Masaryk university
Faculty of Informatics

Conceptual modeling using HIT method in *Visual Paradigm for UML* tool

Ján Demo, Marek Winkler, Michal Oškera

1 Introduction

This document is a guide how to use Entity-Relationship diagram (ERD) of *Visual Paradigm for UML (VP-UML)* for creation of conceptual models according to the HIT method. The knowledge of basic concepts of HIT method is assumed, i.e. this document is not about HIT method itself, it explains how its key concepts are represented by ERD elements in this particular modeling tool.

VP-UML is available at FI MU under the *Academic License Standard Edition*, for more information refer to `http://kore.fi.muni.cz:5080/wiki/index.php/Tools#Visual_Paradigm`. It is also possible to download and use viewer or community edition directly from the producer of *VP-UML* at `http://www.visual-paradigm.com/`.

This document covers the following topics:

- creation of a blank model,
- setting of conceptual type of a model,
- modeling entities of basic types,
- defining entities and relationships,
- modeling supertype-subtype relationships,
- other useful settings of *VP-UML*

2 Creation of a blank model

To create a blank model, a new ERD has to be created in *VP-UML*.

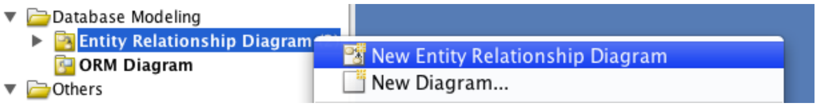


Figure 1: Creation of new ERD – model

3 Setting of conceptual type of a model

Newly created ERD may be one of the following types:

- conceptual,
- logical,
- physical.

The type of ERD can be selected under the menu item *Data Model* in the context menu of the ERD. The context menu pops up after right clicking on the canvas of the ERD (see Figure 2). It is not surprising that the right choice is *Conceptual*.

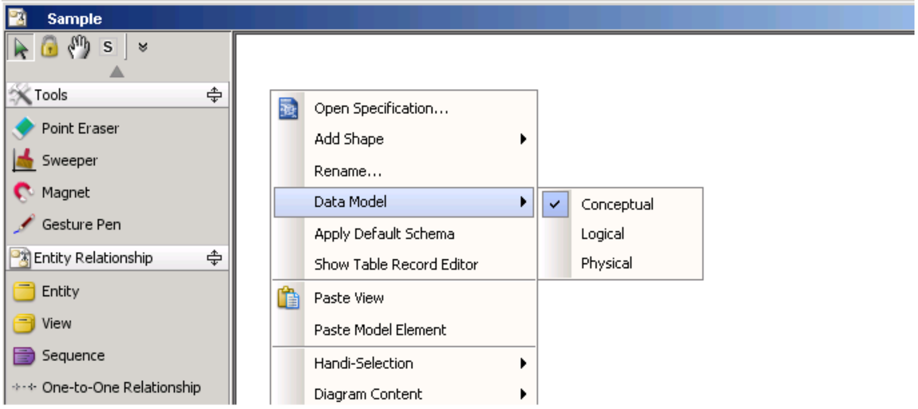


Figure 2: Setting type of a model

4 Modeling entities of basic types

The entities of basic types are modeled in *VP-UML* as follows:

- *kernel entities* are represented by the default entity representation in *VP-UML*,
- *associative entities* are distinguished by the *assoc* stereotype; the stereotype can be selected by right-clicking on an already created entity → *Stereotypes* → *assoc* (see Figure 3),

- *characteristic entities* are distinguished by *characteristic* stereotype; the selection of stereotype is analogous to the selection of *assoc* stereotype.

