This document is Annex C of VDA Recommendation VDA 4967. This annex contains the technical description of SimPDM core data management functionality.

The recommendation is intended to standardize the integration of simulation and computation data in a product data management environment (PDM environment) in the form of simulation data management (SDM) and to define the communication processes between a simulation data management system (SDM system) and CAE systems as well as between an SDM system and other data management systems within the PDM environment.

With this purpose in mind, the recommendation defines use cases and functional requirements for simulation data management and a meta data model to describe simulation and computation data. The meta data model is modularly designed to allow a custom-tailored solution for simulation data management. Furthermore this recommendation describes necessary functionality and its clustering towards use case. Functionality and use cases are modular, too. So it is possible to customize SimPDM to the specific requirements of an enterprise.

The recommendation does not define an interchange data format between different CAE applications.

Version 2.0 of December 2008

Arbeitskreis "PLM"
Exclusion of liability

The VDA guidelines are recommendations which may be applied by everyone. The person using this guideline must however ensure the right application for each case.

They consider the state of technological development at the time of the respective version. No one can be exempted of responsibility for his own actions by using the VDA recommendations. This means that everyone acts at their own risk. A liability for the VDA and for the ones involved in the VDA recommendations is excluded.

Anyone who notices any form of incorrectness or possibility of false interpretation while using the VDA guidelines is asked to inform the VDA about this immediately so that possible faults can be removed.

This Recommendation is also published by the ProSTEP iViP Association, with the same title and version.

This Recommendation has been developed and is supported by the VDA and the ProSTEP iViP Association.
### Table of content

1. **Introduction** ........................................................................................................ 7
   1.1 Scope and Purpose ............................................................................................ 7
   1.2 Notation ............................................................................................................. 7
   1.2.1 Creating new instances of entities of database entities ......................... 7
   1.2.2 Deleting existing instances of database entities ....................................... 7
   1.2.3 Creating new relationships between database objects ......................... 7
   1.2.4 Deleting existing relationships between database objects .................... 8
   1.2.5 Change of attributes ............................................................................... 8
   1.3 Additional Remarks ....................................................................................... 8

2. **Functional description of SimPDM use cases** ...................................................... 8
   2.1 Functionality of use case Administration Management .................................. 8
   2.1.1 Create new administrative data ................................................................. 8
   2.1.2 Delete existing administrative data ........................................................... 10
   2.2 Functionality of use case Analysis Classification Management .................. 11
   2.2.1 Create new analysis type ......................................................................... 11
   2.2.2 Delete existing analysis type ................................................................... 11
   2.2.3 Create new analysis type hierarchy ......................................................... 11
   2.2.4 Delete existing analysis type hierarchy ................................................. 12
   2.3 Functionality of use case Analysis Definition Management .......................... 12
   2.3.1 Create new analysis ................................................................................ 12
   2.3.2 Delete existing analysis .......................................................................... 13
   2.3.3 Create new analysis version .................................................................... 14
   2.3.4 Delete existing analysis version .............................................................. 14
   2.3.5 Define new analysis version dependency .............................................. 15
   2.3.6 Delete existing analysis version dependency ........................................ 15
   2.3.7 Classify existing analysis ....................................................................... 15
   2.3.8 Delete existing analysis classification ................................................... 16
   2.4 Functionality of use case Document Management ......................................... 16
   2.4.1 Create new document ............................................................................ 16
   2.4.2 Delete existing document ....................................................................... 16
   2.4.3 Create new document association .......................................................... 17
   2.4.4 Delete existing document association ...................................................... 19
   2.4.5 Attach external file reference .................................................................. 20
   2.4.6 Delete external files reference ................................................................ 21
   2.5 Functionality of use case Load Case Definition Management ....................... 21
   2.5.1 Create new load ..................................................................................... 21
   2.5.2 Delete existing load ................................................................................ 22
   2.5.3 Assign property to load .......................................................................... 22
   2.5.4 Delete property assignment from load ................................................... 22
   2.5.5 Assign load to a model element .............................................................. 23
   2.5.6 Delete load assignment from model element .......................................... 23
   2.5.7 Assign a model as load (e.g. dummy, barrier) ....................................... 23
   2.5.8 Delete load model assignment from load ............................................... 24
   2.5.9 Assign load to an analysis ..................................................................... 24
   2.5.10 Delete load assignment from analysis .................................................. 25
2.5.11 Add another detailed load to an analysis .............................................. 25
2.5.12 Delete a detailed load assignment ......................................................... 26
2.6 Functionality of use case Model Assembly Management .......................... 26
  2.6.1 Create new model assembly relationship ............................................... 26
  2.6.2 Delete existing model assembly relationship ......................................... 27
2.7 Functionality of use case Model Configuration Management ................. 27
  2.7.1 Create abstract product ........................................................................ 28
  2.7.2 Delete abstract product .......................................................................... 28
  2.7.3 Create product component ..................................................................... 28
  2.7.4 Delete product component ...................................................................... 29
  2.7.5 Create component hierarchy .................................................................. 29
  2.7.6 Delete component hierarchy .................................................................. 29
  2.7.7 Create solution ....................................................................................... 30
  2.7.8 Delete solution ....................................................................................... 30
  2.7.9 Assign model to solution ........................................................................ 30
  2.7.10 Delete model assignment from solution ............................................... 31
  2.7.11 Create configuration within SimPDM .................................................... 31
  2.7.12 Assign configuration to BOM ................................................................. 31
  2.7.13 Delete configuration assignment from BOM ......................................... 32
  2.7.14 Add solution to configuration ................................................................. 32
  2.7.15 Delete solution from configuration ....................................................... 32
  2.7.16 Delete configuration ............................................................................... 33
  2.7.17 Assign configuration to analysis ............................................................. 33
  2.7.18 Delete configuration assignment from analysis .................................... 33
2.8 Functionality of use case Model Definition Management ....................... 34
  2.8.1 Create new model .................................................................................. 34
  2.8.2 Delete existing model ............................................................................. 34
  2.8.3 Create new model version ...................................................................... 35
  2.8.4 Delete existing model version ................................................................. 35
  2.8.5 Connect model version to analysis version ............................................ 36
  2.8.6 Disconnect model version from analysis version ................................... 36
  2.8.7 Define new model version dependency ............................................... 36
  2.8.8 Delete existing model version dependency .......................................... 37
2.9 Functionality of use case Output Specification Management ................... 37
  2.9.1 Create new Output Specification ............................................................. 37
  2.9.2 Delete existing Output Specification ...................................................... 38
  2.9.3 Define new group of interesting elements ............................................. 38
  2.9.4 Delete group definition of interesting elements ....................................... 38
2.10 Functionality of use case Parameter Association Management ................ 39
  2.10.1 Create new parameter .......................................................................... 39
  2.10.2 Delete existing parameter ...................................................................... 40
  2.10.3 Add a parameter unit ........................................................................... 40
  2.10.4 Delete a parameter unit ........................................................................ 41
2.11 Functionality of use case PDM Information Derivation Management ....... 41
  2.11.1 Create new CAD-PDM-Information ....................................................... 41
  2.11.2 Delete CAD-PDM-Information ............................................................... 42
  2.11.3 Derive CAD-PDM-Information .............................................................. 42
  2.11.4 Delete CAD-PDM-Information derivation ........................................... 42
  2.11.5 Assign BOM to analysis ........................................................................ 43
  2.11.6 Delete BOM assignment from analysis ............................................... 43
  2.11.7 Create BOM parts ................................................................................ 43
2.11.8 Deleting BOM parts.................................................................................. 44
2.11.9 Assign part to model.................................................................................. 44
2.11.10 Delete part assignment from model.......................................................... 44
2.11.11 Specify CAE model of shape representation.............................................. 45
2.11.12 Delete CAE model specification from shape representation..................... 45
2.11.13 Assign shape representation to CAE model.............................................. 45
2.11.14 Delete shape representation assignment from CAE model....................... 46
2.11.15 Assign connection element list to analysis................................................ 46
2.11.16 Delete connection element list assignment from analysis........................ 46
2.11.17 Create connection element....................................................................... 47
2.11.18 Delete connection element....................................................................... 47
2.12 Functionality of use case Post-processing Management................................ 48
2.12.1 Create analysis result.................................................................................. 48
2.12.2 Delete analysis result.................................................................................. 48
2.12.3 Create a computation output (result first order)......................................... 49
2.12.4 Delete a computation output (result first order)....................................... 49
2.12.5 Create a key result (result second order).................................................. 50
2.12.6 Delete a key result (result second order).................................................. 50
2.12.7 Create a report (result third order)............................................................. 51
2.12.8 Delete a report (result third order)............................................................. 51
2.12.9 Create a detailed post processing object.................................................... 52
2.12.10 Delete a detailed post processing object................................................... 52
2.12.11 Create a template..................................................................................... 53
2.12.12 Delete a template..................................................................................... 53
2.13 Functionality of use case Property Definition Management........................... 53
2.13.1 Create new property set.............................................................................. 53
2.13.2 Delete a property set.................................................................................. 54
2.13.3 Create new property set version.................................................................. 54
2.13.4 Delete existing property set version............................................................ 54
2.13.5 Create new property set version relationship............................................. 55
2.13.6 Delete existing property set version relationship........................................ 55
2.13.7 Create new property set structure............................................................... 55
2.13.8 Delete existing property set structure........................................................ 56
2.13.9 Create a property....................................................................................... 56
2.13.10 Delete a property....................................................................................... 56
2.13.11 Create property relationship..................................................................... 57
2.13.12 Delete existing property relationship....................................................... 57
2.13.13 Create property constraint....................................................................... 57
2.13.14 Delete existing property constraint......................................................... 58
2.13.15 Assign property set version to model version........................................... 58
2.13.16 Delete property set version from model version....................................... 58
2.13.17 Connect property set version to model element........................................ 59
2.13.18 Disconnect property set version to model element................................. 59
2.14 Functionality of use case Setting Definition Management............................... 59
2.14.1 Create new setting..................................................................................... 59
2.14.2 Delete existing setting................................................................................ 60
2.14.3 Assign setting to an analysis...................................................................... 60
2.14.4 Delete setting assignment from analysis................................................... 61
2.14.5 Create analysis step................................................................................... 61
2.14.6 Delete analysis step................................................................................... 62
2.14.7 Add another detailed setting to an analysis.............................................. 62
2.14.8 Delete a detailed setting assignment ...........................................................63
2.15 Functionality of use case Topology Element Definition Management ..........63
  2.15.1 Create new topological element ...............................................................63
  2.15.2 Delete existing topological element ..........................................................64
  2.15.3 Create topological element relationship ...................................................64
  2.15.4 Delete topological element relationship ...................................................64
  2.15.5 Include model element into model ............................................................65
  2.15.6 Delete model element from model ............................................................65
2.16 Functionality of use case Topology Structure Definition Management ..........65
  2.16.1 Define interface element .......................................................................65
  2.16.2 Delete interface element .......................................................................66
  2.16.3 Create topological model connection .......................................................66
  2.16.4 Delete topological model connection .......................................................67
  2.16.5 Assign a surface to a set .......................................................................67
  2.16.6 Delete a surface from a set .....................................................................67
1 Introduction

1.1 Scope and Purpose

The use cases defined by the recommendation are described by their general purpose within the context of simulation data management with regards to their value progress. The describes the atomic data management system functions which are needed to perform the use case, for instance, ‘Create new model’ as an atomic function of the model definition management. Each function itself is described by the start state and precondition as well as the end state. Each described function requires operations on data objects or the creation of new data objects. These operations are listed for each function and directly refer to the SimPDM data model. The operations are named by a formal code. Last but not least additional notes and remarks are provided, where applicable.

This annex contains the technical description of SimPDM core data management functionality.

1.2 Notation

This section gives an overview of the used notation for the SimPDM core data management functionality. It is also valid for Annex D defining SimPDM system communication functionality.

1.2.1 Creating new instances of entities of database entities

The creation of new instances of entities is described by the following formal code.

- Create.<Package name>.<entity name>

Examples are

- Create.BASE.model_version
- Create.TOPO.model_element
- Create.PROP.property_set

1.2.2 Deleting existing instances of database entities

The deletion of new instances of entities is described by the following formal code.

- Delete.<Package name>.<entity name>

Examples are

- Delete.SETT.general_setting
- Delete.BASE.administration
- Delete.PROP.general_property

1.2.3 Creating new relationships between database objects

The Creation of new relationships between instances of entities is described by the following formal code.

- Connect.<Package name>.<relationship name>

Examples are

- Connect.BASE.result_first_order
- Connect.CONF.solution_model
- Connect.PROP.property_document
1.2.4 Deleting existing relationships between database objects

The deletion of relationships between instances of entities is described by the following formal code.

- Disconnect.<Package name>.<relationship name>

Examples are
- Disconnect.LOAD.applied_load_model
- Disconnect.BASE.model_parameter
- Disconnect.CAD.used_shape

1.2.5 Change of attributes

The setting or changing of attribute values is described by the following formal code

- Set.<package name>.<class name>.<attribute name>

Examples are
- Set.BASE.parameter.name
- Set.BASE.model_version.description
- Set.PROP.general_property.property_specification

1.3 Additional Remarks

The document does not claim the completeness of meta data model operations. Especially the change operations for attribute values are only described in a few functions where the attribute value is significant.

