# Resilience in Sirius Editors: Understanding the Impact of Meta-Model Changes

Pierantonio Alfonso

Juri Di Rocco

Di Ruscio Davide

Hrishikesh Narayanankutty



Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica Università degli Studi dell'Aquila

- 1. Introduction
- 2. MDE Ecosystem
- 3. Metamodel Evolution
- 4. Sirius
- 5. Running Example
- 6. Change impact classification
- 7. How is Sirius affected?
- 8. Examples
- 9. Conclusion

# 'Introduction

- In Model-Driven Engineering the employment of metamodels is pervasive
- They are used to formally describe a wide range of artifacts, including models, transformations, concrete syntaxes, and **editors**
- In essence, metamodels are at the core of any modeling ecosystem and their management is therefore key to success







# Metamodel Evolution

Similarly to any software component metamodels are expected to evolve during their lifecycle:

- Dependencies can emerge at different times during the metamodel life-cycle
- Metamodels can **change**:
  - to satisfy unforeseen requirements;
  - to better represent concepts of the considered domain,
  - ...
- When metamodels are changed the already existing artifacts might be **compromised**





















Sirius\* is an Eclipse project that enables the development of **modeling workbench** by leveraging well-established technologies.

A modeling workbench is composed of a set of Eclipse editors which allow the users to create, edit and visualize EMF models.

The editors are defined by a **model** which defines the complete structure of the modeling workbench.

\* http://www.eclipse.org/sirius/



# -X Sirius - Mapping model

Sirius allows the conceptual separation between the metamodel and the corresponding representation(s) by means of Viewpoint Specification Models (VSMs) wich consistently specifies the structure, appearance, and behavior of an editor according to the domain model and it is given in the .odesign format.

- the <u>representation description</u> specifies how models are represented, i.e., in the form of diagrams, sequence diagrams, tables, and cross-tables, or trees;
- the <u>representation extensions</u> that enable the customization of diagrams, i.e., each representation can be endowed with extra functions;
- the <u>validation rules</u> are given for validating purposes and for suggesting quick-fixes while editing, and finally
- <u>Java extensions</u> define additional functionalities to be associated with editor commands or gestures.



# Running example



શ mindmup.odesign 🛱						
< Sirius Specification Editor						
🔻 👔 platfor	m:/resource/org.mindmup.oc	desig	gn/description/mindmup.odesign			
▼ 🐸 mindmup						
🔻 🌾 mindmupViewPoint						
🔻 🏯 mindmap						
🔻 🔲 Default						
	🔻 📮 TopicNode					
	Square light_gray					
	RelationEdge					
	Edge Style dash_dot					
	Section Palette					
	Node Creation Topic	:				
	Node Creation Val	riabl	e container			
	Container View Va	ariad	de containerview			
	▼ ▶ Begin					
	Change Context var:container.topics					
	▼ <sup>™</sup> Edge Creation Relation	ion	mindingpopic			
	Source Edge Creat	tion	Variable source			
	a Target Edge Creat	tion \	Variable target			
	Source Edge View	/ Crea	ation Variable sourceView			
	arget Edge View	Сгеа	tion Variable targetView			
	Begin					
) e	org.mindmup.odesign.Servio	ces				
Propertie	s ដ 🕄 Problems		F1	$\nabla$		
Create I	nstance mindmun"Topic					
	instance initialitapiiropie					
General	Reference Name*:	?	relations			
1	Гуре Name:	?	mindmup::Topic			
Variable Name: (?) instance						
					_	







# Classified features

Features	Feature classification
Create a Viewpoint Specification Project	Basic
Define a Diagram	Basic
Add a Node to the Layer	Basic
Add a Relation Based Edge	Basic
Add a Palette for edition tools	Basic
Create a Section	Basic
Add a Node Creation Tool	Basic
Define the actions performed by the Create Node tool	Basic
Add a Edge Creation Tool	Advanced



### Uncovered feature

Features	Feature classification
Precondition*	Advanced
Reconnect Edge Tool	Advanced
Advanced Delete Element Tool*	Advanced
Direct Edit Label Tool	Advanced
Style Customization*	Advanced
Layers*	Advanced
Filters*	Advanced
Validation Rules*	Advanced
Quick Fixes*	Advanced
Object-Centered Diagram (Container, Sub Nodes, Border Node, Edge between containers, Double-click, Table, Java Services*)	Advanced



# Change impact classification: editor

**Editor attributes**. The resilience of a Sirius-based editor is explained by its ability to survive meta-model changes and by the degree its expected behavior is still exposed. We characterize the editor resilience by means of the following attributes:

- **non-breaking**: an editor is non-breaking when despite the changes to the meta-model, it is still possible to open and use the editor;
- **complete**: an editor is complete when after a meta-model change, each element in the meta-model has a graphical counterpart within the editor, e.g., as part of the tooling palette;
- **valid**: an editor is valid when it exposes a correct behavior despite the metamodel modifications, i.e., it is still possible to instantiate a model conforming to the new version of the meta-model.



