Cameo Concept Modeler Documentation

Cameo Concept Modeler (Concept Modeler or CCM) is a plugin for MagicDraw, the award-winning software modeling tool. CCM is designed to model real-world concepts, import/export a model from/to an OWL ontology, and generate glossaries in plain English for a clearer and more informative source of knowledge for any domain.

2021x Version News

Cameo Concept Modeler (CCM) Quick Start Guide

Introduction

- MDA
- Concept modeling purpose
- The role of ontologies and reasoners
- Open-world assumption vs. closed-world assumption
- Information modeling purpose

Cameo Concept Modeler Capabilities

Concept Modeling Semantics

- Classes
- Anonymous union classes
- Advanced Modeling Patterns
  - Association class
  - Facets
    - Role
    - Phase
- Equivalent classes
- Conditions
- Property ownership
- Global properties
- Equivalent properties
- Subproperty
- Property chain
- Property restrictions
  - Existential quantification constraint
  - Universal quantification constraint
  - Cascading restrictions
- Cardinality restrictions
- Inverse properties
- Object properties
- Annotation and annotation properties
- Preferred Annotation Property
- Generalization
  - Incomplete and overlapping subclasses
  - Disjoint subclasses
  - Complete subclasses
  - Complete and disjoint subclasses
- Multiplicities
- IRI tagged value
  - Effective IRI meta-property
  - Synchronize UML Package URI and Resource IRI
- Complement Of
- Importing OWL
- Concept model export URI style
- OWL export folder
- UPCM library in CCM
- Equivalent classes in NLG
- Working with superclass intersection
- Intersection

UML to Equivalent OWL in OWL Functional Syntax

- Class
- Class generalization
- Class with Datatype property
- Class with object property
- Class with Self-Referential Object Property
- Class with object property without range
- Class with subproperty
- Class with universal quantification constraint on property I
- Class with universal quantification constraint on property II
- Class with existential quantification constraint on property
- Class with subproperty without a range

- Class with necessary and sufficient property
- Class with property having unspecified multiplicity
- Class with inverse property
- Class with Asymmetric Object Property
- Class with Functional Object Property
- Class with Inverse Functional Object Property
- Class with Irreflexive Object Property
- Class with Reflexive Object Property
- Class with symmetric object property
- Class with Transitive Object Property
- Generalization with disjoint subclasses
- Generalization with subclass completeness
- Anonymous union class
- Association classes
- Property holder with datatype property
- Property holder with self-referential object property
- Property holder with object property
- Property holder with self-referential subproperty
- Property holder with subproperty
- Property with a maximum but no minimum cardinality
- Property with multiple domains and ranges
- Annotation and annotation property
- Asymmetrical inverse property
- Disjoint classes
- Property chains
- Equivalent property
- Equivalent class
- Property restriction from a different namespace
- Necessary and sufficient conditions of anonymous subclasses
    - Intersection subset of Union
    - Intersection equivalent to Union
    - Intersection subset of a Restriction
    - Intersection equivalent to a Restriction
    - Intersection disjoint with a Restriction
    - Intersection subset of an Intersection
    - Intersection equivalent to an Intersection
    - Intersection subset of Complement
    - Intersection equivalent to Complement
    - Intersection disjoint with Complement
    - Intersection subclass of Class
    - Union subset of Union
    - Union equivalent to Union
    - Union subset of Restriction
    - Union equivalent to Restriction
    - Union disjoint with Restriction
    - Union subset of Intersection
    - Union subset of Complement
    - Union equivalent to Complement
    - Union has member Complement
    - Union disjoint with Complement
    - Union subset of Class
    - Complement subset of Union
    - Complement disjoint with Union
    - Complement subset of Restriction
    - Complement subset of Intersection
    - Complement disjoint with Restriction
    - Complement has member Restriction
    - Complement equivalent to Restriction
    - Complement disjoint with Intersection
    - Complement subset of Complement
    - Complement equivalent to Complement
    - Complement has member Complement
    - Complement subset of Class
    - Restriction subset of Union
    - Restriction disjoint with Union
    - Restriction subset of Restriction
    - Restriction equivalent Restriction
    - Restriction disjoint with Restriction
    - Restriction subset of Intersection
    - Restriction disjoint with Class
    - Restriction subset of Class
    - Restriction disjoint with Complement
    - Restriction subset of Complement
    - Restriction disjoint with Intersection
    - Class subset of Union
    - Class subset of Restriction
    - Class equivalent to Restriction
    - Class disjoint with Restriction
    - Class subset of Complement

