3 - Meta-Modeling

3.1 - Definition

A model is an abstract representation of reality. It is used to schematically represent part of the concepts of a program, in order to allow for a better understanding of the way the architecture works.

To understand what is contained in a model as well as the information that it should represent, it is necessary to first agree on the definition of its content. This is the work that is referred to when using the term "meta-modeling".

A meta-model is thus the model of a model.

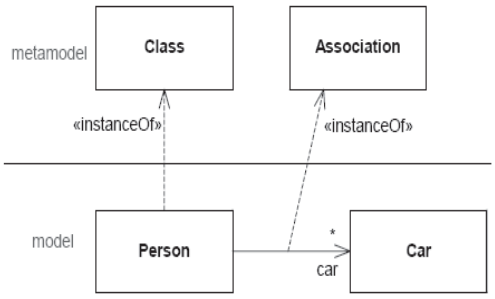
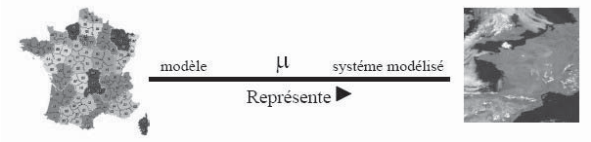


Illustration 1: Example of a model and of its metamodel

To illustrate these concepts, it is possible to build an imaged version of this approach by using the example of a road map. A road map is a model: it is a representation of reality, a schematic representation of the road network, which is real. Its purpose is to allow someone to better understand how it works, so that he or she won't get lost.



The legend of a road map is crucial for its understanding. The legend can be considered as the meta-model for this map. It defines the elements that are represented on the map. Without the legend, it is impossible to understand the map. Of course, we seldom look at the legend because we are used to reading maps of this type, all having the same legend. They all use the same semantics.

In order to allow a certain consistency between maps, all legends are conceived on the same model: one color = one characteristic. It thus becomes simple to move from one map to another. The same method is used in designing software, to share the same formal description of a meta-model, through the notion of MOF meta-meta-models.

3.2 - MOF

Meta-modeling is a vast domain, there are countless ways to represent concepts in an abstract manner. In the same way, it is possible to make an infinite variety of maps presenting different information (city, roads, relief, ...) on the country of France, for example.

Meta-modeling technologies have been formalized by the Object Management Group or OMG. This standardization is well recognized today; it is mainly known as the “MOF” (Meta Object Facility) standard.

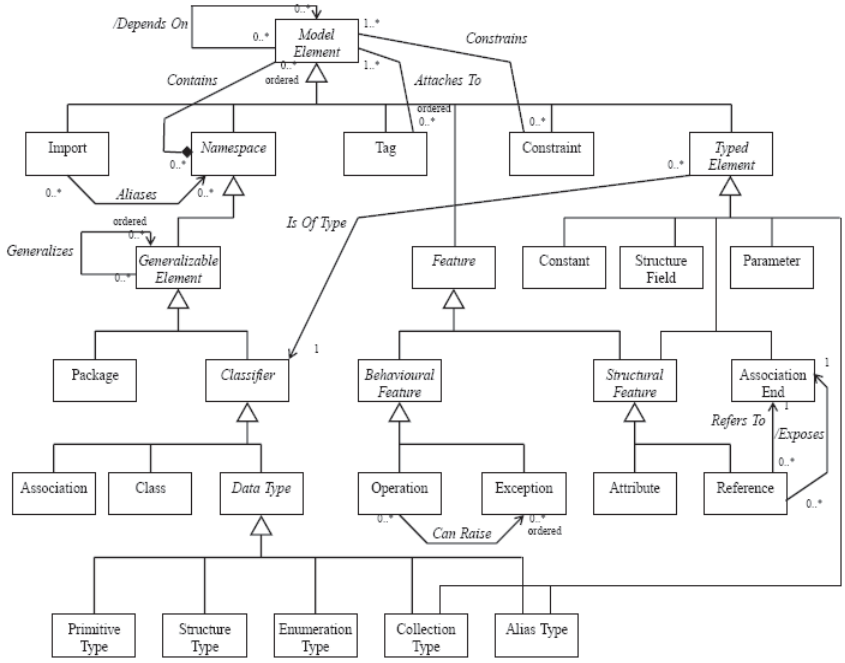


Illustration 2: Overview of MOF 1.4 concepts

MOF allows description of all types of meta-models, using a limited vocabulary. That is why the OMG defines it among the category of meta-meta-models, or into the “meta-model repository”.

