

MILE PAVLIĆ

OBLIKOVANJE BAZA PODATAKA



ODJEL ZA INFORMATIKU, SVEUČILIŠTE U RIJECI

Rijeka, siječanj 2011.

prof. dr. sc. Mile Pavlić
OBLIKOVANJE BAZA PODATKA

Izdavač:

Odjel za informatiku Sveučilišta u Rijeci

Recenzenti:

dr. sc. Vladan Jovanović, redoviti profesor, Georgia Southern University, USA

dr. sc. Neven Vrčak, redoviti profesor, FOI, Varaždin

dr. sc. Krešimir Fertalj, izvanredni profesor, FER, Zagreb

Lektorica:

Marijana Trinajstić

Oblikovanje i grafička priprema:

Ivana Lovrinović

Tisak:

Digital Point tiskara d.o.o., Rijeka

Objavljivanje ovoga sveučilišnog udžbenika odobrilo je Povjerenstvo za izdavačku djelatnost Sveučilišta u Rijeci Odlukom pod brojem – Klasa: 602-09/10-01/39, Ur. broj: 2170-57-05-10-2.

SKRAĆENI PRIKAZ

1	UVOD	6
	Kome je knjiga namijenjena?	6
	Zašto čitati ovu knjigu?	6
	Koje teme obrađuje knjiga?	6
	Prilozi knjizi	7
	Zahvala.....	8
2	INFORMACIJSKI SUSTAVI	9
2.1	OSNOVNI POJMOVI	9
2.1.1	<i>Informacijski sustav</i>	9
2.2	BAZA PODATAKA	13
2.2.1	<i>Projektiranje informacijskoga sustava</i>	18
2.2.2	<i>Proizvodnja softvera</i>	21
2.2.3	<i>Uvođenje u primjenu</i>	23
2.3	SPECIJALIZIRANA METODOLOGIJA MIRIS	23
3	RELACIJSKE BAZE PODATAKA	27
3.1	UVOD U RELACIJSKU METODU	27
4	MODELI PODATAKA	28
4.1	PROBLEMI, STRATEGIJE I KRITERIJI MODELIRANJA PODATAKA.....	28
4.2	OSNOVNI POJMOVI O MODELIRANJU PODATAKA	30
4.3	APSTRAKCIJA PODATAKA.....	32
5	METODA ENTITETI -VEZE	35
5.1	UVOD U METODU ENTITETI - VEZE.....	35
	REFERENCES	36

Predgovor

S osobitim zadovoljstvom preporučam ovu knjigu informatičarima zainteresiranima za stjecanje neophodnih znanja i vještina pri izradi modela podataka u kontekstu integriranih informacijskih sustava. Iako je modeliranje ključna kompetencija u oblikovanju baza podataka, na žalost, čak i u svijetu, posvećen joj je malen broj kvalitetnih knjiga. Smatram da je ovakva situacija posljedica nedostatka iskustva, tj. prilike za aktivno oblikovanje komercijalnih baza podataka. Profesor Pavlić je jedan od rijetkih koji je znanje modeliranja stjecao u praksi, a posjeduje sposobnost apstrakcije neophodnu pri koncipiranju generalnih znanja.

Pred vama je knjiga pisana jasnim stilom, koncizan priručnik bogatoga sadržaja na manje od 200 stranica. Posebno poglavlje čini čak dvanaest praktičnih primjera. Odabir izdvojenih primjera doprinosi bržem čitanju i istovremeno daje punu slobodu da se primjeri biraju u slijedu primjerenom interesu svakoga čitatelja. Jedno je sigurno, nakon korištenja knjige i ponuđenih primjera, čitatelji će biti bogatiji za značajno iskustvo i steći će samopouzdanje kojim će se uspješno posvetiti oblikovanju baze podataka.

Pitanje izbora notacije modeliranja, tj. dijagrama reprezentacije, nije jednostavno. Knjizi je cilj što lakše naučiti čitatelja kako modelirati i u tu svrhu koristiti semantički bogat model. Očito je da će nakon usvajanja semantički bogatih koncepta biti lakše naučiti modelirati. Nakon naučenoga modeliranja, sama notacija neće predstavljati problem. Alternative, kao što su UML ili IDEF1X, popularne su i podržavaju ih CASE alati, autor je iz vlastita iskustva zaključio da je predložena notacija uspješna u fazi učenja konceptualnoga modeliranja te da će pomoći usvajanju drugih notacija kada se za njima ukaže potreba. Pitanje notacije uvijek je kontroverzno jer ne postoji jedna notacija odgovarajuća svim zahtjevima. Za crtanje modela u bilo kojoj notaciji može se adaptirati Visio stencil i izbjeći kupovina skupih alata.

Slijed autorova izlaganja u skladu je s modernim pristupom predavanju i praksi oblikovanja baza podataka što se u punoj mjeri odnosi i na dva pomoćna poglavlja, treće i sedmo. Kao preduvjet validaciji modela podataka zahtijeva se umijeće oblikovanja upita nad bazom podataka korištenjem SQL-a jer se ne može očekivati koristan model, ukoliko ne postoji jasna predodžba o pitanju koje se otvara mogućoj budućoj bazi podataka. Kao drugo, kada se logički dizajn (relacijska shema) izvede iz modela podataka, potrebna je validacija kvalitete samoga dizajna. Normalizacija, obrađena u sedmome poglavlju, služi prvenstveno izbjegavanju anomalija u ažuriranju podataka, do anomalija može doći u slučaju primjene baze s redundantnim sadržajem. Vrijednost dobrog modeliranja očituje se u kvalitetnome dizajnu koji ne treba dalje normalizirati jer je dobrim modeliranjem normalizacija već ostvarena, svaki podatak na jednome mjestu i jedno mjesto za svaki podatak.

Na kraju, zadovoljstvo je vidjeti kako škola modeliranja podataka napreduje u rukama sposobnog učenika. Prije 25 godina autor knjige bio je gdje ste sada, dragi čitatelji, i Vi kao studenti, na početku karijere, a sada se, jasnoćom ove knjige, i Vama otvara mogućnost uspješne karijere.

21.08.2010.

Vladan Jovanovic, Ph.D.
Professor Computer Sciences
Georgia Southern University
Statesboro GA, USA

1 UVOD

Cilj je ove knjige pomoći oblikovati bazu podataka. Većina poslovnih aplikacija podatke čuva u relacijskim bazama podataka te je potrebno poznavati teoriju oblikovanja relacijskih baza podataka (nazovimo je relacijska metoda, skraćeno RM). Oblikovanje baza podataka (engl. Database Design) može se izvesti relacijskom metodom na logičkoj razini i definirati shemu baze podataka. Na osnovi sheme se, uz pomoću softverskih sustava za upravljanje relacijskom bazom podataka (kao Oracle, MS SQL, DB2...) organizira fizička baza podataka. Fizičko i logičko oblikovanje baza podataka dva su odvojena koraka. Oblikovanje baza podataka općenito sastoji se od triju koraka i to konceptualnog, logičkog i fizičkog oblikovanja.

Standardna metoda za konceptualno oblikovanje je EV (metoda entiteta i veza, engl. ER) koja dovodi do konceptualnoga prikaza poslovnog sustava. Model podataka dobiven EV metodom jednostavno je prevesti u relacijsku shemu baze podataka i s njom, uz pomoć sustava za upravljanje bazom podataka, oblikovati fizičku bazu podataka.

U ovoj knjizi bit će prikazane navedene metode, EV i relacijska metoda te proces modeliranja podataka obadvojema metodama, kao i proces i pravila prevođenja EV modela podataka u relacijsku shemu.

Kome je knjiga namijenjena?

Ova je knjiga proistekla iz želje i s nadom da će spoznaje biti korisne praktičarima koji projektiraju informacijske sustave (oblikuju bazu podataka i oblikuju programski proizvod) i studentima, kako za modeliranje podataka tako i za razvoj informacijskih sustava nakon formalnoga školovanja.

Knjiga je namijenjena projektantima informacijskih sustava, analitičarima, organizatorima, programerima, rukovoditeljima informatičkih centara, studentima i svima onima koji u poslovno - proizvodnim sustavima sudjeluju u razvoju informacijskih sustava i polaznicima seminara o projektiranju informacijskih sustava, projektiranju baza podataka, modeliranju podataka, sistem analizi i dizajnu. Knjiga će poslužiti studentima redovitog i postdiplomskog obrazovanja.

Zašto čitati ovu knjigu?

Ukoliko imate problema s informacijskim sustavom, bazom podataka i programskom podrškom, potrebno Vam je znanje o tome kako izgraditi informacijski sustav, oblikovati bazu podataka i oblikovati programsku podršku.

Knjiga obiluje objašnjenjima i definicijama vezanim uz metode za oblikovanje baza podataka.

Knjiga je rezultat višegodišnjeg iskustva u projektiranju i gradnji informacijskih sustava, baza podataka, različitih projekata i primjeni metoda za oblikovanje pa su u knjizi iznijeta iskustva i mišljenje o metodama.

Prikazane metode primjenjive su kako za razvoj novih informacijskih sustava tako i za reinženjering postojećih programskih rješenja i baza podataka.

Koje teme obrađuje knjiga?

Središnji je dio knjige metoda entiteta i veza. Prikazani su svi koncepti potrebni za izradu modela podataka. Prikazana je relacijska metoda. Većina baza podataka su danas relacijske baze podataka. Modeliranje podataka je lakše izvesti metodom entiteta i veza te se ona preporuča kao metoda za oblikovanje baza podataka. Nakon oblikovanja pristupamo postupku prevođenja EV modela podataka u relacijsku shemu baze podataka. Posebno poglavlje detaljno opisuje ovaj proces prevođenja. Prikazana je normalizacija podataka i njezino tumačenje dano konceptima metoda entitetima veza. Proces izrade modela podataka prikazan je kao zasebna metoda VATAK (opisana u 8. poglavlju) koja pomaže projektantima pri brzom i kvalitetnoj izradi modela podataka. Na kraju je dano desetak primjera dokumenata i njihovih modela podataka.

