1 Objektno-orijentisana analiza (OOA)

Različite metode analize informacionog sistema (na primer, funkcionalna dekompozicija, dijagrami toka podataka kod strukturne sistem analize, model entiteta i odnosa) mogu se opisati načinom na koji koriste i organizuju podatke i aktivnosti. Jedan način za taj opis su jednačine sledeég tipa.

Funkcionalna dekompozicija =
    Funkcije +
    Pod-funkcije +
    Funkcijski interfejsi

Dijagram toka podataka =
    Tokovi podataka (& kontrole) +
    Transformacija podataka (& kontrole) +
    Skladišta podataka (& kontrole) +
    Kriterijumi kraja +
    Specifikacija procesa +
    Rečnik podataka

Model entiteta i odnosa =
    Objekti (entiteti) +
    Atributi +
    Odnosi +
    Nadtip/Podtipovi +
    Asocijativni objekti (entiteti)

Objektna analiza ne odbacuje ove, ranije metode, već koristi njihova dobra svojstva, odvacuje loša, i kombinuje ih.

Za svaku od ovih, ranijih metoda, postoje "loša" svojstva koja je potrebno prevazići novom metodom, metodom OOA.

Na primer, funkcionalna dekompozicija zahteva da čovek preslika domen problema (npr. kontrolu saobraćaja) u funkcije i podfunkcije. Ne postoji mehanizam koji eksplicitno preslikava izgradjenu funkcionalnost (sistem funkcija i podfunkcija) nazad u domen problema, tj. ne postoji način provere kvaliteta dobijene dekompozicije. Elemente ove metode koristi i OOA, ali u vrlo ograničenom kontekstu koji omogućuje takvu proveru. Taj kontekst je definisanje metoda OO modela, koji definišu specifično ponašanje koje objekat treba da ispolji. Sama funkcionalna dekompozicija je tada korisna kod razlaganja kompleksnog servisa na jednostavnije celine – funkcije kojima se precizira šta se zahteva. Sama metoda funkcionalne dekompozicije, tj. koraci obrade, u OOA se, medjutim, ne koristi kao osnovni okvir organizacije za vreme analize, jer je suviše podložna promenama.

Problem sa SSA je da ne daje kriterijum gde se zaustaviti, tako da se sa procesom analize može ići, u sveobuhvatnosti, dosta daleko, potrošiti puno vremena i sredstava. Veliki problem predstavlja i veličina rečnika podataka (do 1000 stranica). Mada CASE-alti mogu da pomognu u održavanju sintaksne ispravnosti rečnika podataka, njegova semantika postaje nesavladiva za čoveka.

Još jedan problem sa SSA je slaba podrška podacima kroz skladišta podataka, zbog čega se sama metoda često vezuje za model entiteta i odnosa. Najzad, problem sa SSA je u bitnoj razlici u notaciji od notacije strukturnog projektovanja. Dok dijagram toka podataka predstavlja mrežnu reprezantaciju procesa i skladišta, strukturni dijagram (dobijen u fazi projektovanja) je striktna hijerarhija modula.

Cilj OOA je da se razvije i koristi ista notacija kao i u fazi OOP.

Model entiteta i odnosa preslikava neposredno domen problema u objekte modela, ali je sama metoda zasnovana na njemu parcijalna jer joj nedostaju koncepti OO metoda, učurenih sa atributima tako da predstavljaju celinu, poruka, kao usko definisane sumedje prema objektu, nasledjivanja koje ukazuje na zajedničko medju objektima, i strukture, kao što je struktura celina/deo, koja predstavlja osnovnu metodu čovekove organizacije.

Jednačina kojom se definiše objektno-orijentisani pristup je sledeća:

Objektno-orijentisan = Klase i Objekti + Nasledjivanje + Komunikacija porukama.

U ukupnom pristupu, OOA se sastoji od pet osnovnih aktivnosti:

Nalaženje Klasa&Objekata

Identifikovanje strukture (generalizacija/specijalizacija, celina/deo)

Identifikovanje tematskih celina

Definisanje atributa

Definisanje metoda

Ovo su aktivnosti u smislu da se ne moraju izvršavati ovim redom (mada to obično pomaže). Tako se OOA model može predstaviti sa pet nivoa:

1. Nivo tematskih celina
2. Nivo Klasa&Objekata
3. Nivo strukture
4. Nivo atributa
5. Nivo metoda

Motivacija i koristi od OOA su višestruke.

1. OOA daje naglasak na razumevanje domena problema
2. OOA poboljšava interakciju analitičara i eksperta u domenu problema
3. OOA povećava unutrašnju konzistentnost rezultata analize
4. OOA koristi nasledjivanje da identifikuje zajedničke atribute i metode
5. OOA gradi specifikaciju koja je otporna na promene
6. OOA omogućuje ponovno korišćenje rezultata analize
7. OOA obezbedjuje konzistentnu reprezentaciju sa OOP reprezentacijom i implementacijom.

OOA definiše objekat i klasu na način koji reflektuje i domen problema i ono što je, iz domena problema, od interesa za informacioni (ili programski uopšte) sistem.

Objekat je apstrakcija nečega u domenu problema, koja reflektuje mogućnost sistema da pamti informaciju o tome (nečemu) i da interaguje sa nime; objekat je učaurenje vrednosti atributa i njihovih ekskluzivnih metoda. Sinonim za objekat je instanca.

Klasa je opis jednog ili više objekata sa uniformnim skupom atributa i metoda, uključujući opis kako se novi objekat klase kreira.