### Change impact classification: mapping model

**Mapping model attributes**: A meta-model change can severely affect the editor specification given by the mapping model stored in the .odesign file. The mapping model resilience is characterized by the following attributes:

- **complete**: a mapping model is complete, if after a meta-model change it still contains the graphical denotations for all model elements in the meta-model;
- **valid**: a mapping model is valid, if it consistently defines the graphical denotation for the elements in the meta-model, despite the meta-model changes 8 ;
- **resolvable**: a mapping model is resolvable, if after meta-model modifications its validity and completeness can be restored by means of an automated procedure.



17

# How is Sirius affected?

We apply a catalog of atomic changes [1] on running example.

Chango	E	Mapping model				
Change	non breaking	complete	valid	complete	valid	resolvable
1.1 Add empty, optional, concrete class	YES	NO	YES	NO	YES	NO
1.2 Add empty, mandatory, concrete class	YES	NO	NO	NO	YES	NO
2 Add empty, abstract class	YES	YES	YES	YES	YES	-
3 Add specialization	NO	-	-	NO	NO	-
4 Delete concrete metaclass	NO	-	-	YES	NO	YES
5 Rename metaclass	YES	-	-	YES	YES	NO
6.1 Add attribute	YES	YES	YES	YES	YES	-
6.2 Add reference	YES	NO*	NO*	NO	YES	NO
7 Delete property	YES	NO	NO	NO	NO	YES

[1] Davide Di Ruscio, Ralf Laemmel and Alfonso Pierantonio, Automated co-evolution of GMF editor models, in: 3rd International Conference on Software Language Engineering (SLE 2010), Eindhoven.



# How is Sirius affected?

We apply a catalog of atomic changes [1] on running example.

Chango	Editor			Mapping model			
Change	non breaking	complete	valid	complete	valid	resolvable	
8.1 Rename Attribute	YES	NO	NO	YES	NO	YES	
8.2 Rename Reference	YES	NO	NO	YES	NO	YES	
9 Move property	YES	NO*	NO*	YES	NO	NO	
10 Pull property	YES	YES	YES	YES	YES	-	
11.1 Change attribute type	YES	YES	YES	YES	YES	-	
11.2 Change reference type	NO	-	-	YES	NO	YES	

[1] Davide Di Ruscio, Ralf Laemmel and Alfonso Pierantonio, Automated co-evolution of GMF editor models, in: 3rd International Conference on Software Language Engineering (SLE 2010), Eindhoven.



If we consider the running example...

### 1.2 - Add empty, mandatory, concrete class



🔹 disim 🗋

### 1.2 - Add empty, mandatory, concrete class

	Mapping model				
non breaking complete valid		valid	complete	valid	resolvable
YES	NO	NO	NO	YES	NO





#### 6.1 Add attribute





#### 6.1 Add attribute

Editor			Mapping model				
non breaking	complete	valid	complete	valid	resolvable		
YES	YES	YES	YES	YES	-		
	■ ■ III III III - ÖÖ- theater - ÖÖ- art ! - ÖÖ- music	Y I I I I I I I I I I I I I I I I I I I	B I Palette Palette → → → → → Palette → → → → → → Palette → → → → → → → → Palette	↓ ★ </td <td></td>			
	🖹 Problems 📮 Console 🕙	Error Log 🔲 Prope	erties 🛛 📑 🎽				
	Main	es ring: ⑦ spoi	rt				
	Appearance description	n : EString: 🕐					



#### 8.1 Rename attribute





#### 8.1 Rename attribute

Editor			Mapping model				
non breaking	complete	valid	complete	valid	resolvable		
YES	NO	NO	YES	NO	YES		
	■ • <	Image: Second	B Palette Palette Palette Palette Topic Relation Relation re as name re as name re as name re as name re as name re as name				
	<ul> <li>Feature: labelExpre</li> <li>Feature: labelExpre</li> <li>Feature: labelExpre</li> </ul>	ession Unknown feat ession Unknown feat	ure name name ure name name				



# Conclusion

The work in this paper is preliminary to the analysis of more extended and **composite refactoring patterns** and to the definition of (semi) automated procedures for the consistency restoration.

Besides more extended changes, future work includes also the analysis of changes operated on meta-model OCL invariants and how they affect editor resilience.



Thank you for your attention!

**Questions?** 

Github Repository: <u>http://github.com/MDEGroup/sirius-</u> coevolution

Juri Di Rocco