

Notacije za modeliranje podataka

Mataja, Ema

Undergraduate thesis / Završni rad

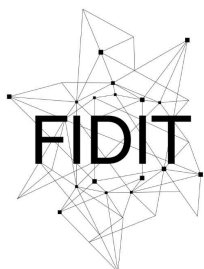
2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka / Sveučilište u Rijeci**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:195:688304>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-04-20**



Sveučilište u Rijeci
**Fakultet informatike
i digitalnih tehnologija**

Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository](#)



zir.nsk.hr

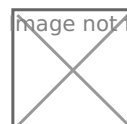


Image not found or type unknown

Sveučilište u Rijeci – Odjel za informatiku

Preddiplomski jednopredmetni studij Informatike

Ema Mataja

Notacije za modeliranje podataka

Završni rad

Mentorica: Doc. dr. sc. Martina Ašenbrener Katić

Rijeka, Rujan 2020

Rijeka, 01.06.2020.

Zadatak za završni rad

Pristupnik: Ema Mataja

Naziv završnog rada: Notacije za modeliranje podataka

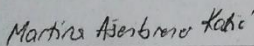
Naziv završnog rada na eng. jeziku: Notations for data modeling

Sadržaj zadatka:

U završnom radu potrebno je detaljno opisati nekoliko odabranih notacija za modeliranje podataka. Za svaku od notacija potrebno je definirati primjer te na kraju usporediti opisane notacije.

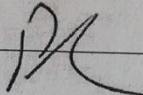
Mentor

Doc. dr. sc. Martina Ašenbrener Katić



Voditelj za završne radove

Doc. dr. sc. Miran Pobar



Zadatak preuzet: 10.06.2020.



(Ema Mataja)

Sadržaj

Sažetak	4
Ključne riječi	5
Kazalo	6
Uvod	7
1 Metoda entiteta – veze	8
1.1 Koncepti strukture metode entiteta – veza	8
1.1.1 Model podataka za dokument Račun napravljen EV notacijom	12
2 Crow's Foot notacija	15
2.1 Koncepti modela	15
2.2 Model podataka za dokument Račun napravljen Crow's Foot notacijom	17
3 Barkerova notacija	18
3.1 Koncepti modela	18
3.2 Model podataka za dokument Račun napravljen Barkerovom notacijom	19
4 IDEF1X notacija	21
4.1 Koncepti metode IDEF1X	21
4.2 Model podataka za dokument Račun napravljen IDEF1X notacijom	24
5 Unified Modeling Language (UML)	26
5.1 UML dijagrami	26
5.1.1 Dijagram klase	26
5.2 Dijagram klase za dokument Račun	28
6 Usporedba notacija	30
7 Zaključak	32
Popis slika	33
Literatura	34

Sažetak

U ovom radu razrađene su različite notacije za modeliranje podataka. Opisane su najvažnije notacije, istaknute su njihove definicije te najvažniji pojmovi koji se odnose na svaku od njih. Za svaku od notacija navedeni su primjeri koji su detaljno opisani i razrađeni. Notacije odabrane i obrađene u ovom radu su: Entiteti veze (metodologije MIRIS), Crow's foot ili takozvana IE notacija, Barkerova notacija, IDEF1x notacija (metodologije IDEF) te Dijagram klasa (iz UML). Za istaknute primjere notacija dana je i detaljna usporedba istih.

Ključne riječi

Notacija

Metoda Entiteti veze

Crow's Foot metoda

Barkerova metoda

IDEF1X metoda

UML

Dijagram klase

Entitet

Veze

Atribut

Kazalo

A

Agregacija, 11
atribut, 8, 15, 18
Atribut, 9

B

Barker, 18

C

CASE, 15, 18
Chen, 8
Crow's Foot, 15, 18

D

Dijagram klase, 26

E

Egzistencijalni i identifikacijski tip veze, 11
Egzistencijalni tip veze, 10
Entitet, 4, 8
entiteti – veza, 8
EV, 21, 30
Entiteti -veze, 8, 18

I

identifikacijske veze, 24
Identifikacijski tip veze, 10

K

kardinalnost, 15

M

MIRIS, 4
model, 8, 15

N

nadtip, 12
NORMA, 30
notacija, 4, 15, 18

O

ORACLE, 15, 18
ORM, 30

P

podtip, 12
Primarni ključ, 9, 15

U

UML, 18

V

Veza, 10
vidljivosti, 26

Uvod

Razvijajući informatiku kao nauku razvija se i njena tehnologija. Tehnologija se razvija da bi nam u mnogočemu pomogla olakšavajući nam razne procese. Prilikom izrade raznih poslovnih procesa, raznih dijagrama tražile su se učinkovitije metode. Za razvoj informacijskog sustava potrebno je prikupiti potrebne dokumente te izraditi pripadajući model. Pokušavajući smanjiti dokumentaciju kako bi se pojednostavio proces izrade raznih modela, a uz to povećala djelotvornost izrade modela, uvode se notacije. Notacije za modeliranje podataka su grafički prikazi za specificiranje i izradu podatkovnih modela za informacijski sustav primjenom određenih formalnih tehnika (Wikipedia BPM, 2020). Notacije za modeliranje podataka su jednostavne i efikasne te su olakšale izradu raznih modela. Neke od važnijih vrsta notacija koje su pokrenule tu efikasnost i pospješile jednostavnost izrade te se svakodnevno nadograđuju za modeliranje podataka su: metoda entitet – veze kao dio metodologije MIRIS, Crow's Foot notacija, Barkerova notacija, IDEF1X notacija kao dio metodologije IDEF te Dijagram klasa (iz UML–a). Iste će detaljno biti obrazložene u nastavku. Poznate notacije koriste se i za izradu modela koje su prisutne i u CASE alatima. Computer – aided softwer enineering (CASE) domena je softverskih alata koji se koriste za dizajn i implementaciju aplikacija (Wikipedia CASE, 2020). Tako je Object Management Group (OMG) objedinjeni jezik modeliranja UML prihvaćen kao standard za objektno orijentirano modeliranje i razvili set alata CASE (Wikipedia OMG, 2020).

1 Metoda entiteti – veze

Metoda entiteti – veze poznata pod imenom model entiteti – veze (engl. Entity – relationship model, skraćeno EV) je prvi put objavljena u Chenovu članku (Chen 1976.). Metoda EV je jedna od prvih semantički bogatijih metoda za modeliranje podataka (Pavlič, 2011).

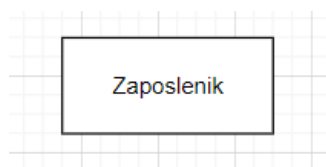
Metoda EV prikazuje odnos međusobno povezanih stvari ili grupa podataka u tom sustavu koji se odnose na određenu domenu. Koncept te metode napravljen je da bude blizak i jednostavan za razumijevanje prilikom komunikacije između projektanta i korisnika. EV je postala metoda koja se najčešće koristila te se u međuvremenu i izmjenjivala i nadopunjavala kao dio MIRIS metodologije. Metodologija MIRIS (Metodologija za Razvoj Informacijskih Sustava) je skup metoda čiji je cilj projektirati i izgraditi informacijski sustav (IS). Metodologiju MIRIS razvio je prof. dr. sc. Mile Pavlič, a objavljena je 1995. godine (Pavlič, 2009).

Pomoću EV metode dalje se izgrađuje informacijski sustav uz pomoć dijagrama odnosno DEV-a (dijagram entiteta - veze). Obično se gradi pomoću informacijskog sustava koji se sastoji od najmanje dva dijela. Tu dolazimo do koncepta strukture te metode koje će biti opisane u sljedećem poglavlju.

1.1 Koncepti strukture metode entiteti – veza

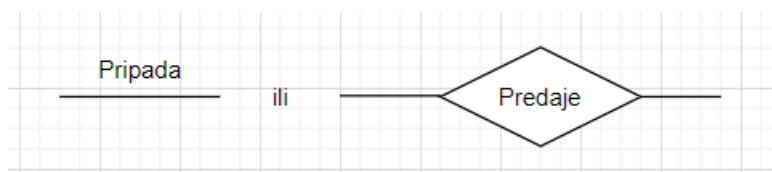
Osnovni koncepti metode entiteti – veza od kojih se gradi struktura modela entiteti – veze su: entitet i tip entiteta, veza i tip veze, atribut tipa entiteta, slab tip entiteta, specijalni tipovi veza, agregirani tip entiteta, povratni tip veze te generalizacijski tip veze (Pavlič, 2011).

Entitet su imenice i njima se imenuju stvari ili neka postojanja ili pojam, nešto što možemo imenovati. Za primjer imamo entitet Nastavnik. Kako bi pojednostavili prikaz modela možemo klasificirati te entitete odnosno razvrstati ih u tipove entiteta sa sličnim karakteristikama. Tip entiteta je skup pojedinačnih entiteta dobiven procesom klasifikacije apstrakcije (Pavlič, 2011). Odnosno svrstavajući entitete istog tipa u skupinu, dobivamo tipove entiteta, npr. Nastavnik i Ravnatelj, njih možemo svrstati u tip entiteta Zaposlenik.



Slika 1 Tip entiteta sa nazivom

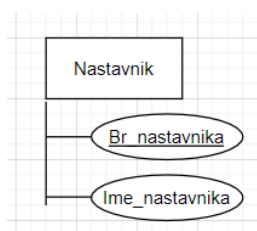
Prilikom nastanka modela entiteti - veza potrebno je povezati tipove entiteta i odrediti vezu između njih. Relacijama odnosno vezama postićemo povezanost. Vezu možemo definirati kao pridruživanje između tipova entiteta. Prilikom odabira entiteta iz pojedinih tipova entiteta postićemo uređeni par dvaju entiteta koja ostvaruje pojedinu vezu pojedinih entiteta (Pavlič, 2011). Poput klasifikacije entiteta postoji klasifikacija veze. Kada veze između istih entiteta sjedinimo u skup veza dobijemo tip veze. Primjer oznake tipova veze prikazane su na slici 2.



Slika 2 Simbol tipa veze

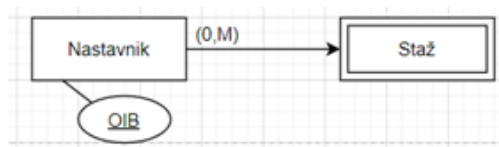
Svaki tip entiteta sadrži neka svojstva koja ga opisuju i nazivaju se atributi. Atribut dodjeljuje tipu entiteta vrijednost. Atribut može biti jednostavni ili složeni. Jednostavni atribut je prikazan na slici 3 Ime_nastavnika gdje se on definira kao svojstvo entiteta. Češće se upotrebljava jednostavan atribut upravo zbog jednostavnosti da je lakše razumjeti model. Nasuprot jednostavnom, složeni atribut se može podijeliti na više atributa koji gledamo kao jedan novi. Složeni atribut se još naziva i grupni atribut jer spaja više atributa u jedan (Pavlič, 2011). Primjerice atribut Naziv nastavnika se može podijeliti na attribute Ime i Prezime. Atribut i ne mora imati vrijednost odnosno može imati nul vrijednost. Nul vrijednost predstavlja neodređenu vrijednost atributa nekog elementa (Pavlič, 2011). Takva vrijednost iako postoji nije dobrodošla u modelu upravo zbog svog svojstva koja nije primjenjiva u modelu. Na primjeru tipa entiteta Telefon s atributom Telefonski_broj, vrijednost tog atributa ne mora biti poznata jer nema telefonski broj, ali se to kasnije može izmijeniti ukoliko isti nabavi.