3.3 - XMI

The XMI format has been developed in order to allow the serialization of models and meta-models in a physical format. As its name suggests, XMI uses the XML syntax, a tag-based syntax.

XMI is thus employed to permit interoperability in the exchange of models between Engineering workshop software, reverse engineering and rationalization tools such as Acceleo.

Finally, it is important to note that a given meta-model version can be serialized in several different XMI versions.

3.4 - EMF

EMF (Eclipse Modeling Framework) was developed to simplify the loading, handling and storage of models within the Eclipse environment.

EMF is not specific to a meta-model, thus it may handle all types of models.

It is based on a standard of meta-model description named “Ecore”. This standard is a subset of the EMOF standard (Essential MOF), which in turn is a subset of MOF2.

3.5 - UML

Even if people don't always realize it, the most commonly used meta-model is UML. Indeed, UML is the modeling language that is the most widely used today. UML has its own meta-model. For example, UML allows the representation of class diagrams with attributes, methods and relationships. It is the UML meta-model that defines the fact that a class may possess attributes and relationships…

UML is fitted to represent a good number of concepts, mostly in an informal way, yet it is not necessarily the best language to represent everything. For that reason, we may have to define new meta-models allowing a more formal vision of reality. As a concrete example, the electronics industry models complex systems such as microprocessors using its own standards.

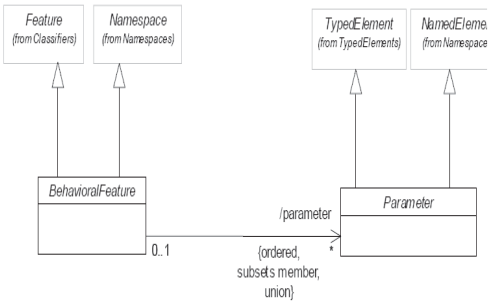
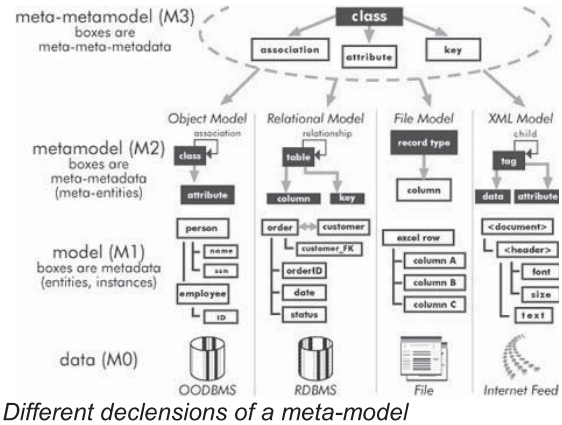


Illustration 3: Part of the UML 2.0 Meta-model

3.6 - Other Meta-Models

UML is often used as an example because it is the best known meta-model, but it is not the only one. Some others include Merise, SADT, DTD, SPEM, ...

Remember: Model ≠ UML



Different declensions of a meta-model

UML is not a standard that is well adapted to accomplish all types of tasks. It is possible to extend its range by using UML profiles and the concept of stereotypes, but this practice reaches its limits as soon as we get too far from the original vocation of the UML elements. Dedicated meta-models can therefore be created to meet specific needs. These are called DSL (Domain Specific Language).

As DSLs become more widely used, the development of new meta-models should become better understood and more common.

To develop a meta-model, the following steps are followed:

- Defining the concepts that should be modeled and the best way of doing it
- Representing these concepts in the form of MOF or EMF diagrams

The difficulties inherent to meta-modeling are:

- Difficulty in figuring out the best way to represent a concept. This task necessitates a vast experience and hindsight from the domain being treated.
- Difficulty in mastering the concepts being modeled. Each new abstract concept is a new element to be learned for the readers and users of the resulting models.
- Modelers for Meta-models are seldom and poorly understood today.

Acceleo has been designed to work at the meta-level. It accepts any type of model as its input, even those based on proprietary meta-models.