Chapter 3

Graphical Editor Development with Sirius

3.1 Preliminaries

For Sirius to be able to work with custom metamodels, either it has to be installed into the host Eclipse, or Sirius editing needs to be executed in a runtime Eclipse instance. To install the metamodel, use the Export/Deployable plugins and fragments wizard, and use the Install into host option. Restart as needed.

Some tips and tricks:

- When working with Sirius, always use the Model browser instead of the usual Package Explorer. This is always available in the Sirius or Modeling perspectives.
- Always set up the IDs of the created view specification elements.
- In case of problems, always check labels, etc.
- Domain classes can usually be entered via content assist; feature names generally cannot.
- In case of Acceleo expressions, either enter the `[/]` entirely or using the content assist, otherwise Acceleo content assist will not work.

3.2 Initializing a Viewpoint specification project

1. Create a New viewpoint specification project (in the Sirius category)
2. Define a new viewpoint and a new diagram representation in the model
   (a) Set the supported file extensions to either `socialnetwork` or `socialnetwork xmi` on the new viewpoint
   (b) Add the socialnetwork metamodel on the Metamodels property tab of the diagram representation

3.3 Open model in new diagram

1. Create a new Modeling project
2. Add a new instance model of the Social network example (e.g. copy the example from the modeling project)
3. Right click on the project, and use the Select Viewpoint option to enable the SocialNetwork diagram
4. Open the model in a new editor

3.4 Person Mappings and Tools

1. Create Node mapping for Person
   (a) Set up ID and Domain Class
   (b) Add style definition
2. Create Relation mapping from Entity for Acquaintance
   (a) Set up ID and Domain Class
   (b) Source and target mappings can be set to Person mapping
   (c) Set up source and target features to the corresponding Ecore model feature names
   (d) Set up style definition, with Label Expression `feature:type`
3. Add a Tool Section
4. Add Creation Tool for Person
   (a) Create Node Creation Tool (Element Creation submenu)
   (b) Set up id, label and mapping!
   (c) Under begin, Change context to `container`, Create a Person instance and Set up name (Simple version: `person`, more complex version: `['person' + container.eContents()->size()/]`)

3.5 Community Mappings and Tools

1. Create Containment mapping for Top-level Communities
2. Domain Class: Community, Children presentation: Freeform, Style: Gradient
3. Create Containment mapping for subcommunities (below community mapping)
4. Limit Container mapping with Semantic Expression: 'feature:entities'
   (a) Domain class: Community
   (b) Semantic expression: change to feature:eContents!
   (c) Import tab: Reused Container mapping - reuse own mapping
5. Create Community Tool
6. Two mappings: community and subcommunity
7. Create Operations described via switch, see Figure 3.1

3.6 Membership Mapping and Tool

1. Create Relation-based mapping for membership
2. Two source mappings: community and subCommunity
3. Target mapping: person
4. Target finder: 'feature:members'
5. Create membership tool (Simple Edge creation tool)
6. However, setting the values requires tricky OCL expression. Main idea: get the existing list of members and extend it with new value (see Figure 3.2.