Uz navedene koncepte imamo i ključeve. Svaki tip entiteta ima svoj ključ koji ga definira. Primarni ključ ili glavni ključ odabiremo sa nekakvim značajem da može zamijeniti taj tip entiteta. Svaki tip entiteta, u ovom slučaju Nastavnik ima svoj atribut Br_nastavnika pod kojim se isti identificira. Pritom taj atribut Br_nastavnika kao tip entiteta Nastavnik izabiremo za primarni ključ. Svaki ključ osim primarnog ima uvjete koje mora zadovoljiti; uvjet jedinstvenosti i uvjet neredundantnosti. Uvjet jedinstvenosti nam kazuje da ne postoje dva pojedinačna pojavljivanja entiteta u tipu entiteta takva da imaju istu vrijednost atributa koja čine ključ. To znači da ne postoje dva tipa entiteta koji imaju isti skup atributa za ključ (Pavlič, 2011). Uvjet neredundantnosti nam ukazuje, ukoliko se iz ključa izostavi bilo koji od atributa, gornja osobina se gubi (Pavlič, 2011). Sažeto, svaki tip entiteta se mora razlikovati po vrijednosti atributa i po skupu atributa za ključ te je bitno da se ne izgubi niti jedna osobina tog atributa, npr. tip entitet Osoba ima svoje attribute OIB i Matični broj.



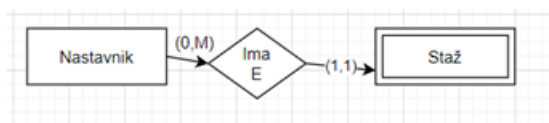
Slika 3 Primjer tipa entiteta Nastavnik sa atributima i pripadajućim primarnim ključem

Osim definiranja tipova entiteta postoje i vrste tipova entiteta. Kao vrsta entiteta postoje jak tip entiteta i slab tip entiteta. Jaki tip entiteta ne ovisi o drugim tipovima entiteta i ima svoj primarni ključ koji ga definira. Na slici je prikazano da tip entiteta Nastavnik ima svoj primarni ključ OIB i on ne ovisi o drugim entitetima. Nasuprot jakom tipu entiteta, slab tip entiteta je tip entiteta koji na neki način ovisi o drugom tipu entiteta (Pavlič, 2011). Slabi tip entiteta je tip entiteta Staž, koji ovisi o godinama nastavnikovog rada što imamo na slici 4.



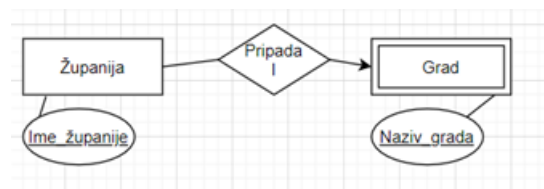
Slika 4 Primjer egzistencijale ovisnosti Staža o Nastavniku

Ovisnost slabog tipa entiteta možemo i prikazati vezom specijalnog tipa. Specijalan tip veze prikazuje povezanost jakog i slabog tipa entiteta i ta povezanost je nazvana egzistencijalnom vezom. Egzistencijalni tip veze (E) je tip veze koji povezuje dva tipa entiteta od kojih postojanje jednog tipa entiteta ovisi o drugome. Tip entiteta koji egzistencijalno ovisi o jakom tipu entiteta je slabi tip entiteta (Pavlić, 2011). Pa tako jaki tip entiteta Nastavnik ima slabi tip entiteta Staž i povezuje ih relacija Ima koja je definira kao egzistencijalna veza što je i prikazano na slici 5.



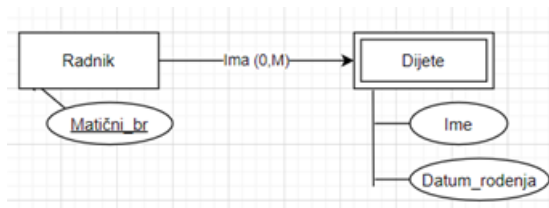
Slika 5 Primjer egzistencijalne veze

Za razliku od egzistencijalnog tipa veze, gdje svaki od tipova entiteta sadrži svoj ključ, identifikacijski tip veze (I) označava da se jedan tip entiteta u modelu ne može jedinstveno identificirati vlastitim ključem. Umjesto vlastitog ključa koristi se identifikator nekog drugog tipa entiteta da bi se postigla jedinstvena identifikacija. Na primjer: tip veze Pripada povezuje slabi tip entiteta Grad (s identifikatorom Naziv grada) i tip entiteta Županija (s identifikatorom Ime Županije). Veza Pripada je identifikacijski tip veze jer se Gradovi ne mogu jedinstveno identificirati samo s ključem Naziv grada. (Pavlić, 2011) . Identifikacijski tip veze prikazan je na slici 6.



Slika 6 Primjer identifikacijskog slabog tipa entiteta (Pavlić, 2011)

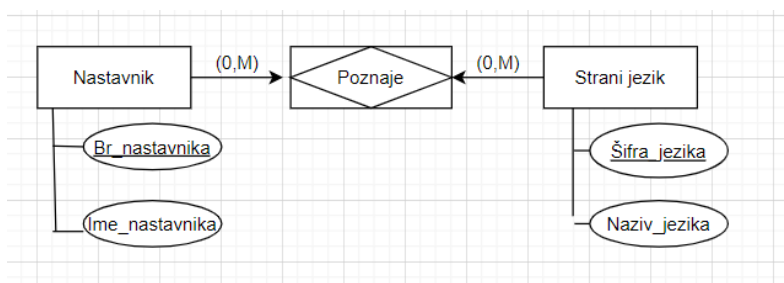
Uz egzistencijalnu i identifikacijsku vezu postoji i egzistencijalni i identifikacijski tip veze (E&I) kod kojeg je jedan tip entiteta egzistencijalno i identifikacijski zavisao od drugom tipu entiteta. Na primjer, tip entiteta Dijete je egzistencijalno i identifikacijski ovisan o tipu entiteta Radnik (Pavlić, 2011). Egzistencijalno i identifikacijski tip veze imamo prikazano na slici 7.



Slika 7 Primjer slabog entiteta koji može biti istovremeno i ovisan i identificiran drugim tipom entiteta (Pavlić, 2011)

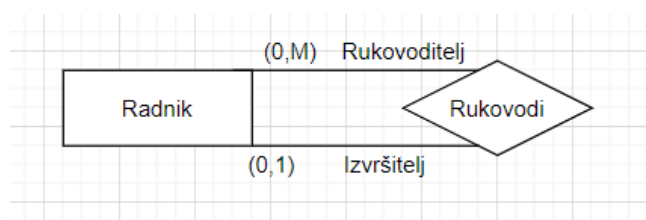
Uz relacije koje povezuju tipove entiteta navesti ćemo i brojnost tipa veze. Tip veze sastoji se od preslikavanja između tipova entiteta. Ta preslikavanja ograničavamo, te dolazimo do pojma brojnosti preslikavanja. Brojnost tipa veze je broj koji ujedno upisujemo na dijagram EV koji nam govori koliko entiteta pojedinog tipa entiteta sudjeluje u tipu veze s entitetom iz drugog tipa entiteta (Pavlič, 2011). Brojnost tipa veze imamo prikazano i na primjeru slike 5 gdje tu brojnost detaljnije i preciznije čitamo, jedan nastavnik iz tipa entiteta Nastavnik ima najmanje nula i najviše mnogo staža iz tipa entitet Staž.

Agregacija (mješoviti, agregirani tip entiteta) je apstrakcija u kojoj se tip veze između dva ili više tipova entiteta tretira kao novi tip entiteta (Pavlič, 2011). U jednom od slučajeva koji ćemo i spomenuti dolazi do agregacije kad su brojnosti sa obje strane (0,M) ili (1,M). Tako tipove entiteta Nastavnik i Strani jezik povezuje agregacija Poznaje kao što je prikazano na slici 8. Jedan entitet prvog tipa entitet pridružuje se najmanje nula i najviše većem broju entiteta drugog tipa entiteta. Detaljnije i preciznije jedan nastavnik iz tipa entiteta Nastavnik poznaje najmanje nula i najviše mnogo stranih jezika iz tipa entiteta Strani jezik. Obrnuto, jedan strani jezik iz tipa entiteta Strani jezik poznaje najmanje nula i najviše mnogo nastavnika iz tipa entiteta nastavnik



Slika 8 Primjer tipova entiteta koje povezuje agregacija sa priadajućim atributima i ključevima

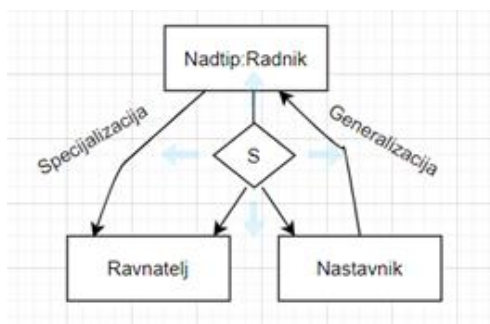
Pored relacija koje spajaju dva različita tipa entiteta postoji i povratni tip veze. Povratni (rekurzivni) tip veze je tip veze koji povezuje neki tip entiteta sam sa sobom. Na primjer, jedno pojavljivanje radnika na višoj razini hijerarhije (dakle, šefa) je povezano s više pojavljivanja entiteta na nižoj razini (Radnik), dok je pojavljivanje entiteta na nižoj razini (Radnik) u vezi samo s jednim pojavljivanjem na višoj razini hijerarhije. Dakle jedna osoba može biti šef više radnika, dok jedan radnik može imati najviše jednog šefa (Pavlič, 2011).



Slika 9 Tip entiteta Radnik s povratnom vezom Rukovodi. (Pavlič, 2011)

Za vrijeme povezivanja tipova entiteta relacijama, tip entiteta možemo još ili generalizirati ili specijalizirati. Prilikom generalizacije i prilikom specijalizacije također dobivamo veze. Neki tipovi entiteta E1, E2, ... su podtipovi nekog tipa entiteta E, kojeg zovemo nadtip ako su sva pojavljivanja entiteta podtipova ujedno i pojavljivanja entiteta nad-tipa. Nad-tip i podtipovi su povezani generalizacijskim tipom veze. Podtipovi i nadtipovi se razlikuju u atributima. Tip veze je generalizacijski ako povezuje jedan ili više entiteta podtipova sa jednim entitetom nadtipom. Svi atributi nadtip su i atributi podtipa, dakle podtip nasljeđuje entitete nadtipa. Tip generalizacijske veze se predstavlja

rombom sa nazivom S od specijalizacija ili G od generalizacija (Pavlič, 2011). U navedenom modelu prikazan je primjer generalizacije tipa entiteta Radnik. Prikazuje nam Slika 10 da se tip entiteta Radnik specijalizira ili za tipove entiteta Ravnatelj ili tip entiteta Nastavnik.



Slika 10 Primjer generalizacije i specijalizacije

1.1 Model podataka za dokument Račun napravljen EV notacijom

Napravljen je dijagram na temelju dokumenta račun koji je na slici 11 gdje su prikazani glavni elementi navedene notacije. Račun je poslovna isprava koju prodavač ispostavlja kupcu za proizvode koji su prodani ili izvršene usluge. Jedne od bitnih stavaka koje se moraju prikazati na računu su: naziv proizvoda, količina kupljenog proizvoda, cijena proizvoda i za kraj ukupan iznos kupljenog. Kao temelj svih dijagrama je dokument račun.



d.o.o. za zastupanje i posredovanje u prometu roba i usluga - Trading Company
Kvarnerska cesta 35 - 51211 Matulji - Hrvatska/Croatia - tel. +385 51 277557 - fax +385 51 277556
Prodajni centri: Rijeka - Matulji, Rijeka - Kastav, Zagreb - Sesvete, Pula, Split
PC Rijeka: Rubeši 54, Kastav, tel. +385 51 225 051, fax. +385 51 225 053

Račun - Invoice Rač21-00040/12

Datum izdavanja - Issue date: 10.01.2012. 0:00
Mjesto izdavanja - Issued at: Rubeši
Datum plaćanja - Date of payment: 24.02.2012.
Datum isporuke - Date of delivery: 10.01.2012.

