Concept Modeler Capabilities

Concept Modeler's major capabilities are explained as follows:

SME friendly graphical notation

- Uses consultant proven “SME (subject matter expert) friendly” graphical notation.
- Facilitates real-time interactions with SMEs to model real world business concepts and their relationships.
- Requires no camel case names.
- Sets, by default, visibility of properties to public.
- Encourages clean, hyperlinked micro-subject-area diagrams.

Automatic styling of concept model

Large-scale models often contain information that is tangled in a complex web of relationships and therefore are difficult to read and focus on. Even though MagicDraw is the best modeling tool on the market, without the AutoStyler capability, untangling requires quite a bit of effort, so many modelers don't bother to try.

The Concept Modeler, through the AutoStyler plugin, assists the modeler in producing diagrams that are concise, focused, and presentable to stakeholders.

Central to AutoStyler is a style named “Defined Elsewhere” that is applied to diagrammed model elements when they are not defined on the current diagram. Traditionally, this style collapses all compartments and fades the normal element colors, including association ends that are not defined on the current diagram. The modeler can fine tune this style in any way the MagicDraw style system allows, and, if desired, even retain the default style for some or all kinds of elements.

AutoStyler examines two factors to determine whether the current diagram is eligible to be the defining diagram for an element being added to a diagram. The first is whether or not a non-descendant package owns the added element. In other words, when the added element is owned by an element that is not a child of the diagram's owner, it is defined elsewhere. An example of this is when a package other than the diagram's owning package owns the added class. The second is whether or not the added element has a hyperlink to a diagram, which indicates the defining diagram for the element. When an added element's hyperlink is not the current diagram, it is clearly defined elsewhere.

AutoStyler has a mode to automatically assign a hyperlink that points to the current diagram when it is eligible to be the defining diagram for an element being added to it. This automatic mode can be turned off as well.

The "Defined Elsewhere" style can be defined differently for each project, or defined the same across all projects. For example, one might use a UPDM architectural model and a UML software model with one standard "Defined Elsewhere" style that works for both kinds of models. This style is usually a clone of the "Default" style that has been adjusted to fade fill colors, text colors, and line colors, which can usually be done at the top level using opacity settings.

AutoStyler allows the modeler to select one or more elements on a diagram as the defining diagram for them, as long as they meet the criteria described above for defining diagram eligibility. AutoStyler cannot automatically change the style of an element on a diagram when its defining diagram status changes on any but the current diagram. AutoStyler allows the modeler to select these elements and “repair” the styles to reflect this change in status.

The “Default” and “Defined Elsewhere” styles are tuned for working with concept models. The symbol properties within those styles are, by default, set as follows:

- Visibility and stereotypes for properties are suppressed.
- Tagged values for association ends and classes are suppressed.
- Tagged values for attributes are shown.
- Subsets, redefines, and constraints for properties are shown.
- Constraint names for properties are shown (instead of constraint expressions).
- Properties of Property Holders are always shown.
- Association-end properties for Property Holders are shown as attributes when the “Defined Elsewhere” style is applied, and the Association Ends are not shown.

AutoStyler allows the modeler to change any of these settings element by element, or by changing them in the “Default” and “Defined Elsewhere” styles.

AutoStyler is currently a separate plugin for MagicDraw, but is expected to become part of the base MagicDraw product in the future.

Automatic glossary generation

Using a glossary saves time by ensuring consistent usage of terminology in the organization. It also improves the communication between team members since terms are understood in the same way and definitions become visible everywhere the terms are used.

Depending on project options, automatic glossary generation in a concept model can create a glossary containing the names and descriptions of classes, association ends, attributes, enumerations, and / or enumeration literals used in the owning concept model. These glossaries are generated on import, element creation in the containment tree, or element creation on the diagram. When creating a new class, association end, attribute, enumeration, or enumeration literal in the containment tree or on the diagram, the element will not be added to the glossary until the user names the element.

For automatic glossary generation, a user is provided with five project options:

- Add classes to a glossary. When a class is created in the Containment tree, created on a diagram, or imported from an ontology, the class name and documentation will be added to a glossary in the owning concept model.
- Add association ends to a glossary. When an association end is created in the Containment tree, created on a diagram, or imported from an ontology, the class name and documentation will be added to a glossary in the owning concept model.
- Add attributes to a glossary. When an attribute is created in the Containment tree, created on a diagram, or imported from an ontology, the attribute name and documentation will be added to a glossary in the owning concept model.
- Add enumerations to a glossary. When an enumeration is created in the containment tree, created on a diagram, or imported from an ontology, the enumeration name and documentation will be added to a glossary in the owning concept model.
- Add enumeration literals to a glossary. When an enumeration literal is created in the Containment tree, created on a diagram, or imported from an ontology, the enumeration literal and documentation will be added to a glossary in the owning concept model.

