

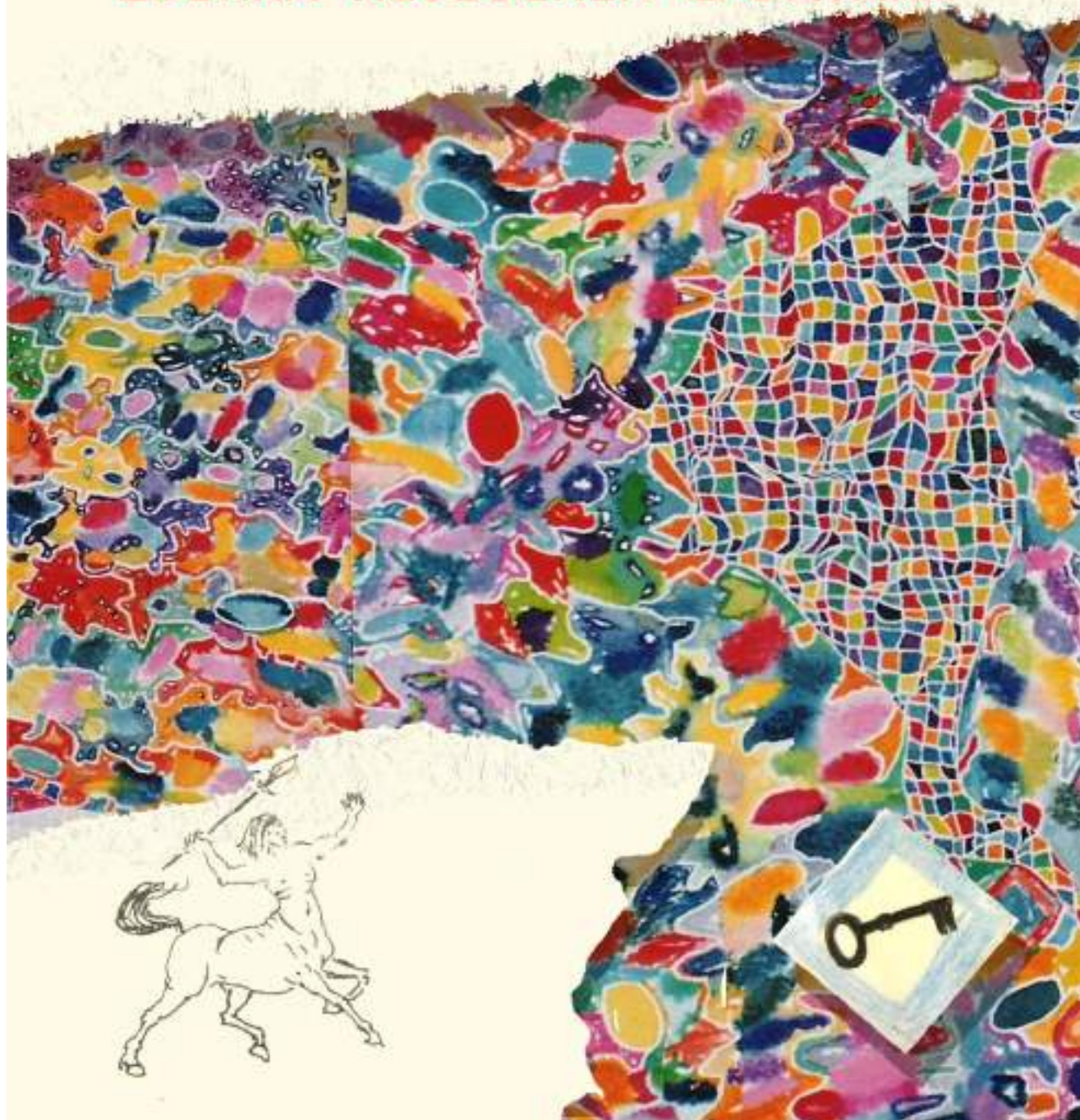
Slavko Marić

Dražen Brđanin

Goran Banjac

# RELACIONE BAZE PODATAKA

ZBIRKA RIJEŠENIH ZADATAKA



## Glava 1

# Konceptualno modelovanje

## 1.1 Uvod

**Projektovanje baze podataka** jeste jedan od najznačajnijih segmenata razvoja sistema sa bazama podataka. Cilj je da se dođe do detaljne specifikacije sveukupne strukture baze podataka (modela) za odgovarajući DBMS (sistem za upravljanje bazom podataka), odnosno do **logičke šeme baze podataka** na implementacionom nivou. Zbog kompleksnosti problema potreban je sistematizovan pristup koji podrazumijeva korištenje odgovarajuće metodologije.

Početna faza projektovanja baze podataka je **konceptualno modelovanje**, koje omogućava specifikaciju sveukupne strukture baze podataka na visokom nivou apstrakcije (konceptualna šema), za šta se najčešće koristi **Model Objekti-Veže (MOV)**. U praksi se često koriste i alternativne, konceptualno slične notacije, kao što je npr. **IE** (*Information Engineering*) notacija. Sa intenzivnim razvojem UML-a posljednjih godina, sve širu primjenu u konceptualnom modelovanju baze podataka ima i **UML dijagram klasa** (*class diagram*). U ovoj glavi ilustrovana je primjena sve tri prethodno pomenute notacije za konceptualno modelovanje.

## 1.2 Model Objekti-Veže

Koncept modela objekti-veže je jednostavan i intuitivan, zasniva se na percepciji realnog svijeta, identifikaciji klasa sličnih entiteta i specifikaciji veza između njih putem odgovarajućih grafičkih koncepata. Model objekti-veže predstavlja **semantički model** podataka jer omogućava ne samo reprezentaciju strukture podataka i veza između njih, nego i specifikaciju njihovog značenja kao i različitih ograničenja. Osnovni MOV koncepti su entiteti i veze.

**Entitet** (objekat) je nešto što postoji u realnom svijetu i što se razlikuje od drugih postojećih objekata. Svaki entitet realnog svijeta karakterišu određena svojstva koja se u MOV modelu reprezentuju **atributima**. Određeno svojstvo entiteta se iskazuje odgovarajućom kvantitativnom vrijednošću odnosnog atributa.

Skup dopuštenih vrijednosti nekog atributa, predstavlja njegov **domen**. Atributi mogu biti: *prosti* ili *složeni*, *jednoznačni* ili *višeznačni*, *bazni* ili *izvedeni*.

Određene entitete realnog svijeta karakterišu ista svojstva, odnosno ti entiteti se mogu reprezentovati istim skupom atributa. Za takve entitete kažemo da su slični, odnosno da su istog tipa. **Tip entiteta** je određen imenom tipa i skupom atributa koji reprezentuju svojstva entiteta tog tipa. Tip entiteta opisuje strukturu i karakteristike entiteta koji pripadaju istom entitetskom skupu. **Entitetski skup** je skup entiteta istog tipa u bazi podataka, u nekom određenom vremenskom trenutku.

**Veza** je asocijacija između određenog broja entiteta. **Tip veze** definiše karakteristike i mogući skup veza između entiteta odnosnih entitetskih tipova. **Vezni skup** je skup veza istog tipa. **Uloge** entiteta u vezi su najčešće implicitno određene. U rekurzivnim vezama između entiteta istog entitetskog skupa potrebno je eksplicitno naznačiti ulogu pojedinih entiteta u vezi, putem odgovarajućeg naziva uloge. Veze mogu imati svoje vlastite, **opisne** attribute.

**Stepen tipa veze** (i odgovarajućeg veznog skupa) je određen brojem entitetskih tipova (i analogno entitetskih skupova) koji participiraju u tipu veze (veznom skupu). Najčešće su **binarne veze**, koje povezuju dva entitetska tipa (entitetska skupa). Veze koje povezuju više od dva entitetska tipa (skupa) su **n-arne veze**.

Tip veze ne samo da definiše koji tipovi entiteta učestvuju u vezama, nego specifikuje i dodatna ograničenja na učešće entiteta u vezama. **Kardinalnost mapiranja** određenog tipa veze, specifikuje maksimalan broj veza u kojima neki entitet odnosnog entitetskog tipa (skupa) može da učestvuje. Kardinalnost mapiranja za binarni tip veze može biti *jedan:jedan*, *jedan:više*, *više:jedan* i *više:više*. **Ograničenje učešća entiteta u vezi** specifikuje, za određeni tip veze, minimalni broj veza u kojima neki entitet odnosnog entitetskog tipa (skupa) mora da učestvuje. Učešće entiteta u vezi može biti *totalno* i *parcijalno*. Učešće nekog entitetskog tipa (skupa) u veznom tipu (skupu) je totalno ako svaki entitet tog entitetskog tipa (skupa) mora učestvovati barem u jednoj vezi iz veznog skupa, a parcijalno ako samo neki entiteti tog entitetskog tipa (skupa) učestvuju u vezama iz veznog skupa.

Značajna forma ograničenja u MOV modelu su ključevi. **Superključ** je skup atributa entitetskog tipa, na kojem svaki entitet entitetskog skupa (tog tipa) ima jedinstvenu vrijednost. **Kandidatski ključ** je minimalni superključ. Neki entitetski tip može sadržavati više kandidatskih ključeva. Za entitetski tip (skup) jedan od njegovih kandidatskih ključeva selektuje se kao **primarni ključ**, odnosno kao osnovni ključ za jedinstvenu identifikaciju entiteta u tom entitetskom skupu. Slično, superključ tipa veze je skup atributa koji jedinstveno identifikuje svaku vezu u veznom skupu odnosnog tipa. Kandidatski/primarni ključ tipa veze je minimalni superključ tipa veze.

Neki entitetski tipovi nemaju attribute koji čine primarni ključ. Takvi entitetski tipovi se nazivaju **slabi entitetski tipovi**, a entitetski skupovi tog tipa **slabi**

**entitetski skupovi.** Entitetski tip koji ima attribute koji čine primarni ključ se naziva **jaki entitetski tip** (i analogno tome, odnosni entitetski skupovi su **jaki entitetski skupovi**). Tip veze koji povezuje slabi i jaki entitetski tip se naziva **identifikujući tip veze**.

Osnovni MOV model nije dovoljan za pogodno predstavljanje specifičnih svojstava entiteta i odnosa između entiteta, pa je MOV model proširen dodatnim konceptima (generalizacija/specijalizacija, unija, agregacija) i u literaturi se često naziva **EER** (*Enhanced Entity-Relationship*) model.

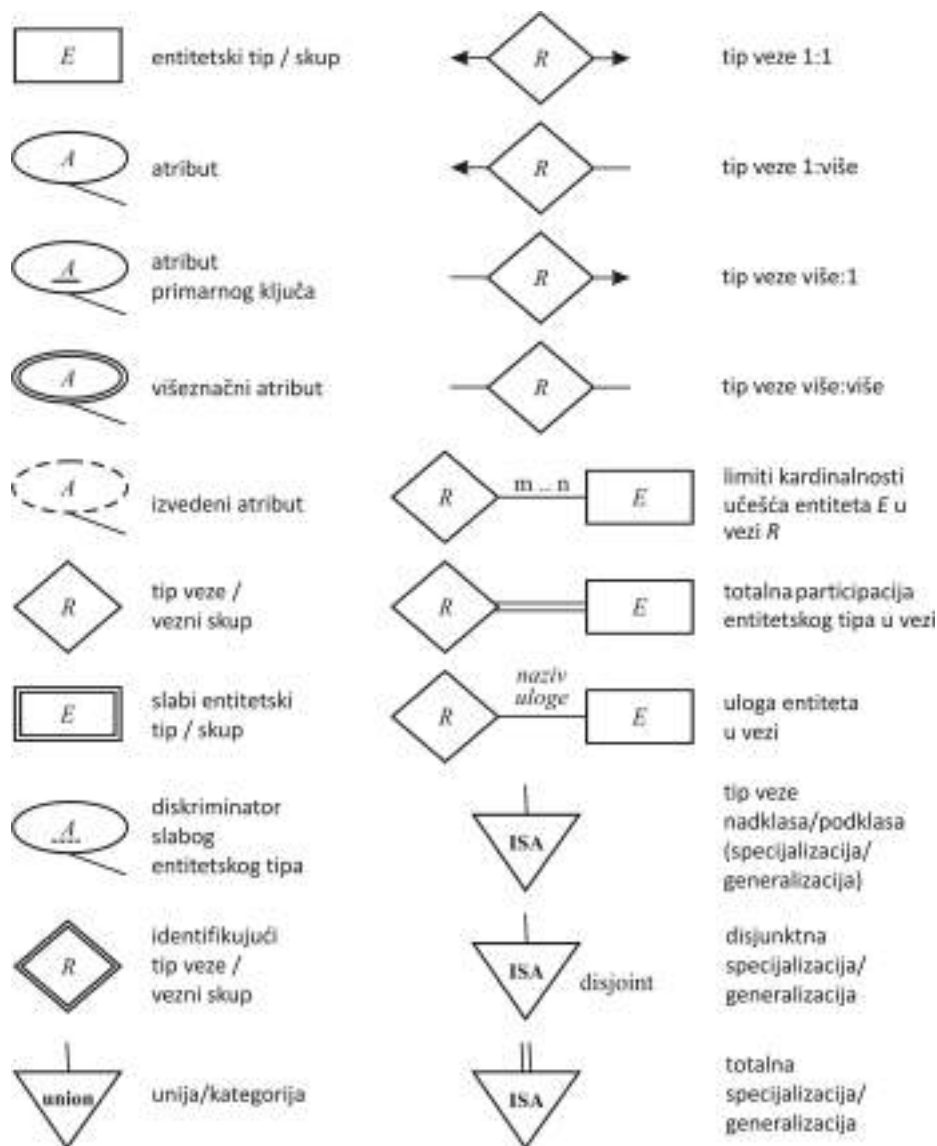
U nekim entitetskim skupovima mogu se uočiti podgrupe entiteta koje se razlikuju po nekim specifičnim svojstvima u odnosu na druge entitete iz entitetskog skupa. Proces identifikacije i reprezentacije ovih podgrupa u okviru nekog entitetskog skupa naziva se **specijalizacija**. Podgrupe entiteta sa specifičnim svojstvima unutar neke grupe se modeluju **entitetskim tipovima nižeg nivoa** koji su sa generalnijim **entitetskim tipom višeg nivoa** povezani **ISA** vezom. ISA veza reprezentuje činjenice da su (specijalizovani) entiteti entitetskog tipa nižeg nivoa ujedno i entiteti višeg entitetskog tipa. Za entitetske tipove višeg nivoa koristi se i termin **superklasa** ili **natklasa**, a za entitetske tipove nižeg nivoa termin **potklasa**. Entitetski tipovi nižeg nivoa (potklase) nasljeđuju sve attribute entitetskog tipa višeg nivoa (superklase) sa kojim su povezani ISA vezom, i imaju, dodatno, svoje specifične attribute. Analogno nasljeđivanju atributa, nasljeđuju se i tipovi veza u kojima entitetski tip višeg nivoa (superklasa) učestvuje. Do identifikacije generalnijih entitetskih tipova višeg nivoa i entitetskih tipova nižeg nivoa može se doći i **generalizacijom**, obrnutim procesom u odnosu na specijalizaciju. Ukoliko se u toku analize sistema prvo identifikuju neki entitetski tipovi (skupovi) a zatim utvrdi njihova sličnost, onda se može izvršiti generalizacija i definisanje entitetskog tipa (skupa) višeg nivoa (superklase), čiji bi atributi bili zajednički atributi polaznih entitetskih tipova.

Za određenu specijalizaciju/generalizaciju se mogu specificovati slijedeća ograničenja: uslov članstva entiteta entitetskog skupa višeg nivoa u entitetskom skupu nižeg nivoa (*definisano uslovom* ili *korisnički definisano*), ograničenje na participaciju entiteta višeg entitetskog skupa u entitetskim skupovima nižeg nivoa (*disjunktno* i *preklapajuće*), i ograničenje na kompletnost učešća entiteta entitetskog tipa višeg nivoa u entitetskim skupovima nižeg nivoa (*totalno* i *parcijalno*).

Modelovanje unija i agregacija uglavnom nije uključeno i podržano u alatima za konceptualno modelovanje. Koncept **unije** ili **kategorije** omogućuje modelovanje veza superklasa/potklasa, pri čemu veza povezuje više superklasa različitog tipa sa jednom potklasom. Potklasa (kategorija/unija) može da sadrži entitete različitih tipova, koji su elementi superklasa sa kojima je kategorija/unija povezana, odnosno kategorija (unija) je podskup unije entiteta različitih tipova, članova superklasa. **Agregacija** je apstrakcija kojom se veze tretiraju kao entiteti višeg nivoa, i koje, kao takvi apstraktni entiteti višeg nivoa, mogu učestvovati u drugim vezama.

### 1.2.1 MOV dijagram

U MOV modelu, konceptualna šema reprezentuje se grafičkom notacijom poznatom pod nazivom **MOV dijagram**<sup>1</sup>. MOV notacija (niti osnovna niti proširena) nije standardizovana. Na sl.1.1 dat je sumarni pregled MOV notacije koja se koristi u ovoj knjizi.



Slika 1.1 Sumarni pregled MOV notacije

<sup>1</sup> U literaturi se često koristi i termin E-R (*Entity-Relationship*) dijagram.

## 1.2.2 Proces projektovanja konceptualnog modela

Ne postoji metodologija koja propisuje precizne korake koji bi uvijek vodili ka jedinstvenom i jednoznačnom konceptualnom modelu nekog realnog sistema. U procesu projektovanja projektanti se susreću sa različitim mogućnostima i dilemama. Specifične percepcije realnog sistema mogu rezultovati različitim zaključcima i različitim konceptualnim modelima za isti realni sistem. Proces izgradnje konceptualnog modela baze podataka je gotovo uvijek iterativan, pri čemu se prvo kreira početni model, a zatim se u iterativnom procesu u svakoj iteraciji vrši analiza, otklanjaju nedostaci, rješavaju projektne dileme i poboljšava model.

Najopštiji pristup jeste da se inicijalna identifikacija entitetskih skupova (tipova) i veza vrši na osnovu specifikacije zahtjeva, odnosno informacionih potreba sistema za koji se projektuje baza podataka, pri čemu se (posebno u alatima za automatsko projektovanje) koriste heuristička, odnosno neformalna pravila za identifikaciju karakterističnih koncepata. Tako su opšte imenice, koje reprezentuju objekte, pojave ili događaje realnog sistema, kandidati za entitetske tipove, a glagoli kandidati za vezne tipove. To ne znači da sve opšte imenice treba reprezentovati entitetskim tipovima, niti sve glagole veznim tipovima. Imenice mogu da imaju različite uloge u rečenicama (subjekti, objekti, apozicije itd.). *Subjekti* u rečenicama najčešće reprezentuju entitete koji treba da se modeluju odgovarajućim entitetskim tipom. *Predikati* (glagoli i glagolske fraze) koji u rečenicama povezuju subjekte i objekte (već reprezentovane entitetskim tipovima) reprezentuju se veznim tipovima.

U primjerima koji slijede, konceptualni model podataka projektovan je inkrementalnim pristupom na osnovu zadate specifikacije informacionih potreba, što podrazumijeva analizu zadate tekstualne specifikacije rečenicu po rečenicu i identifikaciju entiteta i njihovih veza sa prethodno identifikovanim entitetima. Ovakav pristup inkrementalno (korak po korak) vodi ka ciljnom konceptualnom modelu i pogodan je za ilustraciju primjene različitih (alternativnih) koncepata.

---

**Primjer 1.1**    Projektovati konceptualni model podataka univerziteta primjenom MOV notacije.

---

### Rješenje.

Pretpostavimo da smo u procesu analize identifikovali informacione potrebe hipotetičkog univerzitetskog sistema. S obzirom na njihov obim, razmatraćemo tekstualnu specifikaciju i projektovati konceptualni model po dijelovima koji predstavljaju logičke cjeline.



### **Specifikacija informacionih potreba (I. dio)**

*Univerzitet je obrazovna institucija koja ima naziv, adresu (sjedište) i određen broj telefonskih brojeva za kontakt, u čiji sastav ulazi određeni broj fakulteta. Svaki fakultet ima naziv, adresu i određen broj telefonskih brojeva za kontakt. Na fakultetima se izvodi nastava na jednom ili više studijskih programa. Studijski program karakteriše naziv studijskog programa, ciklus studija (I ciklus – dodiplomski, II ciklus – master/postdiplomski, III ciklus – doktorski studij), trajanje (broj godina/semestara), ukupan broj ECTS kredita koji se stiče studiranjem na studijskom programu, kao i naziv zvanja koji se stiče završetkom studijskog programa. Nastavnim planom definišu se predmeti koje sadrži studijski program, semestar u kojem se po nastavnom planu izvodi nastava na predmetu i tip predmeta (obavezni/izborni). Predmete karakteriše jedinstven identifikator/kod, naziv predmeta i broj ECTS bodova. Svi predmeti su jednosemestralni. Jedan predmet može da se izvodi na više studijskih programa, jednog ili više fakulteta. Za svaki predmet je matičan određeni fakultet.*

### **Projektovanje konceptualnog modela**

Analizom prve rečenice date tekstualne specifikacije zaključujemo da je *univerzitet* entitet od interesa, tj. entitet čija su svojstva bitna u konkretnom sistemu. Dakle, prvi entitetski tip u ciljnom MOV dijagramu jeste *UNIVERZITET*. Univerzitet karakterišu *naziv, adresa, brojevi telefona*, kao i *fakulteti koji ga sačinjavaju*. Prva karakteristika predstavlja jednostavno svojstvo univerziteta, koje se modeluje prostim (atomičnim) atributom *Naziv*. Druga karakteristika predstavlja složeno svojstvo univerziteta, jer adresu čine *naziv ulice, broj, naziv mjesta i poštanski broj*. U inicijalnoj fazi razvoja konceptualnog modela i ovo svojstvo može da se reprezentuje prostim atributom *Adresa*, koji u kasnijoj fazi može da se modeluje složenim atributom. Treća karakteristika ukazuje na to da univerzitet ima veći broj telefonskih brojeva. Ovo svojstvo može da se modeluje višeznačnim atributom *Telefon*. Četvrtu karakteristiku univerziteta (*fakulteti koji ga sačinjavaju*) mogli bismo (slično telefonskim brojevima) da modelujemo višeznačnim atributom. Međutim, fakulteti nisu samo svojstva univerziteta, kao što bi se moglo zaključiti iz prve rečenice date tekstualne specifikacije, nego se oni višestruko pominju u narednim rečenicama kao entiteti koji imaju svoja svojstva. Dakle, fakulteti treba da se modeluju zasebnim entitetskim tipom *FAKULTET*.

Tipično u sistemu, za koji se kreira konceptualni model, postoji veći broj entiteta svakog entitetskog tipa. Identifikovani entitetski tip *UNIVERZITET* je specifičan u pogledu broja entiteta tog tipa. U konkretnom slučaju imamo samo jedan univerzitet. I bez obzira na to, treba da selektujemo primarni ključ datog entitetskog tipa. Atribut *Naziv* jedinstveno identifikuje univerzitet, pa možemo da ga izaberemo za primarni ključ. Preostali atributi nisu pogodni za primarni ključ, jer je atribut *Telefon* višeznačan, a *Adresa* nema semantiku identifikatora i može više puta da se mijenja (promjena lokacije sjedišta univerziteta, promjena naziva ulice i sl.).

U ovoj fazi projektovanja, MOV dijagram je kao na sl. 1.2.



Slika 1.2 Inicijalni MOV dijagram

Svojstva svakog fakulteta su, slično univerzitetu, njegov *naziv*, *adresa* i *brojevi telefona*, pa po analogiji dolazimo do zaključka da entitetski tip *FAKULTET* ima jednoznačne atribute *Naziv* i *Adresa*, te višeznačni atribut *Telefon*. Karakteristika fakulteta jesu i studijski programi na kojima se izvodi nastava. Međutim, modelovanje studijskih programa korespondentnim višeznačnim atributom nije pogodno, jer se u nastavku tekstualne specifikacije studijski programi višestruko pominju kao entiteti koji imaju svoja svojstva i veze sa drugim entitetima. Dakle studijski programi treba da se modeluju entitetskim tipom *STUD\_PROGRAM*.

Svi fakulteti koji ulaze u sastav univerziteta imaju različita imena, tj. ne postoje dva istoimena fakulteta u okviru univerziteta. Dakle, vrijednost atributa *Naziv* je jedinstvena za svaki fakultet, pa se atribut *Naziv* može izabrati za primarni ključ entitetskog tipa *FAKULTET*.

Identifikujemo još veze između entiteta tipa *UNIVERZITET* i *FAKULTET*. S obzirom na to da univerzitet u svom sastavu ima određen broj fakulteta, veze između entiteta ova dva tipa možemo da modelujemo veznim tipom *IMA\_U\_SASTAVU*. Opciono bismo mogli, s obzirom na pripadnost fakulteta univerzitetu, ovaj vezni tip da nazovemo *PRIPADA*.

Univerzitet u svom sastavu ima *više* fakulteta, a svaki fakultet pripada samo *jednom* univerzitetu. Dakle, kardinalnost mapiranja je 1:*više* (posmatrano sa strane univerziteta). *Svaki* fakultet mora da pripada univerzitetu, što znači da *svi* fakulteti učestvuju u vezama tipa *IMA\_U\_SASTAVU*. Dakle, učešće entiteta tipa *FAKULTET* u vezama tipa *IMA\_U\_SASTAVU* je *totalno*. S druge strane, univerzitet mora da ima fakultete u svom sastavu, što znači da je i učešće entiteta tipa *UNIVERZITET* u ovom tipu veze *totalno*.

U ovoj fazi projektovanja, MOV dijagram je kao na sl. 1.3.



Slika 1.3 MOV dijagram nakon identifikacije entitetskih tipova *UNIVERZITET* i *FAKULTET*



Opciono bismo telefonske brojeve (univerziteta i fakulteta) mogli modelovati zasebnim entitetskim tipovima, a ne višeznačnim atributima. Ovo bi posebno bilo bitno ako bi, osim broja telefona za kontakt, bili značajni i drugi podaci kao što je mjesto kontakta (npr. centrala, rektor/dekan, prorektori/prodekani, računovodstvo itd.), model telefona, serijski i/ili inventarni broj, itd. S obzirom na to da se u datoj specifikaciji ne navode dodatna svojstva, brojeve telefona smo modelovali višeznačnim atributima.

Svaki studijski program ima slijedeća svojstva: *naziv*, *ciklus studija*, *trajanje*, *ukupan broj ECTS kredita* te *naziv zvanja*, koja modelujemo odgovarajućim prostim atributima: *Naziv*, *Ciklus*, *Trajanje*, *UkupnoECTS* i *Zvanje*, respektivno. Dodatna karakteristika svakog studijskog programa jesu i predmeti na kojima se izvodi nastava u okviru studijskog programa. Međutim, iz nastavka tekstualne specifikacije slijedi da su predmeti također entiteti koji imaju svoja svojstva. Dakle, predmeti treba da se modeluju zasebnim entitetskim tipom *PREDMET*.

Opciono bismo ciklus studija također mogli da modelujemo zasebnim entitetskim tipom, a ne atributom. Međutim, prema datoj specifikaciji, ovo svojstvo studijskog programa reprezentuje samo činjenicu o nivou univerzitetskog obrazovanja, pri čemu se o ciklusima ne vode nikakve dodatne informacije. Zato u konkretnom slučaju nema potrebe za modelovanjem ciklusa studija zasebnim entitetskim tipom.

