [0..1]
The lower_limit specifies the minimum allowed value.

public Upper_limit [0..1]
The upper_limit specifies the maximum allowed value.

Package: ::KBL::6_Foundation

This packages contains the basic classes for the KBL like numerical values, units, materials and also more complex descriptions like change information.

Classes

Class: Alias_identification

Alternative Identifikation
An Alias_identification is a mechanism to associate an object with an additional identifier that is used to identify the object of interest in a different context, either in another Organization, or in some other context. The scope of the Alias_identification shall be specified by the attributes 'Scope' and/or by the attribute 'Description'.

Attributes

public Alias_id
The alias_id specifies the identifier used in the context specified by scope and description.

public Description [0..1]
The description specifies the type of the Alias_identification.
Example: The description may be, e.g., 'inventory number'.

public Scope [0..1]
The scope specifies the organization in which the Alias_identification is valid.

Class: Approval

An Approval is a judgment concerning the quality of those modules or harnesses that are subject of the Approval. An Approval represents a statement made by technical personnel or management personnel whether certain requirements are met. The absence of approval information does not imply any approval status by default.

Attributes

public Date
The date specifies the date when the Approval actually became valid.

public Department [0..1]
The department specifies the department name of the personnel responsible for the Approval.

public Name [0..1]
The name specifies the name of the personnel responsible for the Approval.

public Type_of_approval
The type_of_approval specifies the terms characterizing the Approval.
Note: There are no values pre-defined. Special values have to be negotiated between exchange partners.

Associations

Is_applied_to: Part_with_title_block [1..*]
The is_applied_to specifies the hobjects to which the Approval is assigned.

Class: B_spline_curve

A B-spline curve is a piecewise parametric polynomial or rational curve described in terms of control points and basis functions. The B-spline curve has been selected as the most stable format to represent all types of polynomial or rational parametric curves. With appropriate attribute values it is capable of representing single span or spline curves of explicit polynomial, rational, Bezier or B-spline type.
Within the Harness Engineering Information Model the definition has been restricted to a uniform B_spline_curve, where the knots are evenly spaced. Suitable default values for the knots and knot multiplicities are derived in this case. A B-spline is uniform if and only if all knots are of multiplicity 1 and they differ by a positive constant from the preceding knot. In this case the knot spacing is 1.0, starting at -d , where d is the degree.
Note: If the B-spline curve is uniform and degree=1, the B-spline is equivalent to a polyline.

Attributes

public Degree
The algebraic degree of the basis functions.

Associations

Control_points: Cartesian_point [2..*]
The list of control points for the curve.
The list shall contain at least 2 elements.

Class: Cartesian_point

A Cartesian_point is a point that is defined by its coordinates in a rectangular Cartesian coordinate sys-

tem.

Attributes

public Coordinates [2..3]
The coordinates specify the coordinates of the Cartesian_point. The third coordinate will not exist in the case of a two-dimensional point.

Class: Change

A Change is a mechanism to keep track of the change history. The set of changes assigned to a Part covers all modification numbers up to the last modification level of the version of the Part.

Attributes

public Approver_department [0..1]
The approver_department specifies the department of the personnel who approved the Part.

public Approver_name [0..1]
The approver_name specifies the name of the personnel who approved the Part.

public Change_date [0..1]
The change_date specifies the date the change was performed.

public Change_request [0..1]
The change_request specifies the activity which triggers one ore more changes.
Example: change request, AEKO, VV; "Modellpflegepunkt"

public Description [0..1]
The description specifies additional information about the change.

public Designer_department
The designer_department specifies the department of the responsible design engineer.

public Id [0..1]
The id specifies the identifier by which a certain change can be referenced.
Example: reference of a fax, note, etc.

public Responsible_designer
The responsible_designer specifies the responsible design engineer.

Class: Creation

A Creation assigns creation information to a Module or Harness.

Attributes

public Date
The date specifies the creation date.

public Department
The department specifies the department the creator belongs to.

public Name
The name specifies the name of the creator.

Associations

Is_applied_to: Part_with_title_block [1..*]
The is_applied_to specifies the harness or module objects to which the Creation is assigned.

Class: Cross_section_area

A Cross_section_area

Attributes

public Area

The area specifies the value of the Cross_section_area.

public Value_determination
The value_determinates specifies whether the value is calculated, reserved, or measured.

Class: External_reference

An External_reference is a mechanism to provide information of the documents associated with the Part. Example: 3D model, form board drawings, etc.

Attributes

public Change_level
The change_level specifies the version of the document.

public Creating_system [0..1]
The creating_system specifies the computer application or the machine which is used to create the document.

public Data_format
The data_format specifies the convention that was used to structure the information in the document.

public Document_number
The document_number specifies the identifier of the document.

public Document_type
The document_type specifies the kind of the document.
Where applicable the following values shall be used:
- 3D-Data set (wiring, construction unit)
- 2D-Data set (Ltgs design, symbol, plug face)
- cable connection diagram
- set of cables database (VW: LCS-Container)
- specification of set of cables
- standards, technical guidelines

public File_name [0..1]
The file_name specifies the name which is used to locate the file either on a computer system or in a repository of paper documents.

public Location [0..1]
The location specifies where a document can be found in a digital or physical data storage system.

Class: Installation_instruction

An Installation_instruction is the description of the methods that can be used to install a Part.

Attributes

public Instruction_type

public Instruction_value
The instruction_value specifies the value for the kind of the Installation_instruction defined by instruction_type.

Class: Material

A Material is the substance out of which a Part is or can be made.

Attributes

public Material_key
The material_key specifies a code by which the material can be identified.

public Material_reference_system [0..1]
The material_reference_system specifies the system in which the material_key and its meaning is defined.

Example: 'IMDS' for the International Material Database System.

Class: Numerical_value

A Numerical_value is a quantity expressed with a numerical value and a unit.

Attributes

public Value_component
The value_component specifies the quantity of the Numerical_value.

Class: Processing_instruction

A Processing_instruction is the description of the methods that can be used to process a Part.

Attributes

public Instruction_type
The instruction_type specifies the kind of the Processing_instruction.
Note: There are no values pre-defined. Special values have to be negotiated between exchange partners.

public Instruction_value
The instruction_value specifies the value for the kind of the Processing_instruction defined by instruction_type.

Class: SI_prefix

Attributes

public centi

public kilo

public micro

public milli

Associations

: Unit [1]

Class: SI_unit_name

Attributes

public gram

public metre

Associations

: Unit [1]

Class: Transformation

A Transformation is a geometric transformation composed of translation and rotation. Scaling is not included.

Attributes

public U [2..3]
The u specifies the rotation by means of the y-axis.

public V [2..3]
The v specifies the rotation by means of the z-axis.

Associations

Cartesian_point: Cartesian_point [1]
The cartesian_point specifies the coordinates of the translation.

Class: Unit

A Unit is a quantity chosen as a standard in terms of which other quantities may be expressed.
Example: A unit of square milli metre is defined as a SI_unit with Si_unit_name "metre", Si_prefix "milli" and Si_dimension "square".

Attributes

public Si_dimension [0..1]
The Dimension specifies the dimension of a SI_unit.
Example: "square", "cubic".

public Si_prefix [0..1]
The si_prefix specifies the prefix for a SI unit.
Example: 'milli', 'kilo'

public Si_unit_name [0..1]
The si_ unit_name specifies the name for a SI unit.
Example: 'gram', 'metre'

public Unit_name [0..1]
The unit_name specifies the name for a non SI unit.
Note: This attribute shall be used to specify units like 'kg/100 pieces'

Class: Unit_dimension

Attributes

public cubic

public square

Class: Value_determination

Attributes

public calculated

public measured

public reserved

Class: Value_range

A Value_range is a pair of numerical values representing the range in which the value shall lie.

Attributes

public Maximum
The maximum specifies the maximum acceptable value that is constrained by the Value_range.

public Minimum
The minimum specifies the minimum acceptable value that is constrained by the Value_range.

Class: Value_with_unit

A Value_with_unit is either a single numerical measure, or a range of numerical measures with upper, lower, or upper and lower bounds.

Associations

Unit_component: Unit [1]
The unit_component specifies the unit in which the Value_with_unit is expressed.

Class: Wire_colour

A Wire_colour is a mechanism to define a colour for a wire together with a description of the kind of the colour.

