GMF graphical editor tutorial

How to create a FSM graphical editor with GMF?

Marie Gouyette

Abstract

This tutorial shows how to create a GMF editor for Fsm models. This tutorial consitutes only an example that illustrate the global fsm example. For more informations and tutorials about GMF please refer to the GMF documentation and its tutorials (ht-tp://www.eclipse.org/modeling/gmf/ (Eclipse GMF Project)).

Published Build date: 3-November-2010 19/07/2006

Preface.	v
Chapter 1. Introduction	1
Chapter 2. Getting started with GMF	2
2.1. Mechanisms of GMF	2
2.2. Create a new GMF project	3
2.3. Generation Process	7
2.3.1. Domain model	7
2.3.2. Domain Gen Model	7
2.3.3. Graphical Definition Model	11
2.3.4. Tooling Definition Model	17
2.3.5. Mapping Model	19
2.3.6. Diagram Editor Gen Model	25
2.4. Use the generated editor	26

List of Figures

2.1. GMF process	. 3
2.2. Create a new GMF project	. 4
2.3. Give a name to the new GMF project	. 5
2.4. Show Dashboard view	6
2.5. DashBoard at the beginning	. 7
2.6. Choose name for the genmodel file	. 8
2.7. Select ecore model	. 9
2.8. Load ecore model	10
2.9. Generate EMF Java code	10
2.10	11
2.11. Select elements from domain	12
2.12. fsm.gmfgraph at the beginning	13
2.13. Edit Rounded Rectangle 's name	14
2.14. Edit Label 's name	15
2.15. Rounded rectangle	16
2.16. Set Target Decoration	17
2.17. Tooling definition	18
2.18. fsmStatic.gmftool	19
2.19. wizard creation .gmfmap	20
2.20. fsmStatic.gmfmap	20
2.21. Node and tool	21
2.22. Select feature and add it	22
2.23. Add diagram label into Feature Mapping	23
2.24. Edit the LinkMapping Transition	24
2.25. fsm.gmfmap	25
2.26. fsmStatic.gmfmap	25
2.27. fsmStatic.gmfmap	26
2.28. Create Fsm Diagram model	27
2.29. Edit Fsm Diagram model	27

CHAPTER

Preface

Kermeta is a Domain Specific Language dedicated to metamodel engineering. It fills the gap let by MOF which defines only the structure of meta-models, by adding a way to specify static semantic (similar to OCL) and dynamic semantic (using operational semantic in the operation of the metamodel). Kermeta uses the object-oriented paradigm like Java or Eiffel. This document complete the metamodel overview showing how to create a graphical editor for fsm.

Important

Kermeta is an evolving software and despite that we put a lot of attention to this document, it may contain errors (more likely in the code samples). If you find any error or have some information that improves this document, please send it to us using the bug tracker in the forge: ht-tp://gforge.inria.fr/tracker/?group_id=32 or using the developer mailing list (kermeta-developers@lists.gforge.inria.fr) Last check: v0.3.1

Tip

The most update version of this document is available on line from http://www.kermeta.org .

CHAPTER 1

Introduction

This tutorial presents how create a GMF Editor for the FSM metamodel.

Important

This tutorial consitutes only an example that illustrate the global fsm example. For more informations and tutorials about GMF please refer to the GMF documentation and its tutorials.

