

EMF-GMF-Tutorial: Petrinet

Petrinet-Metamodell anlegen

1. File/New/Other: Empty EMF Project
Project Name: de.upb.agengels.se.petrinet
2. Rechtsklick auf model-Verzeichnis => New/Other: Ecore Diagram
Domain File Name: Petrinet.ecore
3. EClasses anlegen: Petrinet, Place, Transition, Token, NamedElement
4. Vererbungsbeziehungen anlegen: Alles erbt von NamedElement
5. NamedElement rechts klicken => Show Properties View
Häkchen bei „Is Abstract“ setzen
6. EAttribute in NamedElement anlegen
7. Attribut selektieren, in der Properties View einstellen:
Name: name, Lower Bound und Upper Bound: 1, EType: EString
8. EReferences anlegen:
Petrinet -> Place (places), Petrinet -> Transition (transitions), Place
-> Token (tokens)
Upper Bound: *, Häkchen bei „Is Containment“ setzen
9. EReferences anlegen:
Transition -> Place (sourcePlaces), Transition -> Place (targetPlaces)
Upper Bound: *
10. Speichern, visuellen Editor schließen
11. Petrinet.ecore doppelklicken
12. Wurzelement im Editor rechtsklicken => Validate

Generatormodell erstellen, Code generieren

13. Rechtsklick auf Petrinet.ecore => New/Other: EMF Generator Model
Model Importer (zweiter Schritt): Ecore model
14. Im GenModel-Editor:
das Wurzel-Petrinet ausklappen
das Package-Petrinet selektieren, in der Properties View:
Base Package: de.upb.agengels.se
15. Speichern

16. Rechtsklick auf das Wurzel-Petrinet => Generate All

Runtime Workbench starten

17. Hauptmenü: Run/Run Configurations...
18. „Eclipse Application“ selektieren, „New Launch Configuration“ klicken
Name: Runtime Workbench, Location: \${workspace_loc}/../runtime-
PetrinetWorkspace
19. Run klicken, um Runtime Workbench zu starten

Runtime Workbench: Beispiel-Petrinet erstellen

20. File/New/Other: Project
Project name: Petrinets
21. Projekt Petrinets rechtsklicken
=> New/Other: Petrinet Model
File name: Simple.petrinet, Model Object: Petrinet
22. Rechtsklick auf Petrinet => Show Properties View
Name in Properties View setzen, z.B. Simple Petrinet
23. Funktionalität des Editors testen:
Rechtsklick auf Petrinet => New Child: Place (Name : p1)
Rechtsklick auf Petrinet => New Child: Place (Name : p2)
Rechtsklick auf erzeugten Place p1 => New Child: Token
Rechtsklick auf Petrinet => New Child: Transition
In Properties View: p1 zu Source Places der erzeugten Transition hinzufügen
In Properties View: p2 zu Target Places der erzeugten Transition hinzufügen
24. Runtime Workbench schließen

Graphischen Editor erstellen

25. Projekt de.upb.agengels.se.petrinet selektieren, Window/Show View/Other:
GMF Dashboard
26. Dashboard: Select Domain Model klicken => Petrinet.ecore
27. Dashboard: Select Domain Gen Model klicken => Petrinet.genmodel
28. Dashboard: Derive Graphical Def Model klicken
Diagram Element: Petrinet
29. Petrinet.gmfgraph öffnen, Canvas rechtsklicken => New Child/Compartment

30. Compartment selektieren, in der Properties View:
Name: PlaceCompartment, Figure: PlaceFigure
31. Figure Gallery Default/Figure Descriptor TransitionSourcePlacesFigure/
PolylineConnection TransitionSourcePlacesFigure:
Source Decoration und Target Decoration tauschen (Source Decoration ist vorher leer)
32. Figure Gallery Default/Figure Descriptor PlaceFigure:
Enthaltenes Rectangle löschen
Rechtsklick auf Figure Descriptor => New Child/Ellipse (Name: PlaceFigure)
Rechtsklick auf Ellipse => New Child/Label (Name: PlaceNameFigure)
Child Access getFigureNull selektieren, in Properties View:
Figure: Label PlaceNameFigure
siehe auch Abb. 1 und 2
33. Schritt 32 entsprechend für Figure Descriptor TokenFigure ausführen, zusätzlich:
Rechtsklick auf Ellipse TokenFigure => New Child/Background Color Constant
Color (Value: black)
34. Speichern
35. Dashboard: Derive Tooling Def Model klicken
Diagram Element: Petrinet
36. Dashboard: Combine klicken
Class: Petrinet
Mapping: Token bei den Links entfernen
37. Petrinet.gmfmap anpassen:
ChildReference, inneres Node Mapping und Compartment Mapping erstellen
Properties wie in Abb. 3-8
Transition entsprechend Node (Abb. 3-5, Place wird Transition etc.)
Zweites Link Mapping entsprechend erstem Link Mapping (Abb. 9, Source wird Target etc)
38. Speichern
39. Dashboard: Transform klicken (kein Häkchen bei RCP machen)
40. Rechtsklick auf generiertes Petrinet.gmfgen => Generate diagram code