Attributes

public Colour_type
The colour_type specifies the type of the colour.
Note: There are no values pre-defined. Special values have to be negotiated between exchange partners.
Example: 'base colour', 'second', 'third'

public Colour_value
The colour_value specifies the value of the colour.
Note: There are no values pre-defined. Special values have to be negotiated between exchange partners.
Example: 'red'

Class: Wire_length

Kabellänge
A Wire_length is a mechanism to define a length for a wire together with a description of the kind of the length (e.g. for DMU, for manufacturing).

Attributes

public Length_type
The length_type specifies the type of the length.
Note: There are no values pre-defined. Special values have to be negotiated between exchange partners.

public Length_value
The length_value specifies the length of the wire.

Package: ::KBL::A_Diagrams

Diagrams

Diagram: (00) Connection Overview

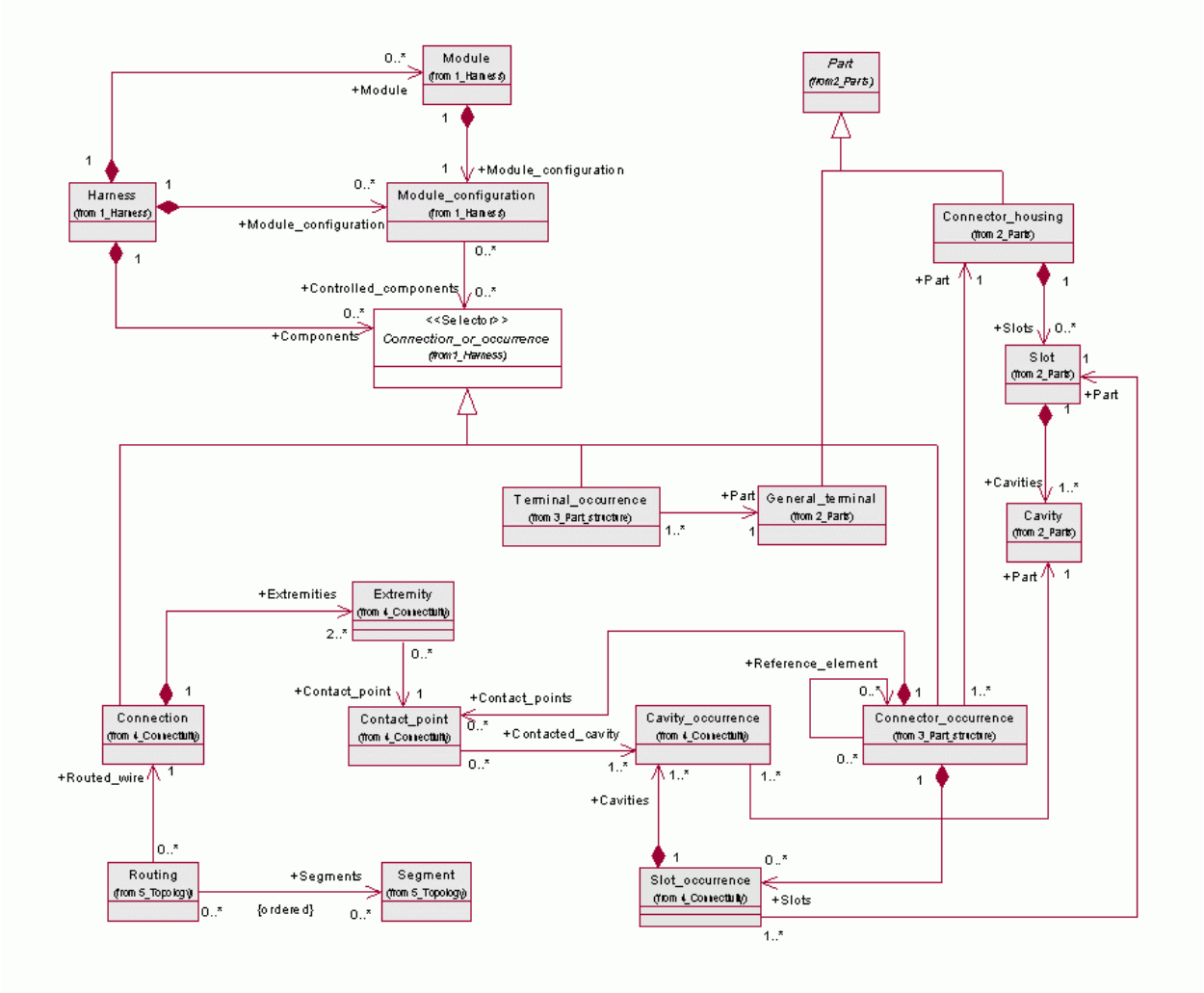

Diagram: (00) Connection Overview

Diagram: (01) Common part attributes and properties

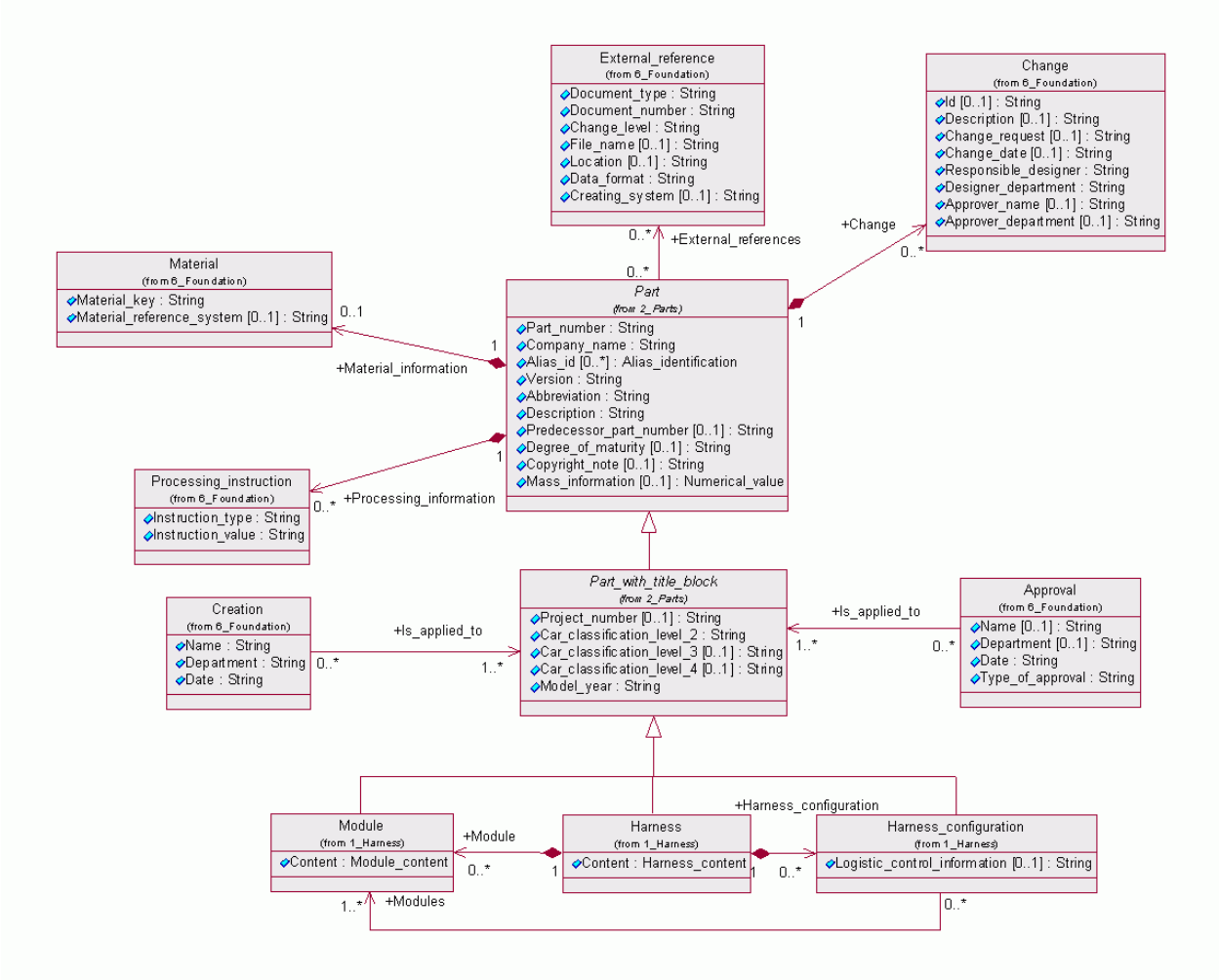

Diagram: (01) Common part attributes and properties

Diagram: (02) Part specializations