Odredimo primarni ključ entitetskog tipa *STUD\_PROGRAM*. Nijedan od atributa ne može samostalno da predstavlja primarni ključ, jer se ne može očekivati da će neki od identifikovanih atributa garantovano imati vrijednosti koje omogućavaju jedinstvenu identifikaciju svakog studijskog programa ponaosob. Može se očekivati da neće postojati dva istoimena studijska programa na različitim fakultetima, ali se može očekivati da fakultet ima više istoimenih studijskih programa na različitim ciklusima. Drugim riječima, može se očekivati da će kombinacija naziva studijskog programa i ciklusa studija biti jedinstvena na univerzitetu, pa kombinaciju atributa *Naziv* i *Ciklus* možemo da selektujemo kao primarni ključ. Međutim, često se u praksi umjesto kombinacije atributa, kao primarni ključ koristi tzv. *surogat* ključ, tj. dodatni atribut koji se uvodi u cilju jedinstvene identifikacije entiteta datog entitetskog tipa. U konkretnom primjeru možemo da uvedemo dodatni atribut *IdSP*, koji predstavlja šifru (identifikacioni kod) studijskog programa.

Između entiteta tipa *FAKULTET* i *STUD\_PROGRAM* postoje veze tipa *IMA* u smislu *izvodi se nastava*. Iz specifikacije se vidi da *svaki* fakultet ima *jedan ili više* studijskih programa, a da *svaki* studijski program pripada samo *jednom* fakultetu. Dakle, kardinalnost mapiranja je 1:*više* posmatrano sa strane fakulteta, a učešće entiteta oba entitetska tipa *totalno* (*svaki* fakultet ima studijske programe i *svaki* studijski program pripada nekom fakultetu).

U ovoj fazi projektovanja, nakon potpune identifikacije entitetskog tipa *STUD\_PROGRAM* i veznog tipa *IMA*, MOV dijagram je kao na sl. 1.4.



**Slika 1.4** MOV dijagram nakon identifikacije entitetskog tipa *STUD\_PROGRAM*

Svaki predmet ima slijedeća svojstva: *jedinstveni identifikator/kod*, *naziv* i *broj ECTS bodova*, koja modelujemo odgovarajućim atributima: *IdPredmeta*, *Naziv* i *ECTS*, respektivno. Činjenica da su svi predmeti jednosemestralni ne treba da se modeluje dodatnim atributom, jer bi svi entiteti tipa *PREDMET* na tom atributu imali istu vrijednost. Pošto atribut *IdPredmeta* predstavlja jedinstveni identifikator predmeta, selektujemo ga kao primarni ključ.

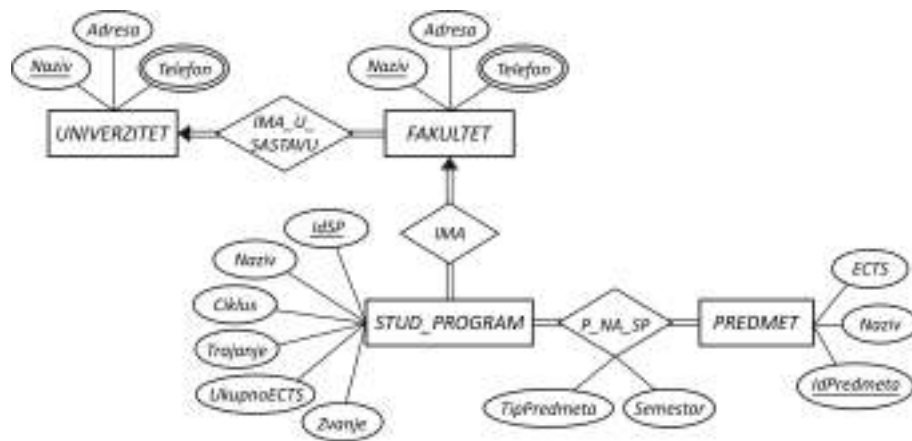
Prema zadatoj specifikaciji, *studijski program* ima *nastavni plan*. Ta činjenica može da se modeluje zasebnim entitetskim tipom *NASTAVNI\_PLAN* koji je vezama tipa *IMA* povezan sa entitetskim tipom *STUD\_PROGRAM*. Entitetski tip *NASTAVNI\_PLAN* bio bi slabi entitetski tip (jer je nastavni plan egzistencijalno zavisian od studijskog programa), a vezni tip *IMA* bio bi identifikujući sa kardinalnošću mapiranja 1:1 i totalnom participacijom entiteta oba tipa (svaki studijski program mora da ima nastavni plan, a svaki nastavni plan pripada nekom studijskom programu). S obzirom na to da se u specifikaciji ne ukazuje na mogućnost postojanja više različitih nastavnih planova (koji važe za različite generacije studenata) za isti studijski program, nema potrebe da se uvodi zaseban entitetski tip za modelovanje nastavnih planova, nego se i studijski program i njegov nastavni plan mogu smatrati jedinstvenim entitetom sa jedinstvenim atributima. U konkretnom slučaju, nastavni plan nema svoja dodatna svojstva (npr. datum usvajanja, broj odluke itd.), pa entitetskom tipu *STUD\_PROGRAM* ne treba dodavati nove atribute.

Prema specifikaciji, *nastavni plan* definiše *predmete* koji pripadaju studijskom programu, *semestar* u kojem se izvodi nastava na svakom predmetu i *tip predmeta* (obavezni/izborni). Dodatno je još specifikovano da neki predmet može da se izvodi na više studijskih programa različitih fakulteta. Dakle, entiteti tipa *STUD\_PROGRAM* (koji ujedno predstavlja i studijski program i njegov nastavni plan) i *PREDMET* vezani su vezama tipa *P\_NA\_SP* (predmet na studijskom programu). *Svaki* studijski program (u svom nastavnom planu) sadrži

više predmeta, a *svaki* predmet je sadržan u nastavnom planu *jednog ili više* studijskih programa. Dakle, kardinalnost mapiranja je *više:više*, a učešće entiteta oba entitetska tipa *totalno* (niti može da postoji studijski program koji u svom nastavnom planu nema predmete, niti mogu da postoje predmeti koji ne pripadaju nekom studijskom programu).

Prema specifikaciji, neki predmet može da se izvodi u okviru različitih studijskih programa, u različitim semestrima i pod različitim uslovima – negdje kao obavezni, a negdje kao izborni. Dakle, *semestar* i *tip predmeta* svojstva su veze (koja je definisana nastavnim planom) konkretnog predmeta i konkretnog studijskog programa, pa se modeluju odgovarajućim atributima *Semestar* i *TipPredmeta* veznog tipa *P\_NA\_SP*. Treba uočiti da vezni tip *P\_NA\_SP* opisuje nastavni plan odnosno studijskog programa, tj. raspored i status predmeta po semestrima.

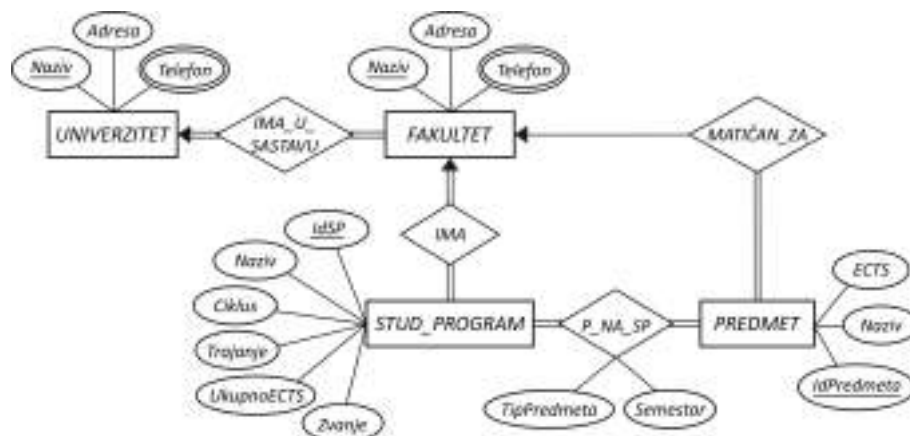
U ovoj fazi projektovanja, nakon identifikacije entitetskog tipa *PREDMET* i veznog tipa *P\_NA\_SP*, MOV dijagram je kao na sl. 1.5.



**Slika 1.5** MOV dijagram nakon identifikacije entitetskog tipa *PREDMET*

Još treba da se identifikuju veze fakulteta i predmeta po osnovu matičnosti. Prema zadatoj specifikaciji, za *svaki* predmet matičan je određeni (*jedan*) fakultet. S druge strane, fakultet može biti matičan za *više* predmeta. Dakle, kardinalnost mapiranja veza tipa *MATIČAN\_ZA* je *1:više* posmatrano sa strane fakulteta. Ako pretpostavimo da mogu da postoje i fakulteti za multidisciplinarne studije, koji nisu matični ni za jedan predmet, zaključujemo da je učešće entiteta tipa *PREDMET totalno* (za *svaki* predmet matičan je neki fakultet), a da je učešće entiteta tipa *FAKULTET parcijalno* (mogu da postoje fakulteti koji nisu matični ni za jedan predmet).

Dodavanjem veznog tipa *MATIČAN\_ZA* dobijamo MOV dijagram kao na sl. 1.6, koji predstavlja konceptualni model podataka kreiran na osnovu zadate tekstualne specifikacije.



**Slika 1.6** MOV dijagram kreiran na osnovu tekstualne specifikacije informacionih potreba hipotetičkog univerzitetskog sistema (1. dio)

Kreirani konceptualni model rezultat je detaljne analize zadate tekstualne specifikacije, koja je sprovedena korak po korak. Do istog rezultata bi se došlo i primenom ranije navedenih neformalnih pravila. Sve opšte imenice (koje predstavljaju subjekte u rečenicama, a reprezentuju objekte, pojave i događaje realnog sistema) bile bi reprezentovane odnosnim entitetskim tipovima. Tako bi se modelovali *univerzitet*, *fakultet*, *studijski program* i *predmet*, a *nastavni plan* bi se nakon dodatne analize (nema sopstvene atribute, zavisi od studijskog programa i postoji samo jedan nastavni plan za svaki studijski program) objedinio sa *studijskim programom*. S druge strane, glagolske fraze *ima u sastavu*, *matičan za*, itd. bile bi reprezentovane odgovarajućim veznim tipovima između odnosnih entitetskih tipova.

#### **Specifikacija informacionih potreba (2. dio)**

Na univerzitetu je neophodno voditi evidenciju o različitim osobama. Svaka osoba ima jedinstven identifikator (matični broj), prezime, ime, datum rođenja i adresu. Za odvijanje osnovnih procesa na univerzitetu – nastavnih i naučno-istraživačkih, ključne grupe osoba su nastavnici, asistenti/saradnici i studenti. Ostale osobe su u funkciji podrške za odvijanje osnovnih procesa na univerzitetu. Nastavu na fakultetu izvode zaduženi nastavnici i saradnici (asistenti). Na izvođenju nastave moraju biti angažovani nastavnici, dok na nekim (teorijskim) predmetima vježbe nisu obavezne. Nastavnici i asistenti su osobe koje mogu biti zaposleni na univerzitetu, ili spoljni saradnici. Svaki zaposleni ima odgovarajuću platu. Spoljne saradnike takođe karakteriše institucija u kojoj su stalno zaposleni. Za angažovanje na univerzitetu, spoljni saradnici zaključuju ugovor. Nastavnici i asistenti imaju nastavnička, odnosno asistentska zvanja. Nastavnici i asistenti mogu izvoditi nastavu na više predmeta i na više studijskih programa/fakulteta univerziteta. Dekan koji rukovodi fakultetom i rektor koji rukovodi univerzitetom, biraju se iz redova zaposlenih nastavnika univerziteta.

### Projektovanje konceptualnog modela

Na osnovu analize prve dvije rečenice zaključujemo da se na univerzitetu vodi evidencija o različitim osobama te da su svojstva svake osobe: *jedinstveni identifikator (matični broj), prezime, ime, datum rođenja i adresa*. Dakle, osoba je entitet od interesa i treba da se reprezentuje odgovarajućim entitetskim tipom *OSOBA*, čiji su atributi: *JMB, Prezime, Ime, DatumRođenja i Adresa*. Primarni ključ je *JMB*, jer predstavlja jedinstveni identifikator svake osobe.

To što se na univerzitetu vodi evidencija o različitim osobama, još uvijek ne ukazuje na potrebu uvođenja nekog novog entitetskog tipa niti tipa veze između osoba i nekih drugih entiteta. Inicijalni MOV dijagram za drugi dio specifikacije informacionih potreba, nakon identifikacije entitetskog tipa *OSOBA*, dat je na sl. 1.7 (radi preglednosti, dio sa sl. 1.6 nije prikazan).



**Slika 1.7** Inicijalni MOV dijagram za drugi dio tekstualne specifikacije

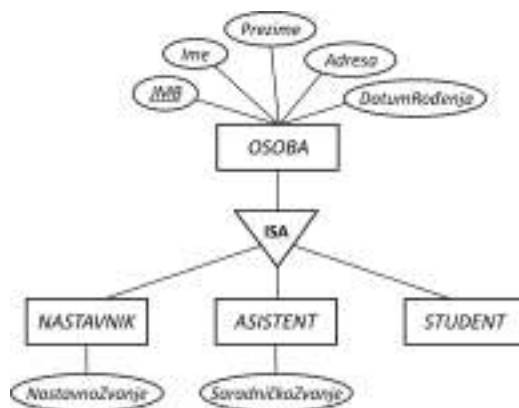
Za odvijanje osnovnih procesa na univerzitetu ključne grupe osoba su *nastavnici, asistenti i studenti*. Svaka od tih grupa ima karakteristične *uloge* (izvode ili slušaju nastavu, rukovode fakultetom itd.) i *svojstva* (plata, zvanje, itd.), pa treba da se modeluje korespondentnim entitetskim tipom *NASTAVNIK, ASISTENT i STUDENT*, respektivno. Ovi entitetski tipovi reprezentuju svojevrsne podvrste (*potklase*) osobe, a entitetski tip *OSOBA* uopštenu reprezentaciju (*natklasa*) svih osoba. Veze natklase i potklasa modelujemo specijalizacijom (ISA veza).

Prema tekstualnoj specifikaciji mogu da postoje i druge osobe, koje su u funkciji podrške osnovnim procesima. Dakle, postoje i osobe koje nisu ni nastavnici ni asistenti ni studenti, što znači da je specijalizacija entitetskog tipa *OSOBA*, po datom kriterijumu, *parcijalna*. S obzirom na to da su asistenti najčešće i studenti postdiplomskih ili doktorskih studija, data specijalizacija *nije disjunktna, već preklapajuća*.

Svojstva koja su zajednička za entitete potklasa već su identifikovana i reprezentovana atributima natklase. Zato te attribute (u konkretnom slučaju: *JMB, Prezime, Ime, DatumRođenja i Adresa*) ne treba dodavati potklasama. Treba identifikovati i dodati samo one attribute koji reprezentuju svojstva karakteristična za konkretnu potklasu.

Analizom specifikacije uočavamo da nastavnici i asistenti imaju nastavnička, odnosno asistentska (saradnička) zvanja. Pošto zvanja nisu detaljno specifikovana niti se očekuje da se za nastavnike i asistente vode evidencije o izborima (datum

izbora, broj odluke, naziv zvanja, period itd.), zaključujemo da se zvanja mogu modelovati odgovarajućim atributima *NastavnoZvanje* i *SaradničkoZvanje* odnosnih entitetskih tipova *NASTAVNIK* i *ASISTENT*, respektivno. U suprotnom, bilo bi neophodno uvesti odgovarajuće entitetske tipove *NASTAVNO\_ZVANJE* i *SARADNIČKO\_ZVANJE*, a zatim identifikovati i tipove veza između entiteta tipa *NASTAVNIK* i *NASTAVNO\_ZVANJE*, te *ASISTENT* i *SARADNIČKO\_ZVANJE*. U ovoj fazi projektovanja, MOV dijagram je kao na sl. 1.8 (dio sa sl. 1.6 nije prikazan).



**Slika 1.8** MOV dijagram nakon identifikacije potklasa *NASTAVNIK*, *ASISTENT* i *STUDENT*

Specifikacija definiše dvije vrste angažmana na univerzitetu. Osim *stalno zaposlenih* radnika, univerzitet može da angažuje i *spoljne saradnike*. Ovo su dvije specifične podvrste osoba. Zaposlene karakteriše *plata*, a spoljne saradnike *ugovor* i *institucija* u kojoj su zaposleni. Zbog toga ove dvije grupe treba da se modeluju odgovarajućim potklasama. Potklasa *ZAPOSLeni*, sa atributom *Plata*, reprezentuje stalno zaposlene, a potklasa *HONORARNI*, sa atributima *Ugovor* i *Institucija*, reprezentuje spoljne saradnike. Ne ostvaruju sve osobe na univerzitetu primanja (npr. studenti). Zato je specijalizacija po ovom osnovu *parcijalna*. Angažovani ne mogu istovremeno da budu i spoljni saradnici i stalno zaposleni na univerzitetu, pa je specijalizacija *disjunktna*. Nakon identifikacije potklasa *ZAPOSLeni* i *HONORARNI*, MOV dijagram je kao na sl. 1.9 (dio sa sl. 1.6 nije prikazan).

Prema specifikaciji, s obzirom na angažman, razlikujemo dvije vrste nastavnika, i to: stalno zaposlene nastavnike i honorarne nastavnike. To znači da entitetski tip *NASTAVNIK* treba da se specijalizuje potklasama *NASTAVNIK\_ZAP* i *NASTAVNIK\_HON* koje reprezentuju te dvije vrste. Pošto nastavnik ne može istovremeno da bude stalno zaposlen na univerzitetu i honorarni saradnik, ova specijalizacija je *disjunktna*, a ujedno i *totalna*, jer ne može da postoji nastavnik koji ima drugačiji radnopravni status.



Slika 1.9 MOV dijagram nakon identifikacije potklasa ZAPOSLENI i HONORARNI

Stalno zaposleni nastavnik nije samo vrsta nastavnika, nego je ujedno i vrsta zaposlene osobe. Zbog toga potklasa NASTAVNIK\_ZAP treba da specijalizuje, ne samo natklasu NASTAVNIK, nego i natklasu ZAPOSLENI, a entiteti tipa NASTAVNIK\_ZAP imaju svojstva karakteristična za entitete obje natklase. Tako će zaposleni nastavnici imati i nastavničko zvanje (kao nastavnici) i platu (kao stalno zaposleni). Budući da se potklasom NASTAVNIK\_ZAP modeluje samo jedna grupa zaposlenih (stalno zaposleni nastavnici), ova specijalizacija natklase ZAPOSLENI je *parcijalna*.

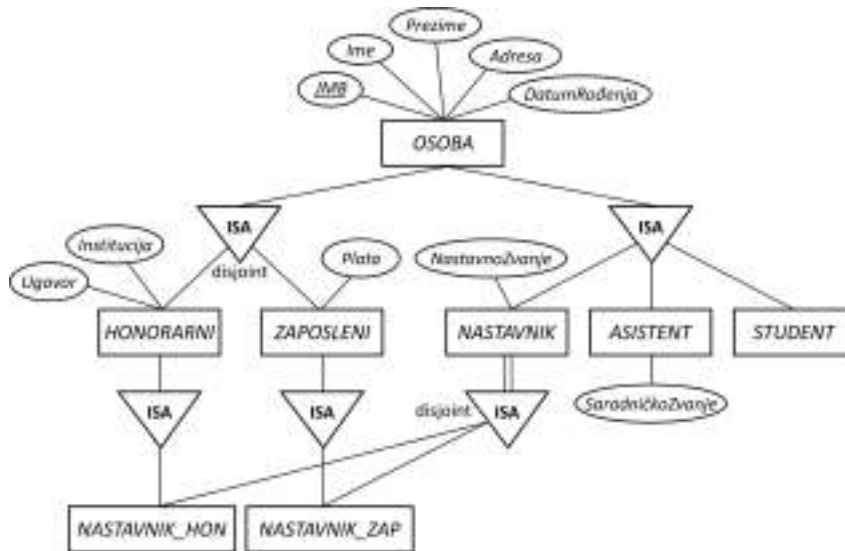
Slično, honorarni nastavnik ujedno je i vrsta honorarnog saradnika, pa potklasa NASTAVNIK\_HON treba da specijalizuje, ne samo natklasu NASTAVNIK, nego i natklasu HONORARNI. Na taj način će honorarni nastavnici imati nastavničko zvanje (kao nastavnici) te ugovor i matičnu instituciju (kao honorarni saradnici). Ova specijalizacija natklase HONORARNI takođe je *parcijalna*, jer potklasa NASTAVNIK\_HON reprezentuje samo honorarne nastavnike. U ovoj fazi projektovanja, MOV dijagram je kao na sl. 1.10 (dio sa sl. 1.6 nije prikazan).

Analogno prethodnom, razlikujemo i dvije vrste asistenata: stalno zaposlene i honorarne, što možemo da modelujemo *totalnom* i *disjunktnom* specijalizacijom natklase ASISTENT potklasama ASISTENT\_ZAP i ASISTENT\_HON. Stalno zaposleni asistent ujedno je i vrsta zaposlene osobe, pa potklasa ASISTENT\_ZAP treba da specijalizuje, ne samo natklasu ASISTENT, nego i natklasu ZAPOSLENI. Tako će zaposleni asistenti imati i asistentsko zvanje (kao asistenti) i platu (kao stalno zaposleni). Slično, honorarni asistent ujedno je i vrsta honorarnog saradnika, pa potklasa ASISTENT\_HON treba da specijalizuje, ne samo natklasu ASISTENT nego i natklasu HONORARNI.

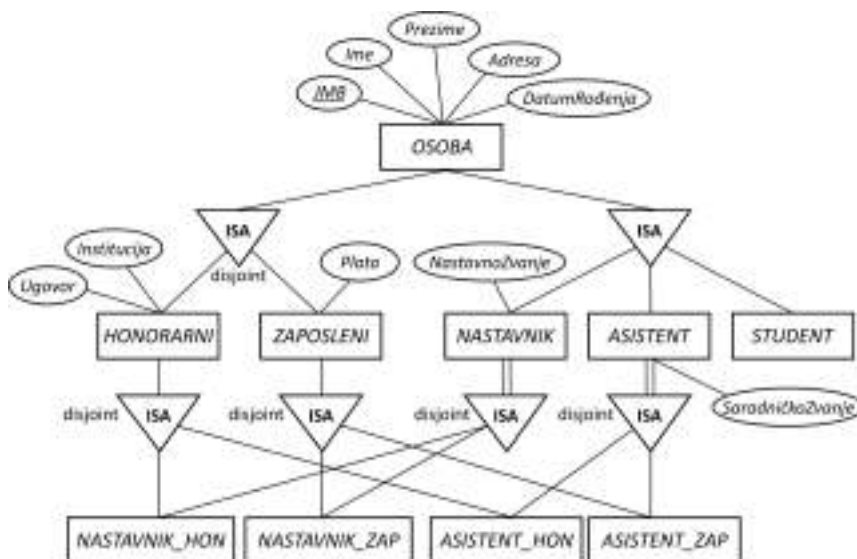
Budući da natklasu ZAPOSLENI specijalizuju dvije potklase (odranije NASTAVNIK\_ZAP i sada ASISTENT\_ZAP), ova specijalizacija treba da se modeluje kao *disjunktna*, jer ne može stalno zaposlena osoba istovremeno da bude i nastavnik i asistent. Analogno, i specijalizacija natklase HONORARNI treba da se modeluje kao *disjunktna*. Objе specijalizacije (i natklase ZAPOSLENI



i natklase *HONORARNI*) su *parcijalne*, jer osim nastavnika i asistenata postoje i druge osobe koje na univerzitetu imaju angažman (npr. administrativno osoblje). U ovoj fazi projektovanja, MOV dijagram je kao na sl. 1.11 (radi preglednosti, dio sa sl. 1.6 nije prikazan).



Slika 1.10 MOV dijagram nakon identifikacije potklasa *NASTAVNIK\_ZAP* i *NASTAVNIK\_HON*



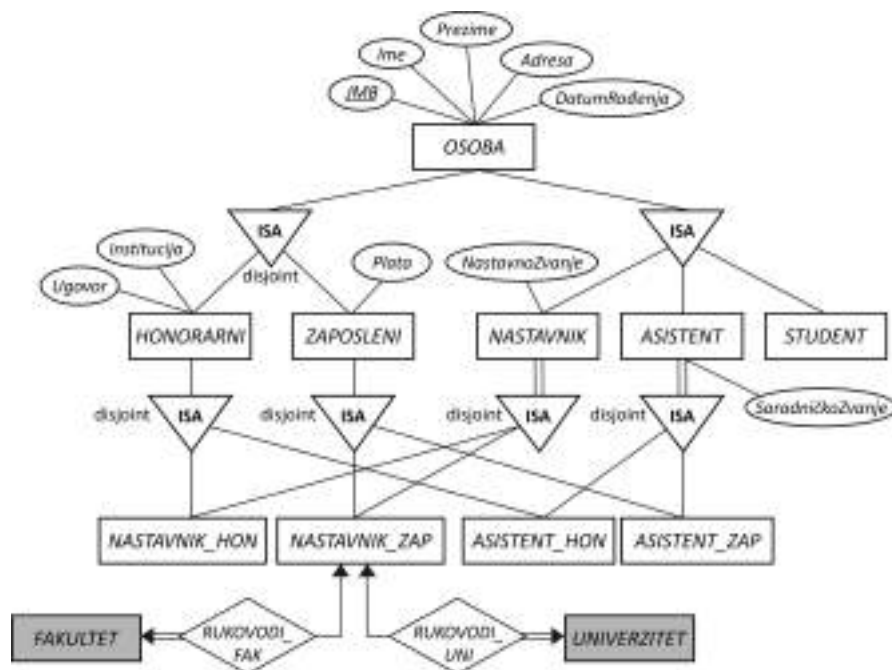
Slika 1.11 MOV dijagram nakon identifikacije potklasa *ASISTENT\_ZAP* i *ASISTENT\_HON*

U odnosu na entitete natklasa, entiteti potklasa imaju neka specifična svojstva (koja se reprezentuju atributima u potklasama) i/ili specifične uloge (koje se reprezentuju tipovima veza u kojima participiraju entiteti potklasa). Sva specifična svojstva entiteta identifikovanih potklasa, koja su specifikovana u datom tekstualnom opisu, reprezentovana su odgovarajućim atributima na MOV dijagramu sa sl. 1.11. Potrebno je još da se identifikuju tipovi veza između entiteta potklasa i drugih entitetskih tipova (ranije identifikovanih na osnovu prvog dijela specifikacije i prikazanih MOV dijagramom na sl. 1.6).