Runtime Workbench: Petrinet graphisch erstellen

41. Runtime Workbench starten

42. Rechtsklick auf Projekt Petrinets => New/Other: Petrinet Diagram
Filnamen: MoreComplex.petrinet_diagram bzw. MoreComplex.petrinet
43. Petrinet erstellen
44. Speichern
45. MoreComplex.ecore doppelklicken, abstrakte Syntax mit konkreter Syntax vergleichen

Erstellte Projekte exportieren

1. File/Export: Archive File
Nur Projekt de.upb.agengels.se.petrinet selektieren (Rest wird generiert)
Zieldatei wählen (Browse), Save in zip format => Finish
2. Zip-Datei zumailen (zur Not per webmail.upb.de), auf USB-Stick kopieren etc.
3. Andere Eclipse-Installation:
File/Import: Existing Projects into Workspace

Eclipse auf den Poolraum-Rechnern starten

1. Terminal öffnen
2. cd /software/eclipse
3. ./eclipse

Eclipse zu Hause installieren

1. Eclipse herunterladen
<http://www.eclipse.org/downloads/> => Eclipse Modeling Tools
2. Entpacken, eclipse.exe starten
3. Help/Install New Software => Work with: Helios Update Site => Modeling/Graphical Modeling Framework SDK installieren

