

# Usporedba MIRIS i IDEF metodologija

---

**Kirinčić, Andrea**

**Undergraduate thesis / Završni rad**

**2018**

*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:195:185127>

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

*Download date / Datum preuzimanja:* **2024-12-26**



*Repository / Repozitorij:*

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



Sveučilište u Rijeci – Odjel za informatiku

Preddiplomski jednopredmetni studij Informatike

Andrea Kirinčić

# Usporedba MIRIS i IDEF metodologija

Završni rad

Mentorica: Dr. sc. Martina Ašenbrener Katić

Rijeka, 21.9.2018

Rijeka, 12.03.2018.

## Zadatak za završni rad

Pristupnik: Andrea Kirinčić

Naziv završnog rada: Usporedba MIRIS i IDEF metodologija

Naziv završnog rada na eng. jeziku: Comparison of MIRIS and IDEF methodologies

Sadržaj zadatka:

Prilikom projektiranja i izgradnje poslovnih informacijskih sustava javlja se potreba za upotrebom određenih jasnih, sveobuhvatnih i standardiziranih metoda za njihovo modeliranje. Zadatak završnog rada je usporediti dvije metodologije za projektiranje i izgradnju informacijskih sustava odnosno potrebno je opisati Specijaliziranu metodologiju za razvoj informacijskog sustava (MIRIS) te Metodologiju IDEF (Integrated Computer Aided Manufacturing Definition). Poseban naglasak staviti će se na metode za modeliranje podataka odnosno metodu entiteta i veza (iz metodologije MIRIS) te IDEF1x (iz metodologije IDEF) koje će biti uspoređene na praktičnom primjeru.

Mentor

Dr. sc. Martina Ašenbrener Katić

*Martina Ašenbrener Katić*

---

Voditelj za završne radove

Dr. sc. Miran Pobar

*Miran Pobar*

---

Zadatak preuzet: 23.03.2018.

*Andrea Kirinčić*

---

(Andrea Kirinčić)

# Sadržaj

<b>Sažetak</b> .....	<b>4</b>
<b>Ključne riječi</b> .....	<b>5</b>
<b>Kazalo</b> .....	<b>6</b>
<b>Uvod</b> .....	<b>7</b>
<b>1 Metodologija MIRIS</b> .....	<b>8</b>
<b>1.1 Model Procesa</b> .....	<b>11</b>
1.1.1 Metoda Dekompozicije .....	11
1.1.2 Dijagram toka podataka .....	12
<b>1.2. Metoda entiteti-veze</b> .....	<b>17</b>
1.2.1 Koncepti strukture metode entiteti – veze .....	18
<b>2 Metodologija IDEF</b> .....	<b>28</b>
<b>2.1 Povijest metodologije IDEF</b> .....	<b>28</b>
<b>2.2 Metode IDEF metodologije</b> .....	<b>28</b>
2.2.1 IDEF0 .....	29
2.2.2 IDEF1 .....	29
2.2.3 IDEF2 .....	30
2.2.4 IDEF3 .....	30
2.2.5 IDEF4 .....	30
2.2.6 IDEF5 .....	31
2.2.7 IDEF6 .....	31
2.2.8 IDEF7 .....	31
2.2.9 IDEF8 .....	32
2.2.10 IDEF9.....	32
2.2.11 IDEF10.....	32
2.2.12 IDEF11.....	32
2.2.13 IDEF12.....	32
2.2.14 IDEF13.....	32
2.2.15 IDEF14.....	32
2.2.16 Metoda IDEF1X.....	33
<b>3 Usporedba metode entiteti-veze metodologije MIRIS s metodom IDEF1X metodologije IDEF50</b>	
<b>3.1 Usporedba koncepata EV metode i metode IDEF1X</b> .....	<b>50</b>
3.1.1 Entitet.....	50
3.1.2 Domena i tip vrijednosti .....	51
3.1.3 Atribut .....	52
3.1.4 Veza .....	53
3.1.5 Primarni i vanjski ključ.....	53
<b>3.2 Usporedba koncepata metode entiteta i veza i metode IDEF1X</b> .....	<b>54</b>
<b>4 Metode entiteti-veze i IDEF1X na praktičnom primjeru</b> .....	<b>56</b>
<b>4.1 Opis sustava</b> .....	<b>56</b>
<b>4.2 Priloženi dokumenti s pripadnim opisom i dijagramima podataka</b> .....	<b>58</b>
<b>Zaključak</b> .....	<b>71</b>
<b>Literatura</b> .....	<b>72</b>
<b>Popis slika</b> .....	<b>73</b>
<b>Popis tablica</b> .....	<b>74</b>

## Sažetak

U ovom radu ću pisati o metodologijama MIRIS i IDEF, te njihovoj usporedbi s naglaskom na metode entiteta i veza (metodologije MIRIS) i metode IDEF1x (metodologije IDEF). Bit će navedene najvažnije definicije i pojmovi kao i opis i primjena svake metodologije. Fokusirat ću se na pojedinačne metode unutar svake metodologije kao što je izrada i opis dijagrama toka podataka koji su dio glavnog projekta kod metodologije MIRIS. Zatim ću govoriti o nastanku dijagrama entiteta i veza kao dio izvedbenog projekta MIRIS-a. U nastavku će biti opisana i metodologija IDEF, najvažnije metode koje su dio te metodologije, s naglaskom na metodu IDEF1X koju ću usporediti s metodom za modeliranje podataka odnosno metodom entiteta i veza koja je dio metodologije MIRIS. Na kraju ću prikazati primjenu obje metode na konkretnom primjeru sustava.

# Ključne riječi

**Metodologija**

**MIRIS**

**IDEF**

**Metoda entiteti – veze**

**IDEF1X**

# Kazalo

- Agregacija, 26, 53
- Atribut, 18, 21, 42, 52
- ATRIBUT, 52
- Bazična domena, 35, 51
- Dekompozicija**, 11
- Dijagram Entiteti – Veze**, 18
- Dijagram toka podataka, 12
- dokument**, 14, 15, 20, 66
- Domena**, 35
- Entitet, 18, 20, 33, 34, 36, 37, 39, 50
- Entity – Relationship*, 17, 18
- Glavni projekt**, 8
- IDEF, 1, 7, 28, 29, 33, 52, 53, 71
- IDEF0, 28, 29
- IDEF1, 29
- IDEF10**, 32
- IDEF11**, 32
- IDEF12, 32
- IDEF13, 32
- IDEF14, 32
- IDEF1X, 17, 28, 33, 34, 36, 37, 38, 47, 50, 51, 52, 53, 71
- IDEF2**, 30
- IDEF3, 30
- IDEF4, 30, 31
- IDEF5, 31
- IDEF6**, 31
- IDEF7, 31
- IDEF8**, 32
- IDEF9**, 32
- Identifikacijska veza, 39
- Izvedbeni projekt**, 8
- Jak tip entiteta**, 24
- Klasifikacija**, 20
- Lista vrijednosti**, 35, 52
- Metoda Entiteti – Veze**, 17
- Metodologija, 8, 28, 71
- Mile Pavlić, 8
- MIRIS, 1, 7, 8, 10, 17, 50, 51, 52, 53, 71
- Model procesa, 11
- modeliranje, 8, 12, 17, 28, 29, 33
- modliranje, 17
- Narudžba dobavljaču**, 59, 60
- Ne-identifikacijska veza, 39, 53
- Ne-specifična veza, 33, 43
- Opcionalna ne-identifikacijska veza, 39, 40
- POGLED**, 47
- Ponikve Krk, 56
- PONIKVE VODA d.o.o., 56
- Primarni (glavni) ključ**, 21
- Primarni ključ**, 45, 52, 53
- Primary key*, 21, 45
- Proces**, 13, 16, 28
- Proizvodnja softvera**, 8
- projekt**, 10, 39, 44
- Raspon vrijednosti**, 35, 52
- Slab tip entiteta, 18
- Slabi tip entiteta**, 24
- Specifična veza spoja, 38, 40
- Spremište (skladište) podataka**, 13, 15
- Spremište podataka, 15
- Strateško planiranje**, 8
- sustav, 7, 8, 11, 12, 13, 14, 29, 30, 33, 36, 56
- tip entiteta, 18, 25
- tip veze, 18, 22, 23, 24, 25, 26
- Tip veze, 22, 23, 24, 25, 26
- TIP VRIJEDNOSTI, 51
- Tok pdataa**, 14
- Tok podataka**, 13, 14, 15
- Upisna domena, 35
- Uvođenje**, 10
- VANJSKI KLJUČ, 46
- Vanjski sustav**, 13
- Veza, 18, 22, 25, 33, 40, 41, 44, 53
- VEZA KATEGORIZACIJE, 41

# Uvod

Razvojem informacijske tehnologije, razvijaju se i mnoge metode koje su namijenjene informatizaciji poslovnih procesa, smanjenju broja nepotrebnih dokumenata kako bi se povećala učinkovitost rada i unaprijedilo poslovanje. S obzirom kako nam je za dobro poslovanje i organizaciju sustava važan informacijski sustav razvilo se niz metoda koje su namijenjene što boljem, kvalitetnijem, ali i bržem razvoju takvih sustava.

Jedni od važnijih pristupa koji su se nastali u području razvoja i nadogradnje kao i modeliranja podataka informacijskih sustava su metodologije MIRIS i IDEF.



# 1 Metodologija MIRIS

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 (Pavlic, Informacijski sustavi, 2009).

Životni ciklus projektiranja informacijskih sustava možemo podijeliti u 3 faze (Pavlic, Informacijski sustavi, 2009):

1. Faza: apstraktno modeliranje sustava i podjela sustava na podsustave
2. Faza: modeliranje podsustava i određivanje odgovarajuće metode za modeliranje procesa
3. Faza: modeliranje podataka procesa i definicija arhitekture aplikacije prema modelu procesa i modelu podataka

MIRIS koristi tri osnovne metode: metodu za modeliranje podataka, metodu za modeliranje procesa i metodu za modeliranje aplikacija.

Faze razvoja informacijskih sustava grupirane su u 2 skupine:

- logičko oblikovanje (projektiranje IS)
- fizičko oblikovanje (izgradnja IS)

Postoji šest faza razvoja IS-a od kojih se svaka sastoji od aktivnosti (Pavlic, Informacijski sustavi, 2009):

## 1. Strateško planiranje informacijskog sustava (SP)

- 1.1. Analiza: Definiranje i obuka tima, dekompozicija procesa, popis dokumentacije i kretanje kroz sustav
- 1.2. Podsustavi: Određivanje podsustava i veza
- 1.3. Prioriteti: Određivanje prioriteta
- 1.4. Resursi: Definiranje cjelovite infrastrukture
- 1.5. Plan: Planiranje glavnih projekata i aktivnosti

## 2. Glavni projekt (GP)

- 2.1. PZ: Izrada projektnog zadatka
- 2.2. DTP: Intervjuiranje, raščlanjivanje i modeliranje procesa (DTP)
- 2.3. Procesi GP: Analiza procesa, problema i prijedloga poboljšanja
- 2.4. Podaci GP: Opisivanje podataka
- 2.5. Plan GP: Planiranje izvedbenih projekata
- 2.6. Resursi GP: Definiranje modela resursa glavnog projekta

## 3. Izvedbeni projekt (IP)

- 3.1. DEV: Intervjuiranje, apstrakcija i modeliranje podataka (EV)
- 3.2. Prevođenje: Prevođenje modela podataka u shemu BP (RM)
- 3.3. Arhitektura IP: Definiranje arhitekture programskog proizvoda (APP)
- 3.4. Operacije IP: Projektiranje operacija nad shemom BP

## 4. Proizvodnja softvera (PS)

- 4.1. *Planiranje proizvodnje*
  - 4.1.1. Planiranje aktivnosti proizvodnje SW
  - 4.1.2. Određivanje izvršitelja za pojedine zadatke i određivanje rokova
  - 4.1.3. Određivanje i kreiranje produkcijske, testne i razvojne okoline
- 4.2. *Oblikovanje baze podataka*
  - 4.2.1. Prevođenje logičkog modela podataka u fizički model sheme baze podataka
  - 4.2.2. Kreiranje razvojne okoline za svakog pojedinog programera
  - 4.2.3. Punjenje sheme baze podataka u razvojnoj okolini iz postojeće BP
  - 4.2.4. Dodavanje novih koncepata iz modela podataka u razvojnu shemu BP (tip entiteta i dr.)
  - 4.2.5. Kreacija razvojne baze podataka
  - 4.2.6. Inicijalno punjenje testne baze podataka
- 4.3. *Razvoj programskog proizvoda*
  - 4.3.1. Izgradnja glavnog izbornika (aplikacijskog stabla) ili dorada već postojećeg
  - 4.3.2. Izrada ekrana za pregled redaka svake tablice po jednom ili više ključeva
  - 4.3.3. Izrada ekrana za operacije nad jednim retkom tablice (unos, izmjena, brisanje i pregled)
  - 4.3.4. Izrada programskih modula različitih vrsta i namjena: obračuna, procedura, funkcija kontrola, *look-upova* nad tablicama (s prvim testiranjem modula)
  - 4.3.5. Izrada izvještaja (s prvim testiranjem modula)
- 4.4. *Testiranje u testnoj okolini*
  - 4.4.1. Prijenos razvijenih programskih modula u testno okruženje
  - 4.4.2. Spajanje novih modula s postojećim
  - 4.4.3. *Back-up* verzija softvera
  - 4.4.4. Testiranje prototipa softvera nad testnom bazom podataka
  - 4.4.5. Ažuriranje planova proizvodnje softvera
  - 4.4.6. Izvođenje prema potrebi aktivnosti iz ranijih skupina aktivnosti i ponovno testiranje
- 4.5. *Testiranje i ispravljanje u radnoj okolini*
  - 4.5.1. Prijenos razvijenih programskih modula u radno okruženje
  - 4.5.2. Spajanje novih modula s postojećima
  - 4.5.3. *Back-up* verzija softvera
  - 4.5.4. Punjenje baze podataka
  - 4.5.5. Testiranje prototipa softvera nad produkcijskom bazom podataka od strane programera
  - 4.5.6. Ažuriranje planova proizvodnje softvera
  - 4.5.7. Izvođenje prema potrebi aktivnosti iz ranijih skupina aktivnosti i ponovno testiranje
- 4.6. *Testiranje od strane korisnika*
  - 4.6.1. Prezentacija softvera korisniku
  - 4.6.2. Testiranje od strane korisnika
  - 4.6.3. Izrada popisa primjedbi korisnika
  - 4.6.4. Ažuriranje planova proizvodnje softvera

4.6.5. Izvođenje prema potrebi aktivnosti iz ranijih skupina aktivnosti i ponovno testiranje

4.6.6. Izrada zapisnika o testiranju i prihvaćanju faze uvođenja

## **5. Uvođenje (UVO)**

5.1. Instalacija gotovog softvera na produkcijsko okruženje (kod korisnika)

5.2. Izrada uputa

5.3. Presentacija gotovog softvera

5.4. Obuka

5.5. Završne konverzije

5.6. Završno testiranje

5.7. Početak primjene nove aplikacije

5.8. Uspostava novog sustava i potpisivanje primopredajnog zapisnika

## **6. Primjena i održavanje (ODR)**

6.1. Podešavanje novog aplikacijskog sustava

6.2. Izvješće o procjeni novog projekta

6.3. Raspodjela odgovornosti korisnika i programera

6.4. Korištenje aplikacijskog sustava

6.5. Postavljanje zahtjeva za izmjenama od strane korisnika

Izgradanja informacijskog sustava počinje izradom strateškog plana, potom u fazi Glavnog projekta se analizira poslovanje i kreira model podataka.

Faze metodologije MIRIS se mogu preklapati i tijekom razvoja moguće se vratiti na bilo koju aktivnost, te je izmijeniti. Na taj način se prikuplja znanje o sustavu i proširuje model sustava, te upotpunjuje vizija potrebna za razvoj informacijskog sustava (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesu, 2014).

## 1.1 Model Procesa

Model procesa sastoji se od više različitih dijagrama i dokumenata koji opisuju sadržaj modela procesa (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesa, 2014).

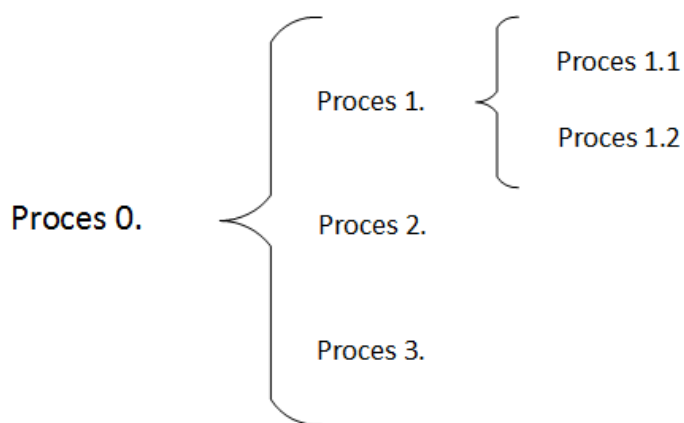
### 1.1.1 Metoda Dekompozicije

**Dekompozicija** je metoda opisivanja sustava od konceptualno općih do detaljnih problema, tj. od cjeline ka dijelovima sustava (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesa, 2014).

Dekompozicijom se složeni sustav rastavlja na više jednostavnijih podsustava koji se dalje mogu dekomponirati. Na taj način dolazimo do hijerarhijskog opisa sustava. Hijerarhijski možemo sustav opisati pomoću više modela pri čemu jedan model prikazuje cijeli sustav, dok ga drugi model još detaljnije opisuje. Postupkom dekompozicije dijelimo sustav na njegove podsustave, dok te podsustave možemo i dalje raščlanjivati i na taj način dolazimo do opisa sustava na raznim razinama detaljnosti, tj. dolazimo do hijerarhijskog opisa sustava (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesa, 2014).

Dekompozicijski dijagrami, tj. dijagrami strukturalne raščlane daju prikaz cijele hijerarhije sustava. Takvi dijagrami opisuju sustav kao skup njegovih dijelova. Postoji više oblika dijagrama dekompozicije, međutim svi koriste istu tehniku raščlanjivanja. Dekompozicija je postupak raščlanjivanja sustava od općeg prema pojedinačnom, te onda od pojedinačnog prema općem (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesa, 2014).

Slika 1. prikazuje dijagram dekompozicije Proces 0. na potprocese Proces 1., Proces 2. i Proces 3. Proces 1. smo još dekomponirali na procese: Proces 1.1 i Proces 1.2.



Slika 1 - dijagram dekompozicije

### 1.1.2 Dijagram toka podataka

**Dijagram toka podataka** (engl. *Dataflow diagram*, skraćeno DTP) je grafičko sredstvo za modeliranje i prezentaciju procesa sustava (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesu, 2014).

DTP je uveo De Marco 1978. godine kao proširenje tehnike „ulaz – proces – izlaz“ (engl. *Input – Process – Output*, skraćeno IPO) i tehnike HIPO (engl. *hierarchial IPO*) tvrtke IBM. Ideja metode HIPO je promatrati sustav i uočiti jedan ili nekoliko ulaza kojima se koristi neki proces, te jedan ili nekoliko izlaza iz tog procesa.

Važne odlike DTP-a (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesu, 2014):

- DTP je grafički prikaz procesa sustava
- Korisnik i analitičar zajedno grade model sustava
- Precizno se definiraju zahtjevi korisnika
- DTP je jezik za komunikaciju korisnika i analitičara
- Broj različitih grafičkih simbola je malen (ima četiri osnovna simbola)

Dijagram toka podataka sastoji se od (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesu, 2014):

1. Ulaznih i izlaznih tokova podataka (izvještaja, dokumenata) koje sustav dobiva ili šalje u okolinu
2. vanjskih objekata (organizacija, drugih sustava) koji šalju podatke prema sustavu ili primaju tokove podataka od sustava
3. procesa sustava (programa, aktivnosti, podsustava) koji transformiraju ulazne u izlazne tokove podataka i
4. spremišta podataka (baze podataka) u kojima se čuvaju podaci potrebni za izvođenje procesa ili dobiveni kao rezultat rada procesa.

DTP je osnovna metoda analize za izradu modela procesa. DTP specificira što sustav radi. Osnovni cilj procesne analize (modeliranja procesa) je određivanje logičkog toka podataka kroz sustav. Procesi na dijagramu toka podataka se paralelno izvršavaju u stvarnom sustavu. DTP nije blok-dijagram (Flow-chart, dijagram toka programa) koji opisuje redoslijed izvršavanja nekog programa. Kod DTP-a se detaljno pomoću grafičkih simbola može opisati sustav, kako bi se mogao lakše, brže i u potpunosti shvatiti (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesu, 2014).

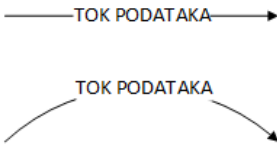


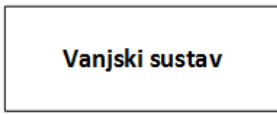
#### 1.1.2.1 Koncepti dijagrama toka podataka

DTP koristi grafičke simbole za opis elemenata sustava. Grafički simboli od kojih se gradi DTP se nazivaju konceptima. Dijagram toka podataka koristi grafičke simbole po Yourdonu.