Prema specifikaciji, samo stalno zaposleni nastavnici mogu da rukovode fakultetom. Dakle, postoje veze tipa *RUKOVODI\_FAK* između entiteta tipa *NASTAVNIK\_ZAP* i *FAKULTET*. Svaka veza ovog tipa reprezentuje činjenicu da je jedan stalno zaposleni nastavnik dekan nekog fakulteta. *Svaki* fakultet ima dekana i u posmatranom trenutku samo jedan nastavnik može da bude dekan na jednom fakultetu (tj. postoje nastavnici koji nisu dekani), pa je učešće entiteta tipa *FAKULTET* u vezi *totalno*, a entiteta tipa *NASTAVNIK\_ZAP* *parcijalno*. Još treba da se odredi kardinalnost mapiranja. Budući da se specifikacijom ne zahtijeva vođenje evidencija o svim dekanima u istoriji fakulteta, već samo informacija o trenutnom dekanu, kardinalnost mapiranja je 1:1 (fakultet ima samo jednog dekana, a neki nastavnik može u nekom trenutku da bude dekan samo jednog fakulteta). Ako bi od značaja bili podaci o prethodnim dekanima, tada bi kardinalnost mapiranja bila *više:više*, jer bi neki nastavnik tokom svoje karijere mogao da bude više puta dekan na istom ili na različitim fakultetima, a svaki fakultet u svojoj istoriji (može da) ima više različitih dekana. Tada bi i odgovarajući vezni tip imao opisne attribute (npr. period rukovođenja).

Analogno prethodnom, samo stalno zaposleni nastavnici mogu da rukovode univerzitetom, što modelujemo vezama tipa *RUKOVODI\_UNI* između entiteta tipa *NASTAVNIK\_ZAP* i *UNIVERZITET*, pri čemu je učešće entiteta tipa *UNIVERZITET* *totalno*, a učešće entiteta tipa *NASTAVNIK\_ZAP* *parcijalno*. Sličnim rezonovanjem usvajamo i da je kardinalnost mapiranja 1:1. U ovoj fazi projektovanja, MOV dijagram je kao na sl. 1.12. Radi preglednosti, dio sa sl. 1.6 nije prikazan u cijelosti, već samo tipovi entiteta (osjenčeno) s kojima su povezani entiteti tipa *NASTAVNIK\_ZAP*.

Preostali dio specifikacije opisuje angažman nastavnika i asistenata u nastavnom procesu. Nastavnici mogu da izvode nastavu na *više* predmeta i na *više* studijskih programa. Iako nije eksplicitno navedeno u specifikaciji, možemo da pretpostavimo da na nekim predmetima može da bude angažovano *više* nastavnika. Dakle, činjenice o angažmanu nastavnika na različitim predmetima različitim studijskih programa treba da modelujemo *ternarnim* tipom veze *PREDAJE* između odnosnih entitetskih tipova *NASTAVNIK*, *PREDMET* i *STUD\_PROGRAM*, pri čemu je kardinalnost mapiranja M:M:M. Ako pretpostavimo da mogu da postoje nastavnici koji (još) nemaju zaduženje na nekom predmetu, kao i studijski programi, odnosno predmeti za koje (još) nisu definisana zaduženja, participacija entiteta odnosnih tipova u vezama tipa *PREDAJE* je *parcijalna*.

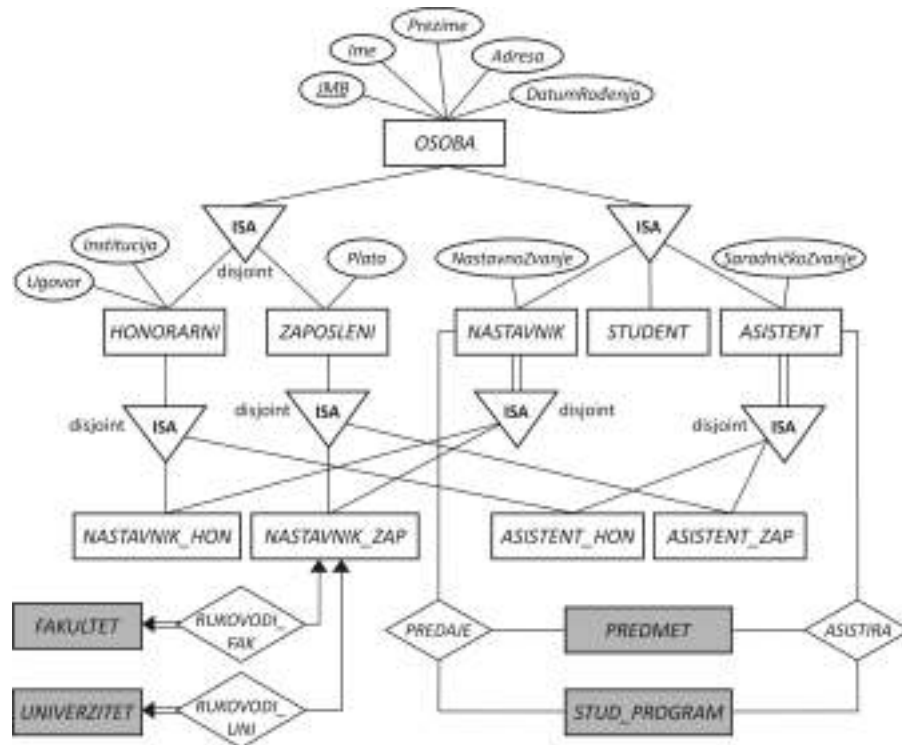


Slika 1.12 MOV dijagram nakon identifikacije veznih tipova *RUKOVODI\_FAK* i *RUKOVODI\_UNI*

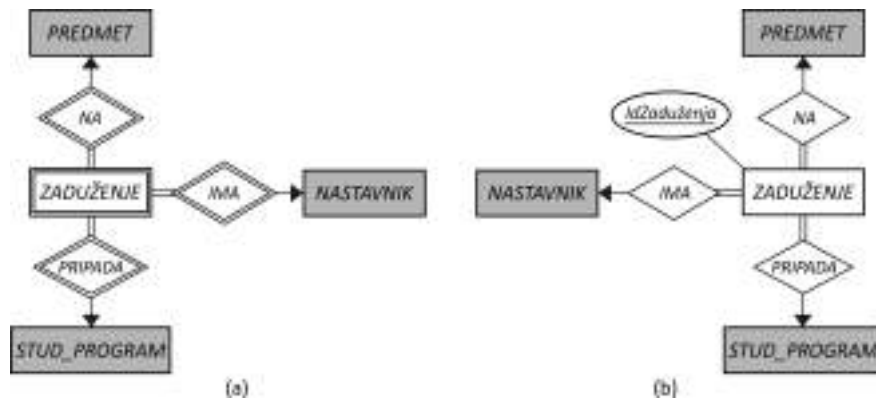
Analognim rezonovanjem dolazimo i do *ternarnog* tipa veze *ASISTIRA* između entitetskih tipova *ASISTENT*, *PREDMET* i *STUD\_PROGRAM*, kojim se reprezentuju činjenice o angažmanu asistenata na različitim predmetima različitih studijskih programa. Kardinalnost mapiranja takođe je M:M:M.

Dodavanjem veznih tipova *PREDAJE* i *ASISTIRA* dobijamo MOV kao na sl. 1.13, koji predstavlja konceptualni model podataka kreiran na osnovu zadate tekstualne specifikacije. Radi preglednosti, dio MOV dijagrama sa sl. 1.6 nije prikazan u cijelosti, već samo tipovi entiteta (osjenčeno) s kojima su povezani entitetski tipovi identifikovani na osnovu drugog dijela tekstualne specifikacije.

Ternarni vezni tip *PREDAJE* (budući da predstavlja *zaduženje* nastavnika na nekom predmetu nekog studijskog programa), alternativno može da se reprezentuje odgovarajućim entitetskim tipom *ZADUŽENJE*, koji je sa entitetskim tipovima *NASTAVNIK*, *STUD\_PROGRAM* i *PREDMET* vezan odgovarajućim binarnim veznim tipovima. Svako *zaduženje* je egzistencijalno zavisno i od konkretnog nastavnika i od konkretnog predmeta i od konkretnog studijskog programa, pa je entitetski tip *ZADUŽENJE* slabi entitetski tip, a odnosni binarni vezni tipovi identifikujući (sl. 1.14a). Dodatno se (ako bi entiteti tipa *ZADUŽENJE* bili u vezama sa entitetima drugih tipova), uvođenjem *surogat* ključa (*IdZaduženja*), slabi entitetski tip *ZADUŽENJE* može prevesti u jaki (jer bi imao primarni ključ), a odnosni binarni vezni tipovi tada bi bili neidentifikujući, kao na sl. 1.14b).



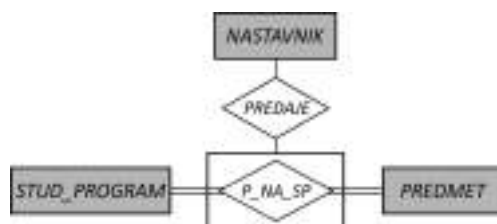
Slika 1.13 MOV dijagram kreiran na osnovu tekstualne specifikacije informacionih potreba hipotetičkog univerzitetskog sistema (2. dio)



Slika 1.14 Alternativne varijante modelovanja zaduzenja nastavnika:  
a) slabim entitetskim tipom, i b) jakim entitetskim tipom

Koncept agregacije omogućava da se veze tretiraju kao entiteti višeg nivoa. U konkretnom slučaju, veze tipa  $P\_NA\_SP$  (ranije identifikovane između entitetskih tipova  $STUD\_PROGRAM$  i  $PREDMET$ ) mogu da se tretiraju kao entiteti višeg nivoa. Svaki *objekat* tipa  $P\_NA\_SP$  reprezentuje činjenicu da je neki predmet

sadržan u nekom studijskom programu. Budući da nastavnici predaju predmete koji pripadaju nekom studijskom programu, vezni tip *PREDAJE* upravo treba da povezuje odnosne entitetske tipove *NASTAVNIK* i *P\_NA\_SP*, kao što je prikazano na sl. 1.15. Kardinalnost mapiranja veznog tipa *PREDAJE* je M:M (jer svaki nastavnik može da predaje više predmeta (na istom ili različitim studijskim programima), a isti predmet može da predaje više nastavnika), pri čemu je participacija objekata oba tipa *parcijalna* (mogu da postoje nastavnici koji nemaju zaduženje te predmeti za koje još niko nije zadužen).



Slika 1.15 Reprezentacija zaduženja nastavnika agregacijom

Ternarni vezni tip *ASISTIRA*, kojim se reprezentuju činjenice o zaduženjima asistenata, takođe može da se tretira na prethodno opisane načine.

### Specifikacija informacionih potreba (3. dio)

*Student je osoba koja je upisana na neki studijski program nekog fakulteta. Prilikom upisa na fakultet studenti dobijaju indeks sa jedinstvenim brojem indeksa za studijski program koji upisuju. Studenti nekog studijskog programa, upisuju odgovarajući semestar i predmete (i tokom semestra prisustvuju nastavi) i polažu ispite koji se organizuju u terminima ispitnih rokova i na određenim lokacijama. Za svaki ispit treba da postoji mogućnost registrovanja broja studenata koji su izašli na ispit i broja studenata koji su položili ispit. Nakon izlaska na ispit, student dobija odgovarajuću ocjenu.*

### Projektovanje konceptualnog modela

Specifikacijom se definiše da je *student* (kao vrsta *osobe*) upisan na neki studijski program. Odnosni entitetski tipovi *STUDENT* i *STUD\_PROGRAM* već su identifikovani i reprezentovani u MOV dijagramu. Još treba da se identifikuje odgovarajući vezni tip *UPISAN\_NA*.

Iako nije eksplicitno specifikovano, treba uzeti u obzir činjenicu da ista osoba (*student*) može da upiše više studijskih programa (najčešće na različitim ciklusima na istom fakultetu). S druge strane, na nekom studijskom programu tipično studira više studenata, pa je kardinalnost mapiranja M:M. Budući da ne može da postoji student koji nije upisan ni na jedan studijski program, učešće entiteta tipa *STUDENT* je *totalno*. Učešće entiteta tipa *STUD\_PROGRAM* je *parcijalno*, jer može da postoji (npr. neki tek osnovan) studijski program na koji nije upisan još nijedan student.

Za svaki upisani studijski program student dobija (novi) indeks, što znači da je indeks (tj. broj indeksa) svojstvo svakog upisa (odnosno veze između studenta i studijskog programa), pa treba da se modeluje odgovarajućim atributom *BrojIndeksa* korespondentnog veznog tipa *UPISAN\_NA*.

Student upisuje semestar na upisanom studijskom programu. Upisani semestar takođe je svojstvo veze između studenta i studijskog programa. Ako pretpostavimo (radi pojednostavljenja) da je dovoljna informacija o trenutnom semestru u koji je student upisan, tada ovo svojstvo možemo da reprezentujemo atributom *Semestar* veznog tipa *UPISAN\_NA*. U suprotnom, upis semestra je entitet sa svojim svojstvima (npr. datum upisa/obnove, itd.) i treba da se modeluje odgovarajućim entitetskim tipom.

Na sl. 1.16 prikazan je relevantni dio MOV dijagrama. Radi preglednosti, prikazani su (osjenčeno) samo entitetski tipovi koji su povezani vezama tipa *UPISAN\_NA*.

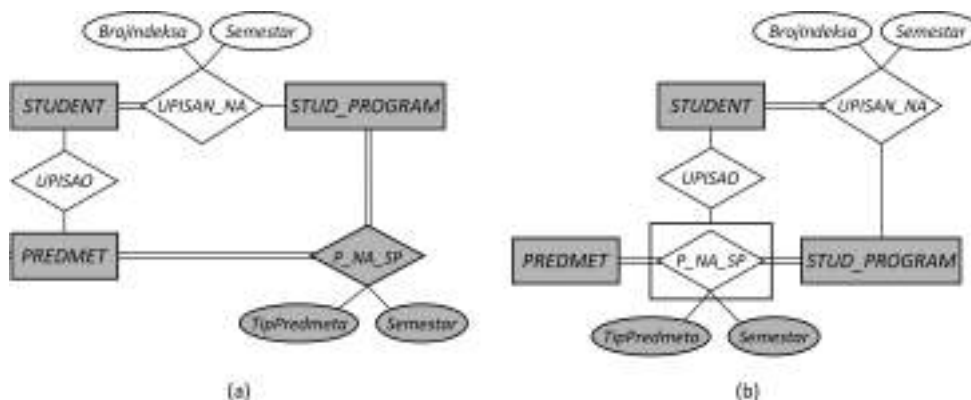


**Slika 1.16** Dio MOV dijagrama nakon identifikacije veznog tipa *UPISAN\_NA*

Prilikom upisa semestra, student bira i upisuje predmete koje će slušati u toku tog semestra. Najjednostavnija reprezentacija činjenice da je neki *student upisao* neki *predmet* jeste veza tipa *UPISAO* između odnosnih entitetskih tipova *STUDENT* i *PREDMET*. Tipično, student u svakom semestru upisuje *više* predmeta, a neki predmet istovremeno sluša *više* studenata, pa je kardinalnost mapiranja M:M. Ako pretpostavimo da student prilikom upisa na neki studijski program, prvo upisuje studijski program, a naknadno upisuje semestar i bira predmete, možemo zaključiti da u nekom periodu mogu da postoje studenti koji (još) nisu upisali nijedan predmet, pa je participacija entiteta tipa *STUDENT* u vezama tipa *UPISAO* *parcijalna*. S druge strane, mogu da postoje izborni predmeti koje nije izabrao nijedan student, pa je participacija entiteta tipa *PREDMET* u vezama tipa *UPISAO* takođe *parcijalna*.

Nakon identifikacije veznog tipa *UPISAO*, relevantni dio MOV dijagrama je kao na sl. 1.17a). Osjenčeno su prikazani koncepti koji su identifikovani u ranijim fazama projektovanja modela.

Usljed veoma pojednostavljenog tretiranja veza studenata i predmeta, kreirani model ima veoma bitan nedostatak, jer ne specifikuje ograničenje da student može da upiše samo predmet sa studijskog programa koji je upisao. Ovaj nedostatak može (djelimično) da se otkloni primjenom koncepta agregacije (slično modelovanju angažmana nastavnika), kao što je prikazano na sl. 1.17b).



**Slika 1.17** Dio MOV dijagrama nakon identifikacije veznog tipa *UPIŠAO* (a), i poboljšani model primjenom agregacije (b)

Studenti polažu *ispite* koji se organizuju u terminima ispitnih rokova i na određenim lokacijama. Dakle, *ispiti* su entiteti od posebnog značaja i treba da se reprezentuju odgovarajućim entitetskim tipom *ISPIT*, čiji atributi *DatumIspita* i *Lokacija* reprezentuju karakteristična svojstva svakog ispita. Budući da se u istom terminu i na istoj lokaciji može održati više ispita, nijedan od atributa pojedinačno niti oba atributa u kombinaciji ne mogu predstavljati primarni ključ. Pošto nema primarni ključ, entitetski tip *ISPIT* je *slabi entitetski tip*.

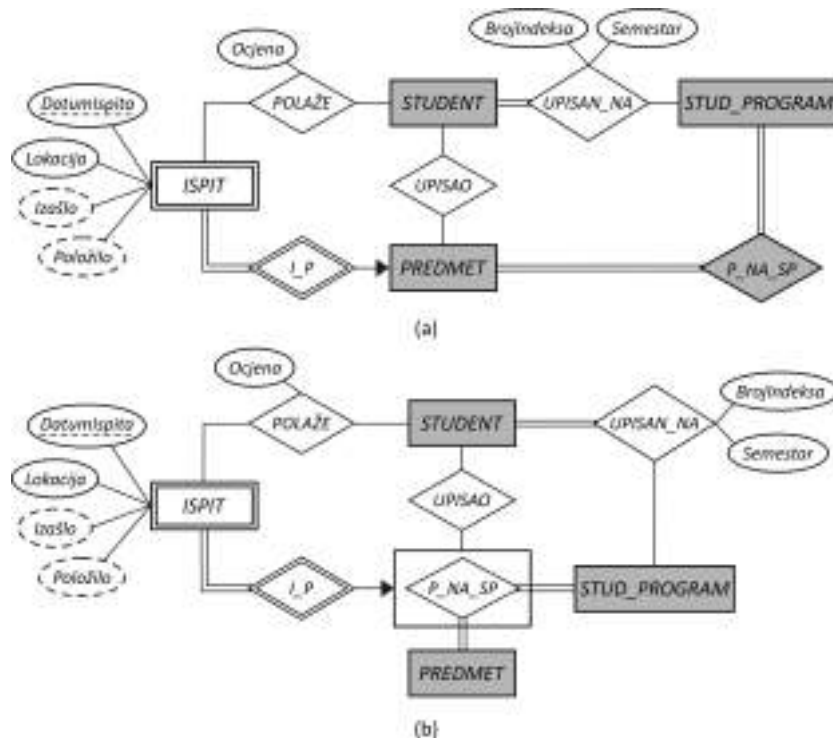
*Ispiti* su egzistencijalno zavisni od *predmeta* (ne može se organizovati ispit iz predmeta koji ne postoji), pa su veze između ovih entiteta *identifikujuće*. Za svaki predmet (može da) se organizuje više ispita, a svaki ispit se odnosi na konkretan predmet, pa je kardinalnost mapiranja 1:više posmatrano sa strane entitetskog tipa *PREDMET*, pri čemu je participacija entiteta tipa *PREDMET* *parcijalna*, a participacija entiteta tipa *ISPIT* *totalna*. Svi ispiti koji se odnose na neki predmet, razlikuju se po terminu održavanja, pa je atribut *DatumIspita* diskriminator entitetskog tipa *ISPIT*.

Student (može da) polaže *više* ispita, a svakom ispitu (može da) pristupa *više* studenata, pa je kardinalnost mapiranja veznog tipa *POLAŽE*, kojim se reprezentuju veze između odnosnih entitetskih tipova *STUDENT* i *ISPIT*, *više:više*, pri čemu su učešća entiteta oba entitetska tipa *parcijalna*. Svojstvo svakog polaganja ispita je *ocjena* (kojom je student ocijenjen) i modeluje se atributom *Ocjena* odnosnog veznog tipa *POLAŽE*.

Prema datoj specifikaciji, za svaki ispit treba da postoji mogućnost registrovanja broja studenata koji su izašli na ispit i broja studenata koji su položili ispit. Ova dva svojstva ispita reprezentujemo atributima *Izašlo* i *Položilo* odnosnog entitetskog tipa *ISPIT*, respektivno. S obzirom na to da se podatak o broju studenata koji su izašli na ispit može izračunati na osnovu broja veza studenata sa ispitom, te da se podatak o broju studenata koji su položili ispit može izračunati analizom ocjena, odnosni atributi *Izašlo* i *Položilo* treba da se modeluju kao *izvedeni* atributi.



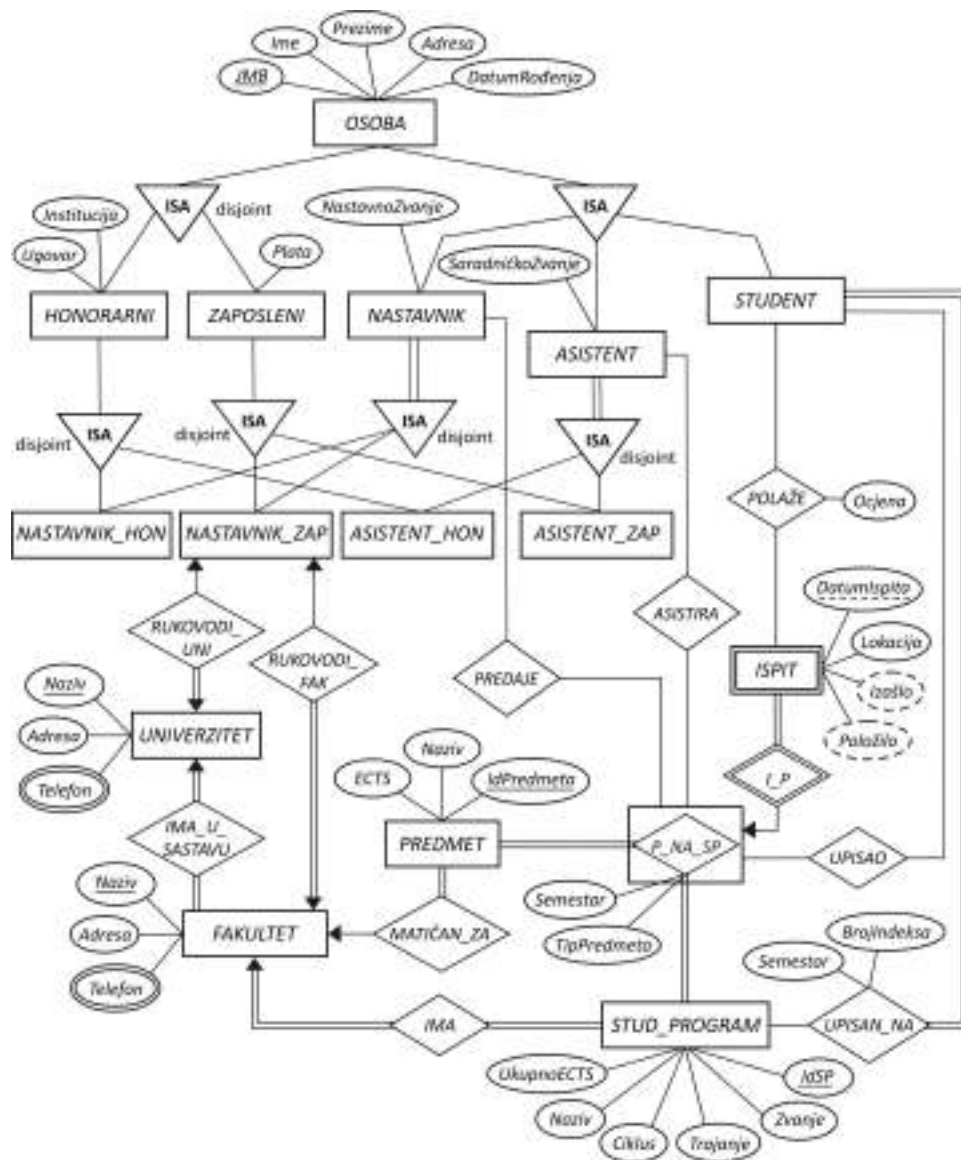
U ovoj fazi projektovanja, relevantan dio MOV dijagrama je kao na sl. 1.18a). Osjenčeno su prikazani ranije identifikovani koncepti. Kreirani model ima nedostatak, jer omogućava da student polaže ispit iz bilo kojeg predmeta na univerzitetu, a ne samo iz predmeta koji pripadaju odgovarajućem studijskom programu ili samo iz predmeta koje je student upisao na odgovarajućem studijskom programu. Ovaj nedostatak djelimično se eliminiše korišćenjem agregacije (sl. 1.18b). Iako poboljšani model sa sl. 1.18b) i dalje omogućava da student polaže ispite iz predmeta koje nije upisao i iz predmeta sa drugih studijskih programa. Međutim, poboljšani model omogućava lakšu manipulaciju relevantnim podacima, jer se lakše dolazi do određenog predmeta na konkretnom studijskom programu.



**Slika 1.18** Dio MOV dijagrama nakon identifikacije entitetskog tipa *ISPIT* (a), i poboljšani model primjenom agregacije (b)

Bez obzira na opisane nedostatke, može se smatrati da kreirani model dovoljno dobro reprezentuje specifikovane informacione potrebe. Kreirani model omogućava određenu fleksibilnost koja je često neophodna u toku eksploatacije baze podataka. U procesu projektovanja treba biti svjestan činjenice da ne mogu sva ograničenja da se uvažavaju i reprezentuju u modelu, odnosno da modelovanje nekih ograničenja može značajno povećati složenost modela, što može da otežava manipulaciju podacima ili degradira performanse sistema u toku eksploatacije. Na kraju, neka ograničenja se mnogo lakše implementiraju na aplikativnom nivou.

Na sl. 1.19 prikazan je cjelokupni MOV dijagram koji reprezentuje konceptualni model podataka univerziteta u skladu sa specificovanim informacionim potrebama. Stvarne informacione potrebe su u određenoj mjeri obimnije, a odgovarajući model kompleksniji (dodatni tipovi entiteta, značajno veći broj atributa, itd.) od modela koji je projektovan u ovom primjeru. I bez obzira na to, prezentovani primjer dovoljno ilustrativno i sistematizovano prikazuje proces projektovanja, ukazujući na najčešće projektne dileme i alternativna rješenja.



