

ZAVOD ZA ELEKTRONIKU, MIKROELEKTRONIKU, RAČUNALNE I INTELIGENTNE SUSTAVE
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
SVEUČILIŠTE U ZAGREBU

GRAPHICAL MODELING FRAMEWORK

Dražen Nežić
SEMINARSKI RAD

Zagreb, 2007.

Sadržaj

1. Uvod.....	1
2. Eclipse Modeling Framework.....	2
2.1. Uporaba Eclipse Modeling Frameworka.....	3
2.1.1. Potrebne Eclipse komponente.....	3
2.1.2. Kreiranje XML sheme (XSD).....	3
2.1.3. Kreiranje EMF projekta.....	4
2.1.4. Generiranje koda iz EMF modela.....	10
2.1.5. Generiranje editora za model.....	11
2.1.6. Pokretanje generiranog editora.....	12
3. Graphical Modeling Framework.....	15
3.1. Potrebni GMF dodaci za Eclipse.....	16
3.2. Primjer uporabe GMF Frameworka.....	16
3.2.1. GMF Dashboard.....	19
3.2.2. Graphical Definition Model.....	19
3.2.3. Tooling Definition Model.....	21
3.2.4. Mapping model.....	23
3.2.5. Kreiranje GMF generator modela i grafičkog editora.....	26
4. Prikaz mogućnosti GMF editora.....	29
4.1. Otvaranje i zatvaranje odjeljka s atributima.....	29
4.2. Direktno editiranje.....	29
4.3. Diagram Assistants.....	30
4.4. Osnovni alati.....	30
4.5. Izvoz dijagrama.....	31
5. Dodatak A – ispis modela korištenih u primjeru.....	32
5.1. knjiznica.ecore.....	32
5.2. Knjiznica.genmodel.....	33
5.3. Knjiznica.xsd2ecore.....	34
5.4. Knjiznica.gmftool.....	35
5.5. Knjiznica.gmfgraph.....	35
5.6. Knjiznica.gmfmap.....	36
6. Dodatak B – Mapping model, prikaz hijerarhije.....	39
7. Zaključak.....	40
8. Literatura.....	41

1. Uvod

*Eclipse*¹ je dio *open source*² zajednice čiji su projekti usmjereni k izgradnji otvorenog razvojnog okruženja sastavljenog od nadogradivih komponenti, pomoćnih alata, izvršnih okruženja za izgradnju, distribuciju i upravljanje *softwareom* tijekom njegovog životnog ciklusa. Eclipse je izgrađen u programskom jeziku Java³ što mu omogućuje fleksibilnost i prenosivost.

Graphical Modeling Framework (GMF), okruženje za grafičko modeliranje je podprojekt *Eclipse Modeling* projekta. Omogućuje poveznicu između Eclipseovog okruženja za modeliranje, *Eclipse Modeling Framework* (EMF), i okruženja za grafičko editiranje, *Graphical Editing Framework* (GEF). *Eclipse Modeling Framework* i *Graphical Editing Framework* se uobičajeno koriste kada se žele vizualizirati modeli opisani EMF-om. Iako postoje načini kako povezati ove dvije tehnologije, njihova primjena je izuzetno teška i dugotrajna. GMF omogućuje infrastrukturu koja olakšava njihovu uporabu pri izgradnji aplikacijskih modela u Eclipseu.

GMF ima veliki potencijal postati glavno okružje za brzi razvoj standardiziranih grafičkih editora modela unutar Eclipse razvojnog okruženja.

S ciljem korištenja *Graphical Modeling Frameworka* prvo se moramo upoznati s *Eclipse Modeling Frameworkom*, kako bismo pomoću njega izgrađen model (*Domain Model*), mogli grafički editirati.

1 www.eclipse.org

2 www.opensource.org

3 www.sun.com/java

2. Eclipse Modeling Framework

Eclipse Modeling Framework je *open source* okruženje koje služi za modeliranje i generiranje izvornog koda na osnovi strukturiranih modela. EMF omogućuje brzo i učinkovito pretvaranje modela u prilagodljiv Java kod. Osnovni EMF paket uključuje *XML¹ Schema Infoset Model (XSD)* i implementaciju *Service Data Objects (SDO²)*. XSD omogućuje model i *API³* za upravljanje komponentama XML sheme s pristupom *DOM⁴* reprezentaciji dokumenta opisanog shemom.

Kompletan kod, potreban za podatkovni model, uporabom EMF-a se generira na osnovi definicije modela. EMF generira pregledan kod uporabom prilagodljivih predložaka. Promjene koje korisnik izvrši nad generiranim kodom neće biti izgubljene regeneriranjem razreda već uvažene. Korisnik može programski vršiti upite o strukturi modela te istu i mijenjati. Također, EMF može spremiti instance modela u XMI format, te generirati editor posebno za korisnikov model.

EMF se sastoji od tri osnovna dijela:

- EMF – jezgra EMF okruženja, uključuje *meta model (Ecore)* za opis modela i upravljanje modelima tijekom izvođenja programa: obavijesti o promjenama modela, *XMF⁵* serijalizacija, učinkovit *API* za upravljanje EMF objektima.
- EMF.edit – okruženje uključuje općenite, nanovo uporabljive razrede za izgradnju editora EMF modela. Omogućuje: razrede koji predstavljaju sadržaje i nazive, razrede koji omogućavaju EMF modelima prikaz uporabom osnovnih *JFACE⁶* pogleda i mapa svojstava, naredbeno okruženje koje uključuje skup osnovnih razreda koji implementiraju naredbe za izgradnju editora.
- EMF.Codegen – Alat za generiranje koda koji omogućuje generiranje svega potrebnog za izgradnju potpunog editora iz EMF modela. Uključuje *GUI⁷* iz kojeg se mogu specificirati postavke generiranja koda te pokretanje samog generiranja.

EMF omogućuje tri stupnja generiranja izvornog koda:

- Model – omogućuje Java razrede, sučelja (engl. *interface*) i implementacijske razrede za sve razrede unutar modela.
- Adapters – generira implementacijske razrede nazvane *ItemProviders*, koji prilagođavaju razrede modela editiranju i prikazu.
- Editor – kreira pravilno strukturirani editor konforman preporučenom stilu EMF model editora i služi kao početna točka za daljnje prilagođavanje.

Svi generatori podržavaju regeneriranje koda, istovremeno zadržavajući korisnikove promjene. Generatori se mogu pozvati uporabom GUI-a ili iz komandne linije.

1 eXtended Markup Language

2 Okruženje za razvoj podatkovnih aplikacija

3 Application Programming Interface

4 Document Object Model

5 XML Metadata Interchange

6 Skup razreda, alatki za UI programiranje u Eclipseu

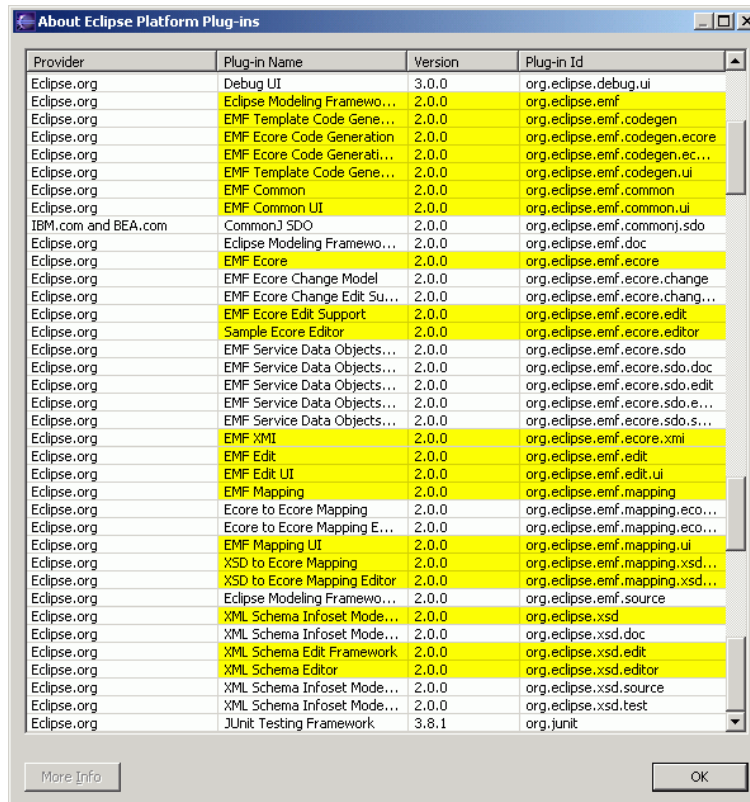
7 Graphical User Interface

2.1. Uporaba Eclipse Modeling Frameworka

Kroz jedan primjer pokazati ćemo i objasniti kreiranje EMF modela i njegovog editora unutar Eclipse IDE-a.

2.1.1. Potrebne Eclipse komponente

Pokrenimo Eclipse IDE¹ te odaberimo u izborniku “Help/About Eclipse Platform”. U novootvorenom prozoru odaberimo “Plug-In Details” te provjerimo da li su žutom bojom označeni dodaci, *plug-inovi*, prisutni:



Provider	Plug-in Name	Version	Plug-in Id
Eclipse.org	Debug UI	3.0.0	org.eclipse.debug.ui
Eclipse.org	Eclipse Modeling Framework...	2.0.0	org.eclipse.emf
Eclipse.org	EMF Template Code Gene...	2.0.0	org.eclipse.emf.codegen
Eclipse.org	EMF Ecore Code Generation	2.0.0	org.eclipse.emf.codegen.ecore
Eclipse.org	EMF Ecore Code Generati...	2.0.0	org.eclipse.emf.codegen.ec...
Eclipse.org	EMF Template Code Gene...	2.0.0	org.eclipse.emf.codegen.ui
Eclipse.org	EMF Common	2.0.0	org.eclipse.emf.common
Eclipse.org	EMF Common UI	2.0.0	org.eclipse.emf.common.ui
IBM.com and BEA.com	CommonJ SDO	2.0.0	org.eclipse.emf.commonj.sdo
Eclipse.org	Eclipse Modeling Framework...	2.0.0	org.eclipse.emf.doc
Eclipse.org	EMF Ecore	2.0.0	org.eclipse.emf.ecore
Eclipse.org	EMF Ecore Change Model	2.0.0	org.eclipse.emf.ecore.change
Eclipse.org	EMF Ecore Change Edit Su...	2.0.0	org.eclipse.emf.ecore.chang...
Eclipse.org	EMF Ecore Edit Support	2.0.0	org.eclipse.emf.ecore.edit
Eclipse.org	Sample Ecore Editor	2.0.0	org.eclipse.emf.ecore.editor
Eclipse.org	EMF Service Data Objects...	2.0.0	org.eclipse.emf.ecore.sdo
Eclipse.org	EMF Service Data Objects...	2.0.0	org.eclipse.emf.ecore.sdo.doc
Eclipse.org	EMF Service Data Objects...	2.0.0	org.eclipse.emf.ecore.sdo.edit
Eclipse.org	EMF Service Data Objects...	2.0.0	org.eclipse.emf.ecore.sdo.e...
Eclipse.org	EMF Service Data Objects...	2.0.0	org.eclipse.emf.ecore.sdo.s...
Eclipse.org	EMF XML	2.0.0	org.eclipse.emf.ecore.xmi
Eclipse.org	EMF Edit	2.0.0	org.eclipse.emf.edit
Eclipse.org	EMF Edit UI	2.0.0	org.eclipse.emf.edit.ui
Eclipse.org	EMF Mapping	2.0.0	org.eclipse.emf.mapping
Eclipse.org	Ecore to Ecore Mapping	2.0.0	org.eclipse.emf.mapping.eco...
Eclipse.org	Ecore to Ecore Mapping E...	2.0.0	org.eclipse.emf.mapping.eco...
Eclipse.org	EMF Mapping UI	2.0.0	org.eclipse.emf.mapping.ui
Eclipse.org	XSD to Ecore Mapping	2.0.0	org.eclipse.emf.mapping.xsd...
Eclipse.org	XSD to Ecore Mapping Editor	2.0.0	org.eclipse.emf.mapping.xsd...
Eclipse.org	Eclipse Modeling Framework...	2.0.0	org.eclipse.emf.source
Eclipse.org	XML Schema Infoset Mode...	2.0.0	org.eclipse.xsd
Eclipse.org	XML Schema Infoset Mode...	2.0.0	org.eclipse.xsd.doc
Eclipse.org	XML Schema Edit Framework	2.0.0	org.eclipse.xsd.edit
Eclipse.org	XML Schema Editor	2.0.0	org.eclipse.xsd.editor
Eclipse.org	XML Schema Infoset Mode...	2.0.0	org.eclipse.xsd.source
Eclipse.org	XML Schema Infoset Mode...	2.0.0	org.eclipse.xsd.test
Eclipse.org	JUnit Testing Framework	3.8.1	org.junit

Slika 2.1: Potrebni dodatci za uporabu EMF-a unutar Eclipsea

Ukoliko nemamo sve potrebne dodatke, možemo ih jednostavno instalirati iz samog Eclipse IDE-a, odabirom u izborniku “Help/Software Updates/Find and Install...”.

