MAP-I PhD Dissertation Proposal

Declarative model-transformation engine

João Pascoal Faria (jpf@fe.up.pt), INESC Porto/FEUP, 3/12/2010

Motivation

The vision of model-driven development (MDD) is to raise models as the primary software development artifact, and generate automatically code in multiple platforms from those models, usually via intermediate model-to-model transformations followed by final model-to-code transformations.

The transformation engines should deal with the technical aspects of the platforms, while the models should concentrate only on the concepts and features of the application domain.

In order to augment the power and flexibility of the MDD approach, the current trend is to specify the abstract syntax of models (and even code) via meta-models, independently of their concrete syntax, and define the transformations as transformations among meta-models. This allows applying the same transformation rules and engines for many different concrete input and output syntaxes.

However, the languages and tools to accomplish this vision are still immature.

Goals

The main goals of this research work are to improve existing declarative model-transformation languages and develop new model-transformation engines that are capable of executing model-to-model and model-to-code transformations based on declarative transformation rules.

The engine receives as input a model that obeys a defined meta-model (e.g., a domain model) and a set of declarative transformation rules, and generates a new model (e.g., a user interface model) or final code.

The work builds upon previous work on the automatic generation of user interface models and prototypes from rigorous domain and use case models. In that work, a declarative model-transformation language was proposed, but an engine capable of processing it was not developed. In the proposed notation, transformation rules are specified declaratively by structural diagrams enriched with constraints, with a left-hand side that should match existing meta-model instances, and a right–hand side to be generated. An example is shown in Figure 1.

The development of a transformation engine poses significant challenges, because of the potential need to solve complex constraints.

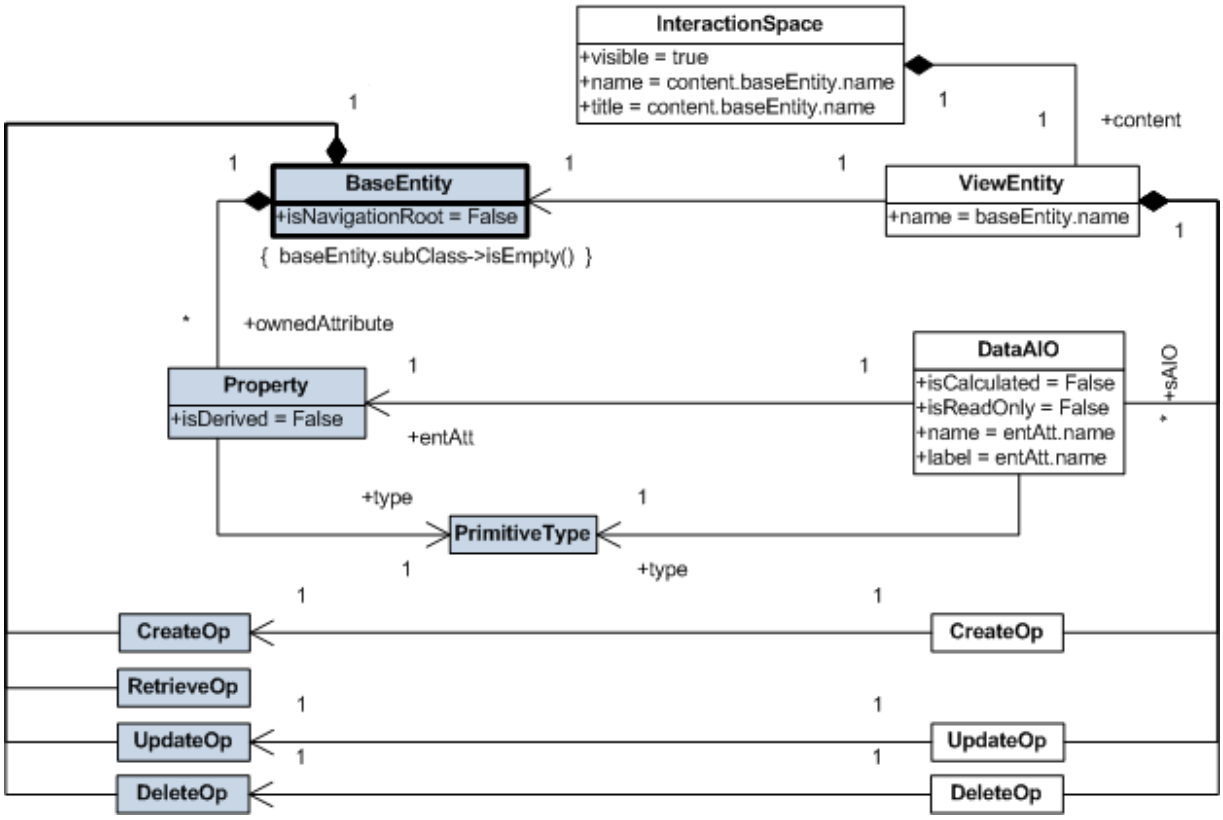

Figure 1 - Example of a declarative model-transformation rule. The left-hand side is shaded.