Slika 1.19 Konceptualni model podataka univerziteta

---

**Primjer 1.2** Projektovati konceptualni model podataka fakultetske biblioteke primjenom MOV notacije.

---

### **Rješenje.**

#### ***Specifikacija informacionih potreba***

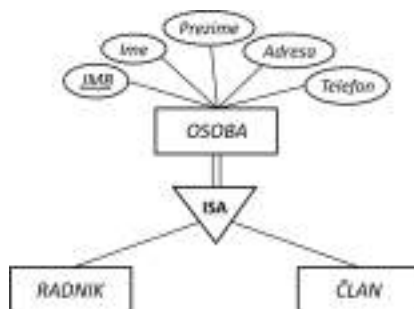
*Predmet projektnog zadatka je fakultetska biblioteka. Radom biblioteke rukovodi bibliotekar. Osim bibliotekara, poslove vezane za članove biblioteke i knjižni fond obavlja nekoliko knjižničara. Članovi biblioteke mogu da budu studenti odgovarajućeg ciklusa studija (I, II, III) nekog od studijskih programa koje organizuje fakultet, te zaposleni na fakultetu (nastavnici, asistenti, administrativno osoblje). Podaci koji se vode za svakog člana su: jmb, prezime, ime, adresa, telefon. Za članove koji su zaposleni na fakultetu još su bitni podaci: broj prostorije u kojoj radi te lokal (broj telefona u prostoriji). Za studente je još bitan broj indeksa. Za svakog člana vodi se evidencija o zaduženjima bibliotečkih jedinica. Studenti imaju ograničenje u pogledu maksimalnog broja jedinica zaduženih u isto vrijeme, dok zaposleni nemaju to ograničenje. Za svakog člana bitan je podatak o trenutnom broju zaduženih bibliotečkih jedinica. Svako zaduženje i razduženje neke jedinice vrši bibliotekar ili knjižničar.*

*Svaka bibliotečka jedinica ima jedinstvenu oznaku (tzv. signatura). Za svaku jedinicu bitan je podatak da li je trenutno raspoloživa (slobodna za posuđivanje). Svaka bibliotečka jedinica pripada ili nekom periodičnom ili nekom neperiodičnom izdanju. Sva periodična izdanja (časopisi) imaju jedinstvenu oznaku (ISSN), naziv, kao i druge podatke o izdanju: godište, broj i godina izdanja. Broj primjeraka svakog broja časopisa varira. Najveći dio neperiodičnih izdanja su knjige. Sve knjige koje pripadaju istom izdanju (obično ima više primjeraka iz istog izdanja) imaju istu jedinstvenu oznaku (ISBN), naziv, godinu izdanja i autore. Časopisi i knjige imaju izdavača. Svakog izdavača karakterišu naziv i sjedište (mjesto i država). Jedan dio neperiodike su radovi pojedinačnih autora, kao što su: diplomski radovi, master radovi, magistarske teze, doktorske disertacije, itd. Ove bibliotečke jedinice takođe imaju naziv, autora i godinu odbrane, ali nemaju ISBN. Većina ovih jedinica je odbranjena na datom univerzitetu, ali postoje i teze koje su odbranjene na drugim univerzitetima.*

#### ***Projektovanje konceptualnog modela***

Analizom prvog dijela tekstualne specifikacije zaključujemo da su za odvijanje poslovnih procesa u fakultetskoj biblioteci bitne dvije grupe osoba: radnici biblioteke (bibliotekar i knjižničari) te članovi biblioteke (nastavnici, asistenti, studenti, administracija). Za radnike biblioteke nisu specifikovana svojstva, ali možemo smatrati da su svojstva koja su specifikovana za članove (jmb, prezime, ime, adresa, telefon), takođe i svojstva radnika biblioteke. U tom slučaju možemo sve osobe (radnici i članovi biblioteke) da reprezentujemo entitetskim tipom OSOBA sa atributima: JMB, Prezime, Ime, Adresa, Telefon.

*Radnici i članovi* mogu da se reprezentuju korespondentnim potklasama *RADNIK* i *ČLAN* superklase *OSOBA*, pri čemu je specijalizacija totalna (u biblioteci nema drugih uloga osim radnika i članova) i preklapajuća (i bibliotekar i knjižničari mogu da budu članovi koji posuđuju knjige). U ovoj fazi projektovanja konceptualnog modela, MOV dijagram je kao na sl. 1.20.



Slika 1.20 Inicijalni MOV dijagram

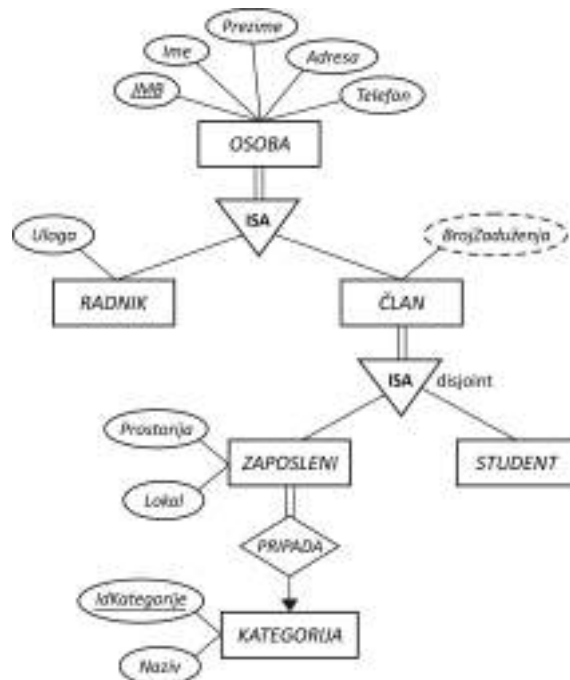
Činjenica da postoje dvije vrste radnika u biblioteci (*bibliotekar* i *knjižničari*) može da se reprezentuje dodatnom specijalizacijom potklase *RADNIK*. Međutim, u specifikaciji informacionih potreba nisu eksplicitno navedene razlike između bibliotekara i knjižničara, niti u svojstvima niti u ulogama u odvijanju poslovnih procesa, pa je za razlikovanje bibliotekara i knjižničara dovoljan jedan atribut (diskriminator<sup>2</sup>) u potklasi *RADNIK* (atribut *Uloga*).

Postoje dvije vrste članova: *studenti* i *zaposleni* na fakultetu (profesori, asistenti, administracija). Budući da se studenti i zaposleni razlikuju u svojstvima, ove dvije vrste članova treba da se reprezentuju korespondentnim potklasama *STUDENT* i *ZAPOSLENI* natklase *ČLAN*. Specijalizacija superklase *ČLAN* je totalna (svaki član je ili student ili osoba zaposlena na fakultetu). Iako su asistenti uglavnom i studenti na višim ciklusima studija, ipak ih treba tretirati kao zaposlene, a ne kao studente (u kontekstu prava i obaveza u fakultetskoj biblioteci), pa je specijalizacija disjunktna.

Bitno svojstvo svakog člana jeste *broj trenutno zaduženih bibliotečkih jedinica* i treba da se reprezentuje odgovarajućim atributom *BrojZaduženja* u natklasi *ČLAN*. S obzirom na to da će se broj zaduženja moći odrediti i na osnovu broja veza tipa *POSUĐUJE* (između entiteta tipa *ČLAN* i *BIB\_JEDINICA*), *BrojZaduženja* je izvedeni atribut. Dodatna svojstva svakog člana iz reda zaposlenih (pored ostalih svojstava karakterističnih za svakog člana) su *broj prostorije* i *broj lokala*, pa treba da se reprezentuju korespondentnim atributima *Prostorija* i *Lokal* potklase *ZAPOSLENI*.

<sup>2</sup> Treba uočiti da se termin *diskriminator* ovdje ne koristi u smislu diskriminatora slabog entitetskog tipa, već u smislu atributa čija vrijednost omogućava razlikovanje bibliotekara i knjižničara.

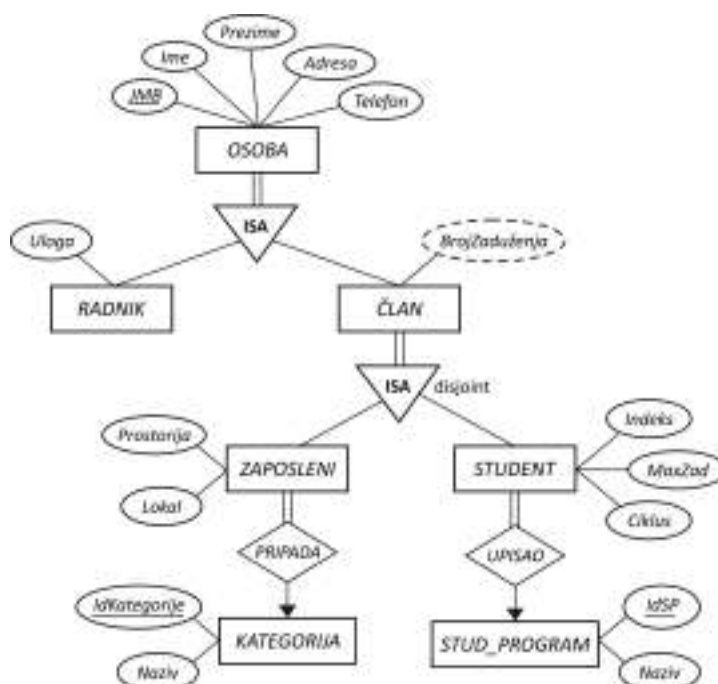
Među članovima iz reda zaposlenih nema dodatnih razlika u svojstvima ili ulogama u biblioteci, pa nema potrebe za dodatnom specijalizacijom potklase *ZAPOSLANI*. Da bi se međusobno razlikovale podgrupe zaposlenih (nastavnici, asistenti, administracija) može da se uvede dodatni atribut (npr. *Kategorija*) kao diskriminator. Alternativno može da se uvede novi entitetski tip *KATEGORIJA* s odgovarajućim atributima (*IdKategorije*, *Naziv*, itd.), koji predstavlja tzv. *šifarnik* ili *registar* kategorija. Uvođenje šifarnika omogućava lakšu manipulaciju podacima u toku eksploatacije baze podataka (konzistentnost podataka, nema potrebe za aplikativnim izmjenama, itd.). Kardinalnost mapiranja veznog tipa *PRIPADA* između entitetskih tipova *ZAPOSLANI* i *KATEGORIJA* jeste *više:1*, pri čemu je participacija na strani *ZAPOSLANI* totalna, a na strani *KATEGORIJA* parcijalna (svaki zaposleni pripada nekoj kategoriji, istoj kategoriji istovremeno može da pripada više zaposlenih, a može da postoji kategorija kojoj niko ne pripada). U ovoj fazi projektovanja konceptualnog modela, MOV dijagram je kao na sl. 1.21.



**Slika 1.21** Inicijalni MOV dijagram nakon identifikacije članova iz reda zaposlenih

Dodatna svojstva svakog člana iz reda studenata (pored ostalih svojstava karakterističnih za svakog člana) su: *broj indeksa*, *ciklus studija*, *studijski program* na koji je student upisan te *maksimalan dozvoljeni broj zaduženih bibliotečkih jedinica*. Broj indeksa, ciklus studija i maksimalni broj zaduženja mogu da se reprezentuju odgovarajućim atributima (*Indeks*, *Ciklus*, *MaxZad*)

potklase *STUDENT*. I studijski program može da se reprezentuje odgovarajućim atributom, ali je (iz istih razloga kao i kod modelovanja kategorija zaposlenih) bolje uvesti dodatni entitetski tip *STUD\_PROGRAM* (sa atributima: *IdSP*, *Naziv*, itd.) kao šifarnik studijskih programa. Kardinalnost mapiranja veznog tipa *UPISAO* između entitetskih tipova *STUDENT* i *STUD\_PROGRAM* jeste *više:1* (posmatrano sa strane entitetskog tipa *STUDENT*), pri čemu je participacija na strani *STUDENT* totalna, a na strani *STUD\_PROGRAM* parcijalna (svaki student je upisan na neki studijski program, na istom studijskom programu može da studira više studenata, a može da postoji studijski program na koji još nisu upisani studenti). Nakon identifikacije svih osoba uključenih u poslovne procese biblioteke, odnosno hijerarhije odgovarajućih entitetskih tipova, MOV dijagram je kao na sl. 1.22.



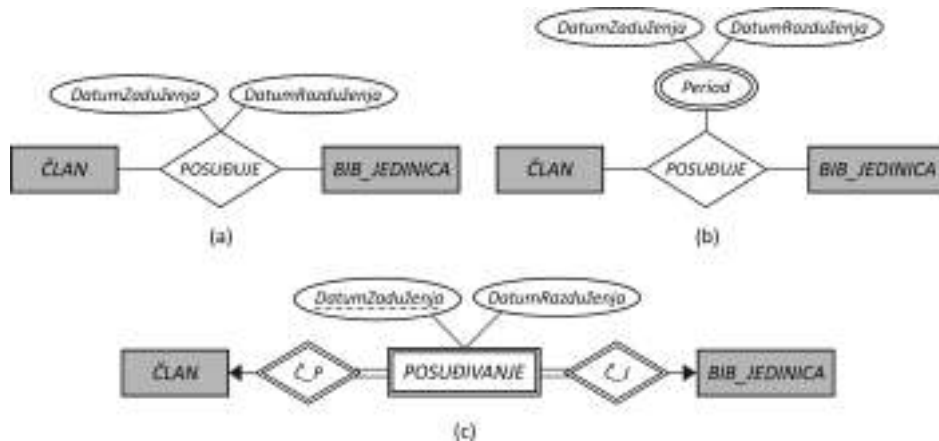
Slika 1.22 MOV dijagram nakon identifikacije svih osoba uključenih u poslovne procese u biblioteci

Za svakog člana vodi se evidencija o zaduženjima *bibliotečkih jedinica*. Svaki član (tokom svog članstva u biblioteci) može da posudi više jedinica, a istu jedinicu (tokom vremena) može da posudi više članova, pa je kardinalnost mapiranja veznog tipa *POSUĐUJE* (između odnosnih entitetskih tipova *ČLAN* i *BIB\_JEDINICA*) *više:više*. Participacija entiteta oba entitetska tipa u vezama tipa *POSUĐUJE* je parcijalna, jer mogu da postoje jedinice koje (još) nisu posuđivane, kao i članovi koji (još) nisu posudili nijednu jedinicu.

Bitna svojstva svakog *posuđivanja* jesu datum posuđivanja bibliotečke jedinice (*datum zaduženja*) i datum vraćanja (*datum razduženja*) jedinice, pa treba da se modeluju odgovarajućim atributima (*DatumZaduženja* i *DatumRazduženja*) odnosno veznog tipa *POSUĐUJE*. Relevantni dio MOV dijagrama prikazan je na sl. 1.23a). Alternativno se period zaduženja može modelovati složenim atributom *Period*, čije su komponente *DatumZaduženja* i *DatumRazduženja*.

Ovakav model omogućava da neki član posudi više različitih bibliotečkih jedinica, ali ne omogućava da više puta posudi istu jedinicu. S obzirom na to da je bitna istorija svih zaduženja svakog člana (zaduženja se ne brišu), te da se često dešava da isti član više puta posudi istu jedinicu (npr. student mora da vrati knjigu nakon mjesec dana, ali kad je vrati može ponovo da je zaduži, ili npr. profesor tokom karijere više puta posudi isti časopis ili knjigu), model treba da omogući svakom članu višestruko posuđivanje iste jedinice. Pošto svako posuđivanje može da se reprezentuje složenim atributom *Period*, višestruka posuđivanja (iste jedinice od strane istog člana) mogu da se reprezentuju višeznačnim složenim atributom *Period* (sa komponentama *DatumZaduženja* i *DatumRazduženja*), kao što je prikazano na sl. 1.23b)

Alternativno, svako *posuđivanje* može da se tretira i kao entitet od posebnog interesa u datom poslovnim sistemu, pa se odnosni vezni tip *POSUĐUJE* može zamijeniti odgovarajućim entitetskim tipom i parom binarnih veznih tipova. Svako *posuđivanje* je egzistencijalno zavisno od konkretnog člana i konkretne bibliotečke jedinice, pa je korespondentni entitetski tip *POSUĐIVANJE* slabiji entitetski tip, a vezni tipovi identifikujućii. Atributi *DatumZaduženja* i *DatumRazduženja* entitetskog tipa *POSUĐIVANJE* su prosti, jer svaki entitet tipa *POSUĐIVANJE* reprezentuje jedno posuđivanje sa jednim datumom zaduženja i jednim datumom razduženja, kao što je prikazano na sl. 1.23c). Budući da se višestruka posuđivanja iste jedinice od strane istog člana razlikuju u datumu zaduženja, atribut *DatumZaduženja* predstavlja diskriminator odnosno slabog entitetskog tipa.



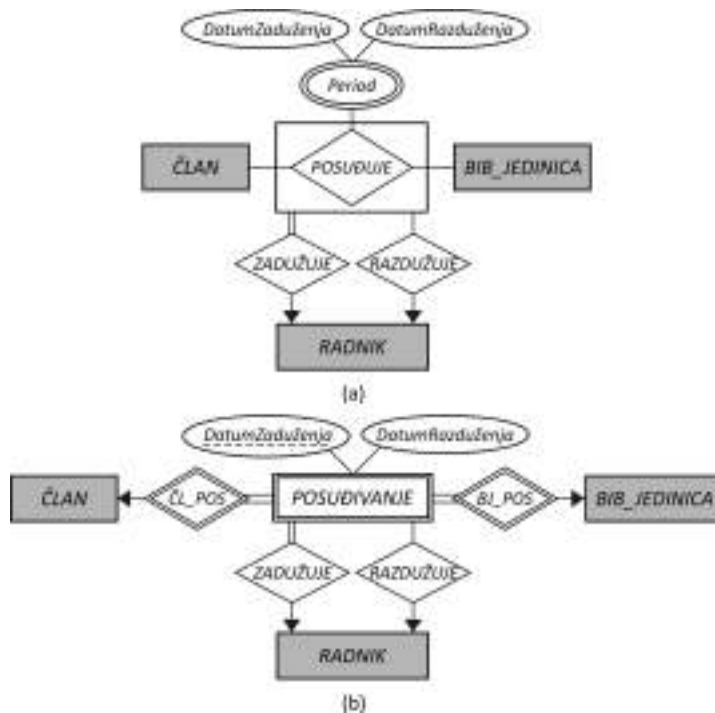
Slika 1.23 Dio MOV dijagrama koji reprezentuje posuđivanje bibliotečkih jedinica



Svako zaduženje i razduženje neke jedinice vrši bibliotekar ili knjižničar. To znači da je za svako *posuđivanje* neke jedinice vezan jedan radnik biblioteke koji evidentira zaduženje te jedan radnik biblioteke koji evidentira razduženje. Iako postoji mogućnost da to bude isti radnik, model treba da omogući da dva različita radnika evidentiraju zaduženje i razduženje (npr. zbog rada u smjenama). Činjenica da jedan radnik evidentira zaduženje modeluje se veznim tipom *ZADUŽUJE*, dok se činjenica da drugi radnik evidentira razduženje modeluje dodatnim veznim tipom *RAZDUŽUJE* između entitetskog tipa *RADNIK* i entitetskog tipa koji reprezentuje *posuđivanje*.

S obzirom na alternative sa sl. 1.23, postoje dvije mogućnosti za modelovanje *posuđivanja*, kao što je prikazano na sl. 1.24. Model na sl. 1.24a) bazira se na posmatranju veza tipa *POSUĐUJE* (u modelu sa sl. 1.23b) kao entiteta višeg tipa, tj. na primjeni koncepta agregacije. Model na sl. 1.24b) temelji se na modelu sa sl. 1.23c), u kojem se *POSUĐIVANJE* posmatra kao slabi entitetski tip.

Kardinalnosti mapiranja oba vezna tipa (*ZADUŽUJE* i *RAZDUŽUJE*) su 1:više posmatrano sa strane entitetskog tipa *RADNIK*, pri čemu je participacija entiteta tipa *RADNIK* parcijalna u oba slučaja. Participacija entiteta tipa *POSUĐIVANJE* (odnosno *POSUĐUJE*) je totalna u vezama tipa *ZADUŽUJE* (svako posuđivanje nastaje evidentiranjem zaduženja), a parcijalna u vezama tipa *RAZDUŽUJE* (sve dok član ne vrati posuđenu jedinicu, odgovarajuće zaduženje nije razduženo).



Slika 1.24 Alternative za modelovanje posuđivanja

Svaka biblioteka jedinica ima jedinstvenu oznaku (*signatura*) i treba da se reprezentuje odgovarajućim atributom (*SIGN*), koji predstavlja primarni ključ odnosnog entitetskog tipa *BIB\_JEDINICA*. Bitno svojstvo svake bibliotečke jedinice jeste i njen trenutni *status*, tj. podatak da li je trenutno zadužena ili je raspoloživa. Iako se podatak o raspoloživosti svake jedinice može odrediti na osnovu evidencije zaduženja (jedinica je slobodna ako ne postoji nijedno posuđivanje koje nije razduženo, ipak je poželjno da se ovo svojstvo (s obzirom na značaj podatka) reprezentuje izvedenim atributom *Status*.

Prema specifikaciji postoje dvije osnovne kategorije bibliotečkih jedinica: periodična izdanja (periodika) i neprodična izdanja (neperiodika). Periodiku čine časopisi, a neperiodiku čine knjige i pojedinačni autorski radovi/teze (diplomski, master, magistarski, doktorski). Dakle, svaka bibliotečka jedinica je ili primjerak nekog *časopisa* ili neka *knjiga* ili neka *teza/rad*, što može da se reprezentuje totalnom i disjunktnom specijalizacijom entitetskog tipa *BIB\_JEDINICA* korespondentnim potklasama *BJ\_ČASOPIS*, *BJ\_KNJIGA* i *BJ\_TEZA*. Dodatno još treba uvesti diskriminator *Vrsta* u superklasi *BIB\_JEDINICA* za razlikovanje bibliotečkih jedinica<sup>3</sup>. U ovoj fazi projektovanja konceptualnog modela, relevantni dio MOV dijagrama je kao na sl. 1.25.



Slika 1.25 Dio MOV dijagrama koji reprezentuje hijerarhiju bibliotečkih jedinica

Svaki entitet potklase *BJ\_KNJIGA* reprezentuje konkretnu knjigu, odnosno primjerak neke knjige. U biblioteci tipično postoji više primjeraka iste knjige, odnosno istog izdanja neke knjige. Budući da svako izdanje ima svojstva, treba da se reprezentuje odgovarajućim entitetskim tipom *KNJIGA*. Kardinalnost mapiranja veznog tipa *K\_PRIPADA* između entitetskih tipova *BJ\_KNJIGA* i *KNJIGA* je više:1 (posmatrano sa strane entitetskog tipa *BJ\_KNJIGA*). Participacija entiteta tipa *BJ\_KNJIGA* je totalna (svaka knjiga pripada nekom izdanju), a sa strane *KNJIGA* parcijalna (mogu da postoje podaci o nekom izdanju, iako u biblioteci ne postoji nijedan primjerak tog izdanja).

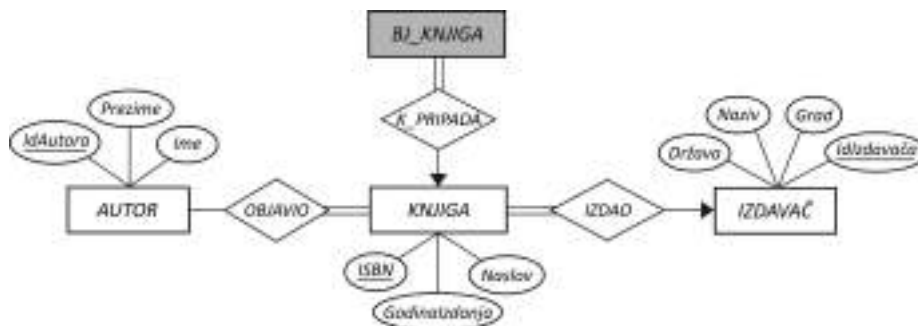
<sup>3</sup> Strogo posmatrano, diskriminator nije neophodan. Međutim, iskustvo ukazuje na to da je tokom eksploatacije baze podataka olakšano pronalaženje konkretnog objekta neke potklase (konkretna biblioteka jedinica) ako je u natklasi uveden diskriminator, tj. atribut na osnovu kojeg se lako određuje potklasa kojoj dati objekat pripada.

Svojstva svake *knjige* su: *ISBN* (jedinствени identifikator), *naslov*, *autori*, *godina izdanja* i *izdavač*. Svojstva *ISBN*, *naziv* i *godina izdanja* treba da se reprezentuju korespondentnim atributima *ISBN*, *Naslov* i *GodinaIzdanja* odnosno entitetskog tipa *KNJIGA*.

Budući da svakog *izdavača* karakterišu *naziv* i *sjedište* (*grad*, *država*), odnosno svojstvo *izdanja* ne treba da se reprezentuje atributom, nego *izdavač* treba da se reprezentuje odgovarajućim entitetskim tipom *IZDAVAČ* sa atributima: *IdIzdavača*, *Naziv*, *Grad*, *Država*, gdje je primarni ključ *IdIzdavača* surogat ključ. Kardinalnost mapiranja veznog tipa *IZDAO* između entitetskih tipova *IZDAVAČ* i *KNJIGA* je 1:više (posmatrano sa strane entitetskog tipa *IZDAVAČ*). Participacija entiteta tipa *KNJIGA* je totalna (svaka knjiga ima izdavača), a sa strane *IZDAVAČ* parcijalna (mogu da postoje podaci o izdavaču čijih izdanja nema u fakultetskoj biblioteci). Atribut *Država* entitetskog tipa *IZDAVAČ*, alternativno može da se reprezentuje entitetskim tipom *DRŽAVA* (šifarnik država) sa atributima: *IdDržave* i *Naziv*.

Svojstvo *autori* može da se reprezentuje višeznačnim atributom *Autor* odnosno entitetskog tipa *KNJIGA*. Budući da isti autor može da objavi više knjiga (i teza), *autori* treba da se reprezentuju zasebnim entitetskim tipom *AUTOR*. Kardinalnost mapiranja veznog tipa *OBJAVIO* između entitetskih tipova *AUTOR* i *KNJIGA* je *više:više*. Participacija entiteta tipa *KNJIGA* je totalna (svaka knjiga ima bar jednog autora), a sa strane *AUTOR* parcijalna (možemo pretpostaviti da postoje autori koji imaju objavljene samo teze).

Iako su autori osobe, entitetski tip *AUTOR* ipak ne treba tretirati kao potklasu *OSOBE*. Naime, svaku *osobu* u modelu karakterišu, između ostalog, jedinstveni matični broj i datum rođenja. Pošto za neke inostrane autore *JMB* ili ne postoji ili nije poznat (kao što nije poznat ni datum rođenja), entitetski tip *AUTOR* ne može da se modeluje kao potklasa *OSOBE*. U ovoj fazi projektovanja konceptualnog modela, relevantni dio MOV dijagrama prikazan je na sl. 1.26.



Slika 1.26 Dio MOV dijagrama koji se odnosi na knjige

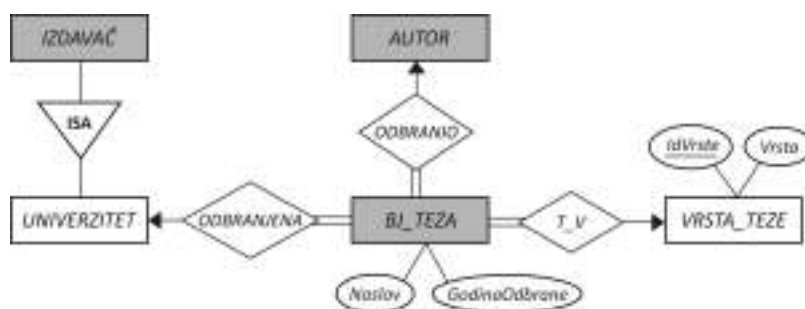
Potklasa *BJ\_TEZA* reprezentuje pojedinačne autorske radove/teze odbranjene na datom fakultetu ili na nekom drugom fakultetu (univerzitetu). U biblioteci

(tipično) postoji samo po jedan primjerak svake teze. Svojstva svake teze su: *naslov*, *autor*, *godina odbrane*. Svojstva *naslov* i *godina odbrane* treba da se reprezentuju odgovarajućim atributima *Naslov* i *GodinaOdbrane* korespondentne potklase *BJ\_TEZA*. Svojstvo *autor* može, takođe, da se reprezentuje odgovarajućim atributom (*Autor*) potklase, ali i posebnim entitetskim tipom *AUTOR*. Budući da ista osoba može da odbrani više radova/teza (npr. diplomski rad, magistarsku tezu i doktorsku disertaciju) te da objavi jednu ili više knjiga, svojstvo *autor* treba da se reprezentuje ranije identifikovanim entitetskim tipom *AUTOR*. Budući da svaka teza ima jednog autora i da neka osoba može da odbrani više teza, kardinalnost mapiranja odnosno veznog tipa *ODBRANIO* između entitetskih tipova *AUTOR* i *BJ\_TEZA* je 1:više (posmatrano sa strane entitetskog tipa *AUTOR*). Participacija entiteta tipa *AUTOR* je parcijalna (samo autori koji su odbranili tezu učestvuju u vezama tipa *ODBRANIO*, a ne i autori koji su samo objavljivali knjige), a entiteta tipa *BJ\_TEZA* je totalna (svaka teza ima autora).