You may change these project options at a later time. You can build or rebuild a glossary table containing only the kinds of entries selected in the project options. When you create a glossary table for any selected «Concept Model» stereotyped package, the Concept Modeler will add only the elements that exist inside the selected package to the glossary. For example, you have a project that has two packages: package A and package B. When you create a glossary for package A, the glossary only includes data from the selected package A. It does not contain any data from package B.

Just like creating a glossary, rebuilding a glossary table works the same way by allowing only the elements from a selected package to be kept. For example, you have created the glossary for package A and you later added some terms or elements (classes, association ends, attributes, enumerations, and enumeration literals) to the glossary manually. When you rebuild that glossary, the terms or elements that you have manually added will not be included in the glossary. The same thing will happen when you move some of the elements from package A to package B and then rebuild the glossary for package A. You will not find the elements that you removed from package A in the glossary.

Attributes and association ends can be suppressed from a glossary when their names are too generic. For example, when a property is called "in", every occurrence of that word in any other description, therefore, undesirably becomes a hyperlink to the property called "in". Additionally, automatic glossary generation can be turned off. Existing glossary entries are not removed when automatic glossary generation is turned off.

The glossary table allows for managing the terms of a concept model in a spreadsheet-like form. Each row in the table represents a term, which can be a word, a phrase, or any element of the model. With the help of this table, a user can easily:

- Create and manage all terms of the model in a single place.
- Customize the representation of the table.
- Export the data into an *.html, *.csv, or *.xlsx file.

For more information, please refer to the user manual for MagicDraw 18.0 SP4 or higher.

Concept model authoring

Concept Modeler can do the following:

- Create a concept model from scratch or by importing an OWL ontology for reuse and/or as a starting point for the creation of a concept model.
- Graphically represent imported RDFS/OWL 2 ontologies.
- Provide graphical concept model authoring with subject matter experts.
- Integrate with any UML model or UML-based standard, such as the Unified Profile for MoDAF and DoDAF, and NIEM-UML.
    - Support the Open World Assumption (i.e., absence of information is interpreted as unknown information, not as negative information)*.
- Export a concept model to an OWL 2 ontology for reasoning over and adding further precision to constrain possible interpretations.
- Support the creation of Closed World Assumption information models.
- Automatically style classes defined in other packages and diagrams.

To see all elements that Concept Modeler can import or export, refer to Concept Modeling Semantics and UML to Equivalent OWL in OWL Functional Syntax.

UML model traceability

The Concept Modeler uses UML to build models. Therefore, concept models built by the Concept Modeler can be traced to any UML model, (e.g., NIEM-UML).

Semantic integration of multiple information models

A concept model built by the Concept Modeler provides the semantics to integrate multiple information models “subsetted” from the concept model:

- Information at rest (e.g., relational and XML databases).
- Information in motion (e.g., XSD schema and NIEM-UML).

Natural language glossary

In addition to glossary tables, the Concept Modeler provides a separate feature for generating a natural-language glossary. Natural language glossaries are intended for technical and non-technical people alike. For instance, concept modelers can ensure that the model indeed says what was intended. Subject matter experts can ensure that the model captures their business knowledge correctly. And system builders can find definitions for the terms used in requirements in much more detail than usual.

A natural language glossary converts the elements in a concept model into natural-language sentences. Every class creates a hyperlinked glossary entry that describes its superclass(es), necessary and sufficient properties, necessary properties, and optional properties. Any user-supplied documentation is transcribed at the end of each glossary entry. That documentation can add supplemental definitions such as examples and counter-examples.

Users will find that the better the model, the clearer the auto-generated glossary.

Annotation properties in the natural language glossary

Cameo Concept Modeler offers a project option that allows the selection of which annotation properties will be shown or hidden in every natural language glossary entry, in addition to the definitions generated from the semantics of a concept model. You can select any number of annotation properties. Elements in the report such as Classes or Properties that are annotated with a «Annotation» stereotyped UML comment that contains one of these annotation properties will display the UML comment body in the report. When no comment body exists the name of the annotation property will display by itself.

In our software, the feature is labeled “Natural Language Glossary annotation property list” and it consists of a list of pre-loaded annotation properties.