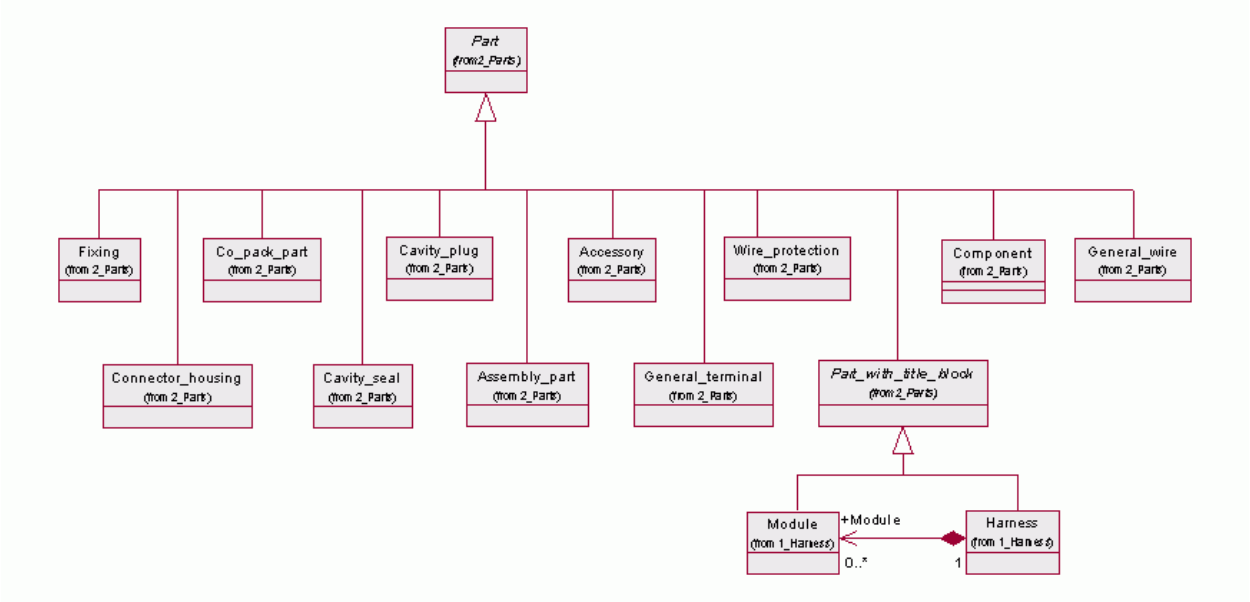

Diagram: (02) Part specializations

Diagram: (03) Harness and Modules

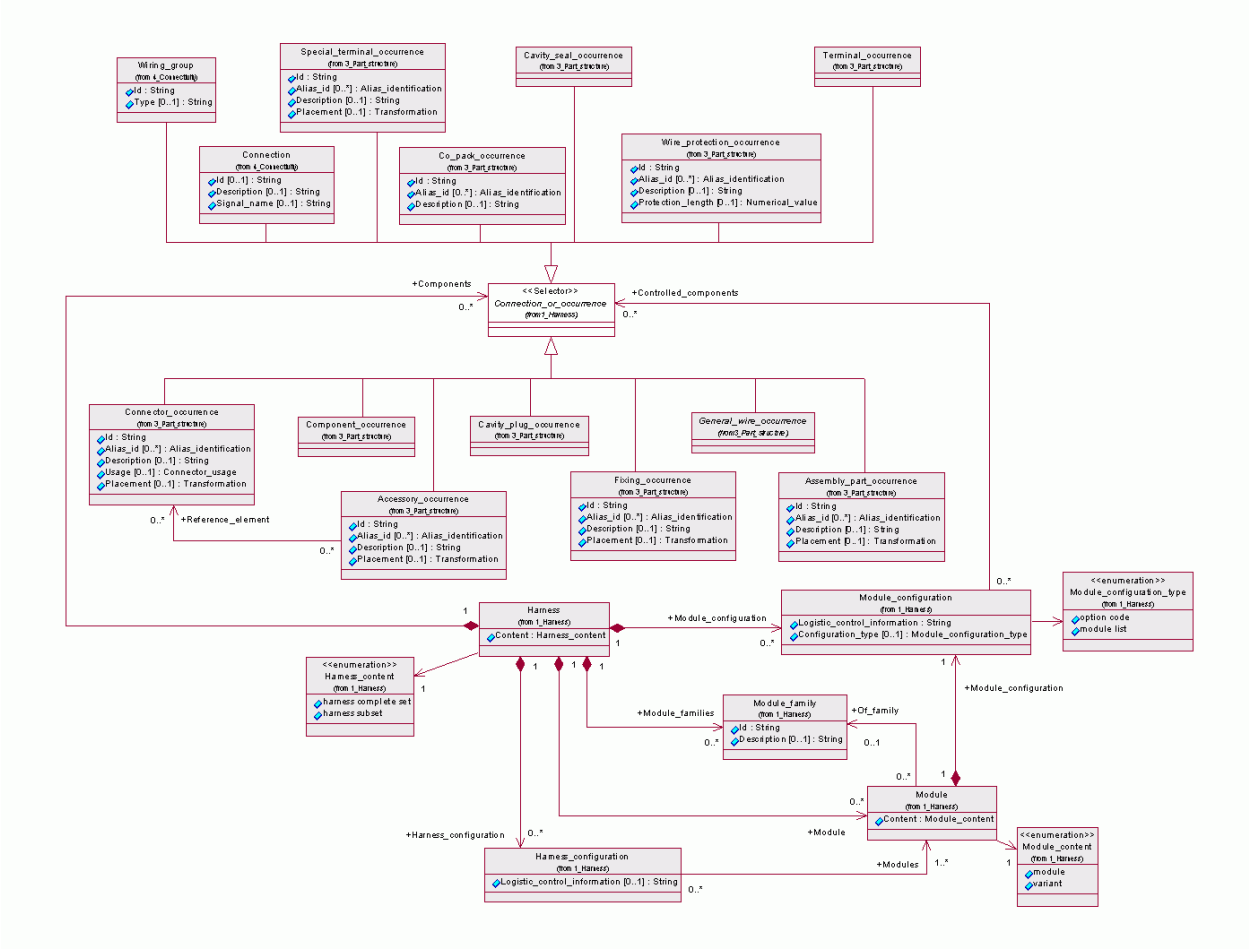

Diagram: (03) Harness and Modules

Diagram: (04) Part master data

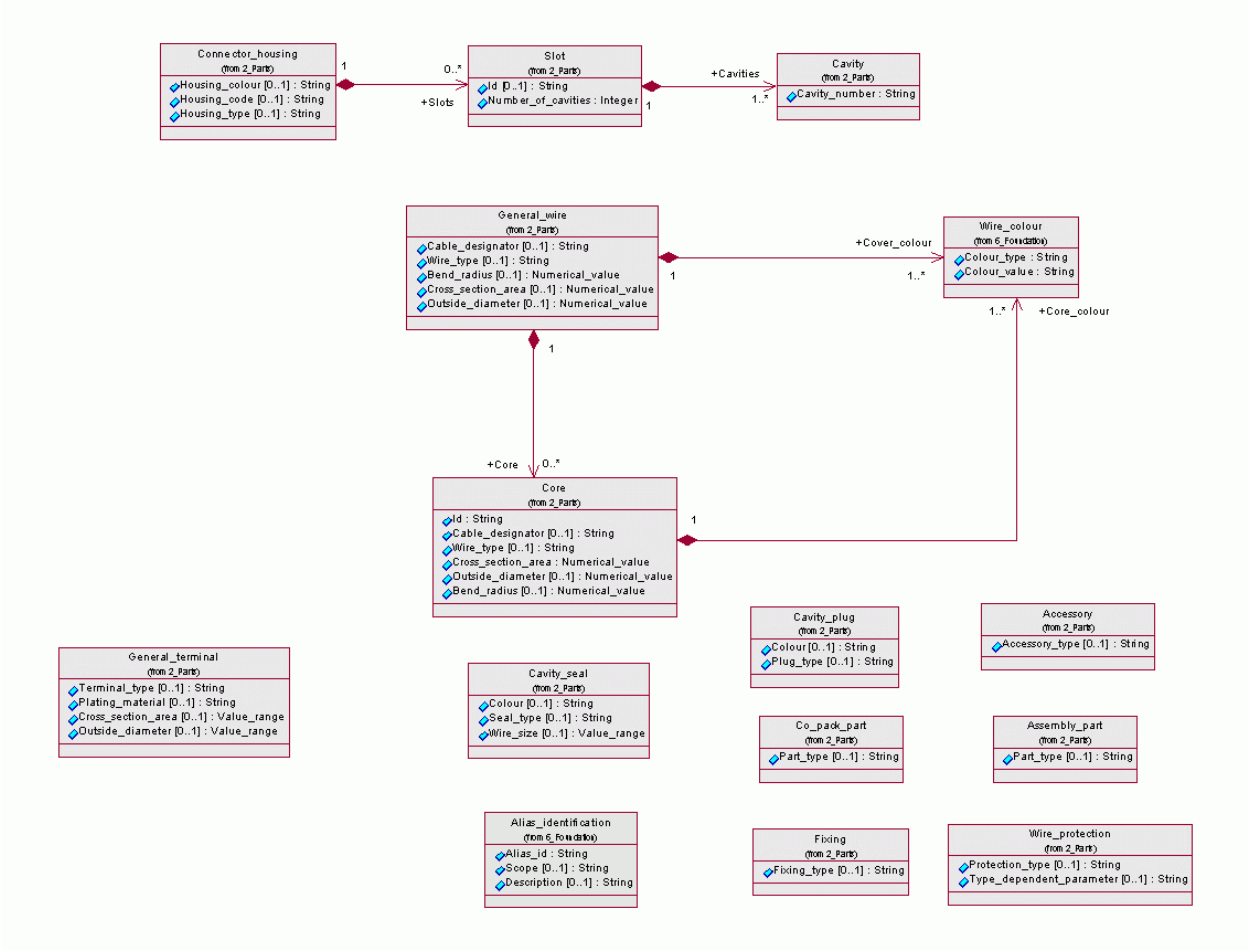

Diagram: (04) Part master data

Diagram: (05) Connectivity

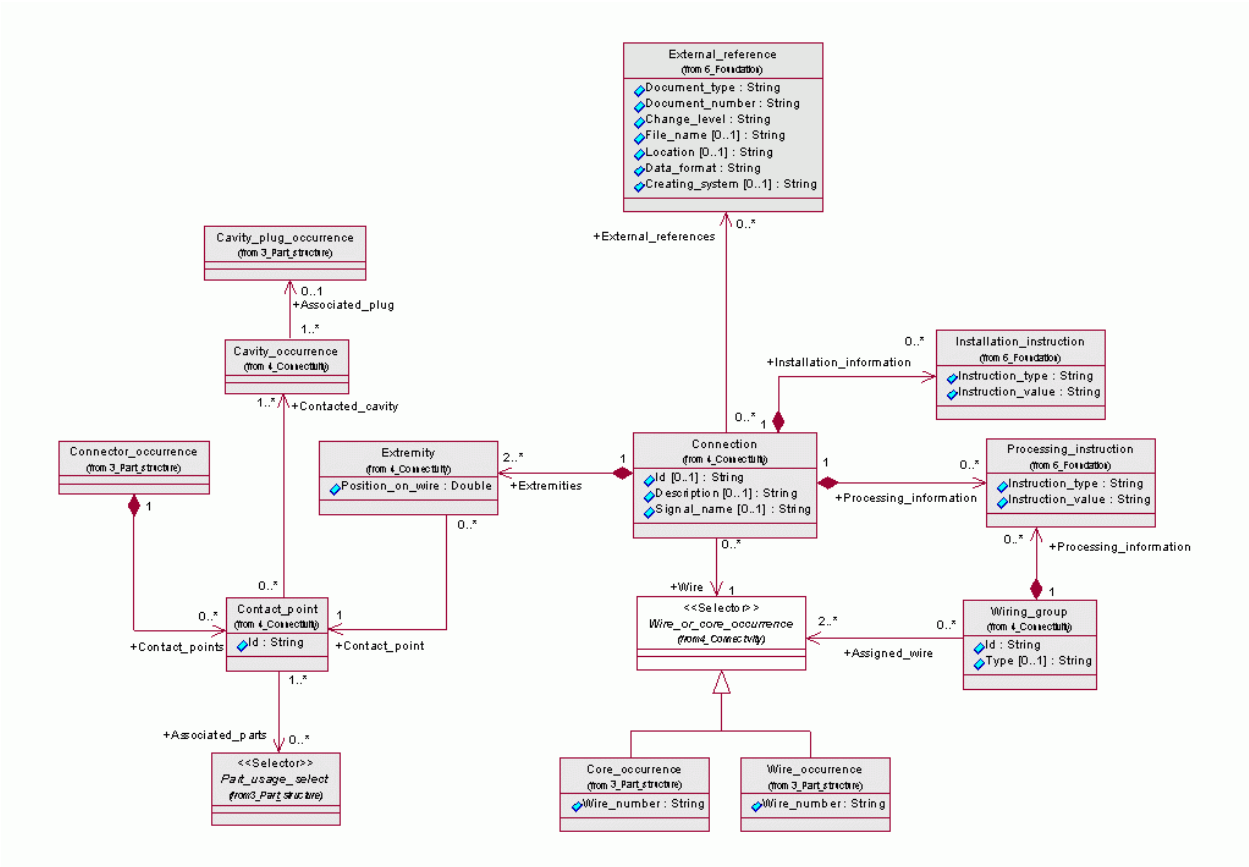

Diagram: (05) Connectivity

Diagram: (06) Part usage list (1)

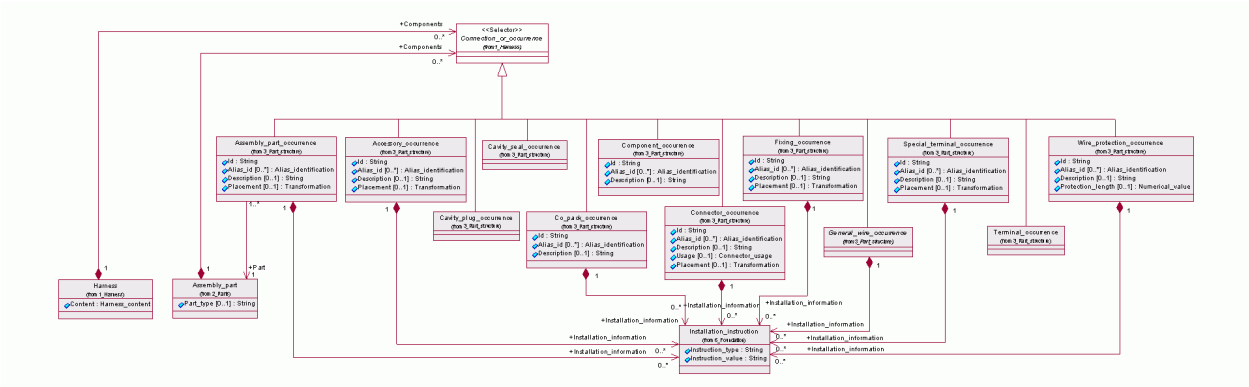

Diagram: (06) Part usage list (1)

Diagram: (07) Part usage list (2)

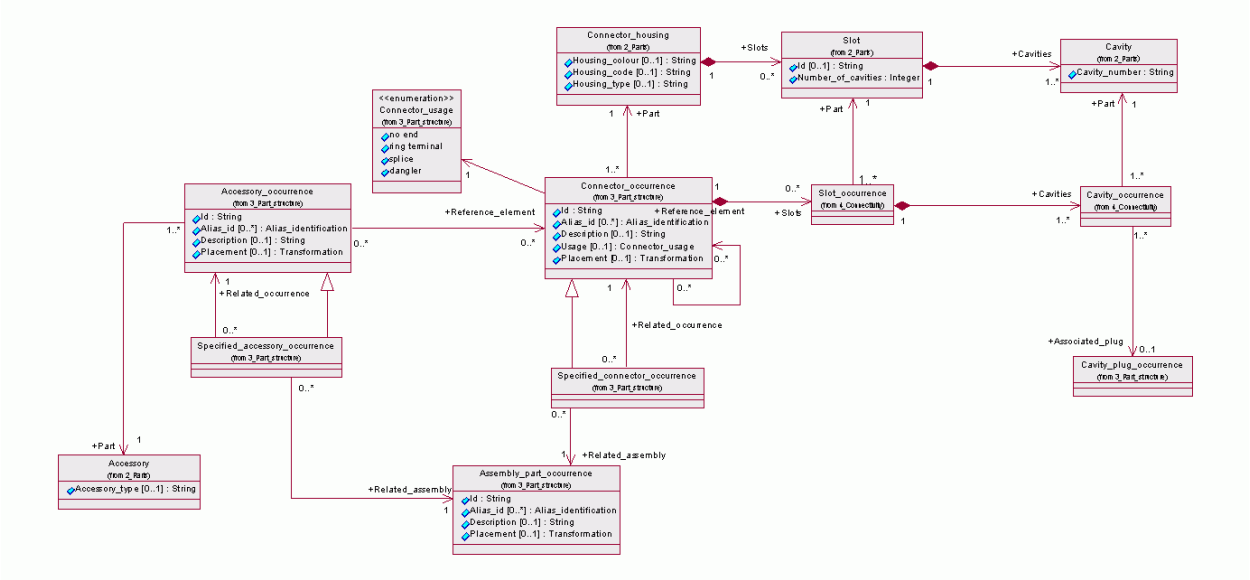

Diagram: (07) Part usage list (2)