The order of the required operations are listed in this document does not imply a mandatory order for the operations to be performed by a data management system.

Generally the meta data model is the fundamental base definition for all described functionality. The required operations directly refer to the meta data model.

2 Functional description of SimPDM use cases

2.1 Functionality of use case Administration Management

2.1.1 Create new administrative data

2.1.1.1 Start state and preconditions

An instance of one of the following classes, able to carry administrative data, does exist.

- BASE.analysis_version
- BASE.analysis
- BASE.computation_output
- BASE.document
- BASE.key_result
- BASE.model
- BASE.model_version
- BASE.report
- BASE.template
- CAD.cadpdm_information
2.1.1.2 Required operations

- Create.BASE.administration
And, depending on the class instance for which the new instance BASE.administration is to be defined, one of the relationships:
  - Connect.BASE.analysis_version_administration
  - Connect.BASE.analysis_administration
  - Connect.BASE.computation_output_administration
  - Connect.BASE.document_administration
  - Connect.BASE.key_result_administration
  - Connect.BASE.model_administration
  - Connect.BASE.model_version_administration
  - Connect.BASE.report_administration
  - Connect.BASE.template_administration
  - Connect.CAD.cad_or_pdm_information_administration
  - Connect.CONF.configuration_administration
  - Connect.CONF.product_administration
  - Connect.CONF.product_component_administration
  - Connect.CONF.solution_administration
  - Connect.LOAD.load_definition_administration
  - Connect.PROP.property_set_administration
  - Connect.PROP.property_set_version_administration
  - Connect.SETT.setting_administration

2.1.1.3 End state and post condition

Administrative data is exclusively assigned to an instance of one of the classes:
  - BASE.analysis_version
  - BASE.analysis
  - BASE.computation_output
  - BASE.document
  - BASE.key_result
  - BASE.model
2.1.1.4 Notes and remarks
No further notes and remarks available

2.1.2 Delete existing administrative data

2.1.2.1 Start state and preconditions
An instance of the class BASE.administration to be deleted does exist.

2.1.2.2 Required operations
- Delete.BASE.administration
- And, depending on the class instance to which BASE.administration is related, one of
  - Disconnect.BASE.analysis_version_administration
  - Disconnect.BASE.analysis_administration
  - Disconnect.BASE.computation_output_administration
  - Disconnect.BASE.document_administration
  - Disconnect.BASE.key_result_administration
  - Disconnect.BASE.model_administration
  - Disconnect.BASE.model_version_administration
  - Disconnect.BASE.report_administration
  - Disconnect.BASE.template_administration
  - Disconnect.CAD.cadpdm_information_administration
  - Disconnect.CONF.configuration_administration
  - Disconnect.CONF.product_administration
  - Disconnect.CONF.product_component_administration
  - Disconnect.CONF.solution_administration
  - Disconnect_LOAD.load_definition_administration
  - Disconnect.PROP.property_set_administration
  - Disconnect.PROP.property_set_version_administration
2.1.2.3 End state and post condition
The administrative data is no longer available.

2.1.2.4 Notes and remarks
The deletion of an instance of the class BASE.administration at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.2 Functionality of use case Analysis Classification Management
See SimPDM recommendation for a general description of the use case Analysis Classification Management.

2.2.1 Create new analysis type
2.2.1.1 Start state and preconditions
No specific precondition to be defined.

2.2.1.2 Required operations
- Create.BASE.analysis_type

2.2.1.3 End state and post condition
A new instance of the class BASE.analysis_type is available to support the analysis classification management.

2.2.1.4 Notes and remarks
Actually, the creation of a new instance of the class BASE.analysis_type is not a daily business, but rather an administrative task.

2.2.2 Delete existing analysis type
2.2.2.1 Start state and preconditions
An instance of the class BASE.analysis_type to be deleted does exist and is in a state allowing it to be deleted, i.e., it is not associated with any instance of the class BASE.analysis and is not associated with another instance of the class BASE.analysis_type with role of a next higher analysis type the within an analysis type hierarchy.

2.2.2.2 Required operations
- Delete.BASE.analysis_type

2.2.2.3 End state and post condition
The deleted instance of the class BASE.analysis_type is no longer available within the data management system.

2.2.2.4 Notes and remarks
Actually, the deletion of an instance of the class BASE.analysis_type is not a daily business, but rather an administrative task.

The deletion of an instance of the class BASE.analysis_type at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.2.3 Create new analysis type hierarchy
2.2.3.1 Start state and preconditions
At least two instances of the class BASE.analysis exist, which have not yet an analysis type hierarchy relationship with each other.

2.2.3.2 Required operations

- Connect.BASE.analysis_type_hierarchy

2.2.3.3 End state and post condition

Two instances of the class BASE.analysis have a hierarchical relationship with each other, where one instance of the class has the role of a sub type and the other instance the role of a next higher analysis type.

2.2.3.4 Notes and remarks

Actually, the creation of an analysis type hierarchy is not a daily business, but rather an administrative task.

If the instance of the class BASE.analysis, supposed to be the subtype, already has the role of a subtype in another analysis type hierarchy relationship, the former relationship is implicitly be deleted. An instance of the class BASE.analysis_type can only be involved in one analysis type hierarchy relationship with the role of a subtype.

2.2.4 Delete existing analysis type hierarchy

2.2.4.1 Start state and preconditions

An instance of the relationship BASE.analysis_type_hierarchy to be deleted does exist and is in a state allowing it to be deleted, i.e., the instance of the class BASE.analysis_type with the role of the subtype is not associated with an instance of the class BASE.analysis.

2.2.4.2 Required operation

- Disconnect.BASE.analysis_type_hierarchy

2.2.4.3 End state and post condition

The analysis type hierarchy relationship is no longer available within the data management system. The two formerly connected instances of the class BASE.analysis_type are still available.

2.2.4.4 Notes and remarks

Actually, the deletion of an analysis type hierarchy is not a daily business, but rather an administrative task.

The deletion of an analysis type hierarchy at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.3 Functionality of use case Analysis Definition Management

See SimPDM recommendation for a general description of the use case Analysis Definition Management.

2.3.1 Create new analysis

2.3.1.1 Start state and preconditions

No start state and precondition to be defined.

2.3.1.2 Required operations

- Create.BASE.analysis
- Create.BASE.analysis_version

Copyright: VDA
2.3.1.3 End state and post condition

A new instance of the class BASE.analysis and a new initial instance of the class BASE.analysis_version for the new instance of the class BASE.analysis are available within the data management system.

2.3.1.4 Notes and remarks

Instances of the classes BASE.analysis and BASE.analysis_version are only administrative data management nodes and only have some identifying information and a description. All detailed information for an analysis / simulation are assigned to the instance of the class BASE.analysis by separate class instances of different types.

Optionally the use cases Administration Management and Document Management might be applied for the new instances of the classes BASE.analysis and BASE.analysis_version.

2.3.2 Delete existing analysis

2.3.2.1 Start state and preconditions

The instance of the class BASE.analysis to be deleted has not more than one associated instances class BASE.analysis_version, i.e. it has only the initial instance of the class BASE.analysis_version. This instance of the class BASE.analysis_version has no relationships with other instances of the class BASE.analysis_version. This instance of the class BASE.analysis_version has no associated information like models, output specification, results; it is not part of one of the relationships BASE.output_specification, BASE.analysis_specific_result or BASE.analysis_version_model_version. The instances of the classes BASE.analysis and the BASE.analysis_version might have associated instances of the classes BASE.administration and BASE.document.

2.3.2.2 Required operations

- Delete.BASE.analysis_version
- Delete.BASE.analysis
- Disconnect.BASE.analysis_version

And if the instance of the class BASE.analysis or the initial instance of the class BASE.analysis_version has associated administrative data...

- Disconnect.BASE.analysis_administration
- Disconnect.BASE.analysis_version_administration
- Delete.BASE.administration

And if the initial instance of the class BASE.analysis_version has an associated document, all required operation for the deletion of documents (see chapter 2.4.2)

2.3.2.3 End state and post condition

The deleted instance of the class BASE.analysis is no longer available within the data management system.
2.3.2.4 Notes and remarks

The deletion of an instance of the class BASE.analysis at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.3.3 Create new analysis version

2.3.3.1 Start state and preconditions

An instance of the class BASE.analysis with an initial instance of the class BASE.analysis_version does exist.

2.3.3.2 Required operations

- Create.BASE.analysis_version
- Connect.BASE.analysis_version
- Connect.BASE.analysis_version_relationship

2.3.3.3 End state and post condition

A new instance of the class BASE.analysis_version for an instance of the class BASE.analysis is available. The new instance of the class BASE.analysis_version is connected with another already existing instance of the class BASE.analysis_version of the same instance of the class BASE.analysis with the role as a succeeding version within a predecessor – successor - relationship.

2.3.3.4 Notes and remarks

Optionally the use cases Administration Management and Document Management might be applied for BASE.analysis_version.

2.3.4 Delete existing analysis version

2.3.4.1 Start state and preconditions

The instance of the class BASE.analysis_version to be deleted has no relationships with instances of the class BASE.analysis_version belonging to separate instances of BASE.analysis, no succeeding instances of the class BASE.analysis_version, and has no associated information like models, output specification, results, i.e., it is not part of one of the relationships BASE.output_specification, BASE.analysis_specific_result or BASE.analysis_version_model_version. The instances of the classes BASE.analysis and the BASE.analysis_version might have associated instances of the classes BASE.administration and BASE.document.

2.3.4.2 Required operations

- Disconnect.BASE.analysis_version
- Delete.BASE.analysis_version

And if the model version has an associated administrative data...

- Disconnect.BASE.analysis_version_administration
- Delete.BASE.administration

And if the initial instance of the class BASE.analysis_version has an associated document, all required operation for the deletion of documents (see chapter 2.4.2)

2.3.4.3 End state and post condition

The deleted instance of the class BASE.analysis_version is no longer available within the data management system.

Copyright: VDA
2.3.4.4 Notes and remarks

The deletion of an instance of the class BASE.analysis_version at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.3.5 Define new analysis version dependency

2.3.5.1 Start state and preconditions

Two instances of the class BASE.analysis_version associated with different instances of the class BASE.analysis do exist.

2.3.5.2 Required operations

- Connect.BASE.analysis_version_relationship

2.3.5.3 End state and post condition

Two instances of the class BASE.analysis_version associated with different instances of BASE.analysis are associated with each other in a general manner to define a dependency.

2.3.5.4 Notes and remarks

No further notes and remarks available

2.3.6 Delete existing analysis version dependency

2.3.6.1 Start state and preconditions

An instance of the relationship BASE.analyze_version_relationship between instances of the class BASE.analysis_version associated with different instances of the class BASE.analysis does exist.

2.3.6.2 Required operations

- Delete.BASE.analysis_version_relationship

2.3.6.3 End state and post condition

The analysis version dependency defined by an instance of the relationship BASE.analyze_version_relationship between two instances of the class BASE.analysis_version associated with different instances of the class BASE.analysis is no longer available within the data management system.

2.3.6.4 Notes and remarks

The deletion of the dependency between two instances of the class BASE.analyze_version has not any influence on the availability and usability of the formally connected instances of the class BASE.analyze_version.

2.3.7 Classify existing analysis

2.3.7.1 Start state and preconditions

An instance of the class BASE.analysis and an instance of the class BASE.analysis_type do exist.

2.3.7.2 Required operations

- Connect.BASE.analysis_classification

2.3.7.3 End state and post condition

An instance of the class BASE.analysis_type is assigned to the instance of the class BASE.analysis
2.3.7.4 Notes and remarks
No further notes and remarks available

2.3.8 Delete existing analysis classification

2.3.8.1 Start state and preconditions
An instance of the relationship BASE.analysis_classification between an instance of the class BASE.analysis and an instance of the class BASE.analysis_type do exist.

2.3.8.2 Required operations
- Disconnect.BASE.analysis_classification

2.3.8.3 End state and post condition
The classification relationship between an instance of the class BASE.analysis and an instance of the class BASE.analysis_type is deleted.

2.3.8.4 Notes and remarks
No further notes and remarks available

2.4 Functionality of use case Document Management

See SimPDM recommendation for a general description of the use case Document Management.

2.4.1 Create new document

2.4.1.1 Start state and preconditions
No start state or precondition to be defined

2.4.1.2 Required operations
- Create.BASE.document

2.4.1.3 End state and post condition
An instance of the class BASE.document is available within the data management system able to carry one or more instances of the class BASE.reference_to_file. The new document is not yet associated to simulation data management objects.

2.4.1.4 Notes and remarks
Optionally the use case Administration Management might be applied for BASE.document.

2.4.2 Delete existing document

2.4.2.1 Start state and preconditions
An instance of the class BASE.document to be deleted does exist, must not carry any external files and is in a stage allowing it to be deleted, i.e. it is not in use, respectively must not have relationships with more than one instance of one of the following classes.
- BASE.analysis_version
- BASE.computation_output
- BASE.key_result
- BASE.load_for_analysis
- BASE.model_version
- BASE.output_request
2.4.2.2 Required operations

Delete BASE.document
And if the document has an associated administration object

- Disconnect BASE.document_administration
- Delete BASE.administration

2.4.2.3 End state and post condition

The deleted instance of the class BASE.document is no longer available within the data management system.