Činjenica da postoji više podvrsta teze (*diplomski rad*, *master rad*, *magistarska teza*, *doktorska disertacija*, itd.) može da se reprezentuje dodatnom specijalizacijom potklase *BJ\_TEZA*. Međutim, u specifikaciji nisu eksplicitno navedene razlike između ovih podvrsta, pa je za njihovo razlikovanje dovoljan atribut *Vrsta* u potklasi *BJ\_TEZA*. Umjesto diskriminatora *Vrsta*, bolje je uvesti dodatni entitetski tip *VRSTA\_TEZE* (sa atributima: *IdVrste*, *Vrsta*, itd.) kao šifarnik vrsta teza (iz istih razloga kao i ranije uvedeni šifarnici). Kardinalnost mapiranja veznog tipa *T\_V* između entitetskih tipova *BJ\_TEZA* i *VRSTA\_TEZE* jeste više:1 (posmatrano sa strane entitetskog tipa *BJ\_TEZA*), pri čemu je participacija entiteta na strani *BJ\_TEZA* totalna (svaka odbranjena teza pripada nekoj vrsti), a na strani *VRSTA\_TEZE* parcijalna (u biblioteci može da postoji više teza iste vrste, ali može da postoji i vrsta za koju još ne postoji nijedna teza).

Svaka teza *odbranjena* je na nekom univerzitetu. I ovo svojstvo može da se reprezentuje odgovarajućim atributom potklase *BJ\_TEZA*. Budući da univerzitet karakterišu *naziv* i *sjedište* (*grad*, *država*), odnosno atribut morao bi da bude složeni atribut. S obzirom na to da će se neki univerziteti pojavljivati i kao *izdavači* knjiga i/ili časopisa, činjenica da je neka teza odbranjena na nekom univerzitetu može da se reprezentuje vezom tipa *ODBRANJENA* između entiteta tipa *BJ\_TEZA* i *UNIVERZITET* (koji je potklasa natklase *IZDAVAČ*, jer je univerzitet specijalna vrsta izdavača), pri čemu je kardinalnost mapiranja više:1 (posmatrano sa strane entitetskog tipa *BJ\_TEZA*), pri čemu je participacija entiteta na strani *BJ\_TEZA* totalna, a na strani *UNIVERZITET* parcijalna (svaka teza je odbranjena na nekom univerzitetu, a mogu da postoje univerziteti na kojima još nije odbranjena nijedna teza). Relevantni dio MOV dijagrama prikazan je na sl. 1.27.

Svaki entitet potklase *BJ\_ČASOPIS* reprezentuje konkretan primjerak nekog broja nekog časopisa. Svaki časopis ima svojstva: *ISSN* (jedinствена oznaka) i *naslov*, pa treba da se reprezentuje odgovarajućim entitetskim tipom *ČASOPIS* sa atributima *ISSN* i *Naslov*.



Slika 1.27 Dio MOV dijagrama koji se odnosi na teze

Budući da časopisi imaju periodična izdanja (*brojevi časopisa*), sa svojstvima: *godište*, *broj* i *godina izdanja*, izdanja časopisa treba da se reprezentuju zasebnim entitetskim tipom *ČAS\_BROJ* sa atributima: *Godište*, *Broj*, i *GodinaIzdanja*. Svaki broj odnosi se na neki časopis i od njega egzistencijalno zavisi. Pored toga, nijedan od atributa (*Godište*, *Broj*, *GodinaIzdanja*) pojedinačno, niti neka od kombinacija nije dovoljna za međusobno razlikovanje brojeva (koji pripadaju različitim časopisima), pa je odnosni entitetski tip *ČAS\_BROJ* slabi entitetski tip (atributi *Godište* i *Broj* predstavljaju diskriminator), a vezni tip *IMA* između entitetskih tipova *ČASOPIS* i *ČAS\_BROJ* identifikujući vezni tip. Kardinalnost mapiranja veznog tipa *IMA* je 1:više (posmatrano sa strane entitetskog tipa *ČASOPIS*). Participacija entiteta tipa *ČAS\_BROJ* je totalna (svaki broj odnosi se na neki časopis), a sa strane *ČASOPIS* parcijalna (npr. časopis je registrovan, ali još ne postoji nijedno izdanje tog časopisa).

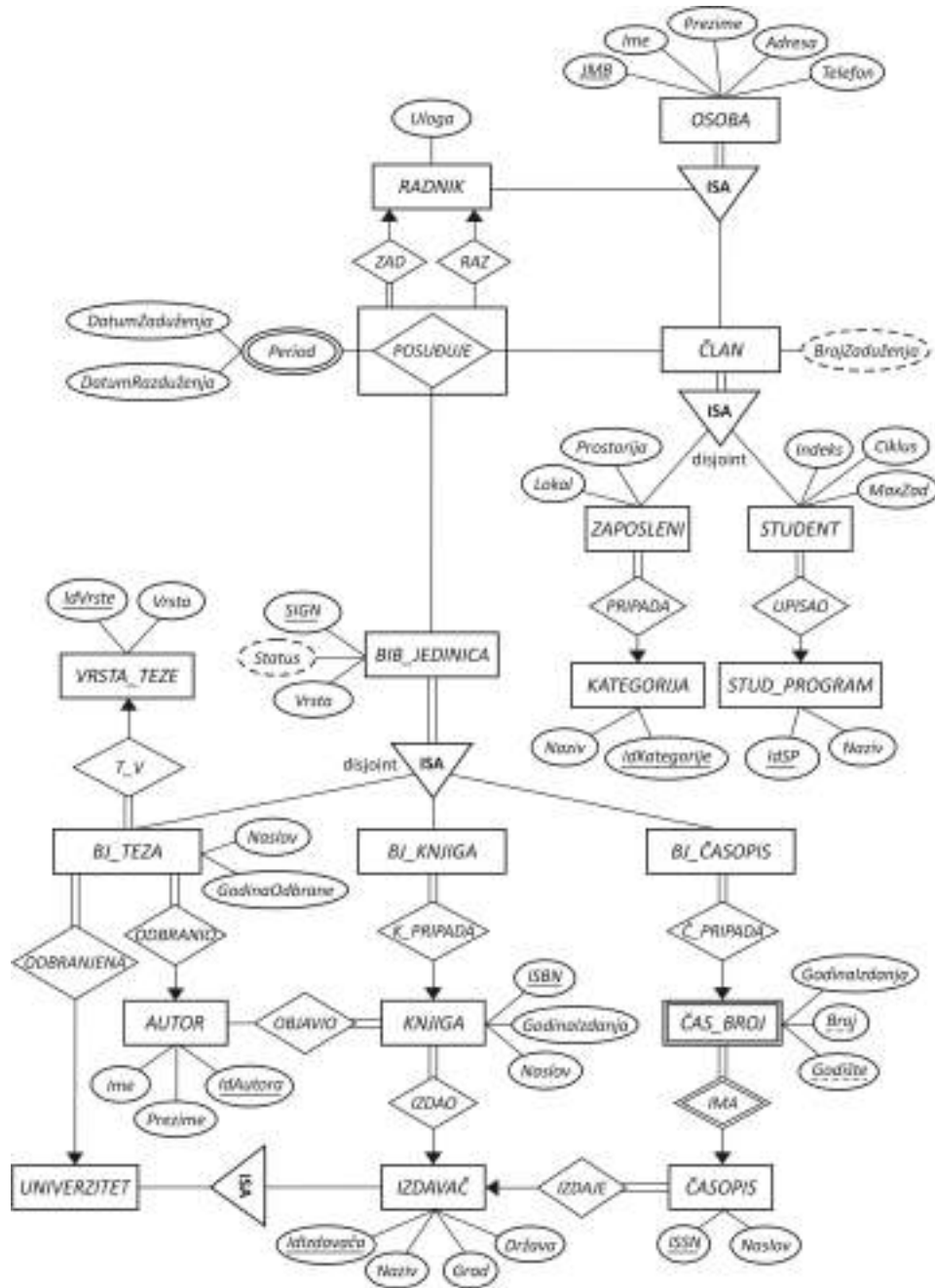
Prema specifikaciji, biblioteka može da raspolaže različitim brojem primjeraka svakog broja nekog časopisa, pa između entiteta tipa *ČAS\_BROJ* i *BJ\_ČASOPIS* postoje veze tipa *Č\_PRIPADA*, čija je kardinalnost mapiranja 1:više (posmatrano sa strane entitetskog tipa *ČAS\_BROJ*). Participacija entiteta tipa *BJ\_ČASOPIS* je totalna (svaka bibliotečka jedinica tipa časopis odnosi se na neki broj nekog časopisa), a sa strane *ČAS\_BROJ* parcijalna (u biblioteci ne postoji nijedan primjerak nekog broja nekog časopisa).

I časopisi imaju izdavača, pa između entiteta tipa *IZDAVAČ* i *ČASOPIS* postoje veze tipa *IZDAJE*, čija je kardinalnost mapiranja 1:više (posmatrano sa strane entitetskog tipa *IZDAVAČ*). Participacija entiteta tipa *ČASOPIS* je totalna (svaka časopis izdaje neki izdavač), a sa strane *IZDAVAČ* parcijalna (među izdavačima postoje i oni izdavači koji ne izdaju časopise). Relevantni dio MOV dijagrama, koji se odnosi na časopise, prikazan je na sl. 1.28.



Slika 1.28 Dio MOV dijagrama koji se odnosi na časopise

Na sl. 1.29 prikazan je cjelokupni MOV dijagram koji reprezentuje konceptualni model podataka fakultetske biblioteke u skladu sa specificovanim informacionim potrebama.



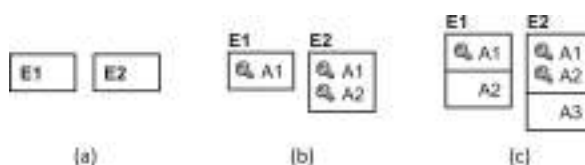
Slika 1.29 Konceptualni model podataka fakultetske biblioteke

### 1.3 IE notacija

**IE** (*Information Engineering*) je često korištena notacija za konceptualno modelovanje u softverskim alatima za projektovanje baza podataka. U ovom odjeljku prikazana je IE zasnovana notacija koja se koristi u ERwin alatu.

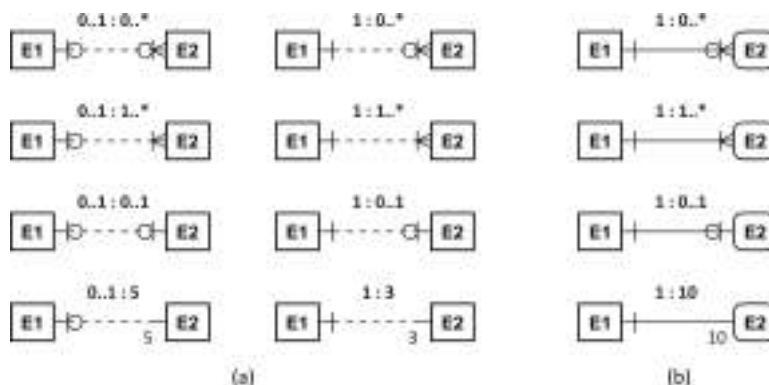
IE notacija ima dosta sličnosti, ali i razlika u odnosu na originalnu Chenovu, kao i MOV notaciju koja je prezentovana u prethodnom odjeljku. Iako IE notacija raspoložuje većinom semantički ekvivalentnih koncepata, ipak postoje neki ER/EER koncepti koji nisu podržani u IE notaciji, npr. *n-arni* tip veze, unija, agregacija, atributi veznog tipa, višeznačni atributi itd.

Entitetski tipovi u IE zasnovanom ERwin konceptualnom modelu mogu da se prikažu na različitim nivoima detaljnosti: bez atributa (sl. 1.30a), samo primarni ključevi (sl. 1.30b), svi atributi (sl. 1.30c). Atributi koji čine primarni ključ (atributi sa simbolom ključa) prikazuju se u gornjoj sekciji pravougaonika, odvojeno od ostalih atributa.



Slika 1.30 Nivoi detaljnosti u prikazu entitetskih tipova

Neidentifikujuće veze prikazuju se isprekidanim linijama (sl. 1.31a), a identifikujuće veze punim linijama (sl. 1.31b). Za specifikaciju kardinalnosti mapiranja koristi se notacija *ptičije stopalo*, tj. kardinalnost *više* prikazuje se u obliku *višestrukog završetka*, a kardinalnost *jedan* u obliku *jednostrukog završetka* linije koja reprezentuje vezni tip pored entitetskog tipa čija se kardinalnost specifikuje.

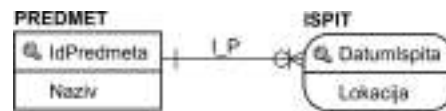


Slika 1.31 Neidentifikujući (a) i identifikujući (b) vezni tipovi



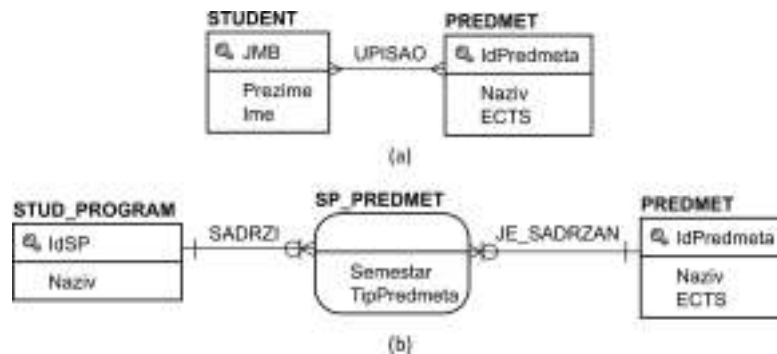
Ograničenja učešća entiteta u vezama specifikuju se na potpuno drugačiji način u odnosu na ranije prikazanu MOV notaciju. Posmatrajmo veze entiteta tipa E1 i tipa E2 (sl. 1.31). Dodatni kružić pored entitetskog tipa E2 specifikuje da postoje entiteti tipa E1 koji nisu u vezi ni sa jednim entitetom tipa E2. Drugim riječima, učešće entiteta tipa E1 je parcijalno (jer postoje entiteti tog tipa koji ne učestvuju u vezama datog tipa). Praktično to znači da se parcijalna participacija entiteta nekog tipa specifikuje kružićem na suprotnoj strani vezi. Na sl. 1.31 dati su tipični primjeri veznih tipova, pri čemu su iznad svakog primjera prikazane kardinalnosti mapiranja i ograničenja učešća entiteta odnosnih entitetskih tipova.

Za razliku od jakih entitetskih tipova, koji se reprezentuju odgovarajućim pravougaonikom (jednodjelnim ili dvodjelnim), slabi entitetski tipovi reprezentuju se pravougaonikom sa zaobljenim vrhovima. Atributi koji predstavljaju diskriminator slabog entitetskog tipa reprezentuju se identično atributima primarnog ključa jakog entitetskog tipa, tj. označavaju se istim simbolom i prikazuju u gornjoj sekciji pravougaonika, kao što je ilustrovano na sl. 1.32.



Slika 1.32 Primjer modelovanja identifikujuće veze i slabog entitetskog tipa

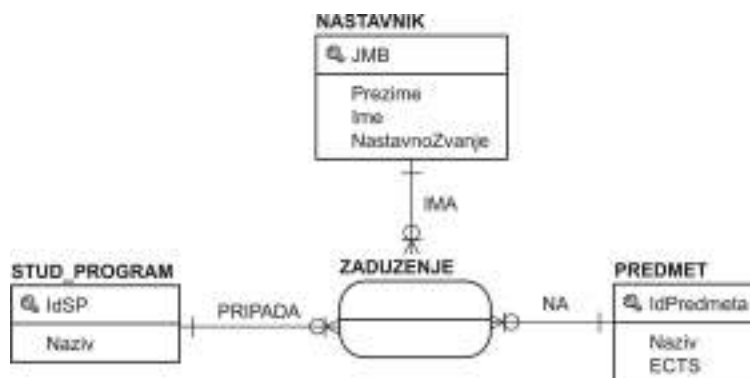
Vezni tip sa kardinalnošću mapiranja *više:više*, ako nema opisne attribute, može da se predstavi kao vezni tip, kao što je ilustrovano na sl. 1.33a. Podrazumijeva se parcijalno učešće entiteta odnosnih tipova u vezama datog tipa, a kružići se ne prikazuju. Međutim, ako se u toku projektovanja identifikuju atributi veznog tipa sa kardinalnošću mapiranja *više:više*, tada takav vezni tip ne može da se predstavi kao vezni tip (jer IE notacija ne omogućava da se veznim tipovima specifikuju atributi), već mora da se reprezentuje odgovarajućim slabim entitetskim tipom (svaka veza datog veznog tipa je egzistencijalno zavisna od entiteta koje povezuje) i identifikujućim veznim tipova sa odnosnim entitetskim tipovima, kao na sl. 1.33b.



Slika 1.33 Reprezentacija veznog tipa sa kardinalnošću mapiranja *više:više*: u slučaju da nema opisne attribute (a) i u slučaju da ima opisne attribute (b)

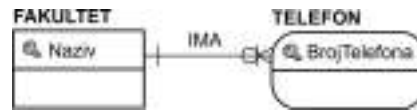
Ako se u toku projektovanja identifikuju atributi nekog veznog tipa čija je kardinalnost mapiranja 1:više, tada postoje dvije mogućnosti. Ako je participacija entiteta na strani *više* totalna, tada se atributi veznog tipa dodaju atributima entitetskog tipa na strani *više*. U suprotnom, dati vezni tip treba da se reprezentuje odgovarajućim slabim entitetskim tipom i identifikujućim veznim tipom sa odnosnim entitetskim tipom na strani *više* i neidentifikujućim veznim tipom sa odnosnim entitetskim tipom na strani 1.

Slično reprezentaciji veznog tipa sa kardinalnošću mapiranja *više:više*, i *n-arni* vezni tipovi reprezentuju se slabim entitetskim tipom i odgovarajućim identifikujućim veznim tipovima, jer IE notacija ne podržava *n-arne* tipove veza. Na sl. 1.34 ilustrovano je modelovanje *ternarnih* veza tipa *PREDAJE* između entiteta tipa *NASTAVNIK*, *STUD\_PROGRAM* i *PREDMET*. Dati ternarni vezni tip reprezentovan je slabim entitetskim tipom *ZADUZENJE* koji je sa odnosnim entitetskim tipovima vezan identifikujućim binarnim veznim tipovima *IMA*, *PRIPADA* i *NA*, respektivno. Svaki entitet tipa *ZADUZENJE* reprezentuje zaduženje jednog nastavnika na konkretnom predmetu nekog studijskog programa i egzistencijalno je zavisno od sva tri odnosna entiteta. S druge strane, ne mora svaki nastavnik da ima zaduženje, mogu da postoje i predmeti za koje nije zadužen nijedan nastavnik i mogu da postoje studijski programi na kojima (još) nema zaduženja.



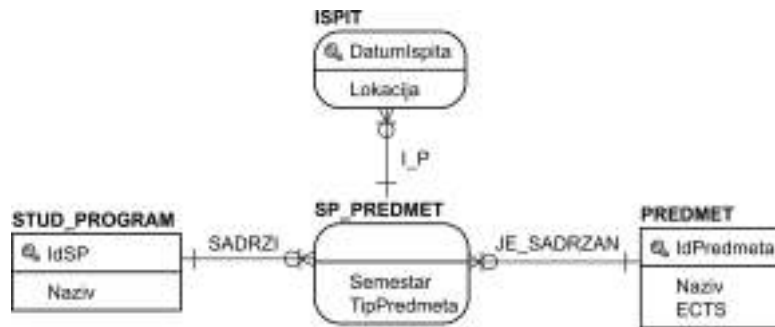
Slika 1.34 Primjer modelovanja ternarnog veznog tipa

IE notacija ne omogućava ni modelovanje višeznačnih atributa, pa višeznačne attribute, koji se identifikuju tokom konceptualnog projektovanja, treba reprezentovati slabim entitetskim tipom i odgovarajućim identifikujućim veznim tipom sa entitetskim tipom kojem pripada identifikovani višeznačni atribut. Na sl. 1.35 ilustrovano je modelovanje višeznačnog atributa *Telefon* entitetskog tipa *FAKULTET*. Osim diskriminatora koji reprezentuje višeznačni atribut, odnosni slabi entitetski tip može da ima i druge attribute. U konkretnom slučaju, osim atributa *BrojTelefona* to mogu da budu atributi koji reprezentuju mjesto kontakta (dekanat, centrala, računovodstvo), serijski broj, inventarni broj, itd.



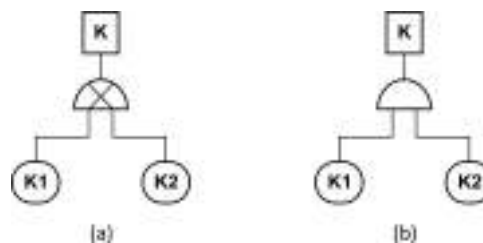
Slika 1.35 Primjer modelovanja višeznačnog atributa

Koncept agregacije nije podržan u IE notaciji. S obzirom na to da se u IE zasnovanom modelu, veze koje imaju vlastita svojstva (a uspostavljanje dodatnih veza je takođe svojevrsno svojstvo) reprezentuju kao entitetski tipovi, modelovanje agregacije u IE notaciji je potpuno u skladu sa definicijom agregacije (agregacija podrazumijeva da se veza posmatra kao entitet višeg nivoa). Na sl. 1.36 ilustrovano je modelovanje agregacije. Entitetski tip *SP\_PREDMET* reprezentuje veze entiteta tipa *PREDMET* i *STUD\_PROGRAM*, pri čemu svaki entitet tipa *SP\_PREDMET* reprezentuje činjenicu da je neki predmet sadržan u nekom studijskom programu. Za svaki takav predmet mogu da se organizuju ispiti, tj. entiteti tipa *SP\_PREDMET* (koji reprezentuju veze) su vezama tipa *I\_P* vezani sa entitetima tipa *ISPIT*. Dakle, entitetski tip *SP\_PREDMET* predstavlja agregaciju.



Slika 1.36 Primjer modelovanja agregacije

IE notacija omogućava modelovanje veza specijalizacije/generalizacije. Podržano je modelovanje *disjunktne (ekskluzivne)* i *preklapajuće (inkluzivne)* specijalizacije. Za razliku od ranije prikazane MOV notacije, nije podržano modelovanje ograničenja učešća (totalno/parcijalno) entiteta natklase. Disjunktna specijalizacija ilustrovana je na sl. 1.37a, a preklapajuća na sl. 1.37b). Potklase se u IE dijagramu reprezentuju slabim entitetskim tipovima.



Slika 1.37 Modelovanje specijalizacije: disjunktna (a) i preklapajuća (b)

**Primjer 1.3** Prikazati konceptualni model podataka univerziteta IE notacijom.**Rješenje.**

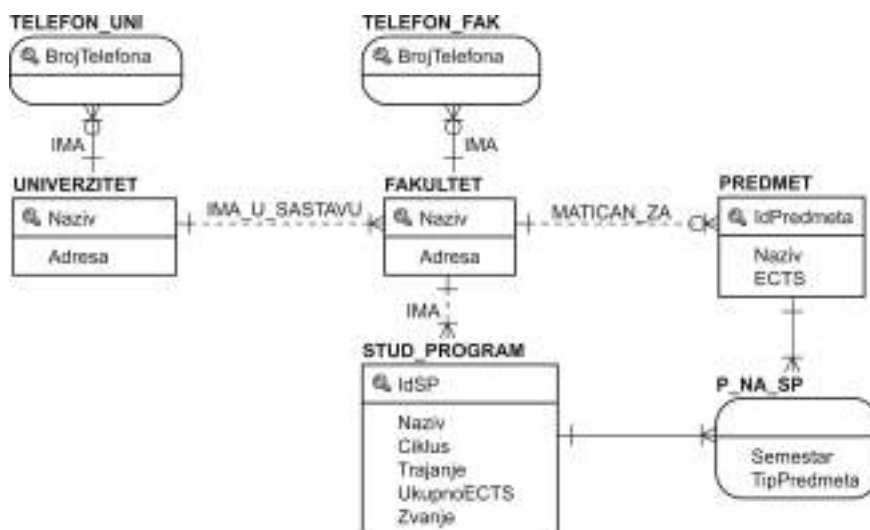
Tekstualna specifikacija informacionih potreba univerziteta i odgovarajući MOV dijagram dati su u primjeru 1.1. Ovdje ćemo samo za svaki od dijelova specifikacije dati ekvivalentni IE dijagram uz odgovarajuća pojašnjenja.

**IE dijagram za prvi dio specifikacije**

IE dijagram koji odgovara prvom dijelu tekstualne specifikacije i koji je ekvivalentan MOV dijagramu sa sl. 1.6, prikazan je na sl. 1.38.

Budući da IE notacija nema podršku za modelovanje višeznačnih atributa, slabi entitetski tipovi *TELEFON\_UNI* i *TELEFON\_FAK* reprezentuju višeznačne attribute *Telefon* entitetskih tipova *UNIVERZITET* i *FAKULTET*, respektivno. Primijetimo da su oba višeznačna atributa (i bez obzira na to što reprezentuju ista svojstva) reprezentovana zasebnim entitetskim tipovima (jer reprezentuju svojstva različitih entiteta).