Način plaćanja - Payment method: Transakcijski račun
Uvjeti plaćanja - Payment terms: Virman
Veza - Ref. Izd21-00042/12 od 10.01.2012.
Bank: Erste&Steiermarkische bank, Jadranski trg 3a, 51000 Rijeka
IBAN: HR6524020061100074306; BIC/SWIFT: ESBCHR22

AKVASET INŽENJERING D.O.O.

DRAŽIČKA 34
51000 RIJEKA
HRVATSKA

OIB: 38743179852

R br. Pos.	Šifra Code	Naziv artikla Item Description	JM U.M.	Količina Quantity	Jed cijena Unit price Kn	Rabat Rebate %	Netto cijena Net price Kn	PDV %	Ukup Total Kn
1.	00105637	LIV S-371 VODOKOTLIČ VISION	KOM	1,000	129,40	0,00	129,40	23	179,16

Prodajna vrijednost:	Kn	129,40
Subtotal:		
Iznos rabata:	Kn	0,00
Rebate value:		
Iznos popusta (0 %):	Kn	0,00
Discount total (0 %):		
OSNOVICA:	Kn	129,40
Amount before VAT:		
Iznos poreza:	Kn	29,76
VAT amount:		
IZNOS RAČUNA:	Kn	159,16
Invoice total:		

Molimo Vas da račun platite na IBAN: HR42 2330 0031 1002 1087 4

Vrijednost računa prema srednjem tečaju HNB na dan izdavanja robe iznosi 21,00 EUR.

Ukoliko dođe do porasta tečaja eura prema kuni više od 2% u odnosu na tečaj koji vrijedi na dan izdavanja računa, Termag si pridržava pravo zaračunavanja tečajnih razlika prema srednjem tečaju Hrvatske narodne banke na dan plaćanja kupca.
U slučaju neplaćanja u roku, zaračunavamo zakonske zatezne kamate. U slučaju spora, utvrđuje se nadležnost Trgovačkog suda u Rijeci.

Osoba odgovorna za izdavanje računa - Document issued by:
Udović, PC Rubeši

Paritet - Del. terms:	Bruto tež. - Gross Weight: 0,00 (kg)	Netto tež. - Net Weight: 0,00 (kg)	Broj koleta - Parcels Nr. 0
Stopa VAT %:	Osnovica Value before VAT:	Porez VAT amount:	Osnov oslobođenja PDV-a: VAT exemption basis:
23 %	129,40	29,76	
TOTAL:	129,40	29,76	Iznos računa Invoice total: 159,16 Kn

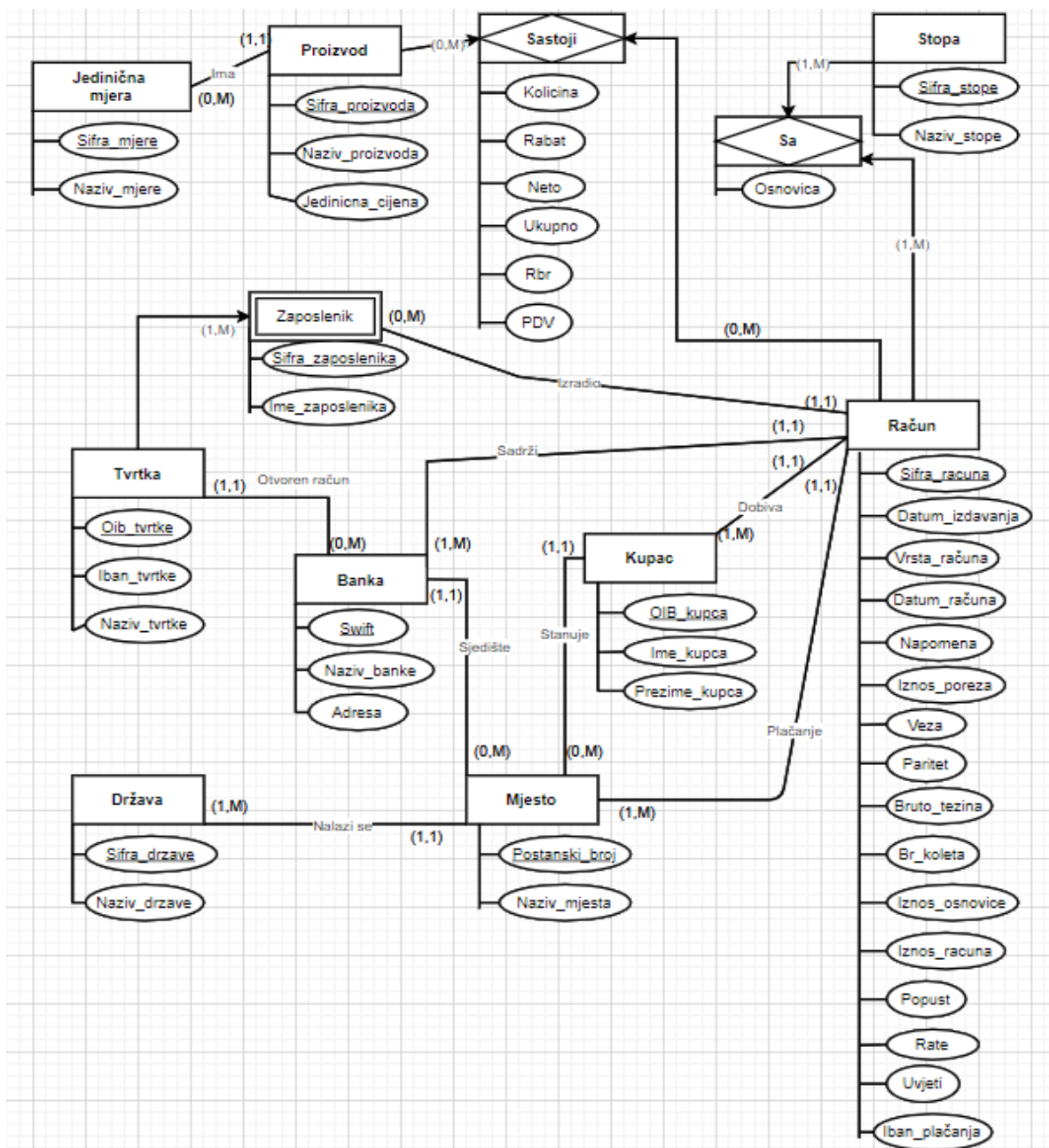
Splitška Banka d.d. - IBAN HR4223300031100210874/SWIFT: SOGEHR22
Erste & Steiermarkische Bank Rijeka - IBAN HR6524020061100074306/SWIFT: ESBCHR22
Zagrebačka Banka d.d. - IBAN HR2723600001102071592/SWIFT: ZABAHR2X
Privredna Banka d.d. - IBAN HR782340009110165959/SWIFT: PBZGHR2X

OIB: 23906639896
VAT nr. HR23906639896
www.termag.hr

Trgovački sud u Rijeci, MBS 040049266 - Temeljni kapital 24.293.200,00 kn uplaćen u cijelosti - Član uprave: Miljenko Lukanić - MB 3464814

Slika 11 Dokument račun za tvrtku Termag

Kao tipove entitete navedeni su: Račun, Proizvod, Jedinичna mjera, Kupac itd. s atributima i ključevima. Ključ Šifra_računa je primarni ključ tipa entiteta Račun. „Sastoji“ i „Sa“ predstavljaju agregacije navedenog modela koji povezuju određene tipove entiteta. Tip entitet Zaposlenik je slab tip entiteta koji ovisi o tipu entiteta Tvrtka. Tipove entitete i agregacije povezuju veze koje mogu i ne moraju imati istaknuti naziv. Kao primjer relacije koja spaja dva tipa entiteta je jedna tvrtka iz tipa entiteta Tvrtka koja ima otvoren račun u najmanje nula i najviše više banaka iz tipa entiteta Banka. Između tipa entiteta Proizvod i tipa entiteta Račun je relacija sa brojnosti tipa veze M:M. Detaljnije i preciznije se brojnost tipa veze na relaciji čita: jedan račun iz tipa entiteta Račun pridružuje se najmanje nijednom i najviše većem broju proizvoda drugog tipa entiteta Proizvod, obrnuto jedan proizvod tipa entitet Proizvod pridružuje se najmanje nijedan i najviše većem broju računa tipa entitet Račun. Takva veza zbog svojih brojnosti postaje agregacija Sastoji.

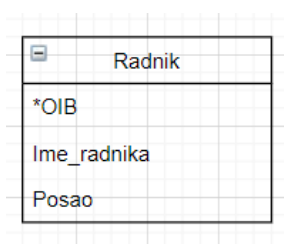


2 Crow's Foot notacija

Osim Chenova modela za notaciju koji je najkorišteniji model, Crow's Foot notacija je druga široko korištena notacija za dijagram Entiteti - Veze. Također, Crow's Foot notacija se naziva i Martinova notacija po James Martinu. James Martinova metoda znatno je pomogla stvoriti temelj za alat CASE¹. U ORACLE² se danas u svojim CASE alatima koristi slična takva notacija (Gupta, 2018).

2.1 Koncepti modela

U modelima napravljenim Crow's Foot notacijom tipovi entiteta su predstavljeni kao pravokutnici. Pravokutnik je razdijeljen na dio sa nazivom tog tipa entiteta i atributima. Naziv se piše u gornjem dijelu pravokutnika. U pravokutnik pišemo i pripadajuće attribute koji su navedeni u donjem dijelu s pripadnim ključevima. Primarni ključ OIB identificira taj tip entiteta Radnik. Što je prikazano na Slici 12.



Slika 12 Primjer tipa entiteta sa određenim atributima i primarnim ključem

Crow's Foot notacija se inače koristi za opisivanje relacije između tipova entiteta sa jedan na jedan ili više ili više prema više. Samo se binarna relacija koristi u ovoj notaciji. Notacija izgleda doslovno kao otisak noge od vrane sa tri razapeta prsta. Ti „prsti“ predstavljaju relacije. Na kraju veze postoje različiti oblici koji prikazuju relativnu kardinalnost relacije. Kardinalnost relacije prikazuje se crtanjem ravnih linija. Tri različita simbola koristimo za kardinalnost. To su uz crow's foot (>, <), još i „nula“ (0) te „crtica“ (|). Simboli se koriste u paru na relaciji sa četiri vrste kardinalnosti, odnosno sa svake strane entiteta po dvije.