2.4.2.4 Notes and remarks

The deletion instances of the class BASE.document at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.4.3 Create new document association

2.4.3.1 Start state and preconditions

An instance of the class BASE.document and at least one instance of one of the following classes does exist.

- BASE.analysis_version
- BASE.computation_output
- BASE.key_result
- BASE.load_for_analysis
- BASE.model_version
- BASE.output_request
- BASE.report
- BASE.setting_for_analysis
- BASE.template
- CAD.cadpdm_information
- LOAD.load_definition
- PROP.property
- PROP.property_constraint
2.4.3.2 Required operations

Depending on the class instance to which the document is to be associated, one of

- Connect.BASE.analysis_version_document
- Connect.BASE.computation_output_document
- Connect.BASE.key_result_document
- Connect.BASE.load_for_analysis_document
- Connect.BASE.model_version_document
- Connect.BASE.output_request_document
- Connect.BASE.report_document
- Connect.BASE.setting_for_analysis_document
- Connect.BASE.template_document
- Connect.CAD.cad_or_pdm_information_document
- Connect.LOAD.load_definition_document
- Connect.PROP.property_document
- Connect.PROP.property_constraint_document
- Connect.SETT.setting_document
- Connect.TOPO.model_element_document

2.4.3.3 End state and post condition

An instance of the class BASE.document is associated to one instance of one of the following classes.

- BASE.analysis_version
- BASE.computation_output
- BASE.key_result
- BASE.load_for_analysis
- BASE.model_version
- BASE.output_request
- BASE.report
- BASE.setting_for_analysis
- BASE.template
- CAD.cadpdm_information
- LOAD.load_definition
- PROP.property
- PROP.property_constraint
- SETT.setting
- TOPO.model_element
2.4.3.4 Notes and remarks

An instance of BASE.document can only have one exclusive relationship to a specific instance of one of the classes listed above. Nevertheless, an instance of the class BASE.document might have several relationships to instances of different classes or different instances of one and the same class.

2.4.4 Delete existing document association

2.4.4.1 Start state and preconditions

A relationship between an instance of the class BASE.document and one instance of one of the following classes does exist,

- BASE.analysis_version
- BASE.computation_output
- BASE.key_result
- BASE.load_for_analysis
- BASE.model_version
- BASE.output_request
- BASE.report
- BASE.setting_for_analysis
- BASE.template
- CAD.cadpdm_information
- LOAD.load_definition
- PROP.property
- PROP.property_constraint
- SETT.setting
- TOPO.model_element

2.4.4.2 Required operations

Depending on the class instance to which BASE.documentation is related, one of

- Disconnect.BASE.analysis_version_document
- BASE.computation_output
- BASE.key_result
- BASE.load_for_analysis
- Disconnect.BASE.model_version_document
- Disconnect.BASE.output_request_document
- BASE.report
- BASE.setting_for_analysis
- Disconnect.BASE.template_document
- Disconnect.CAD.cad_or_pdm_information_document
- Disconnect.LOAD.load_definition_document
- Disconnect.PROP.property_document

Copyright: VDA
2.4.4.3 End state and post condition
A specific instance of the class BASE.document is no longer associated with one specific instance of one of the following classes.

- BASE.analysis_version
- BASE.computation_output
- BASE.key_result
- BASE.load_for_analysis
- BASE.model_version
- BASE.output_request
- BASE.report
- BASE.setting_for_analysis
- BASE.template
- CAD.cadpdm_information
- LOAD.load_definition
- PROP.property
- PROP.property_constraint
- SETT.setting
- TOPO.model_element

Nevertheless, the instance of the class BASE.document still exists and may still have associations to other instances of one of the listed classes.

2.4.4.4 Notes and remarks
The deletion of relationships at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.4.5 Attach external file reference
2.4.5.1 Start state and preconditions
An instance of the class BASE.document as an administrative container object ("external file in context") exists.

2.4.5.2 Required operations
- Create.BASE.reference_to_file
- Connect.BASE.document_file

2.4.5.3 End state and post condition
An instance of the class BASE.reference_to_file is available within the data management system and is, using an instance of the class BASE.document as the administrative container ("external file in context"), associated with an instance of the one of the following classes.

- BASE.analysis_version

Copyright: VDA
2.4.5.4 Notes and remarks
An instance of the class BASE.document can carry more than one instances of the class BASE.reference_to_file but an instance of the class BASE.reference_to_file can only be associated to one instance of the class BASE.document. Nevertheless, an instance of the class BASE.document should not be redundantly associated with more than one instance of the class BASE.reference_to_file referring to the same external file. Within a data management system a specific external file can be referred by more than instance o the class BASE.references_to_file associated to different instances of the class BASE.document.

2.4.6 Delete external files reference

2.4.6.1 Start state and preconditions
An instance of the class BASE.reference_to_file does exist.

2.4.6.2 Required operations
- Delete.BASE.reference_to_file
- Disconnect.BASE.document_file

2.4.6.3 End state and post condition
The instance of the class BASE.reference_to_file is no longer available within the data management system.

2.4.6.4 Notes and remarks
The SimPDM data model does not define the deletion of the external file itself, since deletion of files not only depends on the data management functionalities and rights but also on operation system functionalities and rights.

2.5 Functionality of use case Load Case Definition Management
See SimPDM recommendation for a general description of the use case Load Case Definition Management.

2.5.1 Create new load
2.5.1.1 Start state and preconditions
No start state and precondition to be defined.

2.5.1.2 Required operations
- Create.LOAD.load_definition (or one of the subtypes)

2.5.1.3 End state and post condition
A new LOAD.load_definition definition is available within the data management system.

2.5.1.4 Notes and remarks
No further notes and remarks available.

2.5.2 Delete existing load

2.5.2.1 Start state and preconditions
A LOAD.load_definition to be deleted does exist and is in a state allowing change, i.e. it has no further objects like BASE.document, etc. assigned. The LOAD.load_definition might have an associated BASE.administration which is also subject to be deleted then (see chapter 2.1).

2.5.2.2 Required operations
- Delete.LOAD.load_definition (or one of the subtypes)

2.5.2.3 End state and post condition
The LOAD.load_definition definition is no longer available within the data management system.

2.5.2.4 Notes and remarks
The deletion of LOAD.load_definition at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.5.3 Assign property to load

2.5.3.1 Start state and preconditions
A LOAD.load_definition or one of its subtypes exists and is in a state allowing change. Furthermore a PROP.property_set_version does exist.

2.5.3.2 Required operations
- Connect.LOAD.load_property_set_version

2.5.3.3 End state and post condition
A LOAD.load_definition or one of its subtypes has assigned a PROP.property_set_version within the data management system.

2.5.3.4 Notes and remarks
The PROP.property_set_version already existed within the data management system. Creation is subject of another use case (see chapter 2.13).

2.5.4 Delete property assignment from load

2.5.4.1 Start state and preconditions
A LOAD.load_definition or one of its subtypes has assigned a PROP.property_set_version and is in a state allowing change.

2.5.4.2 Required operations
2.5.4.3 **End state and post condition**
A LOAD.load_definition or one of its subtypes has no longer assigned a PROP.property_set_version within the data management system.

2.5.4.4 **Notes and remarks**
The PROP.property_set_version still exists within the data management system. Deletion is subject of another use case (see chapter 2.13).

The deletion of the relation LOAD.load_property_set_version at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.5.5 **Assign load to a model element**

2.5.5.1 **Start state and preconditions**
A LOAD.load_definition or one of its subtypes exists and is in a state allowing change. Furthermore a TOPO.model_element or one of its subtypes does exist.

2.5.5.2 **Required operations**
- Connect.LOAD.load_definition_model_element

2.5.5.3 **End state and post condition**
A LOAD.load_definition or one of its subtypes has assigned a TOPO.model_element or one of its subtypes within the data management system.

2.5.5.4 **Notes and remarks**
The TOPO.model_element or one of its subtypes already existed within the data management system. Creation is subject of another use case (see chapter 2.15).

2.5.6 **Delete load assignment from model element**

2.5.6.1 **Start state and preconditions**
A LOAD.load_definition or one of its subtypes has assigned a TOPO.model_element or one of its subtypes and is in a state allowing change.

2.5.6.2 **Required operations**
- Disconnect.LOAD.load_definition_model_element

2.5.6.3 **End state and post condition**
A LOAD.load_definition or one of its subtypes has no longer assigned a TOPO.model_element or one of its subtypes within the data management system.

2.5.6.4 **Notes and remarks**
The TOPO.model_element or one of its subtypes still exists within the data management system. Deletion is subject of another use case (see chapter 2.15).

The deletion of the relation LOAD.load_definition_model_element at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.5.7 **Assign a model as load (e.g. dummy, barrier)**

2.5.7.1 **Start state and preconditions**
A LOAD.applied_load_case exists and is in a state allowing change. Furthermore a BASE.model_version does exist.
2.5.7.2 Required operations

- Connect.LOAD.applied_load_model

2.5.7.3 End state and post condition

A LOAD.applied_load_case has assigned a BASE.model_version within the data management system.

2.5.7.4 Notes and remarks

No further notes and remarks available.

2.5.8 Delete load model assignment from load

2.5.8.1 Start state and preconditions

A LOAD.applied_load_case has assigned a BASE.model_version and is in a state allowing change.

2.5.8.2 Required operations

- Disconnect.LOAD.applied_load_model

2.5.8.3 End state and post condition

A LOAD.applied_load_case or one of its subtypes has no longer assigned a BASE.model_version within the data management system.

2.5.8.4 Notes and remarks

The LOAD.applied_load_case and the BASE.model_version still exist within the data management system. Deletion is subject of another use case (see chapter 2.8).

The deletion of the relation LOAD.applied_load_model at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.5.9 Assign load to an analysis

2.5.9.1 Start state and preconditions

A BASE.analysis_version does exist and is in a state allowing change. For high granularity load management existence of LOAD.load_definition or one of its subtypes can be necessary.

2.5.9.2 Required operations

- Create.BASE.load_for_analysis
- Connect.BASE.load_for_analysis_version

And, in case of high granularity level data management, additionally

- Create.LOAD.applied_load_case
- Connect.LOAD.analysis_applied_load_case

And, at least one of ...

- Connect.LOAD.load_applied_load_case
- Connect.LOAD.boundary_condition_applied_load_case
- Connect.LOAD.script_applied_load_case
- Connect.LOAD.start_condition_applied_load_case

2.5.9.3 End state and post condition
A `BASE.load_for_analysis` is assigned to a `BASE.analysis_version`. In case of high level granularity this is further detailed in the assigned `LOAD.applied_load_case` and its assignments.

### 2.5.9.4 Notes and remarks

`BASE.load_for_analysis` can be defined in a document. Document attachment in general is subject of another use case (see chapter 2.4).

### 2.5.10 Delete load assignment from analysis

#### 2.5.10.1 Start state and preconditions

A `BASE.load_for_analysis` is assigned to a `BASE.analysis_version` which is in a state allowing change. In case of high level granularity the load is further detailed in the assigned `LOAD.applied_load_case` and its assignments which also exist and are in a state allowing change. There is no `BASE.document` attached to `BASE.load_for_analysis` (see chapter 2.4).

#### 2.5.10.2 Required operations

- `Delete.BASE.load_for_analysis`
- `Disconnect.BASE.load_for_analysis_version`

And, in case of high granularity level data management, additionally

- `Delete.LOAD.applied_load_case`
- `Disconnect.LOAD.analysis_applied_load_case`

And, depending on assigned loads,

- `Disconnect.LOAD.load_applied_load_case`
- `Disconnect.LOAD.boundary_condition_applied_load_case`
- `Disconnect.LOAD.script_applied_load_case`
- `Disconnect.LOAD.start_condition_applied_load_case`

#### 2.5.10.3 End state and post condition

A `BASE.load_for_analysis` is no longer available within the data management system. Also no longer available are the `LOAD.applied_load_case` and the assignments.

#### 2.5.10.4 Notes and remarks

The `LOAD.load_definition` or one of its subtypes and the `BASE.model_version` still exist within the data management system. Deletion is subject of another use case (see chapter 2.8).

The deletion of load assignments at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

### 2.5.11 Add another detailed load to an analysis

#### 2.5.11.1 Start state and preconditions

A `LOAD.applied_load_case` does exist and is in a state allowing change. A `LOAD.load_definition` or one of its subtypes exists.

#### 2.5.11.2 Required operations

One of …

- `Connect.LOAD.load_applied_load_case`

Copyright: VDA
2.5.11.3 End state and post condition
Another LOAD.load_definition or one of its subtypes is assigned to a LOAD.applied_load_case within the data management system.

2.5.11.4 Notes and remarks
No further notes and remarks available.

2.5.12 Delete a detailed load assignment

2.5.12.1 Start state and preconditions
A LOAD.load_definitions or one of its subtypes are assigned to a LOAD.applied_load_case within the data management system. The LOAD.applied_load_case is in a state allowing change.

2.5.12.2 Required operations
Depending on assigned loads, one of …
- Disconnect.LOAD.load_applied_load_case
- Disconnect.LOAD.boundary_condition_applied_load_case
- Disconnect.LOAD.script_applied_load_case
- Disconnect.LOAD.start_condition_applied_load_case

2.5.12.3 End state and post condition
A LOAD.load_definition or one of its subtypes is no longer assigned to a LOAD.applied_load_case within the data management system.

