

Over the last years, models turned out to be one cornerstone for the development of complex software systems, leading to intensive teamwork and continuous evolution of models. Both issues are handled by optimistic model versioning systems. The urgent demand for optimistic version control support for models triggered intensive research and first approaches for model merging were realized. Those approaches primarily focus on the comparison of concurrently evolved versions of a model and the detection of conflicting changes. The visualization of detected merge problems has experienced little attention.

We employ UML profiles, the language inherent extension mechanism of UML and propose the **Versioning Profile**

, an approach to visualize changes and conflicts within UML modeling environments without requiring tool extensions. As a result, modelers may resolve conflicts in the graphical concrete syntax using their familiar UML editors. The screenshot below depicts a tentative, automatically merged

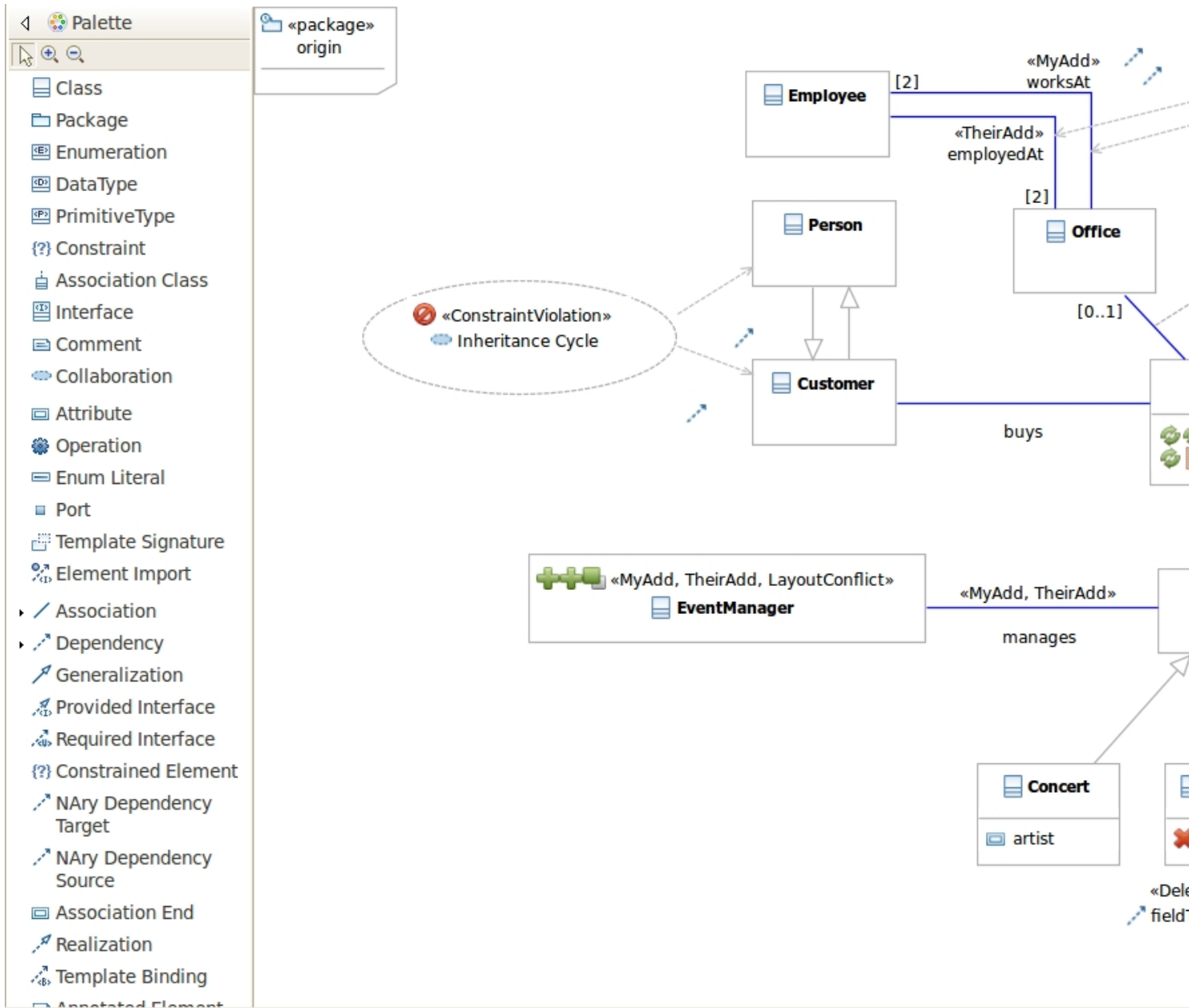
Conflict Diagram

. The conflict diagram is enriched with UML stereotypes of the versioning profile holding change and conflict information and thus, acts as kick-start for manually merging two conflicting model.