2.1.2. Kreiranje XML sheme (XSD)

U ovom primjeru generirati ćemo EMF model uporabom XML sheme. Potrebna nam je XSD datoteka slijedećeg sadržaja:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.example.eclipse.org/Knjiznica"
  xmlns:ecore="http://www.eclipse.org/emf/2002/Ecore"
  xmlns:lib="http://www.example.eclipse.org/Knjiznica"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:complexType name="Knjiga">
    <xsd:sequence>
      <xsd:element name="naziv" type="xsd:string"/>
      <xsd:element name="stranica" type="xsd:int"/>
      <xsd:element name="kategorija" type="lib:Kategorija"/>
    
```

¹ Integrirano razvojno okruženje (engl. *Integrated Development Environment*)


```

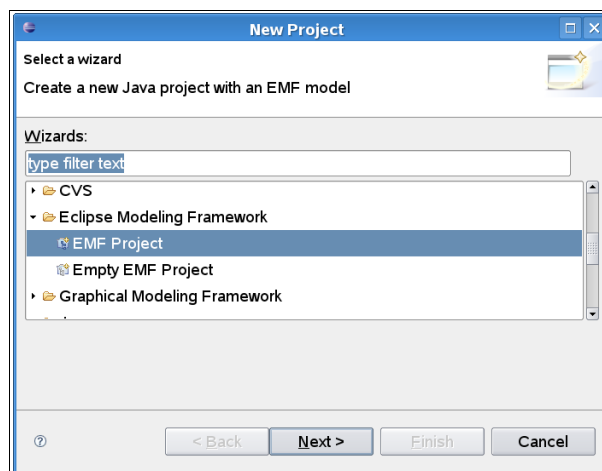
    <xsd:element name="autor" type="xsd:anyURI"
      ecore:reference="lib:Pisac" ecore:opposite="knjige"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Pisac">
  <xsd:sequence>
    <xsd:element name="ime" type="xsd:string"/>
    <xsd:element maxOccurs="unbounded" minOccurs="0" name="knjige"
      type="xsd:anyURI" ecore:reference="lib:Knjiga" ecore:opposite="autor"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Knjiznica">
  <xsd:sequence>
    <xsd:element name="ime" type="xsd:string"/>
    <xsd:element maxOccurs="unbounded" minOccurs="0"
      name="pisci" type="lib:Pisac"/>
    <xsd:element maxOccurs="unbounded" minOccurs="0"
      name="knjige" type="lib:Knjiga"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="Kategorija">
  <xsd:restriction base="xsd:NCName">
    <xsd:enumeration value="Misterij"/>
    <xsd:enumeration value="ZnanstvenaFantastika"/>
    <xsd:enumeration value="Biografija"/>
  </xsd:restriction>
</xsd:simpleType>
</xsd:schema>

```

Spremimo gore navedeni sadržaj u datoteku na disk pod nazivom *Knjiznica.xsd*. Kako *XML Schema* nije dovoljno ekspresivna, u datoteci se mogu vidjeti atributi *ecore:reference* koji nam omogućuju povezivanje *Knjiga* s *Piscima*. Ova referenca će nam poslužiti da bolje pokažemo mogućnosti *Graphical Modeling Frameworka*, te smo je također mogli i izostaviti za potrebe nekog drugog strukturnog modela.

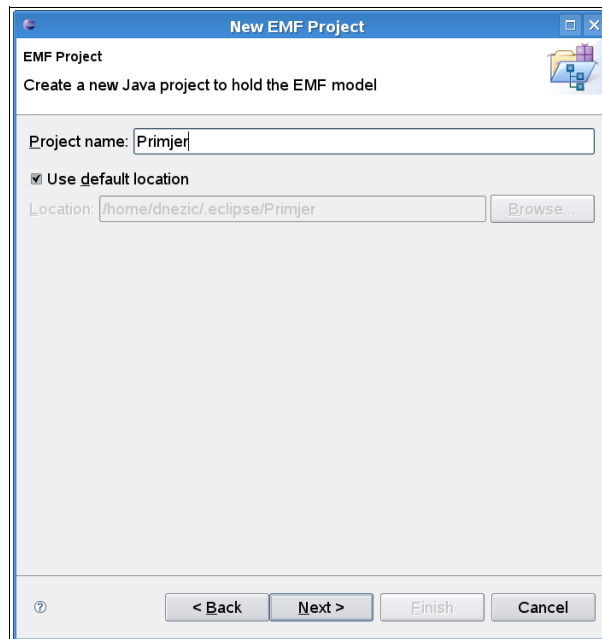
2.1.3. Kreiranje EMF projekta

Unutar Eclipse IDE-a, odaberemo u izborniku “File/New/Project...” dijalog.



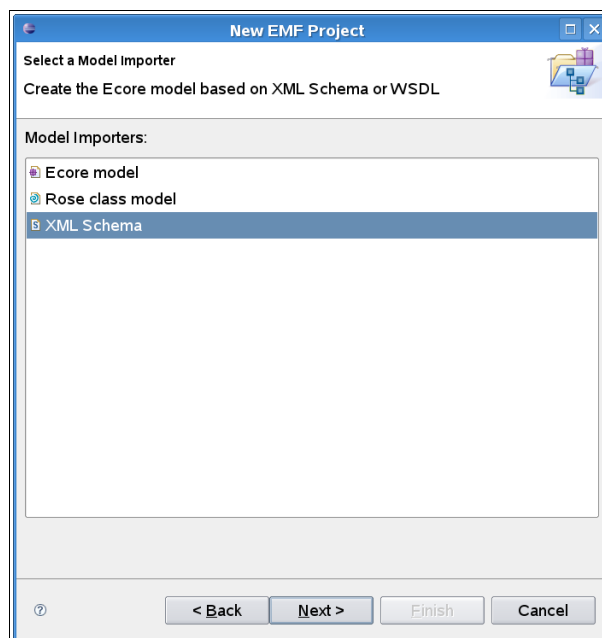
Slika 2.2: Odabir novog EMF projekta

Odabirom projekta “EMF Project”, pritiskom na dugme “Next” u novom dijalogu odaberimo ime projekta (*Primjer*):



Slika 2.3: Odabir naziva novog projekta

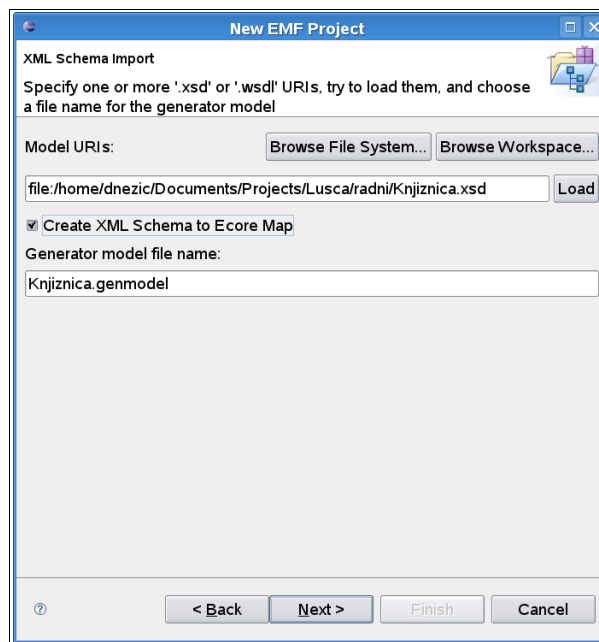
Pritiskom na dugme “Next”, u novom izborniku, odaberemo stavku “XML Schema”, što će predstavljati naš model knjižnice opisan XSD datotekom, XML shemom.



Slika 2.4: Odabir Ecore modela baziranog na XML shemi

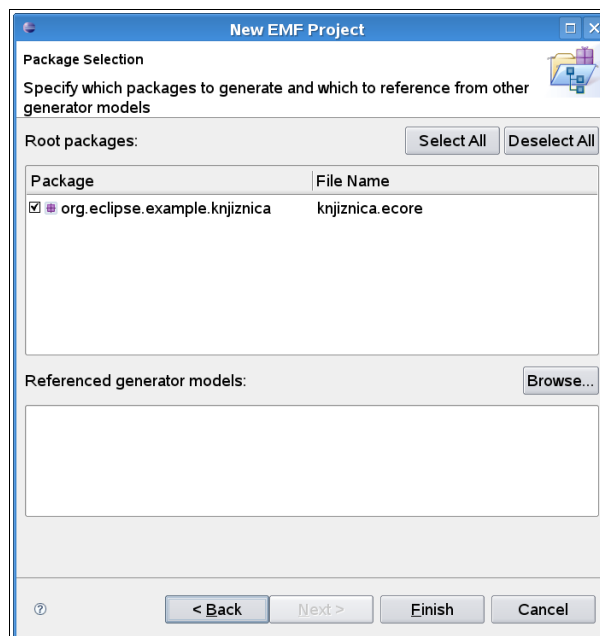
U slijedećem izborniku moramo upisati putanju do naše datoteke *Knjiznica.xsd*, koju smo prethodno spremili, nakon čega će se validirati njen format. Automatski će se predložiti naziv model generatora,

Knjiznica.genmodel, te odabirom opcije “Create XML Schema to Ecore Map”, generirati će se poveznica Ecore modela s našom XML shemom.



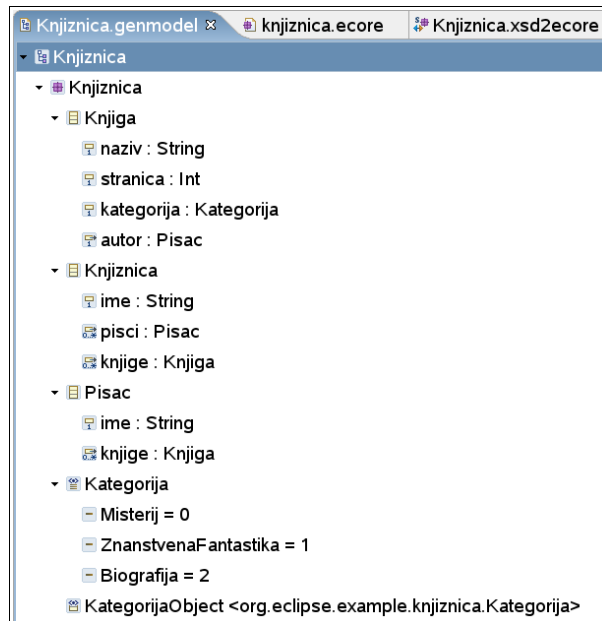
Slika 2.5: Odabir "XML Schema" datoteke

S obzirom da više od jednog paketa može biti kreirano iz jedne XML sheme, ako uključuje različite prostore imena (engl. *namespace*), potrebno je odabrati paket za koji želimo generirati EMF model, (slika 2.6).



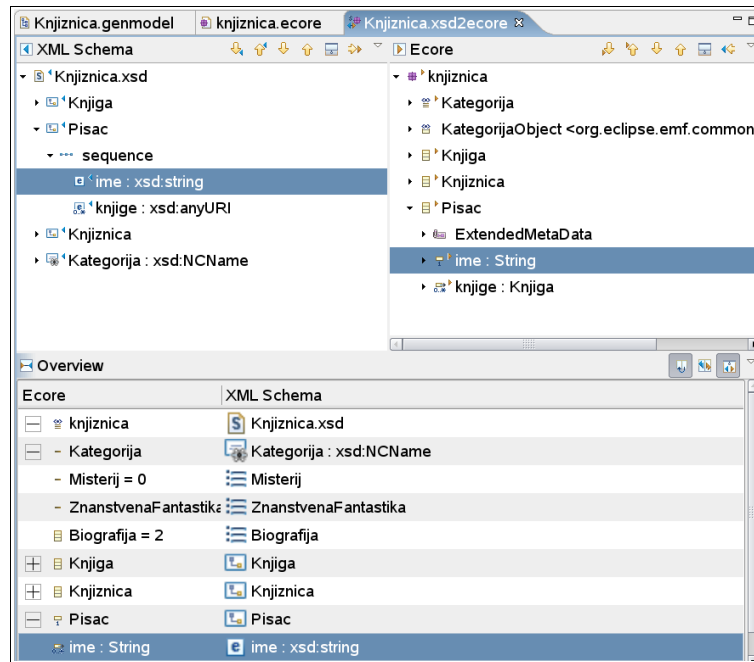
Slika 2.6: Odabir paketa za koji želimo generirati EMF model

Pritiskom na dugme “Finish”, generirati će se *core model* (*knjiznica.ecore*), *generator model* (*Knjiznica.genmodel*) i, ukoliko smo odabrali opciju u izborniku sa slike 2.5, “Create XML Schema to Ecore Map”, datoteka *Knjiznica.xsd2ecore*, koja točno prikazuje kako je XML shema povezana (engl. *mapped*) s *Ecoreom* (str. 3). *Generator model* prikazuje korijenski objekt (engl. *root object*) koji predstavlja cijeli model. Djeca ovog objekta predstavljaju njegove pakete, čija djeca predstavljaju klasifikatore (razredi, podatkovni tipovi, pobrojani tipovi). Djeca razreda su atributi razreda, reference i operacije. Djeca pobrojanih tipova su literalni. *Generator model* kontrolira kreiranje EMF modela, a *Ecore model* služi kao *meta model* za bilo koji EMF model koji kreiramo.



Slika 2.7: Generator model

Na slici 2.8 prikazana je *Knjiznica.xsd2ecore* datoteka koja opisuje kako je *Generator model*, opisan XML shemom, XSD datotekom, povezan sa *Ecore* generičnim modelom.



Slika 2.8: xsd2ecore datoteka

Odmah je uočljivo da je tip *xsd:string* iz XML datoteke, kojim je predstavljeno ime pisca, u *Ecore* modelu predstavljen Javinim tipom *String*. Osnovna pravila mapiranja (engl. *mapping rules*) iz XML sheme u *Ecore* su:

- *Schema* se mapira u *EPackage*. Svaka dodatno uključena shema koja deklarira različiti prostor imena mapira se u vlastiti, novi, *EPackage*.
- *ComplexType* tip se mapira u *EClass*.
- *SimpleType* tip se mapira u *EDataType*.
- Deklaracija atributa ili deklaracija ugnježđenog elementa mapira se u *EAttribute* ili *EReference*, ovisno o njegovom tipu.
- *EClass* imena *DocumentRoot* je kreiran kako bi sadržavao bilo koji element u gornjem nivou ili deklaracije atributa.

Iz perspektive modeliranja, XML shema nije toliko moćna i ekspresivna kao *Ecore*, ne može specificirati tip reference ili definirati bidirekionalne reference, no EMF omogućuje načine da se nadvladaju ova ograničenja (npr. *ecore:reference* atribut).

Osim XML sheme, EMF model može se definirati direktno editiranjem XMI dokumenta (zahtjeva poznavanje sintakse), uporabom nekog alata u kojem ćemo moći nacrtati model (UML) te naposljetku, direktno, iskorirajući nepotpuna Java sučelja koje EMF “razumije” i može pridodati ostatku modela. Ova posljednja metoda u dokumentaciji se navodi kao *Annotated Java*. Prema našem primjeru, model *Knjiga* u *Annotated Javi* izgledao bi:

