



ConML Release Notes

version 1.5.2 · 1 June 2020



ConML by Institute of Heritage Sciences (Incipit), Spanish National Research Council (CSIC)



is licensed under a Creative Commons Attribution 4.0 International License.

This document and its contents were created by Cesar Gonzalez-Perez.

Partial funding was provided by Incipit · CSIC and projects MIRFOL (grant number 09SEC002606PR, INCITE Programme, Xunta de Galicia, Spain), ARIADNE (grant number 313193, FP7-INFRASTRUCTURES-2012-1) and MARIOL (grant number HAR2013-41653-R, Retos de la Sociedad, Plan Estatal 2013-2016, Spain).

1 June 2020 19:18 revision 1029

1 What's New in Version 1.5.2

The following list summarizes the new features in ConML version 1.5.2 as compared to the previous version:

- **Facets now separated by comma instead of semicolon.** Values in value sets, and references in reference sets, are now shown separated by commas rather than semicolons. Please see 5.3. *Object Diagrams*.
- **Removed support for multiple expressions per qualifier.** Previous versions supported multiple expressions per qualifier. This feature has not been useful and increases complexity, so it has been removed. Please see 5.3.4. *Qualification*.

2 Previous Versions

The following list summarizes features that were incorporated in version 1.5.1:

- **New notation for multi-node boxes.** A new notation artefact, using a dashed-lined box, has been introduced to simplify connectors attached to multiple nodes in a diagram. Please see 5.1.8. *Multi-Node Boxes*.
- **Removed redundant association in Language.** The `IsDefaultIn` association has been removed, as it was redundant with the `IsDefault` attribute. Please see 4.1.3. *Language*.

The following list summarizes features that were incorporated in version 1.5.0:

- **Model extension has been completely redesigned.** Model extension in previous versions of ConML was very restrictive in terms of what changes were allowed to a model during extension. A new and more permissive model extension approach has been designed. Please see *Type Model Extension*.
- **Metainformation is now supported.** The metamodel now includes support for metainformation objects documenting model elements. Please see *Metainformation*.
- **The semantics of model referencing have been developed.** References between models have been described and fleshed out in terms of the dependencies they generate and the reuse possibilities of model elements in the dependee models. Please see *References between Models*.
- **Multilingualism is now supported.** Type and instance models may now be expressed in multiple languages through a new language aspect. Please see *Multilingualism*.
- **Model views have been removed.** Model views were an early mechanism to allow for model customization. They have been removed from the specification in preparation for a simpler and more powerful customization mechanism that will be incorporated in the future. This change has allowed for a much simpler language architecture.
- **Constant semantics have been introduced.** In addition to temporal semantics, features may now be marked as constant. Please see *Feature*.
- **Temporal and perspective selectors are now called qualifiers.** For better consistency with existential qualifiers (see below), the term “qualifier” is now used instead of “selector” for the objects indexing phases and perspectives.
- **Existential aspect qualifiers have been introduced.** Previous versions of ConML allowed for the qualification of facets through temporal and subjective predicate qualifiers (previously called “selectors”). Now, aspect qualifiers may also be used for the existential qualification of objects. Please see *Soft Issues*.

- **Certainty qualifiers have been introduced.** The degree of certainty about the existence of an object or the predication of a facet can now be expressed. Please see *Vagueness*.
- **Notation for indirect relationships has been added.** Indirect generalization and instantiation relationships can be depicted now by using dashed lines. Please see *Generalization Relationships* and *Elements of the Type Model*.
- **The notation for predication qualifiers has been improved.** Now, predication-qualified facet sets are depicted through in-line qualifier expressions, rather than relying on the object box top section to bear an overall qualifier. Please see *Object Diagrams*.

The following list summarizes features that were incorporated in version 1.4.7:

- **A section has been added to the specification on informal variations.** These variations include localization as well as gender alternation. In particular, localized variants of the language are now regarded as informal. Please see *Informal Variations*.
- **The layout of specification tables is not part of the specification anymore.** Please see *Specification Tables*.
- **A “compact box” notation has been added for classes and objects.** Please see *Classes, Properties and Attributes*; and *Objects and Values*.
- **The colon character (“:”) used as namespace separator has been replaced by a full stop (“.”).** This makes namespaced class names in diagrams less ambiguous as compared to object boxes. Please see *Namespaces* and *Notation*.

The following list summarizes features that were incorporated in version 1.4.6:

- **Data type classes have been renamed for better consistency.** Class `SimpleType` has been renamed to `DataType`. Class `BaseType` has been renamed to `SimpleDataType`. Attribute `BaseType.DataType` has been renamed to `SimpleDateType.Base`. Adjustments to the definitions of these elements have also been made.
- **A “sha” option has been added to compact notation for associations.** Please see *Compact Style*.
- **Notational improvements have been made for temporal/subjective objects.** Please see *Phases and Perspectives*.

The following list summarizes features that were incorporated in version 1.4.5:

- **Tags have been introduced.** Tags allow you to label model elements with custom-defined strings. See *Tag*.
- **Classes representing abstract model elements have been introduced.** These are necessary for the ConML infrastructure.
- **Explicit naming attributes have been added for Cluster and Bundle.** Please see *Cluster* and *Bundle*.

The following list summarizes features that were incorporated in version 1.4.4:

- **A Version attribute has been added to the Model class.** This attribute will help with the management of models and their evolution. See *Model*.
- **Clear semantics for references between models have been defined.** The concepts of sealed and open models have been introduced. See *References between Models*.
- **The Model Management package has been removed.** This package was a vestige of older version of the metamodel, and had become unnecessary. The `Package` class is now part of the `Types` package.