Generating Conflict Diagrams for Evolving UML Models

Written by Administrator

Monday, 28 March 2011 08:55 - Last Updated Monday, 28 March 2011 12:51



Properties

Operation

Core	Property	Value
Tagged Values	- My Update	
	Change Element	Attribute name : String in <Operation> purch
	Old Value	buy
	State	APPLIED
	Updated Feature	name : String
	Updated Value	purchase
	User	<Actor> Sally

Legend

Changes

+ Add X Delete Update Composite Operation

Conflicts

Update/Update Violation Layout

The Versioning Profile

The versioning profile depicted below is a revised version of our previously presented [conflict profile](#). It

consists of two parts for defining

Changes

and

Conflicts

.

Changes. The change part of the versioning profile provides stereotypes for each kind of change, i.e., for atomic changes (add, update, delete) and for composite changes, like refactorings (see [EMF Modeling Operations](#) for the definition of composite changes). To provide provenance information, each Change has tagged values to make the responsible user explicit. Additionally, as the stereotypes are not only used for mere visualization purposes, but also for supporting the merge process in terms of dedicated tooling, a status information indicating whether the change is already applied, is introduced. To complement tooling related information, each change may be traced back to the corresponding change element. An AtomicChange,

i.e.,

Add

,

Delete

, or

Update

, may be applied to any concrete UML element, i.e., Class, Generalization, Property, etc., and

thus, is defined to extend the UML metaclass

element

. As updates are changes to existing elements, they have additionally tagged values for pointing

to the affected feature of the changed element including its old and new value. Composite

changes like refactorings, incorporate a set of indivisible atomic changes. To highlight this fact,

a new UML

collaboration

is introduced and annotated with the stereotype

CompositeChange

. The collaboration connects all model elements concerned by the composite change via UML

relationships

. Each specific kind of change stereotype is finally defined in the form of

MyChange

and

TheirChange

to indicate which changes were originally performed by the other user and which changes were applied by the user in charge of merging.

Generating Conflict Diagrams for Evolving UML Models

Written by Administrator

Monday, 28 March 2011 08:55 - Last Updated Monday, 28 March 2011 12:51

Conflicts. The conflict part of the versioning profile defines stereotypes for different conflict types. Conflicts are either due to OverlappingChanges, i.e., parallel add/add, update/update, or update/delete operations are performed on the same model element or shape, or due to violations.

Vio

lations

occur, if the merged version violates some constraint (metamodel constraint or user defined OCL constraint restricting the UML diagram) or a composite operation is not applicable because of a violated precondition.

Update/Update

,

Delete/Update

, and

LayoutConflict

stereotypes extend the UML metaclass

element

, as these conflicts result from two atomic changes on the same model element. In contrast,

Add/Add

collaborations

to hint at the involved changes.

ConstraintViolation

further state violated constraints. In case of a

OperationContractViolation

, the UML

relationships

interlinking the involved elements to the UML

collaboration

, are annotated with stereotypes (inspired from graph transformation theory) indicating how the

contract is violated by the model element. The stereotypes

ChangeUse

and

DeleteUse

are applied on model elements already existing in the original model, which are involved in a

composite operation and changed or deleted by the other user, respectively.