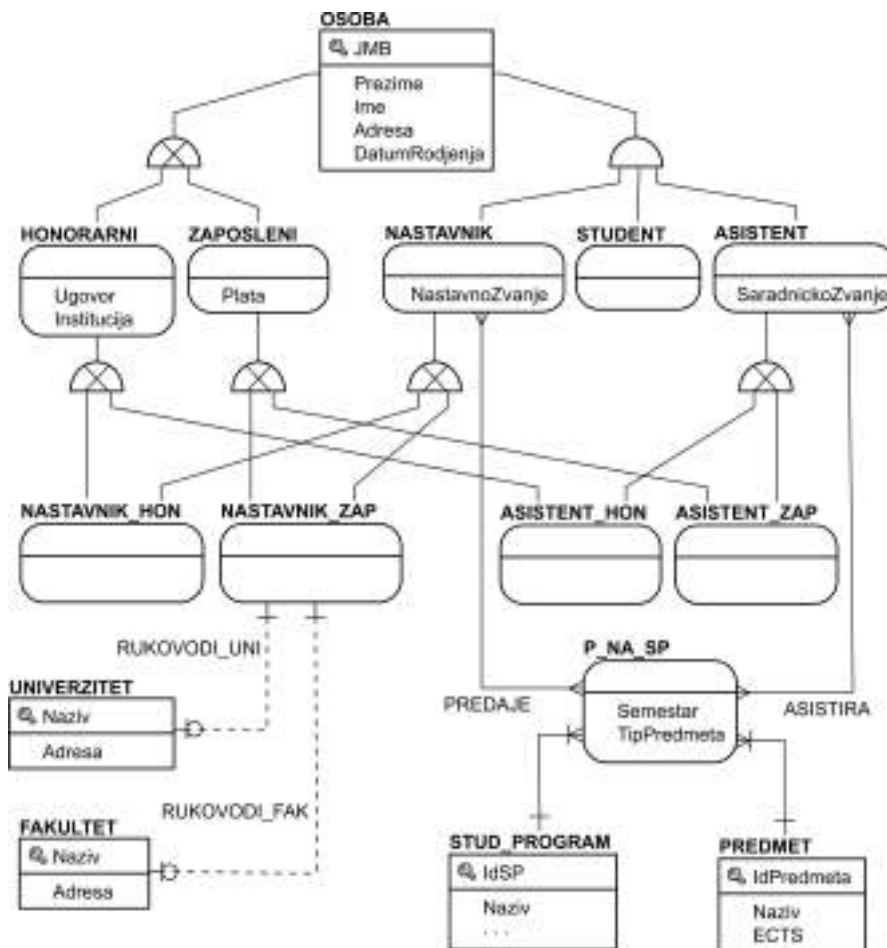
IE notacija onemogućava specifikaciju atributa veznim tipovima. Zbog toga je vezni tip *P\_NA\_SP* (iz modela sa sl. 1.6) sa kardinalnošću mapiranja *više:više* i *totalnom participacijom* entiteta oba tipa, reprezentovan odgovarajućim slabim entitetskim tipom i identifikujućim veznim tipovima prema odnosnim entitetskim tipovima. Kardinalnost mapiranja oba identifikujuća vezna tipa je 1:1..\*, jer svaki studijski program mora da ima definisane predmete, a svaki predmet mora da pripada nekom studijskom programu.



**Slika 1.38** IE dijagram koji odgovara prvom dijelu specifikacije informacionih potreba univerziteta iz primjera 1.1 i koji je ekvivalentan MOV dijagramu sa sl. 1.6

### IE dijagram za drugi dio specifikacije

IE dijagram koji odgovara drugom dijelu specifikacije informacionih potreba i koji je ekvivalentan MOV dijagramu sa sl. 1.13, prikazan je na sl. 1.39.



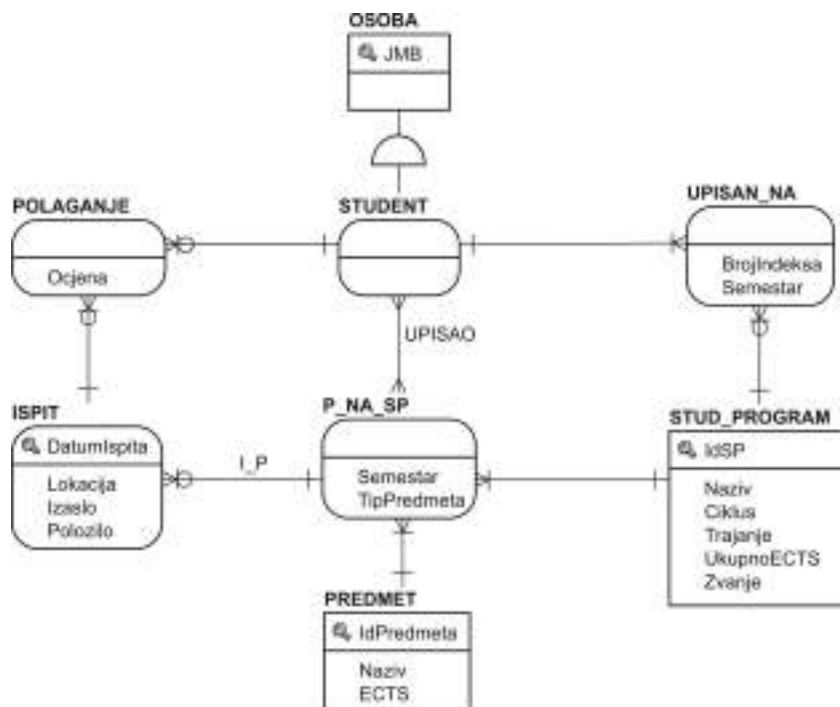
**Slika 1.39** IE dijagram koji odgovara drugom dijelu specifikacije informacionih potreba univerziteta iz primjera 1.1 i koji je ekvivalentan MOV dijagramu sa sl. 1.13

Zaduženja nastavnika i asistenata modelovana su agregacijom, tj. nastavnici i asistenti imaju zaduženja na predmetima koji pripadaju nekom studijskom programu (predmet koji pripada studijskom programu reprezentovan je slabim entitetskim tipom *P\_NA\_SP*).

Kao što je ranije navedeno, potklase se u IE dijagramu reprezentuju slabim entitetskim tipovima. Takođe, ne postoji mogućnost da se specifikuje totalna, odnosno parcijalna specijalizacija natklase po nekom kriterijumu, nego samo da se specifikuje da li je data specijalizacija disjunktna ili preklapajuća.

### IE dijagram za treći dio specifikacije

IE dijagram koji odgovara trećem dijelu specifikacije informacionih potreba i koji je ekvivalentan MOV dijagramu sa sl. 1.18b), prikazan je na sl. 1.40.



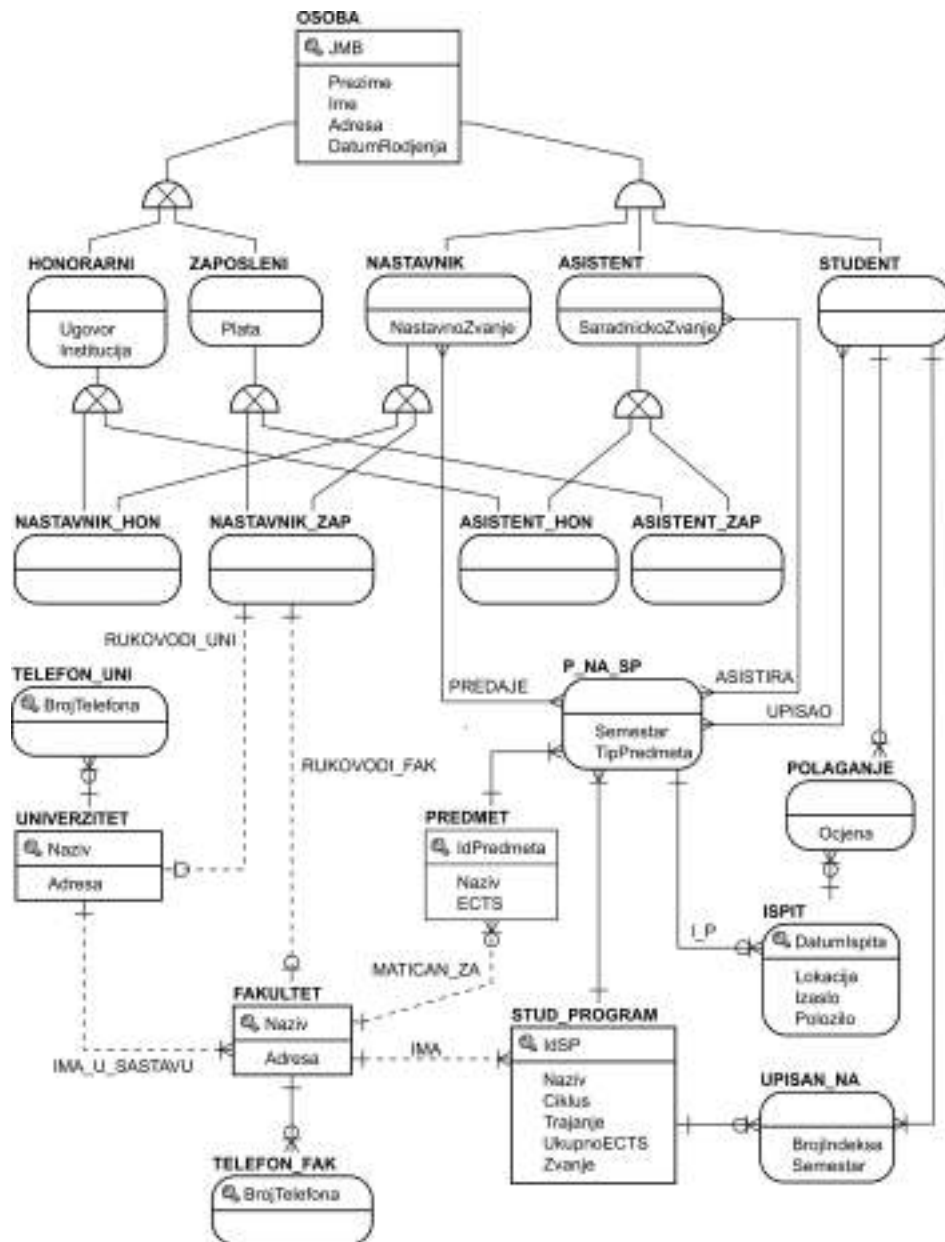
**Slika 1.40** IE dijagram koji odgovara trećem dijelu specifikacije informacionih potreba univerziteta iz primjera 1.1 i koji je ekvivalentan MOV dijagramu sa sl. 1.18b)

Vezni tipovi sa *UPISAN\_NA* i *POLAŽE* (sa MOV dijagrama na sl. 1.18b), čija je kardinalnost mapiranja *više:više* i koji imaju opisne atribute, reprezentovani su odgovarajućim slabim entitetskim tipovima (*UPISAN\_NA* i *POLAGANJE*) i parovima identifikujućih veznih tipova. Nazivi veznih tipova, odnosno uloge entiteta u vezama nisu navedene kako bi se naglasilo da je riječ o entitetskim tipovima koji reprezentuju vezne tipove.

Vezni tip *UPISAO* (sa MOV dijagrama na sl. 1.18b) također ima kardinalnost mapiranja *više:više*, ali nema opisne atribute, pa je reprezentovan istoimenim veznim tipom.

IE notacija nema podršku za izvedene atribute. S obzirom na to koliki značaj u sistemu imaju, (izvedeni) atributi *Izaslo* i *Polozilo* su zadržani u modelu i reprezentovani kao (osnovni) atributi slabog entitetskog tipa *ISPIT*.

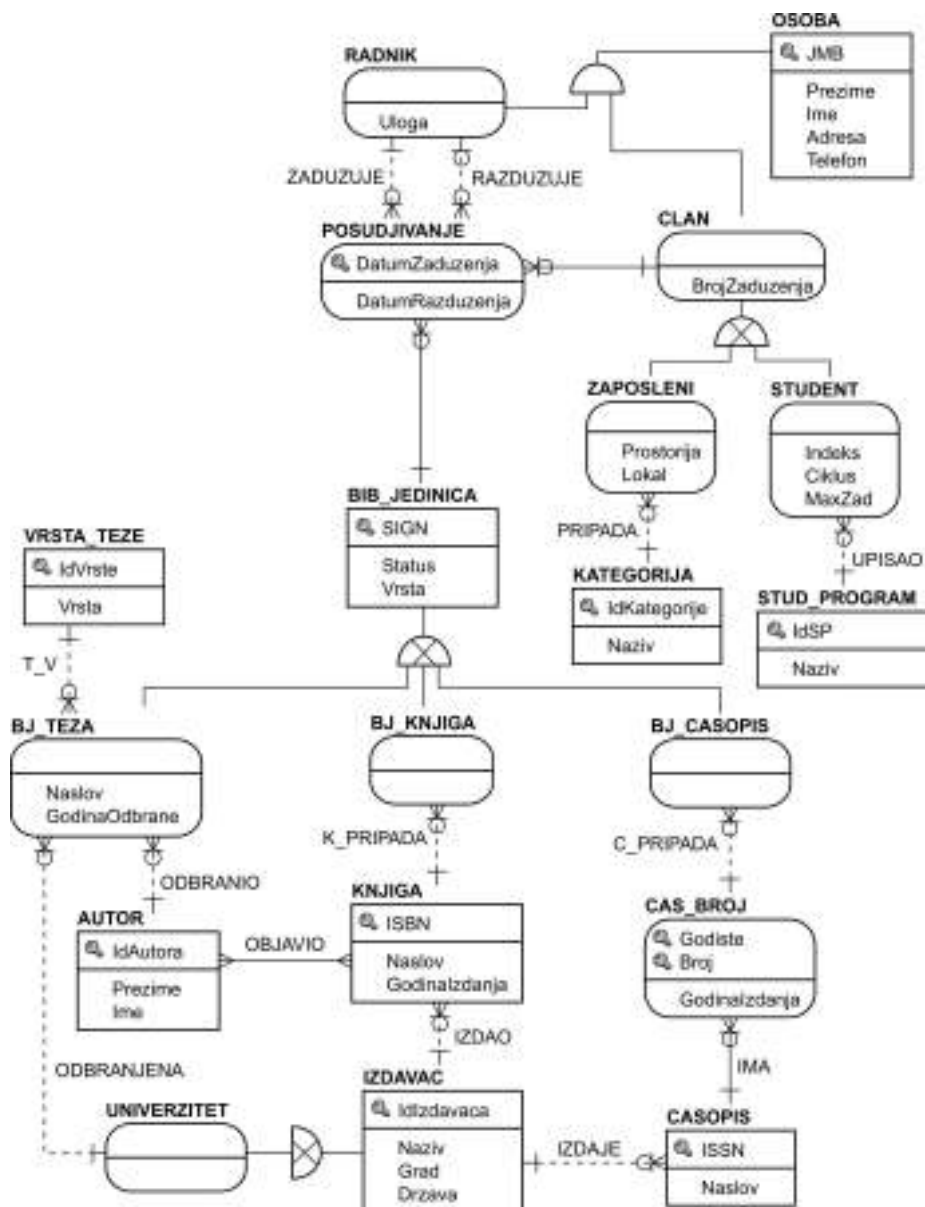
Na sl. 1.41 prikazan je cjelokupni IE dijagram koji reprezentuje konceptualni model podataka univerziteta u skladu sa specifikovanim informacionim potrebama i koji je semantički ekvivalentan MOV dijagramu sa sl. 1.19.



Slika 1.41 IE dijagram koji reprezentuje uprošteni konceptualni model podataka univerziteta

**Primjer 1.4** Prikazati IE konceptualni model podataka fakultetske biblioteke.**Rješenje.**

Specifikacija informacionih potreba i odgovarajući MOV dijagram dati su u primjeru 1.2. Ekvivalentni IE dijagram prikazan je na sl. 1.42.



Slika 1.42 IE dijagram fakultetske biblioteke ekvivalentan MOV dijagramu sa sl. 1.29



---

**Primjer 1.5** Primjenom IE notacije projektovati konceptualni model podataka stanice za otkup svježeg mlijeka od individualnih proizvođača u nekom mjestu. Sistem treba da omogući vođenje evidencija o proizvođačima, otkupu i isplatama. Za svaku otkupljenu količinu bitna je informacija o kvalitetu. Proizvođač za svaku predatu količinu dobija potvrdu o količini i kvalitetu predatog mlijeka. Proizvođačima se isplate vrše povremeno za prethodno predate neisplaćene količine.

---

### Rješenje.

Predmet projektnog zadatka je (relativno) jednostavan sistem u kojem mogu da se identifikuju, na osnovu zadate specifikacije, tri vrste entiteta o kojima se vode evidencije (proizvođači, otkupljene količine, isplate).

Proizvođači se modeluju entitetskim tipom *PROIZVODJAC* sa atributima: *JMB*, *Prezime*, *Ime*, *Adresa*, *Telefon*. Odnosna svojstva proizvođača su tipična svojstva svake osobe (prema specifikaciji, radi se o *individualnim* proizvođačima).

Budući da je predmet otkupa u otkupnoj stanici samo jedna vrsta robe, odnosno sirovine (svježe mlijeko), nije potrebno uvoditi poseban entitetski tip kojim bi se ta roba reprezentovala, jer se uvijek zna da je riječ o sirovom mlijeku. Dovoljno je samo reprezentovati otkupljene količine. Podaci o otkupljenim količinama modeluju se entitetskim tipom *OTKUP* sa atributima: *BrojPotvrde*, *Datum*, *Kolicina*, *Kvalitet*. Za svaku predatu količinu proizvođaču se izdaje potvrda o otkupu pa je *BrojPotvrde* atribut koji jedinstveno identifikuje svaki otkup i predstavlja primarni ključ odnosnog entitetskog tipa.

Isplate se modeluju entitetskim tipom *ISPLATA* sa atributima: *BrojIsplate*, *Datum*, *Iznos*. Dodatno još može da se uvede i izvedeni atribut *Kolicina* koji reprezentuje ukupnu količinu mlijeka obuhvaćenu konkretnom isplatom (izvedeni, jer se ukupna količina može izračunati kao zbir svih otkupljenih količina koje su obuhvaćene nekom isplatom).

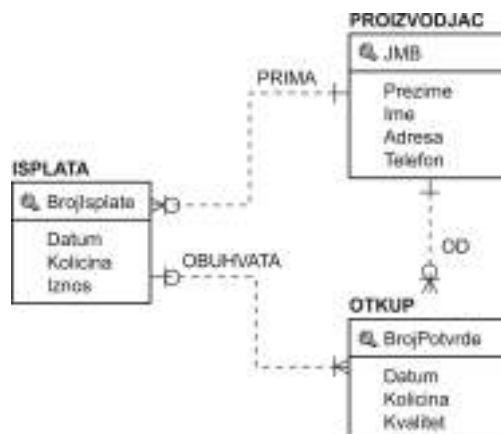
Veze tipa *OD* između entitetskih tipova *OTKUP* i *PROIZVODJAC* nisu identifikujuće (oba entitetska tipa su jaka, jer imaju primarni ključ). Kardinalnost mapiranja je 0..\*:1 (posmatrano sa strane entitetskog tipa *OTKUP*), jer je svaki otkup *od* nekog proizvođača, a mogu da postoje proizvođači koji su tek registrovani i još nisu predavali mlijeko.

Veze tipa *OBUHVATA* između entitetskih tipova *ISPLATA* i *OTKUP* nisu identifikujuće (oba entitetska tipa imaju primarni ključ). Kardinalnost mapiranja je 0..1:1..\*: (posmatrano sa strane entitetskog tipa *ISPLATA*), jer svaka isplata obuhvata bar jedan otkup, a svaki otkup tek nakon nekog vremena biva obuhvaćen nekom isplatom.

Veze tipa *PRIMA* između entitetskih tipova *PROIZVODJAC* i *ISPLATA* nisu identifikujuće (oba entitetska tipa imaju primarni ključ). Kardinalnost mapiranja je

1:0..\*: (posmatrano sa strane entitetskog tipa *PROIZVODJAC*), jer svaka isplata pripada nekom proizvođaču, a mogu da postoje proizvođači koji još nisu dobili nijednu isplatu.

Odgovarajući IE dijagram prikazan je na sl. 1.43.



Slika 1.43 IE zasnovani konceptualni model podataka otkupne stanice u jednom mjestu

**Primjer 1.6** Prilagoditi konceptualni model podataka iz primjera 1.5 tako da zadovoljava informacione potrebe lanca otkupnih stanica u različitim mjestima u kojima se otkupljuje sirovo mlijeko različitog porijekla (npr. kravlje, kozje, ...), ne samo od individualnih proizvođača, nego i od drugih preduzeća.

### Rješenje.

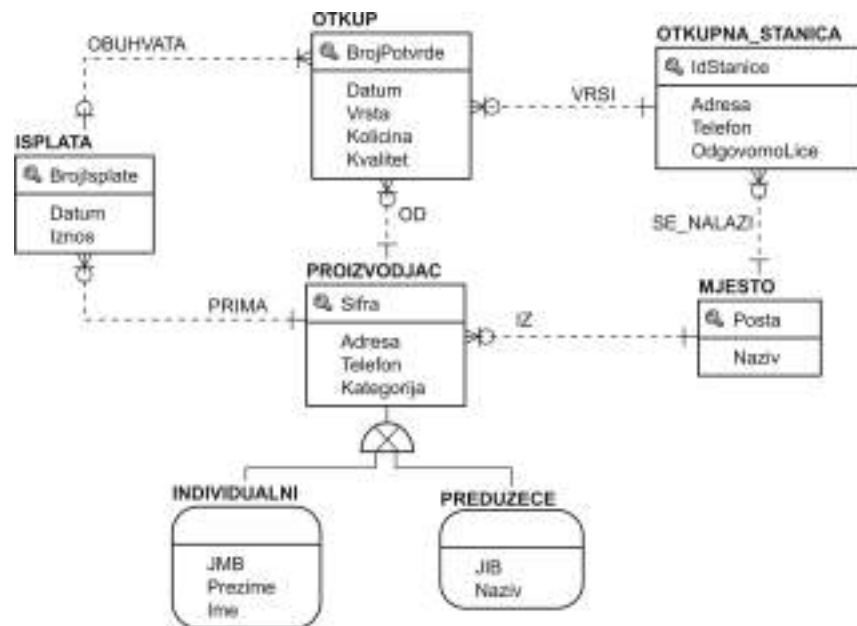
U odnosu na sistem iz primjera 1.5, ovaj poslovni sistem ima dvije kategorije (podvrste) proizvođača: *individualni proizvođači* i *preduzeća*. Ove dvije podvrste dijelom karakterišu ista svojstva (*adresa, telefon, mjesto* u kojem žive ili imaju sjedište), ali postoje i svojstva specifična za svaku podvrstu. Pojednice karakterišu: *JMB, prezime* i *ime*, dok preduzeća karakterišu: *JIB* (jedinstveni identifikacioni broj) i *naziv*. Zbog toga se entitetski tip *PROIZVODJAC* specijalizuje odgovarajućim potklasama *INDIVIDUALNI* (sa atributima: *JMB, Prezime, Ime*) i *PREDUZECE* (sa atributima: *JIB, Naziv*). Zajednička svojstva reprezentuju se atributima superklase. Tako su modelovana svojstva *adresa* i *telefon*. Svojstvo *mjesto* može da se modeluje odgovarajućim atributom superklase. Međutim, bolje rješenje je uvođenje entitetskog tipa *MJESTO* (i odgovarajućeg veznog tipa *IZ*) kao šifarnika mjesta. Dodatni razlog za uvođenje entitetskog tipa *MJESTO* jeste i činjenica da se svaka otkupna stanica takođe nalazi u nekom mjestu. Budući da entitetski tip *PROIZVODJAC* treba da bude jaki entitetski tip, još treba dodati surogat ključ *Sifra*. Dodatno još superklasa ima diskriminator *Kategorija*.

Svaki otkup vrši se u nekoj *otkupnoj stanici* (koja se nalazi u nekom mjestu), pa u model treba uvesti i entitetski tip *OTKUPNA\_STANICA* sa odgovarajućim atributima: *IdStanice* (surogat ključ), *Adresa*, *Telefon*, *OdgovornoLice*. Ovakav model omogućava da u svakoj otkupnoj stanici postoji jedno odgovorno lice. Dodatno bi se, umjesto atributa *OdgovornoLice*, mogao uvesti entitetski tip *ZAPOSLjeni* (sa atributima: *JMB*, *Prezime*, *Ime*, *Uloga*), koji je sa entitetskim tipom *OTKUPNA\_STANICA* vezan vezama tipa *IMA*, pri čemu bi kardinalnost mapiranja bila 1:1..\* (posmatrano sa strane entitetskog tipa *OTKUPNA\_STANICA*).

Kardinalnost mapiranja veznog tipa *VRSI* između entitetskih tipova *OTKUPNA\_STANICA* i *OTKUP* je 1:0..\*, jer se svaki otkup vrši u nekoj stanici, a mogu da postoje stanice (novoformirane) u kojima još nije bilo otkupa.

Prema zadatoj specifikaciji, otkupne stanice vrše otkup sirovog mlijeka različitog porijekla. Dakle, bitno svojstvo svake otkupljene količine (svakog otkupa) je i porijeklo, odnosno *vrsta* mlijeka. Ovo svojstvo može da se reprezentuje odgovarajućim atributom *Vrsta*. Alternativno, može da se uvede entitetski tip *VRSTA\_MLIJEKA* (kao šifarnik).

Preostali dio modela iz primjera 1.5 ne treba da se modifikuje. Odgovarajući IE dijagram prikazan je na sl. 1.44.



Slika 1.44 IE zasnovani konceptualni model podataka lanca otkupnih stanica

Treba imati u vidu da proizvođač za svaku predatu količinu dobija potvrdu o otkupu. To znači da će proizvođač koji preda dvije vrste mlijeka u istom danu (npr. i kozje i ovčje) dobiti dvije potvrde.

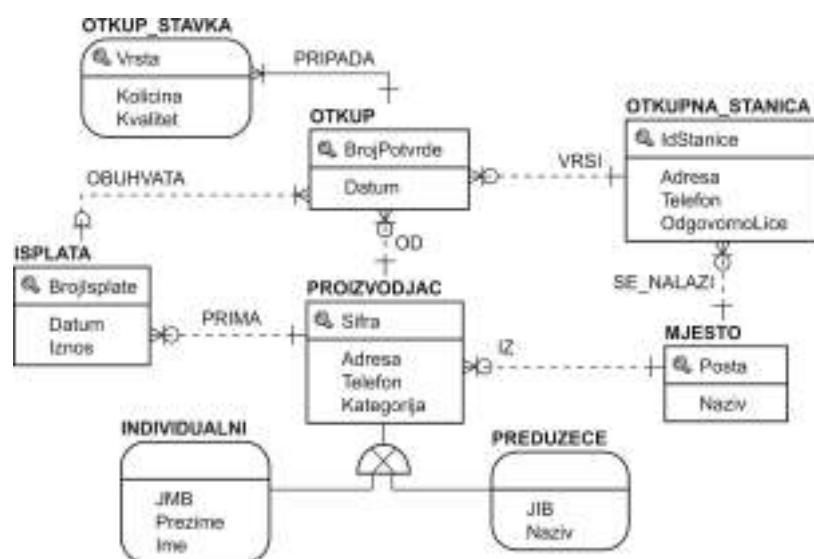
**Primjer 1.7** Prilagoditi konceptualni model iz primjera 1.6 tako se proizvođaču prilikom otkupa izdaje samo jedna potvrda za sve predate količine.

### Rješenje.

Prema modelu iz primjera 1.6, proizvođač koji u jednom danu preda dvije različite vrste mlijeka mora da dobije dvije potvrde o otkupu – za svaku vrstu mlijeka po jednu potvrdu. Ovdje se zahtijeva da se takvom proizvođaču izdaje samo jedna potvrda.

Dakle, proizvođaču se izdaje jedna potvrda o otkupu u jednom danu, a svaki otkup obuhvata određenu količinu jedne ili više različitih vrsta mlijeka određenog kvaliteta. To znači da treba uvesti dodatni entitetski tip *OTKUP\_STAVKA*, kojim se reprezentuju tzv. *stavke* otkupa, pri čemu se svaka stavka odnosi na jednu vrstu mlijeka. Svojstva svake stavke su: *vrsta* mlijeka, *količina* i *kvalitet*, pa su atributi entitetskog tipa *OTKUP\_STAVKA*: *Vrsta*, *Kolicina*, *Kvalitet*. Nijedan od ova tri atributa, niti pojedinačno niti u kombinaciji, ne predstavlja primarni ključ. Osim toga, svaka stavka egzistencijalno zavisi od otkupa, pa je *OTKUP\_STAVKA* slabi entitetski tip (atribut *Vrsta* predstavlja diskriminator), a vezni tip *PRIPADA* identifikujući sa kardinalnošću mapiranja 1:1..\* (sa strane tipa *OTKUP*).