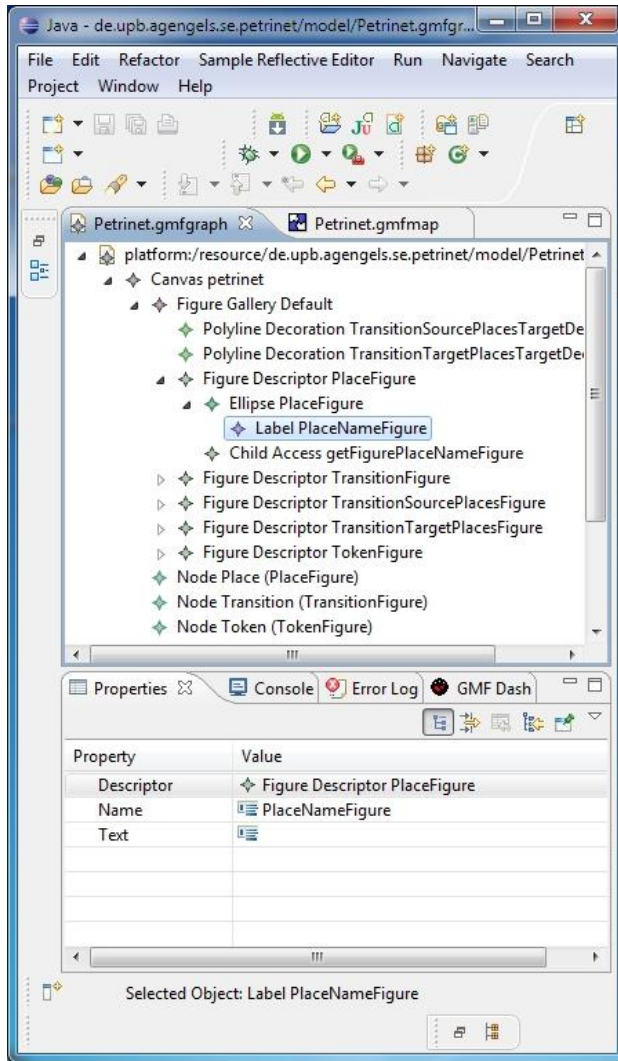


Abbildung 1

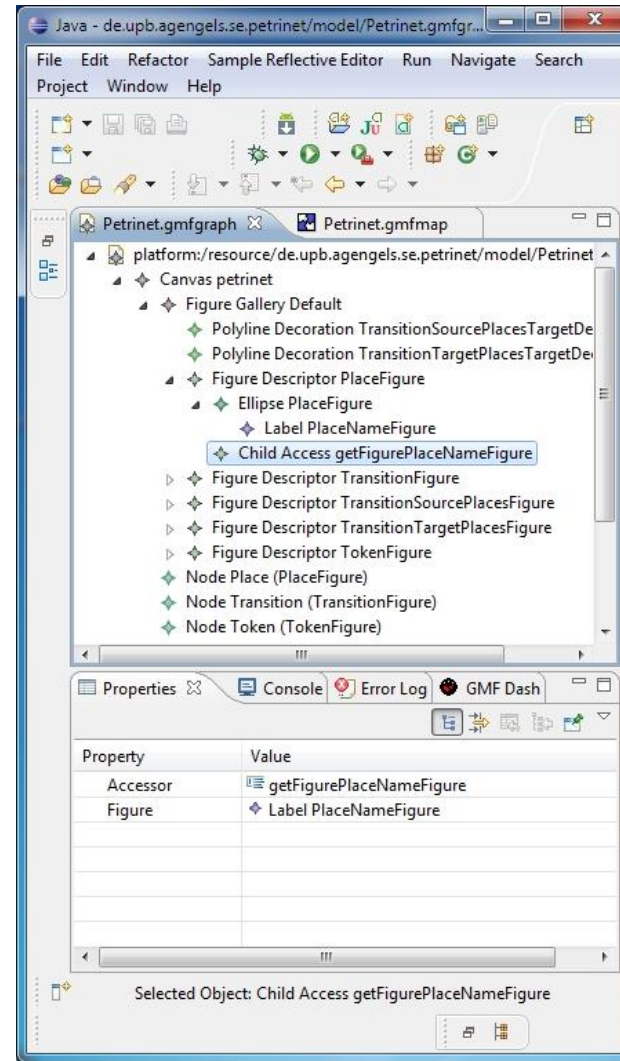


Abbildung 2