Preferred annotation property

Cameo Concept Modeler offers a project option that speeds up the creation of one of many possible kinds of annotations, and specifies which kind of annotation to treat as documentation. You can select a preferred annotation property to be assumed whenever the user adds a UML Comment or an annotation. Comments and Annotations that explicitly use that annotation property will then be shown in the documentation panel, and in the Natural Language Glossary, as the human-specified definition (as opposed to the model-generated definition).

If you import a concept model that contains annotations, they are placed in its owning folder and each annotation has an annotated element and can have an annotation property tagged value. When you select a preferred annotation property, the «Annotation»s owned by the annotated element for the preferred annotation property will appear as documentation in MagicDraw.

Any annotations on ontology itself are imported correctly by CCM as annotations.

If your project is a TWC project, Concept Modeler will attempt to lock the project’s elements. If any of the elements cannot be locked, whether it is locked by another user, then several message windows will appear, notifying you of the problem and allowing you to see which elements are not locked.

Creation of multiple data models from one concept model and connection of multiple existing data models to one concept model

Creation of multiple data models from one concept model (Diagram preservation after ontology import)

In addition to the Concept Modeler’s import capability, the software allows for diagram preservation after an ontology import. More specifically, while importing an ontology, the concept modeler will update the existing concept model. An ontology is imported into a CCM project that contains one or more concept models. Each ontology is imported into a concept model that may already be present in the project in which the ontology is imported. Ontology elements get translated into concept model elements.

The following table describes the conditions, evaluated by Concept Modeler, of each resource from ontology that is being imported. The condition determined by Concept Modeler will dictate how Concept Modeler will create/merge/delete the model element.

Condition	Description
New	An element is not present in the model project in which an ontology is being imported. Some parts of the ontology being imported may be already present in the model project but some of the other parts may be brand new.
Deleted	An element may be present in a concept model in the model project in which an ontology is being imported. This element however may be missing from the ontology being imported. This deletion need to be identified and element removed after the import.
Modified	An element may be present in a concept model in the model project in which an ontology is being imported. This element however may have different properties / values in the ontology being imported. This update to its properties need to be identified and updated after the import.
Same	An element remains unmodified after the import of an ontology.

The table below groups all the concept model elements and explains how they handle the “preservation” for each of the four conditions explained above.

Concept Model Element	Type of Update after Import	What can be modified?
Concept Model	New and Modified	Only two relevant conditions
	Deleted and Same	Irrelevant for the concept model.
Concept	New	New Concept is present only in the imported ontology for the given model.
	Modified	The concept's IRI matches but either or both of the name and owner are different.
	Deleted	Concept is present in the model but missing from the ontology. Mark all the new, same and modified ones in the original model. Delete the unmarked ones from the original model as these are the ones that are deleted in the ontology.
	Same	Following concept property match - Concept's IRI, name and its owner / model.
Concept Generalization	New	New generalization (to be identified by general and special concept) is present only in the ontology
	Modified	Not applicable
	Same	Generalization's general and special concept are same in the ontology as in the model.
	Deleted	Concept Modeler will mark all the new and same generalizations from the original model and delete all the unmarked ones since those are the generalizations deleted in the ontology.
Concept Disjoint Relationship	Explicit disjoint relation between 2 concepts	Identical to concept generalization
Concept Equivalence Relationship	New	New equivalence is present only in the ontology
	Modified	Not applicable
	Same	Equivalent concepts are the same in the ontology as in the model.
	Deleted	Concept Modeler will mark all the new and same generalizations from the original model and delete all the unmarked ones since those are the generalizations deleted in the ontology.
Anonymous Unions	Same	Same (anonymous) union to be identified by looking up constituents (set of concepts in the union) of the union against every existing union.
	Modified	Modified union is a union which has either a) same union constituents AND different (uniquely identifiable) properties or b) different constituents AND same (uniquely identifiable) properties
	New	New set constituents AND new properties
	Deleted	Other 3 cases should mark anonymous unions that are same, new or modified. Any unmarked anonymous unions in the original model are to be deleted.
Properties	New	New object property is present only in the imported ontology (to be determined using IRI)
	Same	Following properties of a object property should match- IRI, Domain, type, multiplicities
	Modified	Object property's IRI matches but one or more of the following values differ - Domain, type, multiplicities
	Deleted	Property is present only in the original model but is missing from the ontology being imported. Mark all the new, same and modified ones in the original model. Delete the unmarked ones from the original model as these are the ones that are deleted in the ontology.