

An iStar 2.0 Editor Based on the Eclipse Modeling Framework

Martin Liebenberg, Kerstin Roßmaier,
Gerhard Lakemeyer

iStar'17 Workshop
13.06.2017



Motivation for a New Editor

Problems with the old editor:

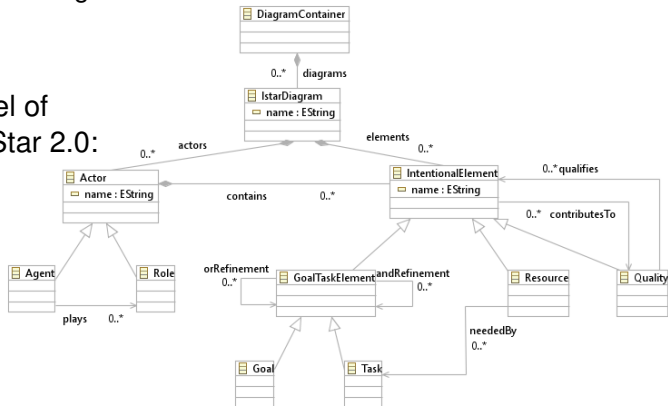
- we used an seven-year-old OpenOME editor so far
- difficult to maintain by now
- we needed new functionalities for editing scenarios for a multi-agent simulation
- the last updates of the OpenOME editor are from 2011

The requirements for a new editor:

- use the Eclipse Modeling Framework (EMF) for editor generation
- as little coding as possible for better maintainability
- we need to change features of the editor easily

Searching for a Suitable Framework

- we wanted to use EMF and the Graphical Editing Framework (GEF) for editor generation
- we examined the following frameworks which provide that:
 - ▶ Graphical Modeling Framework
 - ▶ Graphiti
 - ▶ Sirius
- our Ecore model of a fragment of iStar 2.0:



Brief Comparison of the Frameworks

All three frameworks use Ecore models for the editor generation.

- Graphical Modeling Framework (GMF)
 - ▶ many models to maintain which depend on each other
 - ▶ changing one of these models can be a complex task
- Graphiti
 - ▶ fewer models than GMF and therefore more flexible
 - ▶ but much more coding needed than in the GMF
- Sirius
 - ▶ only one model for representing the Ecore metamodel in the editor
 - ▶ editor generation without code generation
 - ▶ on-the-fly changes of the editor possible
 - ▶ changes of the editor's model appear easier than in GMF or
 - ▶ small limitations in the visualisation

We Went for Sirius

- Editor generation with XML-like Domain-Specific Language (DSL)
- easy to maintain in Sirius' editors

The screenshot displays the Sirius Specification Editor interface. The top pane shows a project tree with the following structure:

- platform:/resource/istarmodel1.design/description/istarmodel1.o.design
 - istarmodel1
 - istar
 - IstarDiagram
 - Validation Actor Validation
 - Default
 - ResourceNode**
 - Workspace Image /istarmodel1.design/icons/Resource3.svg
 - OrRefinementEdge
 - AndRefinementEdge
 - QualifiesEdge
 - ContributionEdge
 - ProvidesEdge

The bottom pane shows the Properties view for the selected **ResourceNode**. The properties are:

Category	Property	Value
General	Id*	ResourceNode
General	Label	ResourceNode
General	Domain Class*	istarmodel1.Resource
General	Semantic Cand...s Expression	feature:elements

Easy Validation in Sirius

- easy access to the Ecore metamodel via the Aceleo Query Language (AQL)
- validation in DSL easily possible

The screenshot displays the Sirius Specification Editor interface. The top pane shows a project tree with the following structure:

- platform:/resource/istarmodel1.design/description/istarmodel1.odesign
 - istarmodel1
 - istar
 - IstarDiagram
 - Validation Actor Validation
 - Semantic Validation Rule AND_REFINEMENT_ERROR
 - Semantic Validation Rule REFINEMENT_ERROR
 - ? Audit aql:self.andRefinement->size()=0 or self.orRefinement->size()=0
 - Semantic Validation Rule SELFLOOP_ERROR
 - ? Audit service.hasNoSelfloop()
 - Semantic Validation Rule CIRCLE_WARNING
 - ? Audit service.isCirclefree()
 - Default

The bottom pane shows the configuration for the selected **Semantic Validation Rule REFINEMENT_ERROR**. The configuration is as follows:

Property	Value
Id*	REFINEMENT_ERROR
Label*	REFINEMENT_ERROR
Level*	Error
Target Class*	istarmodel1.GoalTaskElement
Message:	aql:self.name + ' has both AND and OR refinements'

The Resulting Editor

The screenshot displays the Sirius Eclipse Platform interface for editing an iStar diagram. The main workspace shows a diagram titled "Water usage diagram" with various nodes and relationships. A "Task order tanker water" is selected, and its properties are shown in the Properties view below. The Properties view includes a table with the following data:

Task order tanker water	
Semantic	And Refinement
Style	Contributes To
Appearance	Name
	Or Refinement

The Properties view also shows a list of relationships: "Quality supply reliability" and "order tanker water".

- Drawback: wrong AND-refinement and neededBy relation arrowheads due to uncustomisable link decorators so far

Summary

- We needed a new, easily maintainable iStar 2.0 editor.
- We compared three existing EMF frameworks using Ecore metamodels.
- We chose the Sirius framework for implementing an editor.
- Sirius is easy to maintain on-the-fly with DSL and AQL.
- Small disadvantage: the visualisation of iStar relations