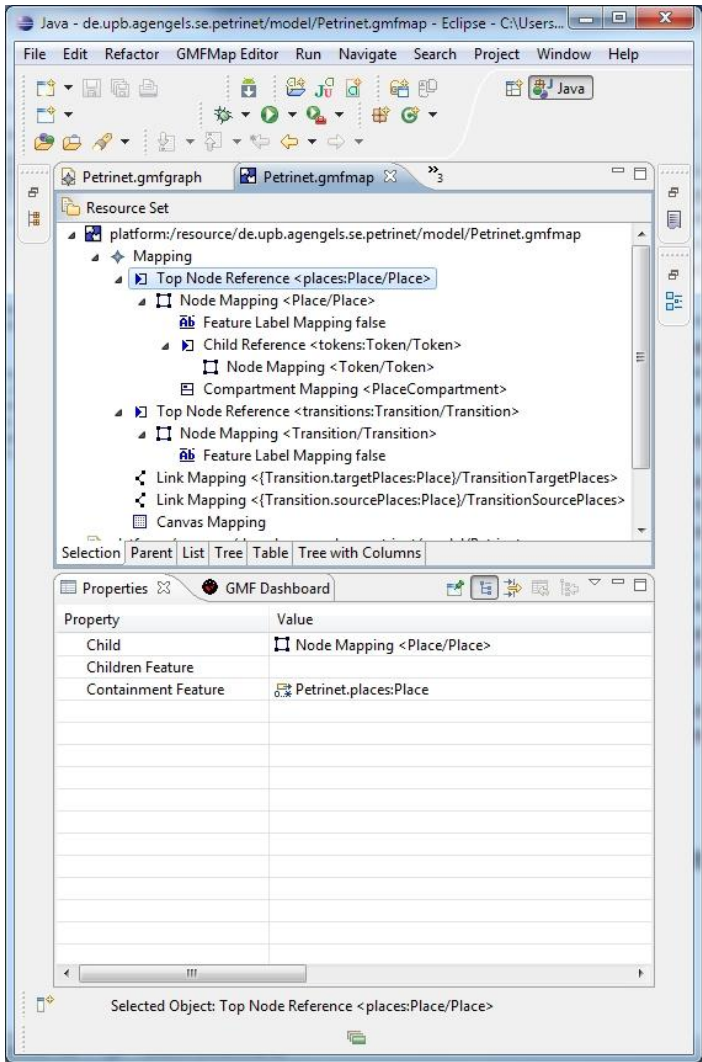


Abbildung 3

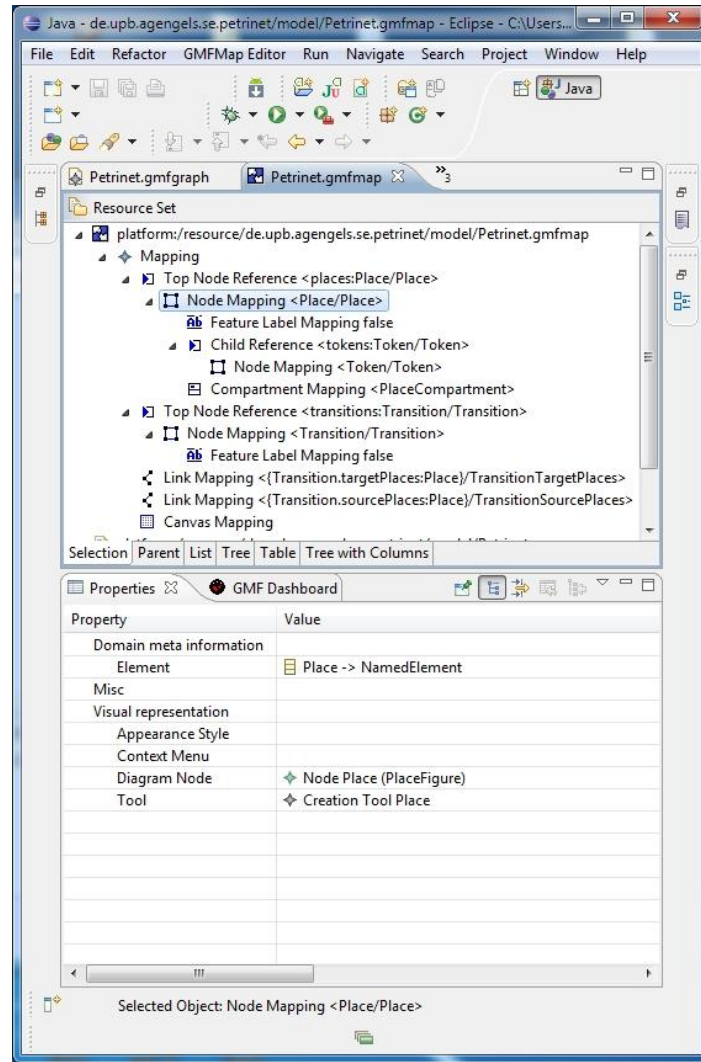


Abbildung 4