```
/**
 * @model
 */
public interface Knjiga
{
    /**
     * @model
     */
    String getNaziv();
}
```

```
/**
 * @model
 */
int getStranica();
}
```

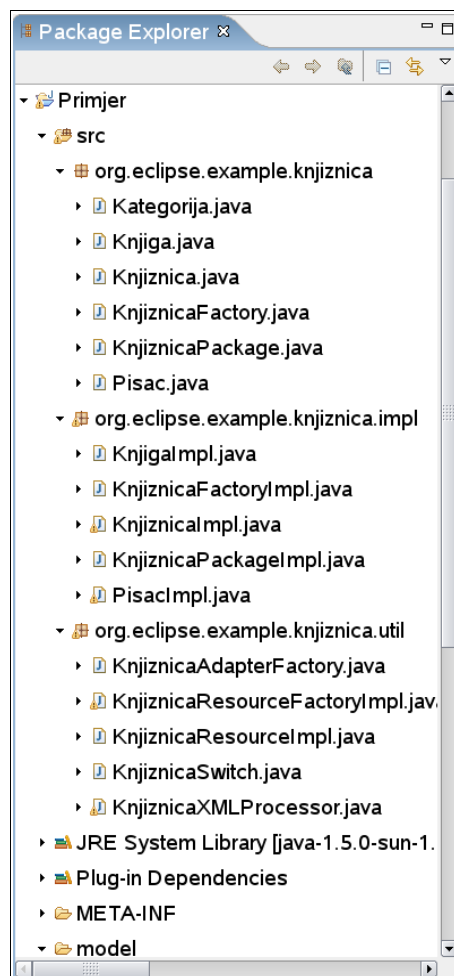
Ovim pristupom definirali smo sve informacije o modelu u obliku Java sučelja standardnim *get* metodama. Oznaka “@model” označava generatoru koda koja sučelja i dijelovi sučelja su dijelovi modela te stoga zahtijevaju generaciju koda. Ako želimo nekom atributu dodati posebno svojstvo, npr. sa zabranom na pisanje (*read-only*), dodat ćemo slijedeći parametar:

```
/**
 * @model changeable="false"
 */
int getStranica();
```

2.1.4. Generiranje koda iz EMF modela

Za svaki objekt u *Generator modelu* možemo vidjeti njegova svojstva. Pritiskom desne tipke miša na objekt odaberemo opciju “Show Properties View”. Prikazana svojstva kontroliraju ponašanje generatora koda. U većini slučajeva neće biti potrebno mijenjati početne vrijednosti svojstava.

U *Generator modelu* može se inicirati generiranje koda, odabirom nekog objekta u modelu te pritiskom desne tipke miša u novootvorenom izborniku odabirom opcije “Generate Model Code”. Nakon generiranja biti će kreirana sučelja razreda i pobrojanih tipova te novi parovi sučelja, za pakete i njihove tvornice razreda (engl. *factory*). Osim njih, kreirat će se i dva nova paketa s *impl* i *util* nastavcima, koja sadržavaju implementacije sučelja i dodatne pomoćne razrede te manifest datoteku modela, *plugin.xml* (slika 2.9).



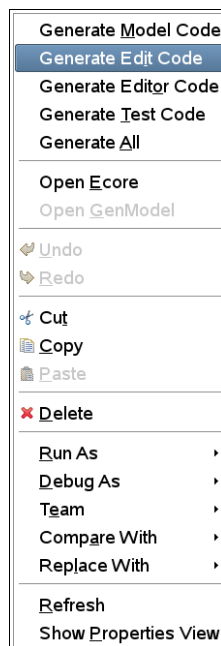
Slika 2.9: Prikaz projektnog stabla nakon generiranja koda iz EMF modela

Ukoliko se želi promijeniti model, kod se može iznova generirati i promjene u modelu i kodu će biti prepoznate i spojene.

2.1.5. Generiranje editora za model

Osim koda, prema EFM modelu može se generirati i Eclipse editor. U osnovi, podijeljen je na dva dodatka: *edit* dodatak, s prilagodnicima (engl. *adapters*) koji omogućavaju strukturni pogled i komandno-linijsko editiranje objekata modela; *editor* dodatak, koji omogućuje UI.

U *generator modelu*, odaberimo vršni objekt modela iz primjera te opciju “Generate Edit Code”. Potom nad istim objektom odaberimo opciju “Generate Editor Code”.

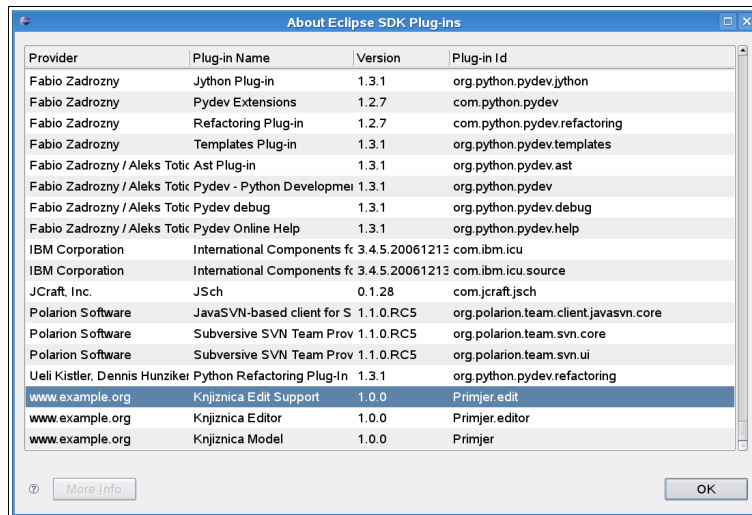


Slika 2.10: Generiranje editora

U pogledu “Package Explorer” generirani su projekti s nastavcima *edit* i *editor* (*Primjer.edit* i *Primjer.editor*). Odabirom opcije “Generate All”, model, edit i editor dodatci se mogu generirati jednim korakom. Nakon toga iz glavnog izbornika odaberimo opciju “Project/Build All”.

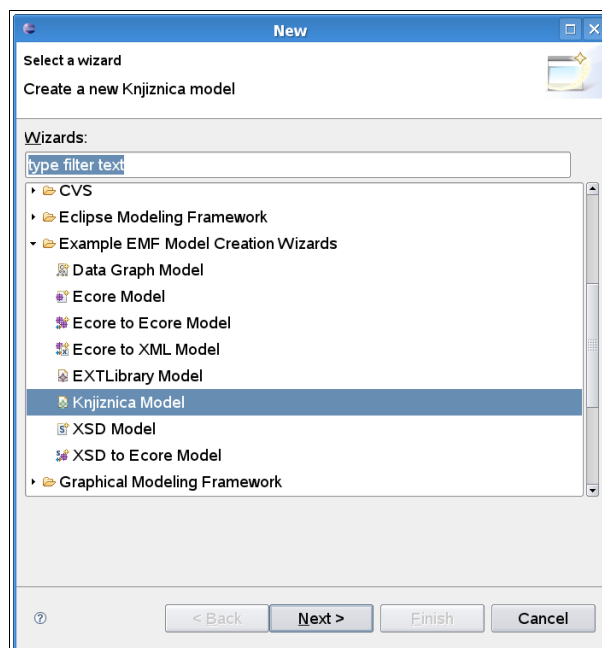
2.1.6. Pokretanje generiranog editora

Da bi mogli testirati nove dodatke, *plug-inove*, mora se pokrenuti nova instanca Eclipse IDE-a. Stoga, naš projekt *Primjer* moramo pokrenuti opcijom iz glavnog izbornika “Run/Run As/Eclipse Application”. U novootvorenojinstanci Eclipse IDE-a, u izborniku “Help/About Eclipse SDK”, u listi instaliranih dodataka, na dnu, možemo vidjeti naše dodatke projekta *Primjer*.



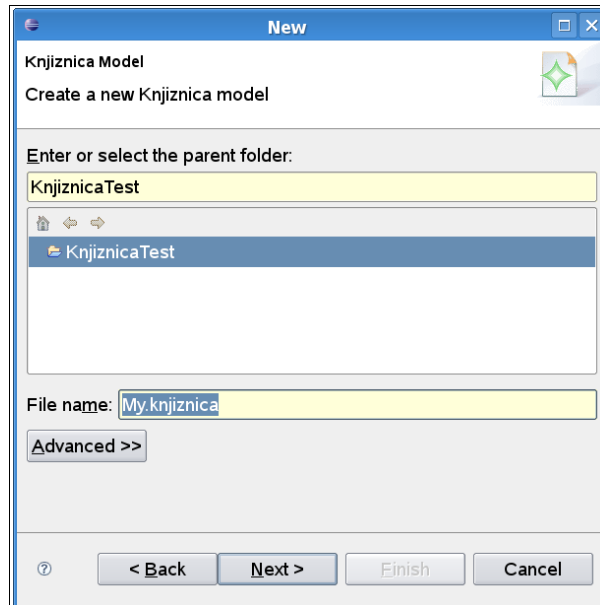
Slika 2.11: "Primjer" dodatak u listi dodataka nove instance Eclipse IDE-a

Sada možemo kreirati novi projekt opcijom izbornika: “File/New/Project..”, a u novom prozoru odaberemo opciju, u mapi *General*, “Project”. Nazovimo novi projekt npr. *KnjiznicaTest*. Potom kreiramo novu instancu modela *Knjiznica* odabirom opcije glavnog izbornika “File/New/Other...”. U izborniku odaberemo, u mapi “Example EMF Model Creation Wizards”, “Knjiznica Model”, što je prikazano na slici 2.12.



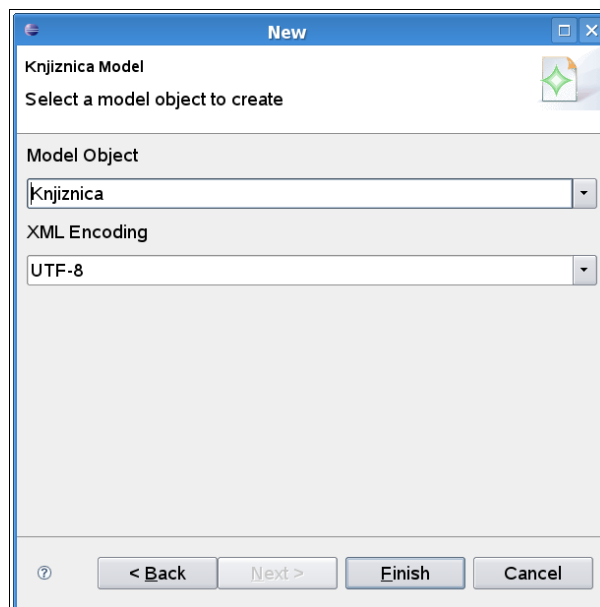
Slika 2.12: Kreiranje novog modela "Knjiznica"

Novi *Knjiznica* model naziva “My.knjiznica” pripadati će projektu *KnjiznicaTest*.



Slika 2.13: Odabir imena novog modela i pridodavanje istog projektu "KnjiznicaTest"

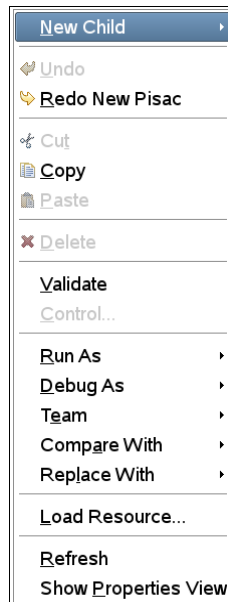
Pritiskom na dugme “Next” u novom izborniku odabiremo koji model želimo kreirati, na raspolaganju su nam: *Knjiga*, *Knjiznica* i *Pisac*.



Slika 2.14: Odabir modela koji se želi kreirati

Pritiskom na dugme “Finish” model je kreiran i pridodan je projektu *KnjiznicaTest* kao model *My.knjiznica*. Otvorimo li model *My.knjiznica*, u izborniku “Resource Set” možemo vidjeti trenutne objekte modela, na početku je to samo objekt *Knjiznica*. U tabulatoru “Properties” možemo vidjeti svojstva atributa objekta *Knjiznica*, u obliku <Property:Value>. Promijenimo vrijednost svojstva

knjižnice, “Ime” u “Glavna knjižnica”. Potom desnim klikom miša na objekt *Knjižnica* u “Resource Setu” modela *My.knjižnica* u novootvorenom izborniku možemo dodavati nove objekte knjižnice, pisce i knjige (opcija “New Child”).



Slika 2.15:
Dodavanje
podobjekata, djece

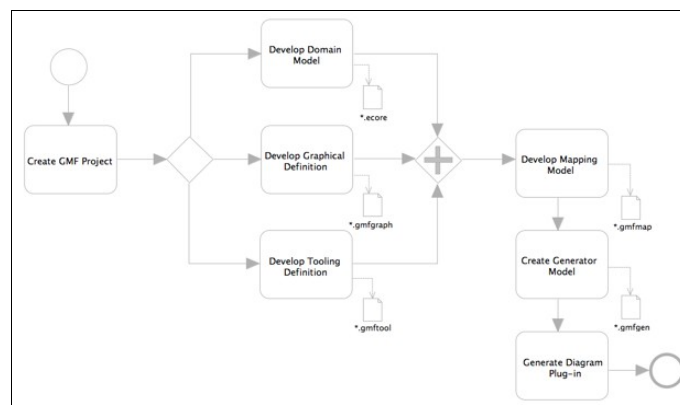
Nakon što smo unijeli pisce i knjige možemo naš model *My.knjižnica* otvoriti pomoću uređivača teksta i na taj način postići XML serijalizaciju, opcijom “Open With/Text Editor”. Rezultat XML serijalizacije:

```
<?xml version="1.0" encoding="UTF-8"?>
<knjiznica:Knjiznica xmlns:knjiznica="http://www.example.eclipse.org/Knjiznica">
  <ime>Glavna knjiznica</ime>
  <pisci>
    <ime>Isaak Asimov</ime>
    <knjige>#//@knjige.0</knjige>
    <knjige>#//@knjige.1</knjige>
  </pisci>
  <knjige>
    <naziv>Foundation</naziv>
    <stranica>236</stranica>
    <kategorija>ZnanstvenaFantastika</kategorija>
    <autor>#//@pisci.0</autor>
  </knjige>
  <knjige>
    <naziv>Foundation And Empire</naziv>
    <stranica>224</stranica>
    <kategorija>ZnanstvenaFantastika</kategorija>
    <autor>#//@pisci.0</autor>
  </knjige>
</knjiznica:Knjiznica>
```

3. Graphical Modeling Framework

Graphical Modeling Framework je okruženje za izgradnju grafičkih editora modela unutar Eclipsea. Primjeri takvih editora su UML editori, ECore editori, editori dijagrama toka, XSD editori, editori dijagrama poslovnih procesa, itd. Dakle, Eclipseov EMF editor možemo “unaprijediti” grafičkim editorom.

Osnovne komponente i modeli koji se koriste tijekom razvoja *Graphical Modeling Framework* editora prikazani su na slici 3.1.



Slika 3.1: Pregled Graphical Modeling Frameworka

Glavni dio GMF-a je *Graphical Definition* model. On sadrži informacije o grafičkim elementima koji će se pojaviti u, naposljetku, generiranom pluginu, editoru. Sam za sebe, ni na koji način nije povezan s *Domain* modelima, za koje će omogućiti grafičko editiranje. *Domain* model je *ecore* model koji smo prethodno generirali u 2. poglavlju. *Tooling Definition* model je opcionalna komponenta koja sadrži specifikaciju dizajna izbornika i alatnih traka koji će se koristiti u editoru.

Osnovna ideja GMF-a je omogućiti *Graphical Definition* i *Tooling Definition* modelima uporabu nad više različitih (sličnih po strukturi i pojavnosti) *Domain* modela. Takva općenitost omogućuje se uporabom *Mapping* modela, posebno definiranog, u svrhu povezivanja definiranih grafičkih komponenti s komponentama modela i njihovim relacijama. Nakon definicije *Mapping* modela, GMF omogućuje *Generator* model u kojem se definiraju detalji implementacije za konačnu fazu generacije. Cilj generacije, koja se odvija prema *Generator* modelu, je *diagram runtime*, odnosno, dodatak za Eclipse, *plug-in*, koji će omogućiti grafičko generiranje i editiranje EMF *ecore* modela.

Prednosti koje imamo ako baziramo vlastiti editor na GMF okruženju su:

- Konzistentni izgled u odnosu na ostale editore bazirane na GMF-u
- Editor može biti izgrađen manualno metodom “od dna prema vrhu” ili uporabom raznih “čarobnjaka” koji se nalaze unutar GMF SDK¹
- Korisnik ne mora brinuti o razvoju editora već se može bolje koncentrirati na model koji izgrađuje
- Proširive je prirode, moguće ga je lako nadograditi

¹ Software Development Kit – Skup alata za izgradnju softwarea

- Njegove mogućnosti su vrlo dobro dizajnirane i testirane
- Buduće nadogradnje moći će se lako integrirati u već postojeći naš editor

Pregled osnovnih mogućnosti GMF editora prikazani su u poglavlju 4.

3.1. Potrebni GMF dodaci za Eclipse

Kako bi mogli koristiti GMF, moramo unutar Eclipse razvojnog okruženja instalirati slijedeće dodatke:

- Graphical Modeling Framework (Callisto Edition)

dodatno će se uz taj dodatak instalirati:

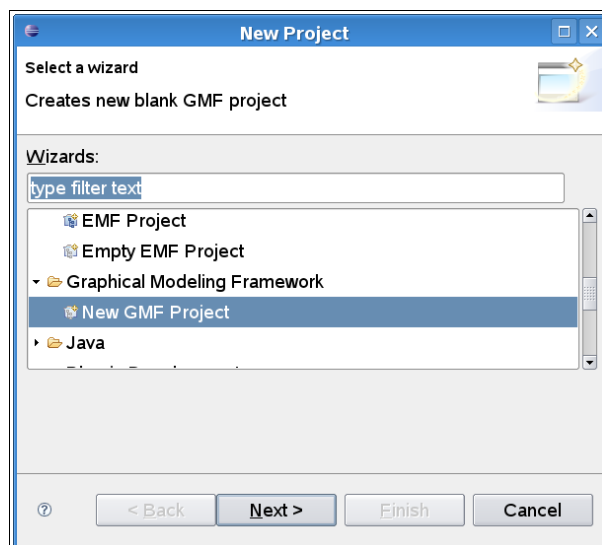
- Eclipse Modeling Framework Technology Query
- Eclipse Modeling Framework Technology Query OCL
- Eclipse Modeling Framework Technology Query Transactions
- Eclipse Modeling Framework Technology Query Validation
- Eclipse Modeling Framework Technology Query Validation OCL
- Eclipse Modeling Framework Technology Workbench Integration
- Graphical Modeling Framework Runtime
- Object Constraint Language (OCL)

S obzirom da je GMF relativno nova tehnologija preporučljivo je učestalo nadograđivati ove pakete jer su podložni čestim izmjenama.

3.2. Primjer uporabe GMF Frameworka

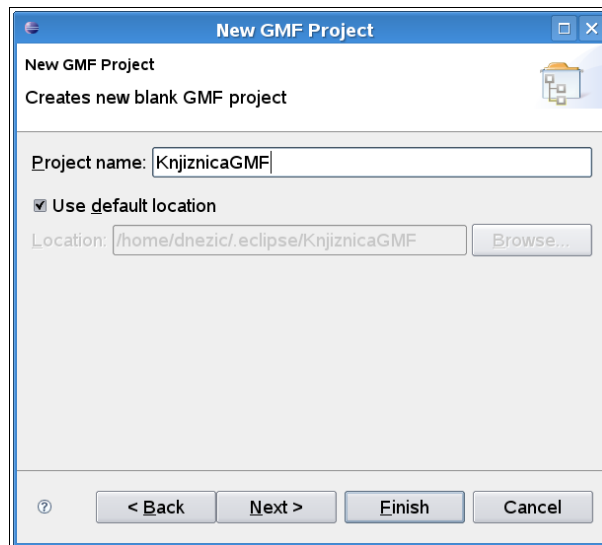
Nakon što smo generirali vlastiti EMF model, u 2. poglavlju, možemo izgraditi grafički editor koji će nam omogućiti dodavanje i editiranje komponenti prema tom modelu.

Odaberemo iz glavnog izbornika opciju “File/New Project/Graphical Modeling Framework/New GMF Project”.



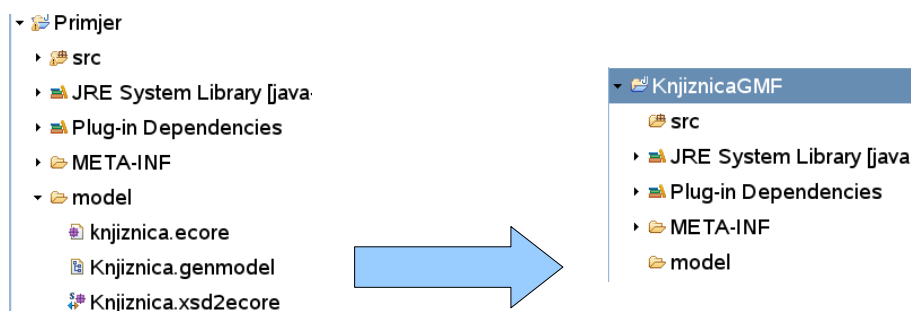
Slika 3.2: Kreiranje novog GMF projekta

U slijedećem izborniku odabiremo ime našeg GMF projekta, npr. “KnjiznicaGMF”.



Slika 3.3: Pridjeljivanje naziva GMF projektu

Pritiskom na dugme “Finish” kreirat će se novi GMF projekt unutar Eclipse okruženja.

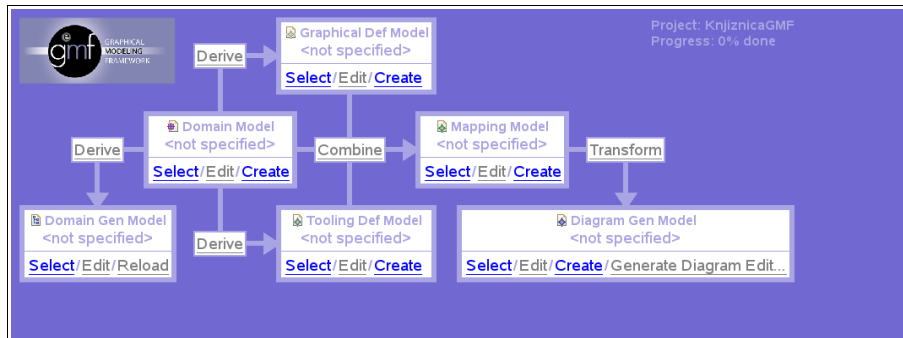


Slika 3.4: Prebacivanje modela iz EMF u novi GMF projekt

Prebacimo, kopiramo, datoteke s nastavcima “.ecore” i “.genmodel” iz našeg EMF projekta *Primjer* u direktorij “model” novog GMF projekta *KnjiznicaGMF*. Dakle, prema našem primjeru, nakon prebacivanja u direktoriju “model”, projekta *KnjiznicaGMF*, nalaze se datoteke *knjiznica.ecore* i *Knjiznica.genmodel*. Ove datoteke su nam neophodne pri generiranju GMF editora za naš model jer ga opisuju. Prisjetimo se, datoteka *Knjiznica.xsd2ecore* opisuje na koji će se način izvršiti transformacija iz XSD datoteke kojom smo opisali model u Eclipseov *Ecore meta model*.

3.2.1. GMF Dashboard

Odmah nakon što smo kreirali novi GMF projekt možemo otvoriti pogled naziva *GMF Dashboard*. Ovaj dijagram pokazuje i objašnjava sve korake i komponente koji su nam potrebni u izgradnji našeg grafičkog editora modela. Direktno, u ovom prozoru, možemo izvršiti sve neophodne procese.

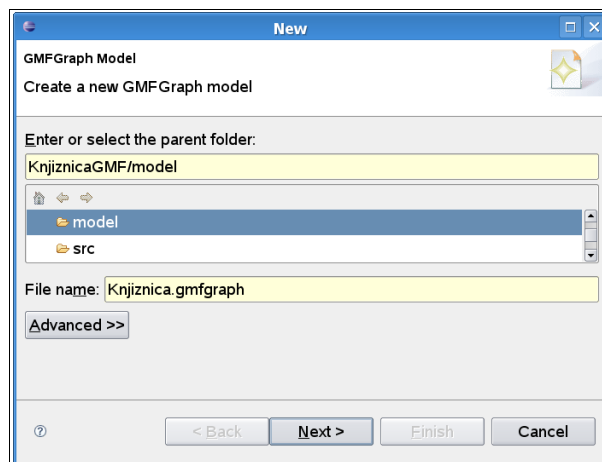


Slika 3.5: GMF Dashboard

Domain Gen Model i *Domain Model* već imamo, to su EMF modeli *Knjiznica.genmodel* i *knjiznica.ecore* koje smo kreirali u poglavlju 2.

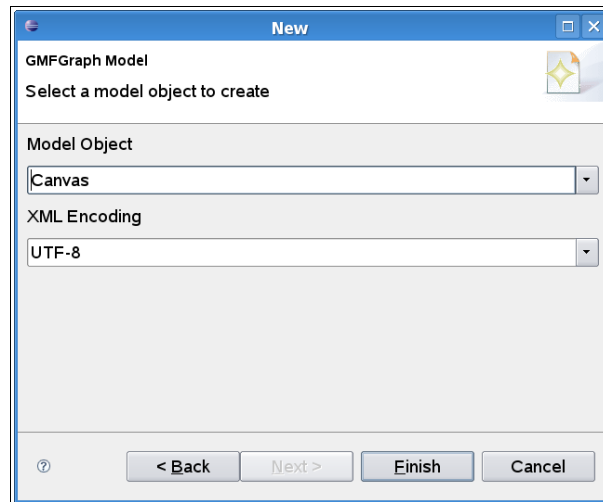
3.2.2. Graphical Definition Model

Prvi model koji je preporučljivo kreirati je *Graphical Definition Model*. *Graphical Definition Model* sadrži definiciju svih grafičkih objekata koje ćemo koristiti u grafičkom editoru našeg modela. U *GMF Dashboardu* odaberimo opciju “Create”, za “Graphical Def Model”. Kao na slici 3.6., smjestimo model u mapu “KnjiznicaGMF/model”.



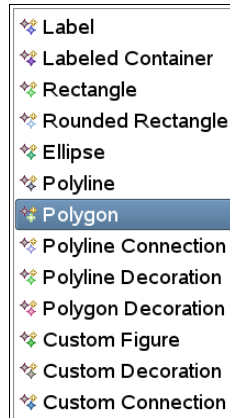
Slika 3.6: Novi Graphical Definition Model

U slijedećem izborniku potom izaberemo jedan od ponuđenih model objekata, u našem primjeru neka to bude *Canvas*, slika 3.7. Taj model će zadovoljiti naše potrebe, za primjer modela *Knjiznica*.



Slika 3.7: Odabir model objekta

U projektnom stablu generirana je datoteka *Knjiznica.gmfgraph*. U “Resource Set” pogledu ove datoteke možemo vidjeti glavni element *Canvas* kojemu možemo, u “Properties” tabulatoru, nadjenuti ime. Pritiskom desne tipke miša na element *Canvas* dodajmo mu dijete, opcijom “New Child/Figure Gallery”. “Figure Gallery” sadrži grafičke elemente koje ćemo koristiti za grafički prikaz našeg modela. Također, u izborniku elementa “Figure Gallery” možemo odabrati “New Child” od ponuđenih na slici 3.8.



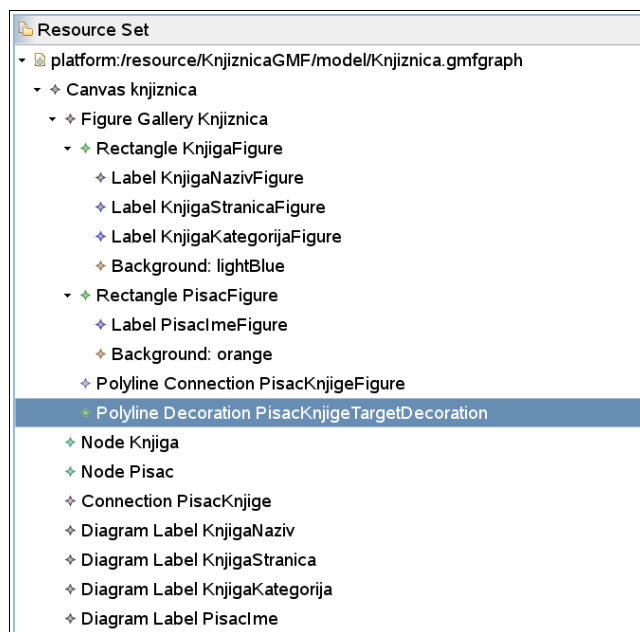
Slika 3.8: Elementi koje sadrži "Figure Gallery"

Elementi koji će biti dovoljni za opis našeg modela su npr. *Rectangle* i “*Polyline Connection*”. Za svaki posebno u “Properties” pogledu možemo definirati opcije prikaza i ime. Nazovimo ih *KnjigaFigure*, *PisacFigure* za tip *Rectangle* te *PisacKnjigeFigure* za tip “*Polyline Connection*”.

Nakon što smo dodali elemente prikaza, sada trebamo dodati elemente modela koji će se prikazivati ovim grafičkim elementima. Pritiskom desne tipke miša na element *Canvas*, pridodajmo mu dva nova djeteta tipa *Node*, jedan za *Knjigu* i drugi za *Pisca*, i jedno dijete tipa *Connection*. Nazovimo elemente tipa *Node* “*Knjiga*”, odnosno “*Pisac*” te pridodajmo im, u izborniku “Properties”, atribut *Figures* “*Rectangle KnjigaFigure*”, odnosno “*Rectangle PisacFigure*” za element *Pisac*. Također, element *Connection* imenujmo “*PisacKnjige*”, a njegov atribut *Figure* postavimo na “*Polyline Connection*”.