Klasa&objekti je termin koji označava klasu i objekte te klase.

Klasa bez objekata se naziva apstraktnom klasom, i ona se ponekad uvodi kao koren hijerarhije generalizacija/specijalizacija (kao generička klasa). U tom slučaju samo specijalizovane klase imaju primerke.

Klase&objekti, koji opisuju domen problema, su relativno stabilni u vremenu i omogućuju ponovno korišćenje (engl. reusability) rezultata analize.

U opadajućem redosledu stabilnosti su sledeći elementi:

podaci,

redosled funkcija

funkcije

sumedje (interfejsi) izmedju komponenti sistema.

1.1 Klase i objekti

Oznaka za klasu&objekte je na slici 1.

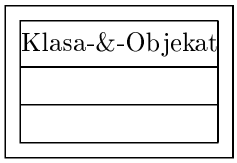

Figure 1: Simbol za klasu&objekte

Spoljašnji pravougaonik označava objekat, dok unutrašnji pravougaonik označava klasu. Pravougaonik je podeljen u tri dela – deo za ime klase, deo za atribute i deo za metode. Ime klase je imenica u jednini ili atribut i imenica, i treba da imenuje pojedinačni objekat klase.

Oznaka za klasu je na slici 2.

Figure 2: Simbol za klasu

Broj klasa u OOA modelu zavisi od širine i dubine domena problema. U proseku je oko 35 klasa, ali ih može biti i mnogo više, razdeljenih u poddomene problema, za kompleksne domene (npr. upravljanje saobraćajem).

Da bi se došlo do prve iteracije u identifikovanju klasa, potrebno je posmatrati, aktivno slušati, proveriti i konsultovati rezultate prethodne OOA, konsultovati druge – srodne sisteme, čitati i graditi prototipove. Sa čitanjem, na primer, treba poći od dokumenta koji opisuje zahtev za sistemom i koji proizvodi klijent (naručilac sistema), bilo po svom nahodjenju bilo na zahtev analitičara. Poželjno je čitati i druge izvore, kao što je detaljni opis domena problema koji se može naći u nekoj od enciklopedija, ili u nekom od klasifikacionih sistema. Cilj je upoznati se što detaljnije sa domenom problema.

Potencijalne klase&objekte treba tražiti u strukturama (generalizacija/specijalizacija, celina/deo), drugim sistemima, uredjajima, stvarima ili dogadjajima koji se pamte, ulogama, procedurama i organizacionim jedinicama. Ne traba uključivati računarske komponente koje su specifične za implementaciju (npr. terminale, disk jedinice), jer su to razmatranja za kasnije faze.

Potencijalne klase&objekti treba da imaju sledeća svojstva (da bi bile usvojene kao klase&objekti u OOA):

1. postoji potreba za njihovim pamćenjem i specifičnim ponašanjem u sistemu
2. obično imaju više od jednog atributa
3. obično ima više od jednog objekta u klasi
4. atributi se uvek mogu primeniti (za sve primerke klase)
5. metode se uvek mogu primeniti (za sve primerke klase)
6. klase odražavaju zahteve domena problema, tj. zahteve koje sistem mora da podrži bez obzira na računarsku tehnologiju koja će se koristiti u izgradnji; klase u OOA nisu, npr. prozori, meniji, ekrani
7. izvedeni rezultati nisu klase u OOA; oni se mogu na kraju, u OOP ili implementaciji, proizvesti u klase radi efikasnosti.

Primer – registracija motornih vozila.

Opis problema: Sistem registracije motornih vozila održava sledeće informacije:

Organizacija – naziv, direktor, adresa, telefon (ovo je organizacija koja je ovlašćena za registraciju vozila i izdavanje tablica)

Operater – ime, adresa, userid, autorizacija, datum zaposlenja

Vlasnik – ime, adresa, telefon

Tablica – broj, evidencija vlasništva, datum izdavanja, taksa

Registracija – datum izdavanja, datum početka važenja, datum prestanka važenja, nalepnica(godina, tip, broj), taksa

Vozilo – broj, godina, model, karoserija, boja, cena, i

za kamione: broj putnika, brzina, nosivost, težina

za motorcikle: broj putnika, brzina

za prikolice: težina

za putničke prikolice: broj putnika, brzina, nosivost, broj karoserije, dužina

Operator vrši registraciju, izdavanje tablica i naplatu takse.

Svaki operator pripada nekoj organizaciji.

Sistem izdaje opomene o obnavljanju registracije.

Moguće klase&objekti su: Vozilo (drugi sistem), dogadjdji koji se pamte (PravniDogadjaj koji se specijalizuje u IzdavanjeTablice i Registracija, uloge (Osoba koja se specijalizuje u VlasnikOsoba, OperatorOsoba, OperatorVlasnikOsoba), i Organizacija (ili njene specijalizacije, ako se organizacija na nivou opštine opisuje različitim metodama i atributima od organizacije na nivou grada, republike, itd).

Nivo klasa i objekata za ovaj sistem predstavljen je na slici 3.

Dodatni detalji se razradjuju u sledećim aktivnostima OOA.

1.2 Identifikovanje struktura

Termin struktura upotrebljava se da opiše strukturu generalizacije/specijalizacije i strukturu celina/deo.

Generalizacija/specijalizacija je struktura nad klasama. Oznaka za strukturu generalizacija/specijalizacija je na slici 4.

