



KONCEPTUALNO MODELIRANJE PODATAKA

Autor: **Zoran Hercigonja, mag.edu.inf.**

ISBN 978-953-59549-5-8

Impressum

Naslov: **Konceptualno modeliranje podataka**

Autor: **Zoran Hercigonja, mag.edu.inf.**

Lektor: **Adela Brozd**

Nakladnik: **Vlastita naklada autora**

Url: <https://issuu.com>

Mjesto i godina izdavanja: **Varaždin, 2017.**

ISBN 978-953-59549-5-8

Sadržaj:

1.Uvod.....	4
2.Konceptualno modeliranje podataka.....	5
2.1. Model entiteti-veze	6
2.2.Grafički prikaz modela entiteti-veze	14
2.3.Izrada modela entiteti-veze	17
2.3.1.Prikupljanje i analiza informacijskih zahtjeva:	17
2.3.2.Izraditi model entiteti-veze	17
2.3.3.Konsolidiranje modela entiteti veze.....	19
2.4. Objektni modeli.....	20
2.5. Izrada objektnog modela.....	24
2.5.1.Prikupljanje i analiza informacijskih zahtjeva:	24
2.5.2.Izraditi objektni model	25
2.5.3.Konsolidiranje modela entiteti veze.....	26
3.Zaključak.....	27
4.Literatura.....	28

1.Uvod

Dobro osmišljen i izrađen konceptualni model veza i entiteta presuđuje u stvaranju uspješne baze podataka. Najvažnije je na samome početku projektirati i konceptualno dotjerati model neke baze podataka, kako bi ona kasnije bila upotrebljiva i korisna. Nije svejedno isprogramirati vezu entiteta koja neće odgovarati na upit baze podataka. Preduvjet dobro modelirane baze s minimumom pogrešaka i maksimalnim učinkom je prethodno kreiranje modela logičkih veza. Konceptualno modeliranje je vrlo često zanemareno u oblikovanju baze podataka, ali najviše doprinosi. Upravo zato tema ovog rada je prikaz i uvod u konceptualni model baze podataka definirajući pritom najbitnije elemente konceptualnog modela: entitet, veza, atribut.

2. Konceptualno modeliranje podataka

Konceptualno modeliranje polazi od specifikacija informacijskih zahtjeva, koji čine zahtjeve na strukturu podataka i zahtjeve za korištenje podataka, a rezultira izrađenim konceptualnim modelom podataka.¹

Prema tome konceptualni model podataka predstavlja cjelovit i konzistentan opis podataka informacijskog sustava. Konceptualni model podataka je potreban korisnicima koji dobro razumiju potrebe za informacijama, a izrađeni su od strane analitičara odnosno projekatana.

Konceptualno općenito znači nešto što nije ovisno o implementaciji dok na razini sustava predstavlja neovisnost načina interakcije korisnika s informacijskim sustavom, načina realizacije procesa informacijskog sustava i načina implementacije modela podataka.

U dobrom konceptualnom modelu podataka, podaci koji opisuju jednu činjenicu, grupirani su na jedno mjesto i neovisni su od podataka druge činjenice.

Na taj način:

- svaka konstrukcija u modelu podataka ima samo jedno značenje, koje se čita iz samo jedne konstrukcije
- izrada modela je jednostavnija jer se projektant u postupku modeliranja koncentrira na dijelove modela i donosi međusobno neovisne odluke
- promjene u modelu podataka lakše je lokalizirati, a mijenjaju se samo konstrukcije zahvaćene promjenom
- model podataka može se graditi po blokovima kao kocke

U daljnjem kontekstu razumijevanja konceptualnog modeliranja podataka, treba sagledati i opisati dva modela koja objašnjavaju i prikazuju koncept modeliranja ovakve vrste.

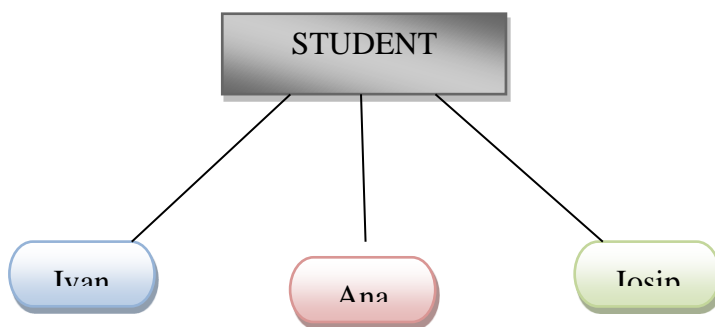
¹ Mladen Varga Baze podataka, Konceptualno, logičko i fizičko modeliranje podataka DRIP Zagreb, 1994. str.43

2.1. Model entiteti-veze:

Model entiteti-veze predstavlja kao što i sam naziv govori entitete i njihove veze. Preteča ovakvog modela jesu tehnike za prikaz strukture podataka, koje su kasnije proširene novim koncepcijama i kasnije prozvan proširenim modelom entiteti-veze. Pri objašnjavanju ovakvog modela poslužiti ćemo se s dva grafička prikaza Martinovi dijagrami i Chenovi dijagrami, koje ćemo zasebno prikazati, a koji će doprinjeti boljem razumijevanju modela entiteti-veze.

Model entiteti-veze promatra svijet kroz entitete i njihov odnos tj. veze. U promatranju svijeta koristi se nužno postupak **apstrakcije**. Apstrakcija predstavlja uočavanje glavnog, općeg, nužnog ili bitnog, a hotimično ispuštanje sporednog, posebnog, slučajnog ili nebitnog.² Da bismo što bolje sagledali svijet i odagnali nepotrebne sitnice u cilju što kvalitetnijeg modeliranja, nužna je pojava apstrakcije. Također za postupak modeliranja podataka, tj. uočavanje korisnog, a odbacivanje beskorisnog, koristimo i postupke klasifikacije, generalizacije i agregacije, koji će biti u daljnjim stranicama prikazani i objašnjeni.

Klasifikacija je postupak (rezultat) primjene koncepta na neki objekt³. Predstavlja jednu vrstu apstrakcije u kojoj se entiteti opisuju i grupiraju u klase odnosno grupe i razrede prema zajedničkim obilježjima. Na primjer osobe Ivan, Ana i Josip se mogu klasificirati u zajednički entitet na primjer student. Dakle oni imaju zajednička obilježja kao što su JMBAG, ime, prezime, godina studija, a međusobno se razlikuju po vrijednostima tih obilježja.

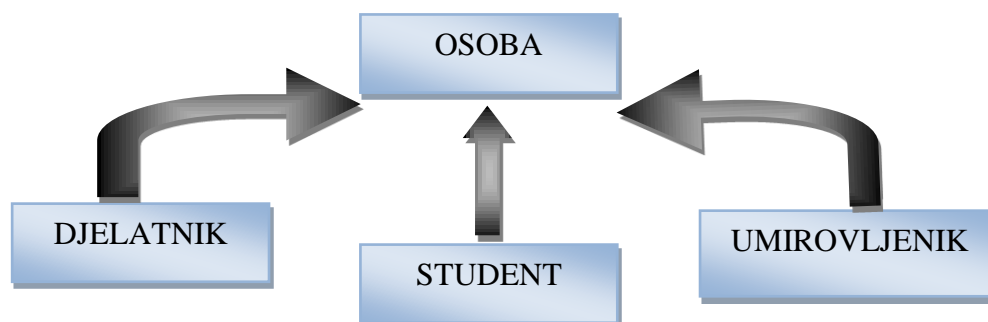


sl.1.Klasifikacija

² Mladen Varga Baze podataka, Konceptualno, logičko i fizičko modeliranje podataka DRIP Zagreb, 1994. str. 44.