GMF (Graphical Modelling Framework) is an Eclipse framework that permits to create a specific graphical editor of models conform to a meta model following the MVC architecture (Model View Controller). It is based on the Framework EMF (Eclipse Modelling Framework)in charge of domain model and on the Framework GEF (Graphical Editing Framework) charged on the controllers. On this tutorial we use EMF 2.5.0 and GMF 2.2.1. For more informations about GMF please refers to the following links :

- 1. http://www.eclipse.org/modeling/gmf/ (Eclipse GMF Project)
- 2. http://wiki.eclipse.org/index.php/GMF_Tutorial A tutorial about GMF
- 3. http://wiki.eclipse.org/index.php/GMF_Tutorial_BPMN Another tutorial about GMF
- 4. http://www.ibm.com/developerworks/opensource/library/os-ecl-gmf/ (Quickly getting started)

CHAPTER 2

Getting started with GMF

In this tutorial, we suppose that GMF is already installed into Eclipse. We will copy and use the ecore metamodel fsm.ecore stored into fr.irisa.triskell.samples.fsm.demoAspect/metamodels/ after creating a new general project. The solution of this tutorial is available clicking on File -> New -> Example -> Kermeta Samples -> FSM Solutions. So, you can retrieve the following plugins :

- 1. fr.irisa.triskell.kermeta.samples.fsm.gmf
- 2. fr.irisa.triskell.kermeta.samples.fsm.gmf.edit
- 3. fr.irisa.triskell.kermeta.samples.fsm.gmf.editor
- 4. fr.irisa.triskell.kermeta.samples.fsm.gmf.diagram
- 5. fr.irisa.triskell.kermeta.samples.fsm.gmf.tests

Even if these plugins do not have the same name as the plugins you will created it is the same diagram editor .

2.1. Mechanisms of GMF

GMF uses six files to create a generated graphical editor for instances of a given metamode like shown in the following DashBoard (all the steps are finished). For more information about DashBoard View please refer to the section Create a new GMF project.



Figure 2.1. GMF process

As you can see, the generation of a GMF graphical editor contains six steps :

- 1. **Domain model :** the metamodel we want to use to create the graphical editor. For this metamodel, you have the choice between several kinds of metamodels : Annotated Java code , Ecore model, Rose class model, UML model or XML Schema). In this tutorial we will use the ecore metamodel of FSM.
- 2. **Domain Gen Model (.genmodel) :** this file is used to generate the domain model code with EMF (it is the EMF file genModel)
- 3. Graphical Def Model (.gmfgraph) : this file is used to define the graphical elements for your domain model
- 4. Tooling Def Model (.gmftool) : this file is used to define the palette of tools that you can use in the graphical editor
- 5. **Mapping Model (.gmfmap) :** this file links the domain model, the graphical model (.gmfgraph) and the tooling model (.gmftool)
- 6. **Diagram Editor Gen Model (.gmfgen) :** this final file us used to generate the GMF graphical editor in addition to the EMF code generated by the .genmodel file

Tip

You can generate two kind of graphical editor with GMF : a plugin graphical editor integrated with Eclipse or as an RCP (Rich Client Platform) application which consists in an autonomous application. To generate a RCP application, just click into RCP on the DashBoard. In this tutorial we will create a GMF editor as an Eclipse plugin.

2.2. Create a new GMF project

To start with, click on File-> New->Other-> Graphical Modeling Framework -> New GMF Project.

New	
Select a wizard	
Creates new blank GMF project	
Wizards:	
type filter text	
🗁 Eclipse Modeling Framework	*
🗁 Ecore Tools	
Example EMF Model Creation Wizards	
GMF-Xpand	E
Graphical Modeling Framework	
New GME Project	
Reconcile Graphical Definition Model	
Reconcile Tooling Definition Model	
Simple Graphical Definition Model	
Simple Tooling Definition Model	
🔁 Default	-
	Cancel

Figure 2.2. Create a new GMF project

Then, you can give to it the name org.kermeta.fsm.gmf like shown below.

Image: Sew GMF Project	
New GMF Project	
Creates new blank GMF project	
Project name: org.kermeta.fsm.gmf	
✓ Use default location	
Location: C:\Users\mgouyett\Marie\Work\workspacePourKermeta\org	Browse
Choose file system: default]
Over the second seco	Cancel

Figure 2.3. Give a name to the new GMF project

Select the DashBoard view, it will be useful to use GMF. This dashboard resumes all the necessary steps to create an editor with GMF and you can navigate between these steps with it.