   ```
   [self.oclAsType(Community).members->including(target)/]
   ```

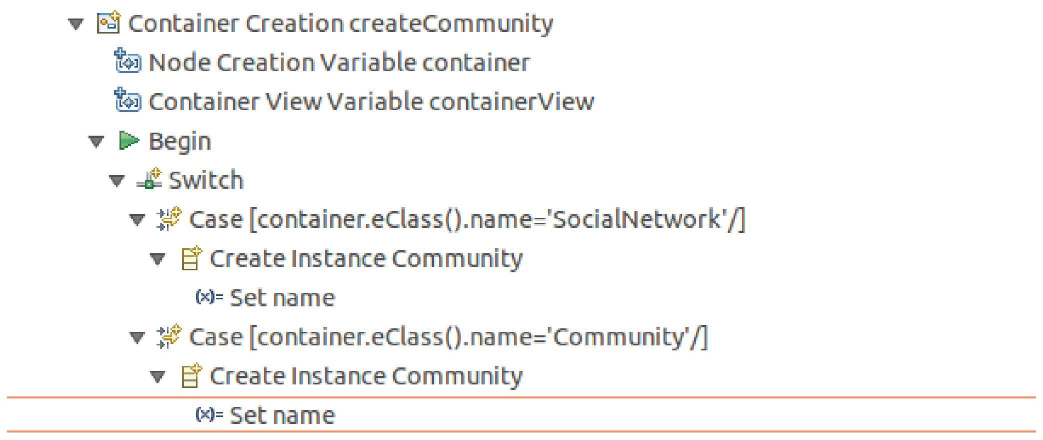
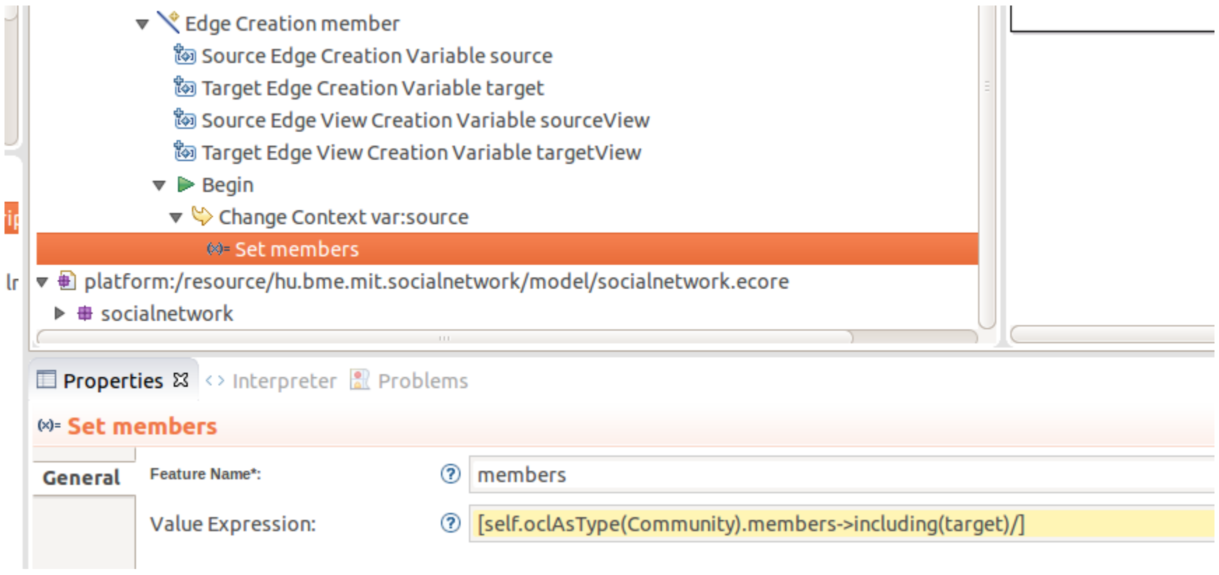

Container Creation createCommunity
Node Creation Variable container
Container View Variable containerView
Begin
Switch
Case [container.eClass().name='SocialNetwork'/]
Create Instance Community
Set name
Case [container.eClass().name='Community'/]
Create Instance Community
Set name

Figure 3.1: Create Community Operations


Edge Creation member
Source Edge Creation Variable source
Target Edge Creation Variable target
Source Edge View Creation Variable sourceView
Target Edge View Creation Variable targetView
Begin
Change Context var:source
Set members
platform:/resource/hu.bme.mit.socialnetwork/model/socialnetwork.ecore
socialnetwork

Set members

Feature Name*: members
Value Expression: [self.oclAsType(Community).members->including(target)/]

Figure 3.2: Setting membership reference