PisacKnjigeFigure”. Nakon što smo definirali sve objekte koje ćemo koristiti poželjno je u izborniku objekta *Canvas* opcijom “Validate” provjeriti korektnost modela.

Grafička definicija modela naposljetku izgleda ovako:

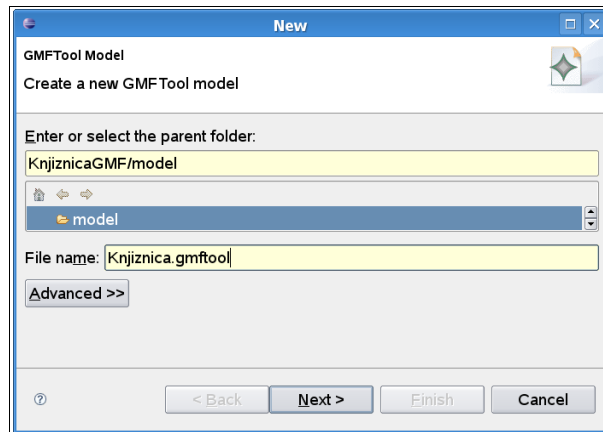


Slika 3.9: Grafička definicija modela

Kao što je prikazano na slici, dodali smo još nekoliko elemenata tipa *Label* te svojstvo *Background* kao dodatna svojstva objekata *KnjigaFigure* i *PisacFigure*. Osim toga dodali smo objekt tipa “*Polyline Decoration PisacKnjigeTargetDecoration*” koji definira izgled objekta “*Polyline Connection PisacKnjigeFigure*”. Objekti “*Diagram Label*” potrebni su nam da bi mogli preko njih mapirati objekte tipa *Label*, to će biti, na neki način, instance tipova *Label*, kao što je npr. *Node* imenovana instanca definirana tipom *Rectangle*.

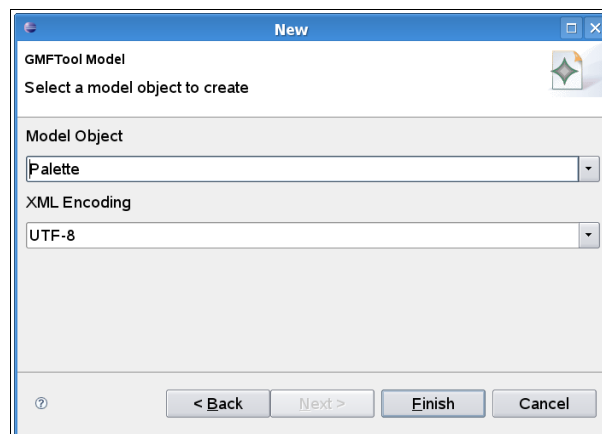
3.2.3. Tooling Definition Model

Ovaj model opisuje kako će izgledati naš *plug-in*, odnosno na koji način, u kojem obliku, će se moći koristiti grafički editor. Unutar *GMF Dashboarda* odaberemo opciju “Create” za “Tooling Def Model”. Nazovimo novi model ***Knjiznica.gmftool***.



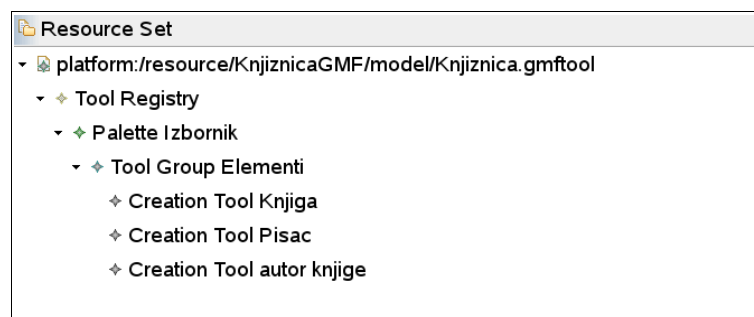
Slika 3.10: Dodavanje Tooling Definition modela

U novom prozoru odabiremo vrstu prozora, tip elementa, unutar kojeg će se u našem editoru nalaziti objekti koje ćemo koristiti za grafičko editiranje našeg modela. U ovom primjeru odaberimo tip *Palette*.



Slika 3.11: Odabir Model objekta

Za potrebe ovog primjera, elementu “*Tool Registry*” dodajmo element *Palette*, a svaki element tipa “*Creation Tool*” će se moći odabrati u *Palette* izborniku našeg *plug-ina*.



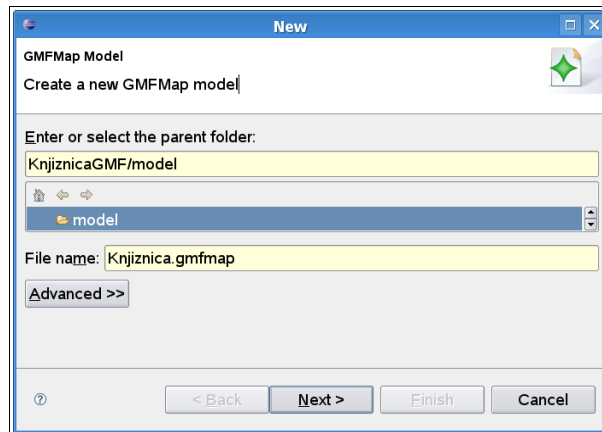
Slika 3.12: Izgled Tool modela

Svakom elementu tog tipa poželjno je zadati opis, koji će se pojaviti kao poruka kad pozicioniramo miš na taj objekt u izborniku, te ime kojim će biti predstavljen.

3.2.4. Mapping model

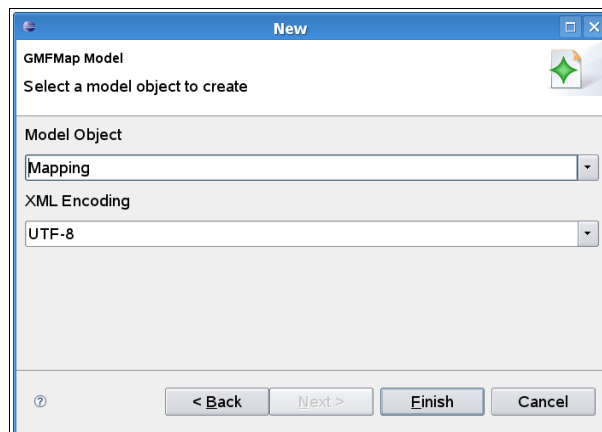
Mapping model će nam omogućiti da povežemo dosadašnja tri modela, *Ecore (opis modela)*, *Graphical Definition* i *Tooling Definition* model.

Kreirajmo novi *Mapping* model u *GMF Dashboardu*.



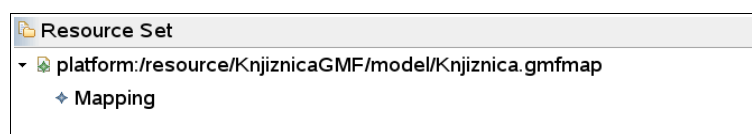
Slika 3.13: Novi GMF Mapping model

Smjestimo novi *Mapping* model u mapu “model” projekta *KnjiznicaGMF*.



Slika 3.14: Generiranje novog Mapping modela

Dodajmo modelu objekt *Mapping* koji će biti osnovni element modela, što je vidljivo u pogledu “Resource Set” (slike 3.14. i 3.15.).



Slika 3.15: Resource Set Mapping modela

U izborniku elementa *Mapping*, odaberimo opciju “Load Resource/Browse Workspace” te iz mape “model” učitajmo *Tooling* model *Knjiznica.gmftool*, *Graphical Definition* model *Knjiznica.gmfgraph* i *Ecore* model *Knjiznica.ecore*.

Dodajmo elementu *Mapping* dijete “*Canvas Mapping*”. Svojstva elementa “*Canvas Mapping*” u pogledu “*Properties*” postavimo na slijedeći način:

- Domain meta information
 - Domain Model: EPackage knjiznica
 - Element: EClass Knjiznica
- Misc
 - Palette: Palette Izbornik
- Visual representation
 - Diagram Canvas: Canvas knjiznica

Nakon toga, dodajmo element “*Top Node Reference*”. Kako bi nam prvi “*Top Node Reference*” predstavljao *Pisac* moramo tom elementu dodati dijete “*Node Mapping*” sa slijedećim svojstvima:

- Domain meta information
 - Element: EClass Pisac
- Visual representation
 - Diagram Node: Node Pisac
 - Tool: Creation Tool Pisac

Svojstva “*Top Node Reference*” objekta potrebno je postaviti:

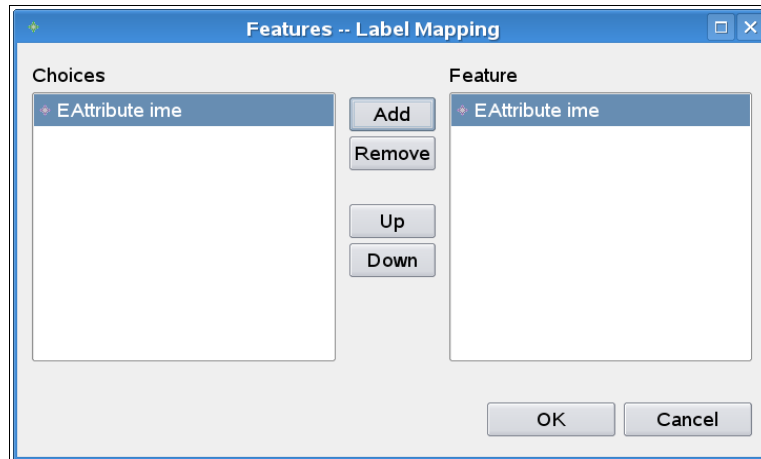
- Child: Node Mapping <Pisac/Pisac>
- Containment Feature: EReference pisaci

Potom elementu “*Node Mapping*” <Pisac/Pisac> možemo dodati element “*Label Mapping*” sa svojstvima:

- Diagram Label: Diagram Label PisacIme
- Features: EAttribute ime
- ViewPattern: Ime: {0}

Ovdje je uočljiv predložak '{0}' koji će zamijeniti korisnički unešenu varijablu. Koji će atribut ta varijabla predstavljati odaberemo u izborniku svojstva *Features*, slika 3.16. Ukoliko istoj labeli želimo dodati više atributa uporabiti ćemo na primjer slijedeći predložak:

- ViewPattern: Ulica: {0}, Broj: {1} ... itd.



Slika 3.16: Podešavanje Label Mapping objekta

Osim *Pisca* potrebne su nam i *Knjige*, stoga moramo dodati još jedan “*Top Node Reference*” element s “*Node Mapping*” elementom <Knjiga/Knjiga>, svojstva tog elementa su:

- Domain meta information
 - Element: EClass Knjiga
- Visual representation
 - Diagram Node: Node Knjiga
 - Tool: Creation Tool Knjiga

uz tri labele sa svojstvima:

- Diagram Label: Diagram Label KnjigaNaziv
- Features: EAttribute naziv
- ViewPattern: Naziv: {0}

za kategoriju:

- Diagram Label: Diagram Label KnjigaKategorija
- Features: EAttribute kategorija
- ViewPattern: Kategorija: {0}

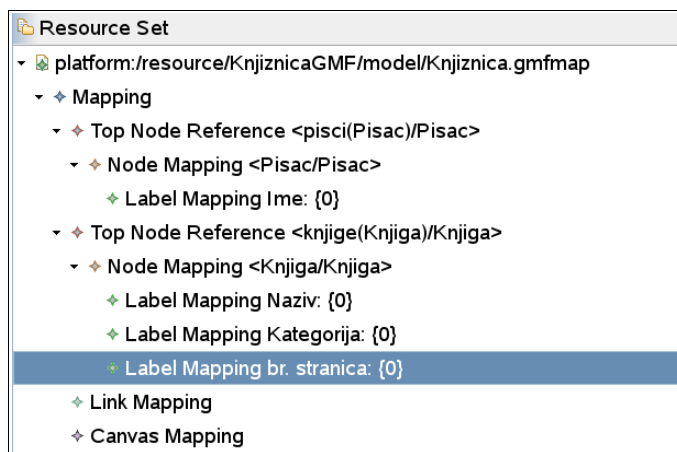
i broj stranica:

- Diagram Label: Diagram Label KnjigaStranica
- Features: EAttribute stranica
- ViewPattern: br. stranica: {0}

Osim čvorova, *Nodes*, potrebna nam je i poveznica između njih za što će nam poslužiti “*Link Mapping*” objekt. Svojstva tog objekta potrebno je podesiti:

- Target Feature: EReference knjige
- Diagram Link: Connection PisacKnjige
- Tool: Creation Tool autor knjige

Ovom poveznicom možemo pridodjeljivati knjige piscima. Konačni izgled Mapping modela prikazan je na slici 3.17.



Slika 3.17: Mapping model

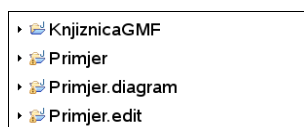
Umjesto novog praznog GMF *Mapping* modela, kao u ovom primjeru, možemo kreirati i model opcijom “File/New/Graphical Modeling Framework/Guide GMFMap”. U tom će slučaju većina povezivanja biti obavljena automatski kroz nekoliko dijaloga. Naknadno možemo “ručno” intervenirati ukoliko nismo u potpunosti zadovoljni rezultatom.

3.2.5. Kreiranje GMF generator modela i grafičkog editora

Nakon što smo definirali *Mapping* model možemo pristupiti kreiranju *Generator* modela, opcijom “Create” u “Diagram Gen Model” izborniku *GMF Dashboarda* ili opcijom “Create Generator model...” u izborniku *Mapping* modela, *Knjiznica.gmfmap* u mapi “model” projekta.

Ukoliko je proces bio uspješan, u mapi “model” projekta *KnjiznicaGMF* stvorit će se datoteka *Knjiznica.gmfgen*. Ova datoteka predstavlja *Generator* model i u njenom izborniku odaberimo opciju “Generate Diagram Code”, što možemo učiniti i iz *GMF Dashboarda*.

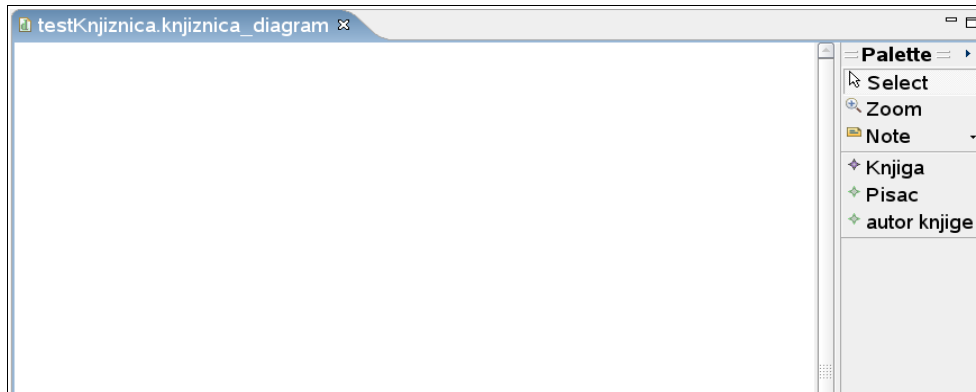
Po završetku procesa, u pogledu “Package Explorer” mora se nalaziti projekt *Primjer.diagram*.



Slika 3.18: Projekti potrebni za editor

Odaberimo projekt *KnjiznicaGMF* te ga pokrenimo kao Eclipse aplikaciju (“Run As/Eclipse Application”).

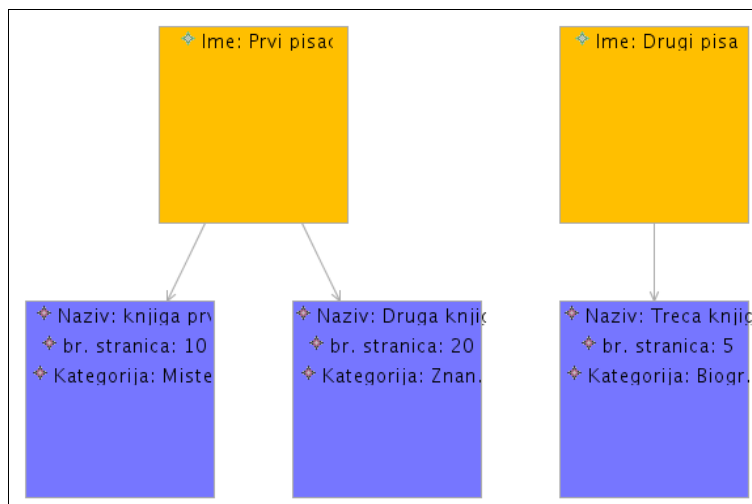
U novoj instanci Eclipse IDE-a kreirajmo novi “prazan” Eclipse projekt te mu dodajmo “Knjiznica Diagram”, opcijom “File/New/Other/Knjiznica Diagram”.



Slika 3.19: Izgled editor modela s radnom površinom

Sada možemo dodati neke elemente, pridodati im attribute te ih povezati poveznicom “*autor knjige*” iz *Palette* izbornika.

Kada smo završili sa dodavanjem elemenata dijagram možemo posložiti opcijom iz glavnog izbornika “Diagram/Arrange All/All”. Možemo ga snimiti kao sliku opcijom “File/Save As Image File” u izborniku, pritiskom desne tipke miša na radnu površinu.



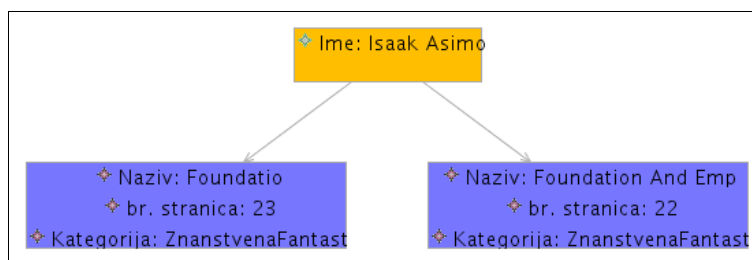
Slika 3.20: Izgled našeg dijagrama

Također, dijagram možemo serijalizirati, u format ovisno o vrsti modela u kojem je originalno definiran, u našem slučaju, *XSD Schema*. Otvorimo li datoteku *testKnjiznica.knjiznica* našeg dijagrama unutar nekog obrađivača teksta dobit ćemo serijaliziran dokument.