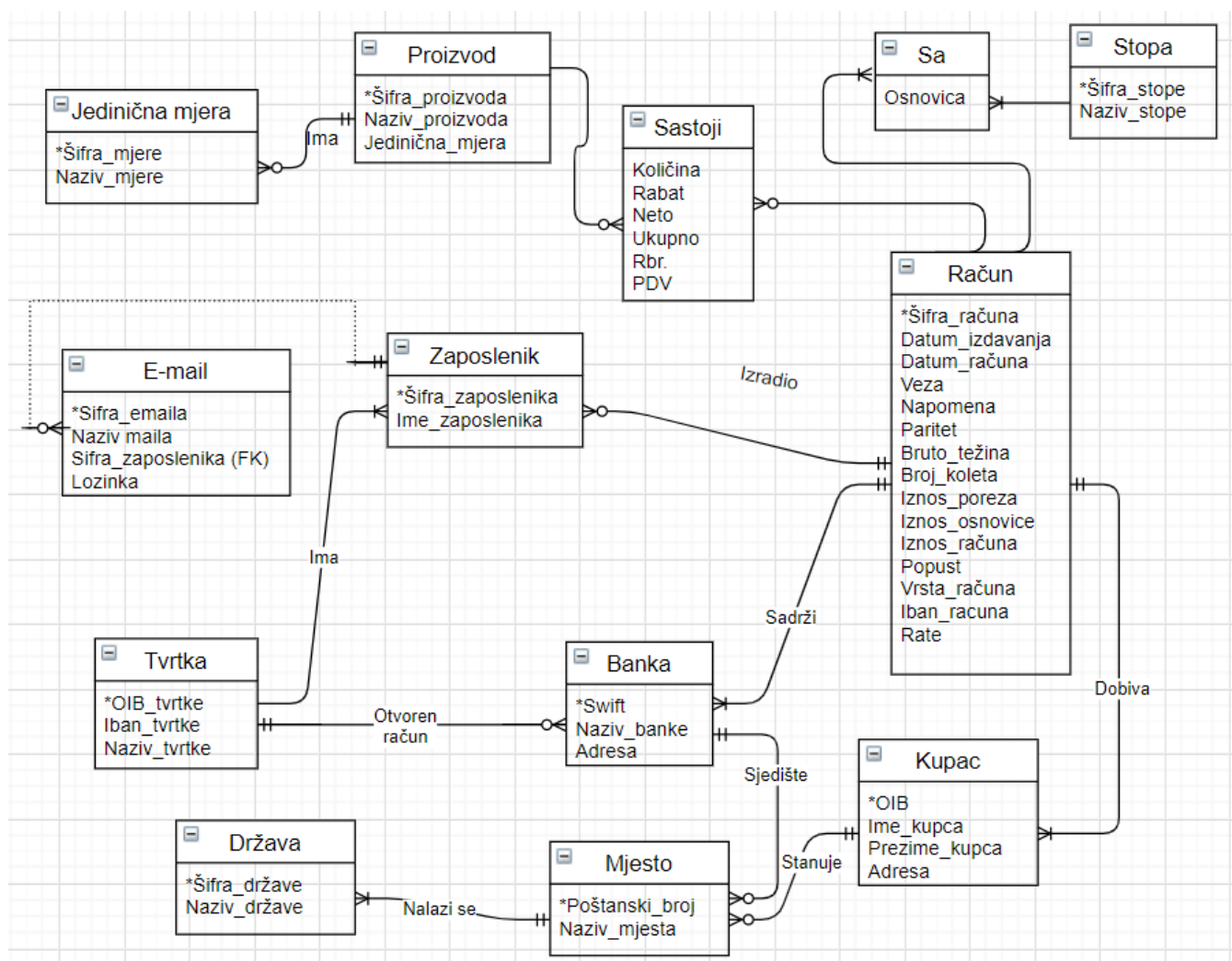
Postoje identifikacijske i ne identifikacijske relacije. Kod identificirajućeg odnosa podređena tablica ovisi o roditeljskoj tablici. Na prvom modelu (Slika 13) prikazana je notacija 1:1 gdje jedan student tipa entiteta Student sjedi na najmanje i najviše jednoj stolici tipa entiteta Stolica. Također jedan na jedan odnos se često koristi za uklanjanje NULL vrijednosnih polja. Nasuprot tome kod M:M notacije najmanje jedan i najviše više studenata tipa entitet Student (Slika 14) sudjeluje u najmanje jednoj i najviše više predmeta tipa entitet Predmeta. Time opisujemo da za svaki zapis u jednoj tablici postoji mnogo mogućih evidencija u drugoj tablici. Sljedeća notacija 1:M (Slika 13), jedan profesor tipa entitet Profesor može održavati najmanje jedan i najviše više predmeta tipa entitet Predmet, te su upravo ti

¹ CASE - Computer-Aided Software Engineering

² ORACLE – sustav upravljanja bazama podataka

2.2 Model podataka za dokument Račun napravljen Crow's Foot notacijom

Modeliran je dokument Račun (sa slike 11), te je napravljen dijagram uz pomoć Crow's Foot notacije. Tipovi entiteta su prikazani kao pravokutnici sa oštrim rubovima tj.: Račun, Proizvod, Jedinična mjera, itd. Unutar tog pravokutnika, tipovi entiteta imaju navedene atribute kao pripadajući kao i ključeve kao što je Šifra_računa primarni ključ tipa entitet Račun. Uz primarni ključ kao što smo naveli postoji i strani ključ. Strani ključ se nalazi u entitetu E-mail, kao preuzeti ključ od tipa entitet Zaposlenik. Entiteti su kao i uvijek povezani relacijama koji mogu i ne moraju imati naziv te veze sa također pripadnom kardinalnosti. Primjerice jedan račun tipa entiteta Račun dobiva najmanje jedan i najviše više kupaca tipa entitet Kupac, dok kupac tipa entitet Kupac dobiva najmanje i najviše jedan račun tipa entiteta Račun.



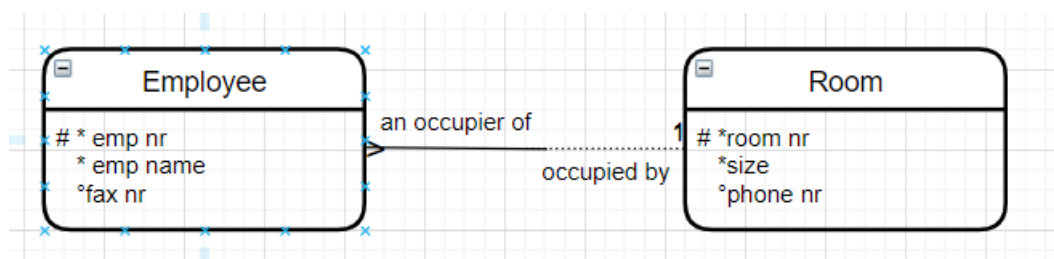
3 Barkerova notacija

Richard Barker u suradnji sa Ian Palmer i Harry Ellis razvili su notaciju. Kasnije je takva notacija nazvana Barkerova notacija u trenutku kada se Barker pridružio ORACLE-u te detaljnije definirao notaciju u svojoj knjizi „Entity Relationship Modelling“. Knjiga je jedna od serija knjiga o CASE Metodi (Halpin & Morgan, 2010). Ova metoda je varijacija metode Crow's Foot notacije. Notaciju je usvojio ORACLE Korporacija za svoje alate u CASE dizajnu. Korištena je zbog svoje učinkovitosti uporabe prostora i čitljivosti modela. Danas se više preferira UML kao alternativa Barkerove notacije, iako se smatra da je Barkerova notacija najbolja EV notacija i dobiva veliku podršku (Halpin & Morgan, 2010).

3.1 Koncepti modela

Barkerova notacija se minimalno razlikuje u pojedinostima od Crow's Foot notacije. Tip entitet je prikazan kao pravokutnik ali sa oblim rubovima i sa imenom tog entiteta u gornjem dijelu pravokutnika. Kao u prethodnoj metodi, atribut se piše u donjem dijelu pravokutnika tog tipa entiteta. U ovoj metodi attribute označavamo malo drugačije. Postoje obavezni i ne obavezni atributi tako ih se i različito označava. Ispred obaveznih atributa pišemo zvjezdicu „*“ ili ispunjenu točkicu. Obavezni kao i što sam naziv kaže su obavezni kod izrade modela. Prazna točkica „°“ pokazuje da je atribut ne obavezan. Ograničenja se pišu prije naziva navedenog atributa. Na slici 16 prikazane su vrste atributa. Emp nr u prijevodu broj radnika je atribut koji je obavezan u tipu entiteta Employee u prijevodu Radnik. Primjer ne obaveznog atributa je fax nr odnosno broj fax. Ljestvama „#“ pokazujemo da je atribut primarni ključ tog entiteta, za primjer je naveden emp nr koji definira tip entiteta Employee.

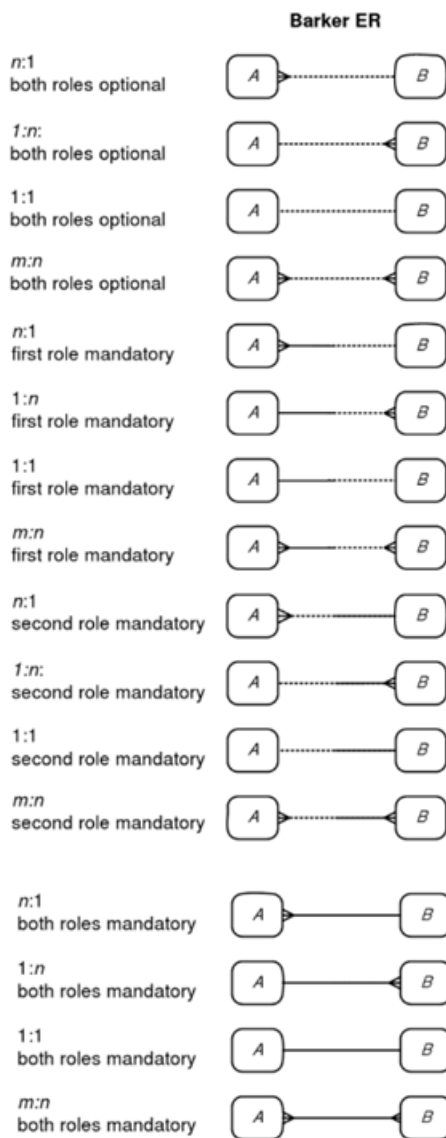
Prilikom povezivanja tipova entiteta pridružena je veza. U Barkerovoj notaciji koristi se binarna veza. Binarna veza je sad podijeljena u dva dijela kao šta je prikazano na Slici 16. Relacije su prikazane svaka sa svojim imenom. Svaki dio linije sadrži svoje ime i binarni odnos je čitljiv u oba smjera. Puna linija ukazuje na obaveznu kardinalnost i znači da entitet radnik tipa entitet Radnik okupira jednu sobu tipa entiteta Soba. Točkasta linija označuje da je ta kardinalnost opcionalna tj. entitet soba tipa entitet Soba okupira najmanje nula i najviše više radnika tipa entiteta Radnik.



Slika 16 Primjer binarne veze između dva entiteta (Halpin & Halpin, 2001)

Kako bi se istaknule opcionalne i obavezne kardinalnosti, Barker je preporučio tehniku imenovanja relacija. Prvi tip entitet označujemo sa A, drugi sa B i relaciju između njih R. Relacija A R B označava njihovu povezanost na način da relacija R povezuje tipove entiteta A i B. Svaki tip entiteta A, odnosno Radnik u našem slučaju, mora okupirati jednu i samo jednu Soba, odnosno tip entitet B, dok Soba može

biti sa jednim ili više Radnika u njoj. Kad je prvi dio veze obavezan, koristimo izraz „mora“, dok izraz „može“, koristimo kada je prvi dio opcionalan. U drugom dijelu veze kada postoji jedan ili više kardinalnosti, koristimo izraz „jedan i samo jedan“ odnosno „jedan ili više“. Postoji 16 uzoraka u slučaju kada je svaki dio binarne relacije označen s opcionalnim ili obaveznim tj. s izrazima „više“ ili „jedan“ (Halpin & Morgan, 2010). U nastavku na Slici 16 i Slici 17 prikazano je svih 16 vrsta dijagrama.