Za sve specijalizacije na njnižem nivou koristi se simbol za klasu&objekat, dok se na drugim mestima u ovoj strukturi može koristiti bilo sombol za klasu&objekat, bilo simbo za klasu, koji je veádekvatan.

Strategija identifikovanja strukture generalizacija/specijalizacija počinje posmatranjem svake klase kao potencijalne generalizacije, i razmatranjem da li njene potencijalne specijalizacije pripadaju domenu problema, sistemu koji se gradi, da li ima nasledjivanja i da li potencijalne specijalizacije imaju ranije navedena svojstva klasa. Na sličan način se svaka postojeća klasa posmatra kao potencijalna specijalizacija, i za njenu potencijalnu generalizaciju se razmatraju ista pitanja.

Struktura generalizacija/specijalizacija nije prosto izdvajanje zajedničkih atributa, već ona mora da odrazi postojanje ovog odnosa u domenu problema. Na primer, izdvajanje zajedničkog atributa LOKACIJA kao generalizacije za klase Pošiljka, Avion, Aerodrom, nema opravdanje u domenu problema (slika 5).

Najčešći oblik strukture generalizacija/specijalizacija je hijerarhija, kao na slici 6.

Medjutim. ova struktura može imati i strukturu rešetke (slika 7).

Druga važna struktura je struktura celina/deo nad objektima. Objekat celina se prikazuje na vrhu a zatim objekat deo (slika 8).

Ova struktura odgovara kompozitnom objektu iz proširenja jezgra OO modela.

Krajnje tačke linija u ovoj strukturi ukazuju da se radi o strukturi nad objektima a ne o strukturi nad klasama.

Svaki kraj linije u ovoj strukturi označen je brojem ili intervalom koji označavaju broj delova koji celina može da sadrži, i u koliko celina jedan deo može biti uključen, u bilo kom trenutku.

Istraživanje strukture celina/deo može da uvede nove klase&objekte.

Kandidate za strukturu celina/deo treba tražiti medju odnosima tipa, npr. sastav/komponenta, sadržalac/sadržaj, kolekcija/član.

Slika 9 prikazuje nivo struktura za primer sistema registracije motornih vozila.

1.3 Identifikovanje tematskih celina

Tematske celine daju pregled većeg OOA modela. One su mehanizam koji vodi čitaoca (analitičara, eksperta u domenu problema, direktora, klijenta) kroz veliki, kompleksni model. Tematske celine pomažu u organizovanju posla na većim projektima, koje se zasniva na početnim istraživanjima OOA. Princip na kome se zasnivaju tematske celine je proširenje odnosa celina/deo koje se zove skaliranje.

Izbor tematskih celina vrši se tako što se na početku vršna klasa svake strukture promoviše u tematsku celinu, kao i svaka kalsa&objekti koja nije u strukturi. Za primer sistema registracije vozila, mogu se klase Organizacija, PravniDogadjaj, Osoba, Vozilo promovisati u tematske celine. Zatim se inicijalni izbor tematskih celina može profinjavati na osnovu zahteva za minimizovanjem medjuzavisnosti i interakcije medju tematskim celinama.

Jedna klasa&objekti može biti u više tematskih celina, ako je to potrebno. Notacija uključuje razne načine predstavljanja tematskih celina, od sažetog (slika 10), preko delimično raširenog (slika 11) do potpuno raširenog (slika 12).

Na slikama **??** i **??** prikazani su OOA model sistema registracije vozila sa nivoima tematskih celina i klasa&objekti, odnosno sa nivoima tematskih celina, klasa&objekti i struktura.

1.4 Definisanje atributa

Atribut u OOA je neki podatak (informacija o stanju) za koji svaki objekat u jednoj klasi ima svoju sopstvenu vrednost. Atributi opisuju vrednosti koje se čuvaju u objektu i kojima jedino mogu manipulisati metode tog objekta. Ako neki drugi deo sistema ima potrebu da pristupi vrednostima objekta, on to može učiniti jedino slanjem poruke (u OOA vezom poruke, engl. Message Connection) koja odgovara metodi definisanoj za taj objekat. Atributi se smeštaju u središnji odeljak simbola za klasu&objekte (slika 13).

(primer neadekvatnog atributa – kreditna kartica)

Atributi se biraju tako da budu atomični u smislu da predstavljaju jednu vrednost ili grupisanje tesno povezanih vrednosti. Eventualna normalizacija, mehanizam identifikovanja ili izdvajanja, kao posebnog atributa, vrednosti koja se može izračunati iz drugih vrednosti, ostavljaju se za fazu projektovanja.

Atribut se dodeljuje najvišoj klasi u strukturi generalizacija/specijalizacija, u kojoj ostaje primenljiv za svaku njenu specijalizaciju.

Kada su definisani atributi, može se uspostaviti još jedna veza medju objektima kao instancama – primercima klasa. U OOA ta veza se zove instancna veza, i odgovara (u OO modelu) horizontalnoj hijerarhiji klasa (hijerarhiji kompozicije klasa) sa složenim klasama, isključujući odnos celina/deo (kompozitne objekte). Instancna veza se predstavlja notacijom na slici 14.

1.5 Definisanje metoda

U OOA metoda se definiše kao specifično ponašanje koje je objekat dužan da ispolji.