Figure 2.4. Show Dashboard view





Figure 2.5. DashBoard at the beginning

Now, we will follow all the steps presented into the last section.

2.3. Generation Process

This section details more all the steps. To start with, copy the file fsm.ecore stored into fr.irisa.triskell.kermeta.samples.fsm.demoAspects/metamodels/fsm.ecore into org.kermeta.fsm.gmf/model. In all this section we use the Dashboard view.

2.3.1. Domain model

On the DashBoard view, in the case Domain model click on Select and choose the fsm.ecore from org.kermeta.fsm.gmf. Then, click on the Derive at the left to create the EMF GenModel.

2.3.2. Domain Gen Model

The following wizard appears to choose the name of the .genmodel file. (You can make appears it clicking on Select in the Domain Gen Model Case).

New EMF Generator Model	- • •
EMF Generator Model	ii-
Create the generator model	- 6
Enter or select the parent folder:	
org.kermeta.fsm.gmf	
⊿	*
🗁 .settings	
	E
src	
•	ł.
File name: fsm.genmodel	
Advanced >>	
Seck Next > Finish	Cancel

Figure 2.6. Choose name for the genmodel file

Click on Next. In the following wizard select ecore model.

Image: Severator Model Image: Severator Model Image: Severator Model Image: Severator Model Image: Severator Severat	
Select a Model Importer Create the Ecore model based on other Ecore or EMOF models	
Model Importers:	
 Annotated Java Ecore model Rose class model UML model XML Schema 	
Over the second seco	Cancel

Figure 2.7. Select ecore model

Click on Next. For the next wizard do not forget to click on load.

😂 New EMF Generator Model	
Ecore Import Specify one or more '.ecore' or '.emof' URIs a	nd try to load them
Model URIs:	Browse File System Browse Workspace
platform:/resource/org.kermeta.fsm.gmf/m	odel/fsm.ecore
? < Back Net	ext > Finish Cancel

Figure 2.8. Load ecore model

Then click on Next and Finish. The file fsmStatic.genmodel appears on the project and the DashBoard. Open it and click on the arrow near Fsm. Right click on Fsm -> Generate All. It generates the EMF Java code for the domain model.

🔋 fsm.genmodel	X	
皆 Fsm		
🖶 Fsm		

Figure 2.9. Generate EMF Java code

For the next steps you can choose between begin with Graphical Def Model or Tooling Def Model. We start

with Graphical Def Model.

2.3.3. Graphical Definition Model

Click in Derive at the top of the Domain Model case. Like for GenModel you can select the name. Then, select the root metaclass of your metamodel, in your case FSM.

New		
Domain Model		
Select file with ecore	e domain model.	
Model URI:	Browse File System Browse Workspace	Find In Workspace
platform:/resource	/org.kermeta.fsm.gmf/model/fsm.ecore	Load
Diagram Element		
FSM		
State		
Transition		
2	< Back Next > Finish	Cancel
J		cuncer

Figure 2.10.

In this tutorial we want simply to create a graphical editor for States with their name and Transition. Select the elements like in the following figure :



Figure 2.11. Select elements from domain

The resulted file fsm.gmfgraph looks like this :



Figure 2.12. fsm.gmfgraph at the beginning

Now, we want to do changes on fsm.gmfgraph file. To start with we want to replace the state rectangle by a rounded state rectangle. Start with suppress the Rectangle State figure (Right click on it -> Delete). Click on FigureDescriptor StateFigure -> New Child -> Rounded Rectangle. Then click on the created Rectangle and use the Properties view to edit its name as StateFigure.

Тір

If you do not see the Properties View go to Window-> Show View -> General -> Properties.