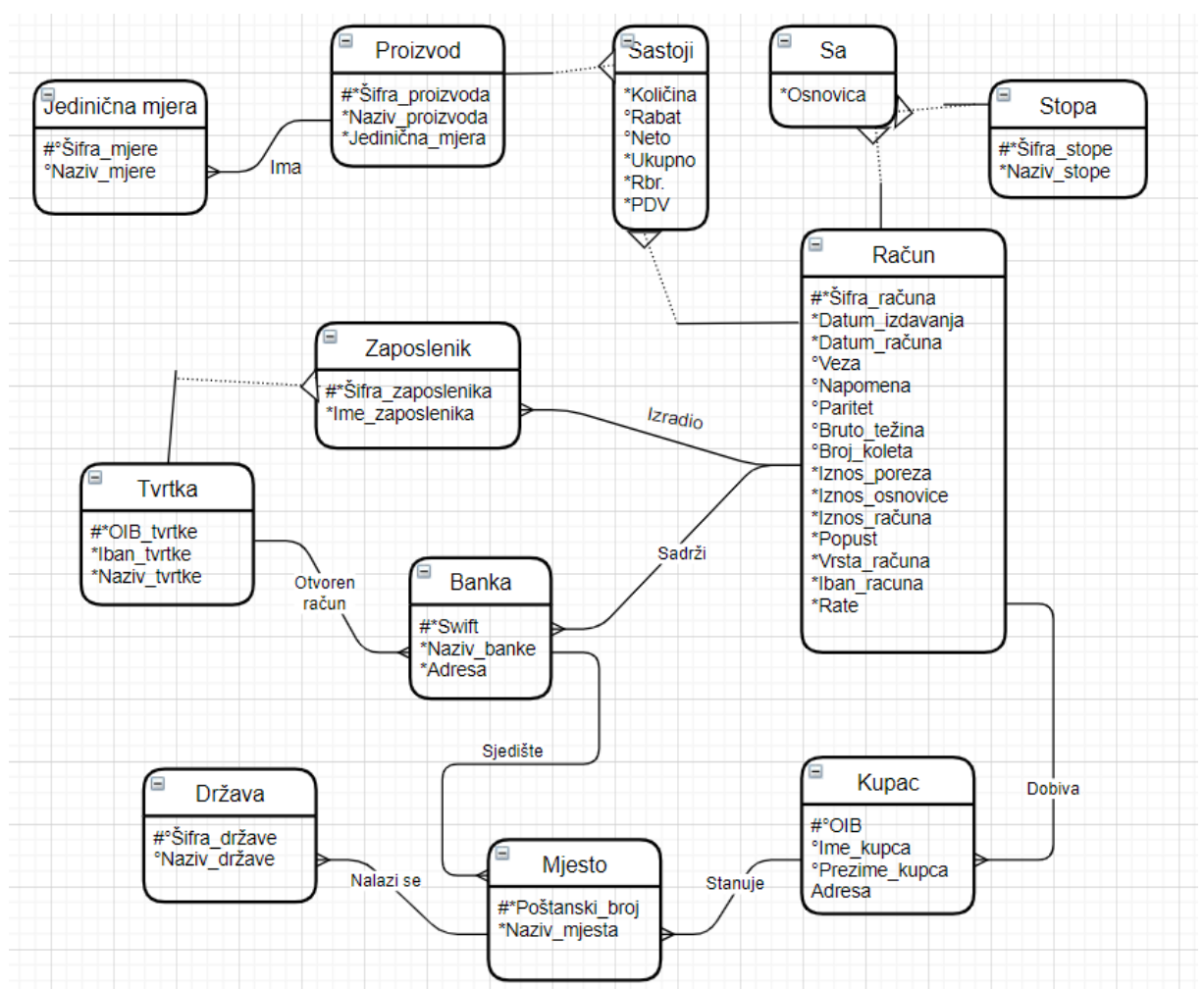


Slika 17 Vrste binarnih veza (Halpin & Morgan, 2010)

3.2 Model podataka za dokument Račun napravljen Barkerovom notacijom

Prikazan je opširni prikaz dijagrama napravljen Barkerovom notacijom iz dokumenta račun sa slike 11. Tipovi entiteti su pravokutnici sa oblim rubovima: Račun, Zaposlenik... Unutar pravokutnika navedeni su svi pripadajući atributi kao i ključevi. Novitet kod atributa, kao što je navedeno već u poglavlju „Koncepti modela“, je određivanja obveznosti atributa. Obavezni atributi su Šifra_računa, Datum_izdavanja kao atributi tipa entiteta Račun i oni su sastavni dio tog računa i obavezni su. Obavezne

atribute moramo uvrstiti u dokument, kao takav dokument ne bi bio potpuni. Atribut kao što je Šifra_računa se mora pojaviti na računu kao i atribut Datum_izdavanja kao dio tipa entiteta Račun. Kao primjer ne obaveznih atributa navedena je Napomena kao atribut tipa entiteta Račun. Atribut Napomena u tipu entiteta Račun kao što je rečeno nije obavezan jer tip entiteta Zaposlenik koji izdaje račun može, a i ne mora napisati napomenu. Kao drugi primjer ne obaveznog atributa je Šifra_mjere tipa entiteta Jedinična mjera. Atribut Šifra_mjere je primarni ključ i definira entitet jediničnu mjeru kao tip entiteta Jedinična mjera. Iako atribut Šifra_mjere je primarni ključ i obavezna je kao dio tipa entiteta Jedinična_mjera, nije obavezna kao dio dokumenta račun i tipova entiteta Račun i Kupac. Relacije koje povezuju tipove entiteta kao što je navedeno mogu biti opcionalne i obavezne. Sve veze na dijagramu su većinom obavezne osim dijelova veze koje spajaju tipove entiteta Proizvod i Sastoji gdje je dio veze opcionalan a dio obavezan. Drugačije bi se mogli izraziti da je entitet proizvod kao dio tipa entiteta Proizvod obavezan element ovog dokumenta, dok entitet sastoji tipa entitet Sastoji nije jer sam kao takav ne može postojati bez tipa entiteta Proizvod.



4 IDEF1X notacija

Notacija IDEF1X odnosno integracijska definicija za informacijsko modeliranje. IDEF1X je jezik za modeliranje podataka za razvoj semantičkih modela podataka (Wikipedia IDEF1X, 2020). Bazirana je na konceptualnom jeziku IDEF1. IDEF1X je danas najprošireniji i najviše se koristi kao dio metodologije IDEF. Uz veliku proširenost ovog jezika podržan je i u CASE alatu (Halpin & Morgan, 2010).

4.1 Koncepti metode IDEF1X

IDEF1X jezik je hibridni jer kombinira više konceptualnih notacija. Koncepti u IDEF1X jeziku koji se koriste za izradu modela su: entitet, relacije i ključevi. Entitet više ne znači „entitet instance“ nego „tip entiteta“. Usvojivši riječ „entitet instance“ praktičari upotrebljavaju riječ „instanca“ umjesto „tipa entiteta“, ali zadržat ćemo se na „tipu entiteta“ kao što je nama znano. Jedan ili više atributa sadrži svaki tip entiteta koji se temelje na domeni. Skup vrijednosti podataka iste vrste podataka imenujemo domenom. Ograničene su binarnom asocijacijom veze između tipova entiteta (Halpin & Morgan, 2010).

Kod izrade IDEF1X modela koriste se bar tri faze prikaza: EV prikaz, „temelj - ključ“ prikaz i „uključeni svi atributi“ prikaz. Razlike u prikazima prikazane su u tablici.

<i>ER view</i>	Entity types: No identification scheme or keys defined No need to specify any attributes No attribute constraints (e.g., mandatory) Relationships: Many:many relationships are allowed Identifying/nonidentifying distinction is not made
<i>Key-based view</i>	Include at least all the key-based attributes Classify connection relationships as identifying or nonidentifying
<i>Fully attributed view</i>	Include all attributes

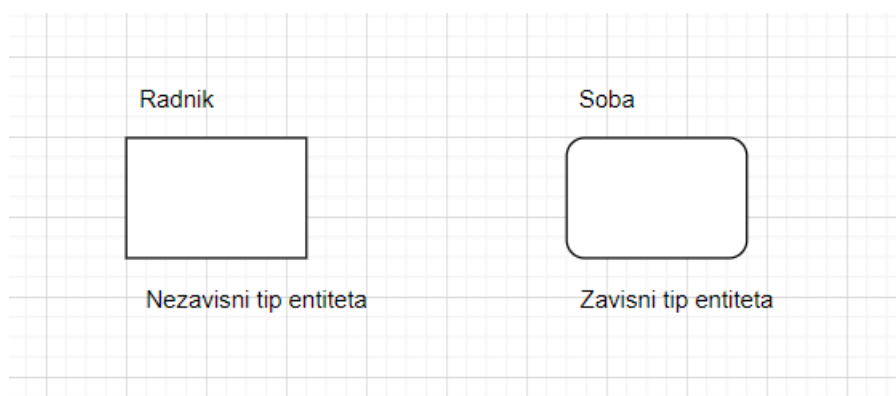
Slika 18 Tri različita prikaza u IDEF1X (Halpin & Morgan, 2010)

Neki određeni CASE alati ne podržavaju EV prikaz i tu je nastao problem. Zbog tog problema IDEF1X jezika nije baziran na potpunoj izradi EV modela. EV prikaz u IDEF1X je nepotpun prikaz EV modela i njega bi se moglo pretvoriti u pravi model. Iako, umjesto da se obnovi i prikaže kao konceptualni model, IDEF1X je okrenut logičkoj razini kako bi se dodali dodatni detalji. Da bi dobili kompletan model, EV prikaz moramo prikazati kao model koji je temeljen na ključu odnosno moramo ga prikazati tako da su svi ključevi nekog tipa entiteta uključeni. Pored ključeva kao sastavni dio modela i tipa entiteta, potrebno je i da su svi atributi uključeni. Takav prikaz sličan je u mnogim pogledima relacijskom modelu.

IDEF1X je kod prikaza temeljenom na ključu i svim atributima u principu tablica odnosa. Te relacije su reference stranog ključa³ primarnom. Referentni tipovi entiteta su nazvani „dijete“ i „roditelj“, te su predstavljeni kao izvor relacije odnosno cilj iste. U atributima se pohranjuju podaci odnosno činjenice.

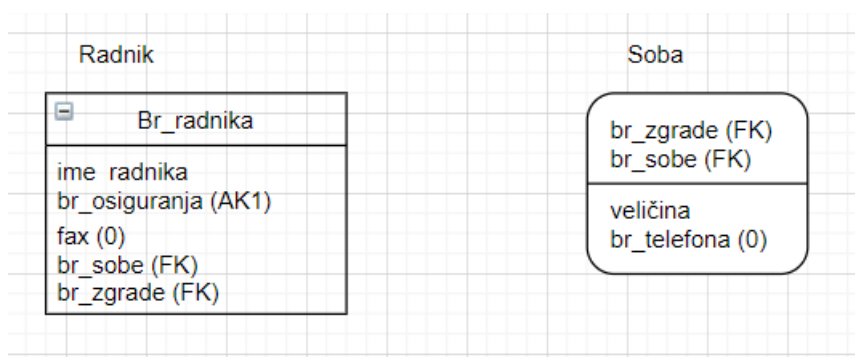
Tip entiteta može biti klasificiran kao ovisan ili ne ovisan o identifikatoru. Neovisnost tipa entiteta ukazuje ukoliko njegov identifikator ne ovisi o drugom tipu entiteta ili njegov odnos nije baziran na shemi identifikacije prema drugom tipu entiteta. Također, primarni ključ ako i samo ako nema komponentu koja je strani ključ iako je moguće da sadrži atribut koji je samo dio stranog ključa može biti ne ovisan tip entitet.

Na slici 19 prikazani su tipovi entitet. Kod nezavisnog tipa to je pravokutnik sa oštrim rubom dok je kod zavisnog to pravokutnik sa oblim rubovima te se ime tipa entiteta piše iznad.



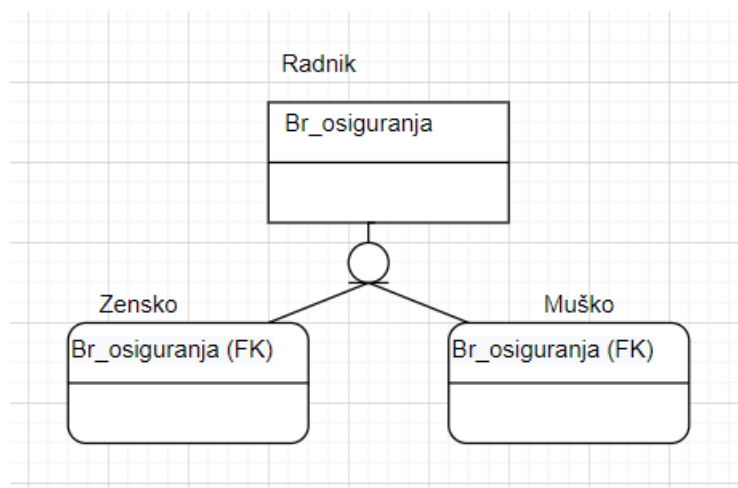
Slika 17 Tipovi entiteta

Atributi kod tipova entiteta smješteni su unutar pravokutnika. Primarni ključ atributa je uvijek naveden u gornjem dijelu i on je izabran kao jedinstveni identifikator tog entiteta. Osim primarnog ključa koji pruža primarnu identifikaciju, razlikujemo još i alternativne te strane ključeve. Kod alternativnog ključa imamo dodatnu oznaku „AKn“ gdje je $n > 0$. Alternativni ili zamjenski ključevi koji su različiti moraju imati različite vrijednosti za n . Iako mogu imati i iste vrijednosti za n , ako je taj zamjenski ključ složen. Kod stranog ključa imamo dodatnu oznaku „FK“. Strani ključ se promatra kroz kategorizaciju te je rezultat takozvane „migracije“ primarnog ključa. Atributi su svi jednoznačni. Ne obaveznost atributa prepoznavamo po „0“. Na slici 20 prikazani su svi navedeni ključevi za tipove entiteta.