Strategija definisanja metoda uključuje sledeće aktivnosti:
1. Identifikovanje stanja objekta
2. Identifikovanje potrebnih metoda
3. identifikovanje poruka
4. Specifikacija metoda
5. Objedinjavanje OOA dokumentacije

Postupak definisanja metoda ne mora da sledi za postupkom definisanja atributa. Neki analitičari primenjuju obrnut redosled, dok se često primenjuje i paralelno definisanje jednih i drugih (malo atributa, malo metoda, pa opet atributa itd).

Simbol za metode u OOA je prikazan na slici 15.

Medju metodama izdvajaju se algoritamski jednostavne metode i algoritamski složene metode.

Algoritamski jednostavne metode odnose se na svaku klasu&objekti u modelu. To su kreiranje objekta, povezivanje objekta (sa drugim objektom), pristup objektu (čitanje ili upis vrednosti atributa objekta), i uklanjanje objekta. Ove metode su implicitne i ne navode se u oznaci klase.

U sistemu registracije motornih vozila, na primer, preko 95% ponašanja sistema ispoljava se kroz algoritamski jednostavne metode.

Algoritamski složene metode dele se na dve kategorije: računanje i nadgledanje.

Računanjem se izračunava rezultat iz vrednosti atributa objekta.

Metode nadgledanja kontrolišu spoljašnji sistem ili uredjaj. One se bave ulazom i izlazom spoljašnjih sistema, prikupljanjem podataka uredjaja i njihovim upravljanjem.

Kada su definisane metode, moguće je identifikovati poruke. Poruka je preslikavanje jednog objekta u drugi objekat (ili u klasu, da se kreira novi objekat), u kome "pošiljalac" šalje poruku "primaocu" da bi obavio neku obradu. Tražena obrada se navodi (imenuje) u specifikaciji metode pošiljaoca, a definiše se u specifikaciji metoda primaoca. Ovakva komunikacija je vrlo ograničena i predstavlja usku sumedju izmedju učaurenja atributa i ekskluzivnih metoda nad atributima (npr. poruka tipa "Zdravo, kako si" se **ne** šalje medju objektima).

Oznaka za poruke je prikazana na slici 16.

Za poruke jednog objekta većem broju drugih objekata koristi se notacija na slici ??.

Da bi se identifikovale potrebne poruke treba razmotriti, za svaki objekat:
– Od kojih još objekata mu je potrebna metoda (usluga)? (Nacrtati strelicu do svakog takvog objekta.)
– Kojim je još objektima potrebna neka od njegovih metoda (usluga)? (Nacrtati strelicu od svakog takvog objekta ka objektu koji se posmatra.)
– Pratiti poruku do sledećeg objekta, i ponoviti pitanja.

Metode treba predstaviti u okviru obrasca klase&objekata, pomoću dijagrama metode za svaku metodu. Obrazac klase&objekata ima sledeću formu:

specifikacija
atribut
atribut
atribut
spoljašnji ulaz
spoljašnji izlaz
...
metoda(ime & dijagram metode)
metoda(ime & dijagram metode)
metoda(ime & dijagram metode)
...

Unutar obrasca, dijagram (toka upravljanja) metode grafički predstavlja zahteve metode. Oznake za korake metode, koje se koriste u ovom dijagramu su uslov (if) (šestougao), blok teksta (pravougaonik), petlja (elipsa) i veznik (crta izmedju koraka metode).

Kompletna dokumentacija OOA modela sastoji se od sledećih komponenti:
1. OOA model u pet nivoa (tematske celine, klasa&objekti, struktura, atributi, metode;
2. specifikacija klasa&objekata
3. dopunska dokumentacija, po potrebi:
tabela kritičnih lanaca izvršavanja
dopunska ograničenja sistema
tabela servisa/stanja objekta u kojima je servis definisan.

Sistem registracije vozila sa svim nivoima (osim atributa) prikazan je na slici 18.

1.6 OOA CASE

CASE-alatom (Computer-Aided Software Engineering, razvoj softvera uz pomoć računara), obično se naziva neko grafičko i tekstuelno sredstvo (alat) ograničenih mogućnosti. P.Coad i E.Yourdon predlažu adekvatnije ime za CASE – CADC, Computer-Aided Drawing and Checking (crtanje i provera uz pomoć računara).

CASE-podrška OOA uključila bi:

1. Notaciju, tj. pet nivoa OOA modela i njihovih simbola, kao i obrazac specifikacije klase&objekata (stanja objekta i prelazi, dijagram metode);

2. Nivoe, kroz uključivanje/isključivanje izbora nekog od pet nivoa OOA;

3. Napredna svojstva, npr. podrška ponovnom korišćenju uz pretraživanje, klasifikaciju i ekstrakciju rezultata prethodne OOA;

4. Proveru modela kroz proveru da:

svaka klasa&objekti ima:

ime

jedinstveno ime unutar modela

više od jednog atributa

jednu ili više instancnih veza

jednu ili više poruka

ima jedinstvena imena atributa (u okviru simbola)

ima jedinstvena imena poruka (u okviru simbola)

svaki obrazac ima:

specifikaciju svakog atributa

specifikaciju svake metode

sadržaj konzistentan sa nivoima

upotrebu svakog imenovanog ulaza-izlaza u okviru teksta iz specifikacije

svaka generalizacija/specijalizacija struktura ima:

više od jednog atributa ili metode po nivou

2 do 4 nivoa

imena atributa jedinstvena unutar prethodnika i sledbenika

imena atributa i metoda koja se ne pojavljuju duž celog nivoa specijalizacije

jedinstvena imena atributa i metoda u generalizacijama za svaki deo rešetke.

