Projektovanje softvera

Arhitektura metamodeliranja

Uvod

- UML je jedan od nivoa 4-nivoske arhitekture metamodeliranja
- 4-nivoska arhitektura je dokazana infrastruktura za definisanje precizne semantike koju zahtevaju kompleksni modeli
- Prednosti pristupa 4-nivoske arhitekture:
  - rafinira semantičke konstrukcije njihovom rekurzivnom primenom na sukcesivne metanivoe
  - obezbeđuje arhitektonsku osnovu za:
    - definisanje budućih proširenja UML metamodela
    - nivelaciju UML metamodela sa drugim standardima zasnovanim na 4-nivoskoj arhitekturi metamodeliranja, posebno OMG *Meta-Object Facility* (MOF)

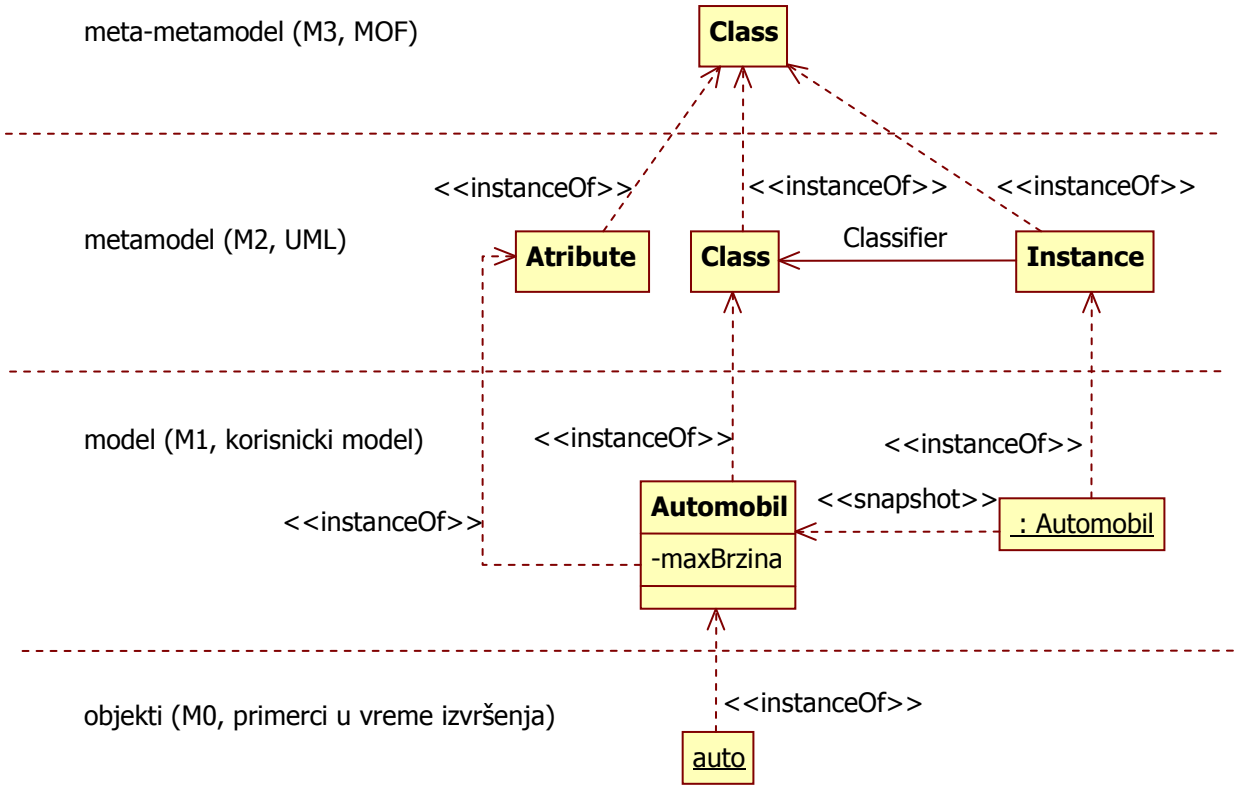
Četvoronivoska arhitektura

- Odnos između 2 susedna nivoa
  - niži nivo je instanca višeg nivoa
- Broj meta-nivoa je teorijski neograničen,
  ali su za praktične primene dovoljna 4 nivoa
- Opšte prihvaćeni radni okvir za metamodeliranje
  je zasnovan na arhitekturi sa sledeća četiri nivoa (sloja):
  - meta-metamodel
  - metamodel
  - model
  - korisnički objekti