Knjiga ima devet poglavlja. Prvo je poglavlje uvodno.

Poglavlje 2 Informacijski sustavi prikazuje položaj baze podataka u informacijskom sustavu. Definirani su pojmovi vezani uz baze podataka. Prikazano je projektiranje informacijskih sustava, proizvodnja softvera i uvođenje gotovih aplikacija. Iz teorije i prakse razvijena je i u knjizi prikazana vlastita specijalizirana metodologija MIRIS, metodologija za analizu i oblikovanje informacijskih sustava, kao mogući okvir svim fazama razvoja od strateškoga planiranja do održavanja. Oblikovanje baza podataka samo je jedna od faza razvoja informacijskoga sustava. Postoje druge brojne slične specijalizirane metodologije koje su standardizirali pojedinci ili tvrtke, ali gotovo svaki programer ima vlastiti nestandardizirani način razvoja informacijskih sustava.

Poglavlje 3 Relacijska baza podataka prikazuje pojmove koji čine strukturu relacijske metode i pojmove ograničenja u relacijskom modelu iz kojih slijedi pojam ključnog atributa. Ukratko su prikazani operatori relacijske algebre pri čemu je SQL - standardni upitni jezik za relacijske baze podataka, prikazan uklatko.

Poglavlje 4 Modeli podataka prikazuje osnovne pojmove i metode modeliranja podataka. Posebno je opisana apstrakcija podataka koja se nalazi u osnovi svih metoda.

Poglavlje 5 Metoda entiteta i veza prikazuje jedan po jedan koncepte metode EV detaljno i predstavlja središnji dio ove knjige. Uz svaki koncept odmah se izlažu njegova ograničenja i prikazuje njegovo prevođenje u shemu relacijske baze podataka.

Poglavlje 6 Prevođenje definira pravila za prevođenje dijagrama dobivenih metodom EV-a u relacijsku shemu baze podataka.

Poglavlje 7 Normalizacija je uobičajen prikaz pogreške pri organiziranju sheme baze podataka te procesa otklanjanja jedne po jedne pogreške sve dok se shema ne dovede u oblik koji ne stvara probleme u održavanju baze podataka. Pored uobičajenog prikaza tablice, prikazan je i model podataka metodom EV-a i to prije i nakon normalizacije. Ovim načinom grafički se predočuju pogreške koje pri modeliranju uzrokuju probleme u bazi podataka.

Poglavlje 8 Proces oblikovanja baze podataka pokazuje proceduru pri izradi modela podataka metodom EV-a. Sama procedura je zasebna metoda modeliranja podataka i kao takva može se, s manjim prilagodbama, koristiti za različite varijante metode EV-a.

Poglavlje 9 Praktični primjeri modeliranja podataka prikazuje konkretne popunjene dokumente iz poslovnoga sustava te za njih daje prikaz modela podataka. Tako čitatelj može vidjeti konkretnu primjenu metode EV-a i prevođenja dijagrama EV-a u relacijsku shemu. Na primjerima su također postavljeni zadaci koje treba dodatno riješiti proširenjem modela podataka zbog korisničkih zahtjeva.

Prilozi knjizi

Na internetskoj adresi: <http://tinyurl.com/oblikovanje-baze-podataka> nalaze se korisni tekstovi i koncepti poput, na primjer projektne dokumentacije koju su izradili analitičari i studenti. Takvim pristupom čitatelju se omogućuje uvid u primjenu navedenih metoda i potpuni oblik informacijskoga sustava dokumentiran fazama analize i oblikovanja.

Zahvala

Zahvaljujem mnogobrojnim prijateljima i suradnicima s kojima sam radio na nizu projekata u okviru kojih smo učili i isprobavali metode.

Zahvaljujem brojnim učiteljima, prijateljima, kolegama i partnerima na savjetima i pomoći u promišljanjima o temama, mogućnostima, primjerima i praktičnim problemima informacijskih sustava.

Zahvaljujem programerima u tvrtki RIS na suradnji pri izradi mnogobrojnih primjera softvera i informacijskih sustava.

Zahvaljujem mnogobrojnim suradnicima s HRT-a, iz Croatia osiguranja, Croatia banke, Elektromaterijala, Konta i drugim poslovnim organizacijama na suradnji, na njihovoj želji za unapređenjem informacijskoga sustava i njihovoj vjeri u naše napore.

Posebno zahvaljujem Martini Ašenbrener, za golem trud u izradi knjige, prikupljanju i obradi izvorne građe, podršci i vjeri u uspjeh.

Zahvaljujem studentima Odjela za informatiku, Sveučilišta u Rijeci, mojim magistrantima i doktorandima, polaznicima jednogodišnje Škole za projektiranje informacijskih sustava na njihovim mnogobrojnim sugestijama, pitanjima, diplomskim, seminarskim, magistarskim, doktorskim radovima koji su pridonijeli nastanku ove knjige.

Rijeka, siječanj 2011.

prof. dr. sc. Mile Pavlić

2 INFORMACIJSKI SUSTAVI

2.1 Osnovni pojmovi

2.1.1 Informacijski sustav

Pojam informacijski sustav (skraćeno IS) sastoji se od dvaju pojmova, termina koje je potrebno definirati: „sustav“ i „informacija“.

Sustav (engl. *system*) je skup dijelova (elemenata), veza između dijelova te osobina dijelova svrsishodno organiziranih za neki proces (funkciju).

Svaki sustav nalazi se u širem sustavu kojemu pripada i s kojim je u vezi. Dio sveukupne cjeline koji nije obuhvaćen sustavom naziva se **okolina sustava**. Okolina je skup okolnih sustava. Vezu sustava i okoline predstavljamo ulazima kada okolina predaje sustavu materiju, energiju ili informacije, a izlazima kada to sustav predaje okolini. Na slici 2.1. prikazan je osnovni model sustava.



Slika 2.1 Osnovni model sustava

U ljudskome društvu postoje mnogobrojne formalne i neformalne organizacije. Svaka organizacija je sustav.

Poslovna organizacija (poslovni sustav, poduzeće, ustanova, tvrtka, engl. *business organization*) je organizacija koja se bavi jednom ili više poslovnih djelatnosti kao svojom misijom. Svaka poslovna organizacija ima informacijski sustav.

Poslovna djelatnost (engl. *business sector*) je sve ono što poslovna organizacija čini kako bi ostvarila prihode, bilo izravno (kao što je izrada proizvoda za prodaju) ili neizravno (kao što je pružanje usluga – na primjer konzultant koji savjetuje poslovne organizacije kako poboljšati svoj proizvod) (CEO, 2005.).

Za poslovne organizacije i informacijske sustave temeljni pojmovi su podatak i informacija. Za podatak kažemo da je nositelj informacija, a informacija je podatak koji su ljudi protumačili. Za te pojmove postoje brojne definicije (samo ISO ima u svojoj dokumentaciji preko 50 definicija pojma informacija, vidi <http://cdb.iso.org/cdb/search.action>). Ovdje ćemo navesti samo neke.

Podatak (engl. *data*) je bilo koji predmet mišljenja koji može prenijeti informaciju; formalizirani znakovni prikaz činjenica, pojmova i naredaba pogodan za priopćavanje, interpretiranje te analognu i digitalnu obradu (Kiš, 2002.).

Informacija (engl. *information*)¹ je takav podatak koji u tijeku privredne aktivnosti donosi nova znanja koja nam na osnovi postojećih znanja i iskustava govore o promjeni nekoga stanja unutar poduzeća ili u okruženju. To je podatak koji je prethodno provjeren i analiziran te kao takav predstavlja prikladnu osnovu za donošenje odluka o utjecaju na daljnji tijek privredne aktivnosti (Papp, 1979.).

Podatak je diskretna vrijednost neke fizičke veličine, na primjer 67 je broj koji se sastoji od dva znaka. Podatak je skup znakova. Znajući vrijednost broja 67, a ne znajući kojoj vrsti stvari u sustavu pripada i što znači, kažemo da imamo neki podatak. Podaci su nositelji informacija. Podaci se mogu zapisivati na različitim medijima.

Ako kažemo (ili pomislimo, ili ako znamo, ili je na papiru zapisana činjenica), „visina stola je 67 cm“, onda imamo informaciju. Ova zapisana činjenica je informacija svakoj osobi koja će je znati protumačiti iz svoga konteksta znanja o stvarnosti. Za osobu koja ne zna pročitati rečenicu na tom jeziku, ta činjenica je podatak. Dakle, ovisno o kontekstu, informacija za nekoga može biti podatak i obrnuto.