1.7 Dometi OOA

Kako se softverska industrija suočavala sa problemima niske produktivnosti i lošeg kvaliteta, kompanije širom sveta su tražile nova rešenja u obliku alata, tehnika i metoda. Strukturne metode (SSA, SP) su važile za rešenje 70-tih i ranih 80-tih godina. CASE alati su se smatrali rešenjem problema druge polovine 80-tih, a 90- tih se sugeriše OO metodologija kao rešenje svih problema.

Ipak, ne postoji magično rešenje za fundamentalno teške probleme vezane sa razvojem softvera. To nisu strukturne metode, to nisu bili CASE alati i neće biti ni OO metodologija. Medju personalom projektnih timova američke industrije aktuelna je poslovica "A fool with a tool is still a fool".

Imajući u vidu prednosti OOA metode, mora se naglasiti da ona ni u kom pogledu ne garantuje uspeh projekta, i da ne može zaštititi razvoj sistema od katastrofe. Neiskusni analitičar može da ne sprovede intervjue na pravi način i ne otkrije sve relevantne objekte sistema, nekooperativni korisnik može nedovoljno da opiše atribute, metode ili instancne veze, i naravno, svaki projekat, bez obzira na alate i metode, može biti unazadjen zbog političkih, upravljačkih problema ili nekompetencije projektnog tima.