³ Prof. dr. sc. Mirko Maleković, predavanja Baze podataka 2 slajd 53.

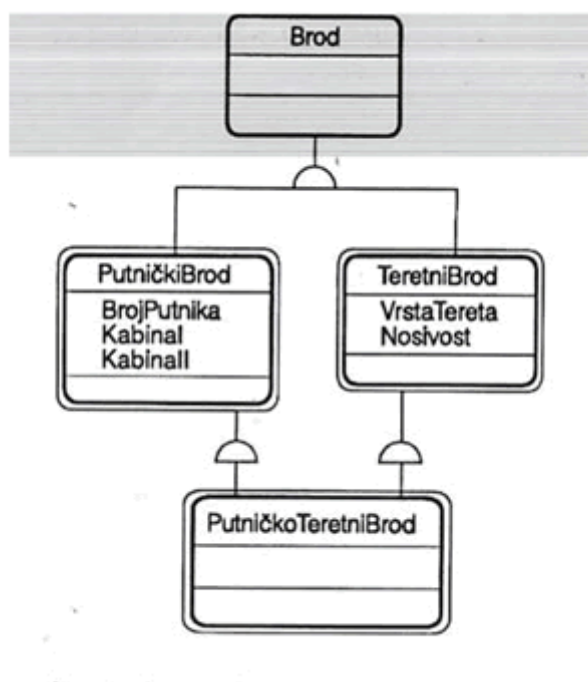
Generalizacija je vrsta apstrakcije slična klasifikaciji. Kod nje se tipovi entiteta niže razine uopćuju s tipom entiteta više razine. Generalizacija je postupak (rezultat) uočavanja koncepta (tipa) koji uključuje drugi koncept (koncepte)⁴. To bi značilo na primjer tipovi entiteta kao djelatnik, student i umirovljenik generaliziraju se tipom entiteta osoba. Tip entiteta više razine je nadtip, a tip entiteta niže razine je podtip.



sl.2.Generalizacija

Generalizacija se opisuje vezim "je" (engl. is a). Obrnuti postupak naziva se specijalizacijom.

Da bih potkrijepio ovaj iskaz prilažem odgovarajući primjer.

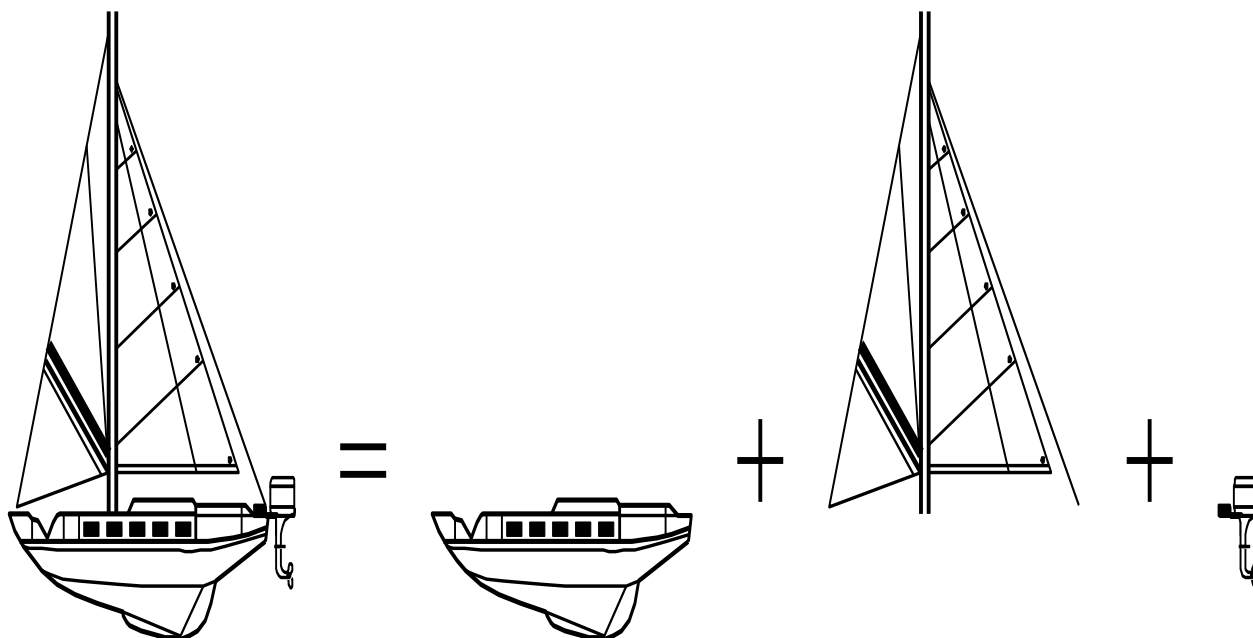


sl.3.Generalizacija/specijalizacija

⁴ Prof. dr. sc. Mirko Maleković, predavanja Baze podataka 2 slajd 66

Ukoliko sliku gledamo odozdo prema gore onda dobivamo ideju generalizacije, a ukoliko je gledamo odozgora prema dole onda dobivamo specijalizaciju.

Agregacija jest formiranje novog pojma, višeg stupnja na temelju odnosa postojećih pojmova. Agregacija je postupak ili rezultat formiranja cjeline (složenog objekta) koristeći druge objekte kao dijelove.⁵



sl.4.Agregacija⁶

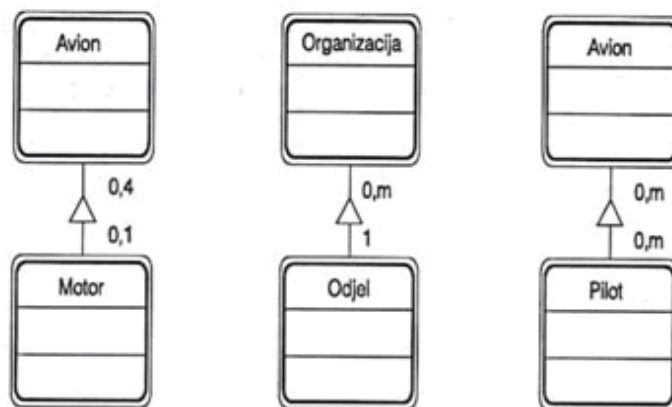
U ovom jednostavnom primjeru, koristeći jednostavne dijelove, stvorena je cjelina. Ova jahta je izgrađena od jedra, trupa i pokretačkog motora. Agregacija specificira da neki objekt može biti komponiran ili da može biti dio drugog objekta.⁷ U ovom primjeru, jedro je dio nekog drugog objekta na primjer jahte ukoliko jahtu promatramo kao objekt. Isti slučaj je i s motorom jahte.

⁵ Prof. dr. sc. Mirko Maleković, predavanja Baze podataka 2slajd 60

⁶ Prof. dr. sc. Mirko Maleković, predavanja Baze podataka 2slajd 60

⁷ Prof. dr. sc. Mirko Maleković, predavanja Baze podataka 2slajd 63

Agregacija opisuje objekt (obično se crta gore) i njegove dijelove (crtaju se dolje). Povezuju se linijom na kojoj simbol ukazuje na odnos objekata. Uz linije se označava broj koji označava broj dijelova u cjelini. Motor je dio ni jednog aviona ili najviše jednog aviona, avion ima u svom sastavu niti jedan ili najviše 4 motora. Odjel pripada točno jednoj organizaciji, a organizacija ima nijedan ili više odjela. Pilot upravlja nijednim ili s više aviona, a avion je upravlján od nijednog ili više pilota.



sl.5. Agregacija (Primjer avion-motor, organizacija-odjel, avion-pilot)

Kako bi bilo lakše razumijeti ove metode temeljene na apstrakciji, valja objasniti i osnovne pojmove kao što su entitet, veza, atribut i ključ.