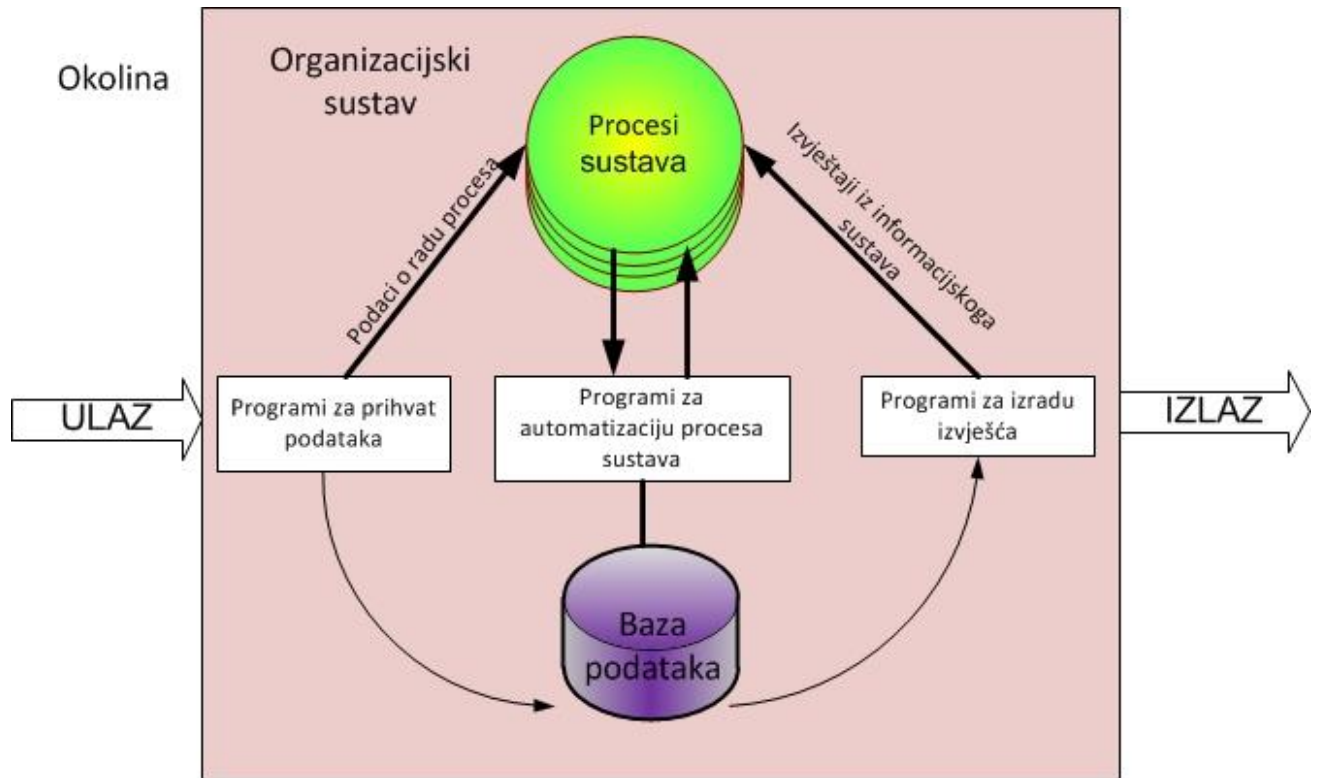
Informacijski sustav (engl. *Information System, skraćeno IS*) je objedinjeni skup komponenti (računalnog hardvera i softvera, baza podataka, telekomunikacijskih sustava, ljudskih resursa te procesa) koje služe za prikupljanje, pohranu, obradu te prenošenje informacija (vidi <http://www.britannica.com/EBchecked/topic/287895/information-system>).

Informacijski sustav je skup povezanih dijelova (softver, hardver, ljudi, procedure, informacije, te komunikacijske mreže) kojima je cilj pribaviti, pohraniti, obraditi i prenijeti informacije za funkcioniranje, planiranje, odlučivanje i/ili upravljanje poslovnom organizacijom.

Informacijski sustav zasnovan na računalu čine ljudi (obrazovani za rad s računalima) te programska i računalna oprema (softver i hardver) koja je napravljena, oblikovana i dovedena u operativno stanje te se pomoću nje informacije skupljaju (engl. *data acquisition*), zapisuju (engl. *recording*), obrađuju (engl. *processing*), spremaju i pronalaze (engl. *information storage and retrieval*) te prikazuju u odgovarajućem obliku (Kiš, 1993.).

Informacijski sustav, sa stajališta osnovnih komponenti strukture, pogodno je prikazati grafički (vidi sl. 2.2.).

¹ U hrvatskome jeziku naziva se još i „obavijest“. Postoji velik broj definicija pojma informacije s različitih aspekata kao: obavijest, neizvjesnost, komunikacija, semantika, događaj, struktura, znanstvena činjenica, svojstvo materije.



Slika 2.2 Dijelovi informacijskoga sustava i njihova povezanost s okolinom

Börje Langefors je definirao osnove teorije informacijskih sustava². Većinu svojih istraživanja posvetio je generalnoj teoriji koja definira tu disciplinu, neovisno o tehnologiji koja se stalno razvija. Langeforsova definicija glasi: "Informacijski sustav temeljen na računalima je tehnološki ostvaren medij za zapisivanje, spremanje i razumijevanje te širenje lingvističkih izraza, kao i zaključivanje iz istih."

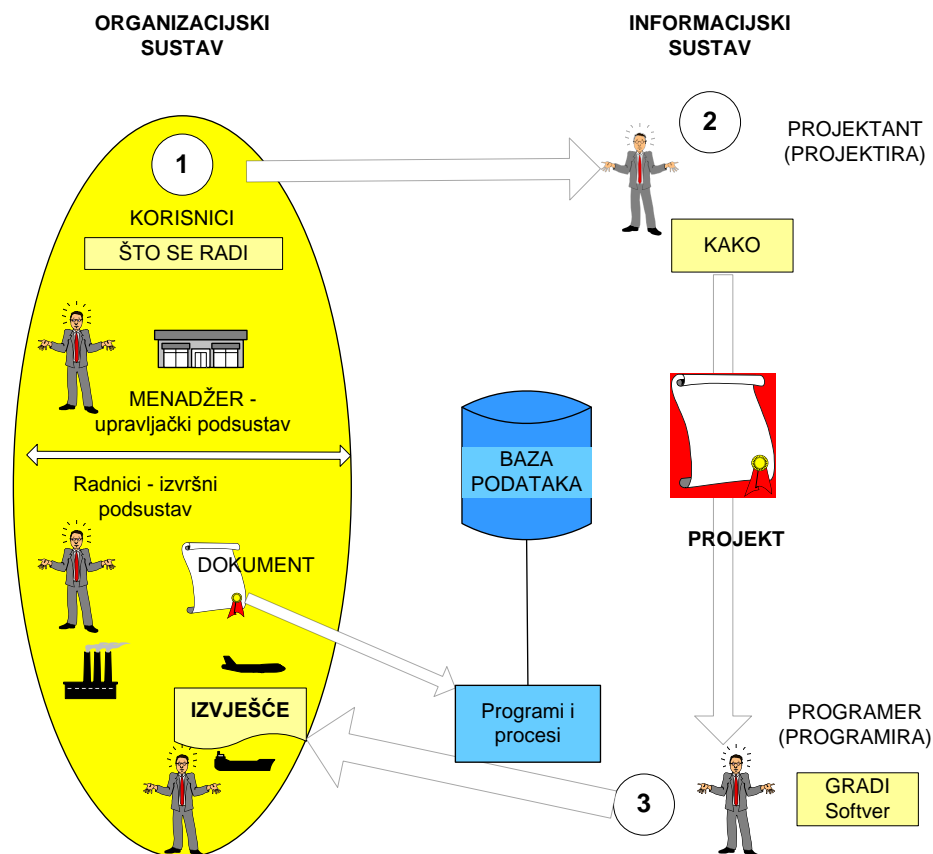
Informacijski sustav, sa stajališta funkcije u organizacijskome sustavu, ima zadatak permanentno opskrbljivati potrebnim podacima poslovne procese i sve razine upravljanja tim procesima te odlučivanja u tim procesima.

Temeljne funkcije informacijskoga sustava su:

- prikupljanje
- upis (pohranjivanje, engl. storage) podataka u bazu podataka
- obrada (procesiranje, engl. processing) podataka
- prikaz i ispostavljanje (izdavanje na korištenje, engl. dissemination) podataka iz baze podataka
- čuvanje (dokumentiranje, trajno pohranjivanje) podataka.

Informacijski sustav je dio organizacijskoga sustava (vidi sl. 2.3) koji podacima povezuje cijeli sustav i koji se kontinuirano održava.

² Vidi http://hr.wikipedia.org/wiki/Informacijski_sustavi, 19.4.2010.



Slika 2.3 Informacijski sustav kao integralni dio organizacijskoga sustava

U širem smislu dijelovi informacijskoga sustava su: dokumenti s podacima, ljudi koji koriste i kreiraju dokumente, uređaji za manipuliranje dokumentima, računalna tehnologija, baza podataka, ljudi koji upisuju podatke u bazu podataka i komunikacijska tehnologija.

Skup elemenata koji iz okoline djeluju na sustav naziva se „**ulaz u sustav**“ (vidi sl. 2.2). Ulazi u sustav su informacije koje sustav prima iz okoline. To mogu biti: nacrti, računi, dostavnice, telefonske poruke, izvodi banke, izvješća, usmene poruke, novine i dr. Podaci s ulaza u sustav prepisuju se u bazu podataka pomoću programa za prihvatanje podataka.

Programi za prihvatanje ulaznih podataka važan su dio informacijskoga sustava, jednako kao i programi za izvješćivanje. Prvi prihvaćaju podatke i smještaju ih u bazu podataka za buduće potrebe kada će iz njih programi za izvješćivanje kreirati potrebne skupove podataka.

Baza podataka je mjesto gdje se čuvaju podaci. Baza podataka može biti različitih vrsta i oblika, na raznim tehnologijama. Najvažnija je značajka baze podataka, tj. ono što je čini bazom podataka, što ona sadrži podatke o prošlosti i sadašnjosti organizacijskoga sustava.

Prihvatom i upisom podataka u bazu podataka informacijski sustav je ostvario svoju prvu temeljnu funkciju. Temeljna je funkcija organizacijskoga sustava korištenje ulaza u sustav kako bi na osnovi njih poslovni procesi sustava kreirali nove informacije koje mogu ići kao ulazi u druge procese ili predstavljati **izlaze iz sustava**. Ako se neki od procesa poslovnoga sustava može potpuno ili djelomice automatizirati i ugraditi u informacijski sustav, onda informacijski sustav svojim radom obavlja temeljne funkcije organizacijskoga sustava.

Druga temeljna funkcija informacijskoga sustava je izvođenje poslovnih procesa organizacijskoga sustava. Što je više poslovnih procesa organizacijskoga sustava automatizirano i ne izvode ih ljudi, već IT (informacijska tehnologija) uz pomoć i kontrolu ljudi, to je organizacijski sustav efikasniji i ekonomičniji. Cilj je automatizirati što više poslovnih procesa (na primjer uvesti bankomate umjesto šalterskih radnika za podizanje novca, uvesti aplikaciju za obradu plaća ili završni račun tvrtke koja će omogućiti izradbu više varijanti plaćanja poreza, uvesti robote za automatsko rezanje limova u tvornici na osnovi nacrtu projektnog ureda i sl.)