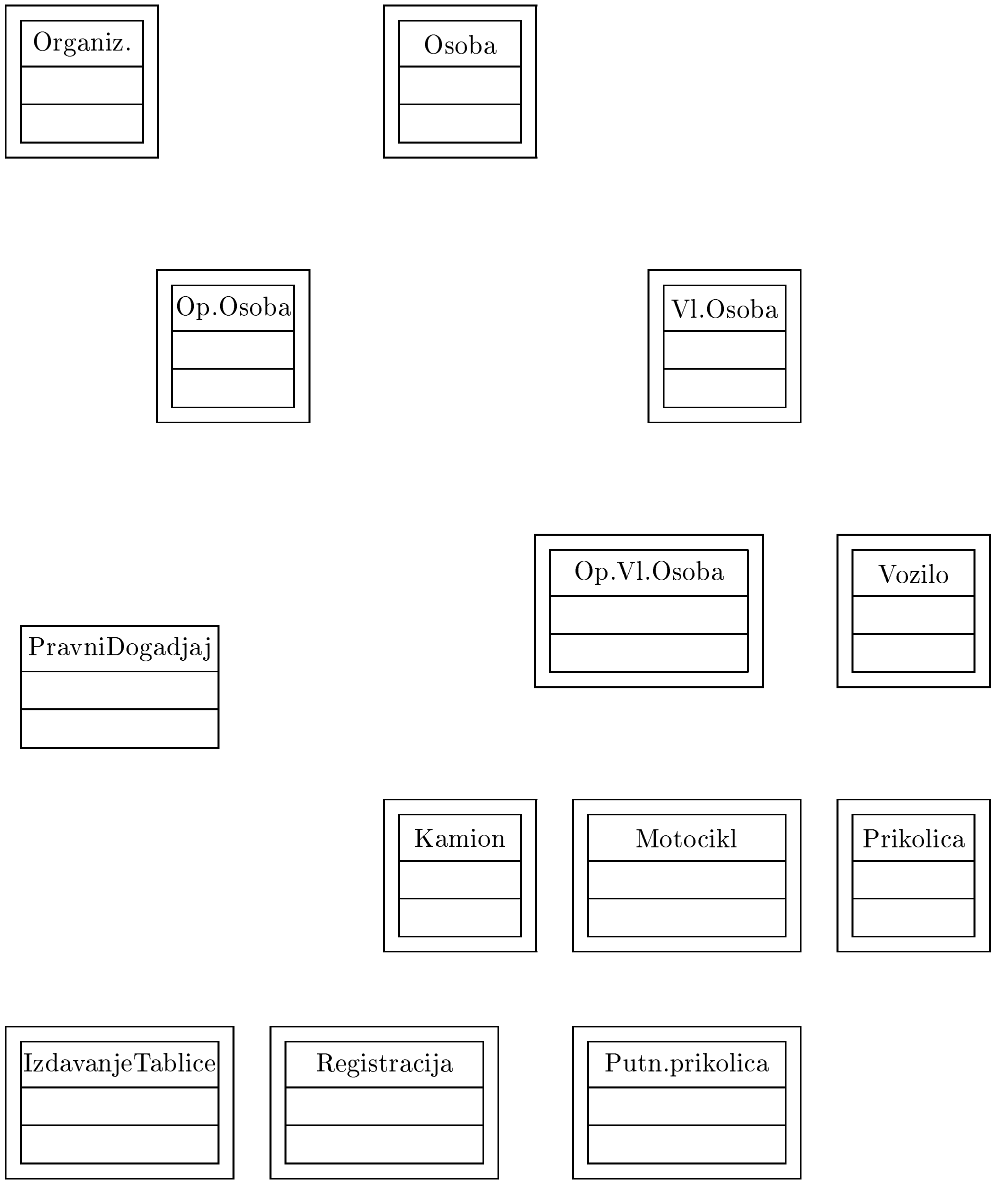

Figure 3: Registracija vozila – nivo klasa&objekata

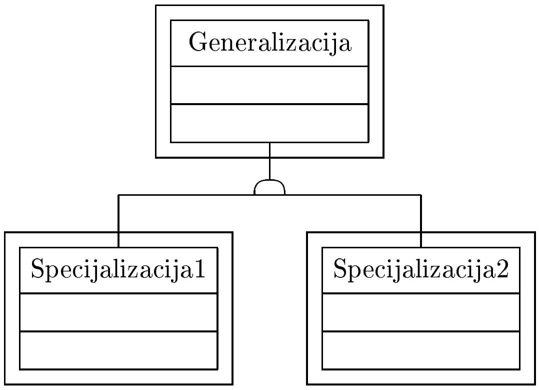

Figure 4: Simbol za strukturu generalizacija/specijalizacija

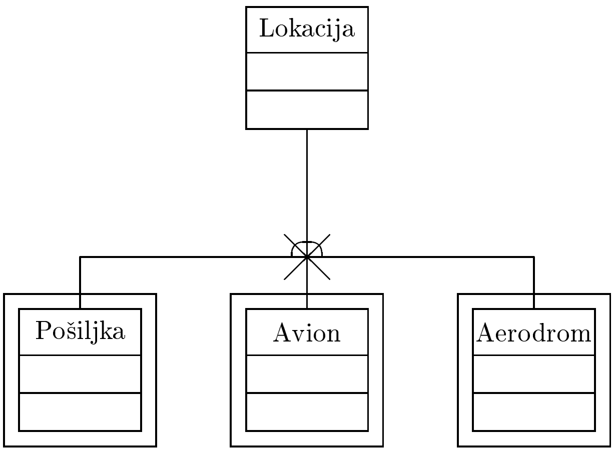

Figure 5: Neodgovarajuća struktura generalizaicija/specijalizacija

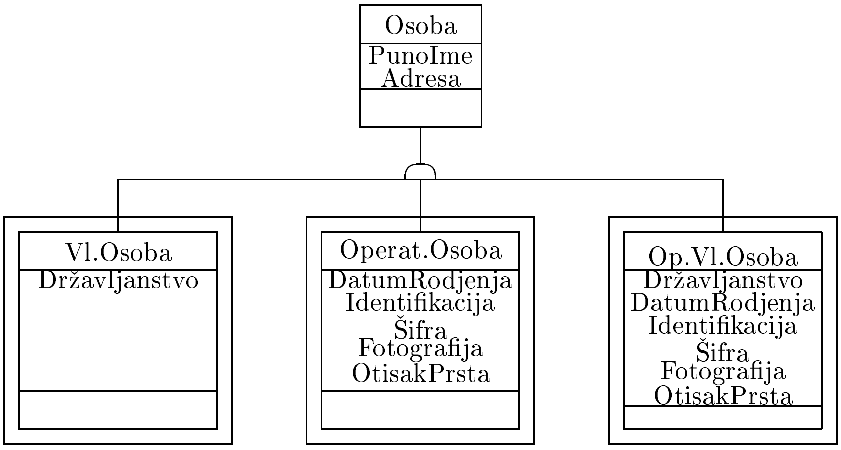

Figure 6: Generalizacija/specijalizacija Osoba, kao hijerarhija

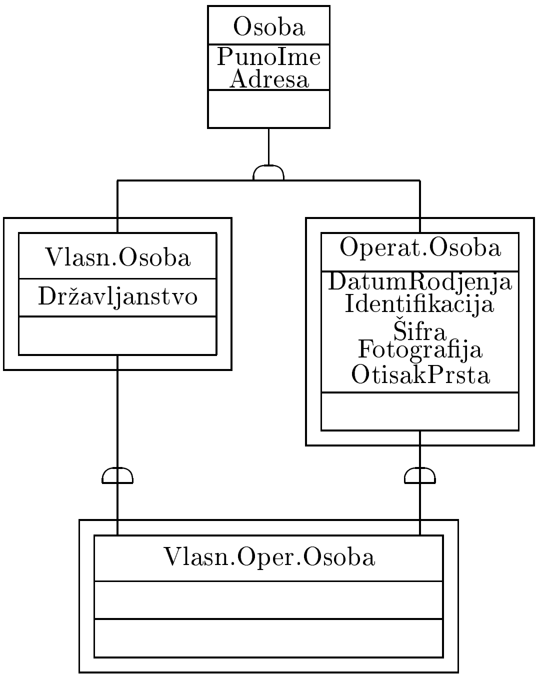

Figure 7: Generalizacija/specijalizacija Osoba, kao rešetka