Nakon uvođenja slabog entitetskog tipa *OTKUP\_STAVKA* i izdvajanja dijela atributa (*Vrsta*, *Kolicina*, *Kvalitet*), entitetskom tipu *OTKUP* preostaju atributi *BrojPotvrde* i *Datum*. Preostali dio modela iz primjera 1.6 ne treba da se modifikuje. Odgovarajući IE dijagram prikazan je na sl. 1.45.



**Slika 1.45** IE zasnovani konceptualni model podataka lanca otkupnih stanica koji omogućava da se proizvođaču prilikom otkupa izdaje samo jedna potvrda

---

**Primjer 1.8** Prilagoditi konceptualni model iz primjera 1.7 tako da zadovoljava informacione potrebe lanca otkupnih stanica, koji otkupljeno mlijeko isporučuje različitim mljekarama.

---

### Rješenje.

Prema specifikaciji, otkupljeno mlijeko isporučuje se različitim *mljekarama*, pa treba da se uvede entitetski tip *MLJEKARA* sa atributima: *JIB*, *Naziv*, *Adresa*, *Telefon*. Svaka mljekara ima *sjedište* u nekom *mjestu*, pa treba uvesti i vezni tip *IMA\_SJEDISTE* između odnosnih entitetskih tipova *MLJEKARA* i *MJESTO*, sa kardinalnošću mapiranja 0..\*:1 (svaka mljekara ima sjedište, a postoje mjesta u kojima ne postoje mljekare).

Mljekare *preuzimaju* određene količine otkupljenog mlijeka različitih vrsta i kvaliteta. Slično otkupu, svaku *isporuku* karakterišu *datum* isporuke i odgovarajući *dokument* (tipično se takav dokument zove *otpremnica*). Dakle, isporuke treba da se modeluju odgovarajućim entitetskim tipom *ISPORUKA* sa atributima *BrojOtpremnice* i *Datum*. Atribut *BrojOtpremnice* je primarni ključ, jer svaka otpremnica odgovara tačno jednoj isporuci nekoj mljekari, a svaka otpremnica ima jedinstveni broj. Kardinalnost mapiranja veznog tipa *PREUZIMA* između entitetskih tipova *MLJEKARA* i *ISPORUKA* je 1:0..\* (svaku isporuku preuzima neka mljekara, svaka mljekara može da preuzme više isporuka, a mogu da postoje i mljekare koje još nisu preuzimale isporuke).

Svaka isporuka ima *stavke*, pri čemu se svaka stavka odnosi na jednu vrstu isporučenog mlijeka. Slično otkupu, svojstva svake stavke (*vrsta*, *količina*, *kvalitet*) mogu da se reprezentuju odgovarajućim atributima (*Vrsta*, *Kolicina*, *Kvalitet*) odnosnog entitetskog tipa *ISPORUKA\_STAVKA*. Budući da se *vrsta* pojavljuje i kao svojstvo isporuke (odnosno stavke isporuke) i kao svojstvo otkupa (odnosno stavke otkupa), bolje je da se odnosno svojstvo reprezentuje odgovarajućim entitetskim tipom (npr. *MLIJEKO*) i odgovarajućim veznim tipom. Pošto se svojstvo *vrsta* reprezentuje entitetskim tipom, atributi entitetskog tipa *ISPORUKA\_STAVKA* su *Kolicina* i *Kvalitet*.

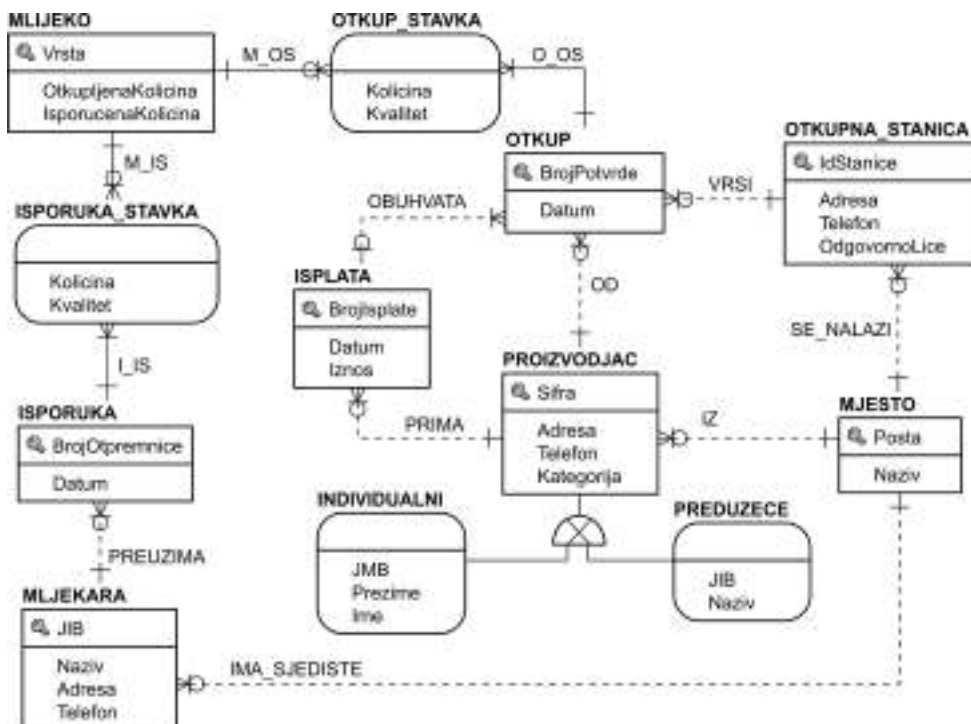
Za svaku vrstu mlijeka može da postoji više stavki (u različitim isporukama), a svaka stavka (neke isporuke) odnosi se na jednu vrstu mlijeka. Kao što je svaka stavka egzistencijalno zavisna od isporuke kojoj pripada (identifikujući vezni tip *I\_IS*), tako je svaka stavka istovremeno egzistencijalno zavisna i od konkretne vrste mlijeka. Zato je odgovarajući vezni tip (*M\_IS*) između entitetskih tipova *MLIJEKO* i *ISPORUKA\_STAVKA* identifikujući vezni tip sa kardinalnošću mapiranja 1:0..\*.

Analogno prethodnim razmatranjima, treba uvesti identifikujući vezni tip *M\_OS* između entitetskih tipova *MLIJEKO* i *OTKUP\_STAVKA*, te eliminisati diskriminator *Vrsta* iz slabog entitetskog tipa *OTKUP\_STAVKA*.

Entitetski tip *MLIJEKO* uveden je (prvenstveno) kao šifarnik vrsta mlijeka. Međutim, uvođenje entitetskog tipa omogućava da se reprezentuju i druga svojstva.

Tako se mogu uvesti atributi *OtkupljenaKolicina* i *IsporucenaKolicina*, koji za svaku vrstu mlijeka sadrže ukupnu otkupljenu i ukupnu isporučenu količinu, respektivno. Iako se ovi podaci mogu izračunati kao sume vrijednosti atributa *Kolicina* u stavkama otkupa i stavkama isporuka, respektivno, ipak se u praksi često uvode ovakvi atributi koji sadrže tzv. *sintetičke* (sumarne ili zbirne) vrijednosti. Razlika vrijednosti atributa *OtkupljenaKolicina* i *IsporucenaKolicina* u svakom trenutku predstavlja *stanje* (raspoloživu količinu) mlijeka u okviru lanca.

Odgovarajući IE dijagram prikazan je na sl. 1.46.



**Slika 1.46** IE zasnovani konceptualni model podataka lanca otkupnih stanica koji otkupljeno mlijeko isporučuje različitim mljekarama

Ovaj primjer ilustruje primjenu tipičnog projektnog obrasca koji se koristi u modelovanju robnih (i finansijskih) tokova u poslovnim sistemima, poznat kao: *DOBAVLJAČ/KUPAC-DOKUMENT-STAVKA-ROBA*. Dati obrazac primijenjen je dva puta u ovom primjeru, i to za modelovanje tzv. *ulazne strane (dobavljači)*: *PROIZVODJAC-OTKUP-OTKUP\_STAVKA-MLIJEKO*, te za modelovanje tzv. *izlazne strane (kupci)*: *MLJEKARA-ISPORUKA-ISPORUKA\_STAVKA-MLIJEKO*. Tipično se, umjesto entitetskih tipova *OTKUP* i *ISPORUKA* koji su korišteni u ovom primjeru, odnosni entitetski tipovi nazivaju prema odovarajućim dokumentima koji prate robne (finansijske) tokove, npr. *NARUDŽBENICA*, *OTPREMNICA*, *FAKTURA*, itd.

**Primjer 1.9** Modifikovati konceptualni model iz primjera 1.8 tako da zadovoljava sljedeću specifikaciju informacionih potreba vezanih za obračun otkupljenog mlijeka i isplate proizvođačima:

**Obračun:**

*Svaka vrsta mlijeka ima osnovnu jediničnu cijenu po litru i jediničnu premiju po litru i masnoj jedinici (masna jedinica je mjera kvaliteta). Ukupna cijena otkupljenog litra jednaka je zbiru osnovne jedinične cijene i jedinične premije pomnožene brojem masnih jedinica:*

$$\text{OtkupnaCijena} = \text{JedCijena} + \text{JedPremija} * \text{Kvalitet}$$

*Otkupna vrijednost mlijeka izračunava se na osnovu važeće cijene na dan otkupa prema formuli:*

$$\text{OtkupnaVrijednost} = \text{OtkupnaCijena} * \text{Količina}$$

**Isplate:**

*Isplate proizvođačima vrše se povremeno za prethodno predate neisplaćene količine. Obračunati iznos isplaćuje se na odgovarajući račun proizvođača u nekoj banci. Svaki proizvođač može da ima više računa kod različitih banaka.*

**Rješenje.**

Svaka vrsta mlijeka ima jediničnu cijenu i jediničnu premiju, pa odnosnom entitetskom tipu *MLIJEKO* (iz modela sa sl. 1.46) treba dodati odgovarajuće atribute *JedCijena* i *JedPremija*, respektivno.

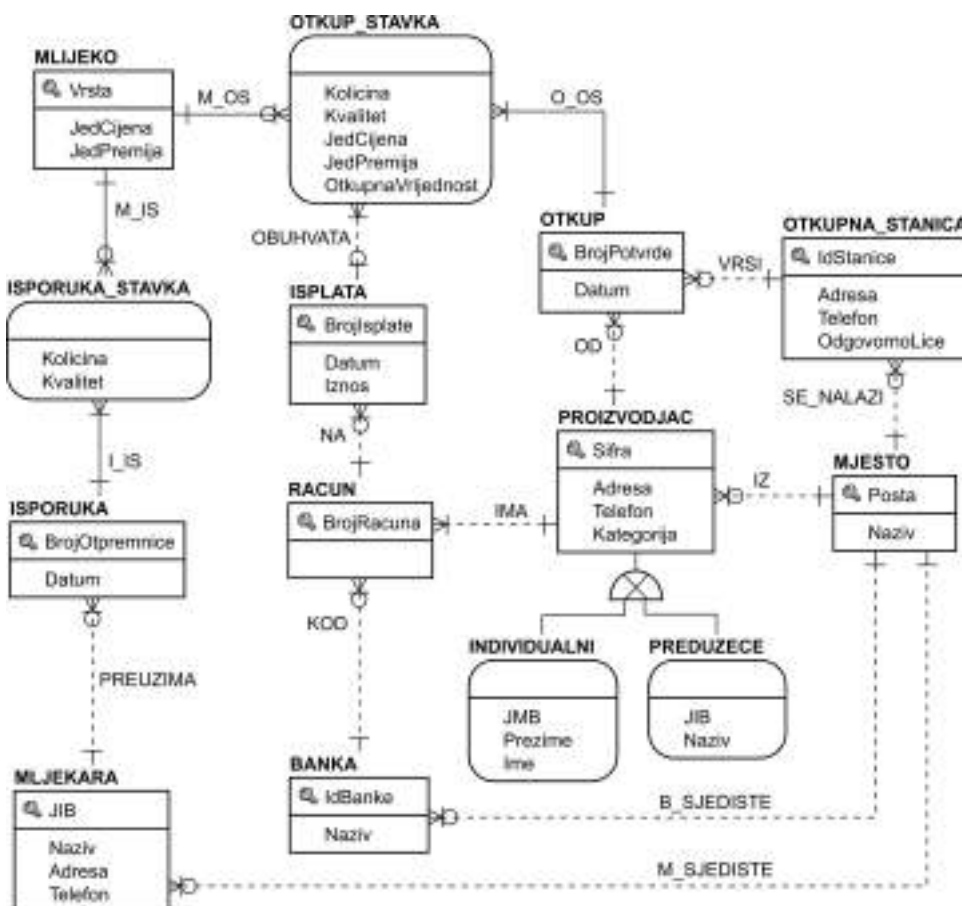
Prema specifikaciji, za svaku otkupljenu količinu obračunava se otkupna vrijednost na osnovu važeće cijene, koja se izračunava na osnovu važeće jedinične cijene, važeće jedinične premije i kvaliteta otkupljenog mlijeka. Veoma je bitna činjenica da se obračun vrši na osnovu važeće jedinične cijene i važeće jedinične premije. Važeća cijena na dan otkupa uvijek je sadržana u atributima *JedCijena* i *JedPremija* entitetskog tipa *MLIJEKO*. Cijena i premija mogu tokom vremena da se mijenjaju, pa mogu postojati količine koje su otkupljene po različitim cijenama. To znači da se za svaku otkupljenu količinu moraju evidentirati jedinična cijena i jedinična premija koje su bile važeće prilikom otkupa. Zato i odnosni entitetski tip *OTKUP\_STAVKA* takođe treba da ima atribute *JedCijena* i *JedPremija*. Pošto se otkupna cijena izračunava (prema prvoj formuli) ne mora da se uvodi (izvedeni) atribut *OtkupnaCijena*. Iako se i vrijednost otkupljene količine izračunava (prema drugoj formuli), ipak je poželjno uvesti atribut *OtkupnaVrijednost*, jer se ta vrijednost koristi prilikom isplate proizvođaču.

Svaka isplata obuhvata određen broj otkupljenih količina, pa je bolje da odnosni vezni tip *OBUHVATA* povezuje entitetske tipove *ISPLATA* i *OTKUP\_STAVKA* (nego *ISPLATA* i *OTKUP*), što u eksploataciji omogućava lakše uspostavljanje veza između isplata i otkupljenih količina.

Svaka isplata vrše se na račun kod neke od banaka, pa treba uvesti entitetski tip *BANKA* sa atributima *IdBanke* i *Naziv*. Svaka banka ima sjedište u nekom mjestu, pa još treba uspostaviti veze tipa *B\_SJEDISTE* prema entitetskom tipu *MJESTO*.

Svaki proizvođač (bez obzira na to kojoj kategoriji pripada) može da ima jedan ili više računa na koje se vrše isplate. Zato treba da se uvede odnosni entitetski tip *RACUN*. Za otkupni lanac bitna su tri svojstva svakog računa: broj računa (atribut *BrojRacuna*), kod koje banke je otvoren (što se reprezentuje vezama tipa *KOD* prema entitetskom tipu *BANKA*, sa kardinalnošću mapiranja 1:0..\* posmatrano sa strane entitetskog tipa *BANKA*), i kojem proizvođaču pripada (što se reprezentuje vezama tipa *IMA* prema entitetskom tipu *PROIZVODJAC*, sa kardinalnošću mapiranja 1:1..\* posmatrano sa strane entitetskog tipa *PROIZVODJAC*).

Svaka isplata ide na tačno jedan račun, a na neki račun može da ide više isplata (tokom vremena), pa između entitetskih tipova *ISPLATA* i *RACUN* treba uvesti vezni tip *NA* sa kardinalnošću mapiranja 0..\*:1.

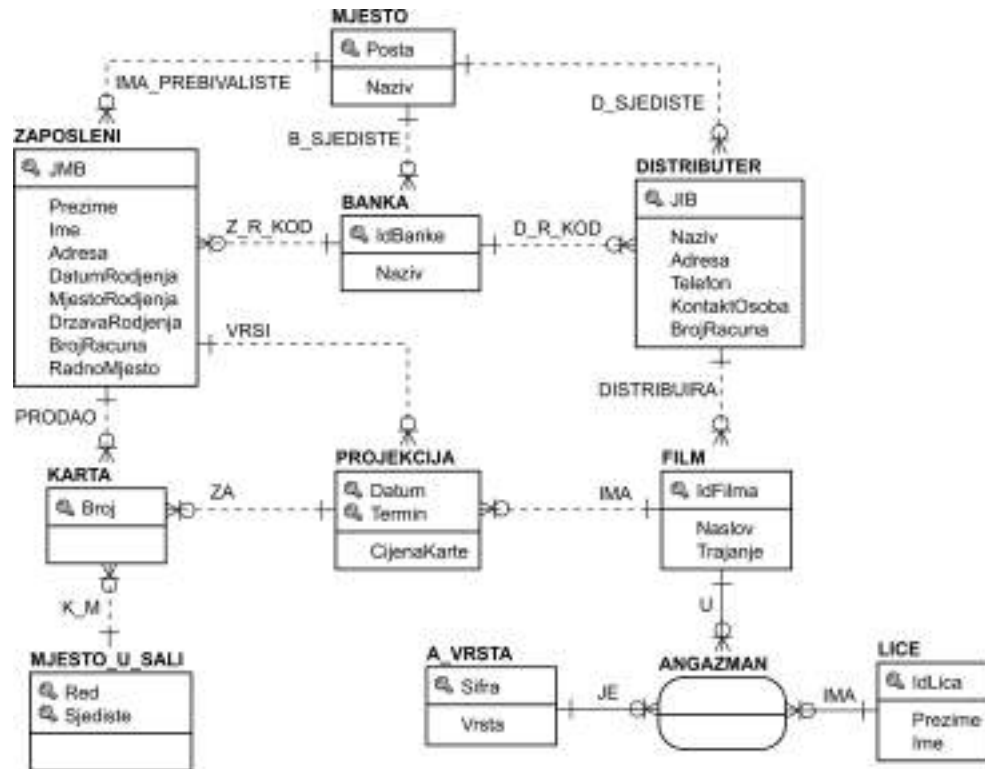


Slika 1.47 IE zasnovani konceptualni model podataka lanca otkupnih stanica



**Primjer 1.10** Projektovati konceptualni model podataka bioskopa sa jednom salom. Ciljna baza podataka treba da omogući vođenje evidencija o: filmovima (naslov, trajanje, režija, uloge, ...) i distributerima filmova (osnovni podaci, sjedište, kontakt, bankovni račun), projekcijama filmova (termini, prodate karte), licima koja su zaposlena u bioskopu (blagajnici, operateri, obezbjeđenje, ...).

### Rješenje.



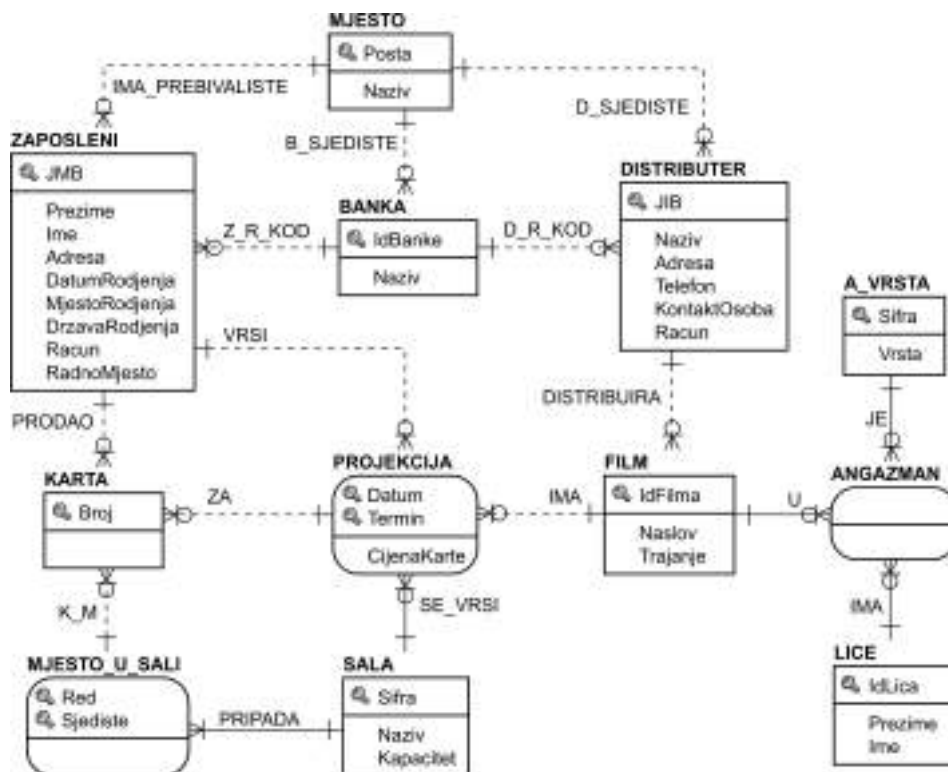
Slika 1.48 IE zasnovani konceptualni model podataka bioskopa

Svaki film može da ima više projekcija. Svaka projekcija jedinstveno je identifikovana datumom i terminom održavanja. Svaku projekciju vrši neko od zaposlenih. Sve karte su numerisane i prodaju se za tačno određeno mjesto u bioskopskoj sali. Svaku kartu prodaje neko od zaposlenih. Svako mjesto u sali jedinstveno je identifikovano brojem reda i brojem sjedišta u redu.

U svakom filmu angažman mogu da imaju različita lica po različitim osnovima. Isto lice može da ima više različitih angažmana u istom filmu (npr. režiser, glavna uloga i producent).

**Primjer 1.11** Modifikovati konceptualni model podataka iz primjera 1.10 tako da zadovoljava informacione potrebe multipleks-bioskopa (bioskop sa većim brojem sala).

**Rješenje.**



**Slika 1.49** IE zasnovani konceptualni model podataka multipleks-bioskopa

Multipleks-bioskop raspolaže sa više projekcionih sala, koje mogu da imaju različite kapacitete. Obično svaka sala ima određeno ime.

Budući da u multipleksu postoji više mjesta (u različitim salama) sa istim brojem reda i sjedišta, odnosno entitetski tip *MJESTO\_U\_SALI* je slabi entitetski tip (diskriminator čine atributi *Red* i *Sjedište*), a vezni tip *PRIPADA* je identifikujući vezni tip.

Slično je i sa projekcijama. Istovremeno može da se vrši više projekcija (u različitim salama), pa je odnosni entitetski tip *PROJEKCIJA*, takođe, slabi entitetski tip (diskriminator čine atributi *Datum* i *Termin*), a veze tipa *SE\_VRSI* (između entiteta tipa *SALA* i *PROJEKCIJA*) identifikujuće.

**Primjer 1.12** Projektovati konceptualni model podataka "kartomata" – uređaja za izdavanje karata na nekoj autobuskoj stanici. Kartomat treba da ima mogućnost izdavanja karata za destinacije na registrovanim linijama. Svaka linija pripada nekom prevozniku. Može da postoji više linija (različitih prevoznika) na istoj relaciji. Svaka linija ima polaznu (početnu) i dolaznu (krajnju) stanicu, a prolazi kroz određen broj usputnih stanica u različitim mjestima. Na svakoj liniji može da postoji više termina polazaka u toku dana. Cijena karte zavisi i od destinacije i od linije (različiti prevoznici mogu da imaju različite cijene karata na istoj relaciji).

### Rješenje.



Slika 1.50 IE zasnovani konceptualni model podataka kartomata

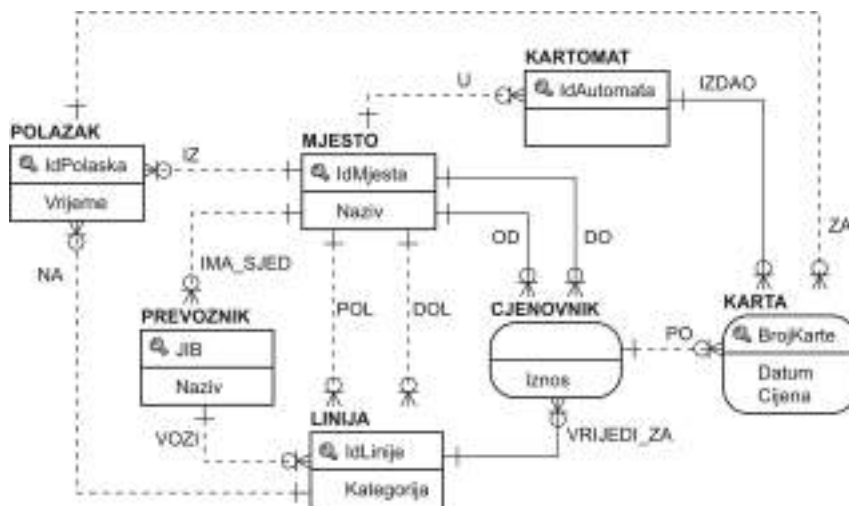
Registrovane linije reprezentovane su entitetskim tipom *LINIJA*. Atribut *Kategorija* reprezentuje kategoriju linije (međugradska, prigradska, ...). Svaka linija pripada nekom prevozniku, a isti prevoznik može da vozi na više linija. Svaki prevoznik ima sjedište u nekom mjestu, a u nekom mjestu može da postoji više različitih prevoznika.

Svaka linija ima početnu stanicu u nekom mjestu (modelovano vezama tipa *POL*) i krajnju stanicu u nekom (drugom) mjestu (modelovano vezama tipa *DOL*). Više linija može da ima istu početnu i/ili istu krajnju stanicu. Polasci na (nekoj) liniji reprezentovani su slabim entitetskim tipom *POLAZAK*. Svi polasci na istoj liniji razlikuju se u terminu polaska (atribut *Termin* je diskriminator).

Karte su reprezentovane entitetskim tipom *KARTA*. Svaka karta izdaje se za određeni polazak na nekoj liniji. Cijena karte odgovara cjenovniku za konkretnu liniju do odgovarajuće destinacije. Entitetski tip *CJENOVNIK* ima dvostruku ulogu. S jedne strane, svaki entitet tipa *CJENOVNIK* reprezentuje činjenicu da se neko mjesto nalazi na nekoj liniji, a sa druge strane, atribut *Iznos* pokazuje kolika je cijena karte do tog mjesta u odnosu na stanicu na kojoj kartomat izdaje kartu.

**Primjer 1.13** Modifikovati konceptualni model podataka iz primjera 1.12 tako da zadovoljava informacione potrebe lanca kartomata koji su postavljani u većem broju autobuskih stanica.

**Rješenje.**



**Slika 1.51** IE zasnovani konceptualni model podataka lanca kartomata

U svakom mjestu može da postoji više kartomata, koji su reprezentovani entitetskim tipom *KARTOMAT*. Svaki kartomat izdaje karte sa sopstvenom numeracijom, pa može da postoji više karata sa istim brojem (koje su izdate na različitim kartomatima). Zato je entitetski tip *KARTA* modelovan kao slabi entitetski tip, a atribut *BrojKarte* predstavlja diskriminator.