Izveščivanje je proces transformacije podataka iz baze podataka u podatke potrebne korisnicima. Podaci se prikazuju na izvješćima u analitičkom ili sintetičkom obliku, u većem ili manjem obimu. Izvješće može biti na različitim medijima, papirnatima, od bušene trake do A0, mikrofilmovima, magnetskim medijima svih vrsta, optičkim diskovima, e-mail i drugim porukama, elektromagnetskim signalima emitiranim različitim komunikacijskim kanalima u uređaje primateljima i dr. Nekada je to bio uglavnom papir, a danas je to najčešće zaslon računala.

2.2 Baza podataka

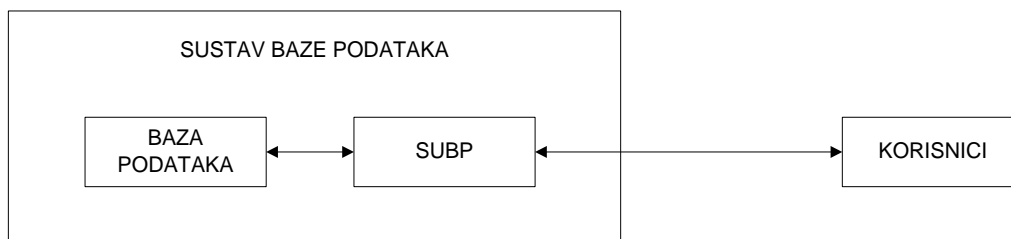
Baza podataka (skraćeno BP, engl. Database) predstavlja neredundantni skup podataka o stanju sustava strukturiran na način opisan u shemi baze podataka. Baza podataka je skup podataka o svim pojavljivanjima entiteta, svih veza i svih njihovih atributa opisanih u shemi baze podataka.

Baza podataka je skupina ustrojjenih, logički povezanih zapisa (engl. *Record*) ili datoteka (engl. *File*); ili skupina datoteka koje sadrže zapise s podacima što su međusobno u nekoj vezi, a korisnici ih mogu rabiti u različite svrhe; može se sastojati i od pomoćnih datoteka (na primjer datoteke s indeksima) (Kiš, 1993.).

Shema baze podataka je zapis konkretne podatkovne strukture i ograničenja među podacima poslovne organizacije odabranom metodom za modeliranje podataka.

Sustav baze podataka je baza podataka i sustav za upravljanje bazom podataka (vidi sl. 2.4.).

Sustav za upravljanje bazama podataka (skraćeno SUBP, engl. Database Management System (DBMS)) je softver koji upravlja korištenjem baze podataka, odnosno ima ulogu posrednika između korisnika i same baze podataka. Neki od poznatijih SUBP-ova dani su u tablici 2.1.



Slika 2.4 Struktura sustava baze podataka

Konkretni SUBP-ovi sastoje se iz dvaju dijelova:

- jezika za opis podataka (skraćeno JOP) i

- jezika za rukovanje podacima (skraćeno JRP)

Jezik za opis podataka (engl. *Data Definition Language*, skraćeno DDL) je skup sintaksnih pravila pomoću kojih je moguće definirati (opisati) shemu baze podataka.

Jezik za rukovanje podacima (engl. *Data Manipulation Language*, skraćeno DML) je računalni jezik pomoću koga programeri ili korisnici mogu dodavati, brisati ili mijenjati podatke u bazi podataka.

Pored operatora u DML jeziku postoje i procedure baze podataka za:

- očuvanje integriteta
- optimiziranje
- praćenje stanja baze podataka
- statističke operacije

Generalizacija tipova SUBP-ova koji postoje, razvijenih na osnovi tipova modela podataka su:

- hijerarhijski
- mrežni
- relacijski
- postrelacijski (entity – relationship)
- objektni

Tijekom razvoja baza podataka razvijan je i DDL te tako postoje generacije arhitekture DDL-a. U ovisnosti o stupnju razdvojenosti fizičke organizacije podataka (interne sheme) u bazi podataka od pogleda na podatke (eksterne sheme) iz aplikacijskoga programa, postoje generacije SUBP-a:

I. generacija: Interna shema → aplikacijski program

II. generacija: Interna shema → konceptualna shema → aplikacijski program

III. generacija: Interna shema → konceptualna shema → eksterna shema → aplikacijski program

Svaka promjena konceptualne sheme zahtijeva izmjenu aplikacijskih programa i potreba za održavanjem softvera i dalje postoji.

ANSI (*American National Standards Institute*) iz skupine SPARC (*Standarts Planning And Requirements Committee*), predložio je zbog prije navedenih razloga, idealnu arhitekturu DDL, organiziranu u tri odvojene razine³ (vidi sl. 2.5).

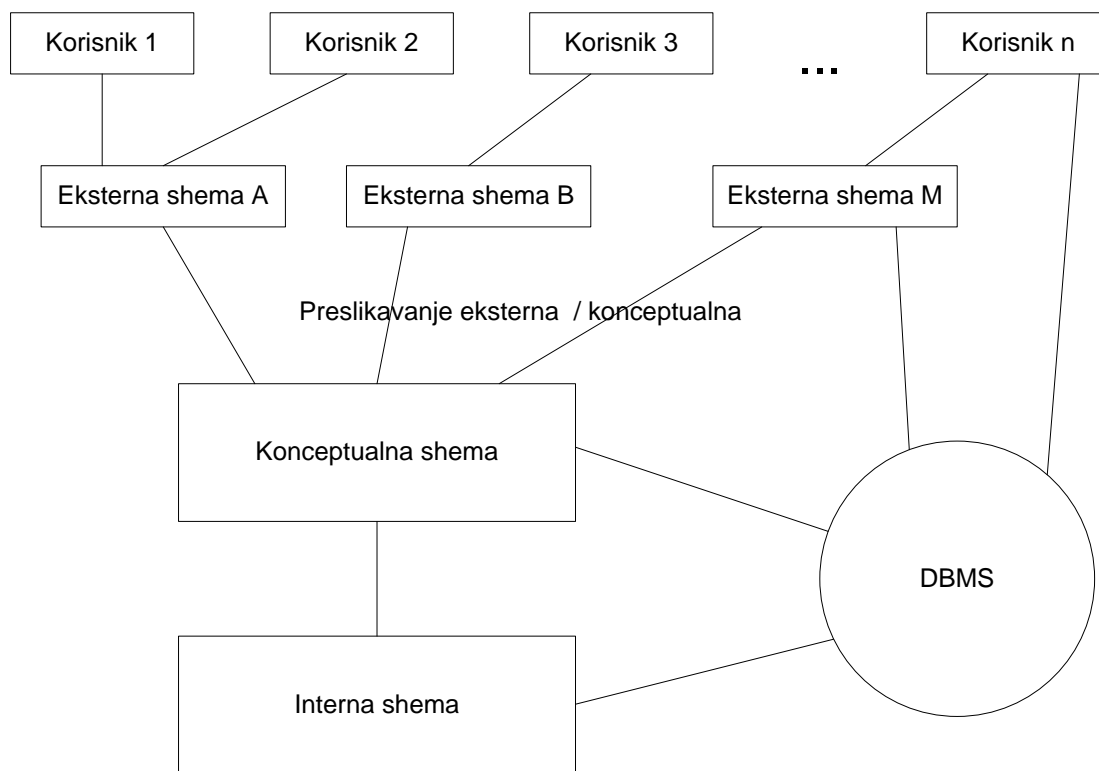
Interna shema (engl. *internal schema*) je skup definicija o fizičkoj organizaciji baze podataka i putevima pristupa specifičnim za pojedine SUBP-ove. Aplikacijski program u I. generaciji, koji pristupa takvoj shemi, mora sadržavati znanje o fizičkoj strukturi i vezama

³ Vidi <http://db.grussell.org/section002.html>, 6.5.2010.

među podacima te je ovisan o fizičkoj lokaciji podataka u bazi podataka. Problem je izmjena definicije u DDL-u jer to zahtijeva izmjenu aplikacijskih programa.

Konceptualna shema (engl. *conceptual schema*) je skup definicija o logičkoj organizaciji baze podataka. Programi nisu ovisni o promjenama u internoj shemi. Postignut je viši stupanj neovisnosti, ali postoji ovisnost programa o logičkim pozicijama podataka (na primjer relacija u relacijskoj bazi podataka).

Eksterna shema (engl. *external schema*) je skup definicija o podacima kako ih vidi aplikacijski program. Pogled na podatke zaseban je za svaku aplikaciju i pri izmjeni aplikacije mijenja se samo pripadni pogled u eksternoj shemi, neovisno o fizičkoj i logičkoj organizaciji podataka. Pri izmjeni konceptualne sheme ne mijenja se niti jedan aplikacijski program koji nije povezan s danom promjenom. Ako je dodan novi atribut, a želimo ga imati u aplikaciji, mijenja se aplikacijski program. Premještanje atributa, proširenje polja, promjena tipa polja, dodavanje novih atributa i entiteta ne utječe na programe.



Slika 2.5 Tri razine arhitekture baze podataka

Eksterna shema izvodi se iz konceptualne sheme.

Konceptualna shema sadrži definicije o svim podacima, bez zalihosti (redundancije), na jedinstven način, neovisno o fizičkoj implementaciji i aplikaciji. Iz nje se mogu izvesti svi pogledi.

Osnovni je cilj projektanta baze podataka modelirati konceptualnu shemu. Administrator baze podataka kreira internu shemu na osnovi konceptualne i priprema eksternu shemu za pojedini aplikacijski program. Projektanti aplikacija zahtijevaju eksternu shemu. Ako u eksternoj shemi ima atributa koji ne postoje u konceptualnoj, oni se dodaju u konceptualnu i internu shemu.

