UML-Based Domain-Specific Modeling using MetaModelAgent

Thomas Wiman

Adocus AB, Stockholm, Sweden
thomas.wiman@adocus.com

Abstract: MetaModelAgent is a UML-modeling tool extension that simplifies the definition and encourages the use of UML-based domain-specific languages (DSL). MetaModelAgent make use of user-defined meta-models to adapt the tools GUI with a number of additional DSL-specific features. This is a presentation of the concept, the meta-model notation used, and the key features of MetaModelAgent.

Keywords: UML, Metamodeling, Domain-Specific Modeling, Model Validation.

1 Introduction

UML is a rich and expressive language, addressing a broad range of needs within the software engineering industry. To be able to use UML efficiently within a specific domain, DSL-extensions, extending the syntax and semantics of UML have to be created. UML allows for extensions in the form of profiles, but there is no simple way to express restrictions on models such as naming conventions, model structure, or diagram contents, in short, to specify a complete DSL in a way which can be monitored by the modeling tool.

MetaModelAgent provides just this functionality. MetaModelAgent makes use of the concept of metamodels to define any UML-based DSL and enables the resulting DSL in an existing UML-modeling tool by providing a set of context sensitive GUI additions dynamically adapted to the metamodel.

2 Defining a UML-Based DSL using a Metamodel

MetaModelAgent is intended for cases where an organization wishes to perform domain-specific modeling using UML. MetaModelAgent makes use of the fact that a UML-based DSL can be represented in a meta-model, using UML notation and semantics. To be able to express all constraints and guidelines needed for a UML-based DSL, a specific metamodel notation has been developed. The metamodel notation allows the expression of many different types of constraint, e.g. naming, permissible element types in packages and diagrams, and model structure. Each constraint is assigned a severity.

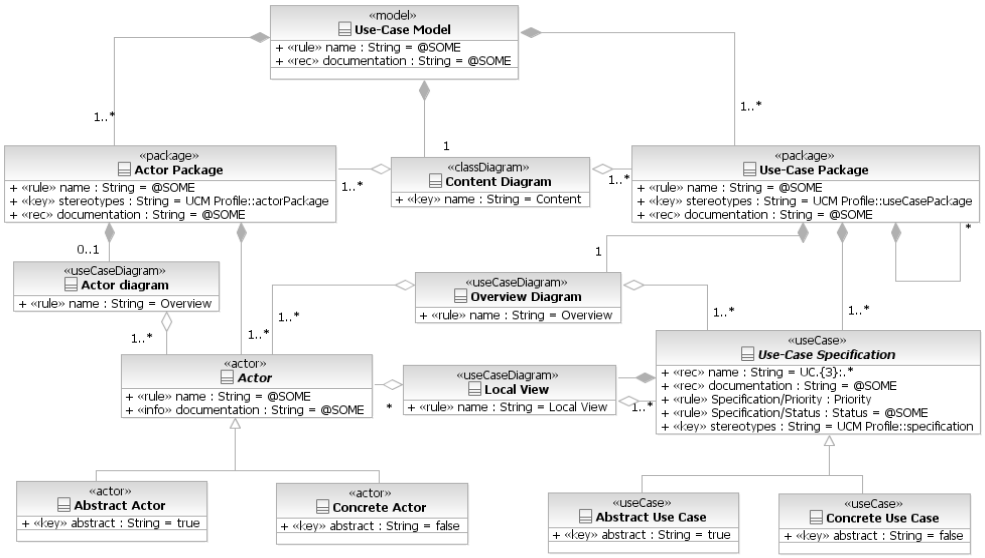

Fig. 1. Sample metamodel defining UML constraints for simple Use-Case modeling

3 The Concept of MetaModelAgent

A domain-specific metamodel essentially replaces guidelines expressed as documents and any DSL-specific tool extensions, other than profiles.

The modeling tool, extended with MetaModelAgent and a domain-specific metamodel, form a domain-specific modeling tool. MetaModelAgent uses the metamodel to provide additional functionality to the UML tool by continuously compare models being developed towards the metamodel and enabling a context-sensitive metamodel adapted GUI to the end user.

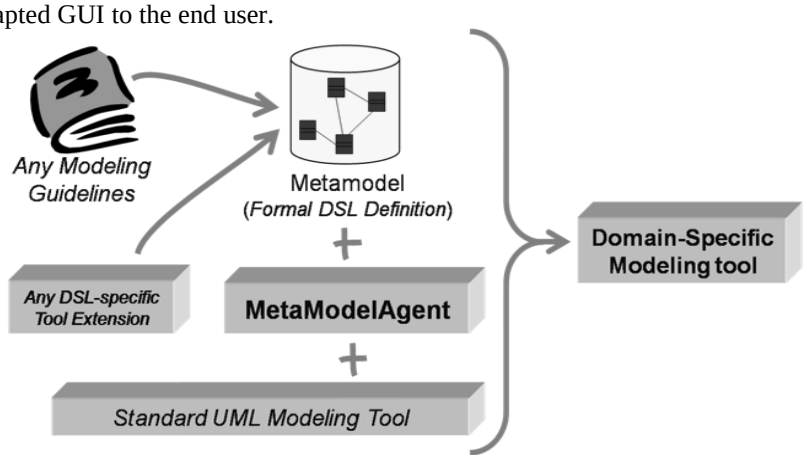

Fig. 2. The concept of MetaModelAgent

4 MetaModelAgent Key Features

MetaModelAgent adds the following key features to the underlying modeling tool:

- Live validation; e.g. quality assurance of DSL-specific constraints. Any problem in the model is presented with extensive explanation of the root cause and suggested actions that resolves the problem. Live validation makes it much easier to ensure that a model fulfills the DSL.
- Quick-fix; whenever it is possible to automatically determine which actions that will resolve a problem, quick-fixes are available to automatically correct the problem, comparable to spell-checking in a word processor.
- A metamodel-based context sensitive Add-wizard that only allows elements that are valid in the current position in the model to be created. The wizard encourages the user to fill in the mandatory and significant element properties with correct

values. The wizard also let the user create any mandatory nested elements in the same operation. The add-wizard will speed up the creation of models significantly.

- A metamodel-based Property View that let the user focus on the significant properties only in a comprehensive view, with active support for entering the correct values. Any invalid values are immediately highlighted.
- An integrated Guidance View presenting textual guidance on any element or element property selected and any problem identified. The guidance presented will be a combination of manually entered documentation in the metamodel combined with automatically generated text from the metamodel.
- An integrated HTML-generator for generating readable modeling guidelines based on the metamodel. Generated guidelines can be uploaded on an intranet for common access within the organization.

5 Benefits of using MetaModelAgent

MetaModelAgent makes UML-based DSLs and modeling guidelines enforceable. Those who are responsible for UML-modeling in an organization or in a project can very easily and cost efficiently create one or several domain-specific metamodels. This task is preferable done in one or several workshops where relevant stakeholders participate. The formalism of a metamodel ensures the consistency, completeness and unambiguously of the defined DSL.

The end users will receive a DSL-enabled modeling tool that will make their modeling tasks more efficient and less error prone than before. The presence of the additional MetaModelAgent functionality can easily be turned on and off making it suitable for inexperienced users in need of active support as well as more experienced users who only want support to validate the models prior to delivery.

6 Availability

MetaModelAgent is developed as an Eclipse-extension mainly based on the open source EMF, UML2, GEF and GMF components of Eclipse. MetaModelAgent is general available for IBM Rational Architect, will soon available for Papyrus UML and is easily extended to any Eclipse-based UML-modeling tool. MetaModelAgent is currently in use in several international organizations.