Skup osnovnih koncepata DTP-a su (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesu, 2014):

- **Tok podataka** – označava se sa strelicom od izvorišta prema odredištu podataka na koju se upisuje naziv toka
- **Proces (funkcija)** – označava se elipsom u koju upisujemo naziv procesa
- **Spremište (skladište) podataka** – naziv skladišta podataka upisujemo između dvije linije
- **Vanjski sustav** – naziv vanjskog sustava upisujemo u pravokutnik

Slika 2. prikazuje koncepte dijagrama toka podataka zajedno sa njihovim grafičkim simbolima.

TOK PODATAKA	
PROCES (funkcija)	
SPREMIŠTE (skladište) PODATAKA	
VANJSKI SUSTAV	

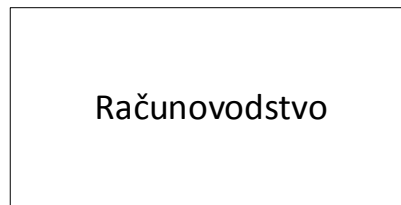
Slika 2 - Koncepti dijagrama toka podataka (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesu, 2014)

#### 1.1.2.1.1 Vanjski sustav

**Vanjski sustav** (VS) je koncept DTP-a koji predstavlja sustave koji je su vezi sa promatranim sustavom, kao izvori ili odredišta podataka, a čiji procesi nam nisu važni za promatrani sustav.

Svi procesi koji ne pripadaju promatranom sustavu smatramo vanjskim sustavom. Vanjski sustav je neki proces izvan promatranog sustava. Vanjski sustav može biti: poduzeće, osoba (šef, klijent, korisnik), organizacijska jedinica (skladište, podsustav plaća), itd. Vanjski sustav se prikazuje pravokutnikom, u koji upisujemo naziv sustava. Naziv vanjskog sustava je najčešće imenica (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesu, 2014).

Na slici 3. je prikazan primjer vanjskog sustava Računovodstvo:



Slika 3 – Vanjski sustav

#### 1.1.2.1.2 Tok podataka

**Tok podataka** je skup podataka koji se kreću od jednog dijela sustava prema drugom, te ih povezuju.

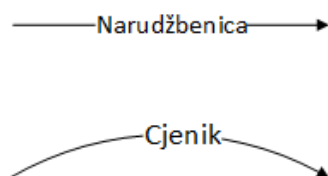
Tok podataka se na dijagramu prikazuje sa ravnom ili zaobljenom crtom na kojima se prikazuje njihov naziv, sa ili bez strelice koja označuje smjer kretanja podataka. Naziv toka može biti imenica ili kombinacija imenica i pridjeva. Naziv toka podataka treba prikazivati podatke od kojih se tok sastoji (Na primjer, narudžba dobavljaču, cjenik). Svaki tok podataka treba imati jedinstveno ime u cijelom dijagramu. Tokovi prema spremištu podataka mogu imati isti naziv kao i spremište, te se ne trebaju imenovati na dijagramu, ali to vrijedi samo ako je sadržaj toka podataka jednak sadržaju spremišta (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesu, 2014).

Tok može biti (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesu, 2014):

- dokument u sustavu, Na primjer, narudžba, radni list, putni nalog,...
- skup dokumenata kao: plan proizvodnje, projektna dokumentacija, itd.
- elementarni podatak (Na primjer, jedinstveni matični broj građana)
- neformatizirani skup podataka kao: knjiga o projektiranju IS-a, neki časopis
- bilo koje znanje, kao npr. znanje eksperta o važno za izvršenje nekog procesa, telefonski ili usmeno prenijet skup podataka, itd.

Tok podataka povezuje komponente DTP-a, što znači da predstavlja vezu u sustavu, odnosno povezuje druge elemente modela. Možemo imati ulazni ili izlazni tok iz drugih elemenata DTP-a (vanjski sustav, spremište podataka ili proces) (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesu, 2014).

Na slici 4. se nalazi primjer toka podataka, tok podataka se može crtati ravnom ili zaobljenom crtom koja ide od izvorišta prema odredištu podataka.



Slika 4 - Tok podataka (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesu, 2014)

### 1.1.2.1.3 Spremište podataka

**Spremište (skladište) podataka** je koncept DTP-a koji služi za prikaz grupa podataka. U spremištu su podaci pohranjeni gdje se čuvaju za kasnije procese obrade. Spremište podataka može biti jedan ili više tokova podataka u mirovanju (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesu, 2014).

Spremište podataka može biti (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesu, 2014):

1. **dokument** kao: upit, ponuda, izvještaj, itd.
2. **skup dokumenata** kao: plan proizvodnje, projektna dokumentacija, itd.
3. **datoteka (file) ili baza podataka (BP)**, na primjer: neki skup podataka na računalu, izvještaj, relacijska BP.
4. **neformatizirani skup podataka** kao: knjiga o projektiranju IS-a, časopis InfoTrend,...
5. **znanje u ljudskom umu**
6. **biblioteka** (časopisi, knjige, dokumenti,...)
7. **skriveni skup podataka** (tajni zaštićeni podaci u BP)
8. **bilo koje znanje**, npr: znanje eksperta važno za izvršavanje procesa ili podaci iz sveukupnog znanja izvršitelja nekog posla, telefonski ili usmeno prenijet skup podataka.

Spremište podataka se grafički predstavlja sa dvije paralelne linije, dok se naziv spremišta upisuje između linija. Ime spremišta se određuje kao i ime toka podataka. Naziv spremišta podataka je imenica koja određuje ime objekta kojeg to spremište opisuje (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesu, 2014).

Na slici 5. se nalazi primjer Spremišta podataka (e-Baza podataka).



Slika 5 - spremište podataka

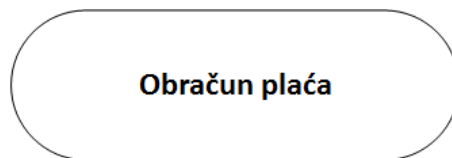


#### 1.1.2.1.4 Proces

**Proces** je skup povezanih aktivnosti i odluka preko kojih objekti sustava ostvaruju dijelove cilja svog postojana, a za njihovo izvršenje su potrebni određeni resursi i određeno vrijeme (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesu, 2014).

Proces je transformacija ulaznog u izlazni tok podataka na svrhovit i koherentan (međusobno povezan) način. Svaki proces treba imati najmanje jedan ulazni i najmanje jedan izlazni tok. Proces se prikazuje krugom ili elipsom unutar kojega upisujemo naziv procesa. Naziv može biti glagol, glagolska imenica ili skup riječi koji opisuje proces, kao npr. Obračun plaća, Skladištenje, itd. Svaki element dijagrama toka podataka treba imati jedinstven naziv (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesu, 2014).

Na slici 6. je prikazan primjer procesa Obračun plaća.



Slika 6 - Proces

## 1.2. Metoda entiteti-veze

**Metoda entiteti – veze (EV)** (engl. *Entity – Relationship model*, skraćeno ER) je metoda za modeliranje podataka. Model Entiteti – Veze je grafički prikaz međusobno povezanih grupa podataka promatranog sustava. EV je semantički bogata metoda za modeliranje podataka. Koncepti ove metode su bliski korisniku, pa je shema podataka jednostavna za razumijevanje i komunikaciju korisnika i projektanta (Pavlic, Oblikovanje baza podataka, 2011).

Metodu Entiteti-Veze je razvio američki znanstvenik i profesor Dr. Peter Pin-Shan Chen koju je objavio u znanstvenom članku 1976. godine. Peter Chen je na „1st International Conference on Very Large Databases“ 1975. godine predstavio prvi rad na temu EV metode koji se smatra jednim od najutjecajnijih radova u informatičkoj povijesti (Pavlic, Oblikovanje baza podataka, 2011).

Razvojem EV-a nastalo je više varijanti te metode i ona je postala jedna od najčešće korištenih metoda za modeliranje podataka. Tako se EV koristi u specijaliziranim metodologijama: CASE\*Method, MIRIS, IEM, itd. Za modeliranje podataka postoje i često korištene standardizirane notacije kao što su ISO UML (dijagrami klasa) i IEEE IDEF1X. Varijanta EV metode se koristi i u metodici MIRIS. Metoda EV i pomoću nje izgrađen dijagram Entiteti – Veze služi za izgradnju dvaju dijelova informacijskog sustava: shemu baze podataka (BP) i arhitekturu programskog proizvoda. Procesom modeliranja podataka metodom EV dolazimo do modela podataka čiji su podaci u 5. normalnoj formi bez zahtjevnoga procesa normalizacije, te je ova metoda značajna za projektiranje baza podataka (Pavlic, Oblikovanje baza podataka, 2011).

### 1.2.1 Koncepti strukture metode entiteti – veze

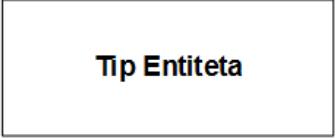

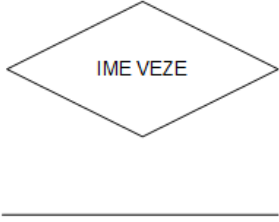
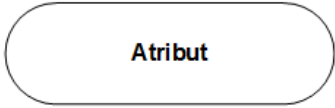

Osnovni koncepti od kojih se gradi struktura modela Entiteti – Veze su (Pavlic, Oblikovanje baza podataka, 2011):

- Entitet i tip entiteta
- Veza i tip veze
- Atribut tipa entiteta
- Slab tip entiteta i specijalni tipovi veza
- Agregirani tip entiteta
- Povratni tip veze
- Generalizacijski tip veze

Model podataka metodom Entiteti – Veze gradi se upotrebom grafičkih simbola.

**Dijagram Entiteti – Veze** (*engl. Entity – Relationship Diagram*) je grafički prikaz modela podataka sustava, metodom EV. Kako bi pravilno izgradili dijagram EV definiran je skup pravila kombiniranja koncepata. Pravila detaljnije određuju semantiku koncepata strukture dijagrama EV (Pavlic, Oblikovanje baza podataka, 2011).

Osnovni simboli dijagrama Entiteti – Veze prikazani su na slici 7. (Pavlic, Oblikovanje baza podataka, 2011):

KONCEPT	SIMBOL
TIP ENTITETA	
SLAB TIP ENTITETA	
TIP VEZE	
ATRIBUT	
AGREGACIJA	

Slika 7 - koncepti dijagrama Entiteta – Veze

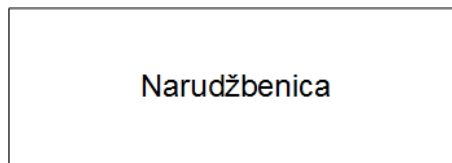
### 1.2.1.1 Entitet

Osnovni pojam metode entiteti-veze je Entitet. **Entitet** (*engl. Entity*) je dio stvarnosti što možemo jednoznačno odrediti. Entitet može biti stvaran predmet, događaj, osoba, apstraktni pojam, dokument ili nešto drugo (Pavlic, Oblikovanje baza podataka, 2011).

Primjeri entiteta mogu biti: Sunce, grad Zagreb, osoba Ana Anić rođena u Rijeci, neki račun, narudžbenica, pojedini kupac, neki konkretan PC, itd. Slični entiteti koji imaju zajednička svojstva se klasificiraju u **tipove entiteta**.

**Klasifikacija** je postupak u kojem se slični entiteti objedinjuju u jedan zajednički skup nazvan tip entiteta. **Tip entiteta** je skup sličnih pojedinačnih entiteta dobiven procesom klasifikacije, npr: Radnik, Organizacijska jedinica, Dobavljač, itd. Tip entiteta se predstavlja pravokutnikom proizvoljnih dimenzija u koji se upisuje naziv tipa entiteta. Jedan tip entiteta sastoji se od više različitih entiteta istog tipa. Entitet možemo nazvati i pojedinačnim pojavljivanjem tipa entiteta. Pojedinačna pojavljivanja entiteta se ne prikazuju na modelu, ona se podrazumijevaju. Svaki tip entiteta može imati jedno ili više pojedinačnih pojavljivanja (Pavlic, Oblikovanje baza podataka, 2011).

Na slici 8. je prikazan primjer tipa entiteta Narudžbenica.



Slika 8 - Tip entiteta

### 1.2.1.2 Atribut

**Atribut** (*engl. Attribute*) tipa entiteta je funkcija koja preslikava tip entiteta u tip vrijednosti ili u kartezijev produkt tipa vrijednosti. Svaki entitet ima niz svojstava (karakteristika, odlika). Atribut je imenovana karakteristika (svojstvo) nekog entiteta. Atribut je funkcija koja pridružuje entitetu neku vrijednost iz tipa vrijednosti. Tip vrijednosti je povezan s tipom entiteta preko atributa. Primjeri atributa mogu biti: boja, datum rođenja, ime radnika, šifra dobavljača, itd. (Pavlic, Oblikovanje baza podataka, 2011).

Atribut daje tipu vrijednosti interpretaciju preko pojavljivanja entiteta u tipu entiteta, tj. prevodi podatak u informaciju. Na primjer: atribut Prezime pridružuje jedno pojavljivanje tipa entiteta Radnik jednome podatku iz tipa vrijednosti Prezimena. Važno je jedinstveno imenovati attribute u modelu podataka. Naziv atributa se piše u elipsu za tip vrijednosti (Pavlic, Oblikovanje baza podataka, 2011).

Na slici 9. je prikazan primjer atributa Ime radnika.



Slika 9 – Atribut (Pavlic, Oblikovanje baza podataka, 2011)

Ključ tipa entiteta (identifikator entiteta) je skup atributa koji neovisno o vremenu ispunjava uvjet jedinstvenosti i neredundantnosti.

#### **Uvjet jedinstvenosti:**

Ne postoje dva pojedinačna pojavljivanja entiteta u tipu entiteta takva da imaju istu vrijednost atributa koji čine ključ i ne postoje dva tipa entiteta koji imaju isti skup atributa za ključ.

#### **Uvjet neredundantnosti:**

Ako se iz ključa izostavi bilo koji od atributa, uvjet neredundantnosti se gubi. Ne postoji niti jedan atribut kao dio ključa koji se može izostaviti, da se pritom uvjet jedinstvenosti ne gubi (Pavlic, Oblikovanje baza podataka, 2011).

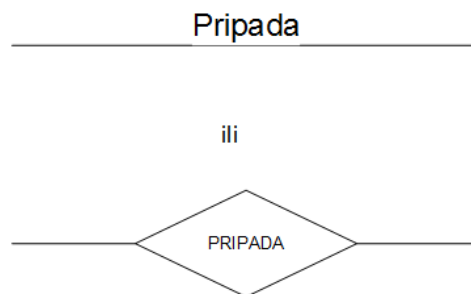
**Primarni (glavni) ključ** (*engl. Primary key*) je jedan od ključeva tipa entiteta koji je odabran zbog nekog važnog razloga da jedinstveno označava svaki entitet tog tipa entiteta. U EV metodi imamo samo jedan ključ za svaki tip entiteta i to primarni (glavni) ključ (Pavlic, Oblikovanje baza podataka, 2011).

### 1.2.1.3 Tip veze

**Veza** (asocijacija) (engl. *Relationship*) je pridruživanje između entiteta. Dva tipa entiteta mogu imati više pojedinačnih veza. Veza predstavlja povezanost među entitetima u stvarnosti ili u mislima. Primjer veze: „Otac Ivan ima sina Marka“, veza „ima“ povezuje entitete „Ivan“ i „Marko“. Dva tipa entiteta mogu imati više pojedinačnih veza – tako dolazimo do Tipa veze (Pavlic, Oblikovanje baza podataka, 2011).

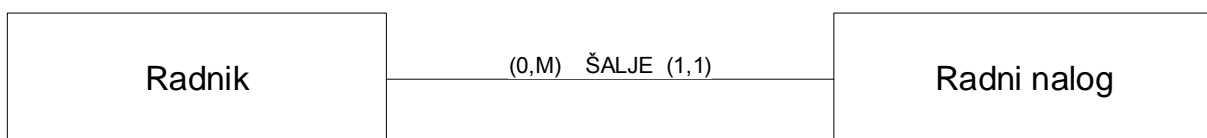
**Tip veze** (engl. *Relationship type*) je skup pojedinačnih veza između istih tipova entiteta. Tip veze se može prikazati pomoću romba ili linije, gdje u romb ili na liniju upisujemo naziv tipa veze. Tipovi veza mogu biti: isporučuje, ima, studira, itd. Tip veze se na dijagramu crta kao linija koja spaja tipove entiteta, a ne pojedinačna pojavljivanja tipa entiteta. Veza se još naziva jedno pojedinačno pojavljivanje tipa veze. **Red tipa veze** je broj tipova entiteta povezanih tim tipom veze. **Naziv tipa veze** je riječ ili grupa riječi koji predstavljaju radnju (glagoli, glagolske imenice) ili odnos među entitetima (imenice) koji nam jasno prikazuje povezanost među entitetima. Naziv tipa veze ne mora biti jedinstven u cijelom dijagramu, ali ne smije postojati još jedan tip veze između istih tipova entiteta s istim imenom. **Uloga** (engl. *role*) entiteta u vezi je imenovana funkcija koja pridružuje taj entitet nekom drugom entitetu (Pavlic, Oblikovanje baza podataka, 2011).

Na slici 10. su prikazana dva načina označavanja tipa veze na primjeru Pripada.



Slika 10 - Tip veze (Pavlic, Oblikovanje baza podataka, 2011)

Na slici 11. se nalazi primjer tipa veze šalje koja povezuje tipove entiteta Radnik i Radni nalog.



Slika 11 - Tip veze

### 1.2.1.3.1 Brojnost tipa veze

Tip veze sastoji se od dvaju preslikavanja (koja nazivamo uloge) između tipova entiteta. Svaki tip veze ime dvije uloge što znači da se sastoji od dva preslikavanja prvog tipa entiteta na drugi i obrnuto. Svako preslikavanje ograničeno je dopuštenim brojem povezanih entiteta. Svaka pojedina veza iz nekog tipa veze povezuje entitete dvaju tipova entiteta (Pavlic, Oblikovanje baza podataka, 2011).

**Brojnost tipa veze** je broj koji kaže koliko entiteta pojedinog tipa entiteta sudjeluje u tipu veze s entitetom iz drugog tipa entiteta (Pavlic, Oblikovanje baza podataka, 2011).

Moguće je odrediti donju (DG) i gornju granicu (GG) brojnosti za obje uloge, tj.:

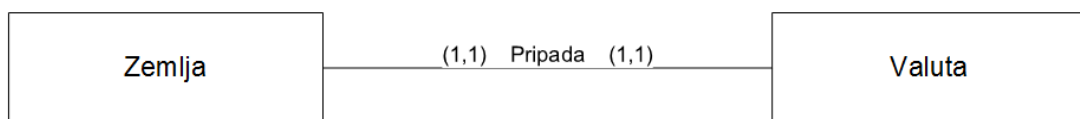
- U koliko će se najmanje i najviše pojedinačnih pojavljivanja tipa entiteta E preslikati pojavljivanje tipa entiteta  $E_1$ ?
- U koliko će se najmanje i najviše pojedinačnih pojavljivanja tipa entiteta  $E_1$  preslikati pojavljivanje tipa entiteta E?

Oba broja, najmanji i najveći broj preslikavanja, upisujemo na dijagram, odvojene zarezom i upisane u zagrade kao uređeni par (DG,GG). Uređeni par (DG,GG) nazivamo brojnost preslikavanja uloge. Brojnost preslikavanja tipa veze su dva uređena para oblika (DG,GG) : (DG, GG), pri čemu prvi par pripada jednoj ulozi, a drugi drugoj ulozi (Pavlic, Oblikovanje baza podataka, 2011).

Neki tipovi brojnosti su (Pavlic, Oblikovanje baza podataka, 2011):

- (1,1) : (1,1) – jedan entitet prvog tipa pridružuje se najmanje nula ili najviše jednom entitetu drugog tipa i obrnuto.  
Npr: Jedna valuta pripada najmanje jednoj i najviše jednoj zemlji, dok jednoj zemlji pripada najmanje jedna i najviše jedna valuta.

Slika 12. prikazuje vezu pripada među tipovima entiteta Zemlja i Valuta.



Slika 12 - Primjer brojnosti veze (Pavlic, Oblikovanje baza podataka, 2011)

- (1,1) : (0,M) – Jedan entitet prvog tipa pridružuje sa najmanje nijednom i najviše jednom entitetu drugog tipa, a jedan entitet drugog tipa može se pridružiti najmanje nula i najviše mnogo entiteta prvog tipa.
- (0,M) : (0,M) – jedan entitet prvog tipa može se pridružiti najmanje nula i najviše mnogo entiteta drugog tipa i obrnuto.

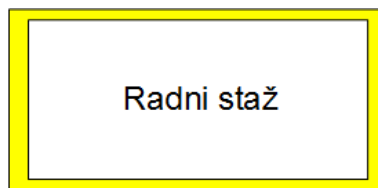


#### 1.2.1.4 Jak i slab tip entiteta

**Jak tip entiteta** (ili samo Tip entiteta, engl. *Strong entity*) je tip entiteta koji ima vlastiti primarni ključ i nije ovisan o drugim tipovima entiteta u modelu podataka.