Ova se knjiga bavi metodama za modeliranje konceptualne sheme, bez obzira na konkretne SUBP-ove.

Administracija baze podataka različita je za različite SUBP-ove, i detaljnije se izučava pri upotrebi konkretnog SUBP-a.

Neki sustavi za upravljanje relacijskim bazama i relacijske baze podataka prikazani su u tablici 2.1.

Tablica 2.1 Sustavi za upravljanje bazom podataka

Naziv alata	Autor	Web
CA-Datcom	Computer Associates	http://www.ca.com/us/products/product.aspx?id=1237
Ingres	University of California, Berkeley	http://www.ingres.com/
DB2	IBM	http://www-01.ibm.com/software/data/db2/
Oracle	Oracle	http://www.oracle.com/index.html
Informix	IBM	http://www-01.ibm.com/software/data/informix/
MySQL	Ulf Michael Widenius	http://www.mysql.com/about/
PostgreSQL	Michael Stonebraker University of California at Berkeley Computer Science Department	http://www.postgresql.org/docs/current/static/intro-what-is.html
Model 204	Computer Corporation	http://www.cca-int.com/prodinfo/m204.html
NonStop SQL	Tandem Computers	http://h20338.www2.hp.com/NonStopComputing/cache/76708-0-0-225-121.html
SQL Server	Microsoft	http://www.microsoft.com/sqlserver/2008/en/us/
SQLBase	Centura Software Corporation	http://www.unify.com/Products/Data_Management/default.aspx
Impromptu	Cognos Inc.	http://www-01.ibm.com/software/data/cognos/products/series7/impromptu/
Teradata	Teradata	http://www.teradata.com/t/
Progress	Progress Software Corporation	http://web.progress.com/en/index.html
Access	Microsoft	http://www.microsoft.com
dBASE	Ashton-Tate Corporation	http://en.wikipedia.org/wiki/Ashton-Tate

Vrlo velike baze podataka (engl. *Very Large Database*, VLDB) dosežu po kapacitetu oko: 128 TB (terabajta), a cijeli sustav baze može imati 64000 takvih baza⁴, što iznosi 8.192.000.000.000.000 znakova ili 8,192 EB-a. Veličina datoteka (ili baze podataka) može se izraziti u jedinicama mjere prikazanima u tablici 2.2.

Tablica 2.2 Jedinice mjere količine podataka

⁴ podaci su preuzeti s <http://www.oracle-base.com/articles/10g/ImprovedVLDBSupport10g.php> (10.4.2008.)

Ime prefiksa	Uobičajen naziv	Simbol	Standard SI5	Potencija broja 2
	jedan	1	100	$2^0 = 1$
deka-	deset	D	101	-
hecto-	sto	h	102	-
kilo-	tisuću	k ili K	103	$2^{10} = 1,024$
mega-	milijun	M	106	$2^{20} = 1,048,576$
giga-	milijarda	G	109	2^{30}
tera-	bilijun	T	1012	2^{40}
peta-	bilijarda	P	1015	2^{50}
exa-	trilijun	E	1018	2^{60}
zetta-	trilijarda	Z	1021	2^{70}
yotta-	kvadrilijun	Y	1024	2^{80}
...				
googol ⁶	gugol	GO ⁷	10100	
googolplex	gugolopleks	GP ⁸	$((10)^{10})^{100}$ $(10)^{\text{googol}}$	

Prefiksi množitelji: *kilo*, *mega*, *giga* i dr. koriste se u fizici i drugim znanostima, ali i u ICT-u za mjerenje količine bita (engl. *bit*, skraćeno „b“) i bajtova (engl. *byte*, skraćeno „B“).

Prefiks *kilo* ima kraticu *k* i predstavlja 10^3 , odnosno 1000 jedinica, međutim u informacijskim znanostima, za termine koji su prefigurirani tim prefiksom, koristi se kratica *K* (dakle veliko slovo) kako bi bilo jasno da vrijednost *kilo* u tome području predstavlja drugu vrijednost: 2^{10} , odnosno 1024 jedinice. Uobičajeno je da se količina prenesenih podataka mjeri u bitima s potencijom na bazi 10. Kapacitet memorije mjeri se uobičajeno u bajtovima s potencijom na bazi 2. Tako na primjer kapacitet nekoga medija od 5 KB (kilobajta) predstavlja 5120 znakova. U većih jedinica nema razlike u velikom i malom slovu i u komunikaciji se podrazumijeva zaokružen broj prema SI standardu, dok je stvarni kapacitet obično malo različit.

Velike baze podataka brzo se šire i proizvođači softvera za upravljanje bazom podataka stalno proširuju mogućnosti svojih proizvoda. VLDB održavaju mnogobrojne ekipe projekatana, analitičara i programera.

⁵ SI je oznaka za međunarodni standard (prema francuskome nazivu *Système International d'Unités*, engl. *International System of Units*).

⁶ Gugol je veliki broj. Može se zapisati kao 1 iza koga slijedi 100 nula (Naprimjer broj tisuću je s tri nule, tj. 1000). Procjenjuje se da u cijelom vidljivom svemiru ima najviše 10^{81} atoma. Nazivom za taj broj motivirano je ime poznate softverske kompanije *Google* (ova je tvrtka ispočetka doista i nosila naziv *googol*, međutim ubrzo se na jednome čeku koji je trebao biti isplaćen na ime tvrtke dogodila pravopisna zabuna. Osnivači tvrtke ocijenili su u tome trenutku da im je spretnije prihvatiti novi način zapisivanja imena, nego tražiti od isplatelja da ispravi svoju pogrešku te je tako tvrtka dobila ime koje nosi i danas).

⁷ Autor nije našao kraticu za taj i sljedeće termine te je predložio kratice.

⁸ *Googolplex* je naziv za broj (izvedeno iz *googol* i *complex*), a predstavlja broj koji se može zapisati kao 1 iza koga slijedi gugol nula. Taj je broj velik kao broj atoma u gugolu svemira malo većih od našega svemira.

U najvećih poslovnih organizacija baza podataka može dnevno porasti za nekoliko gigabajta. Potrebno je, pored upisa podataka, manipulirati i obrađivati velike skupove podataka. Takve obrade nad skupinama informacija u gigabajtima podataka o transakcijama usporavaju rad i ometaju funkcioniranje sustava. Stoga se podaci izdvajaju u zasebnu bazu podataka nad kojom se izvode operacije. Tri koncepta koji omogućuju kopanje po podacima, i predstavljaju proširenje informacijskoga sustava i sa stajališta baze podataka i sa stajališta softverskih i metodoloških rješenja su:

- „skladište“ podataka (engl. *Data warehouse*)
- „tržnica“ podataka (engl. *Data mart*)
- „rudarenje“ (prekopavanje) podataka (engl. *Data mining*).

Data Warehouse (skraćeno **DW**) je baza starih podataka o poslovnim transakcijama organizirana tako da olakša učinkovite upite za donošenje marketinških, taktičkih i strateških odluka.

Data Mart (skraćeno **DM**) je skupina prikupljenih podataka iz baze podataka, skladišta podataka i drugih izvora koja je oblikovana kako bi omogućila analize prošlih trendova i prošlih iskustava za daljnje strateške odluke.

Data Mining (skraćeno **DM**) je analiza velikoga broja podataka za veze koje nisu prethodno bile otkrivene pomoću zasebnih metoda.

Na primjer DM analiza podataka može otkriti prodaji da korisnici koji kupuju skupu hranu, kupuju i skupa pića; koji kupuju dječju hranu, kupuju i dječje igračke; koji kupuju pivo, kupuju odmah i grickalice; koji kupuju keramičke pločice, kasnije kupuju predmete za kućanstvo i sl. Data mining je otkrivanje znanja.

2.2.1 Projektiranje informacijskoga sustava

Informacijski inženjering (engl. *Information Engineering*, skraćeno **IE**) je disciplina koja se bavi svim aspektima razvoja informacijskoga sustava: analizom poslovnih procesa (ne i njihovim inženjeringom), oblikovanjem informacijskih sustava, softverskim inženjeringom, uvođenjem i održavanjem informacijskih sustava. IE obuhvaća sustavnu primjenu prikladnoga skupa metoda i alata u procesu razvoja informacijskoga sustava.

Projekt informacijskoga sustava (engl. *design of information systems*) u biti je skup nacrt, sličan projektima u građevini, brodogradnji i drugim strukama koji logički definira sadržaj i strukturu informacijskoga sustava uz pomoć niza modela informacijskog inženjeringa.

Prvi je korak u projektiranju informacijskoga sustava analiza. Analiza predstavlja spoznaju dijelova sustava kao funkcionalne cjeline. Nakon analize slijedi oblikovanje budućega informacijskoga sustava koji će zadovoljiti potrebe i želje korisnika ustanovljene tijekom analize. Analiza i oblikovanje logičke su faze u kojima se komunicira, promišlja i dokumentira informacijski sustav.

Nakon analize i dizajna informacijskoga sustava potrebno je definirati relacijsku shemu baze podataka i napraviti dizajn arhitekture softvera. Navedeno ulazi u projektnu dokumentaciju informacijskoga sustava. Dokumentacija će odgovoriti na pitanja: što i zašto, kako i kada, tko, gdje i s čim.

Projektiranje informacijskih sustava (engl. *information systems designing*) je nalaženje modela informacijskoga sustava. Model informacijskoga sustava sastoji se od najmanje triju modela: modela procesa, modela podataka i modela resursa. U ovoj knjizi detaljno se definira model podataka.

Model procesa (engl. *process model*) prikazuje skup procesa koji prihvaćaju ulaze u sustav, mijenjaju stanje sustava i pomoću kojih se formiraju izlazi iz sustava. Model procesa je skup poslova nad skupovima podataka. Procesni na modelu skupovi su poslovi koji stvaraju ili koriste informacije za svoje funkcioniranje.