Pojam entitet ima velik broj definicija koje se mogu redom ovdje navesti.

Entitet je nešto što postoji i što se u stvarnom svijetu može identificirati.⁸

Entitet je stvaran ili apstraktan predmet ili događaj o kojem se u informacijskom sustavu prikupljaju podaci.

Entities are the objects of significance for the organization about which information needs to be known⁹.

⁸ Mladen Varga Baze podataka, Konceptualno, logičko i fizičko modeliranje podataka DRIP Zagreb, 1994. str.45.

⁹ R. A. Mata-Toledo, P. K. Cushman: Fundamentals of Relational Databases, Schaum's Outline Series, McGraw-Hill, 2000.

Entitet je objekt opisa odnosno dio realnog sustava koji želimo opisati ili pratiti. Svi entiteti imaju svoja obilježja odnosno attribute i na taj način ih je lakše zapaziti u postupku primjene metode klasifikacije. Skup srodnih entiteta se naziva tip entiteta.

Primjeri tipova entiteta:

- ljudi (njihove uloge):

OSOBA

DJELATNIK

KUPAC

- organizacije (organizirani skup skup ljudi)

PODUZEĆE

AGENCIJA

- stvari (materijalni predmeti)

PROIZVOD

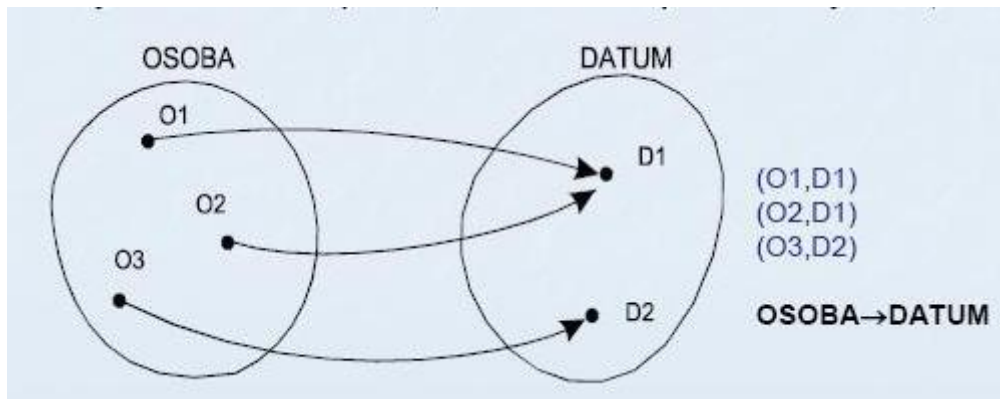
REZERVNI DIO

AUTOMOBIL

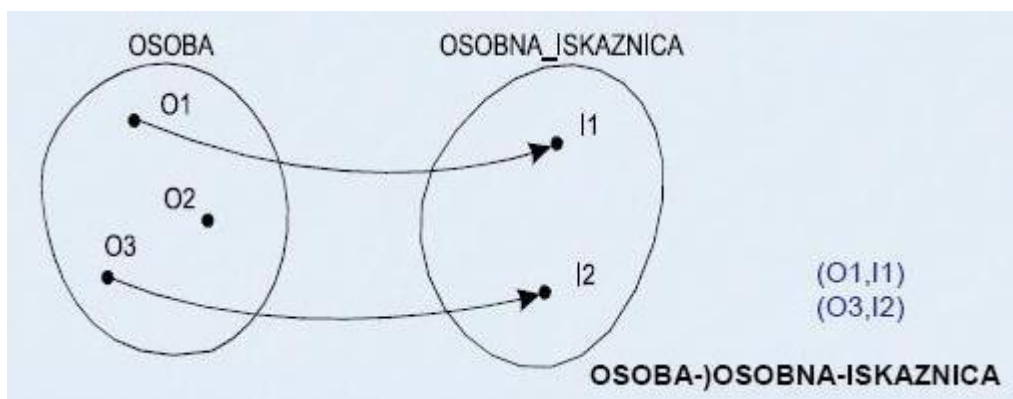
Veza je međusoban skup entiteta odnosno ostvareni odnos između dva ili više entiteta. Može se objasniti matematičkim pojmom pridruživanja elemenata jednog skupa elemenata drugom skupu elemenata. Postoje tri načina združivanja jednoznačno, uvjetno i višeznačno pridruživanje.

Jednoznačno pridruživanje:

Pridruživanje je jednoznačno ($A \rightarrow B$) ako je svakom članu skupa A pridružen jedan i samo jedan član skupa B. Primjer je $OSOBA \rightarrow DATUM$. Drugi naziv ovakvog pridruživanja je jedan prema jedan ili engleski One to one relationship. To potvrđuje i tvrdnja The one to one relationship has the cardinality or degree of one and only one in both directions.¹⁰

sl.6. Jednoznačno pridruživanje¹¹**Uvjetno pridruživanje:**

Pridruživanje je uvjetno ($A \rightarrow B$) ako je svakom članu skupa A pridružen jedan ili nijedan član skupa B. Primjer je $OSOBA \rightarrow OSOBNA\ ISKAZNICA$.

sl.7. Uvjetno pridruživanje¹²

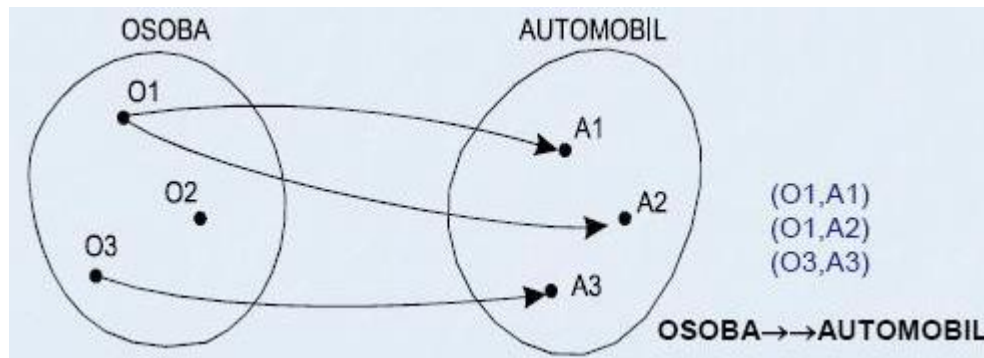
¹⁰ R. A. Mata-Toledo, P. K. Cushman: Fundamentals of Relational Databases, Schaum's Outline Series, McGraw-Hill, 2000. str. 229

¹¹ http://upravljanjepodacima.wikia.com/wiki/KONCEPTUALNO_MODELIRANJE_PODATAKA

¹² http://upravljanjepodacima.wikia.com/wiki/KONCEPTUALNO_MODELIRANJE_PODATAKA

Višeznačno pridruživanje:

Pridruživanje je višeznačno ($A \rightarrow\rightarrow B$) ako je svakom članu skupa A pridružen nijedan, jedan ili više članova skupa B. Primjer je OSOBA $\rightarrow\rightarrow$ AUTOMOBIL. To isto potvrđuje tvrdnja the many to one relationship has a cardinality in one direction of one or more and in the other directions of one and only one.¹³

sl.8. Višeznačno pridruživanje¹⁴

Osim prepoznatljivog binarnih veza modela entiteti-veze postoji i alternativna veza koja se temelji na:

- identifikacijska/egzistencijalna zavisnost
- specijalizacija/generalizacija
- agregacija entiteta u novi entitet

Identifikacijska zavisnost je vezana uz slabije entitete. To znači da je entitet jak ukoliko postoji ili egzistira samostalno. Entitet je slab ako ne postoji samostalno ili se ne identificira samostalno. Ako postojanje entiteta Y zavisi o postojanju entiteta X, tada je entitet Y egzistencijalno slab i egzistencijalno ovisan o entitetu X. Dakle ako se entitet ne identificira samostalno, on je identifikacijski slab. Primjer može biti razred. Razred je egzistencijalno slab jer je ovisan o drugom entitetu a to je škola. Identifikacijska zavisnost se često koristi pri modeliranju atributa koji imaju više vrijednosti. U modelu entiteti-veze preporučljivo je da atribut ima samo jednu vrijednost pa se recimo problem atributa s više vrijednosti modelira otvaranjem novog slabog entiteta s atributom koji ima samo jednu vrijednost. Identifikacijska zavisnost povlači egzistencijalnu zavisnost, ali obrnuti postupak nije ostvariv i moguć.