    - Class disjoint with Class
    - Class disjoint with Complement
    - Class disjoint with Intersection
    - Class disjoint with Union
    - Class equivalent to Class
    - Class subset of Class
    - Class subset of Intersection
    - Complement disjoint with Class
    - Complement disjoint with Complement
    - Complement equivalent to Class
    - Complement has member Class
    - Complement has member Intersection
    - Complement has member Union
    - Intersection disjoint with Class
    - Intersection disjoint with Intersection
    - Intersection disjoint with Union
    - Intersection equivalent to Class
    - Intersection has member Class
    - Intersection has member Complement
    - Intersection has member Intersection
    - Intersection has member Restriction
    - Intersection has member Union
    - Restriction has member Class
    - Restriction has member Complement
    - Restriction has member Intersection
    - Restriction has member Restriction
    - Restriction has member Union
    - Union disjoint with Class
    - Union disjoint with Intersection
    - Union disjoint with Union
    - Union equivalent to Class
    - Union has member Class
    - Union has member Intersection
    - Union has member Restriction
    - Union has member Union

AutoStyler Documentation

- Set Defining Diagram
- Repair Styles
- Enable Auto Diagram Assignment
- Disable Auto Diagram Assignment
- Customizing AutoStyler
    - How to change the color of AutoStyler
    - How to remove package names of classes

Usage

- Creating a concept modeling project
- Creating a concept model
- Converting a UML model into a concept model
- Existing Project Migration
    - Migrate Older Models to Use Relative IRIs
    - Updating symbol styles in existing projects
    - Restrictions
    - Generalization Set Names
    - Sufficient constraint
- Working with non-CCM projects
    - Importing an OWL ontology into a non-CCM project
    - Using Concept Modeling Capabilities with non-CCM projects
- Experimental Features
    - Working with association classes
    - Working with phases
    - Working with roles
- Importing an OWL ontology to a concept model
    - Creating an XML catalog file
    - Updating the XML catalog file
    - Setting the OWL import catalog in MagicDraw
    - Creating a path variable to share OWL import catalog files
    - Using a path variable to share OWL import catalog files
    - Importing an OWL ontology file
    - Using the log file to track changes during import
- Exporting your concept model to an OWL ontology
    - Setting destination for the OWL export folder
    - Setting the concept model export syntax
    - Setting the concept model export URI style
    - Setting the concept model URI
    - Specifying file export paths
    - Specifying IRI ontology versions

    - Use Path Variables to Export a Concept Model to an OWL Ontology
    - Exporting your concept model
    - Exporting models and concept models at any level in package hierarchy
    - View the CCM watermark in an exported OWL
- Adding a concept model to Teamwork Cloud and export it as an OWL ontology
    - Adding a concept model to Teamwork Cloud
    - Exporting a concept model from TWCloud to an OWL ontology
- Logging during OWL importing and exporting
- Freezing and unfreezing IRIs
- Displaying and hiding IRI
- Creating equivalent classes
- Generalizations with read-only subclasses
- Working with Complement Of
- Working with subclasses
    - Making subclasses disjoint
    - Making subclasses complete
    - Making subclasses overlapping
    - Making subclasses incomplete
- Working with properties
    - Creating a property holder
    - Creating a property chain
    - Editing a property chain
    - Deleting a property chain
    - Creating equivalent properties
    - Editing an equivalent property
    - Deleting an equivalent property
    - Redefined property
    - Subsetted property
    - Adding property subsetting
    - Removing property subsetting
    - Creating a datatype property
    - Creating and removing a necessary and sufficient condition
    - Working with subproperties
- Working with restrictions
    - Creating Restrictions
    - Editing Restrictions
    - Removing Restrictions
- Working with glossaries
    - Automatically Generating Glossaries
    - Creating a glossary table
    - Rebuilding a glossary table
    - Viewing a glossary
- Working with annotations
    - Creating annotations
    - Showing annotations on the diagram
    - Showing an annotation in the Documentation pane
    - Working with annotation properties
        - Applying an annotation stereotype to a comment
        - Associating an annotation property with an annotation
        - Defining an annotation property
        - Importing an ontology that defines annotation properties
        - Selecting a Preferred Annotation Property for a UML Comment or Annotation
- Working with the Natural Language Glossary
    - Generating a natural language glossary
    - Exploring the Natural Language Glossary
    - Customizing your Natural Language Glossary
        - Including property definitions in the Natural Language Glossary
        - Selecting an ordered list of annotation properties
        - Variables of the natural language glossary
- Working with intersection

References