🗛 fsm.gmfgraph 🖾

- platform:/resource/org.kermeta.fsm.gmf/model/fsm.gmfgraph
 - Canvas fsm
 - Figure Gallery Default
 - Figure Descriptor StateFigure
 - Rounded Rectangle StateFigure
 - Child Access getFigureNull
 - Figure Descriptor TransitionFigure
 - Polyline Connection TransitionFigure
 - Node State (StateFigure)
 - Connection Transition
 - Diagram Label StateName

🖹 Problems	@ Javadoc	😟 Declaration	🕅 SVN Repositories	🔄 Console
Property		Value		
Corner He	Corner Height			
Corner Wi	idth		L1 8	
Descriptor	r		Figure Descriptor StateFigure	
Fill	Fill		🖙 true	
Line Kind		LINE_SOLID		
Line Widt	Line Width			
Name 🖙 StateFigure				
Outline			🖙 true	
Xor Fill		🖙 false		
Xor Outline 🌆 false		🖙 false		

Figure 2.13. Edit Rounded Rectangle 's name

Then right click on Rounded Rectangle StateFigure and choose Flow Layout. Then right click again and choose Label. Add its name StateFigureLabel in the Properties View.

🗛 *fsm.gmfgraph 🛛			
platform:/resource/org.kermeta.fsm	.gmf/model/fsm.gmfgraph		
Canvas fsm			
Figure Gallery Default			
Figure Descriptor StateFig	ure		
Rounded Rectangle St	ateFigure		
Flow Layout false			
♦ Label StateFigureL	abel		
Child Access getFigure	eNull		
Figure Descriptor Transition	onFigure		
Polyline Connection TransitionFigure			
Node State (StateFigure)			
Connection Transition			
Diagram Label StateName			
🛃 Problems @ Javadoc 😥 Declaration	🔞 SVN Repositories 🗐 Console		
Property	Value		
Descriptor			
Name 🖙 StateFigureLabel			
Text			

Figure 2.14. Edit Label 's name

A Diagram Label StateName appears on the gmfgraph file. Now, Click on the Child Access of the Rounded Rectangle and set its figure property to Label StateFigureLabel. Then right click on Child Access and choose Refresh. You can add a foreground on your Rounded Rectangle. For this right click on Rounded Rectangle StateFigure -> Foreground Color Constant Color and edit the value to blue. The file fsm.gmfgraph should look like this :



Figure 2.15. Rounded rectangle

Then we want to customize the connection and add an open arrow on its end. Start with adding a Polyline-Decoration (right click on PolylineConnection TransitionFigure -> New Child -> Polyline Decoration). Name it targetDecoration. We need to add three Template Point to draw the arrow. Right click on Polyline Target targetDecoration -> New Child -> Template Point (at the bottom of the contextual menu , use the arrow at the bottom to see all this menu) to add one of them. Then edit them respectively to (-1,1), (0,0) and (-1,-1). Now edit Polyline Connection TransitionFigure and set targetDecoration with Polyline Decoration targetDecoration. The file fsm.gmfgraph should look like this :

👧 fsm.gmfgraph 🖾

- A platform:/resource/org.kermeta.fsm.gmf/model/fsm.gmfgraph
 - 🔺 🔶 Canvas fsm
 - Figure Gallery Default
 - Figure Descriptor StateFigure
 - A & Rounded Rectangle StateFigure
 - Flow Layout false
 - Foreground: blue
 - Label StateFigureLabel
 - Child Access getFigureStateFigureLabel
 - Figure Descriptor TransitionFigure
 - Polyline Connection TransitionFigure
 - A + Polygon Decoration targetDecoration
 - (-1,1)

- (-1,-1)
- Node State (StateFigure)
- Connection Transition
- Diagram Label StateName

Figure 2.16. Set Target Decoration

Now, we create the tooling model.

2.3.4. Tooling Definition Model

Click on Derive at the bottom of the Domain model case. Like for the graphical model choose the file name and the root metaclass (FSM). In this palette we want only to use State and Transition, so select the elements like in the following figure :



Figure 2.17. Tooling definition

The file fsm.gmftool looks like this :



Figure 2.18. fsmStatic.gmftool

These images can be customized. For more informations please refer to the GMF documentation.

