

# Strelka - A Visual Rule Modeling Tool

## 1 Introduction

The languages used in the communication between domain analysts and domain experts for analyzing and documenting system requirements should not be 'technical', but should allow visual and/or natural-language-like vocabulary and rule expressions that can be understood by domain experts without extensive technical training.

The UML offers a visual language for specifying vocabularies. Some rule types, for example, integrity constraints (invariants) and derivation rules can be represented in UML models by means of the Object Constraint Language (OCL). The OCL is a formal language, which is difficult to understand for people without a technical background.

In order to simplify rule modeling, the REVERSE Working Group I1 has developed

- a UML-based Rule Modeling Language (URML), which extends UML class models by adding rules, and
- Strelka, a tool for making URML models.

Strelka is implemented as a plug-in for the *Fujaba Tool Suite*, which is an open source UML case tool<sup>1</sup>.

## 2 URML – A UML-Based Metamodel and a Visual Notation for Rule Modeling

URML supports modeling of derivation rules, production rules and reaction rules. A rule is represented graphically as a circle with a rule identifier. Incoming arrows represent rule conditions or triggering events, outgoing arrows represent rule conclusions or produced actions.

### Language Elements

**Condition arrows** refer to a *conditioned model element*, which is a classifier such as a class or an association.

It may come with a *filter expression* selecting instances from the extension of the condition classifier and with an explicit object variable (or object variable tuple, in the case of an association) ranging over the resulting instance collection.

**Negated condition arrows** are crossed at their origin; they denote a negated condition which has to be conjoined with one or more positive condition arrows such that its variables are covered by them.

**Derivation rules** are represented graphically as a circle with an internal label "DR" and a rule identifier attached to it. Incoming arrows represent conditions, outgoing arrows represent conclusions.

**Conclusion arrows** also refer to a classifier model element. Its meaning is to state that the predicate represented by the conclusion classifier applies to any instance that satisfies all rule conditions.

**Production rules** are represented graphically as a circle with an internal label "PR" and a rule identifier attached to it. Incoming arrows represent conditions, outgoing arrows with a double arrowhead represent actions.

---

<sup>1</sup>Fujaba Tool Suite <http://www.fujaba.de>

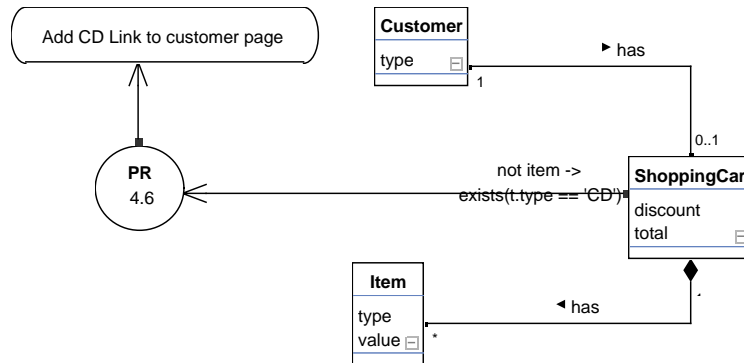


Figure 1: Production rule, create action: If a customer has no items of type CD in her shopping cart, then add a CD page link to the customer page.

**Action arrows** refer either to a class (in the case of a create, delete, assign or invoke action) or to an activity.

**Reaction rules** are represented graphically as a circle with an internal label "RR" and a rule identifier attached to it. There are two kinds of incoming arrows (condition arrows and event arrows) and two kinds of outgoing arrows (action arrows and postcondition arrows, both having a double arrowhead).

**Event arrows** may refer to a class.

A more detailed description of URML is available on the website of the Working Group I1<sup>2</sup>.

### 3 Functionality of Strelka

To create a rule, a user should create a new class diagram first. All rule modeling operations are accessible in the pop-up menu. Rule-specific items are context-dependent and items availability depends on what is currently selected on the diagram.

**To create a rule**, a user should right-click and select "New derivation rule" or new "New production rule".

**To create rule condition arrow**, a user should select condition classifier (class or association) and a rule circle. The selection is done by holding Ctrl button while selecting items with the mouse. After selection is complete, the user should right-click on a classifier and select "Create new condition".

**To add a filter to a condition**, a user should right-click on the condition arrow and select "Edit filter". The filter dialog will pop-up to add/delete/modify filter expression. Currently filters are represented as strings and syntax validation and consistency check are not supported.

**To create a rule conclusion arrow**, a user should select a conclusion classifier and a rule, right click on the classifier and select appropriate conclusion type (classification, association or attribution).

**A property value statement** is used to specify the value of the derived attribute in the attribution conclusion. To edit a conclusion property value statement, a user should right-click on the conclusion arrow and select "Edit property value statement".

**To create a production rule action**, a user should select a production rule and a classifier for the action (class or association), then right-click on the classifier and select action type in the popup menu. Visually, action types are distinguished by the capital character near the arrow head. **A** is used for the assign action, **C** for create, **D** for delete and **I** for the invoke action.

<sup>2</sup>REVERSE Working Group I1 <http://www.reverse.net/I1>

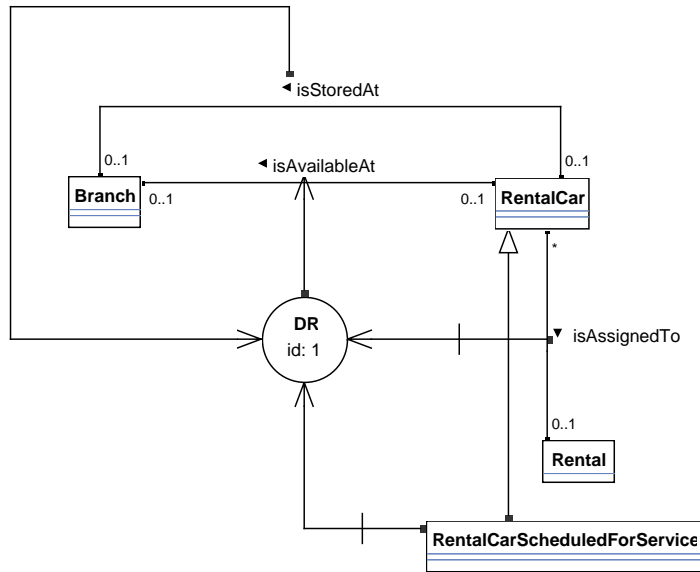


Figure 2: If a rental car is stored at a branch, is not assigned to a rental and is not scheduled for service, then the rental car is available at the branch.

The model can be saved using the standard Fujaba saving function available in the File menu. A model can be exported in any of the graphics formats PDF, SVG, JPG, PNG and GIF.

Two examples of derivation and production rules, created with Strela, are depicted on Figure 1 and 2.

A user manual for Strelka is available on the website of the Working Group I1<sup>3</sup>.

## 4 Future Work

The current work of I1 is focused on the serialization of rule models in the R2ML format<sup>4</sup>. More information about the functionality of the tool, download instructions and sample rule models are available from the Working Group I1 website <http://www.rewerse.net/I1> (go to the Projects section).

<sup>3</sup>REWERSE Working Group I1 <http://www.rewerse.net/I1>

<sup>4</sup>G. Wagner, A.Giurca, S. Lukichev (2005). R2ML: A General Approach for Marking up Rules, Dagstuhl Seminar Proceedings 05371, in F. Bry, F. Fages, M. Marchiori, H. Ohlbach (Eds.) Principles and Practices of Semantic Web Reasoning, <http://drops.dagstuhl.de/opus/volltexte/2006/479/>