

Domain Specific Models and Languages

In software development a domain-specific language (DSL) is a programming language or specification language dedicated to a particular problem domain and is not intended to be able to solve problems outside it. A DSL can be either a visual language, like the languages used by the Eclipse Modeling Framework, or textual languages. A sound language description contains an abstract syntax, one or more concrete syntax descriptions, mappings between abstract and concrete syntax, and a description of the semantics. The abstract syntax of a language is often defined using a metamodel. The semantics can also be defined using a metamodel, but in most cases in practice the semantics aren't explicitly defined, they have to be derived from the runtime behavior.

In model-driven engineering many examples of domain-specific languages may be found like OCL, a language for decorating models with assertions or QVT, a domain specific transformation language. However languages like UML are typically general purpose modeling languages. What is radically new is the idea of creating your own DSL for your own project. Domain specific languages have important design goals that contrast with those of general-purpose languages:

- domain specific languages are less comprehensive.
- domain specific languages are much more expressive in their domain.
- domain specific languages should exhibit minimum redundancy.

A model specified using a DSL is called a Domain-Specific Model (DSM). A complex system is usually described using multiple DSMs specified in different DSLs. These models refer to each other and have to be combined when executing them. Because complex systems ask for a lot of DSMs to model them, it is important to structure the modeling space, the different parts of the system as we can see on Figure 1.

Domain-specific modeling.

The main goal of domain-specific modeling is to raise the level of abstraction by specifying the solution directly using domain concepts. The final product (and maybe several intermediate artifacts, as well) are generated based upon these high-level specifications. It also allows the stakeholders and domain experts to concentrate to the domain only. Domain-specific languages (DSLs) are built in order to capture domain semantics. A very common (but not the only) way of defining DSLs is metamodeling. The previously mentioned Web application design methods contain notations that can be used for describing a model of a Web application so they can be considered as DSLs for Web applications hence.