**Slabi tip entiteta** (engl. *weak entity*) je tip entiteta koji je na neki način zavisian drugom tipu entiteta, a ta zavisnost se prikazuje specijalnim tipom veze među jakim i slabim tipom entiteta. Na primjer, Pošiljka ovisi o Narudžbi, Grad ovisi o Državi, itd. Slabi tip entiteta crtamo sa dvostrukim pravokutnikom. (Pavlic, Oblikovanje baza podataka, 2011)

Na slici 13. je prikazan primjer slabog tipa entiteta Radni staž:



Slika 13 - Primjer slabog tip entiteta (Pavlic, Oblikovanje baza podataka, 2011)

**Specijalni tip veze** je tip veze koja povezuje dva tipa entiteta čiji entiteti su međusobno zavisni. Postoji više vrsta zavisnosti prema kojima se definira vrsta specijalnog tipa veze. (Pavlic, Oblikovanje baza podataka, 2011)

Vrste specijalnog tipa veze su (Pavlic, Oblikovanje baza podataka, 2011):

- egzistencijalna (E)
- identifikacijska (I)
- egzistencijalna i identifikacijska (E&I)
- veza prethođenja (P)
- generalizacijska veza (G)
- hijerarhijska (H)
- agregacijska (veza sastavljena od dvije E&I veze između triju tipova entiteta)

**Egzistencijalni tip veze** (E) je tip veze koji povezuje dva tipa entiteta od kojih postojanje jednog tipa entiteta ovisi o drugom. Tip entiteta koji egzistencijalno ovisi o jakom tipu entiteta je slabi tip entiteta. Npr: Tip veze Pripada povezuje Radnika i Staž. Staž je slabi tip entiteta jer pripada radniku, a veza Pripada je egzistencijalni tip veze (Pavlic, Oblikovanje baza podataka, 2011).

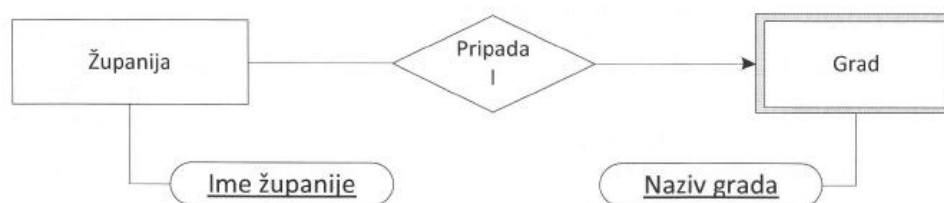
Na slici 14. je prikazan primjer egzistencijalnog tipa veze Pripada koji povezuje jaki tip entiteta Radnik sa slabim tipom entiteta Radni staž.



Slika 14 - Primjer egzistencijalnog tipa veze (Pavlic, Oblikovanje baza podataka, 2011)

**Identifikacijski tip veze (I)** je tip veze koji označava da se jedan tip entiteta u modelu ne može jedinstveno identificirati vlastitim ključem, već se treba koristiti 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 u BP samo s ključem Naziv grada (Pavlic, Oblikovanje baza podataka, 2011).

Na slici 15. je prikazan identifikacijski tip veze Pripada koji povezuje jaki tip entiteta Županija sa slabim tipom entiteta Grad.



Slika 15 - Primjer identifikacijskog tipa veze (Pavlic, Oblikovanje baza podataka, 2011)

**Egzistencijalni i identifikacijski tip veze (E&I)** je tip veze kod kojeg je jedan tip entiteta egzistencijalno i identifikacijski zavisan od drugog tipa entiteta (Pavlic, Oblikovanje baza podataka, 2011).

#### 1.2.1.5 Generalizacija i specijalizacija

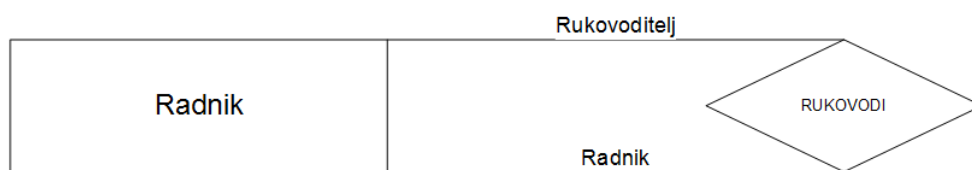
Neki tipovi entiteta  $E_1, E_2, \dots$  su **podtipovi** nekog tipa entiteta  $E$ , kojeg zovemo **nadtip** ako su sva pojavljivanja entiteta podtipova ujedno i pojavljivanja entiteta nad-tipa. Nad-tip i podtipovi su povezani **generalizacijskim tipom** veze. Podtipovi i nadtipovi se razlikuju u atributima. 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 romбом sa nazivom S od specijalizacija ili G od generalizacija (Pavlic, Oblikovanje baza podataka, 2011).

### 1.2.1.6 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 (Pavlic, Oblikovanje baza podataka, 2011).

Na slici 16. je prikazan povratni tip veze kod tipa entiteta radnik:



Slika 16 - Primjer povratne veze (Pavlic, Oblikovanje baza podataka, 2011)

### 1.2.1.7 Agregacija

**Agregacija** (mješoviti, agregirani tip entiteta) je apstrakcija u kojoj se tip veze između dva ili više tipova entiteta tretira kao novi tip entiteta (Pavlic, Oblikovanje baza podataka, 2011).

Tip veze postaje tip **agregirani tip entiteta** ako (Pavlic, Oblikovanje baza podataka, 2011):

- Pojavljuje se atribut tipa veze, tj. atribut ne može pripadati niti jednom pojedinačnom tipu entiteta
- Potrebno je međusobno povezati dva tipa entiteta, npr. veza *Student polaze ispit iz kolegija* treba biti povezana s vezom *Ispituje Profesor*
- Potrebno je razdvojiti višestruki u binarni tip veze, npr. veza Isporuka između entiteta *Roba, Dobavljač, Kupac, Proizvođač, Dostavljač i Skladište*.
- Brojnost tipa veze po gornjoj granici s obje strane iznosi M (na primjer, jedan radnik može poznavati više stranih jezika (M), dok jedan strani jezik može poznavati više radnika (M))

Agregacija može imati i svoje atribute, kao na primjer, stupanj poznavanja stranog jezika. Taj atribut ne možemo pridružiti Radniku jer bi to značilo da radnik ima isti stupanj poznavanja svih stranih jezika, niti stranom jeziku jer bi to značilo da svi radnici znaju strani jezik na jednakom stupnju. Zato Stupanj poznavanja pripada tipu veze *Poznaje*, tako možemo znati koji radnik poznaje koji strani jezik i u kojem stupnju. Agregacija se označava pravokutnikom u kojem je ucrtan romb. (Takav simbol predstavlja da se prvo radilo o tipu veze koji je postao tip entiteta, odnosno da se radi o mješovitom tipu entiteta i veze) (Pavlic, Oblikovanje baza podataka, 2011).

Na slici 17. je prikazan primjer agregacije Poznaje koja povezuje tipove entiteta Radnik i Strani jezik.



Slika 17 - Primjer agregacije (Pavlic, Oblikovanje baza podataka, 2011)

## 2 Metodologija IDEF

**IDEF**, skraćeno od ICAM Definition, kasnije preimenovano u Integration DEFinition, metodologija je koja se sastoji od više metoda za modeliranje na području sistemskog i softverskog inženjerstva. IDEF metode je razvilo američko ratno zrakoplovstvo, i danas se često koriste od strane američke vojske, ali su i javno dostupne. Najkorištenije metode su metoda za modeliranje funkcija: IDEF0 i IDEF1X koja se koristi za modeliranje informacija i baza podataka (Wikipedia, 2018).

IDEF je metodologija koja se sastoji od niza metoda specijaliziranih za određeno područje, s obzirom da ne postoji jedinstvena metoda ili jezik modeliranja koji bi podržao sve faze razvoja od specifikacije, analize do oblikovanja. U razvoju IDEF metoda nastojala se postići povezanost (integracija) više metoda u jednu, smislenu cjelinu (Pavlic, Informacijski sustavi, 2009).

### 2.1 Povijest metodologije IDEF

Zbog potrebe za semantičkim modelima podataka, sredinom 1970-ih godina američko zrakoplovstvo razvija program za računalno podržanu integriranu proizvodnju (engl. *Integrated Computer Aided Manufacturing, ICAM*). Cilj tog programa je bio povećati produktivnost pomoću systemske primjene računalne tehnologije. ICAM program nastao je zbog potrebe za boljom analizom i komunikacijskim tehnikama kako bi se poboljšala produktivnost proizvodnje. Kao rezultat ICAM je razvio seriju metoda pod nazivom IDEF (engl. *ICAM Definition*) (Pavlic, Informacijski sustavi, 2009) (Wikipedia, 2018).

### 2.2 Metode IDEF metodologije

U nastavku su navedene sve metode u sklopu IDEF metodologije, koje su kasnije pobliže objašnjene. Poseban naglasak stavljen je na metodu IDEF1X. Tablica 1. prikazuje popis metoda IDEF metodologije.

Tablica 1 - Popis metoda IDEF metodologije (Pavlic, Oblikovanje baza podataka, 2011)

IDEF 0	Function Modeling	Modeliranje funkcija
IDEF 1	Information modeling	Modeliranje informacija
IDEF 1X	Data modeling	Modeliranje podataka
IDEF 2	Simulation modeling	Modeliranje simulacija
IDEF 3	Process Description Capture	Proces prikupljanja opisa
IDEF 4	Object-oriented design	Objektno orijentirano oblikovanje
IDEF 5	Ontology Description Capture	Ontologija prikupljanja opisa
IDEF 6	Design Rationale Capture	Oblikovanje logičke podloge
IDEF 7	Information System Audit Method	Metoda ocjene informacijskoga sustava

IDEF 8	User Interface Modeling	Modeliranje korisničkog sučelja
IDEF 9	Scenario-driven Info Sys Desing Spec	Specifikacija informacijskog sustava vođena scenarijima
IDEF 10	Implementation Architecture Modeling	Modeliranje arhitekturne implementacije
IDEF 11	Information Artifact modeling	Modeliranje Izrade informacija
IDEF 12	Organization modeling	Modeliranje organizacije
IDEF 13	Three Schema Mapping Design	Oblikovanje sheme triju pridruživanja
IDEF 14	Network Design	Oblikovanje mreže

### 2.2.1 IDEF0

**IDEF0** je metoda za modeliranje odluka, aktivnosti i akcija u nekoj organizaciji ili sustavu. IDEF0 je izveden iz grafičkog jezika Tehnika za Strukturiranu analizu i Dizajn (engl. *Structured Analysis and Design Technique, SADT*). Ova metoda razvijena je za funkcijsko modeliranje kako bi pospješila analizu i komunikaciju iz funkcionalne perspektive sustava.

IDEF0 modeli trebali bi pomoći pri organizaciji, kao i za određivanje širine područja analize sustava, te funkcijske analize. Ova metoda pospješuje komunikaciju analitičara i korisnika, kao i zajedničko donošenje odluka kroz jednostavne i razumljive grafičke simbole. Kao sredstvo za analizu, IDEF0 pomaže u određivanju funkcija u sustavu. IDEF0 modeli najčešće nastaju pri početku izgradnje sustava.

1993. godine Laboratorij za računalne sustave američkog nacionalnog instituta za standarde i tehnologiju (engl. *National Institute of Standards and Technology, NIST*) je proglasio IDEF0 jednim od službenih standarda za funkcijsko modeliranje (Knowledge Based Systems, Idef.com, 2018).

### 2.2.2 IDEF1

**IDEF1** je metoda za modeliranje informacija. IDEF1 je razvijena kao metoda za analizu sustava i komunikaciju.

Ova metoda se najčešće koristi za:

1. Određivanje koje informacije se koriste u poslovanju neke firme
2. koji problemi nastaju pri pogrešnom upravljanju nekim informacijama, kao i
3. koje informacije bi bile važne za buduću primjenu u sustavu

IDEF1 obuhvaća informacije vezane za objekte neke organizacije. Pogled ove metode na informacijski sustav obuhvaća ne samo automatizirane (programska i računalna oprema) već i ostale dijelove sustava kao što su: ljudi, dokumenti, itd. Ova metoda je razvijena kako bi pomogla organizacijama u analizi, te jasnom predstavljanju potreba i zahtjeva za informacijskim resursima (Knowledge Based Systems, Idef.com, 2018).

### 2.2.3 IDEF2

**IDEF2** je započela kao metoda za modeliranje korisničkih sučelja. Međutim zbog potrebe za alatom za simulacije, IDEF2 je prenamijenjena u alat za upravljanje resursima promjenjivih u vremenu u procesu proizvodnje omogućavajući razvojno okruženje za specifikaciju simulacija baziranih na matematičkim modelima (Wikipedia, 2018)

### 2.2.4 IDEF3

**IDEF3** je metoda prikupljanja opisa. Ova metoda koristi se za dokumentiranje i opisivanje procesa. Također opisuje relaciju među situacijama i događajima na razumljiv način omogućavajući strukturiranu metodu za prikazivanje znanja kako sustav, procesi ili organizacija funkcionira.

IDEF3 opisi omogućuju:

- zabilježiti podatke koji su rezultat intervjua sa korisnicima u procesu analize sustava
- odrediti utjecaj informacijskih resursa na važne scenarije operacija unutar firme
- Dokumentiranje procedura odluke koje utječu na stanja i životni ciklus važnih podataka
- Izrada dizajna i analiza sustava
- Upravljanje konfiguracijom podataka i izmjena definicija kontrolnih pravila
- Omogućavanje generiranja modela simulacije

IDEF3 opisuje dinamički dio promatranog ili budućeg modela sustava. Znanje o procesima sustava je strukturirano u kontekstu scenarija što čini IDEF3 intuitivnom metodom za prikupljanje znanja potrebnog za opis sustava. IDEF3 obuhvaća sve temporalne informacije uključujući veze važne za procese sustava. Rezultat ove metode prikazuje strukturiranu bazu znanja važnu za izradu analitičkog i modela dizajna sustava. Za razliku od jezika za izradu simulacija (kao npr: SLAM, GPSS,...) koji se koriste za izgradnju prediktivnih matematičkih modela, IDEF3 gradi strukturirane opise sustava. Ti opisi prikazuju informacije o tome što sustav radi ili što treba raditi, te omogućuju organizaciju i prikaz različitih pogleda na sustav (Knowledge Based Systems, Idef.com, 2018).

### 2.2.5 IDEF4

Zbog intuitivnog načina objektno-orijentiranog programiranja moguće je lakše generirati programski kod. Zbog jednostavnosti proizvodnje softvera moguće su pogreške u dizajnu softvera zbog čega sistem može imati lošu iskoristivost, modularnost ili mogućnost održavanja. IDEF4 je dizajniran kako bi riješio takve probleme.

IDEF4 ne promatra objektno orijentirani pristup samo kao metodu za analizu i dizajn sustava već kao dio šireg procesa razvoja sustava. IDEF4 naglašava važnost objektno-orijentiranog procesa dizajna više nego grafičke sintakse, koristeći dijagrame kao sredstva za komunikaciju oko važnih pitanja dizajna sustava. IDEF4 je različit od drugih objektno-orijentiranih metoda zbog potpore za procjenu utjecaja dizajna vezano za nasljeđivanje klasa, kompoziciju objekata, funkcijsku dekompoziciju i polimorfizam (2018 Knowledge Based Systems, 2018).

### 2.2.6 IDEF5

Cilj ontologije općenito je „podijeliti“ svijet na njegove dijelove, tj. odrediti temeljne kategorije ili vrste koje definiraju objekte svijeta. Tako prirodne znanosti daju dobar uvid u značenje ontologije. Tako, na primjer biologija koristi klasifikaciju kako bi opisala različite vrste živih organizama na Zemlji. Postoje i drugi objekti kao što su planovi proizvodnje, firme, sveučilišta kod kojih je ontologija također važna. Kod takvih objekata ontologija se koristi radi boljeg razumijevanja, dizajna, planiranja, stvaranja i upravljanja takvim sustavima.

IDEF5 predstavlja uspješnu metodu koja se može primijeniti u teoriji i praksi, posebno dizajniranu kako bi pomogla pri stvaranju, izmjeni i održavanju ontologije sustava (Knowledge Based Systems, Idef.com, 2018).

### 2.2.7 IDEF6

**IDEF6** je metoda koja se koristi kako bi se pojednostavio prikaz i upravljanje donošenjem odluka (*engl. design rationale*) koje je važno pri razvoju organizacija. *Rationale* je razlog, opravdanje ili motivacija koja je potaknula dizajnera da koristi određenu strategiju ili dizajn pri razvoju sustava, jednostavnije *Rationale* odgovara na pitanje „Zašto neki dizajn radi na određeni način?“. (Većina metoda dizajniranja sustava se više bazira na konačan proizvod, radije nego zašto je dizajn takav kakav jest?). IDEF6 je metoda koja koristi resurse potrebne za prikazivanje prirode i strukture informacija od kojih se sastoji *design rationale* u konkretnom sustavu, radi povezivanja *design rationale* sa specifikacijom, modelima i dokumentacijom sustava. IDEF6 se najčešće koristi kod faze razvoja informacijskog sustava kroz početni koncept sustava i početne faze dizajna sustava. Slično strateškom planiranju kod MIRIS-a. Također ova metodologija se može koristiti i u fazi izrade softvera (Wikipedia, 2018).

### 2.2.8 IDEF7

Ova metoda se koristi kako bi se ocijenio informacijski sustav. (Pavlic, Informacijski sustavi, 2009)



### 2.2.9 IDEF8

**IDEF8** ili integrirana definicija za dizajn interakcije. IDEF8 metoda je za proizvodnju dizajna kako bi se pojednostavila interakcija između korisnika i sustava s kojim rade. Sustav ne mora biti samo računalni program. Sustavi za interakciju s korisnicima kod ove metode imaju tri razine specifikacije. Prva razina određuje način funkcioniranja sustava i kao rezultat dobivamo modele tekstualnih opisa procesa sustava. Druga razina dizajna određuje scenarije korištenja sustava kroz određene uloge, dok treća razina predstavlja opis sustava od strane korisnika. Na trećoj razini ova metoda sadrži niz pojmova kako bi pomogli korisnicima i dizajnerima odrediti traženi način rada sustava. IDEF8 slična je modeliranju procesa sustava kod MIRIS-a (Wikipedia, 2018).

### 2.2.10 IDEF9

**IDEF9** ili integrirana definicija za određivanje pravila (ograničenja) sustava, je razvijena kako bi pomogla pri određivanju i analizi pravila nekog poslovnog sustava. Jedan od glavnih razloga nastanka ove metode je postojanje mnogih pravila koja tvore neki sustav (poslovnu organizaciju) i koja su najčešće slabo definirana. Znanje o tome koja su ta pravila, te njihova povezanost često nisu dobro poznati. Kako bi mogli učinkovito upravljati procesima sustava važno je znati ta pravila (Wikipedia, 2018).

### 2.2.11 IDEF10

**IDEF10** je metoda za modeliranje arhitekturne implementacije (Pavlic, Informacijski sustavi, 2009).

### 2.2.12 IDEF11

**IDEF11** koristi se za modeliranje izrade informacija (Pavlic, Informacijski sustavi, 2009).

### 2.2.13 IDEF12

Ova metoda koristi se za modeliranje organizacije (Pavlic, Informacijski sustavi, 2009).

### 2.2.14 IDEF13

Ova metoda koristi se za oblikovanje sheme triju pridruživanja (Pavlic, Informacijski sustavi, 2009).

### 2.2.15 IDEF14

**IDEF14** ili integrirana definicija za razvoj mreža, je metoda koja se bavi modeliranjem i dizajnom računalnih i komunikacijskih mreža. Ova metoda se može se koristiti za razvoj već

postojećih, kao i budućih mreža. IDEF14 pomaže korisniku upoznati moguće dizajne mreže. Glavni cilj ove metode je brz i učinkovit dizajn mreža (Wikipedia, 2018).

## 2.2.16 Metoda IDEF1X

**Integration DEFinition for information modelling** (IDEF1X) je metoda za modeliranje podataka koja se koristi za izgradnju modela informacija koji predstavljaju strukturu i semantiku informacija koji opisuju neki sustav uz pomoć grafičkih simbola.

IDEF1X omogućuje izgradnju semantičkih modela podataka koji se mogu koristiti za potporu upravljanja resursima podataka, integraciju informacijskih sustava i gradnju baza podataka. IDEF1X je dio IDEF serije metoda jezika za modeliranje na području softverskog inženjerstva (Wikipedia, 2018).