¹³ R. A. Mata-Toledo, P. K. Cushman: Fundamentals of Relational Databases, Schaum's Outline Series, McGraw-Hill, 2000. str. 229

¹⁴ http://upravljanjepodacima.wikia.com/wiki/KONCEPTUALNO_MODELIRANJE_PODATAKA

Specijalizacijom/generalizacijom naziva se veza koja opisuje odnos entiteta i njegovih podtipova, dobivenih postupcima generalizacije odnosno specijalizacije. Generalizacija/specijalizacija je relacija između koncepata (tipova) koja vodi na subtipove i supertipove (podtipovi, nadtipovi).¹⁵

Ako se svaka pojava nadtipa specijalizira u samo jedan od entiteta podtipova, govori se o **ekskluzivnoj specijalizaciji** odnosno generalizacijskoj hijerarhiji. Ako se svaka pojava entiteta nadtipa istovremeno specijalizira u više entiteta podtipova, govori se o **neekskluzivnoj specijalizaciji** odnosno podskupovnoj hijerarhiji.

Agregacijom se naziva veza u kojoj sudjeluje tri ili više entiteta ili veza s opisnim atributima ili veza koja se ponaša kao entitet, jer sudjeluje u vezi s drugim entitetima.¹⁶

Nakon ovih opisanih metoda koje se temelje na metodi apstrakcije te nekih važnih pojmova, valjalo bi prijeći i na pojmove kao što su atribut i ključ koji će nam biti potrebni za identifikaciju i izradu modela entiteti-veze.

Atribut

Svaki tip entiteta je opisan skupom atributa. Atribut možemo definirati i kao određen odabir svojstava s kojima želimo opisati entitet. Već prije spomenuti primjer klasifikacije je sadržavao neke attribute kao što su JMBAG, ime, prezime i godina studija. Svaki atribut može poprimiti jednu vrijednost iz domene vrijednosti atributa pri čemu domena predstavlja domena je kolekcija objekata u selekcioniranom interesnom području¹⁷. Dakle domena definira moguće vrijednosti atributa. Formalno gledajući, atribut je funkcija koja skupu entiteta pridružuje skup vrijednosti odgovarajuće domene. Normalno vrijednost atributa se s vremenom može mijenjati. Veza može, ali ne mora biti opisana atributima.

¹⁵ Prof. dr. sc. Mirko Maleković, predavanja Upravljanje znanjem prezentacija P08SMkm slajd 27

¹⁶ Mladen Varga Baze podataka, Konceptualno, logičko i fizičko modeliranje podataka DRIP Zagreb, 1994. str 52

¹⁷ Prof. dr. sc. Mirko Maleković, predavanja Upravljanje znanjem prezentacija P08SMkm slajd 5

Ključ

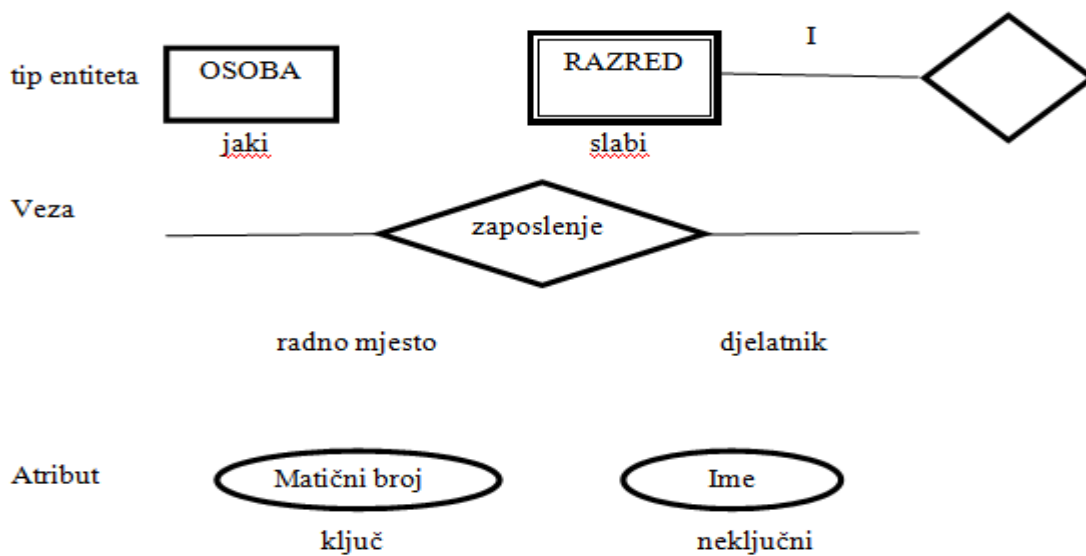
Da bismo identificirali pojavu entiteta, koristimo koncept ključa. Same pojave entiteta se međusobno razlikuju po vrijednostima svojih atributa pa je dovoljno pronaći određen skup atributa čije bi vrijednosti omogućile jednoznačnu identifikaciju svake pojave entiteta u skupu entiteta. To bi na primer u današnjem svijetu bio OIB broj prema kojem se mogu identificirati osobe u velikom skupu osoba. Uvijek se nastoji pronaći minimalan ključ, čiji podskup atributa više nema karakteristike ključa. Minimalnih ključeva može biti više. Njih nazivamo **moogućim ključevima**. Na primjer matični broj građana i broj osobne iskaznice dva su moguća ključa za identifikaciju. Jedan izabrani mogući ključ postaje **primarnim ključem**. Postoji problem na primjer kod slabih entiteta. Slabi entiteti nemaju dovoljno atributa za jednoznačnu identifikaciju svake pojave entiteta pa se može formirati primarni ključ.

2.2. Grafički prikaz modela entiteti-veze:

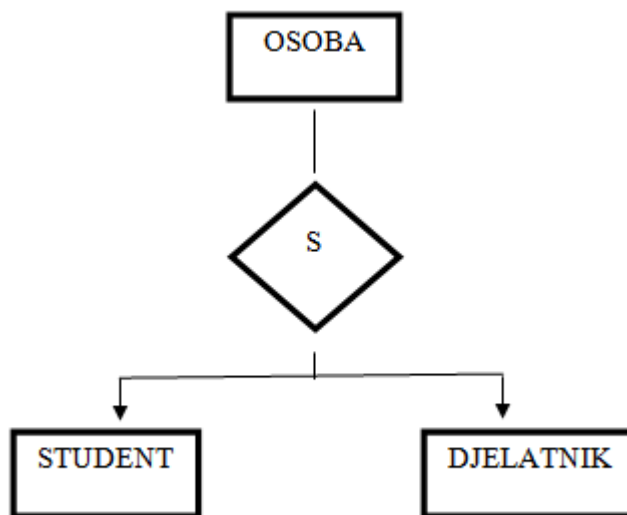
Nacrtni dijagram modela entiteti-veze znatno olakšava razumijevanje modela podataka. Potpuni grafički prikaz kod složenijih modela s mnogo atributa je nepregledan pa se češće koristi jednostavan grafički prikaz poput Chenovog dijagrama i Martinonov dijagrama.