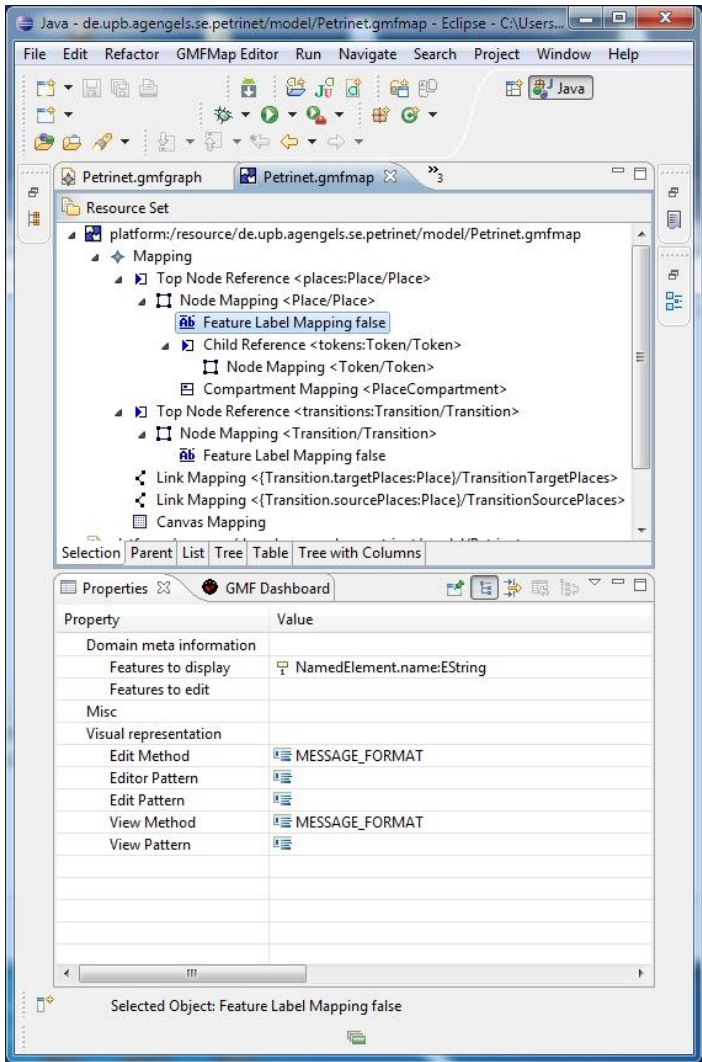


Abbildung 5

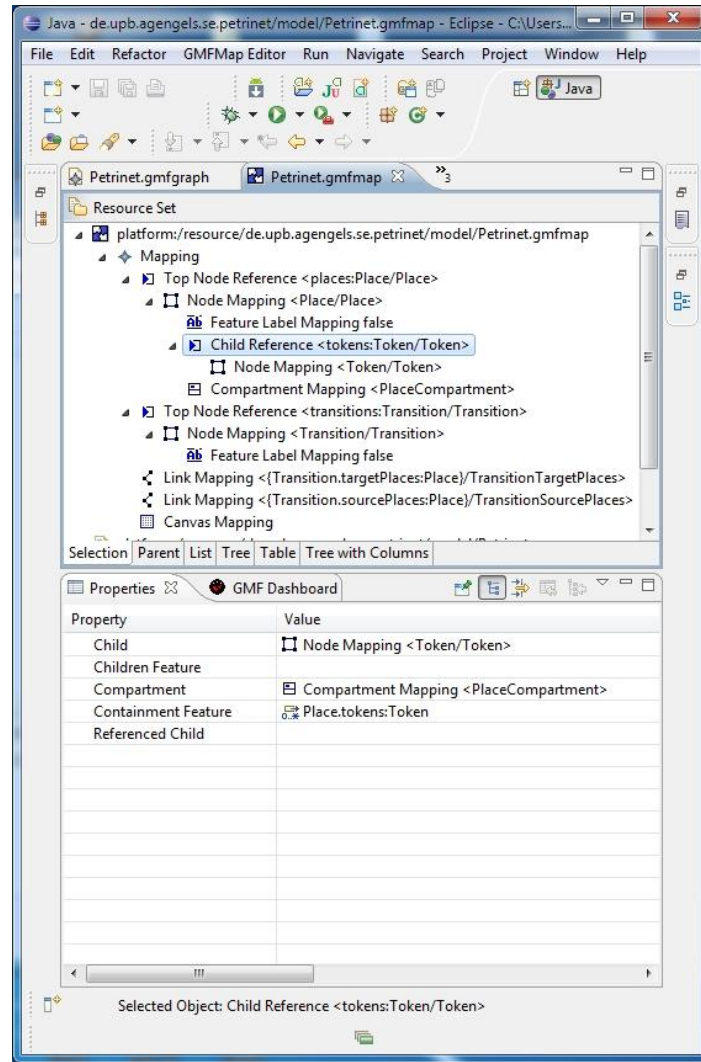


Abbildung 6