### 2.2.16.1 Koncepti metode IDEF1X

Komponente IDEF1X pogleda su (Technology, 1993):

#### 1.) ENTITETI

- Entiteti zavisni o identifikatoru
- Entiteti nezavisni o identifikatoru

#### 2.) VEZE

- Veza spoja zavisna o identifikatoru
- Veza spoja nezavisna o identifikatoru
- Veza kategorizacije
- Ne-specifična veza

#### 3.) ATRIBUTI

#### 4.) KLJUČEVI

- Primarni ključ
- Dodatni ključ
- Vanjski (strani) ključ

#### 2.2.16.1.1 Entitet

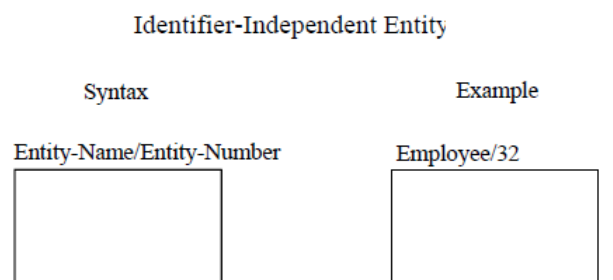
**Entitet** (*engl. Entity*) može predstavljati neke realne i apstraktne objekte kao što su ljudi, objekti, mjesta, događaji, ideje, itd. koji dijele neke zajedničke atribute ili karakteristike.

Jedno konkretno pojavljivanje entiteta nazivamo instanca entiteta. Jedan objekt stvarnog svijeta ili predmet razmišljanja može biti instanca više entiteta u jednom pogledu. Na primjer: Osoba Ivan Ivić može biti instanca entiteta ZAPOSLENIK kao i instanca entiteta KUPAC. Također, instanca entiteta može predstavljati i kombinaciju objekata stvarnog svijeta, tako npr. Ivan i Ana mogu biti instanca entiteta PAR.

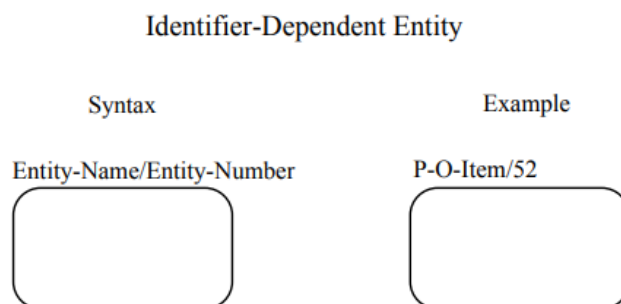
Entitet je „**nezavisan o identifikatoru**“ (*engl. identifier independent*) ili jednostavno „nezavisan“ ako svaku instancu tog entiteta možemo jedinstveno odrediti bez potrebe za određivanjem veza sa drugim entitetima. Entitet je „**zavisan o identifikatoru**“ (*engl. identifier independent*) ili „zavisan“ ako jedinstvena identifikacija instance tog entiteta ovisi o povezanosti s drugim entitetima (Technology, 1993).

Entitet se u metodi IDEF1X prikazuje pomoću pravokutnika u kojeg upisujemo imena atributa, gdje primarne ključeve pišemo iznad horizontalne linije. Ako je entitet zavisan o identifikatoru onda su rubovi pravokutnika zaobljeni. Naziv entiteta pišemo iznad pravokutnika. Svaki entitet treba imati jedinstveni naziv u dijagramu. Neki pozitivan cijeli broj može također biti dio imena entiteta, taj broj se odvaja kosom crtom („/“). Naziv entiteta treba biti imenica u jednini koja opisuje što entitet predstavlja. Naziv entiteta može biti kratica ili akronim. Definicija entiteta kao i popis sinonima ili aliasa definirani su u rječniku. Iako se isti entitet može pojaviti u više dijagrama, u istom dijagramu se može pojaviti samo jednom. Jedan entitet može imati više relacija sa drugim entitetima. Ako cijeli vanjski ključ postaje primarni ključ nekog entiteta tada je taj entitet zavisan o identifikatoru. Međutim ako primarni ključ entiteta ne sadrži ili sadrži samo dio vanjskog ključa onda je entitet neovisan o identifikatoru (Technology, 1993).

Na slici 18. prikazani su sintaksa i primjer entiteta nezavisnog o identifikatoru, dok je na slici 19. prikazana sintaksa i primjer entiteta zavisnog od identifikatora.



Slika 18 - Entitet nezavisan o identifikatoru (Technology, 1993)



Slika 19 – Entitet zavisan o identifikatoru (Technology, 1993)

#### 2.2.16.1.2 Domena

**Domena** (*engl. Domain*) predstavlja skup vrijednosti od kojih atributi poprimaju svoju vrijednost. Domena je klasa koja može imati točan ili beskonačan skup instanci.

Jedna domena se sastoji od podataka istog tipa npr: cijeli brojevi, datumi, itd. Na primjer, *Poštanski broj* može biti domena gdje skup dozvoljenih vrijednosti odgovara definiciji *Poštanskog broja* (tj. Poštanski broj jedinstveno određuje općinu) koji se označava sa peteroznamenastim brojem. Također domena može biti i *Prezime* koji ima velik broj mogućih vrijednosti i koje se tvori od slova abecede [A-Z, a-z]. Domene smatramo nepromjenjivim klasama čije se vrijednosti ne mijenjaju tijekom vremena, za razliku od klase entiteta, čije instance se mogu mijenjati s obzirom na promjenu podataka u bazi. Na primjer, domena *Datum* će sadržavati sve datume, međutim sve instance te domene vjerojatno nećemo trebati upotrijebiti kao vrijednosti atributa entiteta koji sadrže neki datum. Svaka domena se sastoji od jedinstvenih vrijednosti (Technology, 1993).

Postoji dvije vrste domena:

- Bazična domena (*engl. Base domain*)
- Upisna domena (*engl. Typed domain*)

##### 2.2.16.1.2.1 Bazična domena

Standardni tip podataka bazične domene može biti: Znakovni tip, Numerički ili Boolean, ali također se mogu koristiti tipovi podataka kao: datum, vrijeme, binarni, itd. Bazične domene također mogu imati definirana pravila. Ta pravila se koriste kako bi se odredio skup dozvoljenih vrijednosti za neku domenu. Dva najčešće korištena pravila su lista dozvoljenih vrijednosti i raspon vrijednosti.

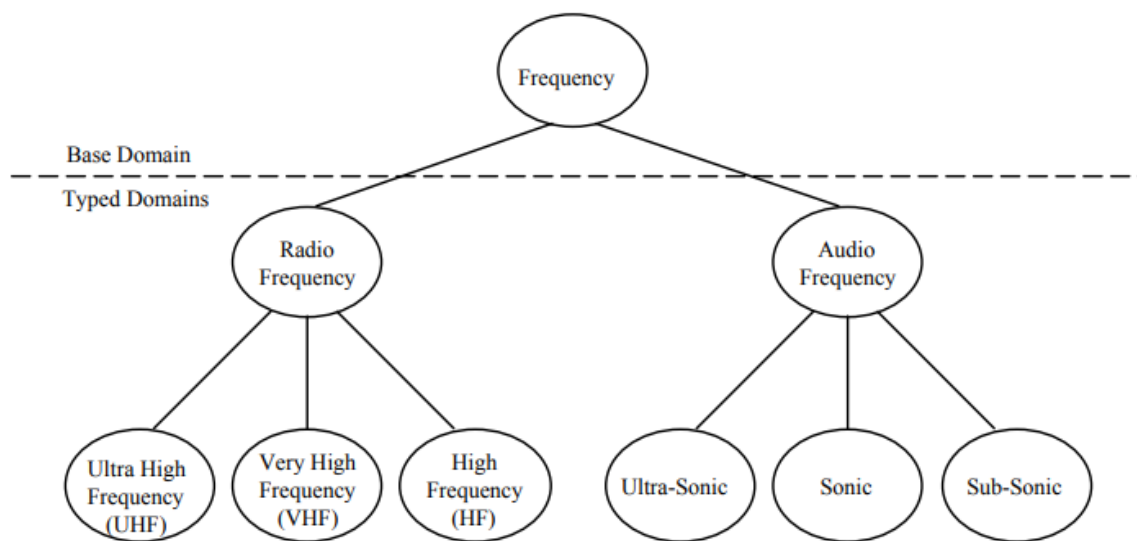
**Lista vrijednosti** definira skup svih dozvoljenih vrijednosti instanci domene. Ovo pravilo se najčešće koristi kako bi se stvorila lista konkretnih vrijednosti iz neke domene, npr. šifra države SAD-a može biti samo jedna od kratica koje pripadaju listi kratica svake savezne države, npr: AZ (Arizona), AK (Alaska), AL (Alabama), itd., ne bilo koja kombinacija dva slova.

**Raspon vrijednosti** definira vrijednosti instanci domene koji su definirani nekom gornjom i donjom granicom. Na primjer Azimut koji može imati vrijednosti od  $-360^\circ$  do  $360^\circ$ . Domena ne treba imati definirano pravilo, u tom slučaju domena je ograničena samo tipom podataka. Npr. vrijednosti domene prezime su znakovne vrijednosti. Bazična domena je određena ako ima definiran tip vrijednosti i ima pravilo domene (Technology, 1993).

##### 2.2.16.1.2.2 Upisna domena

Upisne domene mogu biti pod-tipovi bazičnih domena ili drugih upisnih domena, koje mogu dodatno ograničiti vrijednosti neke domene. Upisna domena treba imati tip vrijednosti i za tu

domenu vrijede pravila nad-domene (*engl. supertype domain*). Na ovaj način možemo definirati hijerarhiju domena. Svaka razina hijerarhije ima definirana pravila. Hijerarhija domena je **generalizacijska**. Za razliku od kategorija entiteta, podtipovi hijerarhije domena se međusobno ne isključuju. Na slici je prikazana domena *Frekvencija* koja nema pravila domene, ali upisna domena *Audio Frequency* je definirana u Hercima (Hz) koja može iznositi između 1 i 250 000 Hz. Pod-domena *Sonic frequency* ima dodatno ograničenje na raspon frekvencije ljudskog sluha između 20 i 20 000 herca. Važno je reći da svaka instanca domene uz vlastita ima pravila kao i njena nad-domena. Informacije o domenama se trebaju nalaziti u rječniku gdje bi trebali biti navedeni tipovi podataka baznih domena, kao i pravila domena. Svaka domena treba imati jedinstveno ime (Technology, 1993). Na slici 20. je prikazana hijerarhija domena Frekvencija.



Slika 20 – Primjer bazične i upisne domene (Technology, 1993)

### 2.2.16.1.3 Pogled

U IDEF1X metodi **pogled** je skup povezanih entiteta i njihovih atributa koji imaju neku svrhu. Pogled može obuhvaćati cijeli sustav, ili samo dio koji modeliramo. IDEF1X model se može sastojati od jednog ili više pogleda prikazanih u dijagramu pogleda kao i definicije entiteta i domena (atributa) koji su dio tog pogleda. Entiteti i domene su definirani u zajedničkom rječniku i međusobno povezani u pogledu. Entitet ZAPOSLENIK se može nalaziti u različitim pogledima, različitim modelima i imati različite attribute u svakom. Važno je da u svakom pogledu entitet ZAPOSLENIK ima isto značenje. Entitet ZAPOSLENIK bi trebao predstavljati klasu svih zaposlenika, tj. različite objekte možemo klasificirati u jednu klasu na temelju zajedničkih obilježja. Pogled ima svoj naziv i dodatne opisne informacije kao naziv autora, datum izrade pogleda, datum zadnje izmjene, itd. Pogled može imati i tekstualni opis koji može sadržavati opise veza u pogledu, kratak opis entiteta i atributa, te

popis pravila koja su definirana u dijagramu. Svaki pogled treba imati jedinstveni naziv (Technology, 1993).

#### 2.2.16.1.4 Atribut

Domena koja je pridružena entitetu u nekom pogledu nazivamo **Atributom** (*engl. Attribute*). U IDEF1X pogledu atributi predstavljaju karakteristiku ili svojstvo nekog entiteta.

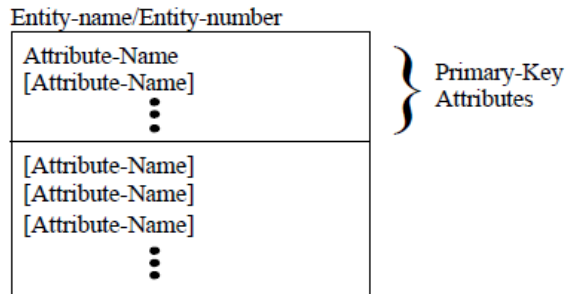
Instanca atributa je karakteristika neke instance entiteta. Instanca atributa koja je definirana tipom karakteristike kao i svojom vrijednošću nazivamo „vrijednost atributa“. Instanca entiteta tako ima posebnu vrijednost za svaki atribut. Na primjer, IME-ZAPOSLENIKA i DATUM-ROĐENJA su atributi entiteta ZAPOSLENK. Instanca entiteta ZAPOSLENIK može poprimiti vrijednosti „Jenny Lynne“ i „27.2.1980“ za attribute IME-ZAPOSLENIKA i DATUM-ROĐENJA. Također, ponekad vrijednost atributa nije poznata, kao npr. Datum rođenja. Ponekad, neki atribut ne možemo primijeniti na sve instance entiteta, na primjer: entitet PROZOR može imati atribut BOJA. Vrijednost BOJE jedne instance entiteta može biti „plava“ dok druga instanca ne mora imati vrijednost (vrijednost je NULL) jer prozor može biti proziran. Dakle, atribut BOJA ne može se primijeniti na sve instance entiteta PROZOR.

Svaki Entitet treba imati atribut ili kombinaciju atributa čije vrijednosti jedinstveno određuju svaku instancu tog entiteta. Takvi atributi tvore „primarni-ključ“ entiteta. Na primjer, ŠIFRA-ZAPOSLENIKA može biti primarni ključ entiteta ZAPOSLENIK, dok su atributi IME-ZAPOSLENIKA i DATUM-ROĐENJA ne-ključni atributi. Entitetu mogu pripadati i atributi koji pripadaju drugima entitetima naslijeđeni kroz vezu spoja ili vezu kategorizacije. Na primjer, ako svakog zaposlenika možemo rasporediti u neki odjel, atribut ŠIFRA-ODJELA može postati atribut entiteta ZAPOLSENIK koji je preko veze spoja prešao iz entiteta ODJEL u entitet ZAPOSLENIK. Samo primarni ključevi mogu prijeći iz jednog entiteta u drugi. Npr. atribut NAZIV-ODJELA neće prijeći u entitet ZAPOSLENIK osim ako nije dio primarnog ključa entiteta ODJEL. Svaki atribut je određen jedinstvenom domenom kojoj pripada.

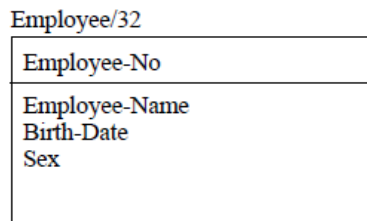
Naziv atributa može biti imenica jednine koja opisuje karakteristiku tog atributa. Kratice i akronimi u imenu atributa su dozvoljeni, međutim atribut treba imati jedinstveno ime u cijelom IDEF1X modelu. Definicija atributa kao i lista sinonima ili aliasa treba biti navedena u rječniku. Atributi su prikazani na modelu unutar pravokutnika entiteta. Atributi koji čine primarni ključ su prikazani na vrhu liste atributa i odvojeni su horizontalnom linijom od ne-ključnih atributa. Jednom entitetu može pripadati proizvoljno mnogo atributa. Entitet može imati više naslijeđenih (*engl. migrated*) atributa, međutim naslijeđeni atributi moraju biti dio primarnog ključa ako se radi o roditeljskom ili kategorijskom entitetu. Svaki ključni atribut mora imati vrijednost, ne smije biti NULL. Svaka instanca entiteta ne smije imati više od jedne vrijednosti za svaki atribut. Atributi koji nisu dio primarnog ključa mogu biti NULL (što znači da im vrijednost nije poznata ili se ne može primijeniti), do imena takvih atributa dodajemo znak „O“ (kao neobavezan atribut, *engl. Optional*). Jedan atribut može imati više aliasa u različitim pogledima istog modela (Technology, 1993).

Na slici 21. je prikazana sintaksa atributa i primarnog ključa, kao i primjer atributa.

#### Attribute And Primary Key Syntax



#### Example



Slika 21 - Sintaksa atributa i primarnih ključeva (Technology, 1993)

#### 2.2.16.1.5 Veza spoja

Kod IDEF1X modela **veza spoja** predstavlja asocijaciju, povezanost među entitetima (Technology, 1993).

##### 2.2.16.1.5.1 Specifična veza spoja

**Specifična veza spoja** ili jednostavno veza spoja je asocijacija među entitetima u kojoj svaka instanca „roditeljskog entiteta“ može biti povezana sa nula, jednom ili više instanci „entiteta djeteta“. Dok svaka instanca entiteta djeteta može biti povezana sa nula ili jednom instancom roditeljskog entiteta. Na primjer, između entiteta KUPAC i NARUDŽBA može postojati veza spoja ako kupac može imati nula, jednu ili više narudžbi, a svaka narudžba može pripadati samo jednom kupcu. Instanca veze povezuje instance entiteta. Na primjer, kupcu Ivanu Iviću pripada narudžba broj 123. Kod veze spoja možemo definirati i kardinalnost, brojnost veze. Kardinalnost definira koliko instanci entiteta djeteta može biti povezano sa jednom instancom entiteta roditelja (Technology, 1993).

Kod IDEF1X-a postoje sljedeće kardinalnosti veze:

- 1.) Jedna instanca roditeljskog entiteta može biti povezana sa nula ili više instanci entiteta djeteta

- 2.) Svaka instanca roditeljskog entiteta treba biti povezana sa najmanje jednom instancom entiteta djeteta
- 3.) Jedna instanca roditeljskog entiteta može biti povezana sa nula ili jednom instancom entiteta djeteta
- 4.) Svaka instanca roditeljskog entiteta može biti povezana sa nekim točnim brojem instanci entiteta djeteta
- 5.) Svaka instanca roditeljskog entiteta može biti povezana sa nekim točnim rasponom instanci entiteta djeteta

Kardinalnost veze možemo također gledati iz perspektive entiteta djeteta (Technology, 1993).

#### 2.2.16.1.5.2 Identifikacijska veza

**Identifikacijska veza** je veza kod koje instancu nekog entiteta možemo jedinstveno identificirati samo pomoću veze sa roditeljskim entitetom, te svaka instanca entiteta djeteta može biti povezana sa točno jednom instancom roditeljskog entiteta. Na primjer, ako neki projekt može imati više zadataka, a zadatke možemo jedinstveno identificirati samo unutar projekta, onda je veza između entiteta PROJEKT i ZADATAK identifikacijska. Entitet dijete u identifikacijskoj vezi uvijek ovisi o postojanju roditeljskog entiteta.

Identifikacijsku vezu označavamo sa punom linijom koja povezuje roditeljski i entitet dijete, sa točkom na strani entiteta djeteta. Ako je veza identifikacijska, onda je entitet dijete zavisen o identifikatoru entiteta roditelja (Technology, 1993).

#### 2.2.16.1.5.3 Ne-identifikacijska veza (engl. Non-Identifying relationship)

Ako svaku instancu entiteta djeteta možemo jedinstveno odrediti bez povezanosti s roditeljskim entitetom onda je veza ne-identifikacijska. Na primjer, iako postojanje entiteta NARUDŽBA ovisi o entitetu KUPAC, narudžbu možemo jedinstveno odrediti po broju narudžbe, bez potrebe da znamo kojem kupcu pripada.

Entitet dijete je nezavisan od entiteta roditelja. Isprekidana linija predstavlja ne-identifikacijsku vezu između roditeljskog i entiteta djeteta. Ako veza nije identifikacijska oba entiteta će biti nezavisni (entitet dijete je nezavisan od roditelja) osim ako bilo koji od tih entiteta već ne sudjeluje u identifikacijskoj vezi (Technology, 1993).

#### 2.2.16.1.5.4 Obavezna ne-identifikacijska veza

U obaveznoj (*engl. mandatory*) vezi spoja svaka instanca entiteta djeteta je povezana sa točno jednom instancom roditeljskog entiteta (Technology, 1993).