2.3.5. Mapping Model

The mapping model (.gmfmap) permits to link .genmodel, .gmfgraph and .gmftool files. In the DashBoard click on Combine. Like for Graphical and Tooling model choose a name and select the root metaclass (FSM). We want to have a very simple editor with simply State and one Transition connector. So, in the last wizard, remove owningFSM and incomingTransition in order to obtain the following wizard:

Create GMFMap model Mapping		
Map domain model elements		
Nodes State (State; ownedState)	As node < As link> Remove Restore	ngTransition)
Structure Element: Transition Containment: outgoingTransition Target Feature: source	Edit	Change
Visual Diagram Element: Transition	Constraints Specialization: Initializer:	Change
?	< Back Next > F	inish Cancel

Figure 2.19. wizard creation .gmfmap

The fsm.gmfmap looks like this

fsm.gmfgraph	🏓 fsm.gmftool	🔂 fsm.gmfmap 🛛			
🍋 Resource Set					
Platform:/resou	ırce/org.kermeta.fsr	m.gmf/model/fsm.gmfmap			
Mapping					
Top Noc	Top Node Reference < ownedState:State/State>				
Node Mapping <state state=""></state>					
Link Mapping <transition{transition.source:state} transition=""></transition{transition.source:state}>					
Canvas Mapping					
📄 platform:/resou	urce/org.kermeta.fsr	m.gmf/model/fsm.ecore			
platform:/resou	ırce/org.kermeta.fsr	m.gmf/model/fsm.gmfgraph			

Figure 2.20. fsmStatic.gmfmap



Do not forget to check if Diagram Node (or Diagram Link) and Tool properties of Node Mapping or Link Mapping have the good values.

🛅 Resource Set			
Platform:/resource/org.kermeta.fsm.gmf/model/fsm.gmfmap			
Mapping			
Top Node Reference < ownedState:State/State> Top Node Mapping < State/State>			
Canvas Mapping			
platform:/resource/org.kermeta.fsm.gmf/model/fsm.ecore			
latform:/resource/org.kermeta.fsm.gmf/model/fsm.gmfgraph 😡			
🏓 platform:/resource/org.kermeta.fsm.gmf/model/fsm.gmftool			
Selection Decret List Tree Table Tree with Columns			
Selection Parent List Tree Table Tree with Columns			
Selection Parent List Tree Table Tree with Columns	ole 🚯 EMF registered packages 🛙		
Selection Parent List Tree Table Tree with Columns Problems @ Javadoc @ Declaration @ SVN Repositories _ Const Property	ole 🚯 EMF registered packages 🛙		
Selection Parent List Tree Table Tree with Columns Problems @ Javadoc 😥 Declaration 👘 SVN Repositories 🚍 Const Property Domain meta information	ole 🚯 EMF registered packages 🛙		
Selection Parent List Tree Table Tree with Columns Problems @ Javadoc 😥 Declaration 🔞 SVN Repositories 🚍 Cons Property Domain meta information Element	ole 🚯 EMF registered packages 🛙 Value 目 State		
Selection Parent List Tree Table Tree with Columns Problems @ Javadoc Declaration Tree with Columns Property	ole 🚯 EMF registered packages 🗍 Value 🗄 State		
Selection Parent List Tree Table Tree with Columns Problems @ Javadoc Declaration Tree with Columns Property Domain meta information Element Misc Related Diagrams	ole 🚯 EMF registered packages 🛙 Value 目 State		
Selection Parent List Tree Table Tree with Columns Problems @ Javadoc Declaration Tree with Columns Property Domain meta information Element Misc Related Diagrams Visual representation	ole 🚯 EMF registered packages 🖡 Value 目 State		
Selection Parent List Tree Table Tree with Columns Problems @ Javadoc Declaration Tree with Columns Property @ Domain meta information Element Misc Related Diagrams Visual representation Appearance Style Output Output	ole 🚯 EMF registered packages 🖡 Value 目 State		
Selection Parent List Tree Table Tree with Columns Problems @ Javadoc Declaration Tree with Columns Tree with Columns Property @ Domain meta information Element Element Misc Related Diagrams Visual representation Appearance Style Context Menu	ole 🚯 EMF registered packages 🖡 Value 🗏 State		
Selection Parent List Tree Table Tree with Columns Problems @ Javadoc Declaration Tree with Columns Tree with Columns Property @ Domain meta information Element Element Misc Related Diagrams Visual representation Appearance Style Context Menu Diagram Node	ole 🚯 EMF registered packages 🗐 Value 🗄 State 🔸 Node State (StateFigure)		