2.5.12.4 Notes and remarks
The LOAD.load_definition or one of its subtypes and the LOAD.applied_load_case still exist within the data management system.

The deletion of load assignments at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.6 Functionality of use case Model Assembly Management

See SimPDM recommendation for a general description of the use case Model Assembly Management.

2.6.1 Create new model assembly relationship

2.6.1.1 Start state and preconditions
At least two instances of the class BASE.model_version of different instances of the class BASE.model exist, which do not yet have a model assembly relationship with each other.

2.6.1.2 Required operations
- Create.BASE.model_version_relationship_with_transformation
- Create.BASE.transformation
- Connect.BASE.related_model
2.6.1.3 End state and post condition

A new model assembly relationship between two instances of the class BASE.model is available. The sub model is positioned within the frame of the next higher model.

2.6.1.4 Notes and remarks

The instance of the class BASE.transformation does only define the positioning of the submodel frame within the frame of the next higher model. The instance of the class transformation does not define the topological connection between two instances of the class model. More else, SimPDM does not enforce the definition of a model topology or the topological connection on the data management level. Alternatively, the model topology and topological connection between models might also be defined in external model files, which are managed by an instance of the class BASE.document and instances of the class BASE.reference_to_file.

The actual coefficients of the transformation are defined by instances of the class BASE.parameter assigned to the instance of the class BASE.transformation. See chapter 2.9 for the functionality of parameter association management.

2.6.2 Delete existing model assembly relationship

2.6.2.1 Start state and preconditions

An instance of the class BASE.model_version_relationship_with_transformation as a definition of a model assembly hierarchy definition to be deleted does exist. The next higher instance of the class BASE.model_version is in a state allowing it to be changed, i.e., it is not used by an instance of the class BASE.analysis_version which does not allow changes.

2.6.2.2 Required operations

- Disconnect.BASE.transformation
- Delete.BASE.transformation
- Disconnect.BASE.related_model
- Disconnect.BASE.relating_model
- Delete.BASE.model_version_relationship_with_transformation

2.6.2.3 End state and post condition

The relationship between the two instances of the class BASE.model_version is entirely deleted, including the instance of the class BASE.transformation. The two instances of the class BASE.model_version are still available. Deletion of BASE.parameter is subject of another use case (see chapter 2.10).

2.6.2.4 Notes and remarks

The deletion of a model assembly relationship at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.7 Functionality of use case Model Configuration Management

See SimPDM recommendation for a general description of the use case Model Configuration Management. The use case Model Configuration Management is not a major use case within SimPDM (though it is a very important topic within product development and overall product data management). Therefore this use case
includes functionalities for the management of an abstract product structure (chapters 2.7.1 to 2.7.10) as well as for the management and procedure of a configuration itself (chapters 2.7.11 to 2.7.18).

2.7.1 Create abstract product

2.7.1.1 Start state and preconditions
No start state and precondition to be defined.

2.7.1.2 Required operations
- Create.CONF.product

2.7.1.3 End state and post condition
A new CONF.product is available to support the abstract product structure.

2.7.1.4 Notes and remarks
Actually, the creation of a CONF.product is not a daily business, but rather an administrative task.

2.7.2 Delete abstract product

2.7.2.1 Start state and preconditions
A CONF.product does exist and is in a state allowing it to be deleted, i.e. it has no CONF.product_component assigned. The CONF.product might have an associated BASE.administration which is also subject to be deleted then (see chapter 2.1).

2.7.2.2 Required operations
- Delete.CONF.product

2.7.2.3 End state and post condition
The CONF.product is no longer available within the data management system.

2.7.2.4 Notes and remarks
Actually, the deletion of a CONF.product is not a daily business, but rather an administrative task.

The deletion of CONF.product at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.7.3 Create product component

2.7.3.1 Start state and preconditions
A CONF.product does exist and is in a stage allowing changes.

2.7.3.2 Required operations
- Create.CONF.product_component
- Connect.CONF.product_component

2.7.3.3 End state and post condition
A new product component CONF.product_component is available and connected to the abstract product CONF.product.

2.7.3.4 Notes and remarks
Actually, the creation of a CONF.product_component is not a daily business, but rather an administrative task.
In case of creating a CONF.product_component hierarchy, the assignment of the CONF.product_component to a CONF.product is needless.

2.7.4 Delete product component

2.7.4.1 Start state and preconditions

A CONF.product_component does exist and is in a state allowing it to be deleted, i.e. it has no CONF.product_component and/or CONF.solution assigned. The CONF.product_component might have an associated BASE.administration which is also subject to be deleted then (see chapter 2.1).

2.7.4.2 Required operations

- Disconnect.CONF.product_product_component
- Delete.CONF.product_component

2.7.4.3 End state and post condition

The CONF.product_component is no longer available within the data management system.

2.7.4.4 Notes and remarks

Actually, the deletion of a CONF.product_component is not a daily business, but rather an administrative task.

The deletion of CONF.product_component at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.7.5 Create component hierarchy

2.7.5.1 Start state and preconditions

Two CONF.product_component exist and are in a stage allowing changes. One of these CONF.product_component is not assigned to a CONF.product.

2.7.5.2 Required operations

- Connect.CONF.product_component_hierarchy

2.7.5.3 End state and post condition

A new product component hierarchy is available within the data management system.

2.7.5.4 Notes and remarks

Actually, the creation of component hierarchies is not a daily business, but rather an administrative task.

2.7.6 Delete component hierarchy

2.7.6.1 Start state and preconditions

Two CONF.product_component which are related exist and are in a state allowing them to be deleted, i.e. the lower CONF.product_component has no further CONF.product_component and/or CONF.solution assigned. The lower CONF.product_component might have associated BASE.administration which is also subject to be deleted then (see chapter 2.1).

2.7.6.2 Required operations

- Disconnect.CONF.product_component_hierarchy
2.7.6.3  End state and post condition

The product component hierarchy is no longer available within the data management system.

2.7.6.4  Notes and remarks

Actually, the deletion of a component hierarchy is not a daily business, but rather an administrative task.

The deletion of a component hierarchy at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.7.7  Create solution

2.7.7.1  Start state and preconditions

A CONF.product_component does exist and is in a stage allowing changes.

2.7.7.2  Required operations

- Create.CONF.solution
- Connect.CONF.product_component_solution

2.7.7.3  End state and post condition

A new solution CONF.solution is available and connected to the abstract product component CONF.product_component.

2.7.7.4  Notes and remarks

Actually, the creation of a CONF.solution is not a daily business, but rather an administrative task.

2.7.8  Delete solution

2.7.8.1  Start state and preconditions

A CONF.solution does exist and is in a state allowing it to be deleted, i.e. it has no BASE.model assigned and/or is not used by a CONF.configuration. The CONF.solution might have an associated BASE.administration which is also subject to be deleted then (see chapter 2.1).

2.7.8.2  Required operations

- Disconnect.CONF.product_component_solution
- Delete.CONF.product_solution

2.7.8.3  End state and post condition

The CONF.solution is no longer available within the data management system.

2.7.8.4  Notes and remarks

Actually, the deletion of a CONF.solution is not a daily business, but rather an administrative task.

The deletion of CONF.solution at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.7.9  Assign model to solution

---

Copyright: VDA
2.7.9.1 Start state and preconditions
A BASE.model does exist. A CONF.solution does exist and both are in a stage allowing changes.

2.7.9.2 Required operations
- Connect.CONF.solution_model

2.7.9.3 End state and post condition
A CONF.solution is related to a BASE.model within the data management system.

2.7.9.4 Notes and remarks
No further notes and remarks available.

2.7.10 Delete model assignment from solution
2.7.10.1 Start state and preconditions
A CONF.solution and a BASE.model which are related exist and are in a state allowing them to be disconnected.

2.7.10.2 Required operations
- Disconnect.CONF.solution_model

2.7.10.3 End state and post condition
A BASE.model is no longer included into a CONF.solution. Within the data management system both objects still exist on their own.

2.7.10.4 Notes and remarks
The deletion of the relation CONF.solution_model at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.7.11 Create configuration within SimPDM
2.7.11.1 Start state and preconditions
One or more CONF.solution exist.

2.7.11.2 Required operations
- Create.CONF.configuration
- Connect.CONF.configuration_solution

And, if the configuration is hierarchic, additionally
- Connect.CONF.configuration_hierarchy

2.7.11.3 End state and post condition
A CONF.configuration with assigned CONF.solution is available within the data management system.

2.7.11.4 Notes and remarks
No further notes and remarks available.

2.7.12 Assign configuration to BOM
2.7.12.1 Start state and preconditions
A CAD.bill_of_material exists. A CONF.configuration exists and is in a state allowing change.
2.7.12.2 Required operations
- Connect.CONF.configuration_bom

2.7.12.3 End state and post condition
A CONF.configuration is assigned to a CAD.bill_of_material.

2.7.12.4 Notes and remarks
No further notes and remarks available.

2.7.13 Delete configuration assignment from BOM
2.7.13.1 Start state and preconditions
A CAD.bill_of_material with an assigned CONF.configuration exists and is in a state allowing change.

2.7.13.2 Required operations
- Disconnect.CONF.configuration_bom

2.7.13.3 End state and post condition
A CONF.configuration is no longer assigned to a CAD.bill_of_material.

2.7.13.4 Notes and remarks
The deletion of the relation CONF.configuration_bom at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.7.14 Add solution to configuration
2.7.14.1 Start state and preconditions
A CONF.solution exists. A CONF.configuration exists and is in a state allowing change.

2.7.14.2 Required operations
- Connect.CONF.configuration_solution

2.7.14.3 End state and post condition
An additional CONF.solution is assigned to a CONF.configuration.

2.7.14.4 Notes and remarks
No further notes and remarks available.

2.7.15 Delete solution from configuration
2.7.15.1 Start state and preconditions
A CONF.configuration with assigned CONF.solution exists and is in a state allowing change.

2.7.15.2 Required operations
- Disconnect.CONF.configuration_solution

2.7.15.3 End state and post condition
A CONF.solution is no longer assigned to a CONF.configuration.

2.7.15.4 Notes and remarks
The deletion of the relation CONF.configuration_solution at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.7.16 Delete configuration

2.7.16.1 Start state and preconditions

A CONF.configuration does exist and is in a state allowing it to be deleted, i.e. it has no CAD.bill_of_material and/or CONF.solution and/or BASE.analysis_version assigned. The CONF.configuration might have an associated BASE.administration which is also subject to be deleted then (see chapter 2.1).

2.7.16.2 Required operations

- Delete.CONF.configuration

And, if the configuration is hierarchic, additionally:

- Disconnect.CONF.configuration_hierarchy

2.7.16.3 End state and post condition

The CONF.configuration is no longer available within the data management system.

2.7.16.4 Notes and remarks

The deletion of CONF.configuration at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.7.17 Assign configuration to analysis

2.7.17.1 Start state and preconditions

A BASE.analysis_version exists. A CONF.configuration exists and both are in a state allowing change.

2.7.17.2 Required operations

- Connect.CONF.configuration_analysis_version

2.7.17.3 End state and post condition

A CONF.configuration is assigned to a BASE.analysis_version.

2.7.17.4 Notes and remarks

No further notes and remarks available.

2.7.18 Delete configuration assignment from analysis

2.7.18.1 Start state and preconditions

A BASE.model_version with an assigned CONF.configuration exists and is in a state allowing change.

2.7.18.2 Required operations

- Disconnect.CONF.configuration_analysis_version

2.7.18.3 End state and post condition

A CONF.configuration is no longer assigned to a BASE.model_version.

2.7.18.4 Notes and remarks

The deletion of the relation CONF.configuration_analysis_version at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.
2.8 Functionality of use case Model Definition Management

See SimPDM recommendation for a general description of the use case Model Definition Management.

2.8.1 Create new model

2.8.1.1 Start state and preconditions

No specific precondition to be defined.

2.8.1.2 Required operations

- Create.BASE.model
- Create.BASE.model_version
- Connect.BASE.model_version

2.8.1.3 End state and post condition

A new instance of the class BASE.model and an associated initial instance of the class BASE.model_version for the new instance of the class BASE.model are available.

2.8.1.4 Notes and remarks

A new instance of the class BASE.model can be defined without initially being embedded within a specific analysis scope. The instances of the classes BASE.model and BASE.model_version are only defined as nodes within the simulation data management system by this function without any topological content defined. Topological content may be defined and associated either by applying the use cases Topological Element Definition Management, Topological Structure Definition Management and Property Definition Management or by applying the use case Document Management. Optionally the use case Administration Management might be applied for BASE.model and BASE.model_version.

2.8.2 Delete existing model

2.8.2.1 Start state and preconditions

The instance of the class BASE.model to be deleted has not more than one associated instances of the class BASE.model_version, the initial instance of the class BASE.model_version for the instance of the class BASE.model to be deleted. The initial instance of the class BASE.model_version has no relationships with other instances of the class BASE.model_version. The associated initial instance of the class BASE.model_version has no associated topological description. The instance of the class BASE.model and the initial instance of the class BASE.model_version each might have associated instances of the class BASE.administration.