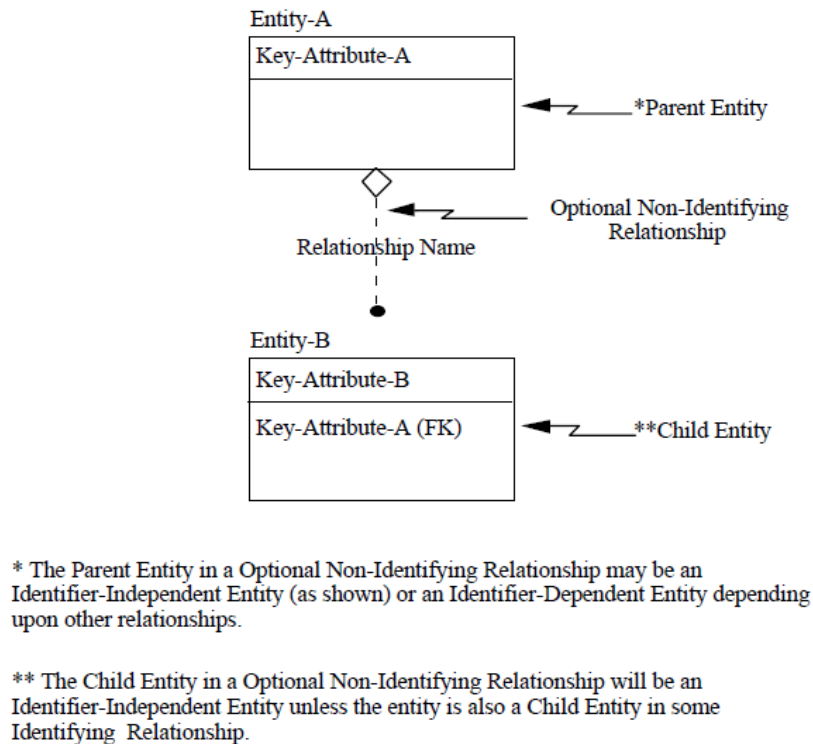
#### 2.2.16.1.5.5 Opcionalna ne-identifikacijska veza

Opcionalnu (*engl. optional*) vezu označavamo isprekidanom linijom sa romбом na strani roditeljskog entiteta i točkom na strani entiteta djeteta. U opcionalnoj vezi svaka instanca



entiteta djeteta je povezana sa nula ili jednom instancom roditeljskog entiteta. Opcionalna ne-identifikacijska veza predstavlja uvjetnu zavisnost. Instanca entiteta djeteta u kojoj svaki atribut vanjskog ključa ima vrijednost treba biti povezana sa instancom roditeljskog entiteta u kojoj su atributi primarnog ključa jednaki vrijednostima atributa vanjskog ključa entiteta djeteta (Technology, 1993).

Slika 22. prikazuje opcionalnu ne-identifikacijsku vezu.



Slika 22 - Opcionalna ne - identifikacijska veza (Technology, 1993)

#### 2.2.16.1.5.6 Specifična veza spoja

**Specifična veza spoja** je na dijagramu prikazana pomoću linije koja spaja roditeljski sa entitetom djetetom, sa točkom n stani entiteta djeteta. Standardna kardinalnost ove veze je nula, jedan ili više.

Brojnost veze označavamo pored točke, a može biti:

- P – kardinalnost je jedan ili više
- Z – kardinalnost je nula ili jedan
- n – ako je kardinalnost točan broj, n je neki broj
- n-m – rang brojnosti od n do m

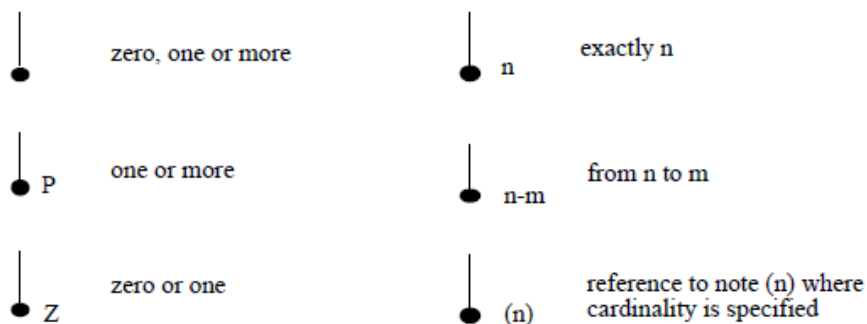
Ime veze je (najčešće) glagol označen pored same veze. Veza između dva ista entiteta treba biti jedinstvena, ali imena veza se mogu ponavljati u istom modelu. Naziv veze se označava u

smjeru od roditeljskog prema entitetu djetetu tako da može tvoriti rečenicu koja se sastoji od imena roditeljskog entiteta, naziva veze, kardinalnosti veze, te imena entiteta djeteta. Na primjer, veza „pripada“ između entiteta roditelja PROJEKT i entiteta djeteta ZADATAK koja ima kardinalnost „P“ možemo opisati kao „Projektu pripada jedan ili više ZADATAKA“. Ako se veza tvori od perspektive entiteta roditelja i entiteta djeteta onda se u naziv veze piše prvo perspektiva entiteta roditelja pa entiteta djeteta koje su odvojene kosom crtom („/“).

Također je važno da naziv veze vrijedi i od smjera entiteta djeteta prema roditeljskom entitetu, ako nije točno zadano da je ime veze definirano od entiteta djeteta prema roditeljskom entitetu. Dakle, veza iz perspektive entiteta djeteta prema roditeljskom entitetu bi bila: „ZADATAK pripada točno jednom PROJEKTU“. Naziv veze iz obje perspektive bi bio: „projektu pripada jedan ili više zadataka/zadatak pripada točno jednom projektu“

Specifična veza spaja uvijek roditeljski entitet sa entitetom djetetom. U identifikacijskoj, kao i u obaveznoj ne-identifikacijskoj vezi svaka instanca entiteta djeteta je povezana sa točno jednom instancom roditeljskog entiteta. Samo ne-identifikacijske veze mogu biti rekurzivne, dakle mogu povezivati jednu instancu entiteta sa drugom instancom istog entiteta (Technology, 1993).

Slika 23. prikazuje vrste kardinalnosti veze.



Slika 23 - Sintaksa kardinalnosti veze (Technology, 1993)

#### 2.2.16.1.6 Veza kategorizacije

**Veza kategorizacije** spaja neki entitet sa kategorijom drugog entiteta.

Veza kategorizacije predstavlja strukturu u kojoj je neki entitet kategorija drugog entiteta. Ovu vrstu veze koristimo kada imamo entitete koji su kategorija drugih entiteta. Na primjer, kada imamo entitet ZAPOSLENIK i potrebno nam je više informacija o vrstama zaposlenika kao što su MJESEČNO-PLAĆENI-ZAPOSLENIK i ZAPOSLENIK-PLAĆEN-PO-SATU. Tako su entiteti MJESEČNO-PLAĆENI-ZAPOSLENIK i ZAPOSLENIK-PLAĆEN-PO-SATU kategorije entiteta ZAPOSLENIK, koje su u dijagramu povezani pomoću veze kategorizacije. Ovu vezu možemo koristiti kod veze koja vrijedi za neku određenu kategoriju, ili kako bi razlikovali veze među različitim kategorijama entiteta. Veza kategorizacije je veza

između „generičkog entiteta“ i „kategorijskog entiteta“. Klaster je skup jedne ili više veza kategorizacije. Instanca generičkog entiteta može biti povezana sa instancom samo jednog kategorijskog entiteta u klasteru, te instanca svakog kategorijskog entiteta može biti povezana sa samo jednom instancom generičkog entiteta.

Svaka instanca kategorijskog entiteta predstavlja isti objekt stvarnog svijeta kao i povezana instanca generičkog entiteta. Na primjer, entitet ZAPOSLENIK je generički entitet, a MJESEČNO-PLAĆENI-ZAPOSLENIK i ZAPOSLENIK-PLAĆEN-PO-SATU kategorijski entiteti povezani vezom kategorizacije sa generičkim entitetom.

S obzirom kako instanca generičkog entiteta ne može biti povezana sa instancama više od jednog kategorijskog entiteta u klasteru, kategorijski entiteti su međusobno isključuju. Tj. instanca generičkog entiteta može biti povezana samo sa instancama jednog kategorijskog entiteta. Što znači da ZAPOSLENIK ne može biti i MJESEČNO-PLAĆENI-ZAPOSLENIK i ZAPOSLENIK-PLAĆEN-PO-SATU. Međutim jedan entitet može biti generički u više klastera, dakle kategorijski entiteti u jednom klasteru se međusobno ne isključuju sa kategorijskim entitetima u drugom klasteru. Na primjer, možemo imati i drugi klaster koji ima ŽENSKIE i MUŠKE zaposlenike, što znači da jedna instanca entiteta ZAPOSLENIK može biti povezana sa jednom instancom ili MJESEČNO-PLAĆENI-ZAPOSLENIK ili ZAPOSLENIK-PLAĆEN-PO-SATU, i istovremeno sa jednom od instanci ŽENSKI-ZAPOSLENIK ili MUŠKI-ZAPOSLENIK. U „potpunom kategorijskom klasteru“ svaka instanca generičkog entiteta treba biti povezana sa instancom kategorijskog entiteta. Npr. svaki zaposlenik treba biti ili mjesečno plaćen ili plaćen po satu – sve kategorije su uključene. Dok kod „nepotpunog kategorijskog klastera“ instanca generičkog entiteta ne mora biti povezana sa instancom kategorijskog entiteta.

Atribut kod generičkog entiteta ili kod nekog od njegovih pod kategorija možemo koristiti kao tzv. „Discriminator“ za specifični klaster tog entiteta. Vrijednost diskriminatora određuje kategorije instanci generičkog atributa. Na primjer, kod klastera koji uključuje mjesečno i po satu plaćene zaposlenike diskriminator može biti TIP-ZAPOLSENIKA.

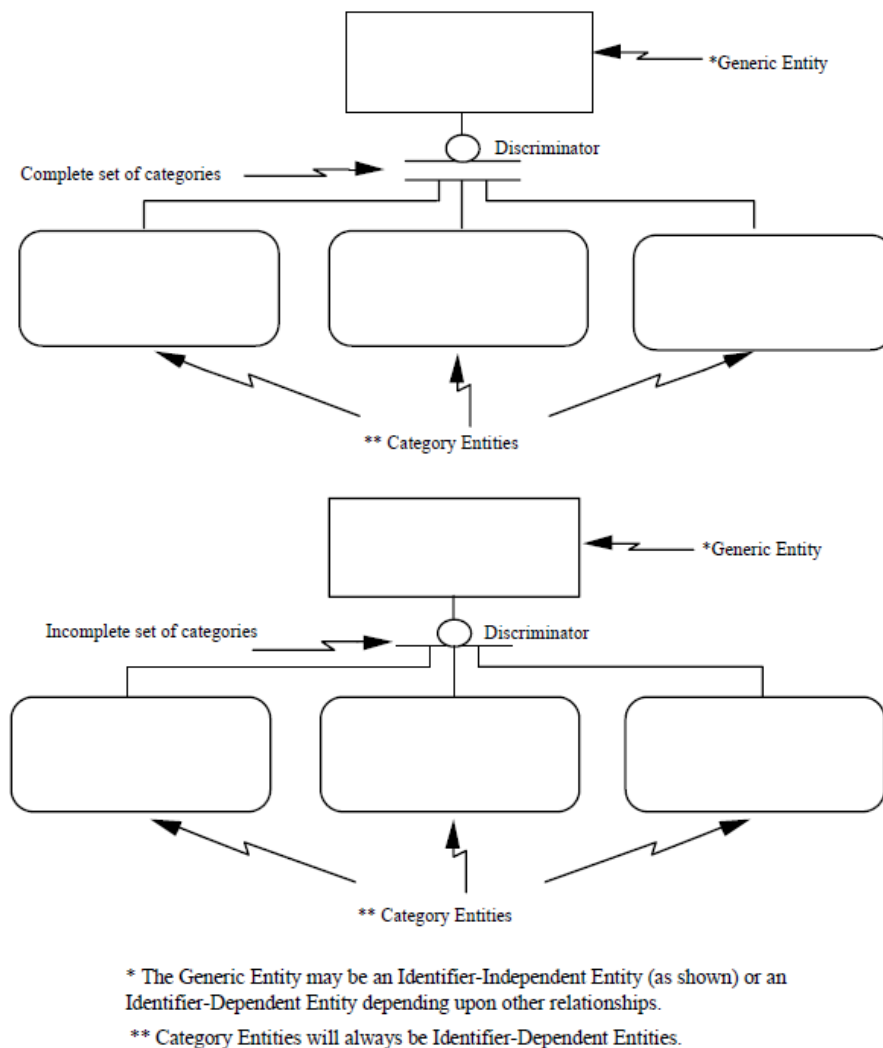
Klaster je prikazan kao linija koja ide od generičkog entiteta prema podcrtanoj kružnici. Više linija ide od podcrtane kružnice do svakog pojedinačnog kategorijskog entiteta. Svaka linija od generičkog entiteta do kružnice pa do kategorijskog entiteta predstavlja jednu vezu kategorije. Kardinalnost veze nije posebno naznačena jer uvijek iznosi nula ili jedan. Kategorijski entiteti su uvijek zavisni o identifikatoru. Generički entitet je nezavisan o identifikatoru osim ako ne sudjeluje u vezi gdje je zavisan o identifikatoru.

Ako je kružnica dvostruko podcrtana onda je skup kategorijskih entiteta kompletan. Jedna crta ispod kružnice označava nepotpun skup kategorija. Naziv diskriminatora (ako ga ima) pišemo uz kružnicu. Iako veze kategorizacije nisu posebno imenovane, vezu od generičkog prema kategorijskom entitetu možemo čitati možemo čitati kao „može biti“. Npr. ZAPOSLENIK može biti MJESEČNO-PLAĆENI-ZAPOSENIK, dok ako se radi o potpunom skupu kategorija vezu možemo čitati kao „mora biti (Technology, 1993).

Pravila veze kategorizacije (Technology, 1993):

- Generički kao i svaki kategorijski entitet trebaju imati isti primarni ključ.
- Kategorijski entitet treba pripadati samo jednom generičkom entitetu.
- Kategorijski entitet u jednoj kategorijskoj vezi može biti generički entitet u drugoj.
- Jedan generički entitet može imati više klastera.
- Sve instance jednog kategorijskog entiteta moraju imati istu vrijednost diskriminatora, koje trebaju biti različite od vrijednosti diskriminatora drugog kategorijskog entiteta.
- Niti jedan generički entitet ne može imati klastera koje imaju isti naziv.
- Diskriminator kompletnog kategorijskog klastera ne smije biti opcionalni atribut.

Slika 24. prikazuje primjer hijerarhije kategorija entiteta.



Slika 24 - Veza kategorizacije (Technology, 1993)

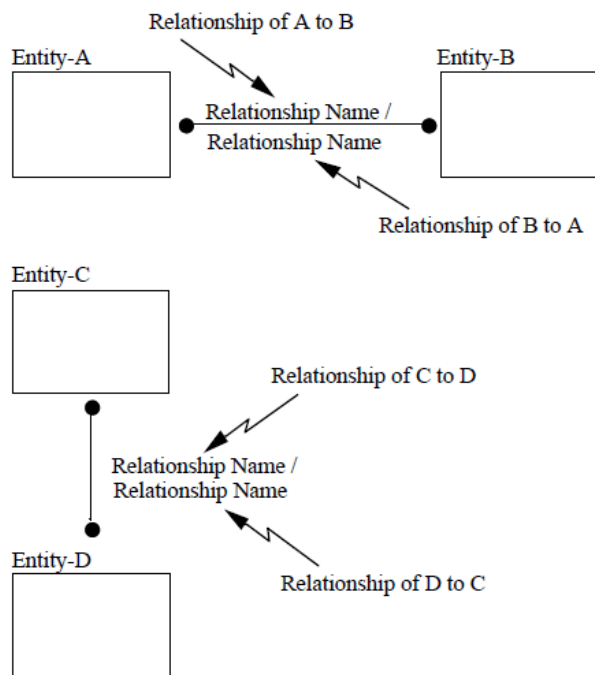
#### 2.2.16.1.7 Ne-specifična veza

Ne-specifične veze se koriste kako bi označile veze u kojima više instanci jednog entiteta može biti povezano sa više instanci drugog entiteta.

Veze roditelj-dijete i kategorizacijska veza su specifične veze jer obje definiraju na koji način su povezane instance jednog entiteta sa instancama drugog entiteta. Pri početku definiranja modela korisno je odrediti ne-specifične veze među entitetima. Ne-specifične veze su kasnije doradene u kasnijim fazama izrade modela. Ne-specifičnu vezu možemo nazvati još i „many-to-many relationship“. Ova veza označava povezanost između entiteta u kojoj instanca jednog entiteta može biti povezana sa nula, jednom ili više instanci drugog entiteta, te svaka instanca drugog entiteta može biti povezana sa nula, jednom ili više instanci drugog entiteta. Npr. ako jedan zaposlenik može raditi na više projekata i na projektu može raditi više zaposlenika, onda je veza među entitetima ZAPOSLENIK i PROJEKT ne-specifična. Ne-specifičnu vezu u kasnijim fazama razvoja modela možemo zamijeniti uvođenjem novog entiteta, kao npr. PRIPADNOST-PROJEKTU koja predstavlja specifičnu vezu spoja među entitetima PROJEKT i ZAPOSLENIK. Novo definirane veze označavaju da zaposlenik ima nula, jednu ili više PRIPADNOSTI-PROJEKTU, te da je svaka PRIPADNOST-PROJEKTU za samo jednog zaposlenika i jedan projekt. Entiteti koje uvodimo kako bi razriješili ne-specifičnu vezu nazivamo „presjekom“ ili „asocijativnim“ entitetima.

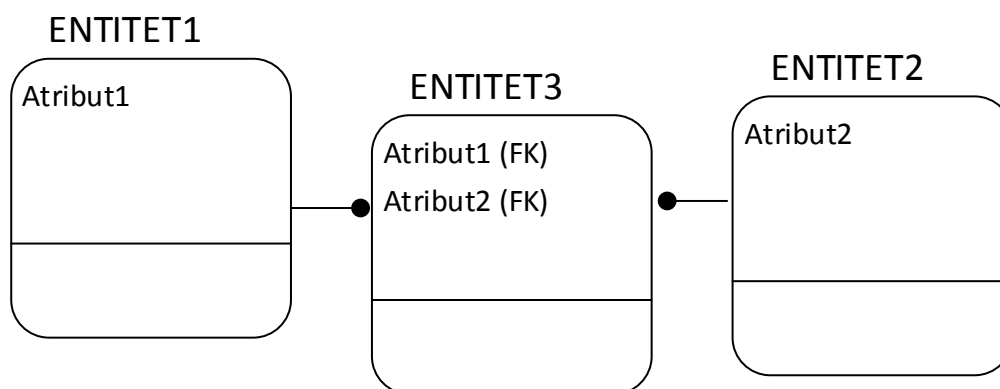
Ne-specifičnu vezu označavamo sa linijom koja ima dvije točke do svakog entiteta. Kardinalnost veze možemo označiti na svakom kraju veze. Veza može biti nazvana iz oba smjera. Naziv veze se tvori od para glagola koja su odvojena kosom crtom „/“. Naziv veze pišemo do linije koja povezuje oba entiteta. Prvi naziv označava vezu s lijeva na desno (ako su entiteti raspoređeni horizontalno) ili odozgora prema dolje (ako su entiteti raspoređeni vertikalno). Ne-specifičnu vezu od Projekta prema zaposleniku možemo čitati: „Projektu može pripadati nula, jedan ili više zaposlenika“, a od Zaposlenika prema projektu: „Zaposlenik može raditi na nula, jedan ili više projekata“. Ne-specifična je veza između dva entiteta (Technology, 1993).

Na slici 25. prikazana je ne-specifična veza između dva entiteta.



Slika 25 - Ne-specifična veza (Technology, 1993)

Pri razrješenju ne-specifične veze najčešće uvodimo novi entitet (asocijativni entitet) koji spaja dva entiteta povezana ne-specifičnom vezom. Novi entitet je zavisian od ta dva i njegov primarni ključ se tvori od primarnih ključeva dva entiteta koje povezuje. Asocijativni entitet je prikazan na slici 26.



Slika 26 - asocijativni entitet

#### 2.2.16.1.8 Primarni ključ

**Primarni ključ** (engl. *Primary key*) jedinstveno označava instancu entiteta.

Kandidat za ključ može biti atribut ili skupa tributa koji jedinstveno označava svaku instancu entiteta. Na primjer, BROJ-NARUDŽEBE jedinstveno označava instancu entiteta

NARUDŽBA. Ponekad više atributa može jedinstveno označavati instance entiteta. Npr. ŠIFRA-ZAPOSLNIKA i OIB mogu jedinstveno određivati instance entiteta ZAPOSLENIK. Ako postoji više kandidata za ključ, onda se bira jedan od njih kao primarni ključ, dok su ostali alternativni ključevi.

Atributi koji čine primarni ključ se nalaze na vrhu kvadrata koji predstavlja entitet, te su odvojeni od ostalih ne-ključnih atributa horizontalnom linijom. Svaki alternativni ključ ima svoj broj, te se označava pored atributa entiteta slovima „AK“ zajedno sa brojem ključa. Jedan atribut može biti dio više alternativnih ključeva. Atributi koji čine alternativni ili primarni ključ mogu biti naslijeđeni od drugih atributa. Primarni i alternativni ključ trebaju se sastojati od atributa koji jedinstveno identificiraju entitet (ako neki atribut koji pridonosi jedinstvenoj identifikaciji instanci entiteta nije uključen u ključ onda neke od instanci entiteta nećemo moći jedinstveno odrediti) (Technology, 1993).

Važna su dva pravila zavisnosti (Technology, 1993):

- Potpuna funkcijska zavisnost – ako se primarni ključ sastoji od više atributa, svi ne-ključni atributu moraju biti funkcijski zavisni o cijelom ključu.
- Tranzitivne zavisnosti nisu dozvoljene – govori nam da svaki ne-ključni atribut mora biti funkcijski zavisan samo o ključnim atributima.