Model resursa (engl. *resource model*) specificira tehnološku osnovicu. On prikazuje „procesore“ (kadrove, organizacijske jedinice, IT (opremu) u pogledu njihovih kapaciteta i dinamike korištenja tih kapaciteta) koji omogućuju smještanje i dinamiku podataka i procesa sustava. U modelu resursa skriveni su svi aspekti različiti od podataka i procesa.

Baza podataka (engl. *Database*) predstavlja neredundantni skup podataka o stanju sustava strukturiran na način opisan u shemi baze podataka. Baza podataka je skup podataka o svim pojavljivanjima entiteta, svih veza i svih njihovih atributa opisanih u shemi baze podataka.

Aplikacija (engl. *application*)⁹ je (Kiš, 2002.)

1. kraći naziv za namjenski program; program posebne namjene
2. područje ili vrsta problema na koje se primjenjuju tehnike obrade podataka.

Sinonimni su termini za *aplikaciju*: aplikativni program, namjenski program, program s posebnom namjenom, aplikativni softver, aplikacijska programska oprema, namjenska programska oprema, poslovna aplikacija.

Aplikacija je računalni program stvoren kako bi obavljao neku zadaću (Anić, 2002). Aplikacija je programski proizvod i baza podataka koji obavljaju određenu poslovnu funkciju ili njezin kohezivni dio.

Proces izrade softvera (softverski proces, faze softverske proizvodnje, engl. *software process*) je definiranje skupina aktivnosti i redoslijeda izvođenja aktivnosti tako da se redom izrađuju dokumentacija i softverski moduli sve do konačnoga integriranog softverskog proizvoda.

Analiza informacijskih sustava je proces u kome analitičar proučava postojeći poslovni sustav i njegov informacijski sustav kako bi omogućio kreiranje modela novoga stanja procesa sustava i novi informacijski sustav. Rezultati analize su: softverski zahtjevi, potrebni podaci, izlazne informacije, zadaci ljudima, logika procedura.

Analitičar (engl. *analyst*) je osoba koja provodi analizu, tj. traži veze, uzroke i posljedice te izvodi zaključke pomoću rastavljanja cjeline na sastavne elemente (Anić, 2002.).

Korisnici (engl. *users*) su radnici u poslovnoj organizaciji i njezinoj okolini koji imaju koristi od sustava (bilo da unose podatke u bazu podataka ili imaju mogućnost pregleda izvještaja). Pod korisnicima podrazumijevamo radnike u sustavu, odnosno izvršitelje, na bilo kojoj hijerarhijskoj razini poslovne organizacije.

Proces (engl. *process*) je skup povezanih aktivnosti i odluka, preko kojih elementi sustava ostvaruju dijelove cilja svoga postojanja, a za njihovo izvršenje potrebni su određeni resursi i određeno vrijeme.

⁹ Pod pojmom aplikacija podrazumijeva se pojam softverska aplikacija.

Procesi se sastoje od potprocesa koji se na najnižoj razini raščlanjivanja poklapaju s aktivnostima.

Aktivnosti (engl. *activities*) su jednostavni poslovi od kojih se sastoje procesi i funkcije.

Funkcija (engl. *function*) je apstraktan naziv za skupinu sličnih poslova koji ostvaruju jedan parcijalni interni cilj jednoga organizacijskog dijela poslovne organizacije.

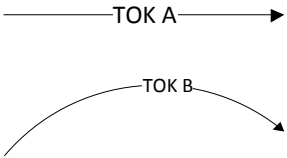
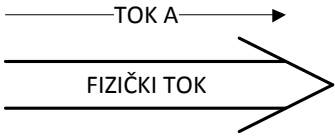

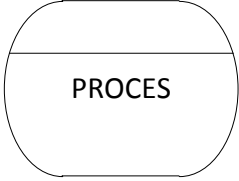

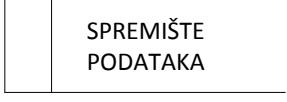

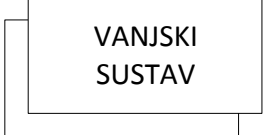
Cilj je analize sustava istražiti procese koji prihvataju ulaz, istražiti što se sve s ulazom događa dok se on ne transformira u izlaz i sustav ne postigne svoj cilj. Transformaciju ulaza u izlaz u okviru sustava obavljaju procesi.

Jedna od poznatih metoda (često u literaturi nazivana tehnika) analize sustava i izrade modela procesa je metoda pod imenom **Dijagram toka podataka** (skraćeno DTP, engl. *Dataflow Diagram*, skraćeno DFD). DTP je grafičko sredstvo za modeliranje i prezentaciju procesa sustava. Mnoge metode za modeliranje procesa koriste jednu od varijanti DTP-a za prikaz modela procesa.

Odlike DTP-a su:

- DTP je grafički prikaz procesa
- korisnik i analitičar zajedno dolaze do modela procesa
- precizno se definiraju zahtjevi korisnika
- DTP je jezik za komunikaciju korisnika i analitičara
- malen je broj različitih koncepata na DTP-u (četiri osnovna simbola).

Skup osnovnih koncepata za gradnju dijagrama toka podataka (Yourdon, 1979.) prikazan je na slici 2.6.

KONCEPT	SIMBOL po DeMarcu, Yourdonu	SIMBOL po Ganeu i Sarsonici
TOK PODATAKA predstavlja se vektorom ili usmjerenim lukom		
PROCES (funkcija) predstavlja se ovalom, elipsom, krugom i slično		
SPREMIŠTE (skladište) podataka predstavlja se dvjema paralelnim crtama		
VANJSKI SUSTAV (izvoriste ili odredište, granični entitet) predstavlja se pravokutnikom		

Slika 2.6 Koncepti DTP-a

Dizajn informacijskoga sustava i metode za modeliranje podataka opisane su detaljnije u drugim poglavljima.

2.2.2 Proizvodnja softvera

Definirajmo neke važnije pojmove vezane uz proizvodnju softvera.

Implementacija (engl. *implementation*) je rad na provođenju čega; provedba, izvršenje, primjena, ostvarenje (Anić, 2002.).

Implementacija informacijskoga sustava je skup aktivnosti koji modele iz projekta informacijskoga sustava pretvara u dijelove strukture budućega informacijskog sustava te novonastale dijelove uvodi u postojeći sustav, mijenjajući postojeći informacijski sustav.

Softversko inženjerstvo (programsko inženjerstvo, engl. *software engineering*) obuhvaća poslove kojima se oblikuje i razvija programska oprema; sastoji se od analiziranja i pobližeg opisivanja postupka koji treba programirati, razvijanja programa, pri čemu se odabire neki od standardnih pristupa, tehnika testiranja, odnosno provjeravanja programa,

pisanja dokumentacije, pokusnih izvedbenih programa, analize vremenskog vođenja itd. (Kiš, 2002.).

Softverski proces (engl. *software process*) je skup aktivnosti, metoda, postupaka i transformacija koji se koriste za oblikovanje i održavanje softvera i pratećih produkata kao što su projektni planovi, kodovi, testovi slučaja i priručnici za korisnike (Kellner, 1999.).

Osnovne aktivnosti unutar softverskoga procesa su:

- specifikacija (analiza zahtjeva korisnika; utvrđuje se što softver treba raditi)
- oblikovanje (oblikovanje građe sustava, načina rada komponenti te sučelja između komponenti; projektiranje rješenja koje određuje kako će softver raditi)
- implementacija (oblikovano rješenje realizira se korištenjem raspoloživih programskih jezika i alata)
- verifikacija (provjera radi li softver prema specifikaciji)
- validacija (provjera radi li softver ono što korisnik želi) te
- održavanje (popravljanje, mijenjanje i nadograđivanje softvera u skladu s promjenama potreba korisnika).

Testiranje programa (ispitivanje) je provjera rada programa njegovim izvođenjem, uz uporabu pripremljenih ulaznih testnih podataka te analizom izlaznih podataka dobivenih kao rezultata obrade ulaznih podataka.

Pravilo je da svaki programer provodi testiranje svoga koda.

Verifikacija programa je testiranje kojim programer utvrđuje obavlja li program ono čemu je namijenjen, odnosno ostvaruje li planiranu funkciju.

Alfa-testiranje (engl. *alpha testing*) ili tzv. verifikacijsko testiranje je probna uporaba programskoga sustava koju provode interno testeri tvrtke u kojoj se izrađuje programski proizvod kako bi provjerili odgovara li softverskom projektu. Alfa-testiranje je simulacija rada programa kao dijela stvarnoga informacijskog sustava (probni podaci su realni iz poslovne organizacije, baza podataka je puna, više testera radi istodobno).

Beta-testiranje (engl. *beta testing*) ili tzv. validacijsko testiranje je korisnička probna uporaba i provjera koja podrazumijeva provjeru rada u stvarnim uvjetima pod vršnim opterećenjem mreže i punom bazom podataka tako da se mjeri stvarna brzina pristupa i vrijeme izrade pojedine funkcije. Za generički softver beta-testiranje može izvoditi vrlo velik broj korisnika.

Funkcionalno testiranje (engl. *black-box, functional testing*) je validacija koju izvodi korisnik i provjerava što program radi. Najčešće korištena metoda funkcijskoga testiranja je testiranje domene¹⁰ (engl. *domain testing, partition testing*) (Beizer, 1990.).

CASE alat (*Computer-Aided Software Engineering*) je softverski proizvod koji omogućuje automatizaciju procesa izrade softvera. CASE alat ima u sebi podršku za: dizajn, programiranje i testiranje softverskoga proizvoda. Vrsta CASE alata su i RAD (engl. Rapid Application Development) alati koji nemaju podršku dizajniranja, ali imaju podršku za automatsko programiranje i testiranje.

Pod **outsourcingom** (engl. *outsourcing*) podrazumijevamo oduzimanje poslovnih operacija (funkcija, procesa) koje nisu temeljna djelatnost organizacije vlastitim

¹⁰ **Domena** je skup svih ulaza kojima se može izvoditi neka testna jedinica.

organizacijskim dijelovima i njihovo delegiranje vanjskim specijaliziranim organizacijama za vođenje tih operacija. Organizacije kreću u *outsourcing*, ne samo u proizvodnji softvera, radi: smanjenja troškova, usmjeravanja cjelokupnoga napora tvrtke ostvarenju misije, korištenja kvalitetnih potencijala kojih u samoj organizaciji nedostaje (ljudi, kapitala, tehnologije, resursa).