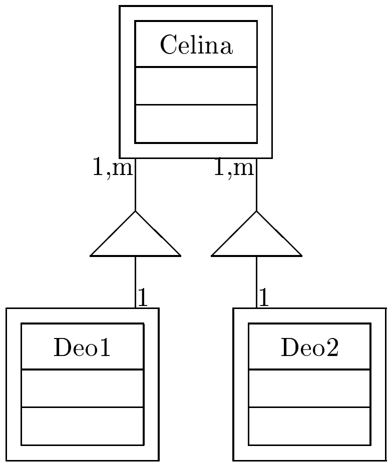

Figure 8: Notacija za strukturu celina/deo

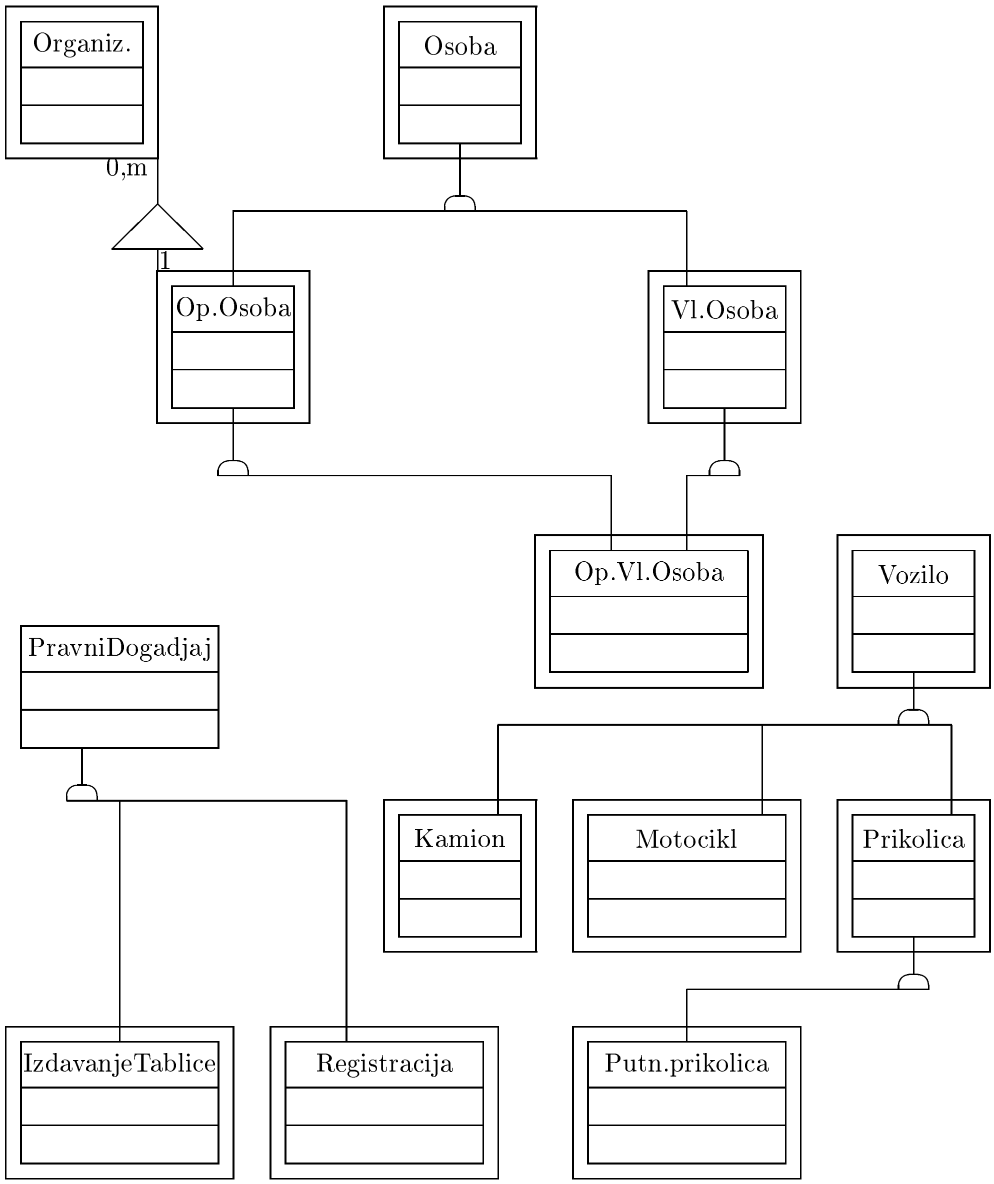

Figure 9: Sistem registracije vozila – nivo struktura

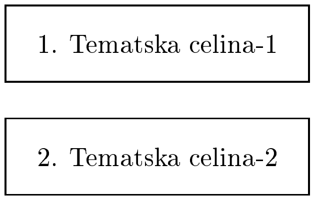

Figure 10: Sažeta notacija za tematsku celinu

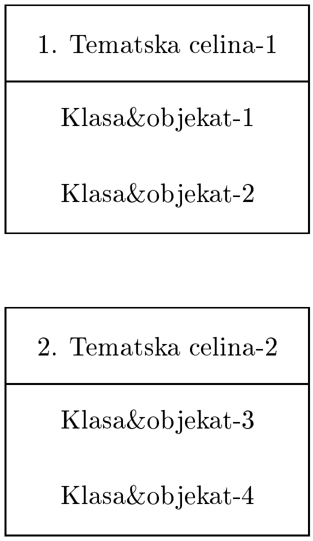

Figure 11: Delimično raširena notacija za tematsku celinu

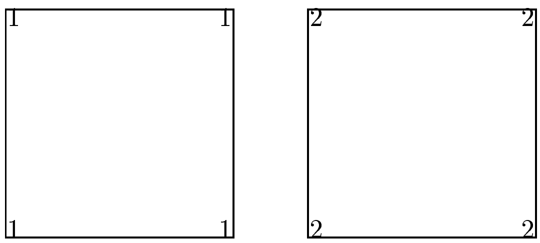

Figure 12: Potpuno raširena notacija za tematsku celinu(kada se uključe i ostali nivoi)