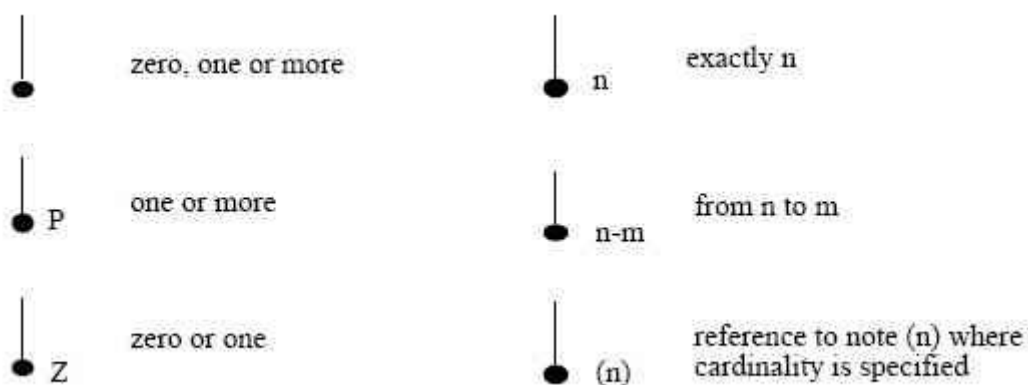
³ Strani ključ je skup jednog ili više atributa čije se vrijednosti moraju pojaviti kao vrijednost nekog primarnog ključa

Kod izrade modela tipove entiteta možemo generalizirati u domenu hijerarhije. Domenom nazivamo skup vrijednosti podataka sa istom vrstom podataka i stvarnom vrijednosti za instancu atributa (Wikipedia IDEF1X, 2020). Svaki određen atribut temeljen je na jednoj osnovnoj domeni. Također, više atributa mogu imati temelj u istoj domeni. Slikom 21 prikazana je jedna hijerarhija nad radnikom. Također je prikazan i odnos ranije rečenog referentnog tipa entiteta „djeteta“ tj. Žensko i Muško i „roditelja“ tj. Radnika.



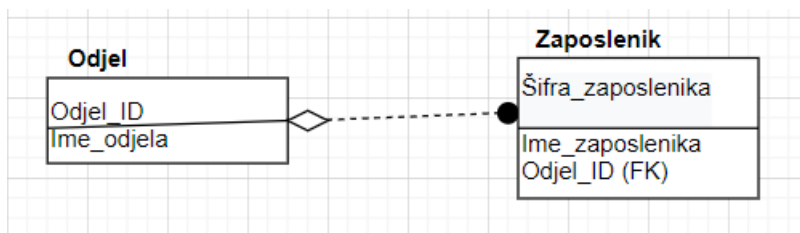
Slika 19 Domena hijerarhije

Povezivanjem instanci između dva tipa entiteta dobivamo veze. Različite vrste veza prikazane su na slici 22. Odnosi su prikazani kao crte koje završavaju punom točkom na podređenom tipu entiteta. Linija kao takva može biti ili puna ili isprekidana, što ovisi o vrsti te veze. Uz navedene veze postoje i identifikacijska i ne identifikacijska veza. Kao što je navedeno u prijašnjem poglavlju, identifikacijski tip veze nema svoj jedinstveni identifikacijski ključ već se koristi tuđim odnosno ključem drugog tipa entiteta. Dok odnos u kojem taj migrirani odnosno drugi atribut postaje dio djetetovog primarnog ključa, onda je taj odnos ne identifikirajući. Pune linije inače prikazuju identifikacijsku vezu, a isprekidana ne identifikacijsku. Tako iz perspektive djeteta gledamo kardinalnost veze i to dijete ovisi o ključu drugog tipa entiteta.



Slika 20 Kardinalnost veze (Wikipedia IDEF1X, 2020)

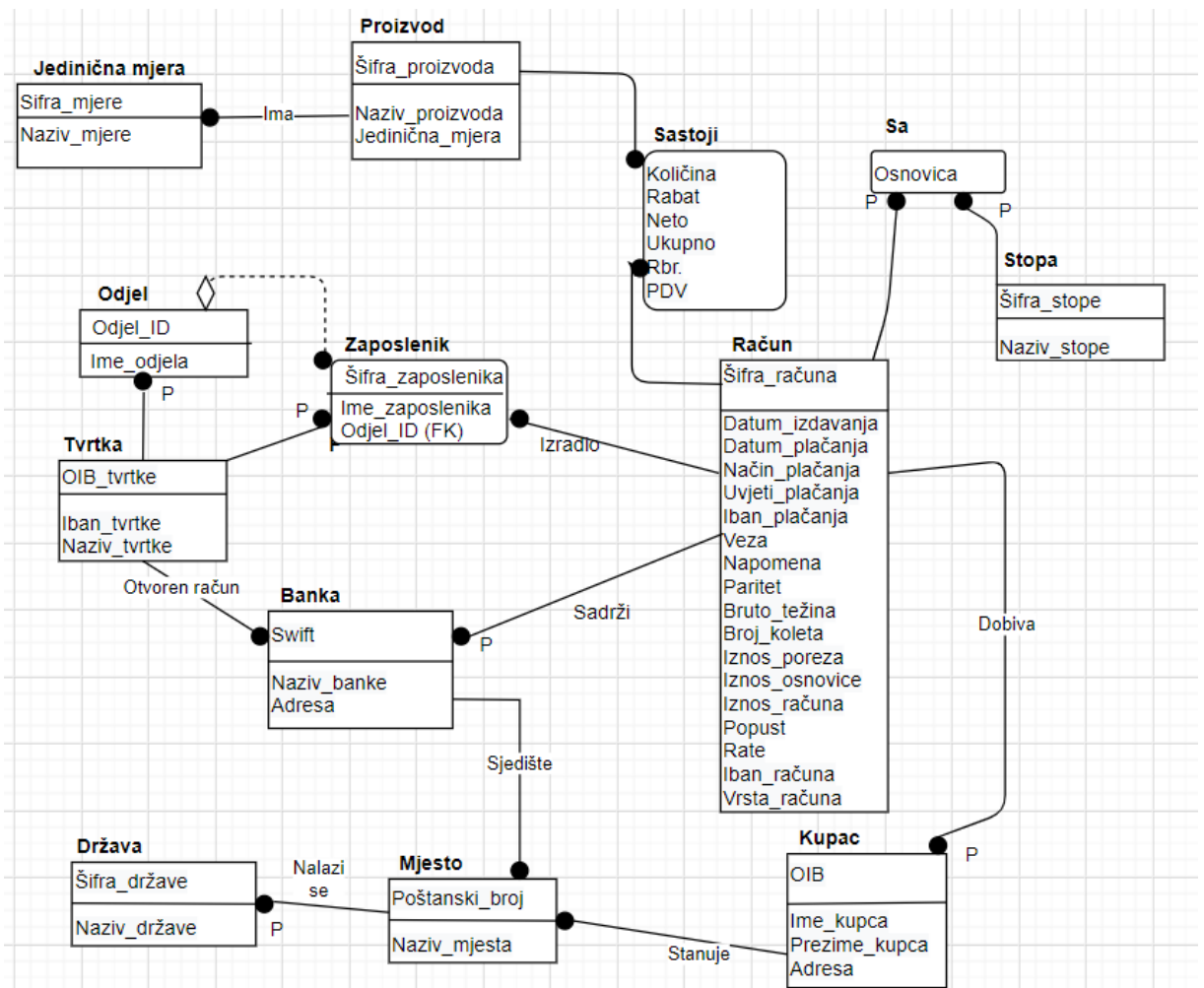
Uz navedene identifikacijske veze, kao šta su prikazane na slici, naveli smo i ne identifikacijske (slika 23). Osim što je crta isprekidana, jezik IDEF1X koristi osim pune točke i prazni dijament da se naznači da je veza ne obavezna. Taj dijament je uvijek na strani nadređenog tj. roditeljskog tipa entiteta. Uz navedeno, prikazan je i strani ključ odnosno `Odjel_ID` koji je primarni ključ u odjelu tipu entitetu `Odjel`.



Slika 21 Ne identifikacijska veza

4.2 Model podataka za dokument Račun napravljen IDEF1X notacijom

Dijagram napravljen IDEF1X notacijom na temelju dokumenta račun sa slike 11 prikazuje detaljan prikaz tog dokumenta. Tipove entiteta crtamo kao pravokutnik. Tip entitet može, a i ne mora ovisiti o drugom tipu entiteta. Ukoliko ovisi o nekom drugom entitetu oni sami za sebe ne mogu postojati. Tako su u dijagramu zavisni tipovi entiteta pravokutnici sa oblim rubom kao što je `Zaposlenik` koji je ovisan o tipu entiteta `Tvrtka` i ne može sam za sebe stajati u ovom dijagramu. `Tvrtka` je ne zavisan tip entiteta sa oštrim rubovima i on ne ovisi o niti jednom elementu. Navedenom notacijom odvojili smo primarni ključ unutar tipa entiteta sa ostalim atributima i ključevima. Primjerice tip entiteta `Račun` u prvom dijelu pravokutnika ima `Šifru_računa` kao primarni ključ navedenog entiteta, dok su ostali atributi u donjem dijelu. Relacije povezuju tipove entiteta, odnosno detaljnije i preciznije čitamo jedan entitet zaposlenik iz tipa entiteta `Zaposlenik` je izradio najmanje i najviše jedan račun iz tipa entiteta `Račun`, odnosno račun iz tipa entiteta `Račun` je napravio najmanje nula i najviše više zaposlenika iz tipa entiteta `Zaposlenik`. Uz identifikacijsku vezu uveli smo i ne identifikacijsku vezu gdje nadređeni tip entiteta, koji ima svoj primarni ključ, postaje strani ključ podređenom tipu entiteta.



5 Unified Modeling Language (UML)

Unified Modeling Language odnosno UML je razvojni jezik za modeliranje u području softverskog inženjerstva koji je namijenjen pružanju standardnog načina vizualizacije dizajna sustava (Wikipedia UML, 2020). Sa UML jezikom počinje prijelaz sa tehnike koja se temeljila na kodovima na tehniku orijentiranu modelu za izradu softvera. Tijekom 90-ih godina razvilo se puno objektno – orijentiranih metoda za razvoj modela. Zbog prevelikog izbora i interpretacija objekta općenito, poticalo je korisnike na prihvaćanje takozvane metode „čekaj i vidi“ gdje se ograničio daljnji napredak korištenom metodom. Tako je 1997. godine standard usvojen, te je službeno dogovoreno da se nadogradi softver i dodaju nužne mogućnosti UML jeziku za modeliranje poslovnih procesa (Wikipedia UML, 2020).