AddForbid

indicates the addition of a new model element which invalidates the precondition of a composite

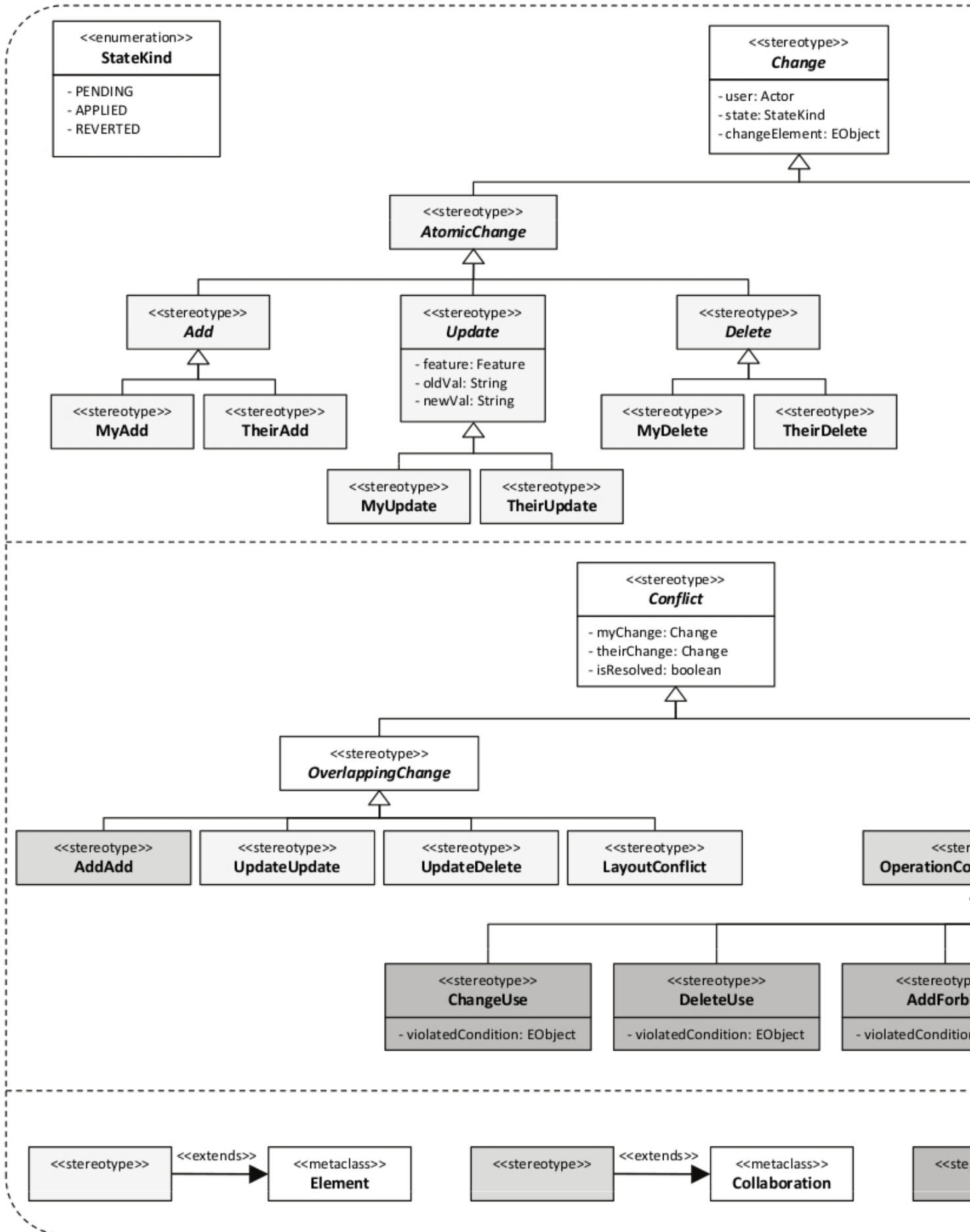
operation. Finally, all conflict stereotypes refer via tagged values to the underlying change

stereotypes, what makes understanding and reproducing the conflicts possible.

Generating Conflict Diagrams for Evolving UML Models

Written by Administrator

Monday, 28 March 2011 08:55 - Last Updated Monday, 28 March 2011 12:51



Generating Conflict Diagrams for Evolving UML Models

Written by Administrator

Monday, 28 March 2011 08:55 - Last Updated Monday, 28 March 2011 12:51

Download

You can download an eclipse-based version of the [Versioning Profile](#) as plug-in. Copy the plug-in into the plugins directory of your

[eclipse modeling edition](#)

and apply the profile to arbitrary UML models. Then, model elements may be annotated with change and conflict stereotypes.