Slabi entitetski tip *CJENOVNIK* modifikovan je u odnosu na model iz primjera 1.12. Svaki entitet tipa *CJENOVNIK* i dalje ima dvostruku ulogu. Međutim, sada reprezentuje cijenu karte između neke dvije stanice na nekoj liniji. To znači da svaki entitet tipa *CJENOVNIK*, osim cijene karte, ujedno sadrži i informaciju o polaznoj i odredišnoj stanici za konkretnu vožnju za koju je izdata karta. Zato nije neophodno uvođenje veznih tipova između entitetskih tipova *KARTA* i *MJESTO* kojima bi se reprezentovale činjenice polazne i odredišne stanice za svaku izdatu kartu.

Svaka karta izdaje se za određeni *polazak* iz nekog mjesta na nekoj liniji. Odnosni entitetski tip *POLAZAK* modelovan je kao jaki entitetski tip uvođenjem surogat ključa *IdPolaska*.

## 1.4 UML dijagram klasa

Dijagram klasa je strukturni UML dijagram koji omogućava modelovanje klasa i njihovih međusobnih veza, i predstavlja jednu od najčešće korištenih notacija u objektno-orijentisanom softverskom inženjerstvu. S obzirom na intenzivan razvoj i primjenu UML-a u svim fazama životnog ciklusa softverskih sistema, dijagram klasa se često koristi i u projektovanju baza podataka, iako standardna notacija nije baš u potpunosti prilagođena za tu namjenu.

**Klasa** je deskriptor za skup entiteta (objekata) sa sličnim svojstvima, ponašanjem i vezama sa drugim entitetima. Klase imaju dvije vrste članova: attribute koji reprezentuju svojstva objekata i operacije koje su definisane nad objektima. Klase bez operacija su analogne entitetskim tipovima u MOV notaciji. Klase se prikazuju u obliku trodjelnog pravougaonika (1.52a), pri čemu gornja sekcija sadrži naziv klase, srednja attribute, a donja operacije. Kad detalji nisu neophodni, atributi i operacije ne moraju da se prikazuju (1.52b). Naziv klase tipično počinje velikim slovom (npr. *Osoba*).

Specifikacija atributa ima opšti oblik:

[*vidljivost*] *ime* [: *tip*] [*multiplikativnost*] [= *vrijednost*] [*uslovi*]

gdje elementi unutar srednjih zagrada nisu obavezni. U specifikaciji atributa jedino je obavezno ime, koje tipično počinje malim slovom (npr. *jmb*).

Članovi klase mogu da budu *javni* (*public*), *zaštićeni* (*protected*) i *privatni* (*private*). To svojstvo naziva se *vidljivost* (*visibility*) i na dijagramu klasa se prikazuje odgovarajućim simbolom, redom: +, #, -. Iz perspektive konceptualnog modelovanja ova distinkcija nije značajna.

Izvedeni atributi prikazuju se sa oznakom '/' ispred naziva atributa.

Višeznačni (*multivalued*) atributi definišu se tako što se za dati atribut specifikuje *multiplikativnost*, tj. minimalan i maksimalan broj vrijednosti u skupu kojim se reprezentuje dati atribut. Tipične multiplikativnosti su: [0..1], [1..1] ili kraće [1], [0..\*] ili kraće [\*], [1..\*], itd. Podrazumijevana multiplikativnost je [1], odnosno [1..1], i ona se tipično ne prikazuje. Multiplikativnost [\*], odnosno [0..\*], znači da dati atribut može da ima više vrijednosti, ali i da dati atribut ne mora da ima vrijednost, odnosno da može da ima *null* vrijednost. Na sl. 1.53c) atributi *adresa* i *telefon* u klasi *Osoba* su višeznačni.

Svakom atributu može da se definiše podrazumijevana vrijednost, koja se unutar definicije atributa specifikuje kao izraz čijim se izračunavanjem dobija podrazumijevana vrijednost datog atributa. Podrazumijevana vrijednost atributa *ects* u klasi *Predmet* na sl. 1.52c) iznosi 6.

Atributima mogu da se specifikuju *uslovi*, odnosno *ograničenja*. Ograničenja mogu da se odnose na dozvoljene vrijednosti koje dati atribut može da ima, ili mogu da reprezentuju neke druge specifičnosti atributa, kao što su *jedinstvenost* (*unique*) ili *uređenost* (*ordered*) skupa vrijednosti kojima se reprezentuje dati višeznačni atribut, itd. Za svaki atribut može da se definiše zaseban skup

ograničenja. Ograničenja se navode unutar velikih zagrada, a međusobno se odvajaju zaptetama, npr.  $\{ordered, unique\}$ . Atribut *ects* (sl. 1.52c) ima ograničenje po vrijednosti  $\{ects>0\}$ .

Standardni UML nema skup simbola koji omogućava da se posebno označe atributi koji čine primarni ključ. Zato neki alati koriste specijalizovanu notaciju. U literaturi se primarni ključ najčešće specifikuje korištenjem stereotipova ili  $\{PK\}$  ograničenja. Ne postoji ni standardna UML notacija za neke druge specifične koncepte koji postoje u MOV, kao što su slabi entitetski skupovi, složeni ili kompozitni atributi, itd. Međutim, postoje semantički ekvivalentni koncepti.



**Slika 1.52** UML notacija za klase: (a) sa implementacionim detaljima, (b) bez atributa i operacija, i (c) uobičajena za konceptualno modelovanje

**Asocijacija** je najopštija vrsta veze u UML-u i podrazumijeva binarnu vezu, tj. vezu između samo dvije klase. Asocijacija se prikazuje punom linijom. Svaka asocijacija ima *naziv* koji se prikazuje na sredini linije. Dodatno se pored naziva asocijacije može prikazati i mali crni trougao čija orijentacija ukazuje na redoslijed čitanja asocijacije. Asocijacija *ima* između klase *StudijskiProgram* i *Fakultet* (sl. 1.53) čita se "fakultet *ima* studijski program".

*Kraj* ili *završetak* asocijacije tipično prikazuje *ulogu* koju objekti date klase imaju u datom tipu veze, *multiplikativnost* i *karakter asocijacije*.

*Uloga* je string koji se tipično prikazuje neposredno uz odgovarajući završetak asocijacije. Uloge se tipično prikazuju kad između klase postoje različite asocijacije u kojima objekti iste klase imaju različite uloge, kod rekurzivnih asocijacija, ili kad uloga ne može direktno da se prepozna iz samog naziva asocijacije. Kada je uloga objekata u nekoj asocijaciji očigledna, uloge ne moraju da se specifikuju. Objekti klase *Nastavnik* u asocijacijama tipa *rukovodi* (sl. 1.53) imaju ulogu *dekan*, tj. nastavnik koji rukovodi fakultetom je dekan.

*Multiplikativnost* krajeva asocijacije služi za specifikaciju ograničenja učešća objekata odnosnih klase u vezama datog tipa. Tipično se jedan kraj asocijacije naziva *source*, a drugi *target*. Multiplikativnost *source* kraja specifikuje minimalan i maksimalan broj objekata *source* klase koji učestvuju u vezi datog tipa sa jednim objektom klase na *target* strani, i obrnuto, multiplikativnost *target* kraja specifikuje minimalan i maksimalan broj objekata *target* klase koji učestvuju u vezi datog tipa sa jednim objektom *source* klase (slično IE notaciji).

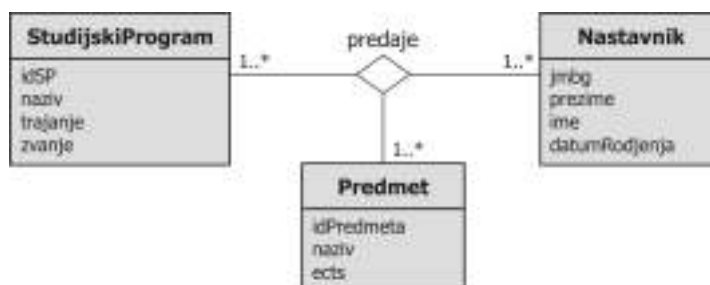
Multiplikativnosti krajeva asocijacije istovremeno specificuju i *kardinalnost mapiranja* i *totalnost učešća* objekata odnosnih klasa u vezi datog tipa (kao i u IE notaciji). S obzirom na to da se ograničenja specificuju jedinstveno, ni u dijagramu klasa nema posebne grafičke notacije za totalnu participaciju. Ako je minimalna multiplikativnost objekata neke klase nula, tada je participacija objekata druge klase parcijalna (jer postoje objekti te druge klase koji nisu u vezi ni sa jednim objektom date klase). Ako je minimalna multiplikativnost objekata jedan, tada je participacija objekata druge klase totalna (svaki objekat te druge klase je u vezi bar sa jednim objektom date klase). Svakim fakultetom (sl. 1.53) rukovodi jedan nastavnik koji ima ulogu dekana. S druge strane, nisu svi nastavnici dekani, tj. ima nastavnika koji nisu u vezama tipa *rukovodi* ni sa jednim fakultetom.



Slika 1.53 Primjeri binarnih asocijacija

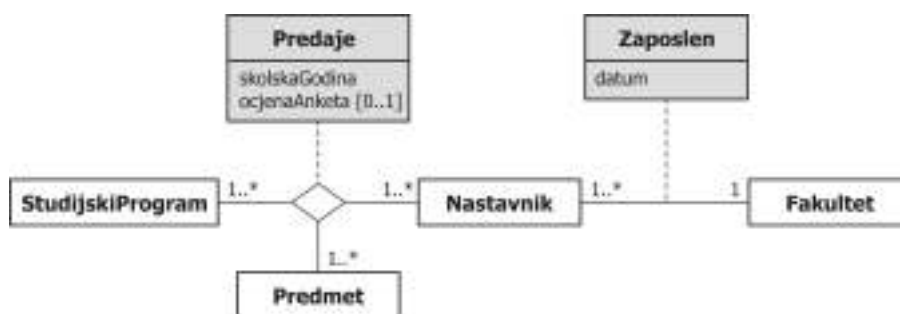
Asocijacije mogu da budu *jednosmjerne* ili *asimetrične* te *dvosmjerne* ili *simetrične*. Završeci simetrične asocijacije prikazuju se bez strelica, dok se kod asimetrične asocijacije jedan završetak predstavlja strelicom koja ukazuje na usmjerenost asocijacije. Pošto usmjerenost asocijacije nema značaj u konceptualnom projektovanju, sve asocijacije u ovoj knjizi su simetrične.

Pod ***n-arnom asocijacijom*** (*n-ary association*) podrazumijeva se istovremena veza tri ili više klasa. Slično MOV notaciji, *n-arna* asocijacija u dijagramu klasa prikazuje se kao prazni romb koji je punim linijama spojen sa odnosnim klasama. Primjer *n-arne* (ternarna) asocijacija dat je na sl. 1.54.



Slika 1.54 Primjer *n-arne* (ternarna) asocijacije

Asocijacija koja ima sopstvene attribute modeluje se kao **vezna klasa** (*association class*), koja se na dijagramu klasa prikazuje kao klasa vezana za odnosnu asocijaciju isprekidanom linijom. Tipično se takva asocijacija ne imenuje, jer vezna klasa ima odgovarajući naziv. Primjeri veznih klasa dati su na sl. 1.55.



Slika 1.55 Primjeri veznih klasa

**Agregacija** je specifična asocijacija između dvije klase kojom se modeluje veza tipa *dio-cjelina*, pri čemu *dio* predstavlja entitet sa vlastitim identitetom, što mu omogućava da egzistira nezavisno od *cjeline* (*cjelina* se obično naziva i *agregat*). Ovakav koncept agregacije razlikuje se od koncepta agregacije u MOV modelu. Na dijagramu klasa agregacija se prikazuje kao asocijacija čiji završetak na strani cjeline ima simbol praznog romba. Nezavisnost od cjeline omogućava dijelovima da istovremeno participiraju u više različitih agregacija. Tako svaki nastavnik pripada nekoj katedri, ali može i da pripada različitim projektnim timovima (sl. 1.56).



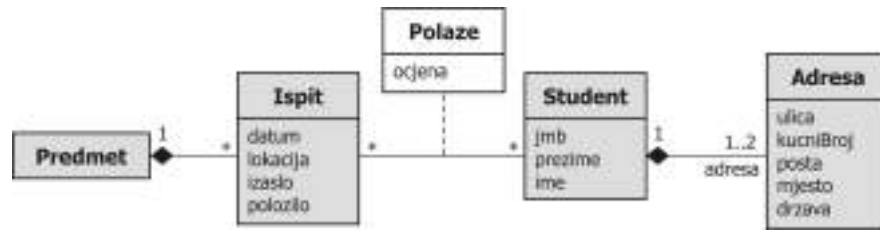
Slika 1.56 Primjeri agregacije

**Kompozicija** je stroži tip asocijacije u odnosu na agregaciju, u kojoj *dio* (obično se naziva *komponenta*) mora da pripada *cjelini* i ne može da egzistira nezavisno od *cjeline*. Na dijagramu klasa kompozicija se prikazuje kao asocijacija čiji završetak na strani cjeline ima simbol punog romba (sl. 1.57).

S obzirom na to da kompozicija (iako se radi o drugačijem konceptu) omogućava modelovanje egzistencijalne zavisnosti, veza jakog i slabog entitetskog skupa, odnosno odgovarajućih klasa, može da se modeluje kao kompozicija. Tako je na sl. 1.57 modelovana egzistencijalna veza objekata tipa *Predmet* i *Ispit*.

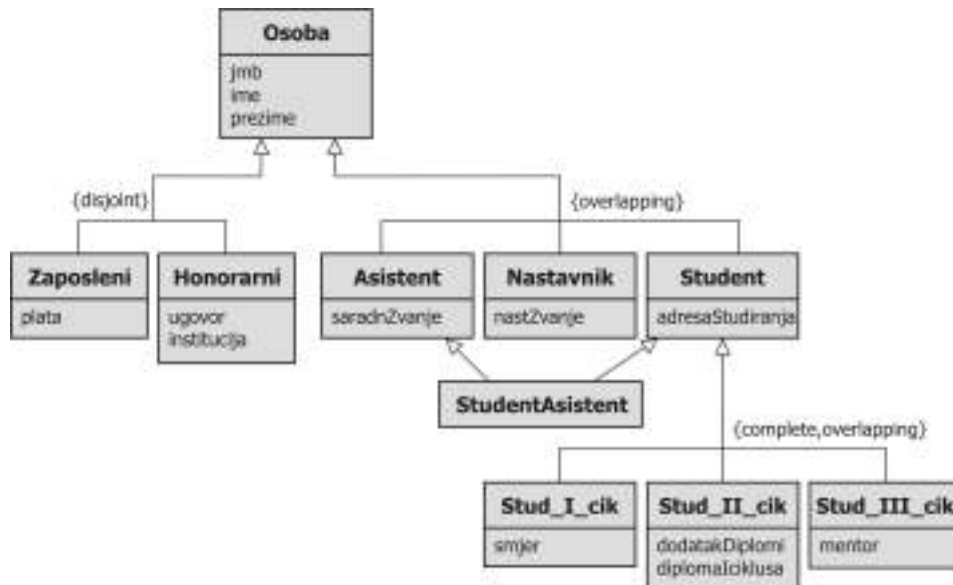
Kompozicija omogućava modelovanje i kompozitnih (složenih) atributa. Naime, atributi neke klase ne mogu direktno da se označe kao kompozitni (kao što je to bilo moguće u MOV), nego se data klasa povezuje kompozicijom sa klasom koja reprezentuje tip kompozitnog atributa. Tako klasa *Student* (sl. 1.57) ima kompozitni atribut *adresa*, pri čemu je taj atribut i višeznačan (jer ima multiplikativnost 1..2).





Slika 1.57 Primjeri kompozicije

**Nasljeđivanje** podrazumijeva da neka klasa nasljeđuje (preuzima) sadržaj jedne ili više drugih klasa uz mogućnost modifikacije preuzetih i dodavanja novih članova. Nasljeđivanje omogućava modelovanje veza generalizacije i specijalizacije. Nasljeđivanje se modeluje strelicom sa praznim trougaonim vrhom, usmjerenom od *nasljednika*, odnosno od *potklase*, prema *pretku*, odnosno *natklasi*. Ne postoji posebna notacija kojom se naglašava razlika između *disjunktne* i *preklapajuće* specijalizacije, pa se karakter specijalizacije specifikuje odgovarajućim ograničenjem, {*disjoint*} ili {*overlapping*}. Ne postoji ni posebna notacija za specifikaciju totalnosti specijalizacije, pa se koristi odgovarajuće ograničenje, {*complete*} ili {*incomplete*}. Primjeri veza generalizacije i specijalizacije dati su na sl. 1.58.



Slika 1.58 Primjeri veza generalizacije i specijalizacije

Primjena dijagrama klasa ilustrovana je u primjeru 1.14, u kojem je prikazan konceptualni model podataka univerziteta, koji je semantički ekvivalentan MOV dijagramu iz primjera 1.1 i IE dijagramu iz primjera 1.3.

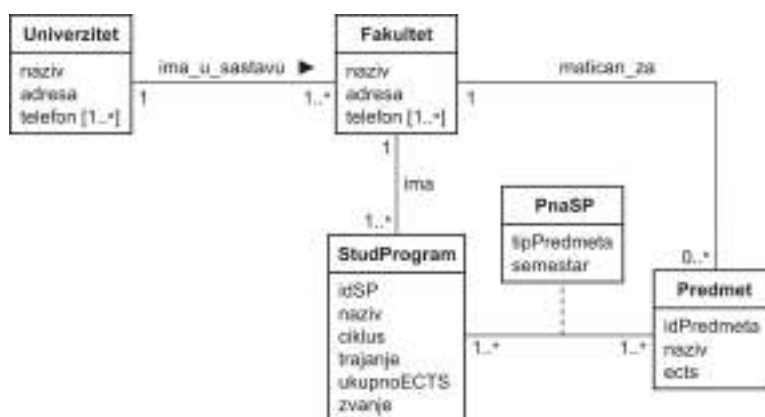
**Primjer 1.14** Predstaviti konceptualni model podataka univerziteta UML dijagramom klasa.

### Rješenje.

Tekstualna specifikacija informacionih potreba univerziteta i odgovarajući MOV dijagram dati su u primjeru 1.1. Ovdje ćemo samo za svaki od dijelova specifikacije dati ekvivalentni dijagram klasa uz odgovarajuća pojašnjenja.

#### *Dijagram klasa za prvi dio specifikacije*

Dijagram klasa koji odgovara prvom dijelu tekstualne specifikacije i koji je ekvivalentan MOV dijagramu sa sl. 1.6, prikazan je na sl. 1.59.



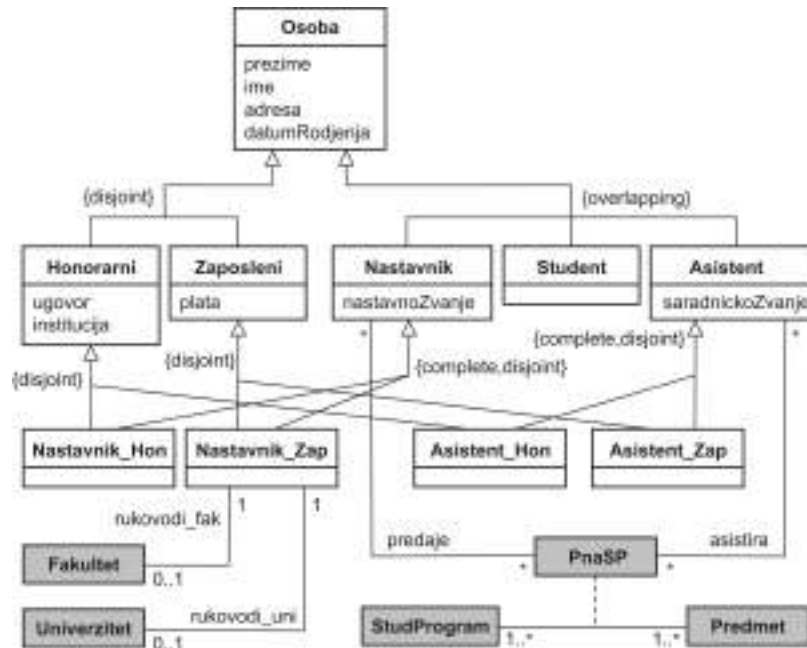
**Slika 1.59** Dijagram klasa koji odgovara prvom dijelu specifikacije informacionih potreba univerziteta iz primjera 1.1 i koji je ekvivalentan MOV dijagramu sa sl. 1.6 te IE dijagramu sa sl. 1.38

Atribut *telefon* (i u klasi *Univerzitet* i u klasi *Fakultet*) je višeznačan. Multiplikativnost je 1..\*, što znači da univerzitet i svaki fakultet imaju bar po jedan broj telefona.

S obzirom na to da vezni tip *P<sub>NA</sub>SP* (iz modela sa sl. 1.6) ima opisne attribute, asocijaciji između klasa *StudProgram* i *Predmet* pridružena je vezna klasa *PnaSP* sa korespondentnim atributima *tipPredmeta* i *semestar*. Multiplikativnosti oba kraja date asocijacije su 1..\*, jer svaki studijski program mora da ima definisane predmete, a svaki predmet mora da pripada bar jednom studijskom programu. Multiplikativnosti završetaka preostalih asocijacija identične su kao u IE dijagramu sa sl. 1.38 (IE dijagram ekvivalentan MOV dijagramu sa sl. 1.6).

#### *Dijagram klasa za drugi dio specifikacije*

Dijagram klasa koji odgovara drugom dijelu tekstualne specifikacije i koji je ekvivalentan MOV dijagramu sa sl. 1.13, prikazan je na sl. 1.60.



**Slika 1.60** Dijagram klasa koji odgovara drugom dijelu specifikacije informacionih potreba univerziteta iz primjera 1.1 i koji je ekvivalentan MOV dijagramu sa sl. 1.13

Vezna klasa daje odnosnoj asocijaciji karakteristike klase, što je ekvivalentno konceptu agregacije u MOV (vezni tip tretira se kao entitetski tip višeg nivoa). Svaka instanca vezne klase *PnaSP* reprezentuje konkretan predmet koji pripada nekom konkretnom studijskom programu, a nastavnici i asistenti upravo imaju zaduženja na takvim predmetima.

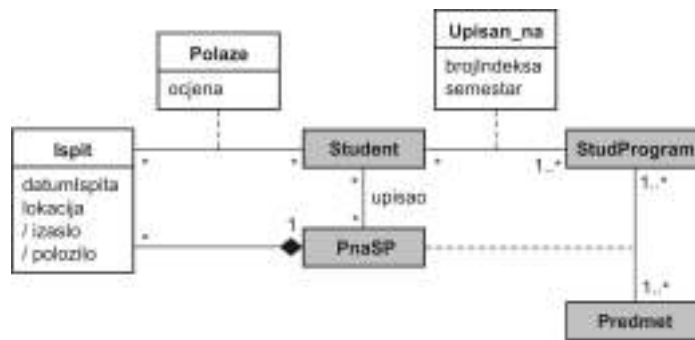
Multiplikativnosti završetaka asocijacija *predaje* i *asistira* direktno slijede iz MOV dijagrama sa sl. 1.13. S obzirom na to da se multiplikativnosti završetaka asocijacija u dijagramu klasa specifikuju analogno veznim tipovima u IE notaciji, izostavljamo detaljan opis asocijacija *rukovodi\_fak* i *rukovodi\_uni* (vidjeti IE dijagram sa sl. 1.40, koji je semantički ekvivalentan MOV dijagramu sa sl. 1.13).

### ***Dijagram klasa za treći dio specifikacije***

Dijagram klasa koji odgovara trećem dijelu tekstualne specifikacije i koji je ekvivalentan MOV dijagramu sa sl. 1.18b), prikazan je na sl. 1.61.

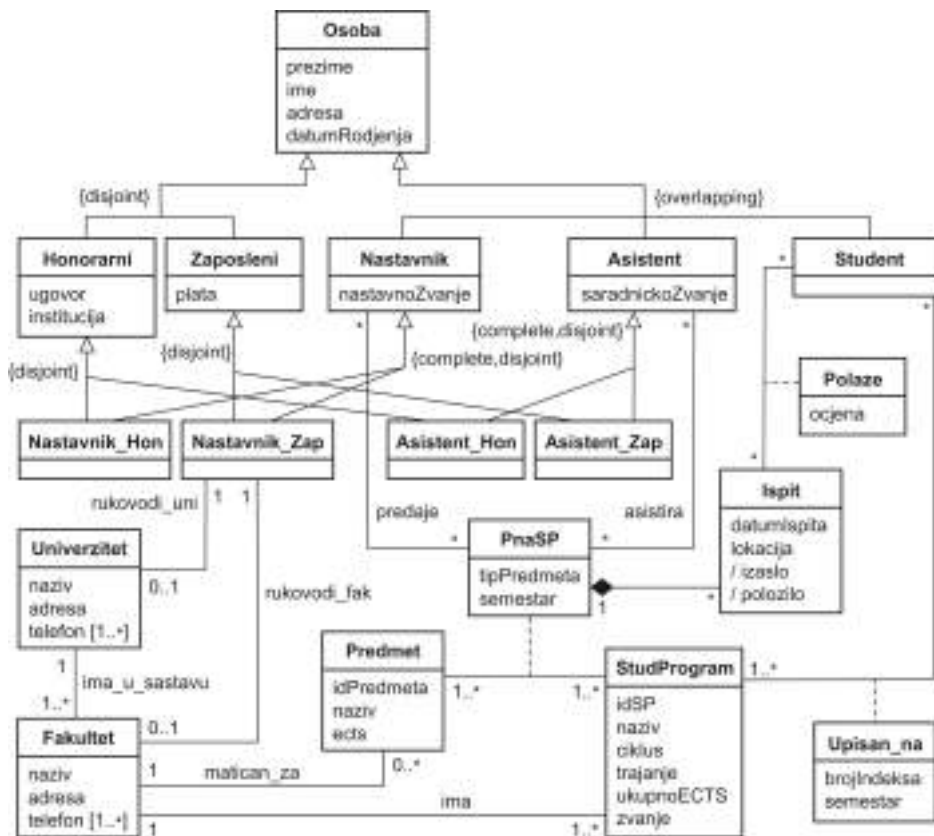
Slabi entitetski tip *ISPIT* (iz MOV dijagrama sa sl. 1.18b) reprezentovan je odgovarajućom klasom *Ispit*, a identifikujući vezni tip *I\_P* odgovarajućom kompozicijom (budući da ispit egzistencijalno zavisi od konkretnog predmeta na nekom studijskom programu).

Vezni tipovi *POLAŽE* i *UPISAN\_NA* imaju opisne attribute, pa su reprezentovani odgovarajućim asocijacijama sa pridruženim veznim klasama.



**Slika 1.61** Dijagram klasa koji odgovara trećem dijelu specifikacije informacionih potreba univerziteta iz primjera 1.1 i koji je ekvivalentan MOV dijagramu sa sl. 1.18b)

Na sl. 1.62 prikazan je cjelokupni dijagram klasa koji reprezentuje konceptualni model podataka univerziteta u skladu sa specifikovanim informacionim potrebama i koji je ekvivalentan MOV dijagramu sa sl. 1.19.



**Slika 1.62** Dijagram klasa koji reprezentuje uprošteni konceptualni model podataka univerziteta