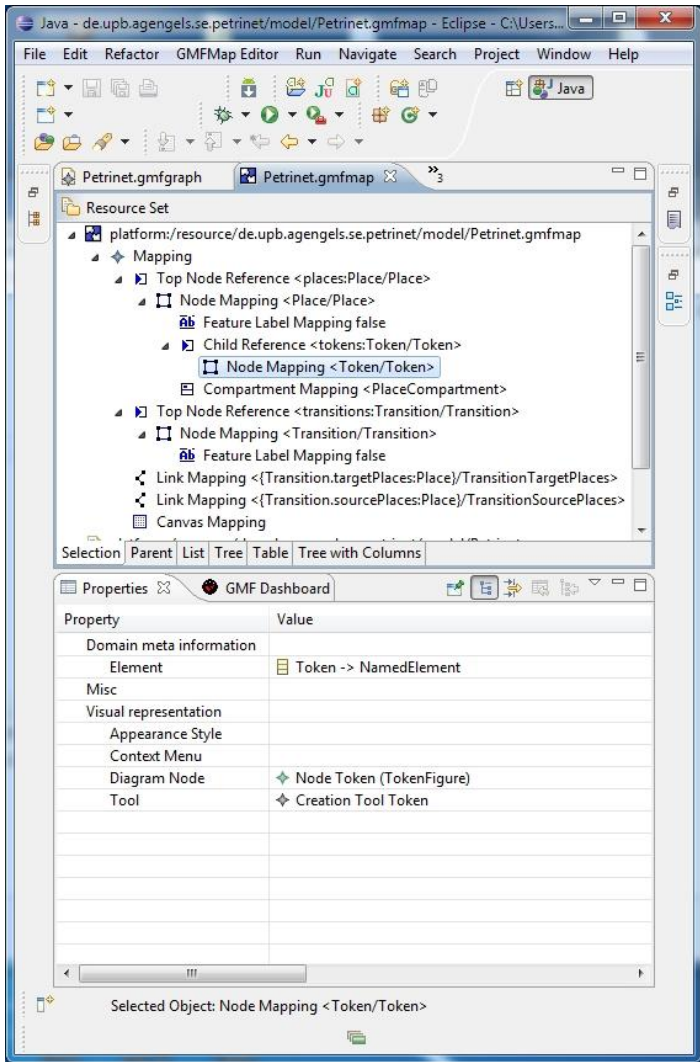


Abbildung 7

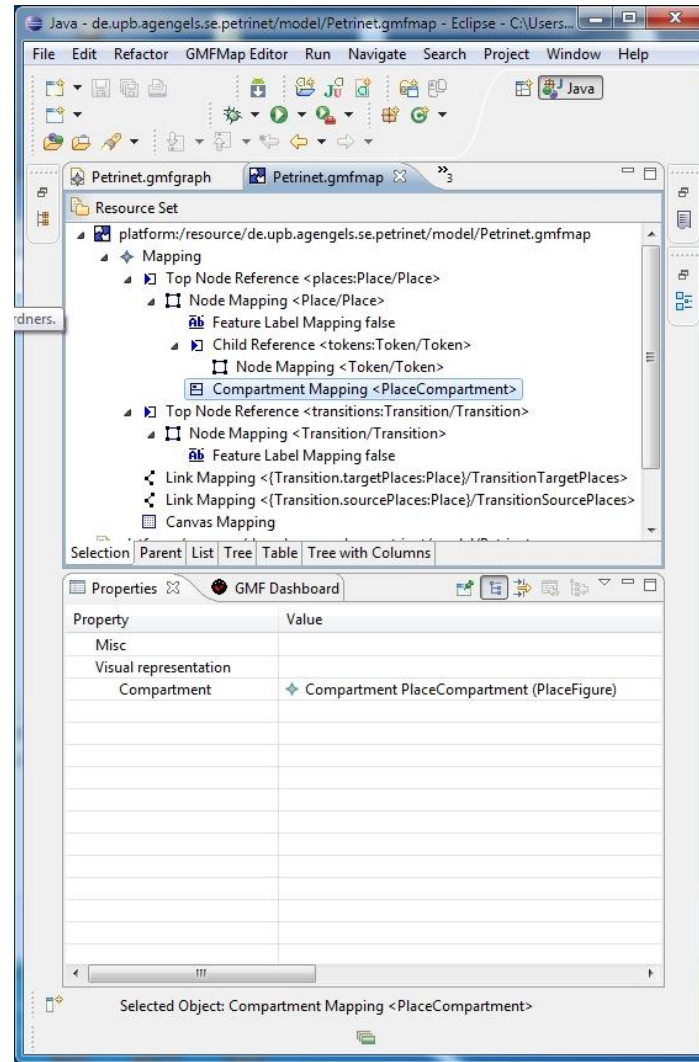


Abbildung 8