2.8.2.2 Required operations

- Disconnect.BASE.model_version
- Delete.BASE.model_version
- Delete.BASE.model

And if the model or the related model version have associated administrative data…

- Disconnect.BASE.model_administration
- Disconnect.BASE.model_version_administration
- Delete.BASE.administration

2.8.2.3 End state and post condition

Copyright: VDA
The instance of the class BASE.model and its associated initial instance of the class BASE.mode_version are no longer available within the data management system.

2.8.2.4 Notes and remarks

The deletion of instances of the class BASE.model at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.8.3 Create new model version

2.8.3.1 Start state and preconditions

An instance of the class BASE.model with at least one instance of the class BASE.model_version, the initial instance of the class BASE.model_version for the instance of the class BASE.model, exists.

2.8.3.2 Required operations

- Create.BASE.model_version
- Connect.BASE.model_version
- Connect.BASE.model_version_relationship

2.8.3.3 End state and post condition

A new instance of the class BASE.model_version of an existing instance of the class BASE.model is available within the data management system. The new instance of the class BASE.model_version is connected with another instance of the class BASE.model_version of the same instance of the class BASE.model in a predecessor–successor relationship.

2.8.3.4 Notes and remarks

The relationship BASE.model_version_relationship can be used to connect two instances of the class BASE.model_version. If these two instances of the class BASE.model_version belong to the same instance of the class BASE.version, the relationship BASE.model_version_relationship defines a predecessor–successor relationship of the instances of the class BASE.model_version. Optionally the use case Administration Management might be applied for the instance of the class BASE.model_version.

2.8.4 Delete existing model version

2.8.4.1 Start state and preconditions

The instance of the class BASE.model_version to be deleted has no relationships with other instances of the class BASE.model_version, no related succeeding instance of the class BASE.model_version, no associated topological description and no associated instance of the class BASE.document. The instance of the class BASE.model_version might have an associated instance of the class BASE.administration.

2.8.4.2 Required operations

- Disconnect.BASE.model_version_relationship
- Disconnect.BASE.model_version
- Delete.BASE.model_version

And if the model version has an associated administrative data...

- Disconnect.BASE.model_version_administration

Copyright: VDA
2.8.4.3 End state and post condition
The instance of the class BASE.mode_version is no longer available within the data management system.

2.8.4.4 Notes and remarks
The deletion of instances of the class BASE.model_version at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.8.5 Connect model version to analysis version
2.8.5.1 Start state and preconditions
One instance of the class BASE.analysis_version and one instance of the class BASE.model_version do exist.

2.8.5.2 Required operations
- Connect.BASE.analysis_version_model_version

2.8.5.3 End state and post condition
An instance of the class BASE.model_version is assigned to an instance of the class BASE.analysis_version.

2.8.5.4 Notes and remarks
No further notes and remarks available

2.8.6 Disconnect model version from analysis version
2.8.6.1 Start state and preconditions
A relationship between an instance of the class BASE.analysis_version and an instance of the class BASE.model_version do exist.

2.8.6.2 Required operations
- Disconnect.BASE.analysis_version_model_version

2.8.6.3 End state and post condition
The instance of the class BASE.model_version is no longer connected with an instance of the class BASE.analysis_version.

2.8.6.4 Notes and remarks
An instance of the class BASE.model_version can be assigned to more than one instances of the class BASE.analysis_version. In this case the deletion on one relationship has no effects on other relationships between an instance of the class BASE.model_version to other instances of the class BASE.analysis_version.

2.8.7 Define new model version dependency
2.8.7.1 Start state and preconditions
Two instances of the class BASE.model_version associated with different instances of the class BASE.model do exist.

2.8.7.2 Required operations
- Connect.BASE.model_version_relationship

2.8.7.3 End state and post condition

Two instances of the class BASE.model_version of different instances of the class BASE.model are associated with each other in a general manner.

2.8.7.4 Notes and remarks

The two different instances of the class BASE.model_version associated with different instances of the class BASE.model_version are generally associated with each other. The data model defined by the version 2.0 of the recommendation does not provide different kind of model version relationships explicitly. A relationship between instances of the class BASE.model_version associated with different instances of the class BASE.model are not supposed to represent a preceding – succeeding relationship of instances of the class BASE.model_version.

2.8.8 Delete existing model version dependency

2.8.8.1 Start state and preconditions

A relationship between two instances of the class BASE.model_version associated with different instances of the class BASE.model does exist.

2.8.8.2 Required operations

- Delete.BASE.model_version_relationship

2.8.8.3 End state and post condition

The dependency between two instances of the class BASE.model_version associated with different with different instances of the class BASE.model is no longer available.

2.8.8.4 Notes and remarks

The deletion of the dependency between two instances of the class BASE.model_version associated with different instances of the class BASE.model has not any effect on the availability and usability of the formally connected instances of the class BASE.model_version.

2.9 Functionality of use case Output Specification Management

See SimPDM recommendation for a general description of the use case Output Specification Management.

2.9.1 Create new Output Specification

2.9.1.1 Start state and preconditions

A BASE.analysis_version exists and is in state allowing change.

2.9.1.2 Required operations

- Create.BASE.output_specification
- Connect.BASE.output_specification

And if required…

- Create.BASE.output_request
- Connect.BASE.output_request

2.9.1.3 End state and post condition

A BASE.output_specification and possibly its BASE.output_request is available within the data management system.

2.9.1.4 Notes and remarks
2.9.2 Delete existing Output Specification

2.9.2.1 Start state and preconditions

A BASE.output_specification exists and is assigned to a BASE.analysis_version which is in a state allowing change. Optionally a BASE.output_request is assigned to the BASE.output_specification and is also in a state allowing change, i.e. no document is assigned.

2.9.2.2 Required operations

- Delete.BASE.output_specification
- Disconnect.BASE.output_specification

And if necessary…

- Delete.BASE.output_request
- Disconnect.BASE.output_request

2.9.2.3 End state and post condition

A BASE.output_specification, its BASE.output_request is no longer available and no longer assigned to a BASE.analysis_version within the data management system.

2.9.2.4 Notes and remarks

The deletion of the BASE.output_specification, its BASE.output_request at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.9.3 Define new group of interesting elements

2.9.3.1 Start state and preconditions

A BASE.output_specification exists and is in a state allowing change. Furthermore TOPO.model_element or one of its subtypes does exist.

2.9.3.2 Required operations

- Create.BASE.group_of_elements
- Connect.BASE.output_group
- Connect.TOPO.element_of_interest

2.9.3.3 End state and post condition

A BASE.group_of_elements is available within the data management system, is assigned to a BASE.output_specification and has assigned some TOPO.model_element or one of its subtypes.

2.9.3.4 Notes and remarks

No further notes and remarks available.

2.9.4 Delete group definition of interesting elements

2.9.4.1 Start state and preconditions

A BASE.group_of_elements exists and has assigned some TOPO.model_element or one of its subtypes. BASE.group_of_elements is assigned to a BASE.output_specification which is in a state allowing change.

2.9.4.2 Required operations

- Delete.BASE.group_of_elements

Copyright: VDA
2.9.4.3 End state and post condition

A BASE.group_of_elements and its assignments are no longer available within the data management system.

2.9.4.4 Notes and remarks

The TOPO.model_element or one of its subtypes still exists within the data management system. Deletion is subject of another use case (see chapter 2.15).

The deletion of the BASE.group_of_elements at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.10 Functionality of use case Parameter Association Management

See SimPDM recommendation for a general description of the use case Parameter Association Management.

2.10.1 Create new parameter

2.10.1.1 Start state and preconditions

One instance of one of the following classes does exist.

- BASE.model_version
- BASE.transformation
- CAD.connection_element_definition
- CAD.connection_type
- PROP.parameter_specified_property
- SETT.setting
- SYNC.parameter_synchronisation
- TOPO.model_element

2.10.1.2 Required operations

- Create.BASE.parameter (one of the subtypes respectively)

And, depending on the instance of a class, for which the new parameter is to be defined, one of …

- Connect.BASE.model_parameter
- Connect.BASE.transformation_parameter
- Connect.CAD.connection_element_definition_parameter
- Connect.CAD.connection_type_parameter
- Connect.PROP.property_parameter
- Connect.SETT.setting_parameter
- Connect.SYNC.parameter_synchronisation_parameter
- Connect.TOPO.element_parameter

2.10.1.3 End state and post condition
A new instance of the class BASE.parameter (one of the subtypes respectively) exists and is assigned to an instance of one of the classes

- BASE.model_version
- BASE.transformation
- CAD.connection_element_definition
- CAD.connection_type
- PROP.parameter_specified_property
- SETT.setting
- SYNC.parameter_synchronisation
- TOPO.model_element

### 2.10.1.4 Notes and remarks

An instance of the class BASE.parameter (one of the subtypes respectively) usually is used once in the context of a certain object. In case of process linking and networking an instance of the class BASE.parameter (one of the subtypes respectively) can be shared by more than one object.

### 2.10.2 Delete existing parameter

#### 2.10.2.1 Start state and preconditions

An instance of the class BASE.parameter (one of the subtypes respectively) to be deleted does exist.

#### 2.10.2.2 Required operations

- Delete.BASE.parameter (one of the subtypes respectively)

And, depending of the instance of a class for which the parameter is defined, one of ...

- Disconnect.BASE.model_parameter
- Disconnect.BASE.transformation_parameter
- Disconnect.CAD.connection_element_definition_parameter
- Disconnect.CAD.connection_type_parameter
- Disconnect.PROP.property_parameter
- Disconnect.SETT.setting_parameter
- Disconnect.SYNC.parameter_synchronisation_parameter
- Disconnect.TOPO.element_parameter

#### 2.10.2.3 End state and post condition

The instance of the class BASE.parameter (one of the subtypes respectively) is no longer available.

#### 2.10.2.4 Notes and remarks

The deletion of an instance of the class BASE.parameter (one of the subtypes respectively) at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

### 2.10.3 Add a parameter unit

Copyright: VDA
2.10.3.1 Start state and preconditions
A BASE.parameter (one of the subtypes respectively) exists and is in state allowing change.

2.10.3.2 Required operations
- Create.BASE.unit
- Connect.BASE.parameter_unit

2.10.3.3 End state and post condition
An instance of the class BASE.unit is available within the data management system and is assigned to a BASE.parameter (one of the subtypes respectively).

2.10.3.4 Notes and remarks
No further notes and remarks available.

2.10.4 Delete a parameter unit

2.10.4.1 Start state and preconditions
An instance of the class BASE.unit to be deleted does exist and is in a stage allowing change.

2.10.4.2 Required operations
- Delete.BASE.unit
- Disconnect.BASE.parameter_unit

2.10.4.3 End state and post condition
The instance of the class BASE.unit is no longer assigned to a BASE.parameter (one of the subtypes respectively) and is no longer available on the database. The instance of the class BASE.parameter (one of the subtypes respectively) is still available, for deletion see according chapter above.

2.10.4.4 Notes and remarks
The deletion of an instance of the class BASE.unit at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.11 Functionality of use case PDM Information Derivation Management
See SimPDM recommendation for a general description of the use case Geometry Derivation Management.

2.11.1 Create new CAD-PDM-Information

2.11.1.1 Start state and preconditions
No start state and precondition to be defined.

2.11.1.2 Required operations
- Create.CAD.cadpdm_information (or one of the subtypes)

2.11.1.3 End state and post condition
A new CAD.cadpdm_information definition is available within the data management system.

2.11.1.4 Notes and remarks
Use of CAD.cadpdm_information is meaningful only in combination with other mechanism, functionalities of use case Geometry Derivation Management like the ones described in the following in this chapter.

2.11.2 Delete CAD-PDM-Information

2.11.2.1 Start state and preconditions
A CAD.cadpdm_information to be deleted does exist and is in a stage allowing change, i.e. it has no further objects like CONF.configuration, BASE.document, etc. assigned. The CAD.cadpdm_information might have an associated BASE.administration which is also subject to be deleted then (see chapter 2.1).

2.11.2.2 Required operations
- Delete.CAD.cadpdm_information (or one of the subtypes)

2.11.2.3 End state and post condition
The CAD.cadpdm_information definition is no longer available within the data management system

2.11.2.4 Notes and remarks
The deletion of a CAD.cadpdm_information object at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.11.3 Derive CAD-PDM-Information

2.11.3.1 Start state and preconditions
A (specialized) CAD.cadpdm_information exists.

2.11.3.2 Required operations
- Create.CAD.cadpdm_information (one of its sub types)
And, depending on the specialization type of CAD.cadpdm_information, one of...
  - Connect.CAD.derived_shape
  - Connect.CAD.derived_cel
  - Connect.CAD.derived_bom

2.11.3.3 End state and post condition
A derived and (specialized) CAD.cadpdm_information is available and assigned to its origin within the data management system.

2.11.3.4 Notes and remarks
Derivation is only possible within a type of specialization, not cross-specialization.

2.11.4 Delete CAD-PDM-Information derivation

2.11.4.1 Start state and preconditions
A (specialized) CAD.cadpdm_information and its origin exist and are in state allowing change.

2.11.4.2 Required operations
- Delete.CAD.cadpdm_information (one of its sub types)
And, depending on the specialization type of CAD.cadpdm_information, one of...
  - Disconnect.CAD.derived_shape
2.11.4.3 End state and post condition

A derived and (specialized) CAD.cadpdm_information is no longer available within the data management system.