Slika 27. pokazuje sintaksu primarnog i alternativnog ključa entiteta

#### Alternate Key Syntax

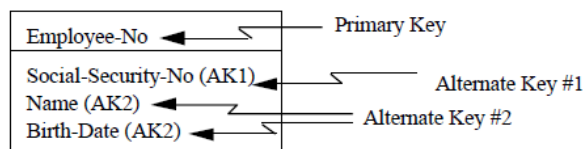
Attribute-Name (AKn) [ (AKm) . . . ]

or

Attribute-Name (AKn [ , AKm . . . ])

Where n, m, etc., uniquely identify each Alternate Key that includes the associated attribute and where an Alternate Key consists of all the attributes with the same identifier.

#### Example



Slika 27 - Primjer sintakse primarnog i alternativnog ključa (Technology, 1993)

#### 2.2.16.1.9 Vanjski ključ

Ako imamo specifičnu ili kategorizacijsku vezu između dva entiteta, onda atributi koji čine primarni ključ entiteta roditelj ili generičkog entiteta se nasljeđuju u entitet dijete ili

kategorijski entitet. Te naslijeđene attribute nazivamo **Vanjskim ključevima** (*engl. foreign keys*). Na primjer, ako su entiteti PROJEKT i ZADATAK povezani vezom spoja, primarni ključ ID-PROJEKTA entiteta roditelja PROJEKT će se naslijediti u entitet djetete ZADATAK, tj. postati će njegov vanjski ključ.

Naslijeđeni primarni ključ može postati dio ili cijeli primarni ključ, alternativni ključ ili ne-ključni atribut u naslijeđenom ili kategorijskom entitetu. Ako svi atributi primarnog ključa entiteta roditelja postaju dio primarnog ključa entiteta djeteta, onda je veza između ta dva atributa identifikacijska. Ako naslijeđeni atributi nisu dio primarnog ključa entiteta djeteta onda je veza ne-identifikacijska. Npr. ako zadatke možemo jedinstveno odrediti samo unutar projekta onda naslijeđeni atribut ID-PROJEKTA uz ID-ZADATKA čini primarni ključ entiteta ZADATAK što znači da je entitet PROJEKT povezan identifikacijskom vezom s entitetom ZADATAK. Međutim ako je atribut ID-ZADATKA uvijek jedinstven bez obzira na PROJEKT, onda bi naslijeđeni atribut ID-PROJEKTA bio ne-ključni atribut entiteta ZADATAK. Takva veza između PROJEKTA i ZADATKA bi bila ne-identifikacijska.

Kod veze kategorizacije primarni ključ svih kategorijskih entiteta se nasljeđuje od primarnog ključa generičkog entiteta. Na primjer, ako su MJESEČNI-PLAĆENI-ZAPOSLENIK i ZAPOSLENIK-PLAĆEN-PO-SATU kategorijski entiteti generičkog entiteta ZAPOSLENIK, onda primarni ključ ŠIFRA-ZAPOSLENIKA entiteta ZAPOSLENIK će biti primarni ključ entiteta MJESEČNI-PLAĆENI-ZAPOSLENIK i ZAPOSLENIK-PLAĆEN-PO-SATU.

Vanjski ključ pišemo u kvadrat entiteta sa „FK“ u zagradama do atributa vanjskog ključa. Ako svi naslijeđeni atributi pripadaju primarnom ključu entiteta djeteta onda se ti atributi pišu iznad horizontalne linije i taj entitet ima zaobljene rubove što znači da je identifikator entiteta djeteta zavisen od primarnog ključa koji se nasljeđuje. Ako naslijeđeni atribut ne pripada primarnom ključu entiteta djeteta, vanjski ključ pišemo ispod horizontalne linije i onda entitet označavamo pravokutnikom, osim ako nije zavisen u nekoj drugoj vezi spoja. Naslijeđeni atributi mogu također biti i alternativni ključevi entiteta. Svaki entitet treba sadržavati vanjski ključ za svaku kategorizacijsku ili vezu spoja u kojoj je taj entitet kategorijski ili entitet djetete. Također broj atributa koji čine vanjski ključ treba biti jednak broju atributa koji čine primarni ključ roditeljskog ili generičkog entiteta (Technology, 1993).

#### 2.2.16.1.10 Pogled

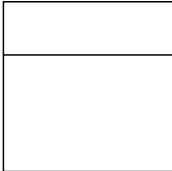
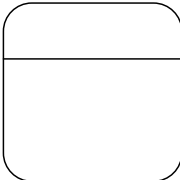

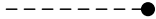
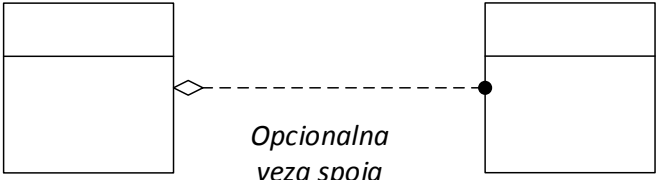
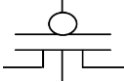
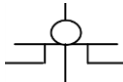
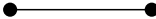
Jedan model podataka kod metode IDEF1X se može sastojati od više pogleda (*engl. views*). Pogled se najčešće sastoji od opisa entiteta, njihovih atributa, veza među entitetima, te ograničenja i pravila. Pogled obično predstavlja dio dijagrama (entitete, veze, attribute) koji su prikladni pojedinom korisniku.

Pogled u dijagramu Entiteti-Veze ne postoje, iako u Bazi podataka različiti pogledi koji su namijenjeni različitim korisnicima kao što su npr. studenti i profesori mogu postojati. Pogled se odnosi na ograničenje prikaza određenih podataka iz baze; entiteta tj. atributa. Pogledi postoje radi sigurnosti, radi očuvanja integriteta baze podataka, kao i ograničenja nastanka



promjena u bazi tako da su svakom korisniku prikazani oni podaci koji su mu namijenjeni (Technology, 1993).

Slike 28. i 29. prikazuju koncepte metode IDEF1X.

KONCEPT	SIMBOL
GENERIČKI, RODITELJSKI ENTITET	
ENTITET DIJETE	
VEZA SPOJA	<p>  <i>Identifikacijska veza</i>  <i>Ne-Identifikacijska veza</i> </p> <p>    <i>Opcionalna veza spoja</i>             Roditeljski entitet                      Entitet dijete         </p>
VEZA KATEGORIZACIJE	<p>    <i>Veza kategorizacije kod kompletnog skupa kategorijskih entiteta</i> </p> <p>    <i>Veza kategorizacije kod nepotpunog skupa kategorijskih veza</i> </p>
NE-SPECIFIČNA VEZA	

Slika 28 - koncepti IDEF1X metode (1)

KONCEPT	SIMBOL
<p style="text-align: center;">ATRIBUT</p>	<p style="text-align: center;">Entitet</p> <div style="border: 1px solid black; padding: 5px; margin: 0 auto; width: fit-content;"> <p style="text-align: center;">Atribut</p> <hr style="border: 0; border-top: 1px solid black; margin: 5px 0;"/> <p style="text-align: center;">Atribut Atribut</p> </div>
<p style="text-align: center;">PRIMARNI I VANJSKI KLJUČ</p>	<p style="text-align: center;">Entitet</p> <div style="border: 1px solid black; border-radius: 10px; padding: 5px; margin: 0 auto; width: fit-content;"> <p style="text-align: center;">Atribut (FK) Atribut</p> <hr style="border: 0; border-top: 1px solid black; margin: 5px 0;"/> <p style="text-align: center;">Atribut Atribut</p> </div>
<p style="text-align: center;">ASOCIJATIVNI ENTITET</p>	<p style="text-align: center;">ENTITET</p> <div style="border: 1px solid black; border-radius: 10px; padding: 5px; margin: 0 auto; width: fit-content;"> <p style="text-align: center;">Atribut (FK) Atribut (FK)</p> <hr style="border: 0; border-top: 1px solid black; margin: 5px 0;"/> <p style="text-align: center;"> </p> </div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> <span>—●</span> <span>●—</span> </div>

Slika 29 - koncepti IDEF1X metode (2)

## 3 Usporedba metode entiteti-veze metodologije MIRIS s metodom IDEF1X metodologije IDEF

MIRIS i IDEF su metodologije za razvoj informacijskih sustava. Metoda entiteti-veze (kod MIRIS-a) i metoda IDEF1X kao dio metodologije IDEF koriste se za modeliranje podataka sustava. Dijagrami metoda entiteti-veze i IDEF1X-a su slični, te imaju slične koncepte, međutim razlikuju se po nekim elementima i pravilima. Dijagrami obje metode se sastoje od entiteta, veza, atributa, domena, primarnih i vanjskih ključeva, te pravila po kojima ih koristimo.

Jedna od najvažnijih razlika IDEF1X-a od drugih metoda koje se uglavnom koriste za konceptualno modeliranje podataka, je to što IDEF1X koristi logičko modeliranje na konceptualnoj razini koristeći prije ne-specifične veze koje bi u toku postupka modeliranja postale specifične, te tvorile logički model podataka (Javor Bogati, 2012).

Identifikacijsku i ne identifikacijsku vezu kod IDEF1X-a označavamo sa punom ili isprekidanom linijom koje imaju na strani entiteta djeteta točku kraj koje se nalazi brojnost veze sa strane entiteta roditelja prema djetetu. Kategorijsku vezu kod IDEF1X-a možemo usporediti sa generalizacijskom vezom kod metode entiteti-veze. Ne specifična veza kod IDEF1X-a označava kada instanca jednog entiteta može biti povezana sa više instanci drugog entiteta, te instanca drugog entiteta može biti povezana sa više instanci prvog entiteta: tada uvodimo novi tip veze (sličan agregaciji kod EV metode). Također kod metode entiteti-veze kada imamo brojnosti veze po gornjoj granici M:M uvodimo agregaciju. Domene su kod obje metode vrlo slične samo što kod IDEF1X-a domene možemo podijeliti u hijerarhiju domena (Bazične i upisne), dok kod EV-a nemamo podjelu na nad-domene i poddomene. Pravila vezana za primarne i vanjske ključeve su vrlo slična samo što se vanjski ključevi kod metodologije MIRIS ne označavaju na dijagramu.

### 3.1 Usporedba koncepata EV metode i metode IDEF1X

#### 3.1.1 Entitet

Entitet IDEF1X metode je sličan tipu Entiteta kod metode EV. Entitet u obje metode predstavlja dio stvarnog svijeta ili neki apstraktni pojam kao što su ljudi, objekti, mjesta, događaji, dokumenti, ideje itd. Jedno konkretno pojavljivanje tipa entiteta u metodologiji MIRIS nazivamo instancom entiteta ili jednostavno entitetom, dok kod metode IDEF1X jedno pojavljivanje entiteta nazivamo instanca entiteta. U IDEF1X metodi slične entitete koje dijele neke zajedničke karakteristike i attribute možemo objediniti u jedan **Entitet**. Slično kao i kod IDEF1X-a entitete koji imaju zajednička svojstva možemo klasificirati u jedan **Tip Entiteta** kod EV metode.

Entitet u metodi entiteta i veza označavamo s pravokutnikom unutar kojeg upisujemo naziv entiteta. Kod IDEF1X metode entitet označavamo pravokutnikom iznad kojeg upisujemo

naziv tog entiteta. Kod metode EV, kao i kod metode IDEF1X pojedinačna pojavljivanja entiteta se ne označavaju na modelu. Svaki entitet, odnosno tip entiteta može imati jedno ili više pojedinačnih pojavljivanja.

### 3.1.2 Domena i tip vrijednosti

Domena kod metode IDEF1X i tip vrijednosti u metodi EV su vrlo slični. U metodi EV atributi mogu poprimiti vrijednosti iz nekog određenog tipa domene, npr: datumi, ocjene, imena, prezimena, nazivi država, itd. Tip vrijednosti (ili domena) je skup pojedinačnih vrijednosti koje dijele neku zajedničku karakteristiku. Tipovi vrijednosti se u EV metodi ne označavaju posebno na dijagramu, ali ih možemo prikazati pomoću elipse unutar koje se nalazi ime, kao i neke instance tog tipa vrijednosti.

Tip podataka (*engl. data type*) je oznaka vrste podataka u tipu vrijednosti. Tip podataka može biti datum, slova, znakovi, alfanumerički znakovi (kombinacija brojeva i slova), brojevi,... Svaki tip vrijednosti sadrži podatke samo jednog tipa. Tako u tipu vrijednosti Imena imamo podatke tipa „slovo“ ili u tipu vrijednosti Datumi podatke tipa „datum“ (Pavlic, Oblikovanje baza podataka, 2011).

Ako za tip vrijednosti odredimo tip podataka onda smo tipu vrijednosti dodali ograničenje. Također, u metodi IDEF1X imamo Domene. Domena je skup vrijednosti iz kojih atributi mogu poprimiti svoju vrijednosti. Domene također mogu imati ograničenje na tip podataka, dok se kod ove metode pojavljuju dvije vrste domena: Bazična i upisna. Bazična domena najčešće ima određen tip podataka, dok kod upisne domene imamo dodatna pravila koja su definirana sa svakom razinom hijerarhije. Svaka nova razina upisne domene sadrži sva ograničenja bazične i domena viših razina hijerarhije ako postoje, kao i svoja vlastita pravila. Različite razine domena u metodi EV ne postoje.

Ograničenja na dopuštene vrijednosti u tipu vrijednosti možemo definirati uz pomoć ograničenja (Pavlic, Oblikovanje baza podataka, 2011):

- Točnim navođenjem svih dopuštenih vrijednosti (na primjer, ocjene od 1 do 5)
- Definiranjem intervala iz kojeg se vrijednosti mogu pojaviti
- Određivanjem tipa podataka dopuštenih u tipu vrijednosti
- Određivanje minimalne i maksimalne dužine zapisa podataka
- Određivanjem oblika zapisa podataka,

Na primjer, nekakav skup datuma u životopisu neke osobe je skup datuma (tip podataka) koji je veći ili jednak od datuma rođenja, a manji od današnjeg datuma (ograničenje na raspon podataka), pri čemu datum treba biti oblika DD.MM.GGGG (DD – dan, MM – mjesec, GGGG – godina) (Pavlic, Oblikovanje baza podataka, 2011).

Također, jedno od ograničenja može biti i ograničenje na dužinu zapisa podataka, npr. Ime osobe može imati 15 znakova. Kao i kod metode EV, u metodi IDEF1X postoje različita ograničenja na domene. Ograničenja mogu biti na tip podataka; najčešći tipovi podataka su

znakovni, numerički ili boolean, ali mogu se pojaviti i drugi tipovi kao što su datumi, binarni, itd. Ograničenja također mogu biti na oblik zapisa podataka.

Slično kao i kod metode EV, domenama mogu biti dodijeljena i pravila:

- Lista vrijednosti
- Raspon vrijednosti

Lista vrijednosti – predstavlja da vrijednosti nekih domena mogu biti samo iz neke određene liste vrijednosti, npr. šifra države SAD-a može biti samo jedna od kratica koje pripadaju listi kratica svake savezne države, npr: AZ (Arizona), AK (Alaska), AL (Alabama),... Raspon vrijednosti – vrijednosti neke domene mogu biti samo u određenom rasponu.

Uglavnom, sva ograničenja metode EV, bez ograničenja na dužinu zapisa podataka, se mogu primijeniti i na IDEF1X domene.

### 3.1.3 Atribut

Atribut je kod metode EV funkcija koja preslikava tip entiteta u tip vrijednosti. Kao i kod metode EV, na IDEF1X modelu atribut predstavlja neku karakteristiku, obilježje entiteta. Atribut kod metode IDEF1X povezuje neku domenu (tip vrijednosti kod MIRIS-a) sa (nekim konkretnim) entitetom (tipom entiteta kod metode EV).

Kao i kod metode EV, u metodi IDEF1X atributi mogu poprimiti NULL vrijednost ako nisu dio primarnog ključa (ključni atributi). Kod metode IDEF1X takvim atributima se dodaje „O“ (kao „Optional“) pored imena. Također u obje metode treba postojati atribut ili skup atributa koji jedinstveno određuje svaku instancu nekog entiteta. Takav atribut se naziva Primarni ključ. Kod IDEF1X metode primarni ključ roditeljskog entiteta postaje dio primarnog ključa naslijeđenog entiteta ako je entitet zavisian o identifikatoru, dok kod entiteta nezavisnog od identifikatora primarni ključ roditeljskog entiteta postaje vanjski ključ naslijeđenog entiteta. Pored svakog naslijeđenog atributa u ovoj metodi se piše nastavak „FK“ u zagradi što označava vanjski ključ (*engl. Foreign key*). U obje metode samo atributi koji pripadaju primarnom ključu mogu prijeći iz jednog entiteta (tipa entiteta) u drugi.

U EV metodi kao i kod metode IDEF1X atributi trebaju imati jedinstveno ime u cijelom dijagramu, samo kod IDEF1X-a atributi mogu imati više različitih aliasa.

Kod metode EV vanjski ključevi se ne označavaju na modelu. Atributi kod metode EV i IDEF1X trebaju biti imenice u jednini. Kod metode EV atributi su prikazani unutar elipse, dok se u IDEF1X metodi atributi nalaze unutar pravokutnika entiteta.

Kod metode IDEF1X atributi koji čine primarni ključ su prikazani na vrhu liste atributa i odvojeni su horizontalnom linijom od ne-ključnih atributa, dok su kod metode EV atributi koji čine primarni ključ podcrtani i nalazi se prije ne-ključnih atributa. Svaki ključni atribut kod obje metode mora imati vrijednost, ne smije biti NULL. Kod obje metode isto vrijedi da svaka instanca entiteta ne smije imati više od jedne vrijednosti.

### 3.1.4 Veza

Ne-identifikacijska veza kod IDEF1X-a povezuje entitet roditelj s entitetom djetetom koji je neovisan o identifikatoru. Takva veza se označava sa isprekidanim crticama sa točkom na kraju entiteta djeteta. Kod ne-identifikacijske veze primarni ključ entiteta roditelja se nasljeđuje u entitet dijete, te postaje vanjski ključ entiteta djeteta. Na strani entiteta djeteta se piše kardinalnost (osim ako se ne radi o standardnoj brojnosti). Ne-identifikacijska veza je slična relaciji između dva entiteta kod metode EV. Primarni ključ se spušta u onaj entitet sa čije je strane veze brojnost (1,1) dok je na drugoj strani veze brojnost (1,M).

Identifikacijska veza kod IDEF-a je slična vezi između slabog i jakog tipa entiteta kod metode EV. Kod identifikacijske veze entitet dijete ne može biti jedinstveno identificiran u dijagramu. Identifikacijsku vezu pišemo punom linijom sa točkom na strani entiteta djeteta. Kod identifikacijske veze primarni ključ roditelja postaje dio primarnog ključa entiteta djeteta. Kod EV metode slabi tip entiteta također ovisi o jakom tipu entiteta. Slabi tip ne može postojati ili se ne može jedinstveno odrediti ako se radi o egzistencijalnoj, odnosno identifikacijskoj vezi. Veza kategorizacije kod IDEF1X-a je slična generalizacijskoj vezi kod metodologije MIRIS. Kod kategorizacijske veze jedan entitet se može podijeliti na više kategorija, slično kao kod generalizacijske veze. Kod obje veze potkategorije se međusobno isključuju, tj. entitet koji pripada kategoriji zaposlenik može pripadati samo jednoj od potkategorija mjesečno plaćeni zaposlenik ili zaposlenik plaćen po satu.

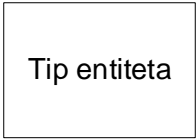
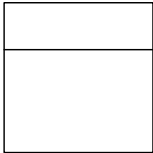

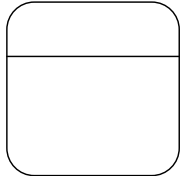
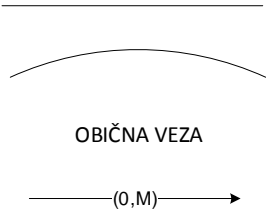
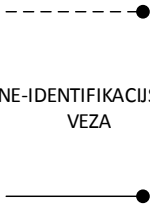
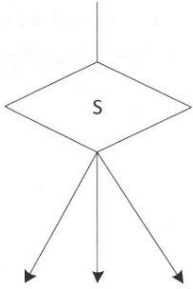
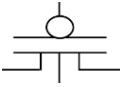
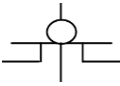
Kod ne-specifične veze kod metode IDEF1X instanca jednog entiteta može biti povezana sa više instanci drugog entiteta kada se uvodi novi entitet koji je zavisan o dva entiteta koja povezuje, te može imati svoje atribute. Slično kao kod IDEF1X-a, u EV metodi kada veza između dva entiteta ima brojnost M:M s obje strane nastaje novi entitet agregacija, koja također može imati vlastite atribute.

### 3.1.5 Primarni i vanjski ključ

Kod metodologije MIRIS primarni i vanjski ključevi su slični kao i kod metode IDEF1X. Primarni ključ je podcrtan i piše se iznad ne-ključnih atributa kod MIRISA, dok kod IDEF1X-a primarni se ključ piše iznad horizontalne linije, dok se ostali ne-ključni atributi pišu ispod linije u pravokutniku entiteta. Nakon ne-ključnih atributa pišemo i vanjske ključeve i dodajemo FK u zagradama do imena vanjskog ključa. Kada se radi o entitetu zavisnom o identifikatoru primarni ključ entiteta roditelja postaje dio primarnog ključa entiteta djeteta. Vanjske ključeve kod metode EV ne označavamo na dijagramu.