The following list summarizes features that were incorporated in version 1.4.3:

- **Augmented Instances package.** The Instances package in the metamodel has been significantly augmented for better support of aspects and richer expression of instance models. See *Instances Package*.
- **Added associations for aspect management.** Explicit associations have been added between Class and TypeModelSpecification for better management of aspects. See *TypeModelSpecification* and *Class*.

The following list summarizes features that were incorporated in version 1.4.2:

- **Added attributes Name and Description to the Model class.** This will help in model management. See *Model*.
- **Attribute FullName of EnumeratedItem has been renamed as AbsoluteName.** This is in line with the use of the term “full name” for namespacing purposes. See *EnumeratedItem* and *Element Names*.
- **Rewording and reorganization of text has been done in various sections.**

The following list summarizes features that were incorporated in version 1.4.1:

- **Redefinition of semi-associations has been improved.** Redefining a semi-association in previous versions of ConML required redefining its inverse as well, which often resulted in an overspecification of the model. This limitation has been removed. See *Feature Redefinition*.

The following list summarizes features that were incorporated in version 1.4.0:

- **Strong semi-associations introduced.** The concept of “strong”, i.e. definitional semi-associations has been introduced. See *SemiAssociation*.
- **Enhanced management of aspects.** Classes can now be declared to constitute the temporal and subjective aspects of a model. Phases and perspectives (i.e. temporal and subjective object versions) of a conforming instance model are explicitly linked to instances of said classes. See *Soft Issues*.
- **Symmetric self-associations are now supported.** Explicit semantic and notational support has been added for self-associations where the only involved class plays a single role. See *Symmetric Self-Associations* and *Associations and Semi-Associations*.
- **Generalization/specialization relationships are now defined for enumerated types.** Enumerated types can now be related to each other through generalization/specialization relationships, so that extension is easily supported. See *EnumeratedType* and *Generalization of Enumerated Types*.
- **Feature redefinition is now supported.** Features can now be redefined by a class when inherited. This allows for greater flexibility and a finer control over the gradual refinement of abstractions. See *Feature Redefinition*.
- **Model views are now supported.** A new Model Views package has been introduced in order to facilitate the creation of views on models for ease of adoption. The clustering mechanism has been improved and reconceptualized as part of this new package, and the concept of model *patch* has been introduced. See *Model Views Package* and *Model Views*.
- **Model extension is explicitly supported.** Extension of models is supported through reuse, model views and other mechanisms in the metamodel. See *Type Model Extension*.

The following list summarizes features that were incorporated in version 1.3.1:

- **Improved package notation.** The notation for packages has been simplified and is now more expressive.

The following list summarizes features that were incorporated in version 1.3.0:

- **Models are now explicitly represented.** Classes `TypeModel` and `InstanceModel` have been added to represent models explicitly in ConML, thus assisting in their management.
- **Added new concepts to represent class clusters and object bundles.** A class cluster is a collection of closely related classes that usually work together. An object bundle is an instance of a class cluster.
- **The concept of packages has been introduced.** A `Package` class has been introduced in order to support the grouping of related type elements into cohesive wholes. This is useful to organize large models and also helps with model management
- **Sorted multi-instanced features are now supported.** A new attribute has been added to the `Feature` class so that multi-instanced features can be marked as being sorted (i.e. ordered).

The following list summarizes features that were incorporated in version 1.2.0:

- **Temporality modelling is now supported.** A “temporal” marker has been introduced, which can be applied to a number of metamodel elements to specify that the resulting entity has a perdurant structure and that its description in the model refers to the corresponding phase parts rather than the whole.
- **Subjectivity modelling is now supported.** A “subjective” marker has been introduced, which can be applied to a number of metamodel elements to specify that the resulting entity has a multivocal structure and that its description in the model refers to the corresponding perspective parts rather than the whole.
- **Basic uncertainty modelling is now supported.** An “unknown” keyword has been introduced to distinguish between ontic absence of information (“null” semantics) and ontic presence of information but epistemic absence of knowledge about it.
- **A compact style notation for associations has been introduced.** Some common types of associations can now be depicted in class diagrams using a compact style that does not require lines between class rectangles.
- **Higher-than-one cardinality attributes have been reformulated as true multi-valued attributes.** In ConML 1.1, attributes were said to admit higher-than-one cardinalities, but this was implemented in the metamodel as an association from `Attribute` to exactly one `Value`, and values were said to be optionally multi-content. This has been reformulated in a manner that is consistent with that of associations and links: the metamodel association from `Attribute` to `Value` has a cardinality of zero to many, and each value corresponds to a simple, single-content data element.

The following list summarizes features that were incorporated in version 1.1.0:

- **Attributes and properties with cardinality specifications are now supported.** This means that attribute values with multiple contents may be specified. This is similar to what is often called “multi-valued attributes” in database parlance. In exchange, the concept of nullability has been removed, since it becomes unnecessary. A nullable, single-content attribute can now be specified through cardinality `0..1`; a non-nullable,

single-content attribute can now be specified through cardinality 1 (which is the default). Higher cardinalities are possible to cater for multi-content attributes.

- The Tuple class has been renamed **Link**.
- **Fully-qualified names** are now defined in ConML. Although the concept of fully-qualified names has been informally used before in the context of ConML, now it is a part of the specification.
- An **ellipsis notation** (“...”) can be used to depict the fact that the contents of a particular value are not relevant or cannot be satisfactorily depicted on a diagram.
- **Localization** support has been added. This means that keywords in ConML (such as “enum” and “null”) can be translated into different languages in order to facilitate the use of language in non-English settings.