```
<?xml version="1.0" encoding="UTF-8"?>
<knjiznica:Knjiznica xmlns:knjiznica="http://www.example.eclipse.org/Knjiznica">
  <ime>Glavna knjiznica</ime>
  <pisci>
    <ime>Prvi pisac</ime>
    <knjige>#//@knjige.0</knjige>
    <knjige>#//@knjige.1</knjige>
  </pisci>
  <pisci>
    <ime>Drugi pisac</ime>
    <knjige>#//@knjige.2</knjige>
  </pisci>
</knjiznica:Knjiznica>
```

```
</pisci>
<knjige>
  <naziv>knjiga prva</naziv>
  <stranica>100</stranica>
  <autor>#//@pisci.0</autor>
</knjige>
<knjige>
  <naziv>Druga knjiga</naziv>
  <stranica>200</stranica>
  <kategorija>ZnanstvenaFantastika</kategorija>
  <autor>#//@pisci.0</autor>
</knjige>
<knjige>
  <naziv>Trecia knjiga</naziv>
  <stranica>50</stranica>
  <kategorija>Biografija</kategorija>
  <autor>#//@pisci.1</autor>
</knjige>
</knjiznica:Knjiznica>
```

Možemo jednostavno uzeti bilo koju, po strukturi jednaku ili dovoljno sličnu, XML datoteku i snimiti je u datoteku *testKnjiznica.knjiznica* te opcijom “Initialize knjiznica_diagram diagram file” generirati dijagram prikazan na slici 3.21.



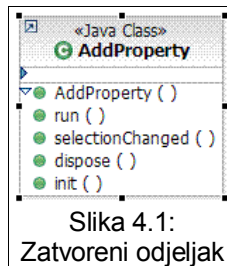
Slika 3.21: Dijagram generiran iz XML datoteke

4. Prikaz mogućnosti GMF editora

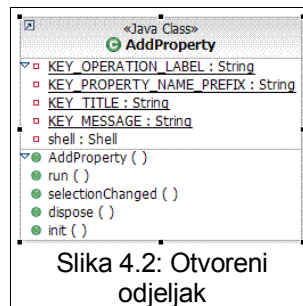
Grafički editori izgrađeni pomoću GMF-a unutar GMF *Runtime* okruženja nasljeđuju brojna korisna i vizualno dojmlija svojstva koje ono nudi.

4.1. Otvaranje i zatvaranje odjeljka s atributima

Composite ili *Compartment* elementi u dijagramu mogu se otvarati i zatvarati npr:



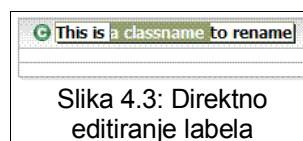
Slika 4.1:
Zatvoreni odjeljak



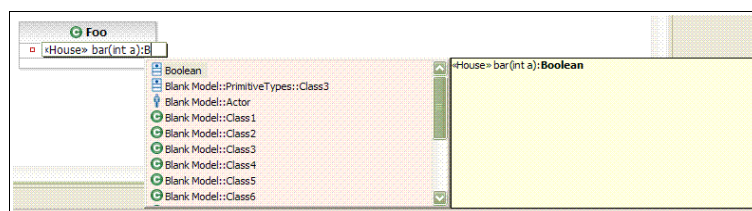
Slika 4.2: Otvoreni odjeljak

4.2. Direktno editiranje

Tekst unutar labela se može direktno editirati i interpretirati na način da ponudi skup komandi.



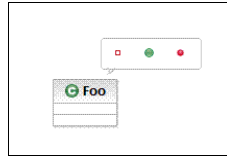
Slika 4.3: Direktno editiranje labela



Slika 4.4: Automatsko nadopunjavanje

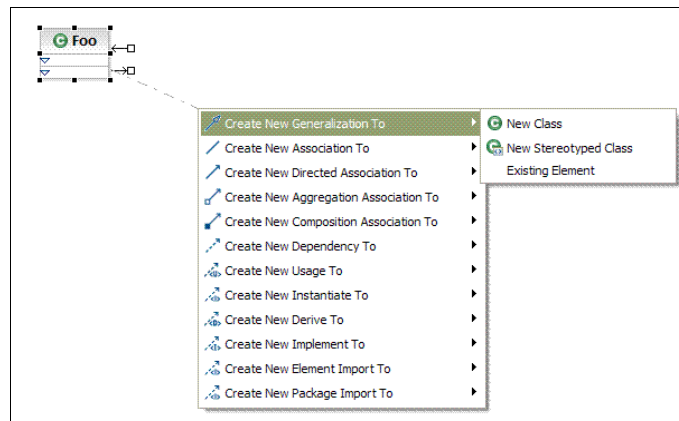
4.3. Diagram Assistants

Diagram Assistants su lebdeći objekti koji se prikazuju unutar radne površine nakon što korisnik određeno vrijeme ne pomiče miša. Postoje dvije vrste: *Pop-up Bars* i *Connection Handles*.



Slika 4.5: Pop-up Bar

Pop-up Bar lebdeći izbornik iznad radne površine omogućuje nam jednostavno dodavanje novih objekata u dijagram.

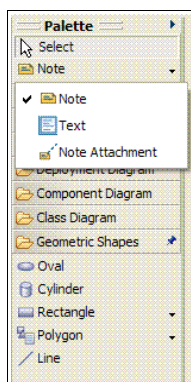


Slika 4.6: Connection handle

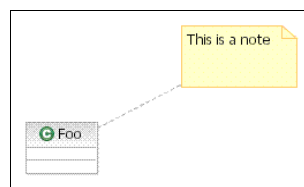
Osim za povezivanje objekata, *Connection Handle* može poslužiti i za generiranje novih objekata koji će odmah biti povezani s izvornim objektom.

4.4. Osnovni alati

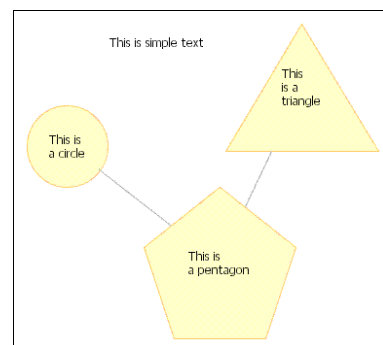
Osim vlastitih elemenata koje možemo dodavati dijagramu, možemo se poslužiti i nekim predefiniranim objektima koji se nalaze u osnovnom GMF *Runtime* okruženju.



Slika 4.7:
Osnovni objekti

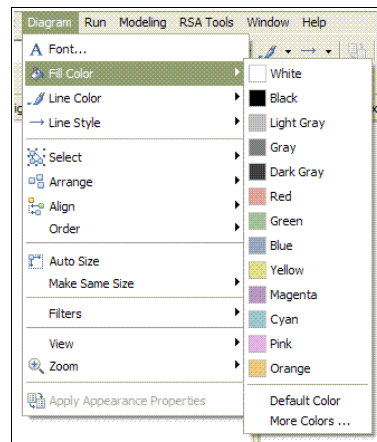


Slika 4.8: Bilješka



Slika 4.9: Osnovni oblici

Pored osnovnih oblika, u izborniku "Diagram" možemo podešavati neke od parametara koji definiraju vizualni izgled dijagrama.



Slika 4.10: Diagram izbornik

Promjenama *Zoom* nivoa i opcijom *Arrange* koje uključuju preraspored elemenata na dijagramu ili neku promjenu njihovog prikaza, valja naglasiti da se odvijaju animirano.

4.5. Izvoz dijagrama

Slika dijagrama se može izvesti u više formata: GIF, PNG, BMP, SVG, JPEG.

5. Dodatak A – ispis modela korištenih u primjeru

U ovom dodatku izloženi su ispisi glavnih datoteka kojima su opisani modeli iz primjera radi lakšeg utvrđivanja ispravnosti slijeđenja uputa.

5.1. knjiznica.ecore