CHENOV PRIKAZ

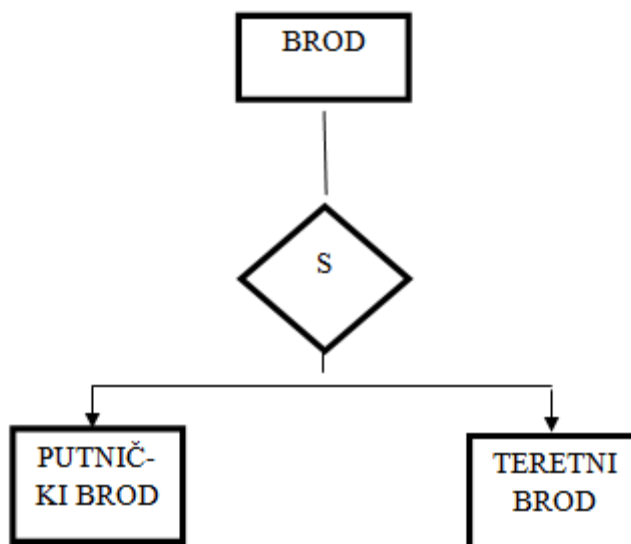
Kod Chenovog dijagrama slabi entiteti se grafički različito prikazuju kao što je prikazano slikom sl.9. Chenov prikaz. Identifikacijski slab entitet se označava oznakom I uz vezu koja ga povezuje s jakim entitetom. Egzistencijalno slab entitet može se prikazati oznakom E. Na grafičkom prikazu se obično navode nazivi veza, a ispuštaju uloge entiteta u vezi. Naziv veze stavlja se u znak veze (romb), a uloga entiteta upisuje se uz liniju veze u kojoj entitet sudjeluje.



sl.9.Chenov prikaz



sl.9.a.Chenov prikaz (neekskluzivna specijalizacija)



sl.9.b. Chenov prikaz (ekskluzivna specijalizacija)

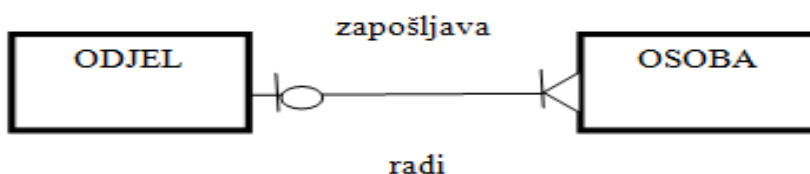
MARTINOV PRIKAZ

U Martinovom prikazu entiteti se prikazuju isto kao i kod Chenovog prikaza. Slabi entiteti se posebno ne razlikuju, ali se koriste samo binarne veze. One se crtaju običnom linijom koja povezuje dva entiteta. Veza se ne imenuje, nego se navode uloge entiteta koji u njoj sudjeluju. Tekst iznad linije u vodoravno nacrtanoj vezi, imenuje ulogu entiteta na lijevoj strani, a ispod linije ulogu entiteta na desnoj strani. Tekst s lijeve strane linije u vezi nacrtanoj okomito, imenuje ulogu donjeg entiteta, a s desne strane linije ulogu gornjeg entiteta. Donja i gornja granica sudjelovanja entiteta u vezi označena je oznakom na liniji veze:

0 (kružić)-donja granica je 0

| (crtica)-donja granica je 1

> (kao znak veće)-gornja granica M (više)



sl.10.Martinov prikaz

U Martinovom prikazu ne postoje veze višeg stupnja. Veze ne mogu imati attribute. Umjesto veze s atributom, otvara se vezni entitet s atributima veze, a on sam se povezuje s entitetima u vezi.

2.3. Izrada modela entiteti-veze:

Model entiteti-veze prikazuje entitete i njihovu strukturu, a za prikaz pravila obavljanja i pokretanja procesa, koriste se tehnike za opis procesa i događaja. To znači da treba prikupiti sve potrebne informacijske zahtjeve, izraditi model entiteti-veze te naposljetku konsolidirati model entiteti-veze. Pritom treba svratiti pozornost da je postupak izraditi model entiteti-veze složen jer se izrada odnosi i na entitete, veze, ključeve i attribute. Sam postupak izrade modela entiteti-veze nije strogo propisan, ali postoji klasična forma po kojoj se sprovodi i nju ćemo opisati u narednim stranicama seminarskog rada.

2.3.1. Prikupljanje i analiza informacijskih zahtjeva:

Modeliranje podataka počinje prikupljanjem informacijskih zahtjeva odnosno poslovnih pravila, koje se obavlja intervjuiranjem korisnika, istraživanjem dokumentacije poslovnog sustava i sličnim postupcima. Nakon intervjuiranja i proučavanja dokumentacije poslovnog sustava analitičar priprema inicijalni poslovni model, koji se sastoji od modela podataka i modela procesa. Analitičar ih razvija istovremeno.

2.3.2. Izraditi model entiteti-veze:

Utvrđivanje entiteta:

Za postupak utvrđivanja entiteta, potrebno je poduzeti pojedine korake.

- **Utvrđiti entitete** intervjuirajući korisnike ili koristeći se dokumentacijom poslovnog sustava. Dakle entiteti su uvijek imenice kojima se opisuju objekti važni za poslovni sustav.
- **Imenovati entitete:** Za imenovanje entiteta je najbolje uzeti imenicu u jednini kojom se služe korisnici. Naziv entiteta mora biti jedinstven, što implicira da se pojava homonima treba preimenovati.
- **Opisati entitete:** Dobar opis sastoji se od punih rečenica koje su smještene u katalogu informacijskog sustava. Također treba dokumentirati i očite činjenice o entitetima, jer one ne moraju biti očite svim korisnicima. Opisom entiteta utvrđujemo najvažnije attribute entiteta.
- **Utvrđiti specijalizaciju entiteta:** Odnosi se na klasičnu definiciju specijalizacije i određivanja podtipova i nadtipova entiteta kako bi bile shvatljive veze među entitetima.

- **Utvrđiti zavisnost entiteta:** Slab odnosno zavisan entitet ne može postojati ako nije vezan uz jaki entitet. Identifikacijski slab entitet uvijek ima složen primarni ključ; ključ jakog entiteta

Utvrđivanje veza

- **Utvrđiti veze** intervjuirajući korisnike ili koristeći se dokumentacijom poslovnog sustava.
- **Imenovati veze** navodeći naziv veze uz pomoć Chenovog ili Martinovog prikaza. Za naziv veze najprikladnija je glagolska imenica, ali se mogu koristiti i glagoli.
- **Opisati veze:** vrlo je slično opisu entiteta
- **Izbaciti redundantne veze:** Redundantna veza može se izvesti iz drugih veza pa je stoga nepotrebna.
- **Utvrđiti veze višeg stupnja:** Većina veza koje utvrđujemo će biti binarne veze. Povremeno se javlja potreba za uspostavljanjem veza višeg stupnja npr. 3, 4, itd. Takve veze treba pažljivo proučiti i utvrđiti.