Diagram: (08) Part usage list (3)

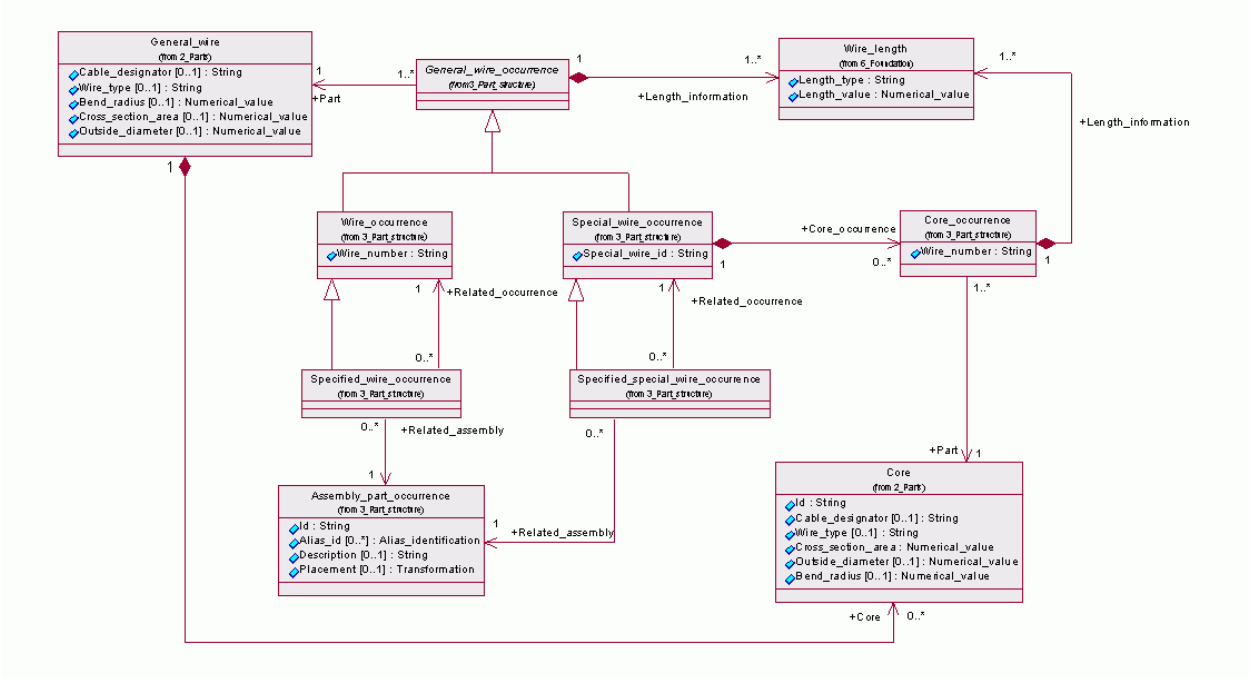

Diagram: (08) Part usage list (3)

Diagram: (09) Part usage list (4)

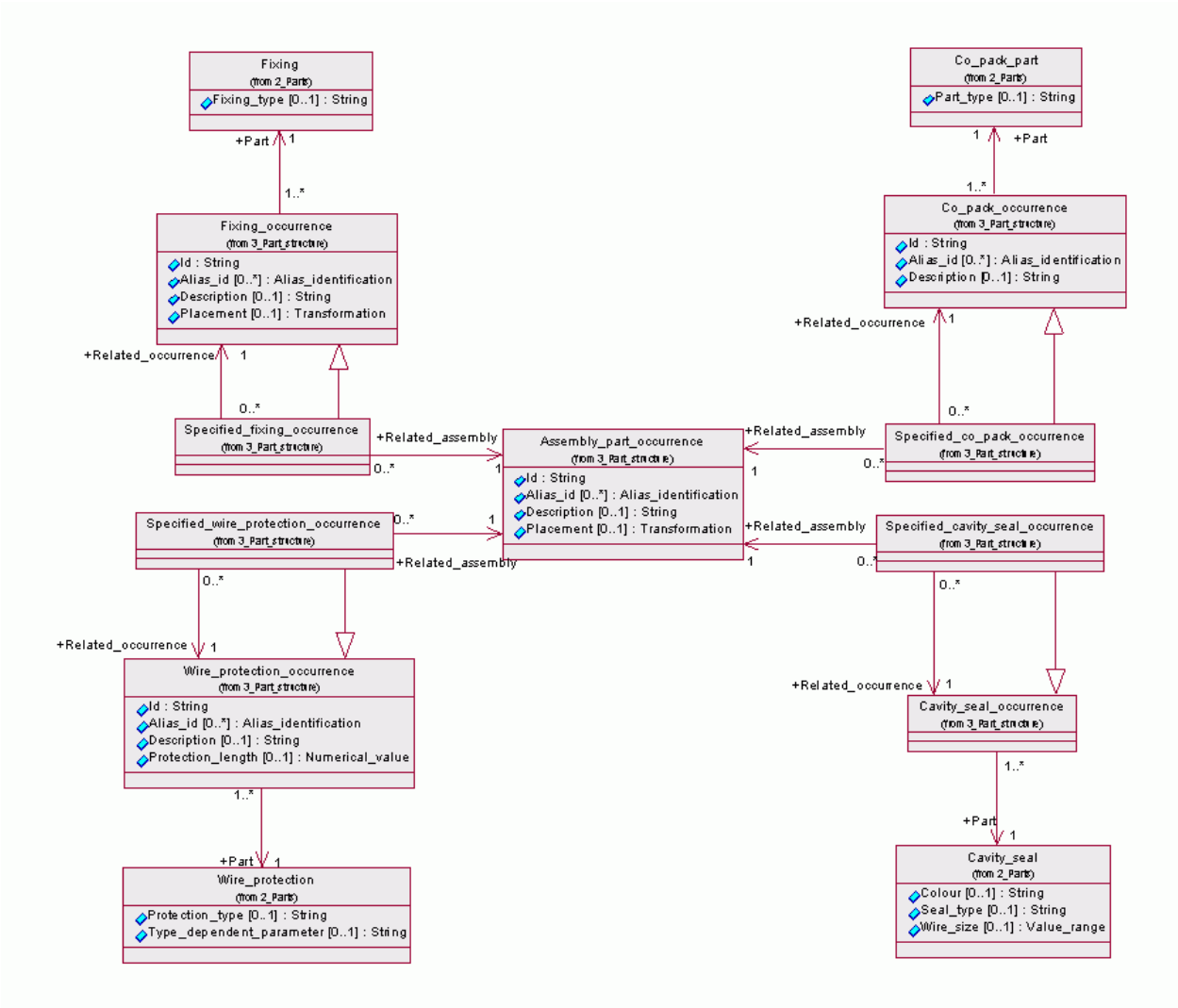

Diagram: (09) Part usage list (4)

Diagram: (10) Part usage list (5)