Figure 2.21. Node and tool

On this file we need to add a Feature Label mapping to manage the name label of the state. Right click on Node mapping State -> New Child -> Feature Label Mapping. Edit the diagram label with Diagram Label StateName. You have also to edit Features to display and Features to edit thanks to the Following wizard :

👪 Features to display Feature Label Mapping false		- • •
Filter Available Choices		
Choice Pattern (* or ?)		
Choices	Feature	
State.name:String	Add Remove Up Down	
	ОК	Cancel

Figure 2.22. Select feature and add it

🛃 fsm.gmfmap 🖾				
Part Resource Set				
platform:/resource/org.kermeta.fsm.gmf/model/fsm.gmfmap				
♦ Mapping				
Top Node Reference < ownedState:State/State>				
Node Mapping <state state=""></state>				
Ab Feature Label Mapping false				
 Link Mapping <transition{transition.source:state} transition=""></transition{transition.source:state}> 				
Canvas Mapping				
盲 platform:/resource/org.kermeta.fsm.gmf/model/fsm.ecore				
🔉 platform:/resource/org.kermeta.fsm.gmf/model/fsm.gmfgraph				
🎤 platform:/resource/org.kermeta.fsm.gmf/model/fsm.gmftool				
Selection Parent List Tree Table Tree with Columns				
Problems 🖉 Javadoc 🚱 Declaration 🕅 SVN Repositories 🗐 Console	EMF registered packages 🔲 Propert			
Property	Value			
Domain meta information				
Features to display	State.name:String			
Features to edit	State.name:String			
⊿ Misc				
Diagram Label	Diagram Label StateName			
Read Only	ur false			
Visual representation				
Edit Method	MESSAGE_FORMAT			
Editor Pattern				
Edit Pattern				
View Method	ESSAGE_FORMAT			
View Pattern				

Figure 2.23. Add diagram label into Feature Mapping

Then, edit the LinkMapping Transition like in the following image :

🛃 fsm.gmfmap 🕱	
Part Resource Set	
 platform:/resource/org.kermeta.fsm.gmf/model/fsm.gmfmap Mapping Top Node Reference < ownedState:State/State> Top Node Reference < ownedState:State/State> Node Mapping < State/State> Feature Label Mapping false Link Mapping < Transition{Transition.source:State}/Transition> Canvas Mapping platform:/resource/org.kermeta.fsm.gmf/model/fsm.ecore platform:/resource/org.kermeta.fsm.gmf/model/fsm.gmfgraph platform:/resource/org.kermeta.fsm.gmf/model/fsm.gmftool 	
Selection Parent List Tree Table Tree with Columns	EMF registered packages 🔲 Properties 🕱
Property	Value
Domain meta information	
Containment Feature Element Source Feature Target Feature	Image: State.outgoingTransition:Transition Image: Transition Image: Transition.source:State Image: Transition.source:State
⊿ Misc	
Related Diagrams	
 Visual representation 	
Appearance Style	
Context Menu	
	Connection Transition A Creation Tool StateIncomingTransition
1001	Treation root statemcoming transition