Utvrđivanje ključeva entiteta

- **Utvrđiti ključeve jakih entiteta:** Kod njih se relativno lako određuje primarni ključ, najčešće je to jedan identifikacijski atribut
- **Utvrđiti ključeve slabih entiteta:** Oni nasljeđuju ključ jakog entiteta i dobivaju najčešće jedan diskriminator
- **Izbjegavati sastavljene ključeve:** Kod ključa koji je sastavljen od više atributa, treba provjeriti postoje li entiteti s ključevima koji odgovaraju komponentama sastavljenog ključa. Ako takvi entiteti postoje, entitet sa sastavljenim ključem nadomješta se vezom

Utvrđivanje atributa entiteta

Kao što je bilo prethodno navedeno, atributi opisuju entitete ili veze, a utvrđuju se na sljedeći način:

- **Utvrđiti attribute** intervenirajući korisnike ili koristeći se dokumentacijom poslovnog sustav.
- **Imenovati attribute:** Naziv atributa može se sastojati od tri djela koji imaju različitu namijenu; korijena, kvalifikatora i klase atributa.
- **Opisati attribute:** Koristiti ista pravila kao i za opis entiteta uz uvjet da dodatno treba definirati domenu vrijednosti koju može poprimiti atribut.

- **Utvrđiti kardinalnost atributa** odgovarajući na sljedeća pitanja na primjer „koliko datuma zaposlenja mora imati osoba“ i „koliko datuma zaposlenja može imati osoba“
- **Verificirati pripadnost atributa:** Iz opisa poslovnih pravila, vidi se da se pripadnost atributa entitetu ili vezi iskazuje na više načina: npr. visina učenika ili učenička visina
- **Grupirati povezane attribute:** Mnogo puta se mogu pronaći nekoliko međusobno povezanih atributa koji čine zasebnu cjelinu
- **Izbaciti derivirane attribute:** Derivirani atributi su oni atributi koji se mogu izvesti iz drugih atributa. Oni najčešće nisu redundantni u odnosu na jednostavne attribute i izbacuju se iz konceptualnog modela.

2.3.3. Konsolidiranje modela entiteti veze:

Nakon prethodno završenih koraka, potrebno je napraviti i posljednji rez uz pomoć konsolidiranja. Model entiteti-veze uobičajeno se radi u manjim segmentima informacijskog sustava tj. po problemskim područjima odnosno podsustavima informacijskog sustava. Konačni je cilj načiniti jedinstveni model podataka informacijskog sustava. Jedinstven model podataka se sastoji od konzistentno uklopljenih podmodela podataka pojedinih problemskih područja. Zbog toga treba pojedine podmodele, nakon što su načinjeni, integrirati u model podataka informacijskog sustava. Integriranje modela podataka znači izradu jedinstvenog modela podataka od više podmodela podataka.¹⁸ Integriranje se dobiva doradom pojedinih podmodela koji u početnom stadiju nisu konzistentni. Cilj konsolidacije je dobivanje skupa međusobno konzistentnih podmodela koji zajednički čine jedinstven model podataka. Jedinstveni model podataka treba zadovoljavati tri uvjeta:

- kompletnost
- neredundantnost
- konzistentnost

U postupku integriranja podmodela u jedinstveni model podataka, rješavaju se sve nekonzistentnosti među podmodelima kao što mogu biti sinonimi ili homonimi entiteta, različito opisane istovrsne veze i slično.

¹⁸ Mladen Varga Baze podataka, Konceptualno, logičko i fizičko modeliranje podataka DRIP Zagreb, 1994. str. 71

2.4. Objektni modeli:

Uz klasičnu primjenu modela entiteti-veze, koristi se i objektni model da bismo postigli konceptualno modeliranje podataka. Kao i kod modela entiteti-veze, prikazat ću osnovne pojmove vezane uz objektnu modelu radi boljeg razumijevanja te ću prikazati način izrade objektnog modela koji (kao što će biti prikazano u narednim stranicama seminara) se izrađuje na jednak način kao i modeli entiteti-veze s malim razlikama.

Dakle zašto se uopće koriste objektni modeli, ako već postoje potonje opisani i savršeni modeli entiteti-veze koji mogu omogućiti modeliranje? „...*Model entiteti-veze, uspješno se primjenjuje u poslovnim informacijskim sustavima u kojima su podaci razmjerno jednostavnijih tipova. Međutim pojavom nestandardnih aplikacija, kao što su uredsko poslovanje, projektiranje, ekspertni sustavi, baze znanja, javlja se potreba za „bogatijim“ tipovima podataka, kojih u ovom (modelu entiteti-veze) nema...*“¹⁹ U objektnom modelu podaci se definiraju kroz objekte, a sam model implementira se kroz objektnu bazu podataka. Object technology is an important discipline in the field of software engineering in general.²⁰ Osnovni pojmovi za razumijevanje objektnog modela su objekt, klasa, struktura, atribut, subjekt i servis.

Objekt predstavlja osnovni pojam objektnih modela; element ili cjelina koja se može samostalno promatrati. Objekt je apstrakcija nečeg u problemskoj domeni, o čemu se prikupljaju podaci i što skriva vrijednost svojih atributa i svojeg ponašanja.²¹ Karakteristike objekta slične su karakteristikama entiteta. Svaki objekt ima više atributa koji ga karakteriziraju. Također objekti imaju prepoznatljivo ponašanje; opisuju se operacijama koje se obavljaju nad njima i stanjem odnosno vrijednostima. Ako usporedimo objekt i entitet, vidjet ćemo razliku. Dakle entiteti nemaju opis ponašanja u samom modelu entiteti-veze. Objekt općenito odgovara pojavi entiteta modela entiteti-veze, osim što sadržava kao dodatak opis svog ponašanja. Do sada smo opisivali jednostavne objekte, no valja napomenuti i pojavu kompleksnih objekata te njihovih daljnjih podjela. Dakle kompleksni objekti postoje kao glavna motivacija za razvoj objektno-orijentiranih sustava. Stoga kompleksne objekte možemo podijeliti na strukturirane i nestrukturirane.

Strukturirani objekti se sastoje od komponenti i definirani su rekurzivnom primjenom konstruktora tipova na različitim razinama. Naravno definirani su strukturom koja rabi više konstruktore tipova podržanih sustavom baze podataka. Struktura objekta, definirana i podržana od strane sustava.

¹⁹ Mladen Varga Baze podataka, Konceptualno, logičko i fizičko modeliranje podataka DRIP Zagreb, 1994. str. 79

²⁰ C. J. Date: An Introduction to Database Systems, Addison Wesley, 2007., str 811

²¹ Mladen Varga Baze podataka, Konceptualno, logičko i fizičko modeliranje podataka DRIP Zagreb, 1994. str. 80

Između kompleksnog objekta i njegovih komponenti postoje dva tipa **referenciranja** na svakoj razini:

- *semantika vlasništva* - primjenjuje se kada su komponente kompleksnog objekta učahurene unutar kompleksne strukture i tako čine sastavni dio objekta
- *semantika referenciranja* - kada su komponente kompleksnog objekta nezavisni objekti, ali ih mogu referencirati drugi objekti

Nestrukturirano kompleksni objekti predstavljaju tip podataka koji zahtijeva znatne memorijske resurse, kao što su slike ili skenirane kartografsko-topografske karte na primjer. U bazi podataka reprezentirani su kao veliki binarni objekti odnosno kao veliki znakovni objekti. Sustav baze podataka ne poznaje njihovu internu strukturu i samo aplikacija koja se njima koristi može interpretirati njihovo značenje.

Naposljetku objekti se smatraju kompleksnima jer zahtijevaju veliki memorijski prostor.

