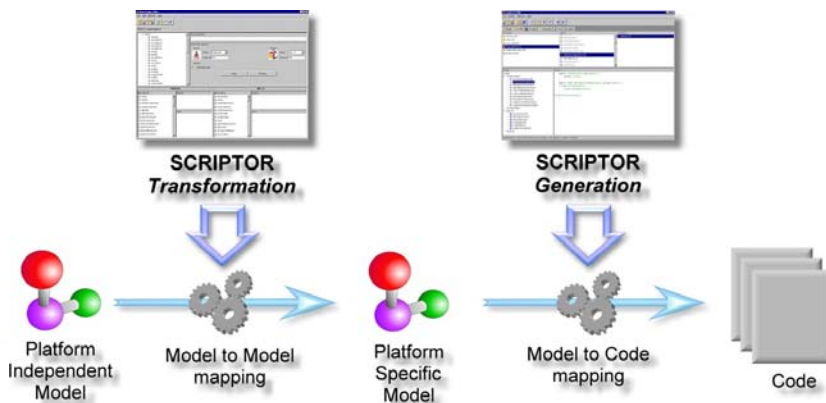


SCRIPTOR is a tool suite that implements OMG MDA™-type approach. The suite is composed of two major tools:

- **SCRIPTOR-Transformation**: to build model-to-model translators
- **SCRIPTOR-Generation**: to build model-to-code translators.



SCRIPTOR supports OMG standards :

- The MOF™ (Meta Object Facility) to define metamodels.
- XMI™ (XML Metadata Interchange) to exchange models with CASE tools.
- SCRIPTOR-Generation supports UML™ Profiles

SCRIPTOR and the MDA approach

SCRIPTOR-Transformation : a model translation processor

SCRIPTOR-Transformation performs translations on a source model to create a target model. The mapping is specified by rules based on the source and target meta-models.

SCRIPTOR-Transformation environment provides a graphical editor to specify the rules, and a code generator to generate the MOF components used to manipulate source and target models.

SCRIPTOR-Transformation has been used in various domains, like :

- Transformations between system engineering tools (Statemate™ and Core™) and UML models
- Transformations between graphical user interface models and UML models
- Transformations of process models to workflow engines.

SCRIPTOR-Generation : a template-based code generator

SCRIPTOR-Generation is a generator development environment.

SCRIPTOR-Generation is open to any source model file format, and can be connected to major CASE tools (Rose™, Rhapsody™, Objectteering™, Together™, Argo, and any XMI compliant tool).

The generation rules are specified with :

- Templates (WYSIWYG scripts), which let you enter the text to be generated instead of programming the code.
- Java™ scripts for more complex operations. The use of Java avoids learning any proprietary language, and lets you reuse your java components for building generation solutions.

SCRIPTOR-Generation is used to build specific code generators based on your architecture and framework. It provides continuity between analysis and design, and supports iterative cycles, by preserving user defined code fragments between successive generations.