Opis nivoa arhitekture

Nivo	Opis	Primer
meta-metamodel	Infrastruktura za arhitekturu metamodeliranja. Definiše jezik za specificiranje metamodela.	*MetaClass, MetaAttribute, MetaOperation*
metamodel	Jedna instanca meta-metamodela. Definiše jezik za specificiranje modela.	*Class, Attribute, Operation, Component*
model	Jedna instanca metamodela. Definiše jezik za opis nekog informacionog domena.	*Dokument, velicina, otvaranje()*
korisnički objekti	Jedna instanca modela. Definiše neki specifičan informacioni domen.	*Dokument dokument = new Dokument("Text.rtf");* *int i=100;*

Primer


meta-metamodel (M3, MOF)
Class
<<instanceOf>>
<<instanceOf>>
<<instanceOf>>
metamodel (M2, UML)
Atribute
Class
Classifier
Instance
model (M1, korisnicki model)
<<instanceOf>>
<<instanceOf>>
<<instanceOf>>
Automobil
-maxBrzina
<<snapshot>>
: Automobil
objekti (M0, primerci u vreme izvršenja)
<<instanceOf>>
auto

Meta-metamodel

- Nivo meta-metamodeliranja daje temelj za arhitekturu metamodeliranja
- Primarna odgovornost nivoa meta-metamodela:
  - da definiše jezik za specificiranje metamodela
- Meta-metamodel definše model na višem stepenu apstrakcije od metamodela i tipično je kompaktniji od metamodela koji opisuje
- Jedan meta-metamodel može definisati više metamodela, i može biti više meta-metamodela pridruženih svakom metamodelu
- Generalno je poželjno da povezani metamodeli i meta-metamodeli dele zajedničke projektne filozofije i konstrukte
  - ovo nije striktno pravilo
- Svaki nivo treba da održava svoj sopstveni projektni integritet
- Primeri meta-metaobjekata u sloju meta-metamodeliranja su:
  - `MetaClass`, `MetaAttribute` i `MetaOperation`

Metamodel

- Metamodel je jedan primerak meta-metamodela
- Primarna odgovornost nivoa metamodela:
  - da definiše jezik za specificiranje modela
- Metamodeli su tipično detaljniji od meta-metamodela koji ih opisuju, specijalno kada definišu dinamičku semantiku
- Primeri metaobjekata u sloju metamodeliranja su:
  - `Class`, `Attribute`, `Operation` i `Component`

Model

- Model je jedan primerak metamodela
- Primarna odgovornost nivoa modela:
  - da definiše jezik koji opisuje neki informacioni domen
- Primeri objekata u sloju modeliranja su:
  - `Dokument`, `velicina`, `otvaranje()`

Korisnički objekti

- Korisnički objekti su jedan primerak modela
- Primarna odgovornost nivoa korisničkih objekata
  - je da opišu specifičan informacioni domen
- Primeri objekata u sloju korisničkih objekata su:
  `Dokument dokument = new Dokument(“Text.rtf”)i int i=100`

Analogija

- Nije samo arhitektura OO meta-modeliranja 4-nivoska
  - i arhitektura proceduralnog projektovanja i realizacije je 4-nivoska:
- Metajezik (meta-metamodel):
  - stanje (podaci): primitivni tip, niz, struktura
  - ponašanje (kontrola toka): sekvenca, selekcija, iteracija
- Jezik (metamodel):
  - podaci: `int`, `[]`, `struct`
  - kontrola toka: `{nar1; nar2;…}`, `if-else`, `for`, `while`
- Program (model):
  - podaci: `typedef unsigned int Uint; struct S{int c; float n[2];}`
  - kontrola toka: `if (a) {b=c; while(d<100){…}} else {…}`
- Izvršenje programa (korisnički objekti):
  - podaci: `Uint i=5;  S s=new S();`
  - kontrola toka: 3. izvršenje ciklusa `while`