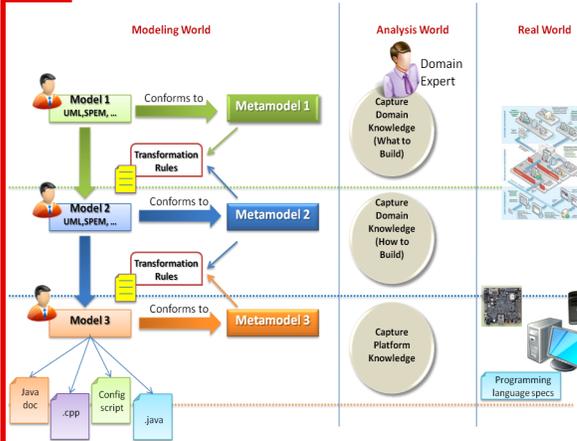


## What is Model Driven Engineering?



Model Driven Engineering (MDE) is an approach that represents the software development lifecycle as **modeling** and **model transformation** activities. A **model** is an abstract and partial representation of a real system or process that changes depending on its objective. MDE goes beyond the pure development activities. It encompasses other model-based tasks of the complete software engineering process such as the model-based evolution of the system, the model-driven reverse engineering of a legacy system, and business process modeling. It aims at promoting models to **drive** the development process not to be just an important artifact in the process.

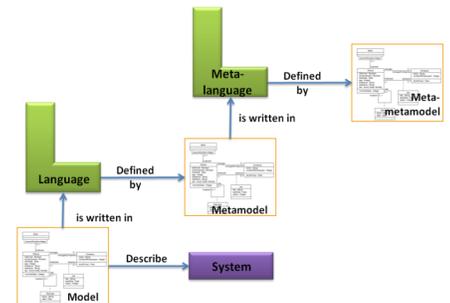
Using MDE moves the focus of the developers from pure coding to analysis so the problem can be solved independent of the technical details of the platform. Thanks to transformations, models can be transformed from one level of abstraction and platform knowledge level to another, inserting new information each step along the way.

## MDE Adoption Basic Concepts

Adopting MDE can be done using current modeling languages such as Unified Modeling Language (UML). Extending UML using profiles is a common technique to add specific concepts not already there. However, not all domains can be modeled this way. For example, in case of creating financial or healthcare applications, it would be easier to use domain concepts to define the solution and automatically generate the rest of the artifacts including documentation and code. Here comes the role of **Domain Specific Modeling (DSM)** and **Domain Specific Languages (DSL)** as key approaches to adopt MDE as they provide more expressiveness through visually expressing domain elements creating your own language. To achieve this, it is required to define the language, define the targets to transform to, define the transformations, and provide the defined framework to developers to create applications in a much **faster** manner and with **higher quality**.

The first step to create a DSL is to capture the domain knowledge in a **metamodel**. A metamodel defines the abstract syntax of the DSL by describing the domain definition and scope, domain terminology, and domain concepts and their relationships. In order to define a modeling language by specifying its metamodel, we will need another language to define the metamodel itself. This is typically called the **metameta-modeling** language. As standardized by the **Open Management Group (OMG)**, one unique meta-meta-model exists known as the **Meta-Object Facility (MOF)**. It is some kind of "top level ontology" from which all other metamodels are derived.

A full DSL might be defined using more than one metamodel each describing a certain level of abstraction or a different viewpoint for the given domain. As a result, writing **transformations** is one of the main activities while defining



DSLs. Model transformation is the automatic generation of a target model from a source model according to a transformation definition. Transformation definition is a set of transformation rules that together describe how a model in the source language (metamodel) can be transformed into a model in the target language (metamodel). Transformations can be either model to model (**M2M**) or model to text (**M2T**) such as documentation, code, or configuration scripts.

## MDE in SECC

So far it cannot be claimed that MDE can be fully adopted in organizations and that there is a tooling support to adopt MDE throughout the whole software development lifecycle. However, some tools provide a good start for metamodeling, transformation, and model verification. It is important to select aspects of the system to use MDE for such as repetitive and error prone sections, or critical components that require reliability such as security.

**Software Engineering Competence Center** provides specialized support to adopt state-of-the-art modeling and model transformation techniques in Egyptian ICT companies helping them to build their capacities by learning MDE basic concepts and getting hands-on experience on the underlying technologies through training courses and specialized consultation support. The provided services help the companies to **focus on problem domain** rather than coding problems, to **increase productivity**, and to **reduce time-to-market**. For more information visit <http://www.secc.org.eg/RECOCAPE/>.