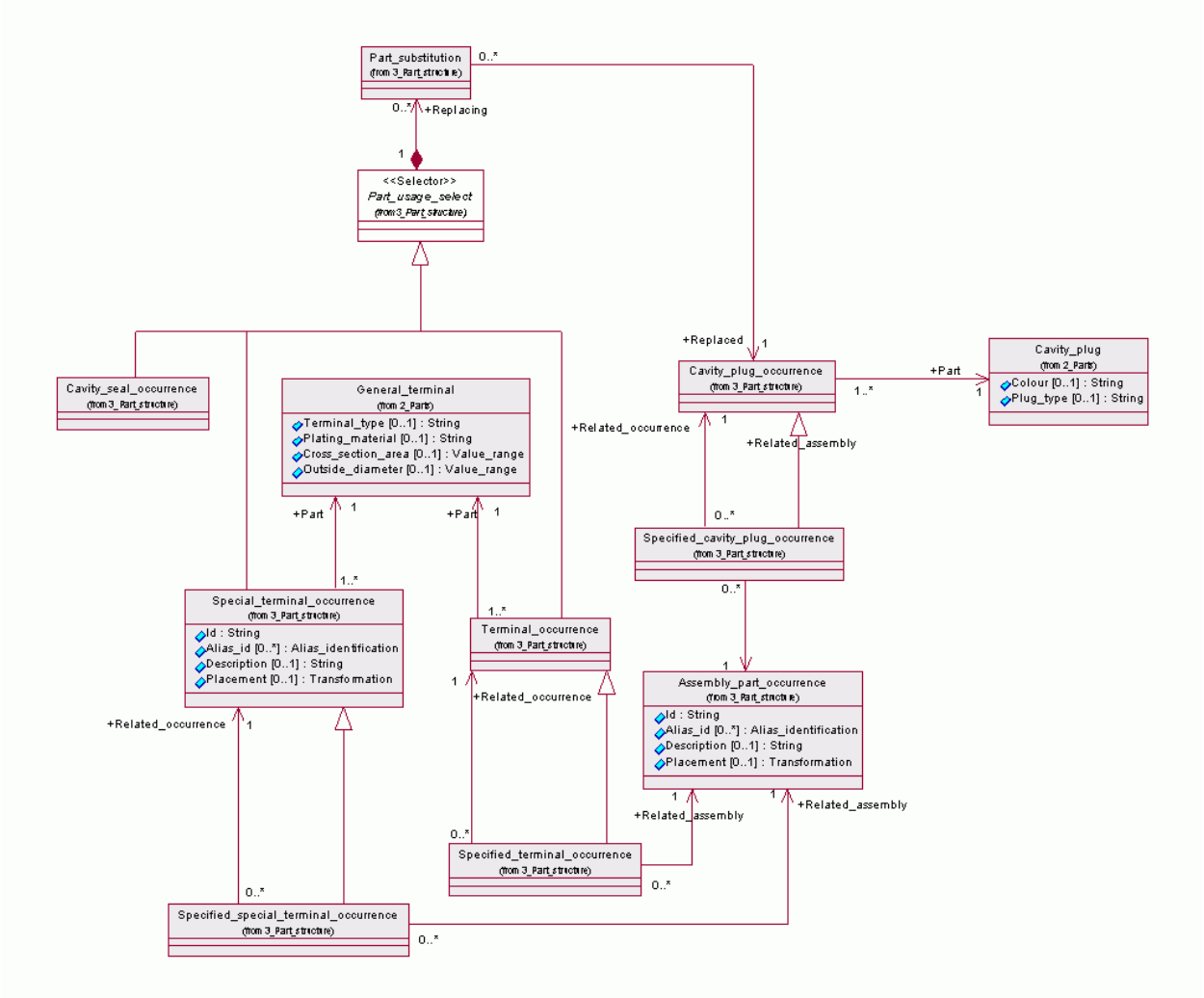

Diagram: (10) Part usage list (5)

Diagram: (11) Part usage list (6)

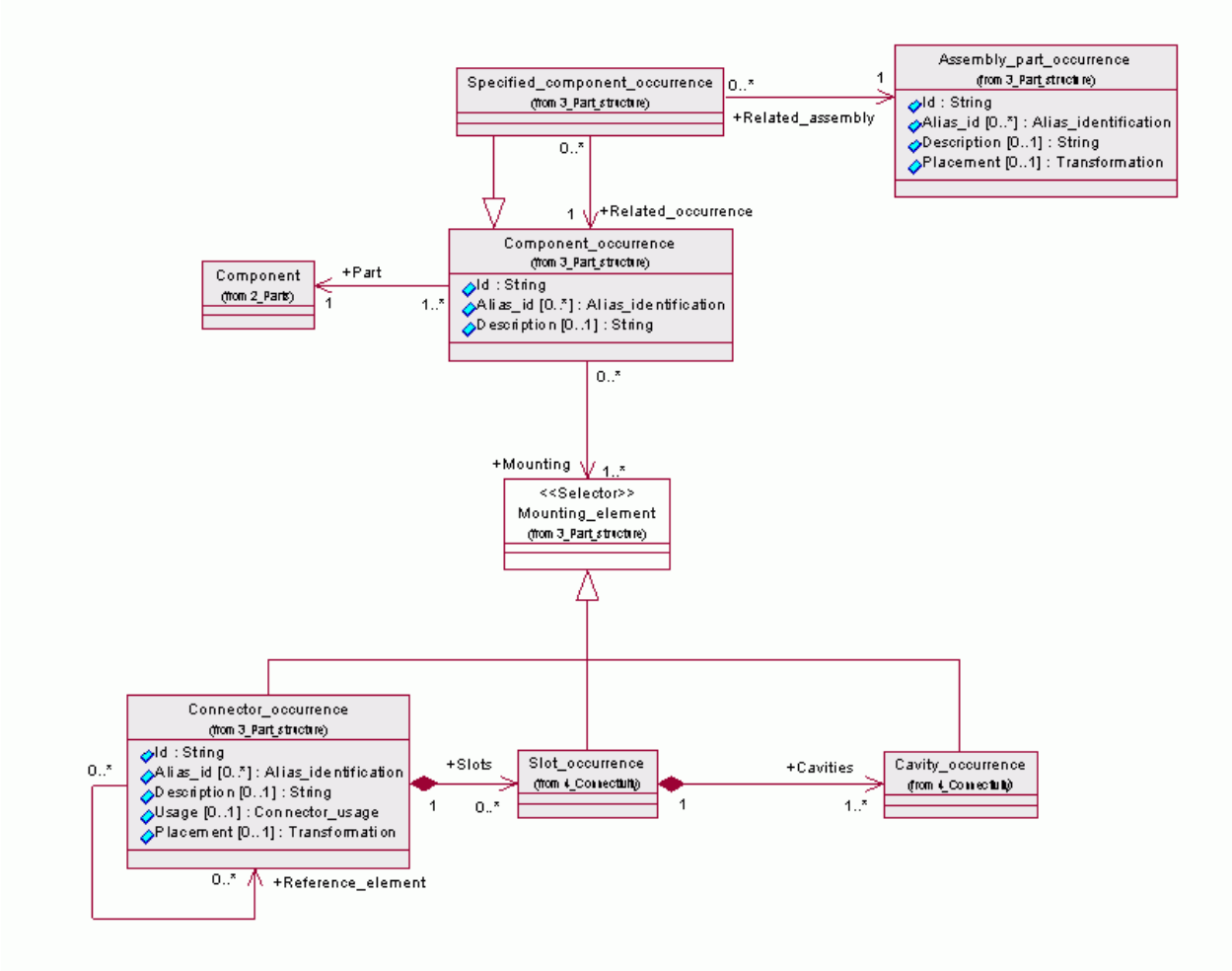

Diagram: (11) Part usage list (6)

Diagram: (12) Topology and routing (1)