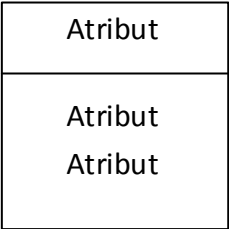

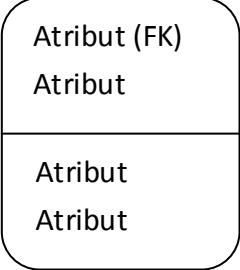

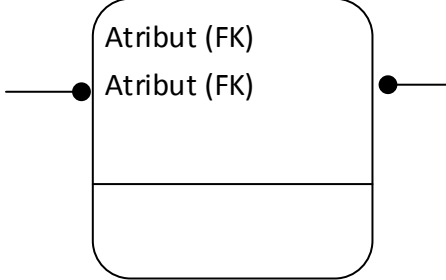
## 3.2 Usporedba konceptata metode entiteta i veza i metode IDEF1X

Slike 30. i 31. prikazuju usporedbu konceptata metode entiteta i veza i metode IDEF1X.

KONCEPT	
MIRIS	IDEF1X
 <p>Tip entiteta</p> <p>JAKI TIP ENTITETA</p>	 <p>GENERIČKI, RODITELJSKI ENTITET</p>
 <p>SLABI TIP ENTITETA</p>	 <p>ENTITET DIJETE</p>
 <p>OBIČNA VEZA</p> <p>VEZA KOJA POVEZUJE JAKI I SLABI TIP ENTITETA</p>	 <p>NE-IDENTIFIKACIJSKA VEZA</p> <p>IDENTIFIKACIJSKA VEZA</p>
 <p>GENERALIZACIJA/ SPEVICIJALIZACIJA</p>	 <p>ILI</p>  <p>VEZA KATEGORIZACIJE</p>

Slika 30 - Usporedba konceptata MIRIS-a i IDEF1X-a (1)

## KONCEPT

MIRIS	IDEF1X
 <p style="margin-top: 10px;"><b>ATRIBUT</b></p>	 <p style="margin-top: 10px;"><b>ATRIBUT</b></p>
 <p style="margin-top: 10px;"><b>PRIMARNI KLJUČ</b></p> <p style="margin-top: 10px;">*metodologija MIRIS ne označava FK na dijagramu</p>	 <p style="margin-top: 10px;"><b>PRIMARNI I VANJSKI KLJUČ</b></p>
	<p style="margin-bottom: 5px;"><b>ENTITET</b></p> 

Slika 31 - Usporedba koncepta MIRIS-a i IDEF1X-a (2)



## 4 Metode entiteti-veze i IDEF1X na praktičnom primjeru

U ovom poglavlju opisat ću primjenu metode entiteti-veze i IDEF1X na praktičnom primjeru firme Ponikve Voda, poslovni sustav skladišta. Ovdje su, uz opis sustava, priloženi dokumenti zajedno s njihovim pripadnim opisima i tablicama podataka. Također prikazani su i dijagrami tih dokumenata modelirani metodama entiteti-veze i IDEF1X.

### 4.1 Opis sustava

Ponikve Krk d.o.o. komunalno je društvo koje se bavi uslugama vodoopskrbe, odvoza i zbrinjavanja komunalnog otpada, pročišćavanjem otpadnih voda, te energetikom u koju spada javna rasvjeta.

Društvo je osnovano 1960. godine s osnovnom djelatnošću proizvodnje i distribucije vode. Od osnutka do danas sjedište Društva nalazi se u Krku. Godine 1986. došlo je do udruživanja PONIKAVA i KOMUNALCA iz Omišlja. Novonastalo društvo djeluje pod imenom PONIKVE. Nakon udruživanja, djelatnost Društva proširuje se na odvoz i odlaganje otpada, održavanje čistoće u poslovnim prostorima, te čišćenje javnih i zelenih površina. U proljeće 2005. godine Društvo je uvelo ekološki sustav zbrinjavanja otpada. Tome je sustavu cilj što veći dio otpada odvojeno prikupiti i reciklirati, a samo ono što se ne uspije odvojiti, odložiti na odlagalištu komunalnog otpada TRESKAVAC kojim upravlja Društvo. Ponikve djeluju na cijelom području otoka Krka.

U skladu sa Zakonom o vodama sva komunalna društva morala su do 31.12.2013. godine iz svog poslovanja izdvojiti sve nevodne djelatnosti. Kako je Ponikve d.o.o. Krk osim javne vodoopskrbe i javne odvodnje obavljala i djelatnost gospodarenja otpadom, provela je nužne statusne promjene koje su na snazi od 01.siječnja 2014. godine. Dosadašnje komunalno društvo Ponikve d.o.o. osim djelatnosti promijenilo je i naziv u **PONIKVE VODA d.o.o.** za javnu vodoopskrbu i javnu odvodnju te pod tim nazivom posluje od 01.01.2014. godine. Osnovane su dva nova društva: **PONIKVE EKO OTOK KRK d.o.o.** za obavljanje komunalnih djelatnosti: gospodarenje otpadom i energetika. **PONIKVE USLUGA d.o.o.** za obavljanje zajedničkih uslužnih djelatnosti (knjigovodstveni poslovi, objedinjena naplata vodnih i komunalnih usluga, razni administrativni poslovi i slično). Članovi društva Ponikve d.o.o. Krk su sve jedinice lokalne samouprave na otoku Krku koje u promjenjivim omjerima sudjeluju u vlasništvu društva.

Skladište Ponikve Voda d.o.o. bavi se uslugama Nabave, skladištenja, te izdavanja robe različitim klijentima zaposlenim u sklopu firme. Skladište čuva, te izdaje različite vrste robe kao što je građevni materijal (ventili, cijevi, poklopci,...), alat te razni uređaji, te sredstva za osobnu, te protupožarnu zaštitu (zaštitna odjeća i obuća, te protupožarni aparati) kao i narudžbom i izdavanjem potrošnog materijala (papir, toneri,...) (Ponkve, 2018).

Na slici 32. je prikazan logo firme Ponikve Voda.



Slika 32 - logo firme Ponikve Voda

## 4.2 Priloženi dokumenti s pripadnim opisom i dijagramima podataka

U ovom poglavlju su prikazani dokumenti sustava Ponikve voda zajedno sa tablicama u kojima je prikazana analiza podataka na dokumentima, te njihovim modelima podataka napravljenim metodama entiteti-veze i IDEF1X.

Na slici 33. prikazan je dokument narudžba dobavljaču, kod kojeg tablica 2. prikazuje analizu tog dokumenta. Također na slikama 34. i 35. prikazani su modeli narudžbe dobavljaču opisani metodama entiteti-veze i IDEF1X. Na slici 36. se nalazi dokument povratnica, uz koji je u tablici 3. prikazana analiza tog dokumenta, kao i model EV na slici 37., te IDEF1X model na slici 38.

**Ponikve voda d.o.o.**  
Vršanska 14, 51500 Krk

**NARUDŽBA DOBAVLJAČU**

Dokument: 1 - 198 / 2016

Datum: 23.11.2016

Način narudžbe: 5 - E-mail  
Način dobave: 2 - kamion dobavljača  
Vrsta dobave: 0101 - prijem Hrvatska  
Vrsta preuzimanja: FCO - franko naše skladište  
Način plaćanja: 5 - Virman - 30 daan

Prijemno skladište: 1002 - CENTRALNO SKLADISTE  
Vršanska 14,

Dobavljač: 87322

Porezni broj: 3519949934

**CNM & CO. d.o.o.**

**Ružiči 1a**

**51213 Jurdani**

Izdavač računa: 87322  
CNM & CO. d.o.o.

Porezni broj: 3519949934

Ružiči 1a  
51213 Jurdani

Dokument dobavljača:  
Ugovor:

Pozicija	Proizvod / usluga	Cijena u valuti		Cijena u kn	Popust%	Cijena s pop.	Porez %	JM Datum isporuke	Cijena s porezom	Količina	Vrijednost
		Valuta	Cijena u valuti								
1	778988 POKLOPAC kan.600x400 tip KRK-15 T "VODOVOD KRK" /	HRK	0,00	685,00	0,00	685,00	25,00	kom 23.11.2016	856,25	20,000	17.125,00
Zajedničke pozicije: 1										<b>Ukupna vrijednost narudžbe:</b>	<b>17.125,00</b>

Rukovoditelj nabave: KIRINČIĆ VEDRAN

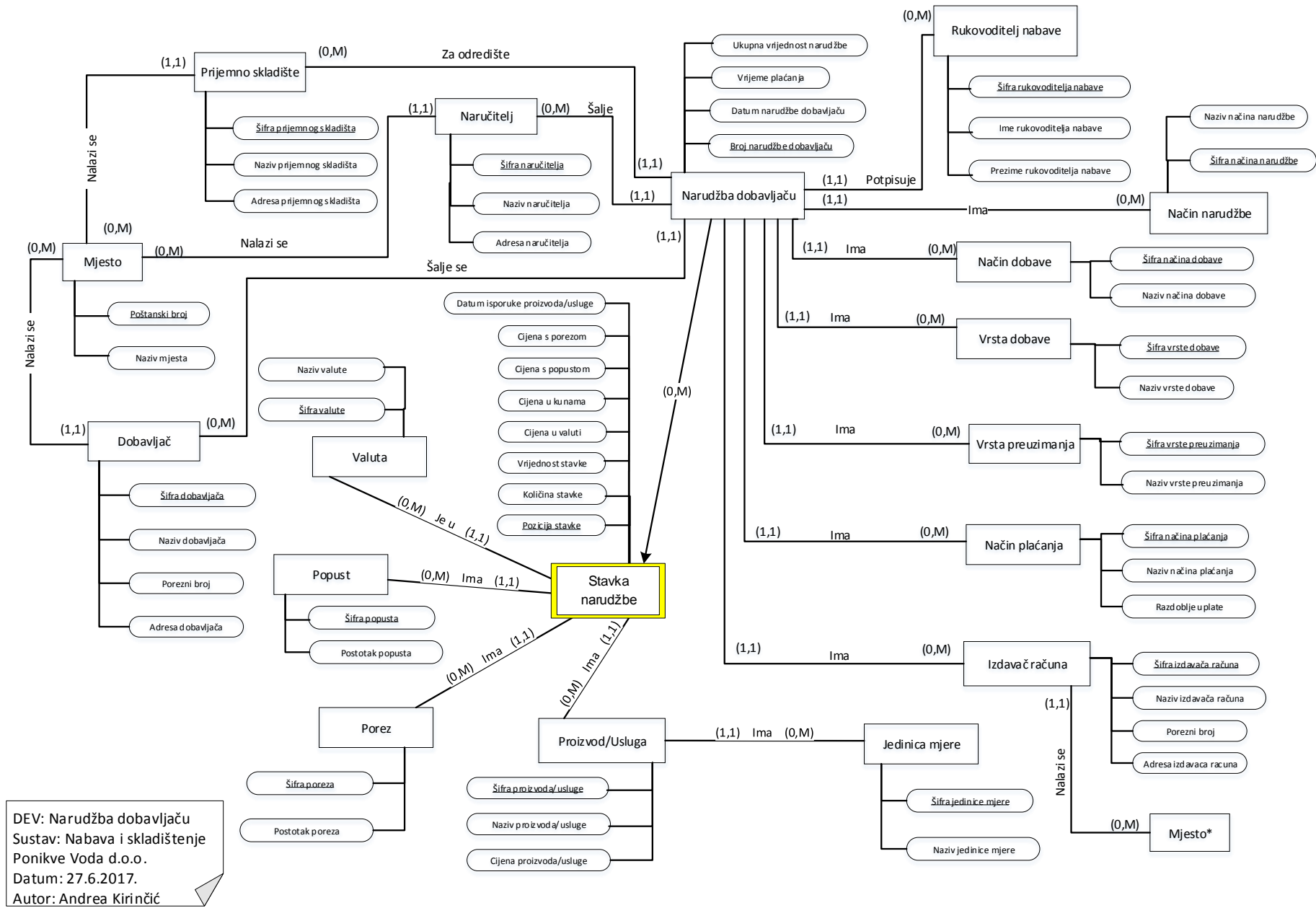
M.P.

Tablica 2 - analiza narudžbe dobavljaču

NAZIV DOKUMENTA		Narudžba dobavljaču		
OPIS DOKUMENTA		Dokument koji naručitelj (skladište) šalje dobavljaču, a na kojemu su definirani nazivi, cijena, količina proizvoda, te ime dobavljača kojemu se šalje narudžba.		
Rbr.	Naziv podatka	Opis podatka	Primjer	Tip i duljina
1.	Naziv tvrtke	Naziv tvrtke koja naručuje robu	Ponikve Voda d.o.o.	S30
2.	Adresa tvrtke	Adresa tvrtke	Vršanska 14	S30
3.	Poštanski broj tvrtke	Poštanski broj tvrtke	51500	N5
4.	Grad/mjesto sjedišta tvrtke	Grad/mjesto sjedišta tvrtke	Krk	S30
5.	Narudžba dobavljaču	Narudžba dobavljaču	Narudžba dobavljaču	S30
6.	Dokument	Broj dokumenta	1-198/2016	S10
7.	Datum	Datum izrade dokumenta	23.11.2016	D06
8.	Šifra načina narudžbe	Šifra načina narudžbe	5	N2
9.	Naziv načina narudžbe	Naziv načina narudžbe	E-mail	S20
10.	Šifra načina dobave	Šifra načina dobave robe	2	N2
11.	Naziv načina dobave	Naziv načina dobave robe	Kamion dobavljača	S30
12.	Šifra vrste dobave	Šifra vrste dobave	0101	N4
13.	Naziv vrste dobave	Naziv vrste dobave	Prijem Hrvatska	S30
14.	Šifra vrste preuzimanja	Šifra vrste preuzimanja	FCO	S3
15.	Naziv vrste preuzimanja	Naziv vrste preuzimanja	franko naše skladište	S50
16.	Šifra načina plaćanja	Šifra načina plaćanja	5	N1
17.	Naziv načina plaćanja	Naziv načina plaćanja	Virman	S20

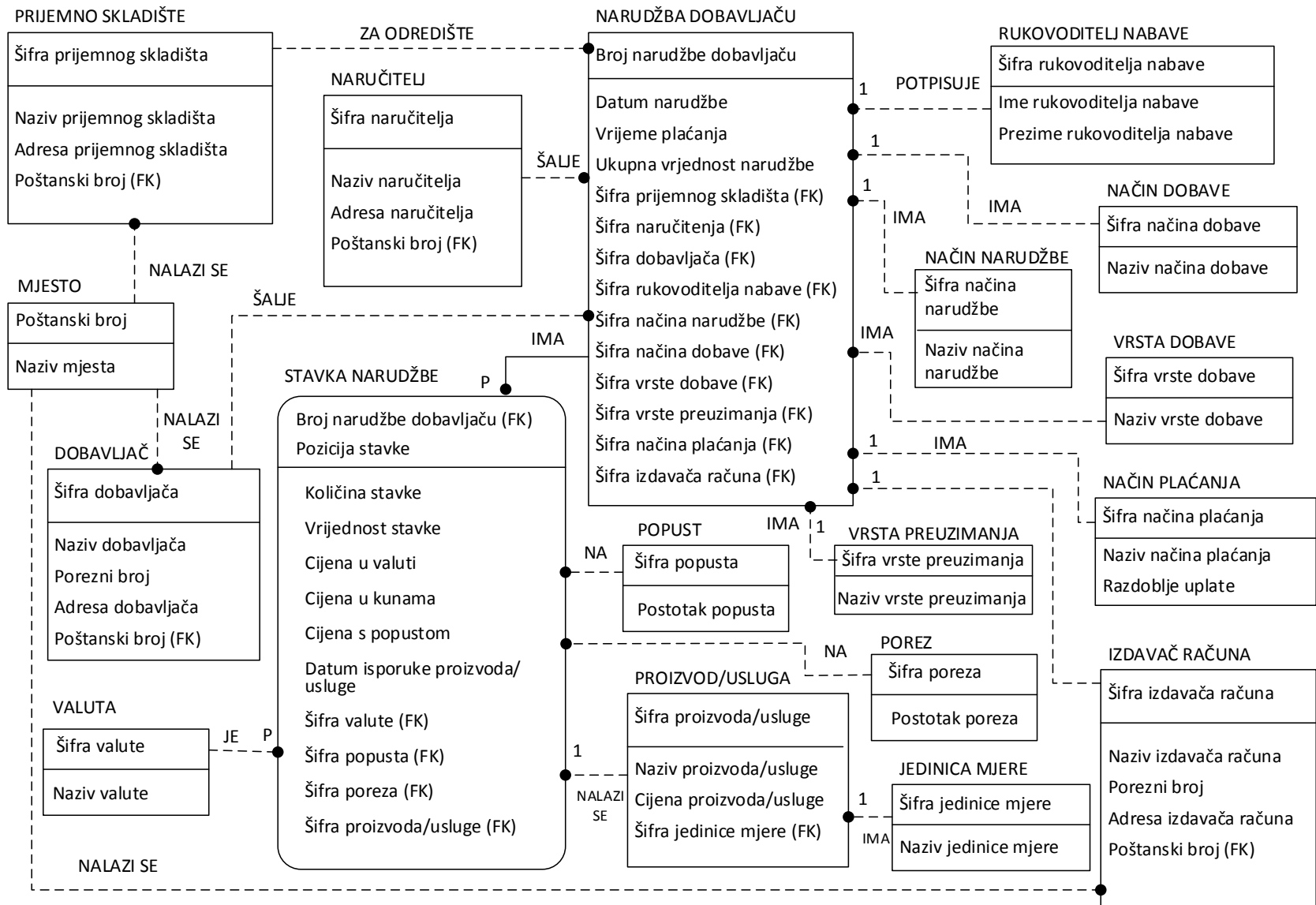
18.	Razdoblje uplate (dani)	Razdoblje uplate (dani)	30	S10
19.	Šifra prijemnog skladišta	Šifra prijemnog skladišta	1002	N4
20.	Vrsta prijemnog skladišta	Vrsta prijemnog skladišta	Centralno skladište	S30
21.	Adresa prijemnog skladišta	Adresa prijemnog skladišta	Vršanska 14	S30
22.	Šifra dobavljača	Šifra dobavljača	87322	N5
23.	Šifra dobavljača	Porezni broj dobavljača	3519949934	N10
24.	Naziv dobavljača	Naziv dobavljača	CNM & CO. d.o.o.	S30
25.	Adresa dobavljača	Adresa dobavljača	Ružići 1a	S30
26.	Poštanski broj	Poštanski broj dobavljača	51213	N5
27.	Grad sjedišta dobavljača	Grad/mjesto sjedišta dobavljača	Jurdani	S30
28.	Šifra izdavača računa	Šifra izdavača računa za naručenu robu	87322	N5
29.	Porezni broj izdavača računa	Porezni broj izdavača računa	3519949934	N10
30.	Naziv izdavača računa	Naziv izdavača računa	CNM & CO. d.o.o.	S30
31.	Adresa izdavača računa	Adresa izdavača računa	Ružići 1a	S30
32.	Poštanski broj izdavača računa	Poštanski broj izdavača računa	51213	N5
33.	Grad/mjesto izdavača računa	Grad/mjesto izdavača računa	Jurdani	S30
34.	Dokument dobavljača	Dokument dobavljača		S30
35.	Ugovor	Ugovor		S50
36.	Pozicija	Pozicija proizvoda/usluge na listi	1	N2
37.	Šifra Proizvoda/usluge	Šifra proizvoda/usluge	778988	N10

38.	Naziv proizvoda/usluge	Naziv proizvoda/usluge (koji se naručuje)	POKLOPAC kan.600x400 tip KRK-15 T „VODOVOD KRK“ /	S50
39.	JM	Mjerna jedinica u kojoj se naručuje proizvod	kom	S3
40.	Datum isporuke	Datum isporuke robe	23.11.2016	D06
41.	Količina	Količina robe	20,000	N10.3
42.	Valuta	Naziv valute u kojoj će se izvršiti plaćanje	HRK	S3
43.	Cijena u valuti	Cijena proizvoda u valuti	0,00	N10.2
44.	Cijena u kn	Cijena proizvoda u kunama	685,00	N10.2
45.	Popust %	Postotak popusta	0,00	N2.2
46.	Cijena s pop.	Cijena proizvoda s popustom	685,00	N10.2
47.	Porez %	Postotak poreza	25,00	N2.2
48.	Cijena s porezom	Cijena proizvoda s porezom	856,25	N10.2
49.	Vrijednost	Umnožak cijene pojedinog proizvoda i količine	17.125,00	N10.2
50.	Ukupna vrijednost narudžbe	Ukupna vrijednost narudžbe u kunama	17.125,00	N10.2
51.	Prezime rukovoditelja nabave	Prezime rukovoditelja nabave	Kirinčić	S30
52.	Ime rukovoditelja nabave	Ime rukovoditelja nabave	Vedran	S30
53.	Potpis odgovorne osobe	Potpis odgovorne osobe		S50
54.	Broj stranice	Broj stranice	Stran 1 od 1	S20



Slika 34 - EV model dokumenta narudžba dobavljaču





Slika 35 – IDEF1X model dokumenta narudžba dobavljaču

Ru 873

Ponikve voda d.o.o.  
SKLADIŠTE: 1002 - CENTRALNO SKLADISTE  
Vršanska 14,

**povratnica**

Dokument: 0502 - 2457 / 2016 - *br. dokument*

Datum: 22.11.2016

51133

Dobavljač/kupac:1  
Ponikve voda d.o.o.