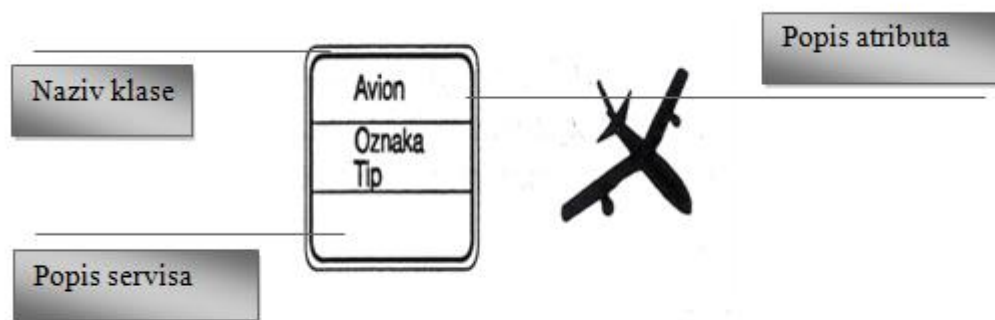
Kompleksne objekte možemo konfigurirati na sljedeće načine tako da:

- kolekcija koja sadrži jednu verziju za svaki objekt, je uređena na način da su verzije objekata kompatibilne i da zajedno tvore ispravnu verziju kompleksnog objekta
- kolekcija verzija različitih objekata koji zajedno tvore kompleksan objekt
- graf verzija opisuje verzije istoga objekta

Kao zaključak o konfiguracijama iznosimo: konfiguracija treba slijediti strukturu tipa kompleksnoga objekta; višestruke konfiguracije istoga kompleksnoga objekta odgovaraju višestrukim verzijama komponentnih objekata.

Klasa je opis jednog ili više objekata koji imaju isti skup atributa i jednak opis ponašanja.²²Klasa odgovara tipu entiteta, osim što dodatno sadržava sadržava opis ponašanja objekata klase.

²² Mladen Varga Baze podataka, Konceptualno, logičko i fizičko modeliranje podataka DRIP Zagreb, 1994. str. 80

sl.11.Primjer klase i objekta²³

Slika predstavlja kombinaciju klase i objekta. Dakle klasa je u ovom primjeru prikazana debelim obrubom pravokutnika, a njezini objekti tankim obrubima pravokutnika. Slika prikazuje realno način na koji se može svoditi združivanje između klasa, objekata ili dvaju objekata. Simbol se sastoji od tri dijela: u prvi dio se upisuje naziv klase, u drugi (srednji) popis atributa, a u donji (zadnji) popis servisa.

Struktura

Kod struktura se koriste isti načini apstrakcije kao i kod modela entiteti-veze. To su generalizacija, agregacija i klasifikacija. Njih sam opisao u prethodnim stranicama seminara pa smatram da ih nije potrebno ponovno navoditi. Primjena te tri strukture na objekte, jednaka je primjeni tri strukture na entitete. Jedina razlika je što u modelu entiteti-veze govorimo o entitetima, a u objektnom modelu o objektima. Nakon objekta, klase i struktura dolazi i pojam atributa koji je istovrsan pojmu atributa kod modela entitet-veze. Vrijednosti atributa su promjenjive servisima definiranim nad objektom. Kao i kod modela entiteti-veze preporučljivo je da svaki atribut ima samo jednu vrijednost. Na taj način se dobivaju jednostavniji modeli. Atributi koji sadržavaju više vrijednosti, eliminiraju se iz modela uvođenjem novog objekta. Naravno u situacijama u kojima vrijednost atributa nije promjenjiva odnosno ne postoji, objektni način modeliranja preporuča da se revidira struktura generalizacije/specijalizacije. Svaki objekt mora imati mogućnost identifikacije u skupu objekta iste klase. Za to koristimo identificirajući attribute ili implicitne identifikatore. Ovisno o poimanju atributa, neki objektni modeli mogu razlikovati jednostavne i kompleksne objekte.

²³ Mladen Varga Baze podataka, Konceptualno, logičko i fizičko modeliranje podataka DRIP Zagreb, 1994. str. 81

Jednostavni objekt je opisan jednostavnim atributima. Može se koristiti za izgradnju bogatijih objekata za manipulaciju na većem stupnju apstrakcije.²⁴ **Kompleksan** objekt je objekt čiji atributi mogu biti i samo objekti. Najbolji primjer za to jest:

Npr. Automobil je kompleksan objekt koji se sastoji od objekata motor i karoserija, a motor može biti kompleksan ili jednostavan ovisno o tome je li opisan sebi podređenim objektima ili samo jednostavnim atributima²⁵.

Servis

Prije nego počinjem objašnjavati što su to servisi, potrebno je napomenuti da u modelu entiteti-veze ne postoje servisi, dok kod objektnog modela ti servisi olakšavaju modeliranje. Servis opisuje ponašanje objekta jedne klase. To znači da svaki informacijski sustav ima podatke i procese koji mijenjaju podatke. U objektnim modelima servisi su ti koji opisuju procese nad podacima objekata. Podaci opisuju stanja objekata i reprezentiraju attribute, a servisi onda predstavljaju sredstvo za promjenu njihovih stanja.

Potrebni servisi, definiraju se utvrđivanjem stanja objekata, utvrđivanjem potrebnih servisa, utvrđivanjem poruka za pokretanje servisa i razradom detalja servisa.

- **Utvrđiti stanja objekata:** svaki objekt prolazi kroz različita stanja od trenutka nastanka do trenutka nestanka (životni vijek objekta). Stanje svakog objekta reprezentira se vrijednostima njegovih atributa odnosno podataka koji opisuju stanje objekata. Dakle svaka promjena atributa znači promjena stanja objekata. Uočiti stanje objekata znači prepoznati attribute i njihove vrijednosti koje utječu na promjenu ponašanja objekta.²⁶
- **Utvrđiti potrebne servise:** Postoji potreba za utvrđivanjem osnovnih servisa kao što su:
 - formiranje novog objekta klasi
 - uspostavljanje ili raskidanje veze pojave objekta s drugim objektom
 - čitanje ili promjena atributa objekta
 - brisanje objekata iz klase

²⁴ Mladen Varga Baze podataka, Konceptualno, logičko i fizičko modeliranje podataka DRIP Zagreb, 1994. str 85

²⁵ Mladen Varga Baze podataka, Konceptualno, logičko i fizičko modeliranje podataka DRIP Zagreb, 1994. str 85

²⁶ Mladen Varga Baze podataka, Konceptualno, logičko i fizičko modeliranje podataka DRIP Zagreb, 1994. str 85

- **Utvrđiti poruke:** To znači utvrditi relevantne poruke za pokretanje servisa odnosno ustanoviti pošiljatelja i primatelja poruke. Primatelj poruke pritom dobiva nalog za obavljanje određene operacije. Operaciju treba navesti u specifikaciji servisa pošiljatelja, a potpuno definirati u specifikaciji servisa primatelja.
- **Razraditi detalje servisa:** Znači dokumentirati ga jednom od tehnika za prikaz logike servisa

2.5. Izrada objektnog modela:

Izrada objektnog modela ima velike sličnosti s izradom modela entiteti-veze. Dakle prikupljanje i analiza informacijskih zahtjeva se obavlja po istim principima, ali s bitnim akcentom da je uočavanje i analiza zahtjeva objektna.

Utvrđivanje objekata, njihovih atributa i strukture u objektnom modelu vrlo je slično postupku u modelu entiteti-veze zato što je entitet modela entitet-veze objekt promatranja u informacijskom sustavu. Dalje osnovna razlika između objektnih modela i modela entiteti-veze jest u definiranju servisa, koji je prikazan na prethodnoj stranici (str. 25.). Model entiteti-veze opisuju samo podatkovnu komponentu informacijskog sustava, što znači da razrađuje strukturu podataka. Model entiteti-veze pritom ne predstavlja konceptualni opis kompletnog informacijskog sustava. Protivno tome objektni model je bogatiji jer opisuje i procesnu komponentu informacijskog sustava pa predstavlja konceptualni opis kompletnog sustava..