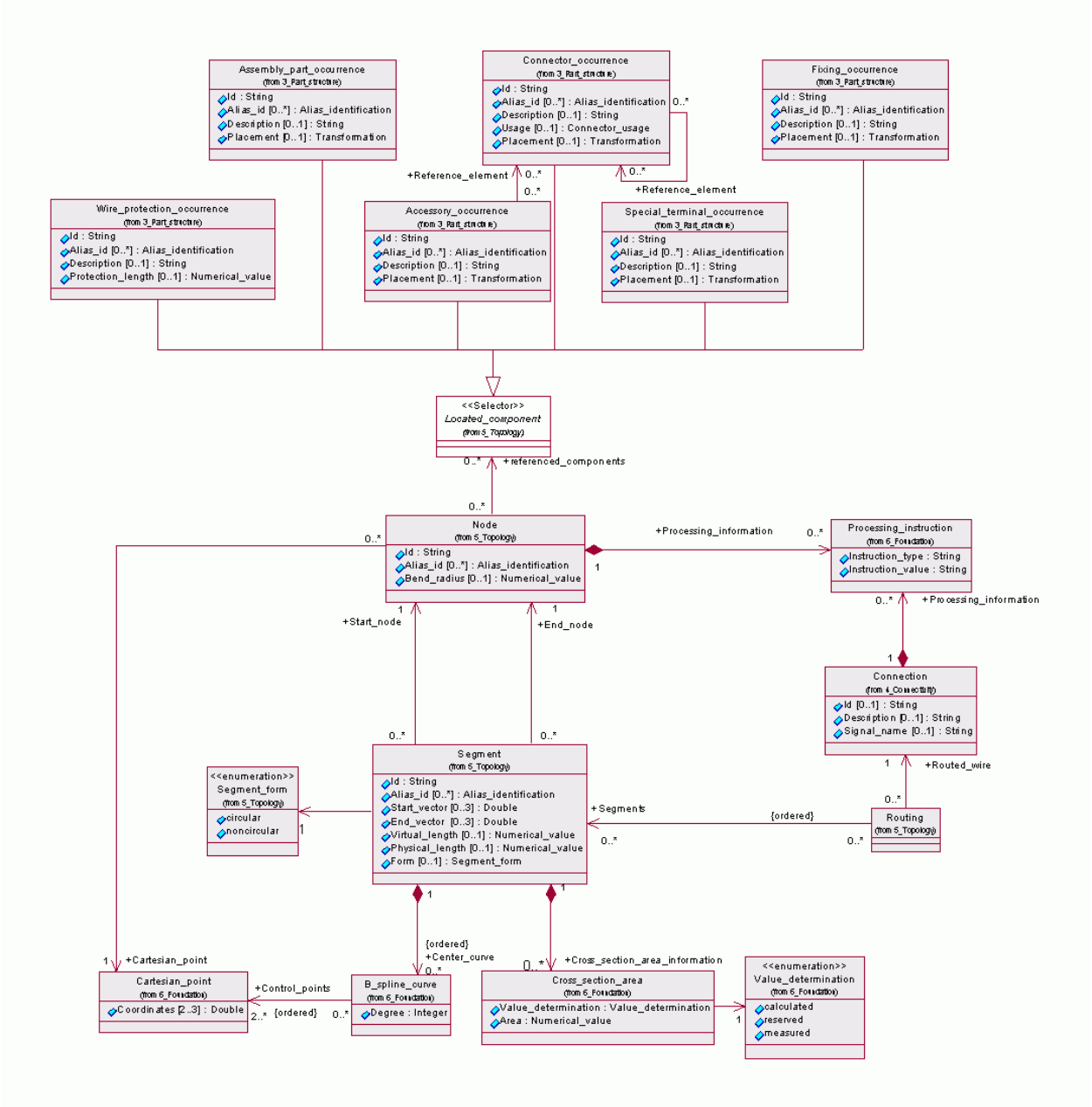

Diagram: (12) Topology and routing (1)

Diagram: (13) Topology and routing (2)

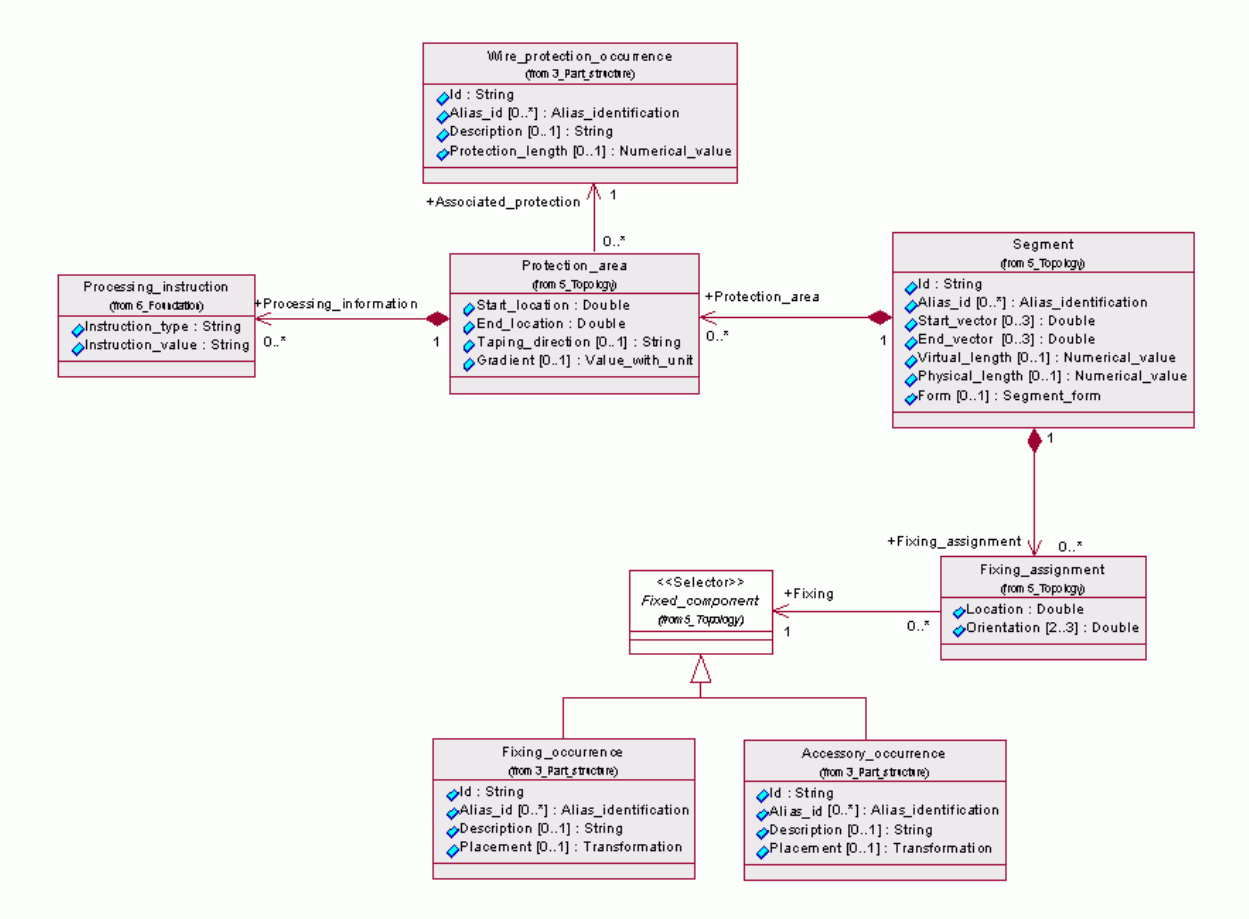

Diagram: (13) Topology and routing (2)