5.1 UML dijagrami

Dijagram je graf i sastoji se od elemenata koji tvore jedan sustav. Razlikujemo devet tipova UML dijagrama. UML dijagrami su podijeljeni na statičke i dinamičke dijagrame. Pod statičke dijagrame ubrajamo: dijagram klase, dijagram objekta, dijagram komponenti, te dijagram rasporeda. Pod dinamičke dijagrame ubrajamo: dijagram aktivnosti, dijagram korištenja, dijagram stanja, dijagram slijeda, te dijagram suradnje. Najkorišteniji dijagrami za komunikaciju i izradu modela su: dijagram korištenja, dijagram slijeda, dijagram aktivnosti, dijagram stanja te dijagram klase. Potonji je u zasebnom poglavlju u nastavku teksta i dodatno pojašnjen, upravo zbog njegove važnosti prilikom usporedbe sa drugim notacijama.

5.1.1 Dijagram klase

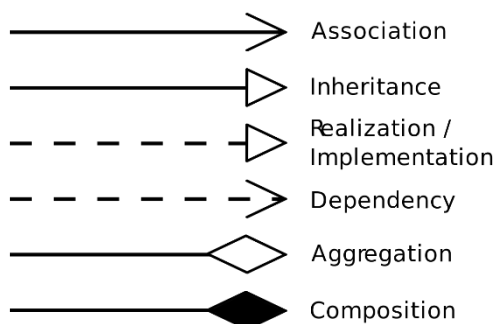
Dijagram klase jest vrsta strukturnog dijagrama u softverskom inženjerstvu. Isti opisuje strukturu sustava objašnjavajući klase unutar sustava, njihove atribute i relacije (Wikipedia CLASS, 2020). Dijagram je vrlo pregledan kod strukture aplikacije. Korisnik bolje može razumjeti shemu aplikacije.

5.1.1.1 Koncepti dijagrama klase

Elementi koji su bitni u dijagramu klase su: naziv klase, atribut i operacija. Kao kod svih dijagrama, naziv klase se piše u gornjem dijelu klase. Klasa ili entitet je prikazana kao pravokutnik koji dijeli atribute i operacije te klase koji su pisani u posebnim odjelima u pravokutniku. Također, kod izrade sustava, klasa se može pojaviti u više dijagrama jer se u jednom dijagramu klase ne moraju uključiti sve klase. Što se tiče atributa, ta komponenta je smještena odmah ispod naziva klase te opisuje taj objekt koji modeliramo. Ispred atributa uvodi se oznaka vidljivost atributa. Vidljivost određuje

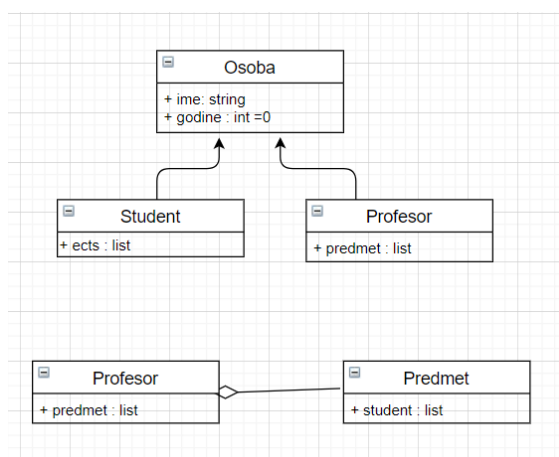
pristup tim atributima, odnosno određuje tko može vidjeti taj atribut ili pak operacije te klase. Oznake koje se koriste kod određivanja vidljivosti su: javno „+“, privatno „-“, zaštićeno „#“ i paket „~“.

UML jezik kod dijagrama klase razvrstava veze po vrstama logičkih odnosa. Te veze se dijele na način kako su prikazane na slici. U nastavku su obrazložene najkorištenije relacije.

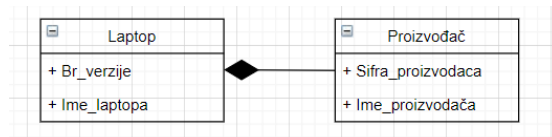


Slika 22 Vrste veza (Wikipedia CLASS, 2020)

Asocijacija prikazuje obiteljsku vezu. Prezntira binarnu vezu koja se crta kao linija. Povezati može navedene dvije ili čak i tri klase. Kao i svaka druga relacija tako se i ova može imenovati. Na jednom kraju, kao što je vidljivo, nalazi se strelica, dok s druge strane možemo dodati i drugo svojstvo. Uglavnom se zna uključiti agregacija, iako postoji još uz to i refleksno, jednosmjerno kao i dvosmjerno pridruživanje. Generalizacija je prikazana na slici 25 sa tipovima entiteta Osoba, Studentom i Profesorom. Kod agregacije nam u prijevodu može poslužiti riječ „ima“. To je binarna veza koja povezuje dvije klase kao npr. klase Profesor i Predmet jer obje klase postoje same za sebe. Spomenute klase odnose se kao jedinice te za razliku od agregacije, kod kompozicije klase se ovisne jedna o drugoj. Veza je 1 na prema 1. Tako npr. tip entiteta Laptop koji posjeduje na primjer tip entiteta Student, ima najmanje jednog i najviše jednog proizvođača tipa entitet Proizvođač.



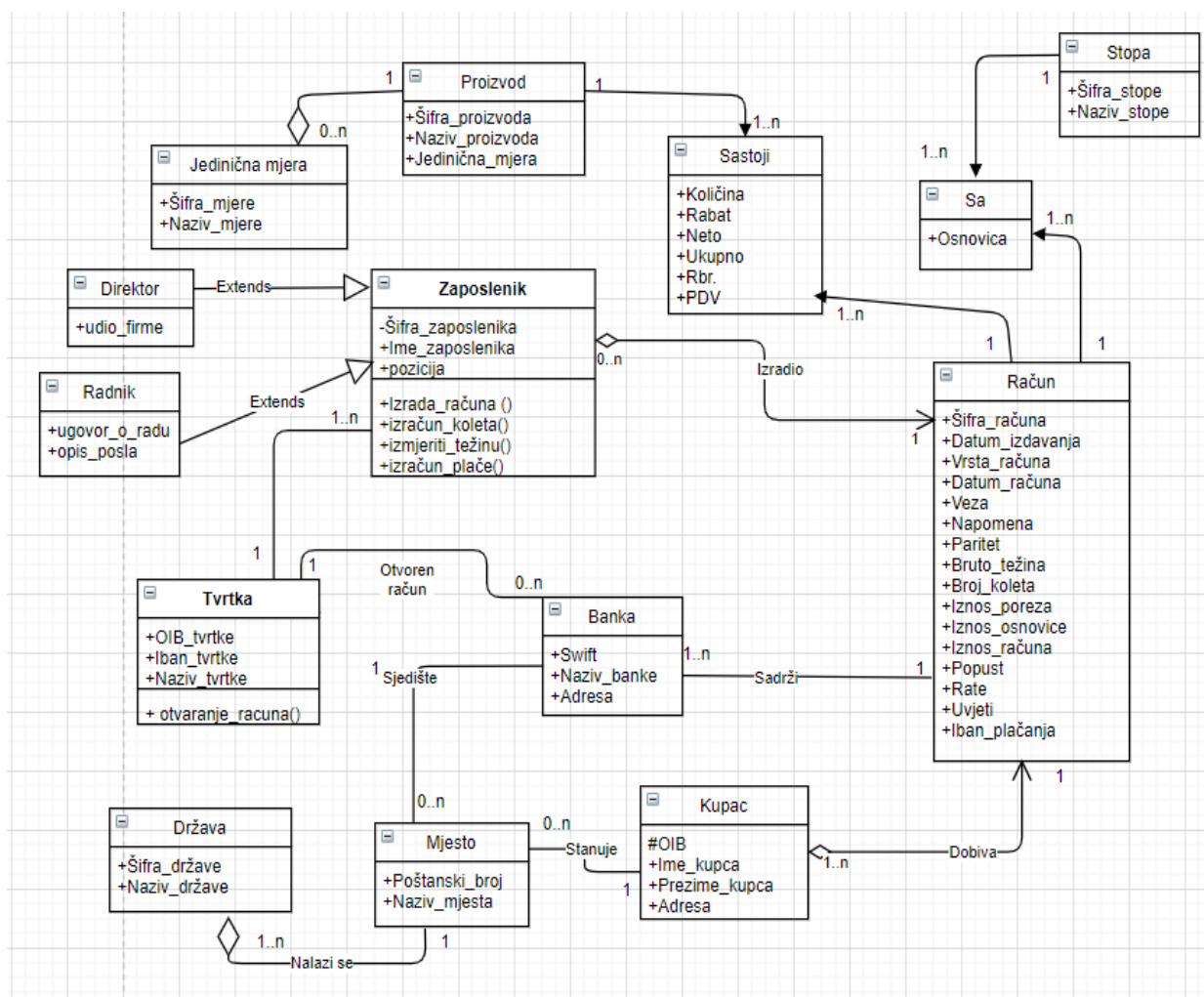
Slika 23 Primjer generalizacije i agregacije dijagrama klase



Slika 24 Primjer kompozicije

5.2 Dijagram klase za dokument Račun

Zbog raznovrsnosti dijagrama UML notacije trebamo naći pravi dijagram za pravi slučaj. U našem slučaju na temelju dokumenta račun sa slike 11 odabran je dijagram klase te je uz pomoć njega napravljen temeljiti prikaz. Entitet u prikazu može biti podijeljen na čak tri dijela. Tip entiteta Zaposlenik ima svoje atribute, točnije Ime_zaposlenika te Šifra_zaposlenika. Šifra zaposlenika je primarni ključ tipa entiteta Zaposlenik. Osim nabiranja atributa, atribut smo razvrstali i možemo odrediti njihovu vidljivost. Slijedi da je Šifra_zaposlenika privatan atribut i ne mora biti vidljiv ostatku, dok je Ime tog zaposlenik kao javan atribut vidljiv svima. Nadalje atribut i primarni ključ OIB, kao dio tipa entiteta Kupac, je zaštićen. Uz navedene vrste atributa u trećem dijelu entiteta možemo imati i operacije. Tip entitet Zaposlenik izrađuje račun uz operaciju Izrada_računa() uz još dodatne operacije koje su nam bitne za određeni entitet. Tip entitet Zaposlenik smo generalizirali na tipove entiteta Direktor i Radnik. Tip entiteta Direktor i Radnik imaju zajedničke karakteristike tipa entiteta Zaposlenik. Svaki tip entiteta koji se generalizira u ovom slučaju sadrži karakteristike Šifra_zaposlenika, Ime_zaposlenika i Poziciju kao atribute tog entiteta. Ono što im je različito se nalazi u pojedinom pravokutniku odnosno tipu entiteta. Tip entitet Direktor ima svoj atribut Udio u firmu koji mu pripada, dok tip entiteta Radnik ima atribute Ugovor_o_radu i Opis_posla. Osim generalizacije kao relacija u dijagramu naveli smo još i agregaciju i asocijaciju. Agregacija pridružuje jednom tipu entiteta Proizvod tip entiteta Jedinična mjera kako je i prikazano u dijagramu. Između tipova entiteta Kupac i Računu se nalazi relacija sa asocijacijom koja udružuje agregaciju. Dvosmjerna asocijacija prikazana je vezom između tipova entiteta Tvrtka i Zaposlenik gdje jedna tvrtka tipa entitet Tvrtka ima najmanje jednog i najviše više zaposlenika tipa entitet Zaposlenik, dok sa druge strane gledano, najmanje jedan i najviše više zaposlenika tipa entitet Zaposlenik može raditi u najmanje i najviše jednoj tvrtki tipa entitet Tvrtka.



6 Usporedba notacija

Usporedbu samih notacija najlakše je prikazati na konkretnim primjerima. Konkretni primjeri su modeli podataka prethodno prikazani u ovom radu rađeni uz pomoć notacija.