2.11.4.4 Notes and remarks

The deletion of derived CAD.cadpdm_information at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.11.5 Assign BOM to analysis

2.11.5.1 Start state and preconditions

A CAD.bill_of_material does exist. A BASE.analysis_version exists and is in a stage allowing change.

2.11.5.2 Required operations

- Connect.CAD.analysis_bom

2.11.5.3 End state and post condition

A CAD.bill_of_material is assigned to a BASE.analysis_version within the data management system.

2.11.5.4 Notes and remarks

No further notes and remarks available.

2.11.6 Delete BOM assignment from analysis

2.11.6.1 Start state and preconditions

A CAD.bill_of_material does exist and is assigned to a BASE.analysis_version which is in a state allowing change.

2.11.6.2 Required operations

- Disconnect.CAD.analysis_bom

2.11.6.3 End state and post condition

A CAD.bill_of_material is no longer assigned to a BASE.analysis_version within the data management system.

2.11.6.4 Notes and remarks

The deletion of the relation CAD.analysis_bom at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.11.7 Create BOM parts

2.11.7.1 Start state and preconditions

A CAD.bill_of_material exists and is in a state allowing change. Additionally existence of a PROP.property_set_version can be a precondition.

2.11.7.2 Required operations

- Create.CAD.part
- Connect.CAD.bom_part
And, if necessary and applicable
  - Connect.CAD.material_of_part

2.11.7.3 End state and post condition
A CAD.part is available and assigned to a CAD.bill_of_material within the data management system. Additionally the CAD.part can be assigned to a PROP.property_set_version.

2.11.7.4 Notes and remarks
The PROP.property_set_version already exists within the data management system. Creation is subject of another use case (see chapter 2.13).

2.11.8 Deleting BOM parts
2.11.8.1 Start state and preconditions
A CAD.part exists and is in a state allowing change, i.e. it has no further objects of BASE.model_version assigned.

2.11.8.2 Required operations
  - Delete.CAD.part
  - Disconnect.CAD.bom_part
And, if necessary,
  - Disconnect.CAD.material_of_part

2.11.8.3 End state and post condition
A CAD.part is no longer available to a CAD.bill_of_material within the data management system.

2.11.8.4 Notes and remarks
The PROP.property_set_version still exists within the data management system. Deletion is subject of another use case (see chapter 2.13).

The deletion of a CAD.part at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.11.9 Assign part to model
2.11.9.1 Start state and preconditions
A CAD.part does exist. A BASE.model_version exists and is in a state allowing change.

2.11.9.2 Required operations
  - Connect.CAD.part_model_version

2.11.9.3 End state and post condition
A CAD.part is assigned to a BASE.model_version within the data management system.

2.11.9.4 Notes and remarks
No further notes and remarks available.

2.11.10 Delete part assignment from model
2.11.10.1 Start state and preconditions
A CAD.part does exist and is assigned to a BASE.model_version which is in a state allowing change.

2.11.10.2 Required operations

- Disconnect.CAD.part_model_version

2.11.10.3 End state and post condition

A CAD.part is no longer assigned to a BASE.model_version within the data management system.

2.11.10.4 Notes and remarks

The deletion of the relation CAD.part_model_version at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.11.11 Specify CAE model of shape representation

2.11.11.1 Start state and preconditions

A CAD.shape_representation does exist. A BASE.model_version exists and is in a state allowing change.

2.11.11.2 Required operations

- Connect.CAD.shape_representation_model_version

2.11.11.3 End state and post condition

A CAD.shape_representation is assigned to a BASE.model_version within the data management system.

2.11.11.4 Notes and remarks

Difference between CAD.shape_representation_model_version and CAD.used_shape is on semantic level. CAD.shape_representation_model_version is used to identify a CAE model belonging to a certain shape in a range of derived shapes. This can be used when multiple CAE models base on a range of derived shapes. CAD.used_shape identifies the origin shape used by a CAE model.

2.11.12 Delete CAE model specification from shape representation

2.11.12.1 Start state and preconditions

A CAD.shape_representation does exist and is assigned to a BASE.model_version where both are in a state allowing change.

2.11.12.2 Required operations

- Disconnect.CAD.shape_representation_model_version

2.11.12.3 End state and post condition

A CAD.shape_representation is no longer assigned to a BASE.model_version within the data management system.

2.11.12.4 Notes and remarks

The deletion of the relation CAD.shape_representation_model_version at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.11.13 Assign shape representation to CAE model

2.11.13.1 Start state and preconditions
A CAD.shape_representation exists and is in a state allowing change. A BASE.model_version exists and is in a state allowing change.

2.11.13.2 Required operations

- Connect.CAD.used_shape

2.11.13.3 End state and post condition

A BASE.model_version is assigned to a CAD.shape_representation within the data management system.

2.11.13.4 Notes and remarks

Difference between CAD.shape_representation_model_version and CAD.used_shape is on semantic level. CAD.shape_representation_model_version is used to identify a CAE model belonging to a certain shape in a range of derived shapes. This can be used when multiple CAE models base on a range of derived shapes. CAD.used_shape identifies the origin shape used by a CAE model.

2.11.14 Delete shape representation assignment from CAE model

2.11.14.1 Start state and preconditions

A BASE.model_version does exist and is assigned to a CAD.shape_representation where both are in a state allowing change.

2.11.14.2 Required operations

- Disconnect.CAD.used_shape

2.11.14.3 End state and post condition

A BASE.model_version is no longer assigned to a CAD.shape_representation within the data management system.

2.11.14.4 Notes and remarks

The deletion of the relation CAD.used_shape at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.11.15 Assign connection element list to analysis

2.11.15.1 Start state and preconditions

A CAD.connection_element_list does exist. A BASE.analysis_version exists and is in a state allowing change.

2.11.15.2 Required operations

- Connect.CAD.analysis_cel

2.11.15.3 End state and post condition

A CAD.connection_element_list is assigned to a BASE.analysis_version within the data management system.

2.11.15.4 Notes and remarks

No further notes and remarks available.

2.11.16 Delete connection element list assignment from analysis

2.11.16.1 Start state and preconditions

A CAD.connection_element_list does exist and is assigned to a BASE.analysis_version which is in a state allowing change.
2.11.16.2 Required operations
- Disconnect.CAD.analysis_cel

2.11.16.3 End state and post condition
A CAD.connection_element_list is no longer assigned to a BASE.analysis_version within the data management system.

2.11.16.4 Notes and remarks
The deletion of the relation CAD.analysis_cel at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.11.17 Create connection element

2.11.17.1 Start state and preconditions
A CAD.connection_element_list exists and is in a state allowing change.

2.11.17.2 Required operations
- Create.CAD.connection_element
- Create.CAD.connection_element_definition
- Create.CAD.connection_element_type
- Connect.CAD.cel_elements
- Connect.CAD.connection_definition
- Connect.CAD.connection_type_definition
- Connect.CAD.connection_material

2.11.17.3 End state and post condition
A CAD.connection_element, its CAD.connection_element_definition and CAD.connection_element_type is available and assigned to a CAD.connection_element_list within the data management system.

2.11.17.4 Notes and remarks
The actual parameters of the CAD.connection_element_definition are defined by BASE.parameter assigned to the CAD.connection_element_definition. See chapter 2.10 for the functionality of parameter association management.

The actual parameters and properties of the CAD.connection_element_type are defined by BASE.parameter resp. PROP.property_set_version assigned to the CAD.connection_element_type. See chapter 2.10 resp. chapter 2.13 for the functionality of parameter association management.

2.11.18 Delete connection element

2.11.18.1 Start state and preconditions
A CAD.connection_element, its CAD.connection_element_definition and CAD.connection_element_type exist and they are assigned to a CAD.connection_element_list which is in a state allowing change, i.e. there are no further objects like parameters or properties assigned.

2.11.18.2 Required operations
- Delete.CAD.connection_element
- Delete.CAD.connection_element_definition
2.11.18.3 End state and post condition
A CAD.connection_element, its CAD.connection_element_definition and CAD.connection_element_type is no longer available and no longer assigned to a CAD.connection_element_list within the data management system.

2.11.18.4 Notes and remarks
The actual parameters of the CAD.connection_element_definition are defined by BASE.parameter assigned to the CAD.connection_element_definition. See chapter 2.10 for the functionality of parameter association management.

The actual parameters and properties of the CAD.connection_element_type are defined by BASE.parameter resp. PROP.property_set_version assigned to the CAD.connection_element_type. See chapter 2.10 resp. chapter 2.13 for the functionality of parameter association management.

The deletion of the CAD.connection_element, its CAD.connection_element_definition and CAD.connection_element_type at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.12 Functionality of use case Post-processing Management
See SimPDM recommendation for a general description of the use case Post-processing Management.

2.12.1 Create analysis result
2.12.1.1 Start state and preconditions
A BASE.analysis_version exists and is in a state allowing no more changes.

2.12.1.2 Required operations
- Create.BASE.analysis_result
- Connect.BASE.analysis_run_specific_analysis_result

2.12.1.3 End state and post condition
A BASE_analysis_result is available and assigned to a BASE.analysis_version within the data management system.

2.12.1.4 Notes and remarks
It has to be stressed that there must be explicit traceability between a BASE.analysis_result and its source, a BASE.analysis_version.

2.12.2 Delete analysis result
2.12.2.1 Start state and preconditions
A BASE.analysis_result exists and is in a state allowing change. There are no further BASE.documents or BASE.administration assigned to the BASE.analysis_result (for disconnecting them see the according chapters 2.1 and 2.4).
2.12.2 Required operations
- Delete.BASE.analysis_result
- Disconnect.BASE.analysis_run_specific_analysis_result

2.12.2.3 End state and post condition
The BASE.analysis_result and its assignment to a BASE.analysis_version are no longer available within the data management system.

2.12.2.4 Notes and remarks
It has to be stressed that there must be explicit traceability between a BASE.analysis_result and its source, a BASE.analysis_result. This is even more important in cases of actual result existence (as BASE.analysis_result is just a folder). In this case deleting a BASE.analysis_result means losing the link between the actual results and their BASE.analysis_version. Deleting a BASE.analysis_result might be suitable in cases of non-converging results.

The deletion of BASE.analysis_result at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.12.3 Create a computation output (result first order)

2.12.3.1 Start state and preconditions
A BASE.analysis_result exists and is in a stage allowing change.

2.12.3.2 Required operations
- Create.BASE.computation_output
- Connect.BASE.result_first_order

2.12.3.3 End state and post condition
A result first order is represented as an instance of BASE.computation_output and is assigned to a specific BASE.analysis_result within the data management system.

2.12.3.4 Notes and remarks
It has to be stressed that there must be explicit traceability between every result and its source, a BASE.analysis_version.

2.12.4 Delete a computation output (result first order)

2.12.4.1 Start state and preconditions
A BASE.computation_result exists and is in a stage allowing change.

2.12.4.2 Required operations
- Delete.BASE.computation_output
- Disconnect.BASE.result_first_order

2.12.4.3 End state and post condition
A result first order represented as an instance of BASE.computation_output is no longer available within the data management system.

2.12.4.4 Notes and remarks
It has to be stressed that there must be explicit traceability between every result and its source, a BASE.analysis_version.
2.12.5 Create a key result (result second order)

2.12.5.1 Start state and preconditions

A BASE.analysis_result and a BASE.computation_output or an already derived BASE.key_result exists.

2.12.5.2 Required operations

- Create.BASE.key_result
- Create.BASE.postprocessing_input (one of its sub types)
- Connect.BASE:result_second_order
- Connect.BASE.base_for_key_result

And, if using a post processing template

- Connect.BASE.template creates_key_result

2.12.5.3 End state and post condition

A result second order is represented as an instance of BASE.key_result and is assigned to a specific BASE.analysis_result. Furthermore its derivation chain is represented.

2.12.5.4 Notes and remarks

It has to be stressed that there must be explicit traceability between every result and its sources, BASE.analysis_version and BASE.postprocessing_input (one of its sub types).

2.12.6 Delete a key result (result second order)

2.12.6.1 Start state and preconditions

A BASE.key_result exists and is in a state allowing change.

2.12.6.2 Required operations

- Delete.BASE.key_result
- Disconnect.BASE.result_second_order
- Disconnect.BASE.base_for_key_result

And, if using a post processing template

- Disconnect.BASE.template creates_key_result

2.12.6.3 End state and post condition

A result second order represented as an instance of BASE.key_result is no longer available within the data management system. Its post processing input still exists.

2.12.6.4 Notes and remarks

It has to be stressed that there must be explicit traceability between every result and its sources, BASE.analysis_version and BASE.postprocessing_input (one of its sub types).

The deletion of BASE.key_result at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

Copyright: VDA
2.12.7 Create a report (result third order)

2.12.7.1 Start state and preconditions

A BASE.analysis_result exists. Depending on the sourcing for the report, a BASE.key_result and/or BASE.computation_result exist.

2.12.7.2 Required operations

- Create.BASE.report
- Connect.BASE.result_third_order

And, depending of the report sourcing, at least one of ...

- Connect.BASE.base_for_report
- Connect.BASE.base_for_report2

2.12.7.3 End state and post condition

A result third order is represented as an instance of BASE.report and is assigned to a specific BASE.analysis_result. Furthermore its derivation chain is represented.