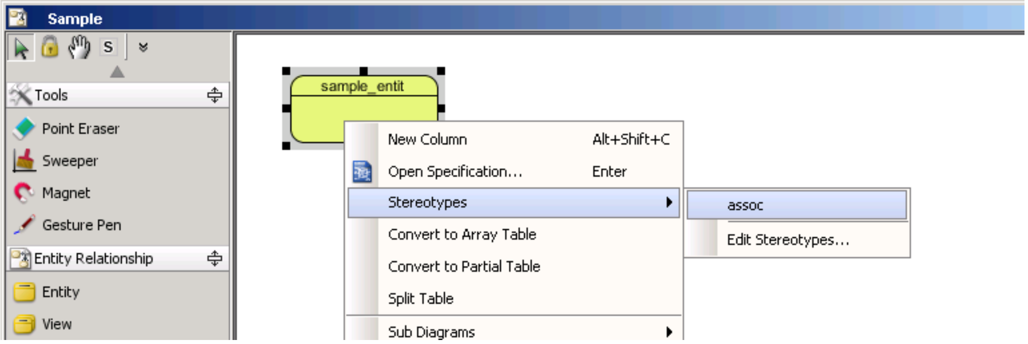


Figure 3: Setting the *assoc* stereotype of an associative entity

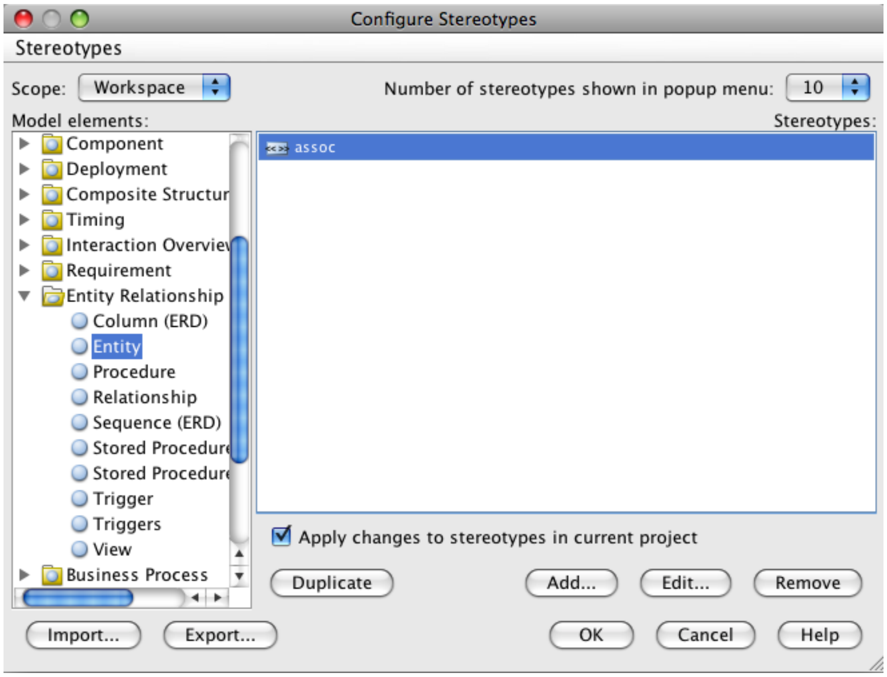
If the requested stereotype is not available in menu, it is necessary to define it:

1. to edit stereotypes click on *Tools* → *Configure Stereotypes...*,
2. in *Model elements* tree select model element that should be stereotyped, in this case it will be always *Entity Relationship* → *Entity* (see Figure 4),
3. press *Add...* button and in opened form fill in the name of the stereotype, for example *assoc*, and documentation, for example: *Denotes that the Entity is an Associative entity.* (Figure 5).

Figure 6 shows how the associative entity with the appropriate stereotype looks like.

5 Defining entities and relationships

The HIT method of conceptual modeling requires all visual model elements, i.e. both entities and relationships, to be appropriately defined. The semantics shall be written in *Documentation* text field in *General* tab within specification dialog of the corresponding model element. This can be achieved in two ways:



Scope: Workspace

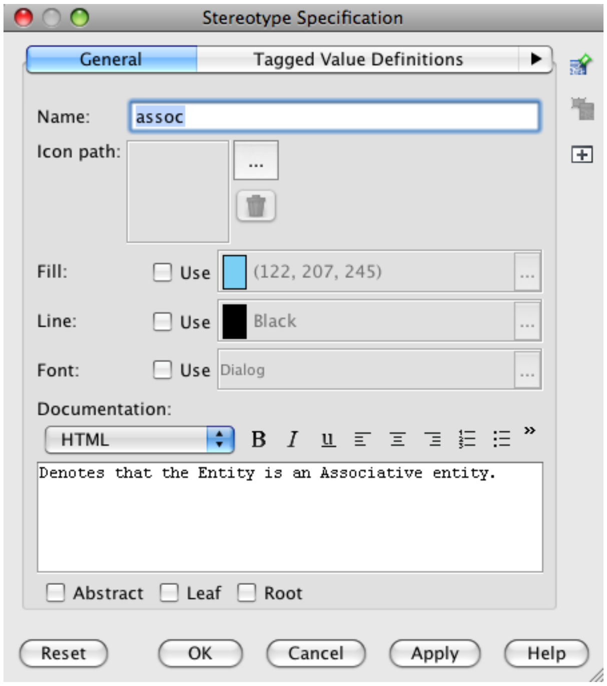
Number of stereotypes shown in popup menu: 10