Metoda EV je najkorištenija metoda u današnje vrijeme. Koristi se kako i u obrazovnim ustanovama tako i šire, i to za izradu dijagrama. Vrlo je jednostavna i razumljiva korisnicima. Iako govoreći o jednostavnosti, ova metoda nije dovoljno detaljna i to primjerice u određivanju obaveznosti atributa, dok je kod Barkerove notacije obaveznost atributa puno bolje određena. Gledajući ostale metode i dijagrame, a što se tiče izgleda entiteta, ništa značajnije se ne mijenja, dok kod atributa i veza vidimo značajniji napredak. Notacija se sastoji od pravokutnika, koji predstavljaju tipove entiteta, i dijamanata ili ravnih linija koje predstavljaju relacije između entiteta. Najveći je problem što danas imamo mnogo verzija Chenove notacije u kojima nemamo jedinstveni standard. Barkerova i IE notacija dolaze kao novije verzije notacija odnosno kao poboljšane verzije pravog EV modela.

Crow's foot notacija iako je sa vezama napredovala, nije najkorištenija notacija. No ipak, kao druga najkorištenija notacija, uvelike je pomogla u razvoju te je dobro podržana u CASE alatima. Notacija se sastoji od pravokutnika koji i dalje predstavljaju entitete te crte koje predstavljaju relacije. Relacijama su pridružene oznake tako zvan „vranin otisak“ koji predstavljaju kardinalnost. Kardinalnost je do tada bila sa Chenovim modelom prikazana samo brojkama. Chenova i Crow's foot notacije, kao najkorištenije, nezgodne su za upotrebu te ih je pomalo teško i za objasniti.

Model EV u Barkerovoj notaciji koristi u konceptualnom dizajnu, i to radi funkcionalnije i više poslovne razine domene. Danas Oracle kao alternativu podržava UML i Barkerovu notaciju. Kod poslovne suradnje i modeliranja te izrade aplikacije za baze podataka, Barkerova je notacija zastupljenija. Temelj notacije su atributi kao i kod UML-ove notacije. Uz atribut, kao temeljem ovog modela, bitne su relacije koje su razrađenije od UML-ovih. Barkerova notacija se danas smatra najboljom entitet – veza dijagramom. Zbog te kompaktnosti NORMA⁴ alati za ORM⁵ se nadograđuju te modelu EV pružaju mogućnost da njegovi dijagrami budu prikazani kao živi prikaz na temelju ORM prikaza.

UML jezik uvelike mijenja zapis odnosno izradu dijagrama kao takvih zbog svoje raznolikosti. Kod ove notacije uvode se novi elementi. Slažu se elementi iz tri cjeline objektnog orijentiranog dizajna. Ti elementi su: Rumbaugh-ovo OMT modeliranje⁶, Boochova objektno orijentirana analiza i dizajn. (Connolly & Begg, 2002) UML notacija omogućila je izradu dijagrama koji su i jednostavni i složeni. Svi dijagrami su povezani međusobno, te je komunikacija od strane korisnika i projektanta znatnije olakšana. Ne slažu se svi da je prikladna notacija za izradu poslovnih zahtjeva, upravo zbog svoje složenosti, može doći do nestabilnosti tog modela. Kroz razna testiranja i podizanja mogućnosti dijagrama UML je odabran za OMG⁷ standard. Uz to što je prihvaćen od strane OMG-e od tad se

⁴ NORMA – Natural ORM Architect for Visual Studio je konceptualni alat za modeliranje koji primjenjuje metodu objekt - uloga

⁵ ORM – Object – relational mapping odnosno objektno – relacijsko mapiranje je tehnika programiranja za pretvoriti podatke koristeći objektno orijentirane programske jezike između nespojivih sustava

⁶ OMT modeliranje – tehnika objektnog modeliranja ili OMT je pristup objektnog modeliranja za softversko modeliranje i dizajn (Wikipedia, 2020)

⁷ The Object Management Group (OMG) je konzorcij za standard računalne industrije. Razvijaju standarde za integraciju poduzeća za niz tehnologija (Wikipedia, 2020)

razvijaju CASE-e alati koji ga također podržavaju. Uz sve vrste dijagrama koje imamo u UML, ovisno o kakvom području i za što nam je potrebno, možemo izraditi određeni model. Kod dijagrama klase koji je i prikazan u našem primjeru, prikaz tipa entiteta se nije mnogo mijenjao, no imamo dodatke za vidljivost, što može pridonijeti izradi modela. Veze iako slične onima kod entitet – veze metode su preglednije. Temeljem svega navedenog razvidno je kako je ispred UML-ovog jezika uspješna budućnost.

Što se tiče IDEF1X notacije, kao dio metodologije IDEF, ova notacija ima neke značajke razvijenije od ostalih IDEF-ovih. Ne samo to već su i proširenije od onih UML-ovih. Nešto što uvelike može pomoći u kreiranju dijagrama općenito jest korištenje UML uz IDEF1X notacije. Da bi se shvatila aplikacija i što korisnik točno želi, UML-ovim dijagramom počinjemo izradu modela. Nakon izrade dijagrama, ideje se prevedu u IDEF1X jezik te se dodaju detalji u bazi podataka. U bazi podataka kod se kreira sa IDEF1X jezikom. Bliži je dizajnu baze podataka i jasnije se prikazuju neki detalji o uzorcima. IDEF1X notacija je spoj Entitet-veze i relacijske notacije. Objektno – orijentirane značajke u bazi podataka kao i u programskim jezicima su proširenije. Tako se u navedenom slučaju vidi prednost IDEF1X notacije ispred UML-ove.

7 Zaključak

U ovom radu razmotrili smo nekoliko notacija za modeliranje podataka. Metode Entitet – veze kao segment metodologije MIRIS, metoda IDEF1X kao segment metodologije IDEF, te uz to i IE metoda, Barkerova i dijagram klase kao dio UML-a služe za izradu modela podataka. Navodeći notacije prikazali smo definicije koje su najbitnije i odnose se na određenu notaciju. Uz navedene definicije prikazani su modeli podataka sa pripadajućim elementima i relacijama. Kako bi predodžba istaknutih modela bila jasnija i radi lakše usporedbe istih, svaki dijagram za sebe je prikazan je sa svojim elementima kao i relacijama. Iz svega iznijetoga proizlazi zaključak kako svaka notacija, odnosno metoda ima određene prednosti, ali i nedostatke. Razlika je kao što smo naveli, u minimalnim sitnicama kod tipova entiteta, dok su veze sa svakim dijagramom sve naprednije te se konstantno uvode određeni noviteti. Chenova notacija je kao najjednostavnija i polazna notacija prema kojoj su i bazirani svi ostali modeli. Uz relacije, kardinalnost se unaprijedila izradom IE notacije te nadalje Barkerovom notacijom. Posebnost tih notacija, kao što je istaknuto, je u kardinalnosti odnosno znaku „vraninog otiska“ koji je zamijenio numeričko označavanje. Prilikom izrade modela i razgovora sa korisnikom kombiniraju se dijagrami. Ovisno o zahtjevima korisnika, kombiniraju se modeli kako bi se dobila što detaljnija slika i konačni krajnji model. Fokusirajući se na dizajn baze podataka upotrebljavamo češće IDEF1X jezik, dok kod UML imamo bolju komunikaciju i raznovrsnost dijagrama. Kod UML-a imamo više vrsta dijagrama. Ta raznolikost nam pruža mogućnost izbora, iako treba biti na oprezu kod odabira, budući da ne možemo svaki dijagram primjeniti na konkretan slučaj. Zbog složenosti moramo pripaziti da ne zakompliciramo korisniku model te da spriječimo eventualno pogrešno razumijevanje cijeloukupne slike. Različitost dijagrama UML velik je plus, ali unatoč svemu, Barkerov dijagram nam omogućuje uz svoju kompaktnost i usavršavanje relacijakao i kardinalnosti veću prihvaćenost među projektantima.

Popis slika

Slika 1 Tip entiteta sa nazivom	8
Slika 2 Simbol tipa veze	9
Slika 3 Primjer tipa entiteta Nastavnik sa atributima i pripadajućim primarnim ključem	9
Slika 4 Primjer egzistencijale ovisnosti Staža o Nastavniku.....	10
Slika 5 Primjer egzistencijalne veze	10
Slika 6 Primjer identifikacijskog slabog tipa entiteta (Pavlič, 2011)	10
Slika 7 Primjer slabog entiteta koji može biti istovremeno i ovisan i identificiran drugim tipom entiteta (Pavlič, 2011)	10
Slika 8 Primjer tipova entiteta koje povezuje agregacija sa priadajućim atributima i ključevima	11
Slika 9 Tip entiteta Radnik s povratnom vezom Rukovodi. (Pavlič, 2011)	11
Slika 10 Primjer generalizacije i specijalizacije	12
Slika 11 Dokument račun za tvrtku Termag	13
Slika 12 Primjer tipa entiteta sa određenim atributima i primarnim ključem	15
Slika 13 Identificirajuće veze (cs.uregina.ca, 2020)	16
Slika 14 Identificirajuće veze (cs.uregina.ca 2020).....	16
Slika 15 Primjer ne identificirajuće veze	16
Slika 16 Primjer binarne veze između dva entiteta (Halpin & Halpin, 2001)	18
Slika 17 Vrste binarnih veza (Halpin & Morgan, 2010)	19
Slika 18 Tri različita prikaza u IDEF1X (Halpin & Morgan, 2010)	21
Slika 19 Tipovi entiteta	22
Slika 20 Ključevi (primarni, alternativni i strani) (Halpin & Morgan, 2010).....	23
Slika 21 Domena hijerarhije	23
Slika 22 Kardinalnost veze (Wikipedia IDEF1X, 2020)	23
Slika 23 Ne identifikacijska veza	24
Slika 24 Vrste veza (Wikipedia CLASS, 2020)	27
Slika 25 Primjer generalizacije i agregacije dijagrama klase	27
Slika 26 Primjer kompozicije.....	27

Literatura

Bezivin J., Pierre – Alain Muller; The Unified Modeling Language „UML“ '98:Beyond the Notation, 2004

Connolly Thomas M., Carolyn E. Begg; Database Systems: A Practical Approach to Design, Implementation, and Management, 2002

Coronel Carlos, Steven Morris, Database Systems: Design, Implementation, & Management, 2012

Gupta G.K., Database Management Systems, 2018

Halpin T.A., Terence Aidan Halpin, Information Modeling and Relational Databases, 2001

Halpin Terry, Tony Morgan, Information Modeling and Relational Databases, 2010

App diagrams, <https://app.diagrams.net/>, datum pristupanja 21.09.2020

Cs Uregina, <http://www2.cs.uregina.ca/~bernatja/crowsfoot.html>, datum pristupanja 15.07.2020,

Pavlić Mile, Alen Jakupović, Sanja Čandrlić, Modeliranje procesa, Odjel za informatiku Sveučilišta u Rijeci, Rijeka, 2014

Pavlić Mile, Informacijski sustavi, Odjel za informatiku Sveučilišta u Rijeci, Rijeka, 2009

Pavlić Mile, Oblikovanje baza podataka, Odjel za informatiku Sveučilišta u Rijeci, Rijeka, 2011

Powell Gavin.; Beginning Database Design, 2006

Silverstone Len, Agnew Paul, The Data Model Resource Book: Volume 3: Universal Patterns for Data Modeling, 2009

Wikipedia BPM, https://en.wikipedia.org/wiki/Business_Process_Model_and_Notation, datum pristupanja, 20.09.2020

Wikipedia CASE, https://en.wikipedia.org/wiki/Computer-aided_software_engineering, datum pristupanja ,18.09.2020

Wikipedia OMG, https://en.wikipedia.org/wiki/Object_Management_Group, datum pristupanja, 18.09.2020

Wikipedia IDEF1X, <https://en.wikipedia.org/wiki/IDEF1X>, datum pristupanja ,15.09.2020

Wikipedia UML, https://en.wikipedia.org/wiki/Unified_Modeling_Language, datum pristupanja, 10.09.2020

Wikipedia CLASS, https://en.wikipedia.org/wiki/Class_diagram, datum pristupanja, 10. 09. 2020