2.12.7.4 Notes and remarks

A BASE.report can be derived from results first and second order. A BASE.report can belong to different BASE.analysis_result, as product performance conclusion usually needs more than a single simulation.

It has to be stressed that there must be explicit traceability between every result and its sources, BASE.analysis_version, BASE.computation_output and BASE.key_result.

2.12.8 Delete a report (result third order)

2.12.8.1 Start state and preconditions

A BASE.report exists and is in a state allowing change.

2.12.8.2 Required operations

- Delete.BASE.report
- Disconnect.BASE.result_third_order

And, depending of the report sourcing, at least one of ...

- Disconnect.BASE.base_for_report
- Disconnect.BASE.base_for_report2

2.12.8.3 End state and post condition

A result third order represented as an instance of BASE.report is no longer available within the data management system. Its sourcing objects like instances of BASE.key_result or BASE.computation_output input still exist.

2.12.8.4 Notes and remarks

It has to be stressed that there must be explicit traceability between every result and its sources, BASE.analysis_version, BASE.computation_output and BASE.key_result.

The deletion of BASE.report at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.
2.12.9 Create a detailed post processing object

2.12.9.1 Start state and preconditions

A BASE.key_result or a BASE.template exists and is in a state allowing change.

2.12.9.2 Required operations

- Create.POST.detailed_postprocessing (one of its sub types)
  Depending on the type of result which is detailed, one of ...
  - Connect.POST.detailing_key_result
  - Connect.POST.detailing_report

- Connect.POST.detailing_key_result
- Connect.POST.detailing_report

Depending on the POST.detailed_postprocessing sub type, one of ...
- Connect.POST.model_version_from_result
- Connect.POST.property_set_version_from_result
- Connect.POST.cadpdm_information_from_result

2.12.9.3 End state and post condition

A detailed post processing objects as an instance of the class POST.detailed_postprocessing is available within the data management system and assigned to an instance of BASE.key_result or BASE.template.

2.12.9.4 Notes and remarks

Technical definition of the detailed result as an instance of the classes BASE.model_version, CAD.cadpdm_information or PROP.property_set_version is not subject of this use case. Therefore see chapters 2.8, 2.13 and 2.11.

2.12.10 Delete a detailed post processing object

2.12.10.1 Start state and preconditions

A POST.detailed_postprocessing (one of its sub types) exists and is in a state allowing change.

2.12.10.2 Required operations

- Delete.POST.detailed_postprocessing (one of its sub types)
  Depending on the type of result which is detailed, one of ...
  - Disconnect.POST.detailing_key_result
  - Disconnect.POST.detailing_report

- Disconnect.POST.detailing_key_result
- Disconnect.POST.detailing_report

Depending on the POST.detailed_postprocessing sub type, one of ...
- Disconnect.POST.model_version_from_result
- Disconnect.POST.property_set_version_from_result
- Disconnect.POST.cadpdm_information_from_result

2.12.10.3 End state and post condition

A detailed post processing objects as an instance of the class POST.detailed_postprocessing is no lonager available within the data management system.

The basing instances of the classes BASE.key_result or BASE_report as well as the technical definition of BASE.model_version, CAD.cadpdm_information or PROP.property_set_version are still available on the data management system.
2.12.10.4 Notes and remarks

Technical definition and deletion of the detailed result as an instance of the classes BASE.model_version, CAD.cadpdm_information or PROP.property_set_version is not subject of this use case. Therefore see chapters 2.8, 2.13 and 2.11.

The deletion of POST.detailed_postprocessing at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.12.11 Create a template

2.12.11.1 Start state and preconditions

No start state and precondition to be defined.

2.12.11.2 Required operations

- Create.BASE.template

2.12.11.3 End state and post condition

A BASE.template is available within the data management system.

2.12.11.4 Notes and remarks

No further notes and remarks available.

2.12.12 Delete a template

2.12.12.1 Start state and preconditions

A BASE.template exists and is in a state allowing change. There are no further BASE.documents, BASE.key_result or BASE.administration etc. assigned to the BASE.template (for disconnecting them see the according chapters).

2.12.12.2 Required operations

- Delete.BASE.template

2.12.12.3 End state and post condition

A BASE.template is no longer available within the data management system.

2.12.12.4 Notes and remarks

The deletion of BASE.template at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.13 Functionality of use case Property Definition Management

See SimPDM recommendation for a general description of the use case Property Definition Management.

2.13.1 Create new property set

2.13.1.1 Start state and preconditions

No start state and precondition to be defined.

2.13.1.2 Required operations

- Create.PROP.property_set (or its sub type)

2.13.1.3 End state and post condition

A new instance of the class PROP.property_set (or its sub type) is available within the data management system.
2.13.1.4 Notes and remarks
No further notes and remarks available.

2.13.2 Delete a property set
2.13.2.1 Start state and preconditions
An instance of the class PROP.property_set exists and is in a state allowing change. This implies there are no further objects like instances of the class PROP.property_set_version, assigned. See the according chapters for deletion.

2.13.2.2 Required operations
- Delete.PROP.property_set (or its sub type)

2.13.2.3 End state and post condition
The instance of the class PROP.property_set is no longer available within the data management system.

2.13.2.4 Notes and remarks
The deletion of PROP.property_set at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.13.3 Create new property set version
2.13.3.1 Start state and preconditions
An instance of the class PROP.property_set does exist.

2.13.3.2 Required operations
- Create.PROP.property_set_version
- Connect.PROP.property_set_version
And optionally
- Connect.PROP.property_set_version_relationship

2.13.3.3 End state and post condition
A new instance of the class PROP.property_set_version does exist, and is optionally associated with the preceding instance of the class PROP.property_set_version in a preceding – succeeding relationship.

2.13.3.4 Notes and remarks
Optionally the use case Administration Management might be applied for the new instance of the class PROP.property_set_version.

2.13.4 Delete existing property set version
2.13.4.1 Start state and preconditions
An instance of the class PROP.property_set_version does exist. The instance of the class PROP.property_set_version belongs to an instance of the class PROP.property_set and is not used in any relationships. The instance of the class PROP.property_set_version might have associated administrative data.

2.13.4.2 Required operations
- Delete.PROP.property_set_version
- Disconnect.PROP.property_set_version
And optionally
Copyright: VDA
2.13.4.3 End state and post condition
The instance of the class PROP.property_set_version is no longer available within the data management system.

2.13.4.4 Notes and remarks
The deletion of instances of the class PROP.property_set_version at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.13.5 Create new property set version relationship
2.13.5.1 Start state and preconditions
Two instances of the class PROP.property_set_version do exist.

2.13.5.2 Required operations
- Connect.PROP.property_set_version_relationship

2.13.5.3 End state and post condition
Two instances of the class PROP.property_set_version are associated with each other in a general manner.

2.13.5.4 Notes and remarks
No further notes and remarks available

2.13.6 Delete existing property set version relationship
2.13.6.1 Start state and preconditions
A relationship between two instances of the class PROP.property_set_version to be deleted does exist.

2.13.6.2 Required operations
- Disconnect.PROP.property_set_version_relationship

2.13.6.3 End state and post condition
The two instances of the class PROP.property_set_version are no longer associated with each other.

2.13.6.4 Notes and remarks
The deletion of instances of the class PROP.property_set_version_relationship at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.13.7 Create new property set structure
2.13.7.1 Start state and preconditions
Two instances of the class PROP.property_set do exist.

2.13.7.2 Required operations
- Connect.PROP.property_set_structure

2.13.7.3 End state and post condition
Two instances of the class PROP.property_set are associated with each other in a structure of property sets.
2.13.7.4 Notes and remarks
A structure of properties can be used to define complex property sets built up by subsets.

2.13.8 Delete existing property set structure
2.13.8.1 Start state and preconditions
A relationship PROP.property_set_structure between two instances of the class PROP.property_set to be deleted does exist.

2.13.8.2 Required operations
- Disconnect.PROP.property_set_structure

2.13.8.3 End state and post condition
The two instances of the class PROP.property_set are no longer associated with each other.

2.13.8.4 Notes and remarks
The deletion of instances of the relationship PROP.property_set_structure at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.13.9 Create a property
2.13.9.1 Start state and preconditions
An instance of the class PROP.property_set_version exists and is in a state allowing change.

2.13.9.2 Required operations
- Create.PROP.property (one of its sub types)
- Connect.PROP.property_of_set
And, depending on the sub type of PROP.property, possibly one of …
- Connect.PROP.element_specification
- Connect.PROP.property_parameter

2.13.9.3 End state and post condition
A property as an instance of the class PROP.property is available within the data management system and is assigned to an instance of the class PROP.property_set_version.

2.13.9.4 Notes and remarks
No further notes and remarks available

2.13.10 Delete a property
2.13.10.1 Start state and preconditions
An instance of the class PROP.property exists and is in a state allowing change. There are no further objects assigned to the property.

2.13.10.2 Required operations
- Delete.PROP.property (one of its sub types)
- Disconnect.PROP.property_of_set

Copyright: VDA
And, depending on the sub type of PROP.property, possibly one of …

- DisconnectPROP.property.element_specification
- DisconnectPROP.property.property_parameter

### 2.13.10.3 End state and post condition
A property as an instance of the class PROP.property is no longer available within the data management system.

### 2.13.10.4 Notes and remarks
The deletion of instances of the class PROP.property at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

### 2.13.11 Create property relationship

#### 2.13.11.1 Start state and preconditions
Two instances of the class PROP.property do exist.

#### 2.13.11.2 Required operations
- ConnectPROP.property_property_relationship

#### 2.13.11.3 End state and post condition
Two instances of the class PROP.property are associated with each other in a general manner.

#### 2.13.11.4 Notes and remarks
To create a property hierarchy it can be useful to delete the relationship PROP.property_of_set of the lower level property.

### 2.13.12 Delete existing property relationship

#### 2.13.12.1 Start state and preconditions
An instance of the relationship PROP.property_relationship between two instances of the class PROP.property does exist.

#### 2.13.12.2 Required operations
- DisconnectPROP.property_property_relationship

#### 2.13.12.3 End state and post condition
Two instances of the class PROP.property are no longer associated with each other.

#### 2.13.12.4 Notes and remarks
The deletion of instance of the relationship PROP.property_relationship at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

### 2.13.13 Create property constraint

#### 2.13.13.1 Start state and preconditions
An instance of the class PROP.property does exist.

#### 2.13.13.2 Required operations
- CreatePROP.property_property_constraint
- ConnectPROP.property_property_constraint_property
2.13.13.3 End state and post condition
An instance of the class PROP.property_constraint is defined and assigned to an instance of the class PROP.property.

2.13.13.4 Notes and remarks
Optionally the use case Document Management might be applied for the new instance of the class PROP.property_constraint.

2.13.14 Delete existing property constraint

2.13.14.1 Start state and preconditions
An instance of the class PROP.property_constraint does exist.

2.13.14.2 Required operations
- Delete PROP.property_constraint
- Disconnect.PROP.property_constraint_property

And if the use case Document Management is applied for the instance of the class PROP.property_constraint,
- Disconnect.PROP.property_constraint_document

2.13.14.3 End state and post condition
The instance of the class PROP.property_constraint for an instance of the class PROP.property is no longer available with the data management system.

2.13.14.4 Notes and remarks
The deletion of instances of the class PROP.property_constraint at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.13.15 Assign property set version to model version

2.13.15.1 Start state and preconditions
An instance of the class PROP.property_set_version and an instance of the class BASE.model_version do exist.

2.13.15.2 Required operations
- Connect.PROP.model_version_property_set_version

2.13.15.3 End state and post condition
An instance of the class PROP.property_set_version is assigned to an instance of the class BASE.model_version.

2.13.15.4 Notes and remarks
No further note and remark available.

2.13.16 Delete property set version from model version

2.13.16.1 Start state and preconditions
An instance of the relationship PROP.model_version_property_set_version between an instance of the class PROP.model_version and an instance of the class PROP.property_set_version does exist.

2.13.16.2 Required operations
- Disconnect.PROP.model_version_property_set_version
2.13.16.3 End state and post condition
The instance of the class PROP.property_set_version is no longer associated with an instance of the class BASE.model_version.

2.13.16.4 Notes and remarks
The deletion of instances of the relationship PROP.model_version_property_set_version at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.13.17 Connect property set version to model element
2.13.17.1 Start state and preconditions
An instance of a subtype of the class TOPO.model_element and an instance of the class PROP.property_set_version do exist.

2.13.17.2 Required operations
- Connect.PROP.associated_property_set_version

2.13.17.3 End state and post condition
An instance of the class TOPO.model_element is described by a set of properties.

2.13.17.4 Notes and remarks
No further notes and remarks available.

2.13.18 Disconnect property set version to model element
2.13.18.1 Start state and preconditions
A relationship between an instance of a subtype of the class TOPO.model_element and an instance of the class PROP.property_set_version does exist.

2.13.18.2 Required operations
- Disconnect.PROP.associated_property_set_version

2.13.18.3 End state and post condition
The assignment of the instance of the class PROP.property_set_version to an instance of the class TOPO.model_element is no longer available.

2.13.18.4 Notes and remarks
No further notes and remarks available.

2.14 Functionality of use case Setting Definition Management
See SimPDM recommendation for a general description of the use case Setting Definition Management.