Diagram: (14) Dimensions

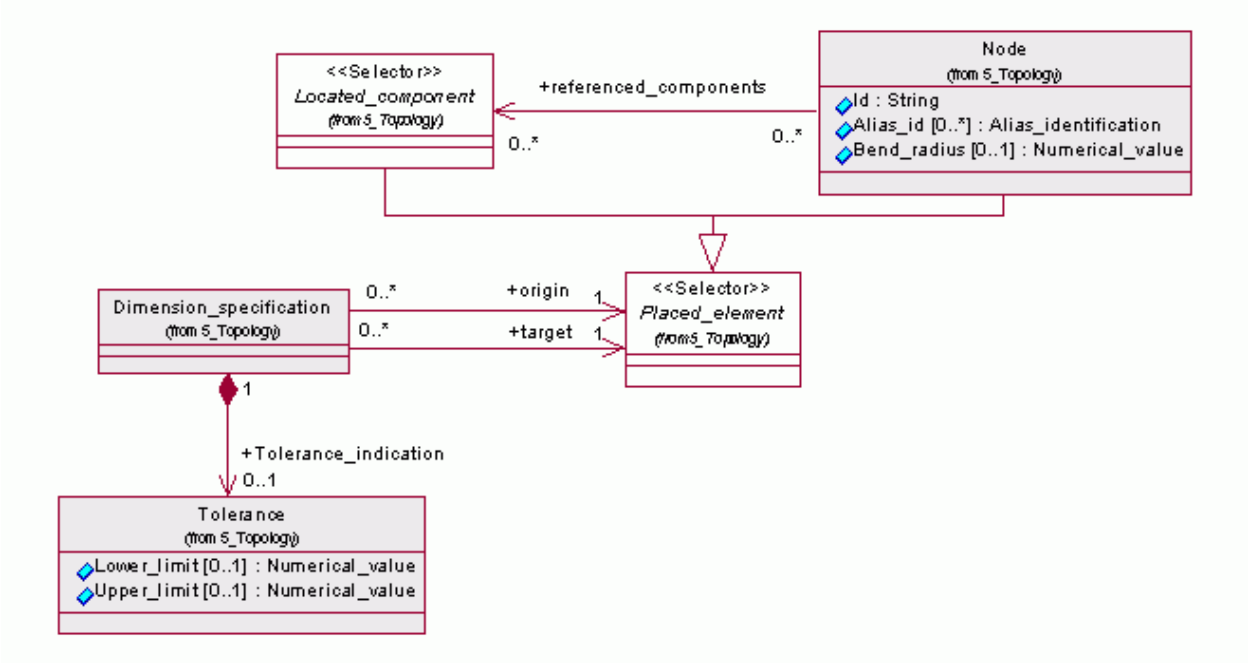

Diagram: (14) Dimensions

Diagram: (15) Miscellaneous

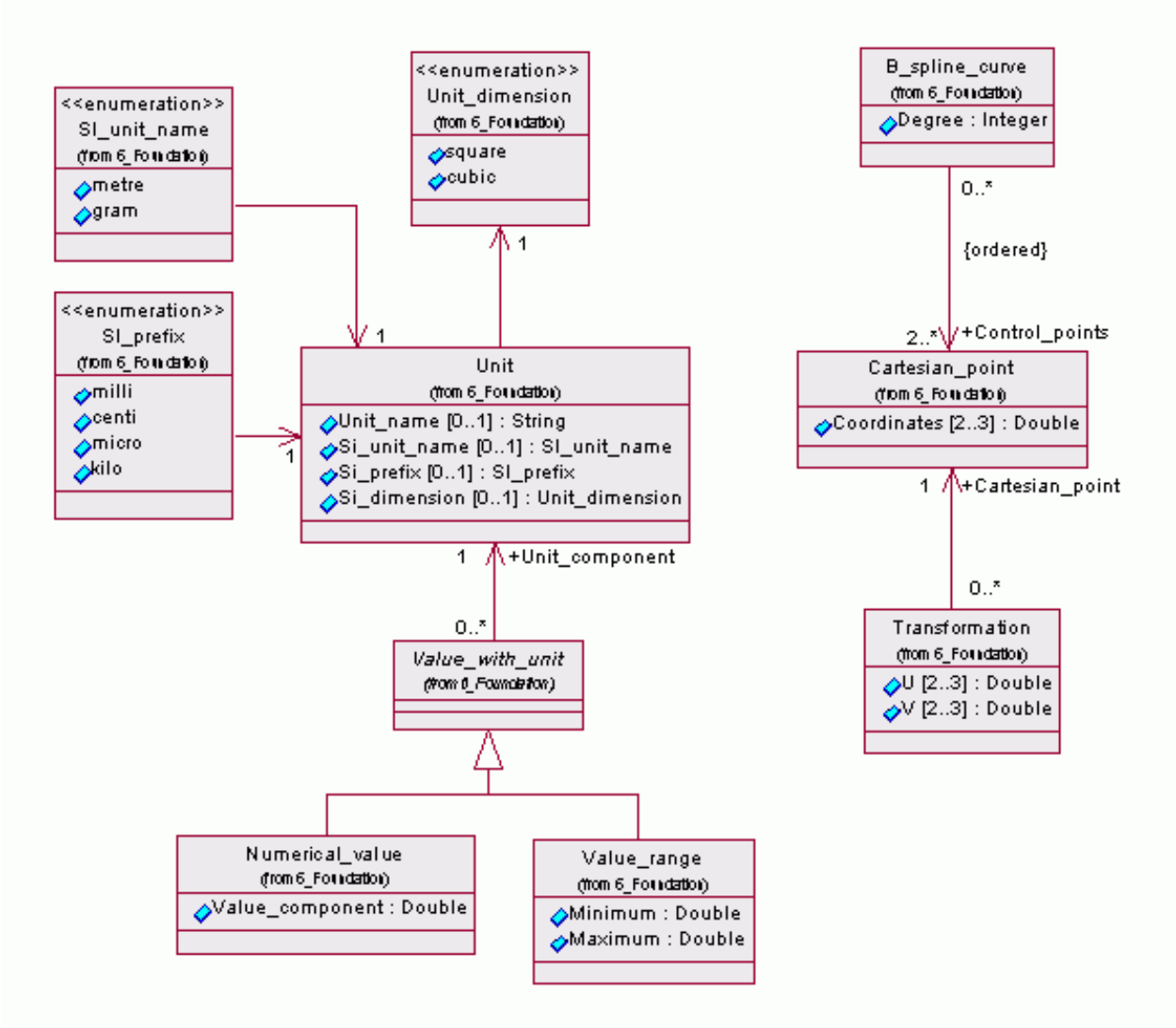

Diagram: (15) Miscellaneous

Diagram: (16) KBL_Container

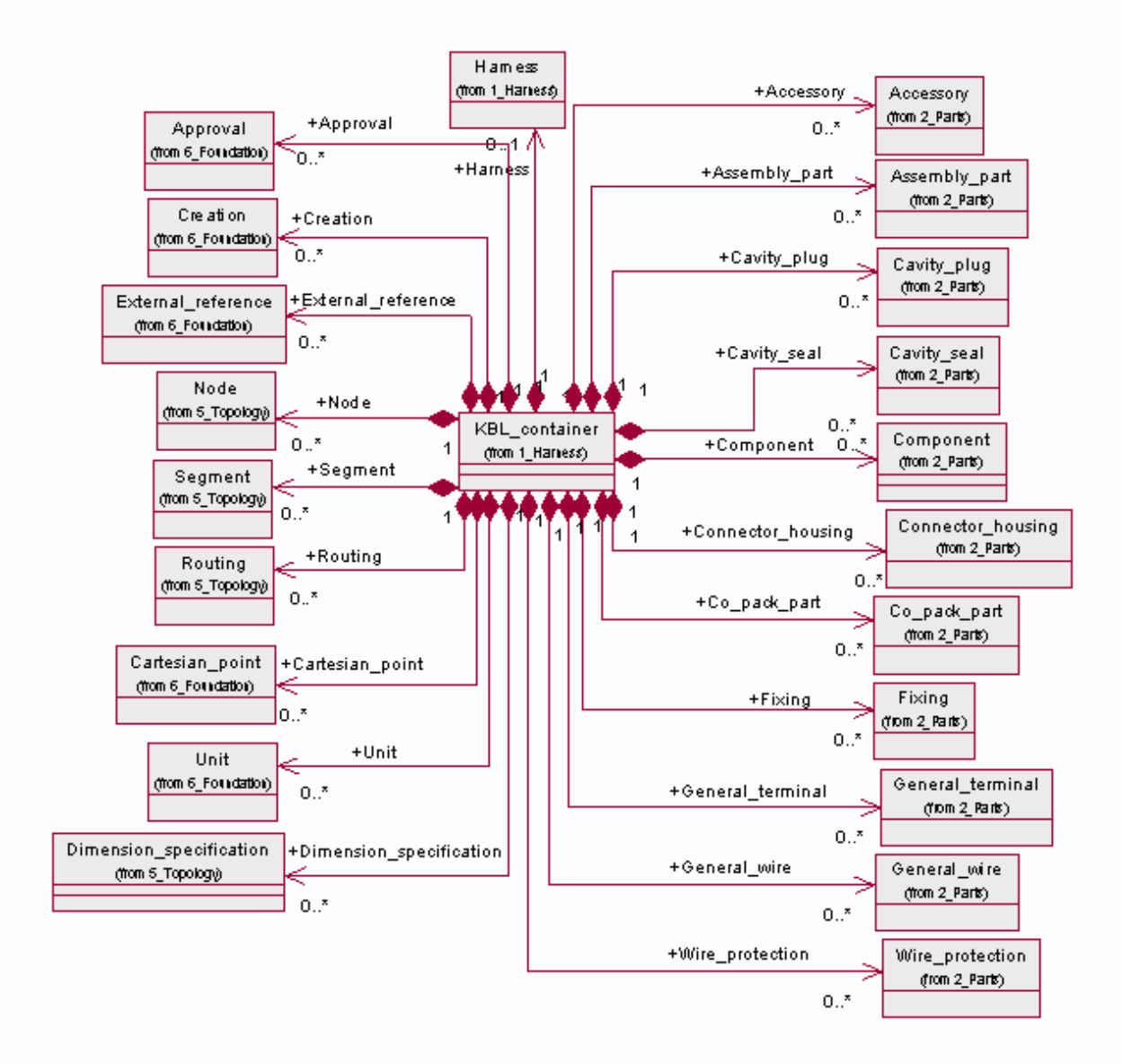

Diagram: (16) KBL_Container

Index

T

U

V

W