```
<?xml version="1.0" encoding="UTF-8"?>
<ecore:EPackage xmi:version="2.0"
  xmlns:xmi="http://www.omg.org/XMI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ecore="http://www.eclipse.org/emf/2002/Ecore" name="knjiznica"
  nsURI="http://www.example.eclipse.org/Knjiznica" nsPrefix="knjiznica">
  <eClassifiers xsi:type="ecore:EEnum" name="Kategorija">
    <eAnnotations source="http://org.eclipse/emf/ecore/util/ExtendedMetaData">
      <details key="name" value="Kategorija"/>
    </eAnnotations>
    <eLiterals name="Misterij"/>
    <eLiterals name="ZnanstvenaFantastika" value="1"/>
    <eLiterals name="Biografija" value="2"/>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EDataType" name="KategorijaObject"
instanceClassName="org.eclipse.emf.common.util.Enumerator">
    <eAnnotations source="http://org.eclipse/emf/ecore/util/ExtendedMetaData">
      <details key="name" value="Kategorija:Object"/>
      <details key="baseType" value="Kategorija"/>
    </eAnnotations>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="Knjiga">
    <eAnnotations source="http://org.eclipse/emf/ecore/util/ExtendedMetaData">
      <details key="name" value="Knjiga"/>
      <details key="kind" value="elementOnly"/>
    </eAnnotations>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="naziv" unique="false" lowerBound="1"
      eType="ecore:EDataType http://www.eclipse.org/emf/2003/XMLType#/String">
      <eAnnotations source="http://org.eclipse/emf/ecore/util/ExtendedMetaData">
        <details key="kind" value="element"/>
        <details key="name" value="naziv"/>
      </eAnnotations>
    </eStructuralFeatures>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="stranica" unique="false"
      lowerBound="1" eType="ecore:EDataType http://www.eclipse.org/emf/2003/XMLType#/Int"
      unsettable="true">
      <eAnnotations source="http://org.eclipse/emf/ecore/util/ExtendedMetaData">
        <details key="kind" value="element"/>
        <details key="name" value="stranica"/>
      </eAnnotations>
    </eStructuralFeatures>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="kategorija" unique="false"
      lowerBound="1" eType="#/Kategorija" defaultValueLiteral="Misterij" unsettable="true">
      <eAnnotations source="http://org.eclipse/emf/ecore/util/ExtendedMetaData">
        <details key="kind" value="element"/>
        <details key="name" value="kategorija"/>
      </eAnnotations>
    </eStructuralFeatures>
    <eStructuralFeatures xsi:type="ecore:EReference" name="autor" lowerBound="1" eType="#/Pisac"
      eOpposite="#/Pisac/knjige">
      <eAnnotations source="http://org.eclipse/emf/ecore/util/ExtendedMetaData">
        <details key="kind" value="element"/>
        <details key="name" value="autor"/>
      </eAnnotations>
    </eStructuralFeatures>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="Knjiznica">
    <eAnnotations source="http://org.eclipse/emf/ecore/util/ExtendedMetaData">
      <details key="name" value="Knjiznica"/>
      <details key="kind" value="elementOnly"/>
    </eAnnotations>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="ime" unique="false" lowerBound="1"
      eType="ecore:EDataType http://www.eclipse.org/emf/2003/XMLType#/String">
      <eAnnotations source="http://org.eclipse/emf/ecore/util/ExtendedMetaData">
```

```

    <details key="kind" value="element" />
    <details key="name" value="ime" />
  </eAnnotations>
</eStructuralFeatures>
<eStructuralFeatures xsi:type="ecore:EReference" name="pisci" upperBound="-1"
  eType="#//Pisac" containment="true" resolveProxies="false">
  <eAnnotations source="http://org/eclipse/emf/ecore/util/ExtendedMetaData">
    <details key="kind" value="element" />
    <details key="name" value="pisci" />
  </eAnnotations>
</eStructuralFeatures>
<eStructuralFeatures xsi:type="ecore:EReference" name="knjige" upperBound="-1"
  eType="#//Knjiga" containment="true" resolveProxies="false">
  <eAnnotations source="http://org/eclipse/emf/ecore/util/ExtendedMetaData">
    <details key="kind" value="element" />
    <details key="name" value="knjige" />
  </eAnnotations>
</eStructuralFeatures>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="Pisac">
  <eAnnotations source="http://org/eclipse/emf/ecore/util/ExtendedMetaData">
    <details key="name" value="Pisac" />
    <details key="kind" value="elementOnly" />
  </eAnnotations>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="ime" unique="false" lowerBound="1"
    eType="ecore:EDatatype http://www.eclipse.org/emf/2003/XMLType#//String">
    <eAnnotations source="http://org/eclipse/emf/ecore/util/ExtendedMetaData">
      <details key="kind" value="element" />
      <details key="name" value="ime" />
    </eAnnotations>
  </eStructuralFeatures>
  <eStructuralFeatures xsi:type="ecore:EReference" name="knjige" upperBound="-1"
    eType="#//Knjiga" eOpposite="#//Knjiga/autor">
    <eAnnotations source="http://org/eclipse/emf/ecore/util/ExtendedMetaData">
      <details key="kind" value="element" />
      <details key="name" value="knjige" />
    </eAnnotations>
  </eStructuralFeatures>
</eClassifiers>
</ecore:EPackage>

```

5.2. Knjiznica.genmodel

```

<?xml version="1.0" encoding="UTF-8"?>
<genmodel:GenModel xmi:version="2.0"
  xmlns:xmi="http://www.omg.org/XMI" xmlns:ecore="http://www.eclipse.org/emf/2002/Ecore"
  xmlns:genmodel="http://www.eclipse.org/emf/2002/GenModel" modelDirectory="/Primjer/src"
  modelPluginID="Primjer" modelName="Knjiznica" importerID="org.eclipse.xsd.ecore.importer">
  <foreignModel>file:/home/Knjiznica.xsd</foreignModel>
  <genPackages prefix="Knjiznica" basePackage="org.eclipse.example" resource="XML"
    disposableProviderFactory="true" ecorePackage="knjiznica.ecore#/">
  <genEnums ecoreEnum="knjiznica.ecore#//Kategorija">
    <genEnumLiterals ecoreEnumLiteral="knjiznica.ecore#//Kategorija/Misterij"/>
    <genEnumLiterals ecoreEnumLiteral="knjiznica.ecore#//Kategorija/ZnanstvenaFantastika"/>
    <genEnumLiterals ecoreEnumLiteral="knjiznica.ecore#//Kategorija/Biografija"/>
  </genEnums>
  <genDataTypes ecoreDataType="knjiznica.ecore#//KategorijaObject"/>
  <genClasses ecoreClass="knjiznica.ecore#//Knjiga">
    <genFeatures createChild="false" ecoreFeature="ecore:EAttribute
knjiznica.ecore#//Knjiga/naziv"/>
    <genFeatures createChild="false" ecoreFeature="ecore:EAttribute
knjiznica.ecore#//Knjiga/stranica"/>
    <genFeatures createChild="false" ecoreFeature="ecore:EAttribute
knjiznica.ecore#//Knjiga/kategorija"/>
    <genFeatures notify="false" createChild="false" propertySortChoices="true"
ecoreFeature="ecore:EReference knjiznica.ecore#//Knjiga/autor"/>
  </genClasses>
  <genClasses ecoreClass="knjiznica.ecore#//Knjiznica">
    <genFeatures createChild="false" ecoreFeature="ecore:EAttribute
knjiznica.ecore#//Knjiznica/ime"/>

```

```

    <genFeatures property="None" children="true" createChild="true".ecoreFeature="ecore:EReference
knjiznica.ecore#//Knjiznica/pisci"/>
    <genFeatures property="None" children="true" createChild="true".ecoreFeature="ecore:EReference
knjiznica.ecore#//Knjiznica/knjige"/>
  </genClasses>
  <genClasses.ecoreClass="knjiznica.ecore#//Pisac">
    <genFeatures createChild="false".ecoreFeature="ecore:EAttribute knjiznica.ecore#//Pisac/ime"/>
    <genFeatures notify="false" createChild="false" propertySortChoices="true"
.ecoreFeature="ecore:EReference knjiznica.ecore#//Pisac/knjige"/>
  </genClasses>
</genPackages>
</genmodel:GenModel>

```

5.3. Knjiznica.xsd2ecore