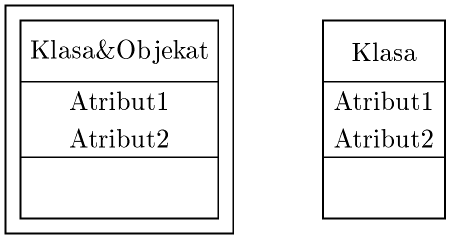

Figure 13: Notacija za atribute

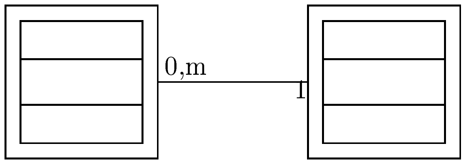

Figure 14: Notacija za instancne veze

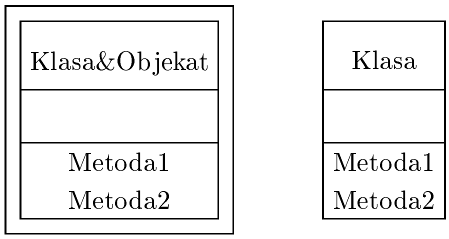

Figure 15: Notacija za metode

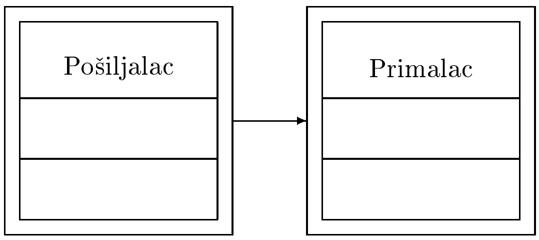

Figure 16: Simbol za poruku

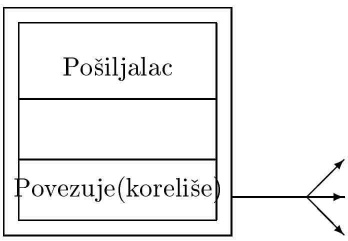

Figure 17: Simbol za poruke većem broju objekata

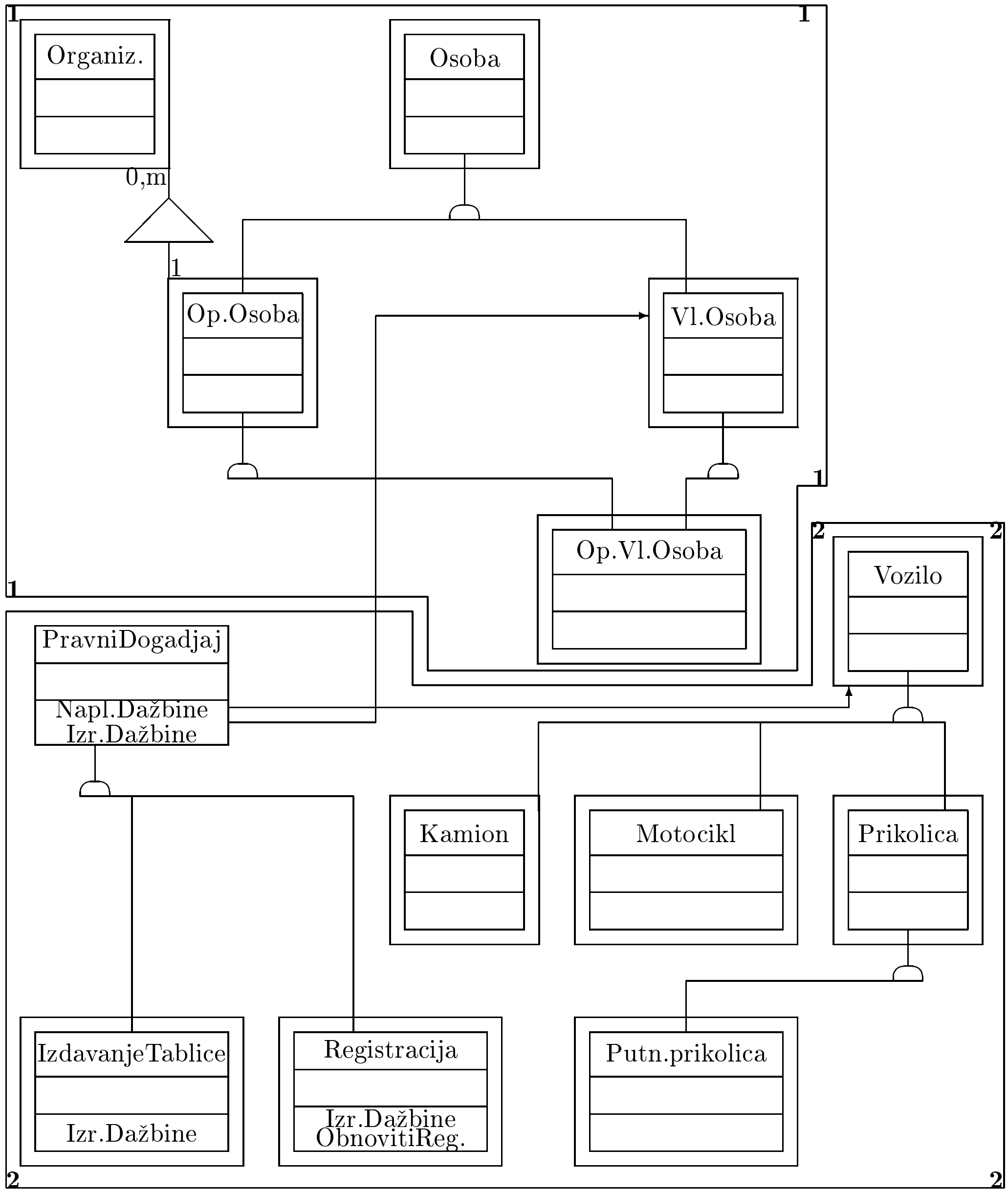

Figure 18: Sistem registracije vozila – svi nivoi osim atributa