Dokument:  
Datum: 00.00.0000

**Izdao**

Zaposlenik: 4026 ZEC VEDRAN  
OJ: 1101 Voda  
TM: 1101511 malinska

Dokument izradio: 4308 KIRINČIĆ VEDRAN

Datum potvrde: 22.11.2016 08:53:56

Razdoblje: 201611 *br. plit*

Vrsta veza dok.: 1002 CENTRALNO SKLADISTE  
Veza dokument: 2016 2213

**Transport:**

Partner:

Veza skladište: 0 - nema podatka nema podatka

Pozicija	Predmet rada	Cijena	Cijena skladištenja	JM	Količina
1	10259 / 1 CIJEV DUKTIL DN 100	186,22	132,02	M	15,000

Br.stavaka: 1

Primio:

*[Handwritten signature]*

Izdao:

*[Handwritten signature]*

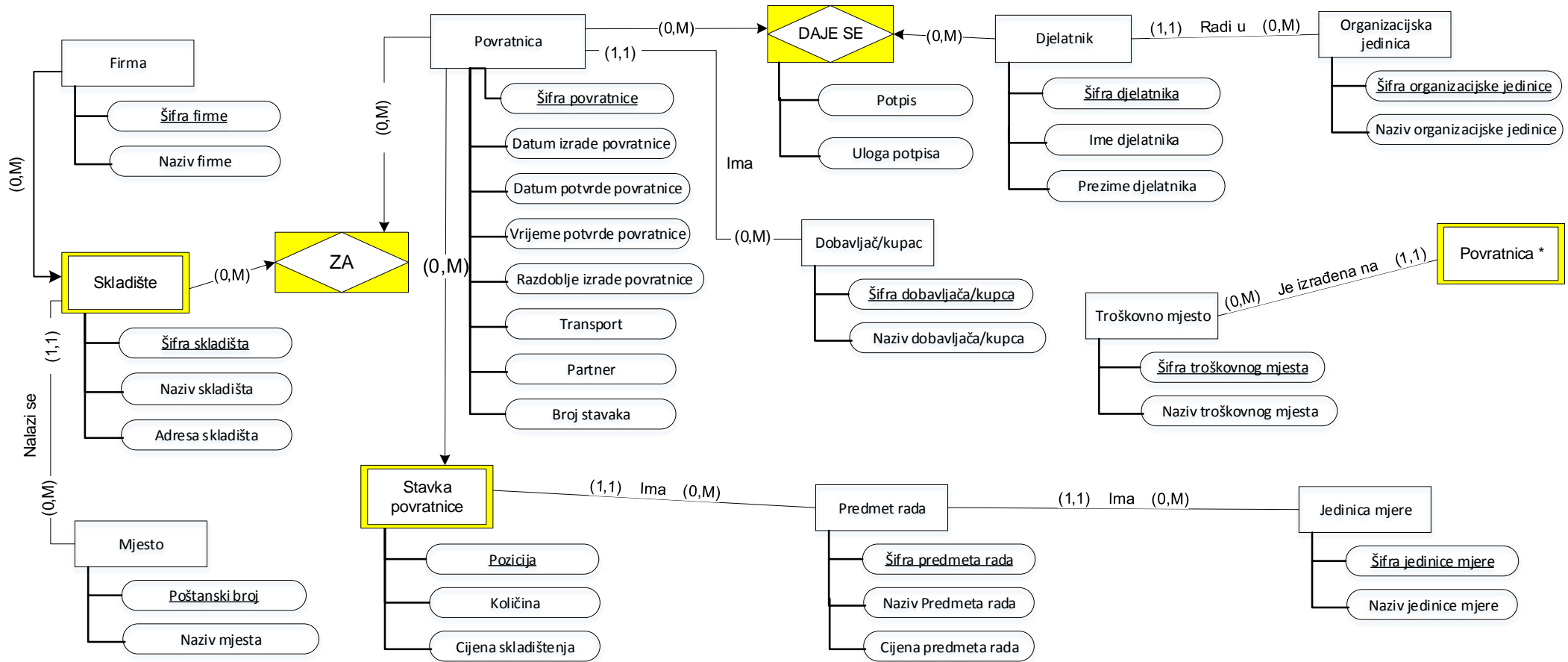
Slika 36 - dokument povratnica

Tablica 3 – analiza dokumenta povratnica

NAZIV DOKUMENTA		<b>Povratnica</b>		
OPIS DOKUMENTA		U slučaju da sav materijal izdan putem izdatnice nije iskorišten, taj materijal se vraća na skladište, te radnik od skladišta dobiva povratnicu. Dokument povratnica također mijenja stanje u e-BP dodajući (vraćajući) robu na stanje.		
Rbr.	Naziv podatka	Opis podatka	Primjer	Tip i duljina
1.	Naziv tvrtke	Naziv tvrtke koja izdaje robu sa skladišta	Ponikve voda d.o.o.	S30
2.	Šifra skladišta	Šifra skladišta	1002	N4
3.	Vrsta skladišta	Vrsta skladišta	Centralno skladište	S30
4.	Adresa skladišta	Adresa skladišta	Vršanska 14	S30
5.	Naziv dokumenta	Naziv dokumenta	Povratnica	S30
6.	Šifra dokumenta	Šifra dokumenta	0502-2457/2016	S14
7.	Datum	Datum izdavanja dokumenta	22.11.2016	D06
8.	Šifra radnika	Šifra radnika koji je izradio dokument	4308	N4
9.	Prezime radnika	Prezime radnika koji je izradio dokument	Kirinčić	S30
10.	Ime radnika	Ime radnika koji je izradio dokument	Vedran	S30
11.	Datum potvrde	Datum potvrde dokumenta	22.11.2016	D06
12.	Sat potvrde	Sat potvrde dokumenta	08.53.56	T
13.	Razdoblje	Mjesec i godina nastanka dokumenta	201611	N6
14.	Šifra skladišta	Šifra skladišta za vezu s BP	1002	N4
15.	Vrsta skladišta	Vrsta skladišta za vezu s BP	Centralno skladište	S30
16.	Datum veze s BP	Datum veze s BP	23.11.2016	D06
17.	Šifra izdatnice	Šifra izdatnice za vezu s BP	2213	N4

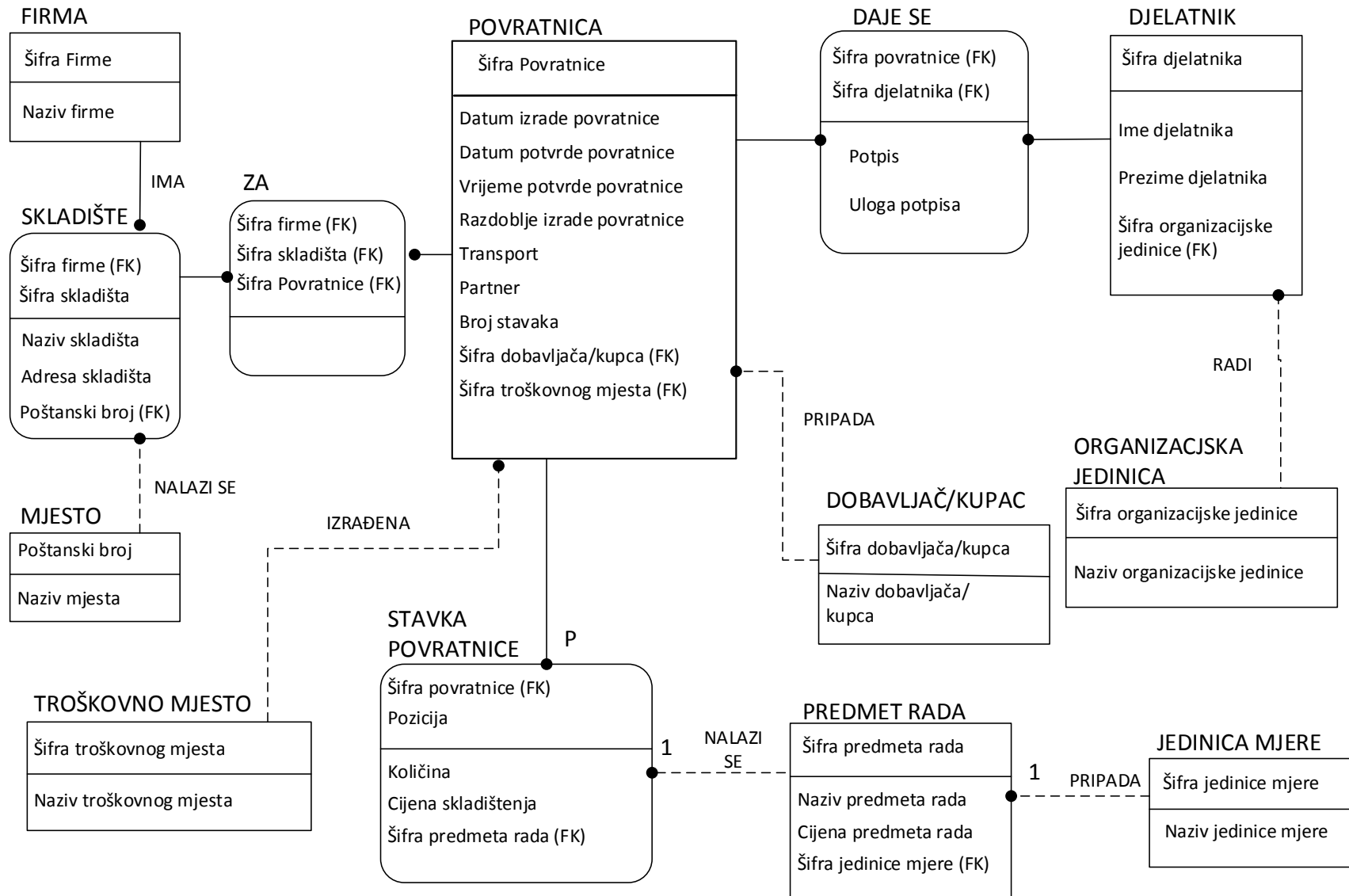
18.	Broj veze	Broj veze sa skladištem	0	N2
19.	Vrsta veze	Vrsta veze sa skladištem	Nema podataka nema podataka	S20
20.	Broj Dobavljača/Kupca	Broj dobavljača/kupaca koji vraćaju robu na skladište	1	N2
21.	Ime firme	Ime firme dobavljača/kupca	Ponikve voda d.o.o.	S30
22.	Datum povrata robe	Datum povrata robe	00.00.0000	D06
23.	Šifra zaposlenika	Šifra zaposlenika koji vraća robu na skladište	4026	N4
24.	Prezime zaposlenika	Prezime zaposlenika	Zec	S30
25.	Ime zaposlenika	Ime zaposlenika	Vedran	S30
26.	Šifra OJ	Šifra organizacijske jedinice radnika koji vraća robu	1101	N4
27.	Ime OJ	Ime organizacijske jedinice radnika koji vraća robu	Voda	S30
28.	Šifra TM	Šifra troškovnog mjesta	1101511	N10
29.	Naziv TM	Naziv troškovnog mjesta	Malinska	S30
30.	Transport	Način transporta robe u skladište		S30
31.	Partner	Partner za pomoć pri transportu robe		S30
32.	Pozicija	Pozicija stavke na dokumentu	1	N2
33.	Šifra Predmeta rada	Šifra predmeta rada koji se vraća na skladište	10259/1	S12
34.	Naziv Predmeta rada	Naziv predmeta rada koji se vraća na skladište	CIJEV DUKTIL DN 100	S50
35.	Cijena	Cijena proizvoda	186,22	N10.2
36.	Cijena skladištenja	Cijena skladištenja proizvoda	132,02	N10.2
37.	JM	Mjerna jedinica	M	S3
38.	Količina	Količina proizvoda	15,000	N5.3
39.	Br. stavaka	Broj stavaka koji se vraćaju na skladište	1	N2

40.	Primio	Potpis radnika koji zaprima vraćenu robu		S50
41.	Izdao	Potpis radnika koji vraća robu na skladište		S50
42.	Broj stranice	Broj stranice	Stranica 1 od 1	S20



DEV: Povratnica  
 Sustav: Nabava i skladištenje  
 Ponikve Voda d.o.o.  
 Datum: 21.09.2018.  
 Autor: Andrea Kirinčić

Slika 37 - EV model dokumenta povratnica



Slika 38 – IDEF1X model dokumenta povratnica

# Zaključak

Metodologija MIRIS jedna je od poznatijih metodologija i vrlo se često primjenjuje u praksi za razvoj informacijskih sustava. MIRIS se sastoji od više faza razvoja sustava kao što su strateško planiranje pri kojem se prije same izrade softverskog proizvoda razmišlja što će se i kako napraviti, priprema se tim, prikuplja dokumentacija i planira se izgradnja sustava. Nakon planiranja slijedi faza glavnog projekta gdje se kroz razgovor s korisnicima gradi slika sustava i modeliraju se procesi. Nakon toga slijedi izvedbeni projekt u kojemu se gradi model podataka sustava, te prevodi taj model u shemu baze podataka, te definira arhitektura programskog proizvoda. Sama proizvodnja softvera sastoji se od više faza poput planiranja proizvodnje, razvoja i testiranja programa, nakon čega slijedi uvođenje softvera u sam sustav. Na kraju sustav se primjenjuje u radnoj okolini i održava. Dakle MIRIS, zahvaljujući jasno definiranim fazama osigurava dobru pripremu i upoznavanje sa sustavom prije izrade samog softvera, te njegovog korištenja u sustavu. Više definiranih faza i aktivnosti daje mogućnost da se temeljno prouči svaki dio razvoja sustava, te što više smanji mogućnost pogrešaka, kao i poboljšaju šanse za uspjeh firme i konkurentnost na tržištu. MIRIS je zbog svoje razumljivosti, dostupnosti i jednostavnosti razvoja sustava od strateškog planiranja pa sve do uvođenja i održavanja jedna od najkorištenijih i najboljih metodologija u praksi.

IDEF metodologija je razvijena od strane američkog zrakoplovstva i u početku se koristila za potrebe vojske, a kasnije se počinje koristiti i u široj primijeni za pomoć organizacijama u poslovanju. IDEF je zbog svog širokog područja primjene i velikog broja metoda također zastupljena u praksi. Prednost ove metodologije je dobra integracija različitih metoda u jednu cjelinu. Prednost IDEF1X-a je standardizacija metode koja je važna za prijenos znanja i informacija među organizacijama. Jedan od nedostataka je što je korisnik treba dobro poznavati metodu kako bi je mogao primijeniti u praksi.

IDEF i MIRIS su metodologije koje su zastupljene u području razvoja IS-a i svaka od njih ima svoje prednosti u različitim područjima poslovanja, od modeliranja podataka, izrade softverskog proizvoda ili primjene u praksi.

Metode entiteta i veza (kao dio metodologije MIRIS) i IDEF1X koriste kod modeliranja podataka.



# Literatura

- 2018 Knowledge Based Systems, I. (2018). *Idef.com*. Preuzeto 12. 8 2018 iz <http://www.idef.com/idef4-object-oriented-design-method/>
- Javor Bogati, D. V. (5. 25 2012). Idef metodologija modeliranja informacijskih sustava.
- Knowledge Based Systems, I. (2018). *Idef.com*. Preuzeto 8. 12 2018 iz [http://www.idef.com/idefo-function\\_modeling\\_method/](http://www.idef.com/idefo-function_modeling_method/)
- Knowledge Based Systems, I. (2018). *Idef.com*. Preuzeto 12. 8 2018 iz [http://www.idef.com/idef1-information\\_modeling\\_method/](http://www.idef.com/idef1-information_modeling_method/)
- Knowledge Based Systems, I. (2018). *Idef.com*. Preuzeto 12. 8 2018 iz <http://www.idef.com/idef3-process-description-capture-method/>
- Knowledge Based Systems, I. (2018). *Idef.com*. Preuzeto 12. 8 2018 iz <http://www.idef.com/idef5-ontology-description-capture-method/>
- Knowledge Based Systems, I. (2018.). *Idef.com*. Preuzeto 12. 8 2018 iz <http://www.idef.com/idef1x-data-modeling-method/>
- Pavlic, M. (2009). *Informacijski sustavi*. Rijeka: Odjel za Informatiku Sveučilišta u Rijeci.
- Pavlic, M. (2011). *Oblikovanje baza podataka*. Rijeka.
- Pavlic, M., Jakupović, A., & Čandrlić, S. (2014). *Modeliranje Proces*. Rijeka: Odjel za Informatiku Sveučilišta u Rijeci.
- Ponkve*. (23. 09 2018). Preuzeto 21. 9 2018 iz [ponikve.hr](http://www.ponikve.hr/): <http://www.ponikve.hr/>
- Technology, N. I. (21. December 1993). INTEGRATION DEFINITION FOR INFORMATION MODELING. *Federal Information Processing Standards Publication 184*, 145.
- Wikipedia*. (27. 11 2017). Preuzeto 12. 8 2018 iz <https://en.wikipedia.org/wiki/IDEF1X>
- Wikipedia*. (22. 2 2018). Preuzeto 12. 8 2018 iz <https://en.wikipedia.org/wiki/IDEF>
- Wikipedia*. (23. 09 2018). Preuzeto 1. Kolovoz 2018 iz <https://en.wikipedia.org/wiki/IDEF1X>

## Popis slika

Slika 1 - dijagram dekompozicije.....	11
Slika 2 - Koncepti dijagrama toka podataka (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesu, 2014) .....	13
Slika 3 – Vanjski sustav .....	14
Slika 4 - Tok podataka (Pavlic, Jakupović, & Čandrlić, Modeliranje Procesu, 2014).....	15
Slika 5 - spremište podataka.....	15
Slika 6 - Proces.....	16
Slika 7 - koncepti dijagrama Entiteti – Veze.....	19
Slika 8 - Tip entiteta .....	20
Slika 9 – Atribut (Pavlic, Oblikovanje baza podataka, 2011) .....	21
Slika 10 - Tip veze (Pavlic, Oblikovanje baza podataka, 2011) .....	22
Slika 11 - Tip veze .....	22
Slika 12 - Primjer brojnosti veze (Pavlic, Oblikovanje baza podataka, 2011).....	23
Slika 13 - Primjer slabog tip entiteta (Pavlic, Oblikovanje baza podataka, 2011).....	24
Slika 14 - Primjer egzistencijalnog tipa veze (Pavlic, Oblikovanje baza podataka, 2011).....	25
Slika 15 - Primjer identifikacijskog tipa veze (Pavlic, Oblikovanje baza podataka, 2011).....	25
Slika 16 - Primjer povratne veze (Pavlic, Oblikovanje baza podataka, 2011).....	26
Slika 17 - Primjer agregacije (Pavlic, Oblikovanje baza podataka, 2011).....	27
Slika 18 - Entitet nezavisan o identifikatoru (Technology, 1993) .....	34
Slika 19 – Entitet zavisian o identifikatoru (Technology, 1993) .....	34
Slika 20 – Primjer bazične i upisne domene (Technology, 1993) .....	36
Slika 21 - Sintaksa atributa i primarnih ključeva (Technology, 1993) .....	38
Slika 22 - Opcionalna ne - identifikacijska veza (Technology, 1993).....	40
Slika 23 - Sintaksa kardinalnosti veze (Technology, 1993).....	41
Slika 24 - Veza kategorizacije (Technology, 1993).....	43
Slika 25 - Ne-specifična veza (Technology, 1993).....	45
Slika 26 - asocijativni entitet.....	45
Slika 27 - Primjer sintakse primarnog i alternativnog ključa (Technology, 1993).....	46
Slika 28 - koncepti IDEF1X metode (1) .....	48
Slika 29 - koncepti IDEF1X metode (2) .....	49
Slika 30 - Usporedba koncepata MIRIS-a i IDEF1X-a (1).....	54
Slika 31 - Usporedba koncepata MIRIS-a i IDEF1X-a (2).....	55
Slika 32 - logo firme Ponikve Voda.....	57
Slika 33 - dokument narudžba dobavljaču .....	59
Slika 34 - EV model dokumenta narudžba dobavljaču .....	63
Slika 35 – IDEF1X model dokumenta narudžba dobavljaču.....	64
Slika 36 - dokument povratnica .....	65
Slika 37 - EV model dokumenta povratnica .....	69
Slika 38 – IDEF1X model dokumenta povratnica .....	70

## Popis tablica

Tablica 1 - Popis metoda IDEF metodologije (Pavlic, Oblikovanje baza podataka, 2011).....	28
Tablica 2 - analiza narudžbe dobavljaču .....	60
Tablica 3 – analiza dokumenta povratnica .....	66