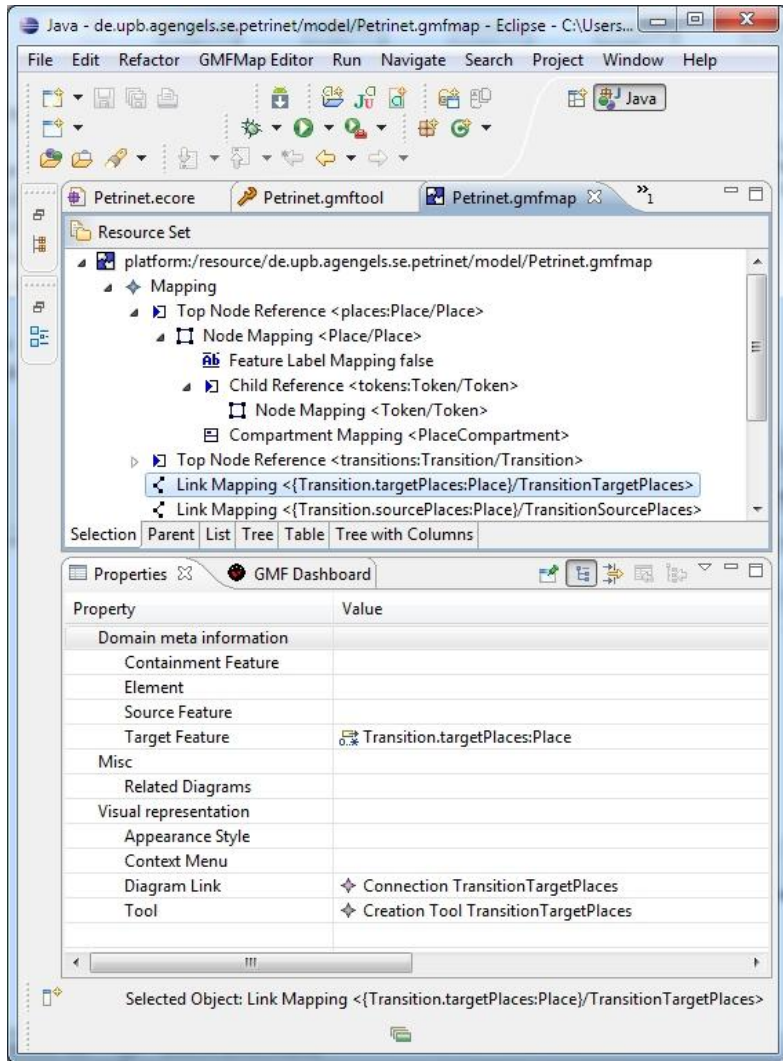


Abbildung 9

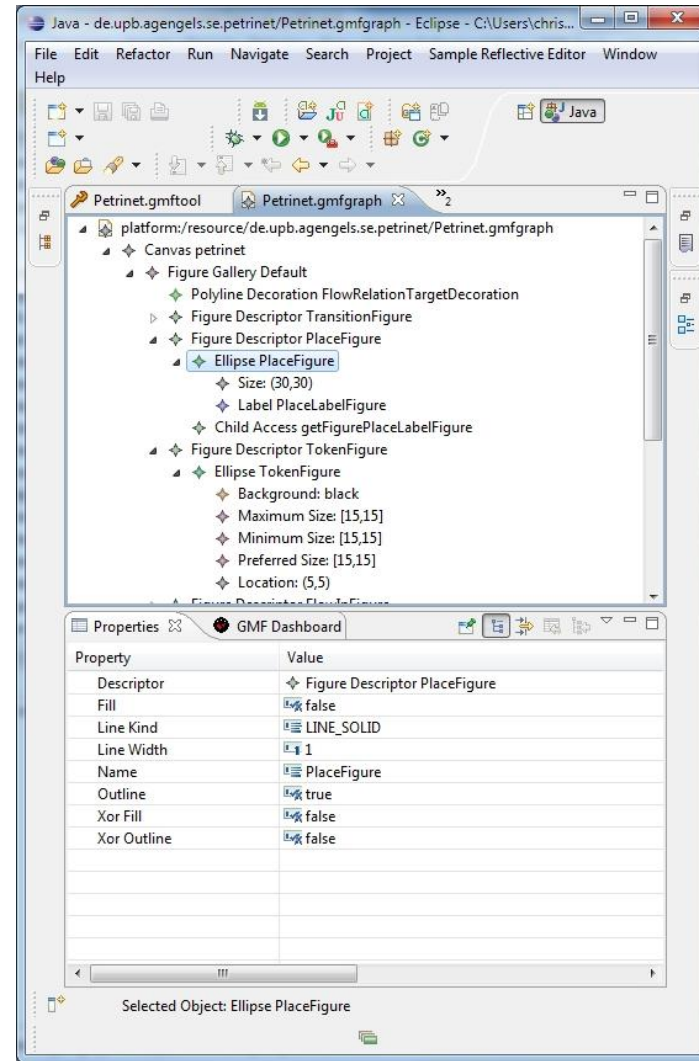


Abbildung 10