2.5.1. Prikupljanje i analiza informacijskih zahtjeva:

Modeliranje podataka počinje prikupljanjem informacijskih zahtjeva odnosno poslovnih pravila, koje se obavlja intervjuiranjem korisnika, istraživanjem dokumentacije poslovnog sustava i sličnim postupcima. Nakon intervjuiranja i proučavanja dokumentacije poslovnog sustava analitičar priprema inicijalni poslovni model, koji se sastoji od modela podataka i modela procesa. Analitičar ih razvija istovremeno. Dakle princip je potpuno isti.

2.5.2. Izraditi objektni model:

Utvrđivanje objekta:

Za postupak utvrđivanja entiteta, potrebno je poduzeti pojedine korake.

- **Utvrđiti objekte** intervjuirajući korisnike ili koristeći se dokumentacijom poslovnog sustava.
- **Imenovati objekte:** Za imenovanje objekata je najbolje uzeti imenicu u jednini kojom se služe korisnici. Naziv objekta mora biti jedinstven, što implicira da se pojava homonima treba preimenovati.
- **Opisati objekte:** Dobar opis sastoji se od punih rečenica koje su smještene u katalogu informacijskog sustava. Također treba dokumentirati i očite činjenice o objektima, jer one ne moraju biti očite svim korisnicima. Opisom objekata utvrđujemo najvažnije attribute objekata
- **Utvrđiti specijalizaciju objekta:** Odnosi se na klasičnu definiciju specijalizacije i određivanja podtipova i nadtipova objekata kako bi bile shvatljive veze među objektima.
- **Utvrđiti zavisnost objekta**

Utvrđivanje veza:

- **Utvrđiti veze** intervjuirajući korisnike ili koristeći se dokumentacijom poslovnog sustava.
- **Imenovati veze**
- **Opisati veze**
- **Izbaciti redundantne veze:** Redundantna veza može se izvesti iz drugih veza pa je stoga nepotrebna.
- **Utvrđiti veze višeg stupnja:** Većina veza koje utvrđujemo će biti binarne veze. Povremeno se javlja potreba za uspostavljanjem veza višeg stupnja npr. 3, 4, itd. Takve veze treba pažljivo proučiti i utvrđiti.

Utvrđivanje atributa objekta

- **Utvrđiti attribute** intervenirajući korisnike ili koristeći se dokumentacijom poslovnog sustav.
- **Imenovati attribute:** Naziv atributa može se sastojati od tri djela koji imaju različitu namijenu; korijena, kvalifikatora i klase atributa.

- **Opisati atribute:** Koristiti ista pravila kao i za opis objekata uz uvjet da dodatno treba definirati domenu vrijednosti koju može poprimiti atribut.
- **Utvrđiti kardinalnost atributa** odgovarajući na sljedeća pitanja na primjer „koliko datuma zaposlenja mora imati osoba“ i „koliko datuma zaposlenja može imati osoba“
- **Verificirati pripadnost atributa:** Iz opisa poslovnih pravila, vidi se da se pripadnost atributa objektu ili vezi iskazuje na više načina: npr. visina učenika ili učenička visina
- **Grupirati povezane attribute:** Mnogo puta se mogu pronaći nekoliko međusobno povezanih atributa koji čine zasebnu cjelinu
- **Izbaciti derivirane attribute:** Derivirani atributi su oni atributi koji se mogu izvesti iz drugih atributa. Oni najčešće nisu redundantni u odnosu na jednostavne attribute i izbacuju se iz konceptualnog modela.

2.5.3. Konsolidiranje modela entiteti veze:

Nakon prethodno završenih koraka, potrebno je napraviti i posljednji „rez“ uz pomoć konsolidiranja. Objektni model uobičajeno se radi u manjim segmentima informacijskog sustava tj. po problemskim područjima odnosno podsustavima informacijskog sustava. Konačni je cilj načiniti jedinstveni model podataka informacijskog sustava. Jedinstven model podataka se sastoji od konzistentno uklopljenih podmodela podataka pojedinih problemskih područja. Zbog toga treba pojedine podmodele, nakon što su načinjeni, integrirati u model podataka informacijskog sustava. Integriranje modela podataka znači izradu jedinstvenog modela podataka od više podmodela podataka.²⁷ Integriranje se dobiva doradom pojedinih podmodela koji u početnom stadiju nisu konzistentni. Cilj konsolidacije je dobivanje skupa međusobno konzistentnih podmodela koji zajednički čine jedinstven model podataka.

²⁷ Mladen Varga Baze podataka, Konceptualno, logičko i fizičko modeliranje podataka DRIP Zagreb, 1994. str. 71

3. Zaključak

Uzimajući u obzir potrebu današnjice za uređenim modelom podataka koji odiše jedinstvenošću i konzistentnošću među podacima, konceptualno modeliranje pruža osnovu stvaranja takvog konzistentnog skupa koji može biti od velikog značaja i služiti svojim korisnicima. Također valja napomenuti važnost konceptualnog modeliranja, koje predstavlja odskočnu dasku i kamen temeljac logičkom i fizičkom modeliranju podataka. Način na koji iz kaosa podataka dolazimo do uređenog skupa podataka je konceptualno modeliranje podataka.

4.Literatura

Knjige:

C. J. Date: An Introduction to Database Systems, Addison Wesley, 2007.

R. A. Mata-Toledo, P. K. Cushman: Fundamentals of Relational Databases, Schaum's Outline Series, McGraw-Hill, 2000.

Mladen Varga Baze podataka, Konceptualno, logičko i fizičko modeliranje podataka DRIP Zagreb, 1994.

Prezentacije:

Prof. dr. sc. Mirko Maleković, predavanja Baze podataka 2

Web:

1. http://upload.wikimedia.org/wikipedia/commons/7/72/ER_Diagram_MMORPG.png učitano 09.10.2011.
2. http://upravljanjepodacima.wikia.com/wiki/KONCEPTUALNO_MODELIRANJE_PODATAKA učitano 09.10.2011

O autoru:



Zoran Hercigonja, rođen je 13.04.1990. godine u Varaždinu. Osnovnu školu "Petar Zrinski " završio je 2005. godine u Jalžabetu. Iste godine upisuje Elektrostrojarsku školu Varaždin, smjer "Tehničar za računalstvo" i uspješno položenom maturom završava 2009. godine. Daljnje školovanje nastavlja na Fakultetu organizacije i informatike Varaždin 2009. godine. Preddiplomski studij "Informatički sustavi" završava 2013. godine. Nastavlja studirati na istom fakultetu na diplomskom studiju za informatičare usmjerenja "Informatika u obrazovanju" koji završava 2015. godine s velikom pohvalom (Magna cum laude). Akademske godine 2013/2014. dobitnik je dekanove nagrade za najboljeg studenta diplomskog studija informatike. Pripravnički staž obavlja na Drugoj gimnaziji Varaždin u razdoblju između 2015. i 2016. godine. Državno-stručni ispit za nastavnika informatike uspješno polaže na V. gimnaziji Zagreb 2017. godine. Radi kao profesor informatike i matematike. Autor je više stručnih radova koje redovito objavljuje u časopisima: Pogled kroz prozor, International Journal of Digital Technology and Economy, Matematika i škola, zborniku Carnetove korisničke konferencije te u Zborniku radova Veleučilišta u Šibeniku.

ISBN 978-953-59549-5-8