2.2.3 Uvođenje u primjenu

Gotov programski proizvod uvodi se u sustav, odnosno primjenjuje u svakodnevnome radu.

Izrađen, novi programski proizvod, treba zauzeti mjesto u poslovnome sustavu i zamijeniti postojeću „staru“ aplikaciju. Proces zamjene stare aplikacije novom aplikacijom nazivamo **uvođenje**. Moguće je da sustav do tada nije informatiziran te da ne postoji „stara“ aplikacija, tada je uvođenje lakše. Kada se aplikacija uvede, počinje njezino dnevno operativno korištenje i tada se sustav nalazi u fazi primjene aplikacije.

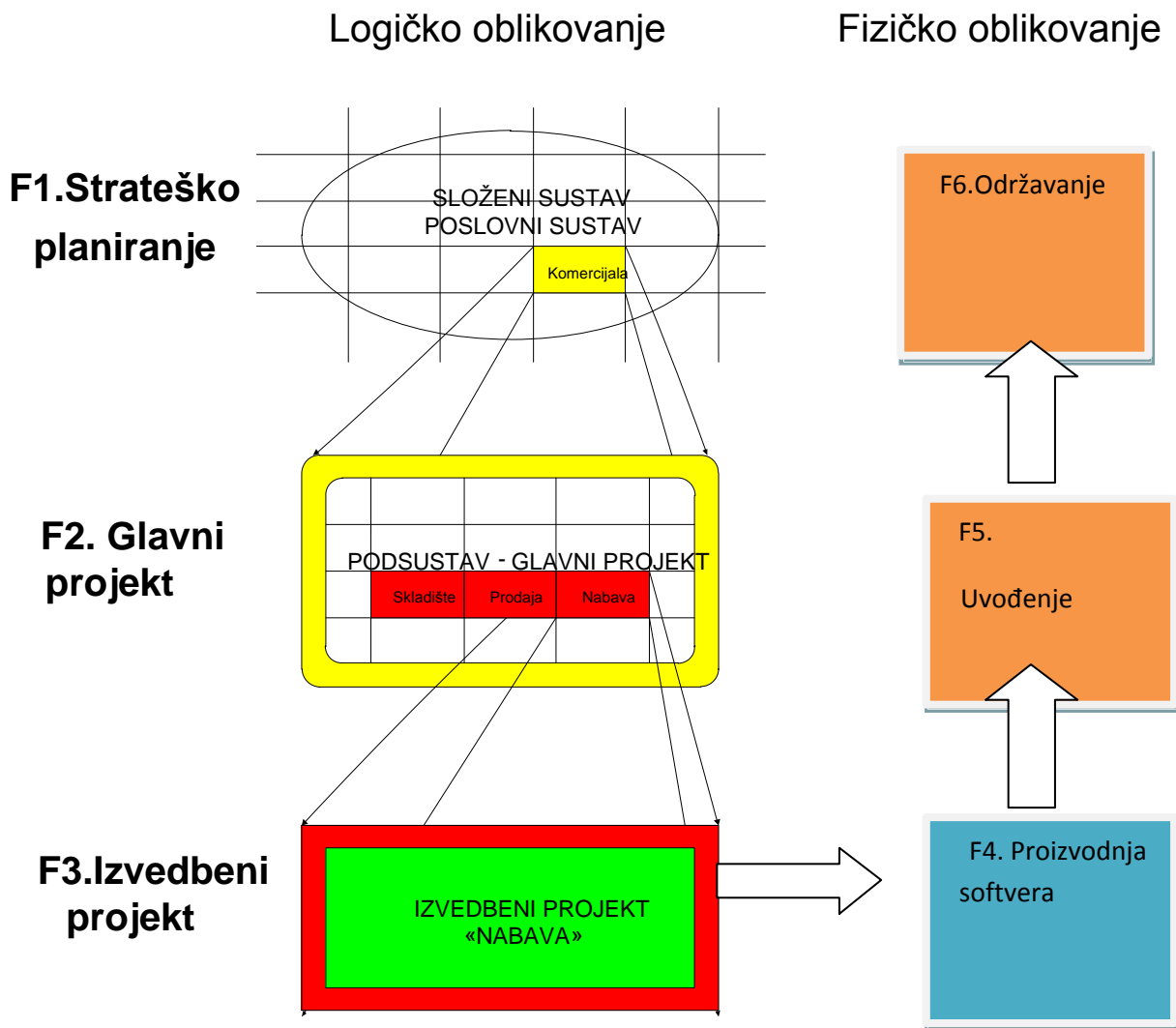
Uvođenje je razdoblje od nekoliko dana do nekoliko mjeseci. Primjena programskoga proizvoda traje više godina. Uvođenje može biti i neuspješno, ako se pokažu nepremostivi problemi. Tada se planiraju korektivne aktivnosti vezane uz probleme: izmjena softvera, dorada baze podataka, dodatna obuka, i dr. Po uklanjanju uzroka odgađanja uvođenja ponovo se pristupa uvođenju. To se može ponoviti više puta. Moguće je i odustati od uvođenja.

Aktivnosti faze uvođenja su prikazane u tablici 2.7. u okviru Specijalizirane metodologije MIRIS.

2.3 Specijalizirana metodologija MIRIS

Specijalizirana metodologija MIRIS (skraćeno od hrvatskog **M**etodologija za **R**azvoj **I**nformacijskog **S**ustava) je skup metoda i uputa čiji je ukupni cilj projektirati i izgraditi informacijski sustav. Poznata je i kao metodika MIRIS. Ta specijalizirana metodologija propisuje faze razvoja i aktivnosti pojedine faze do potrebne razine detalja informacijskih sustava.

Faze životnoga ciklusa grupirane su u dvije skupine faza: logičko oblikovanje (projektiranje informacijskoga sustava) i fizičko oblikovanje (izgradnja informacijskoga sustava). Svaka skupina faza ima tri faze (vidi sl. 2.7). Faze se dalje dijele u aktivnosti.



Slika 2.7 Raščlana sustava i faze specijalizirane metodologije MIRIS

Faze i aktivnosti životnoga ciklusa razvoja informacijskoga sustava prema metodologiji MIRIS prikazane su u sljedećim tablicama.

Tablica 2.3 Strateško planiranje

Faza 1: STRATEŠKO PLANIRANJE INFORMACIJSKOGA 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

Tablica 2.4 Glavni projekt

Faza 2: GLAVNI PROJEKT (GP)

2.1 PZ: Izrada projektnoga 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 glavnoga projekta

Tablica 2.5 Izvedbeni projekt

Faza 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 programskoga proizvoda (APP)
3.4 Operacije IP: Projektiranje operacija nad shemom BP

Tablica 2.6 Proizvodnja softvera

Faza 4: PROIZVODNJA SOFTVERA (PS)
4.1: PLANIRANJE PROIZVODNJE
<ul style="list-style-type: none"> • Planiranje aktivnosti proizvodnje SW • Određivanje izvršitelja za pojedine zadatke i određivanje rokova • Određivanje i kreiranje produkcijske, testne i razvojne okoline
4.2: OBLIKOVANJE BAZA PODATAKA
<ul style="list-style-type: none"> • Prevođenje logičkoga modela podataka u fizički model sheme baze podataka • Kreiranje razvojne okoline za svakoga pojedinog programera • Punjenje sheme baze podataka u razvojnoj okolini iz postojeće produkcijske BP • Dodavanje novih koncepata iz modela podataka u razvojnu shemu BP (tip entiteta i dr.) • Kreacija razvojne baze podataka • Inicijalno punjenje testne baze podataka
4.3: RAZVOJ PROGRAMSKOGA PROIZVODA
<ul style="list-style-type: none"> • Izrada glavnog izbornika (aplikacijskoga stabla) ili dorada već postojećega • Izrada ekrana za pregled redaka svake tablice po jednom ili više ključeva • Izrada ekrana za operacije nad jednim retkom tablice (unos, izmjena, brisanje i pregled) • Izrada programskih modula različitih vrsta i namjena: obračuna, procedura, funkcija kontrola, <i>look-upova</i> nad tablicama (s prvim testiranjem modula) • Izrada izvještaja (s prvim testiranjem modula)
4.4: TESTIRANJE U TESTNOJ OKOLINI
<ul style="list-style-type: none"> • Prijenos razvijenih programskih modula u testno okruženje • Spajanje novih modula s postojećim • <i>Back-up</i> verzija softvera • Testiranje prototipa softvera nad testnom bazom podataka • Ažuriranje planova proizvodnje softvera • Izvođenje prema potrebi aktivnosti iz ranijih skupina aktivnosti i ponovno testiranje
4.5: TESTIRANJE I ISPRAVLJANJE U RADNOJ OKOLINI
<ul style="list-style-type: none"> • Prijenos razvijenih programskih modula u radno okruženje • Spajanje novih modula s postojećima • <i>Back-up</i> verzija softvera • Punjenje baze podataka

• Testiranje prototipa softvera nad produkcijskom bazom podataka koje obavlja programer
• Ažuriranje planova proizvodnje softvera
• Izvođenje prema potrebi aktivnosti iz ranijih skupina aktivnosti i ponovno testiranje
4.6: KORISNIČKO TESTIRANJE
• Presentacija softvera korisniku
• Testiranje od strane korisnika
• Izrada popisa primjedbi korisnika
• Ažuriranje planova proizvodnje softvera
• Izvođenje prema potrebi aktivnosti iz ranijih skupina aktivnosti i ponovno testiranje
• Izrada zapisnika o testiranju i prihvaćanju faze uvođenja

Tablica 2.7 Uvođenje

Faza 5: UVOĐENJE (UVO)
5.1 Instalacija gotovoga softvera na produkcijsko okruženje (kod korisnika)
5.2 Izrada uputa
5.3 Presentacija gotovoga softvera
5.4 Obuka
5.5 Završne konverzije
5.6 Završno testiranje
5.7 Početak primjene nove aplikacije
5.8 Uspostava novoga sustava i potpisivanje primopredajnoga zapisnika

Tablica 2.8 Primjena i održavanje

Faza 6: PRIMJENA I ODRŽAVANJE (ODR)
6.1 Podešavanje novoga aplikacijskoga sustava
6.2 Izvješće o procjeni novoga projekta
6.3 Raspodjela odgovornosti korisnika i programera
6.4 Korištenje aplikacijskoga sustava
6.5 Korisničko postavljanje zahtjeva za izmjenama

Cjelokupni posao započinje izradom strateškoga plana. Potom se u fazi Glavnoga projekta analizira poslovanje i kreira model procesa.