2.14.1 Create new setting
2.14.1.1 Start state and preconditions
No start state and precondition to be defined.

2.14.1.2 Required operations
- Create.SETT.setting (or one of the subtypes)

2.14.1.3 End state and post condition
A new SETT.setting definition is available within the data management system.
2.14.1.4 Notes and remarks
No further notes and remarks available.

2.14.2 Delete existing setting

2.14.2.1 Start state and preconditions
A SETT.setting to be deleted does exist and is in a stage allowing change, i.e. it has no further objects like BASE.document, etc. assigned. The SETT.setting might have an associated BASE.administration which is also subject to be deleted then (see chapter 2.1).

2.14.2.2 Required operations
- Delete. SETT.setting (or one of the subtypes)

2.14.2.3 End state and post condition
The SETT.setting definition is no longer available within the data management system.

2.14.2.4 Notes and remarks
The deletion of SETT.setting at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.14.3 Assign setting to an analysis

2.14.3.1 Start state and preconditions
A BASE.analysis_version does exist and is in a state allowing change. For high granularity load management existence of SETT.setting or one of its subtypes can be necessary.

2.14.3.2 Required operations
- Create.BASE.setting_for_analysis
- Connect.BASE.setting_for_analysis_version

And, in case of high granularity level data management, additionally
- Create.SETT.applied_settings (or one of its subtypes)
- Connect.SETT.applied_settings_for_analysis

And, at least one of...
- Connect.SETT.applied_analysis_setting
- Connect.SETT.applied_environment_setting
- Connect.SETT.applied_general_setting
- Connect.SETT.applied_user_defined_sub_routine

2.14.3.3 End state and post condition
A BASE.load_for_analysis is assigned to a BASE.analysis_version. In case of high level granularity this is further detailed in the assigned SETT.applied_settings and its assignments.

2.14.3.4 Notes and remarks
BASE.setting_for_analysis can be defined in a document. Document attachment in general is subject of another use case (see chapter 2.4).
2.14.4 Delete setting assignment from analysis

2.14.4.1 Start state and preconditions

A BASE.setting_for_analysis is assigned to a BASE.analysis_version which is in a state allowing change. In case of high level granularity the setting is further detailed in the assigned SETT.applied_settings or one of its subtypes and its assignments which also exist and are in a state allowing change. There is no BASE.document attached to BASE.setting_for_analysis (for that see chapter 2.4).

2.14.4.2 Required operations

- Delete.BASE.setting_for_analysis
- Disconnect.BASE.setting_for_analysis_version

And, in case of high granularity level data management, additionally

- Delete.SETT.applied_settings (or one of its subtypes)
- Disconnect.SETT.applied_settings_for_analysis

And, depending on assigned settings ...

- Disconnect.SETT.applied_analysis_setting
- Disconnect.SETT.applied_environment_setting
- Disconnect.SETT.applied_general_setting
- Disconnect.SETT.applied_user_defined_sub_routine

2.14.4.3 End state and post condition

A BASE.setting_for_analysis is no longer available within the data management system. Also no longer available are the SETT.applied_setting or one of its subtypes and the assignments.

2.14.4.4 Notes and remarks

The SETT.setting or one of its subtypes and the BASE.model_version still exist within the data management system.

The deletion of setting assignments at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.14.5 Create analysis step

2.14.5.1 Start state and preconditions

A BASE.setting_for_analysis exists and is in a state allowing change. A LOAD.applied_load_case also exists, is also in a state allowing change and is assigned (indirectly) to the same BASE.analyse_version.

2.14.5.2 Required operations

- Create.SETT.analysis_step
- Connect.SETT.applied_settings_for_analysis
- Connect.SETT.analysis_step_applied_load

2.14.5.3 End state and post condition

A SETT.analysis_step organises as a setting the sequence of load cases within a BASE.analysis_version. This organisation is available within the data management system.
2.14.5.4 Notes and remarks

It is important that assigned objects (settings as well as loads) belong to the same BASE.analysis_version.

2.14.6 Delete analysis step

2.14.6.1 Start state and preconditions

A SETT.analysis_step exists and is in a state allowing change. There are no further BASE.documents, BASE.parameters or BASE.administration assigned to the SETT.analysis_step (for disconnecting them see the according chapters).

2.14.6.2 Required operations

- Delete.SETT.analysis_step
- Disconnect.SETT.applied_settings_for_analysis
- Disconnect.SETT.analysis_step_applied_load

2.14.6.3 End state and post condition

A SETT.analysis_step is no longer available within the data management system.

2.14.6.4 Notes and remarks

The BASE.settings_for_analysis and the LOAD.applied_load_case still exist within the data management system (for deletion refer to the according chapters).

The deletion of SETT.analysis_step at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.14.7 Add another detailed setting to an analysis

2.14.7.1 Start state and preconditions

A BASE.setting_for_analysis does exist and is in a state allowing change. A SETT.setting or one of its subtypes exists.

2.14.7.2 Required operations

- Create.SETT.applied_settings (or one of its subtypes)
- Connect.SETT.applied_settings_for_analysis
- Connect.SETT.applied_analysis_setting
- Connect.SETT.applied_environment_setting
- Connect.SETT.applied_general_setting
- Connect.SETT.applied_user_defined_sub_routine

And one of...

- Connect.SETT.applied_analysis_setting
- Connect.SETT.applied_environment_setting
- Connect.SETT.applied_general_setting
- Connect.SETT.applied_user_defined_sub_routine

2.14.7.3 End state and post condition

Another SETT.applied_settings or one of its subtypes and its defining SETT.setting or one of its subtypes is assigned to BASE.settings_for_analysis within the data management system.

2.14.7.4 Notes and remarks

No further notes and remarks available.
2.14.8 Delete a detailed setting assignment

2.14.8.1 Start state and preconditions

A SETT.applied_settings or one of its subtypes and its defining SETT.setting or one of its subtypes is assigned to BASE.settings_for_analysis within the data management system. The SETT.applied_settings is in a state allowing change.

2.14.8.2 Required operations

- Delete.SETT.applied_settings (or one of its subtypes)
- Disconnect.SETT.applied_settings_for_analysis

And depending on the assigned setting …

- Disconnect.SETT.applied_analysis_setting
- Disconnect.SETT.applied_environment_setting
- Disconnect.SETT.applied_general_setting
- Disconnect.SETT.applied_user_defined_sub_routine

2.14.8.3 End state and post condition

A SETT.applied_settings or one of its subtypes is no longer available at the data management systems. Its relations are also no longer available.

2.14.8.4 Notes and remarks

The deletion of setting assignments at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.15 Functionality of use case Topology Element Definition Management

See SimPDM recommendation for a general description of the use case Topology Element Definition Management.

2.15.1 Create new topological element

2.15.1.1 Start state and preconditions

No start state and precondition to be defined.

2.15.1.2 Required operations

- Create.TOPO.model_element (or one of the subtypes)

2.15.1.3 End state and post condition

A new instance of the class TOPO.model_element definition is available.

2.15.1.4 Notes and remarks

A topological element should either be assigned to an instance of the class BASE.model (use case Topology Structure Definition Management involved) or be managed by a topological element library. Instances of topological elements should not exist unconnected and unmanaged within a data management system. If a topology element library is not available or a topology element is created independently from the library, the new topology element should be assigned to an instance of the class BASE.model_version.

Optionally the use cases Document Management, Administration Management and Parameter Association Management might be applied.
2.15.2 Delete existing topological element

2.15.2.1 Start state and preconditions
An instance of the class TOPO.model_element to be deleted does exist.

2.15.2.2 Required operations
  ▪ Delete.TOPO.model_element (or one of the subtypes)

2.15.2.3 End state and post condition
The instance of the class TOPO.model_element definition is no longer available within the data management system.

2.15.2.4 Notes and remarks
Actually, the deletion of instances of the class TOPO.model_element from a library is not a daily business, but rather an administrative task of a library manager.
The deletion of instances of the class TOPO.model_element at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.15.3 Create topological element relationship

2.15.3.1 Start state and preconditions
At least two instances of the class TOPO.model_element do exist.

2.15.3.2 Required operations
  ▪ Connect.TOPO.model_element_relationship

2.15.3.3 End state and post condition
Two instances of the class TOPO.model_element are related with each other in a general manner.

2.15.3.4 Notes and remarks
No further notes and remarks available

2.15.4 Delete topological element relationship

2.15.4.1 Start state and preconditions
A relationship between two instances of the class TOPO.model_element does exist.

2.15.4.2 Required operations
  ▪ Disconnect.TOPO.model_element_relationship

2.15.4.3 End state and post condition
The relationship between two instances of the class TOPO.model_element is no longer available.

2.15.4.4 Notes and remarks
The deletion of instances of the relationship TOPO.model_element_relationship at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.
2.15.5 Include model element into model

2.15.5.1 Start state and preconditions
At least one instance of the class BASE.model_version and one instance of the class TOPO.model_element do exist.

2.15.5.2 Required operations
- Connect.TOPO.of_model

2.15.5.3 End state and post condition
A model element is included into a model.

2.15.5.4 Notes and remarks
No further notes and remarks available

2.15.6 Delete model element from model

2.15.6.1 Start state and preconditions
At least one instance of the class BASE.model_version with at least one included instance of the class TOPO.model_element or one of its sub classes does exist.

2.15.6.2 Required operations
- Disconnect.TOPO.of_model

2.15.6.3 End state and post condition
A TOPO:model_element is no longer included into a model, but is still available within the data management system.

2.15.6.4 Notes and remarks
An instance of the class TOPO.model_element should not exist outside of a model, if it is not managed by a topology element library. Therefore it should automatically be deleted.

2.16 Functionality of use case Topology Structure Definition Management

See SimPDM recommendation for a general description of the use case Topology Structure Definition Management.

2.16.1 Define interface element

2.16.1.1 Start state and preconditions
An instance of one of the classes TOPO.geo_element, TOPO.node or TOPO.surface or of one their sub classes does exist.

2.16.1.2 Required operations
- Create.TOPO.model_interface
And, depending on the type of model element, that is to be defined as an interface element, one of...
- Connect.TOPO.surface_model_interface
- Connect.TOPO.node_model_interface
- Connect.TOPO.geo_element_model_interface
2.16.1.3 End state and post condition
A specific model element within a topological structure is specified as an interface element.

2.16.1.4 Notes and remarks
Model elements specified as interface elements can be used for automatic topological connections between different models.

2.16.2 Delete interface element
2.16.2.1 Start state and preconditions
An instance of the class TOPO.model_interface to be deleted does exist and is not used by a relationship with another instance of the class TOPO.model_interface.

2.16.2.2 Required operations
- Delete.TOPO.model_interface
And, depending on the type of model element, that is defined as an interface element, one of...
- Disconnect.TOPO.surface_model_interface
- Disconnect.TOPO.node_model_interface
- Disconnect.TOPO.geo_element_model_interface

2.16.2.3 End state and post condition
A specific model element within a topological structure is no longer specified as an interface element.

2.16.2.4 Notes and remarks
The instance of the class TOPO.model_element that was formerly specified as an interface element is still available.

The deletion of instances of the class TOPO.model_interface and the relationships at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.16.3 Create topological model connection
2.16.3.1 Start state and preconditions
Two instances of the class TOPO.interface_element belonging to different instances of the class BASE.model do exist.

2.16.3.2 Required operations
- Connect.TOPO.cross_model_interface_relation

2.16.3.3 End state and post condition
Two different instances of the class BASE.model_version are connected by an instance of the class TOPO.interface_element in each of the model versions represented by instances of the class BASE.model_version.

2.16.3.4 Notes and remarks
An instance of the class TOPO.interface_element can only once at a time be used in a topological model connection.

Copyright: VDA
2.16.4 Delete topological model connection

2.16.4.1 Start state and preconditions
A relationship between two instances of the class TOPO.interface_element belonging to different instances of the class BASE.model does exist.

2.16.4.2 Required operations
- Disconnect.TOPO.cross_model_interface_relation

2.16.4.3 End state and post condition
The two instances of the class TOPO.interface_element are no longer connected.

2.16.4.4 Notes and remarks
The deletion of instances of the relationship TOPO.cross_model_interface_relation at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.

2.16.5 Assign a surface to a set

2.16.5.1 Start state and preconditions
At least one instance of the class TOPO.surface and an instance of the class TOPO.set do exist.

2.16.5.2 Required operations
- Connect.TOPO.belongs_to

2.16.5.3 End state and post condition
The instance of the class TOPO.surface is assigned to an instance of the class TOPO.set.

2.16.5.4 Notes and remarks
No further notes and remarks do exist.

2.16.6 Delete a surface from a set

2.16.6.1 Start state and preconditions
At least one instance of the class TOPO.surface assigned to an instance of the class TOPO.set do exist.

2.16.6.2 Required operations
- Disconnect.TOPO.belongs_to

2.16.6.3 End state and post condition
The instance of the class TOPO.surface is no longer assigned to an instance of the class TOPO.set.

2.16.6.4 Notes and remarks
The instance of the class TOPO.surface formerly assigned to the instance of the class TOPO.set is still available.

The deletion of instances of the relationship TOPO.belongs_to at this point is described with regards to the data model specification. Nevertheless, the deletion of objects might specifically be restricted by data management system restrictions.