```

<?xml version="1.0" encoding="ASCII"?>
<xsd2ecore:XSD2EcoreMappingRoot xmi:version="2.0"
  xmlns:xmi="http://www.omg.org/XMI" xmlns:xsd2ecore="http://www.eclipse.org/emf/2002/XSD2Ecore"
  inputs="file:/home/Knjiznica.xsd#"
  outputs="knjiznica.ecore#" topToBottom="true">
  <nested inputs="file:/home/Knjiznica.xsd#//Kategorija;XSDSimpleTypeDefinition"
  outputs="knjiznica.ecore#//Kategorija">
    <nested
  inputs="file:/home/Knjiznica.xsd#//Kategorija;XSDSimpleTypeDefinition/XSDEnumerationFacet"
  outputs="knjiznica.ecore#//Kategorija/Misterij"/>
    <nested
  inputs="file:/home/Knjiznica.xsd#//Kategorija;XSDSimpleTypeDefinition/XSDEnumerationFacet=1"
  outputs="knjiznica.ecore#//Kategorija/ZnanstvenaFantastika"/>
    <nested
  inputs="file:/home/Knjiznica.xsd#//Kategorija;XSDSimpleTypeDefinition/XSDEnumerationFacet=2"
  outputs="knjiznica.ecore#//Kategorija/Biografija"/>
  </nested>
  <nested
  inputs="bundleentry://422/cache/www.w3.org/2001/XMLSchema.xsd#//string;XSDSimpleTypeDefinition=7"
  outputs="http://www.eclipse.org/emf/2003/XMLType#//String"/>
  <nested
  inputs="bundleentry://422/cache/www.w3.org/2001/XMLSchema.xsd#//int;XSDSimpleTypeDefinition=42"
  outputs="http://www.eclipse.org/emf/2003/XMLType#//Int"/>
  <nested
  inputs="bundleentry://422/cache/www.w3.org/2001/XMLSchema.xsd#//anyURI;XSDSimpleTypeDefinition=23"
  outputs="http://www.eclipse.org/emf/2003/XMLType#//AnyURI"/>
  <nested inputs="file:/home/Knjiznica.xsd#//Pisac;XSDComplexTypeDefinition=1"
  outputs="knjiznica.ecore#//Pisac">
    <nested
  inputs="file:/home/Knjiznica.xsd#//Pisac;XSDComplexTypeDefinition=1/XSDParticle/XSDModelGroup/XSDPar
  ticle"
  outputs="knjiznica.ecore#//Pisac/ime"/>
    <nested
  inputs="file:/home/Knjiznica.xsd#//Pisac;XSDComplexTypeDefinition=1/XSDParticle/XSDModelGroup/XSDPar
  ticle=1"
  outputs="knjiznica.ecore#//Pisac/knjige"/>
  </nested>
  <nested inputs="file:/home/Knjiznica.xsd#//Knjiga;XSDComplexTypeDefinition"
  outputs="knjiznica.ecore#//Knjiga">
    <nested
  inputs="file:/home/Knjiznica.xsd#//Knjiga;XSDComplexTypeDefinition/XSDParticle/XSDModelGroup/XSDPart
  icle"
  outputs="knjiznica.ecore#//Knjiga/naziv"/>
    <nested
  inputs="file:/home/Knjiznica.xsd#//Knjiga;XSDComplexTypeDefinition/XSDParticle/XSDModelGroup/XSDPart
  icle=1"
  outputs="knjiznica.ecore#//Knjiga/stranica"/>
    <nested
  inputs="file:/home/Knjiznica.xsd#//Knjiga;XSDComplexTypeDefinition/XSDParticle/XSDModelGroup/XSDPart
  icle=2"
  outputs="knjiznica.ecore#//Knjiga/kategorija"/>
    <nested
  inputs="file:/home/Knjiznica.xsd#//Knjiga;XSDComplexTypeDefinition/XSDParticle/XSDModelGroup/XSDPart
  icle=3"
  outputs="knjiznica.ecore#//Knjiga/autor"/>

```

```

</nested>
<nested inputs="file:/home/Knjiznica.xsd#/Knjiznica;XSDComplexTypeDefinition=2"
  outputs="knjiznica.ecore#/Knjiznica">
  <nested
inputs="file:/home/Knjiznica.xsd#/Knjiznica;XSDComplexTypeDefinition=2/XSDParticle/XSDModelGroup/XS
DParticle"
  outputs="knjiznica.ecore#/Knjiznica/ime"/>
  <nested
inputs="file:/home/Knjiznica.xsd#/Knjiznica;XSDComplexTypeDefinition=2/XSDParticle/XSDModelGroup/XS
DParticle=1"
  outputs="knjiznica.ecore#/Knjiznica/pisci"/>
  <nested
inputs="file:/home/Knjiznica.xsd#/Knjiznica;XSDComplexTypeDefinition=2/XSDParticle/XSDModelGroup/XS
DParticle=2"
  outputs="knjiznica.ecore#/Knjiznica/knjige"/>
</nested>
</xsd2ecore:XSD2EcoreMappingRoot>

```

5.4. Knjiznica.gmftool

```

<?xml version="1.0" encoding="UTF-8"?>
<gmftool:ToolRegistry xmi:version="2.0"
  xmlns:xmi="http://www.omg.org/XMI"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:gmftool="http://www.eclipse.org/gmf/2005/ToolDefinition">
<palette
  title="Izbornik"
  description="Odabir entiteta knjiznice">
<tools
  xsi:type="gmftool:ToolGroup"
  title="Elementi"
  description="Elementi koje mozemo dodavati u model">
<tools
  xsi:type="gmftool:CreationTool"
  title="Knjiga"
  description="Knjiga"/>
<tools
  xsi:type="gmftool:CreationTool"
  title="Pisac"
  description="Pisac"/>
<tools
  xsi:type="gmftool:CreationTool"
  title="autor knjige"
  description="Veza izmedju knjige i autora"/>
</tools>
</palette>
</gmftool:ToolRegistry>

```

5.5. Knjiznica.gmfgraph

```

<?xml version="1.0" encoding="UTF-8"?>
<gmfgraph:Canvas xmi:version="2.0"
  xmlns:xmi="http://www.omg.org/XMI"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:gmfgraph="http://www.eclipse.org/gmf/2005/GraphicalDefinition" name="knjiznica">
<figures
  name="Knjiznica">
<figures
  xsi:type="gmfgraph:Rectangle"
  referencingElements="Knjiga"
  name="KnjigaFigure">
<children
  xsi:type="gmfgraph:Label"
  referencingElements="KnjigaNaziv"
  name="KnjigaNazivFigure"
  text="&lt;...>/>
<children
  xsi:type="gmfgraph:Label"

```

```
referencingElements="KnjigaStranica"
name="KnjigaStranicaFigure"
text="&lt;...&gt;/>
<children
  xsi:type="gmfgraph:Label"
  referencingElements="KnjigaKategorija"
  name="KnjigaKategorijaFigure"
  text="&lt;...&gt;/>
<backgroundColor
  xsi:type="gmfgraph:ConstantColor"
  value="lightBlue"/>
</figures>
<figures
  xsi:type="gmfgraph:Rectangle"
  referencingElements="Pisac"
  name="PisacFigure">
  <children
    xsi:type="gmfgraph:Label"
    referencingElements="PisacIme"
    name="PisacImeFigure"
    text="&lt;...&gt;"
    <foregroundColor
      xsi:type="gmfgraph:ConstantColor"
      value="black"/>
  </children>
  <backgroundColor
    xsi:type="gmfgraph:ConstantColor"
    value="orange"/>
</figures>
<figures
  xsi:type="gmfgraph:PolylineConnection"
  referencingElements="PisacKnjige"
  name="PisacKnjigeFigure"
  targetDecoration="PisacKnjigeTargetDecoration"/>
<figures
  xsi:type="gmfgraph:PolylineDecoration"
  name="PisacKnjigeTargetDecoration"/>
</figures>
<nodes
  name="Knjiga"
  figure="KnjigaFigure"/>
<nodes
  name="Pisac"
  figure="PisacFigure"/>
<connections
  name="PisacKnjige"
  figure="PisacKnjigeFigure"/>
<labels
  name="KnjigaNaziv"
  figure="KnjigaNazivFigure"/>
<labels
  name="KnjigaStranica"
  figure="KnjigaStranicaFigure"/>
<labels
  name="KnjigaKategorija"
  figure="KnjigaKategorijaFigure"/>
<labels
  name="PisacIme"
  figure="PisacImeFigure"/>
</gmfgraph:Canvas>
```

5.6. Knjiznica.gmfmap

```
<?xml version="1.0" encoding="UTF-8"?>
<gmfmap:Mapping xmi:version="2.0"
  xmlns:xmi="http://www.omg.org/XMI"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ecore="http://www.eclipse.org/emf/2002/Ecore"
  xmlns:gmfmap="http://www.eclipse.org/gmf/2005/mappings"
  xmlns:gmftool="http://www.eclipse.org/gmf/2005/ToolDefinition">
```



```

<nodes>
  <containmentFeature
    href="knjiznica.ecore#//Knjiznica/pisci"/>
  <ownedChild>
    <domainMetaElement
      href="knjiznica.ecore#//Pisac"/>
    <labelMappings
      viewPattern="Ime: {0}">
      <diagramLabel
        href="Knjiznica.gmfgraph#PisacIme"/>
      <features
        href="knjiznica.ecore#//Pisac/ime"/>
    </labelMappings>
    <tool
      xsi:type="gmftool:CreationTool"
      href="Knjiznica.gmftool#//@palette/@tools.0/@tools.1"/>
    <diagramNode
      href="Knjiznica.gmfgraph#Pisac"/>
  </ownedChild>
</nodes>
<nodes>
  <containmentFeature
    href="knjiznica.ecore#//Knjiznica/knjige"/>
  <ownedChild>
    <domainMetaElement
      href="knjiznica.ecore#//Knjiga"/>
    <labelMappings
      viewPattern="Naziv: {0}">
      <diagramLabel
        href="Knjiznica.gmfgraph#KnjigaNaziv"/>
      <features
        href="knjiznica.ecore#//Knjiga/naziv"/>
    </labelMappings>
    <labelMappings
      viewPattern="Kategorija: {0}"
      editPattern="">
      <diagramLabel
        href="Knjiznica.gmfgraph#KnjigaKategorija"/>
      <features
        href="knjiznica.ecore#//Knjiga/kategorija"/>
    </labelMappings>
    <labelMappings
      viewPattern="br. stranica: {0}">
      <diagramLabel
        href="Knjiznica.gmfgraph#KnjigaStranica"/>
      <features
        href="knjiznica.ecore#//Knjiga/stranica"/>
    </labelMappings>
    <tool
      xsi:type="gmftool:CreationTool"
      href="Knjiznica.gmftool#//@palette/@tools.0/@tools.0"/>
    <diagramNode
      href="Knjiznica.gmfgraph#Knjiga"/>
  </ownedChild>
</nodes>
<links>
  <tool
    xsi:type="gmftool:CreationTool"
    href="Knjiznica.gmftool#//@palette/@tools.0/@tools.2"/>
  <diagramLink
    href="Knjiznica.gmfgraph#PisacKnjige"/>
  <linkMetaFeature
    xsi:type=".ecore:EReference"
    href="knjiznica.ecore#//Pisac/knjige"/>
</links>
<diagram>
  <diagramCanvas
    href="Knjiznica.gmfgraph#knjiznica"/>
  <domainModel
    href="knjiznica.ecore#/">
  <domainMetaElement
    href="knjiznica.ecore#//Knjiznica"/>

```

```
<palette
  href="Knjiznica.gmftool#@palette"/>
</diagram>
</gmfmap:Mapping>
```


7. Zaključak

Graphical Modeling Framework je vrlo moćno okruženje za brzo razvijanje aplikacija u Eclipse okruženju. Osim toga, može pomoći i u organizacijsko-poslovnim procesima, za planiranje, modeliranje i prikaz sustava dijagramima. Općenito, nudi grafički editor strukturiranih podatkovnih modela opisanih npr. XML shemom, UML modelom, Rose modelom, ili Java sučeljima. Na jednostavan način možemo dobiti editor našeg modela te se stoga možemo više koncentrirati na njegov dizajn. Pogotovo je utilitaran pri razvoju Java aplikacija zbog njegove vlastite Java prirode, stoga su mu tu mogućnosti najveće. Generiranjem konceptualnog modela *Graphical Modeling Frameworkom* odmah se može dobiti kostur cijelog sustava s kompletnom hijerarhijom potrebnih razreda te obratno, svaka promjena u kodu očituje se na dijagramu što vrlo pojednostavljuje neprestano ažuriranje oba te omogućuje bolju preglednost.

Trenutno, *Graphical Modeling Framework* je relativno nov projekt i u fazi razvoja, mnoge mogućnosti nisu još implementirane a mogu se očekivati i problemi u radu zbog grešaka.

Nakon nekog vremena može se očekivati da će *Graphical Modeling Framework* prerasti u vrlo rado korištenu alatku korisnika Eclipse razvojnog okruženja, privući nove, u svakom slučaju GMF je mnogo obogatio Eclipse razvojno okruženje.

8. Literatura

1. Aniszczyk, C., IBM, Learn Eclipse GMF in 15 minutes, 2007, URL: <http://www-128.ibm.com/developerworks/opensource/library/os-ecl-gmf> (25/04/07).
2. Plante, F., IBM, Introducing the GMF Runtime, 2006, URL: <http://www.eclipse.org/articles/Article-Introducing-GMF/article.html> (26/04/07).
3. GMF Labels, URL: http://wiki.eclipse.org/index.php/GMF_Labels (27/04/07).
4. GMF Programmer's Guide, URL: <http://help.eclipse.org/help32/index.jsp?topic=/org.eclipse.gmf.doc/prog-guide/runtime/index.html> (26/04/07).
5. Graphical Modeling Framework FAQ, URL: http://wiki.eclipse.org/index.php/Graphical_Modeling_Framework_FAQ (26/04/07).
6. GMF Runtime How-To Guide, 2005, URL: <http://help.eclipse.org/help32/index.jsp?topic=/org.eclipse.gmf.doc/howto/diagram/index.html> (26/04/07).
7. GMF Newsgroup Q&A, URL: http://wiki.eclipse.org/index.php/GMF_Newsgroup_Q%26A (28/04/07).
8. Generating an EMF Model using XML Schema (XSD), 2006, URL: <http://dev.eclipse.org/viewcvs/indextools.cgi/org.eclipse.emf/doc/org.eclipse.emf.doc/tutorials/xlibmod/xlibmod.html> (20/04/07).
9. Generating an EMF Model, 2006, URL: <http://dev.eclipse.org/viewcvs/indextools.cgi/org.eclipse.emf/doc/org.eclipse.emf.doc/tutorials/clibmod/clibmod.html> (20/04/07).
10. EMF-FAQ, URL: <http://wiki.eclipse.org/index.php/EMF-FAQ> (20/04/07).
11. Graphical Modeling Framework (proposal), URL: <http://www.eclipse.org/proposals/eclipse-gmf> (19/04/07).