☑ Apply changes to stereotypes in current project

Figure 4: Editing stereotypes

- right-click on entity/relationship to be defined → *Open Specification* (Enter) and write semantics in *Documentation* multiline text field,
- it may be more comfortable to use the toolbar window *Documentation* in some cases. This window is permanently visible and its content is bounded to the currently selected model element. The window can be made visible
  - through menu *View* → *Panes* → *Documentation*,
  - or key shortcut `Ctrl+Shift+U`.

When writing semantics of a model element, for example *An object of type (#Customer) is every . . .* , it is recommended to use function *Add Model Element* (see figure 7) for writing the name of the element, in this example *Customer*. The advantage is that if the model element is renamed the new name is automatically updated in all definitions where the renamed element was referred.



Stereotype Specification

Name: assoc

Icon path:

Fill: □ Use (122, 207, 245)

Line: □ Use Black

Font: □ Use Dialog

Documentation:

Denotes that the Entity is an Associative entity.

□ Abstract □ Leaf □ Root

Figure 5: Editing stereotype *assoc*

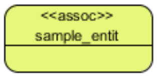


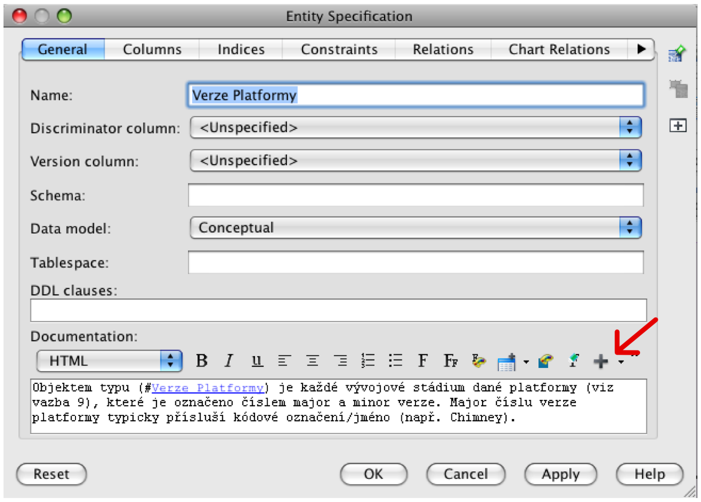
Figure 6: Stereotyped associative entity

6 Modeling supertype-subtype relationship

The *supertype/subtype* (or *general/specialized*) relationship can be modeled in *VP-UML* using the following steps:

1. create 1:1 relationship in direction from intended supertype to intended subtype,

2. in the context menu of the created relationship check that this is an identifying relationship (→ *Identifying*),



Name: Verze Platformy

Discriminator column: <Unspecified>

Version column: <Unspecified>

Schema:

Data model: Conceptual

Tablespace:

DDL clauses:

Documentation:

Objektem typu (#Verze Platformy) je každé vývojové stádium dané platformy (viz vazba 9), které je označeno číslem major a minor verze. Major číslu verze platformy typicky přísluší kódové označení/jméno (např. Chimney).

Figure 7: Using *Add Model Element* function to write name of a model element

3. in the same menu check that this is a subtype kind of relationship (→ *Subtype*).

Figure 8 shows an example of a supertype/subtype relationship, where the *Entity B* is a subtype (or a specialization) of the *Entity A*.

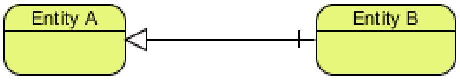


Figure 8: Example of supertype/subtype notation in *VP-UML*

Unfortunately, *VP-UML* (version 8.1) supports modeling of *generalization sets* in *Class diagrams* only[1]. Therefore, you may model supertype/subtype relationships as separate connections without grouping them into generalization sets.

[1] see http://www.visual-paradigm.com/support/documents/vpumluserguide/94/299/7190_drawingclass.html

7 Other useful settings of *VP-UML*

7.1 Spell checking

If the model elements and their definitions have not been written in English, it is recommended to switch off *Spell checking* for better readability:

- *Tools* → *Options*,
- in *Options* window click on the left side on the *Spell Checking* item,
- uncheck the *Enable Spell Checking* check box.