Figure 2.24. Edit the LinkMapping Transition

Now, the file fsm.gmfmap looks like this :



Figure 2.25. fsm.gmfmap

2.3.6. Diagram Editor Gen Model

Click on Transform (at the right of the case mapping model) and the file fsm.gmfgen is created. It looks like :



Figure 2.26. fsmStatic.gmfmap

Then click on generate diagram editor. Your project should be like this :

4	B	org.kermeta.fsm.gmf
	\triangleright	/ src
	\triangleright	➡ JRE System Library [JavaSE-1.6]
	\triangleright	🛋 Plug-in Dependencies
	\triangleright	🗁 META-INF
	⊿	🗁 model
		🖶 fsm.ecore
		脂 fsm.genmodel
		😪 fsm.gmfgen
		🔬 fsm.gmfgraph
		🛃 fsm.gmfmap
		🏓 fsm.gmftool
		🔬 fsm.trace
		build.properties
		plugin.properties
		🚯 plugin.xml
⊳	1	org.kermeta.fsm.gmf.diagram
⊳	è	org.kermeta.fsm.gmf.edit
⊳	e	org.kermeta.fsm.gmf.editor
⊳	e	org.kermeta.fsm.gmf.tests

Figure 2.27. fsmStatic.gmfmap

So, we have generated the graphical editor for FSM. The next section explains how to test it.

2.4. Use the generated editor

To test the generated editor right click on org.kermeta.fsm.gmf.diagram -> Run as -> Eclipse application. It opens a new Eclipse instance (runtime) where the tested plugin is added. Create a new General Project in this new Eclipse named testFSMEditor. Right click on it -> New -> Example -> Fsm Diagam

New Example	
Select a wizard	
Wizards:	
type filter text	
 Fsm Diagram Valueflow Diagram Eclipse Modeling Framework Capabilities Example Extended Library Model Example Generator Extension Example Java Model Example Java Model Example Model Exporter Example Model Exporter Example GMF (Graphical Modeling Framework) Plug-ins Clipboard Support Geoshape Logic 	► II ▼
(?) < Back Next > Finish	Cancel

Figure 2.28. Create Fsm Diagram model

We choose to call our files test.fsm and test.fsm_diagram In this generated editor the model is stored into two files :

- 1. test.fsm : store information on domain model
- 2. test.fsm_diagram : store graphical informations

Open the test.fsm_diagram and use the palette. You can have a result like this :

File Edit Diagram Navigate Search Pr	oject Run Wii	ndow Help		
C1 - 12 🖻			🎄 ▾ 🜔 ▾ Q₄ ▾ 👍 🛱 ତି ▾	🥭 🖋 👻
Segoe UI 👻 9	BIA	- ≫ - ⊿ - → - ฿ ฿ - ฿ - ฿	9 ▼ 🖺 🛒 💥 🗖 ▼ 🛱 ▼ 100%	•
📕 Package Exp 🙁 🍃 Hierarchy 🗖 🗖	test.fsm_dia	gram 🕱		
□ \$ >			^ 😨 Palette	⊳
> 🗁 example			$\searrow \oplus \bigcirc$	
b B MyFirstEMFInstances	e 🔶	t1 📥 st2	♦ State	
> 🔁 test		-	A Transition	
testFSMEditor			✓ Transition	
A test.fsm				
testTutoriall.ogo				
			Ŧ	
			4	
	🔝 Problems 🔎 Javadoc 🔯 Declaration 🗔 Properties 🕸 📎 Error Log 📮 Console			
	♦ State			
	Core	Property	Value	
	Appearance	Incoming Transition	Transition	
	pp salarice	Name	l≡ st2	

Figure 2.29. Edit Fsm Diagram model

We have created a very little graphical editor to illustrate the FSM example . For more informations please refer to the GMF documentation.