U fazi Izvedbenoga projekta modeliraju se podaci i definira logička arhitektura programskoga proizvoda.

Faza „Proizvodnje softvera“ započinje planiranjem proizvodnje. Uzimaju se u obzir postojeći programski proizvodi i baza podataka u sustavu. Cilj je reorganizirati postojeću bazu podataka (stvoriti novu ako ništa ne postoji) i kreirati potreban novi programski proizvod, odnosno izmijenjen ili zamijeniti postojeći programski proizvod.

Gotov programski proizvod uvodi se u sustav, a po uvođenju primjenjuje se u svakodnevnome radu. Dijelovi informacijskoga sustava održavaju se po potrebi kako bi on zadovoljio potrebe korisnika.

3 RELACIJSKE BAZE PODATAKA

3.1 Uvod u relacijsku metodu

Relacijski model podataka objavljen je prvi put u Coddovu¹¹ članku (Codd, 1970.). Edgar Frank Codd (vidi sliku 3.1) rođen je 1923. u Engleskoj.

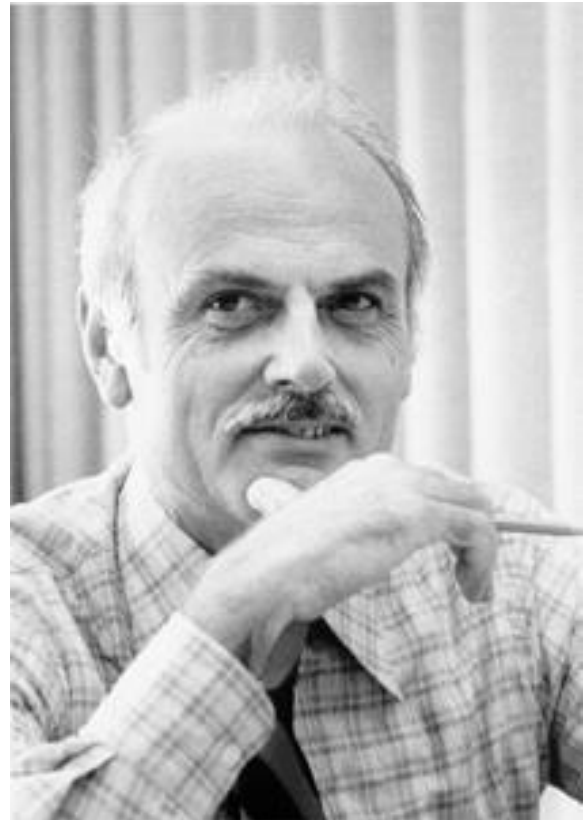
Nakon serije Coddovih radova započeo je razvoj relacijskih modela podataka i relacijskih baza podataka, a teorija modela podataka započinje bržim razvojem. Nakon razvoja teorije modela podataka uočeno je da relacijski model ima niz nedostataka. Na osnovi tih saznanja definiran je prošireni relacijski model.

U osnovi postoje dvije vrste relacijskih metoda, odnosno odgovarajućih modela: relacijski model (RM) i prošireni relacijski model (RM/T).

Cood je objavio prošireni relacijski model podataka (nazvan RM/T) nakon razvoja teorije modela podataka (Codd, 1979.) u okviru koje je razvijeno niz semantički bogatih koncepata. Ovi koncepti omogućuju na formalan način ugradnju interpretacije podataka u model.

Osnovni koncepti za gradnju strukture relacijskoga modela podataka su relacije. Relacija u relacijskome modelu podataka je isto što i relacija u matematici s tom razlikom da su relacije u relacijskome modelu podataka vremenski promjenljive.

U ovome se poglavlju predstavlja teorija relacijskoga modela, upitnih jezika i daje kratak pregled SQL-a kao standardnoga jezika upita. Radi potpunosti prikazana je neophodna teorija za razumijevanje relacijskoga modela i relacijskih baza podataka i dana osnova upitnoga jezika SQL. Kao preduvjet validaciji modela podataka zahtijeva se umijeće oblikovanja upita nad bazom podataka korištenjem SQL-a, jer ne može se očekivati korisnost modela ukoliko se ne stvori jasna predodžba o pitanju budućoj bazi podataka. Ovo znači da bi čitatelj mogao, prilikom validacije strukture modela podataka, odgovoriti može li se danim modelom odgovoriti na raspoloživa razborita korisnička pitanja, poznata i kao tipični zahtjevi korisnika, tj. može li se u principu formulirati SQL upiti (nad danom strukturom buduće baze podataka) koji bi dali tražene odgovore. Na ovo je i u procesu oblikovanja baze podataka važno odgovoriti rano, tj. na osnovi modela podataka kako se ne bi gubilo vrijeme razvijajući neodgovarajuće baze podataka. SQL je predstavljen sažeto i s osnovnom sintaksom kako bi se prenijele ideje i mogućnosti upitnoga jezika što je zadovoljavajuća razina detalja u skladu s ciljem koji ovo poglavlje sadrži u okviru cjeline.



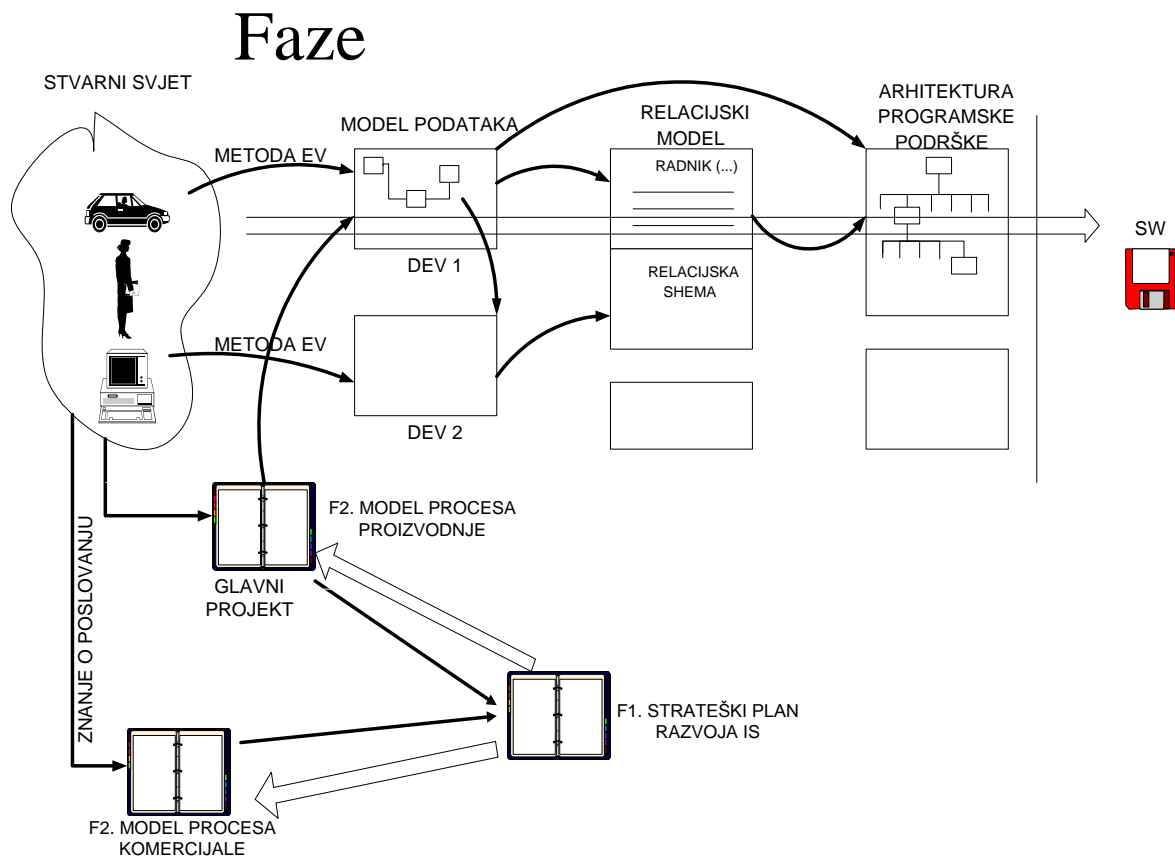
Slika 3.1 Edgar Frank Codd

¹¹ 1970. godine objavio je rad "A Relational Model of Data for Large Shared Data Banks" u kojem je predstavljena relacijska metoda.

4 MODELI PODATAKA

4.1 Problemi, strategije i kriteriji modeliranja podataka

Ovome je poglavlju za cilj izložiti osnovne pojmove o modelu podataka. U poglavlju 2.3 MIRIS prikazane su osnovne aktivnosti razvoja informacijskoga sustava čiji dio je modeliranje podataka, a na slici 4.1 prikazan je položaj metoda za modeliranje podataka u cjelokupnome procesu razvoja informacijskoga sustava.



Slika 4.1 Položaj modela podataka u odnosu na ostale metode

Model podataka je shvaćan i analiziran kao skup entiteta i operacija nad njima.

Jedan od većih problema razvoja informacijskoga sustava je nepostojanje jedinstvene baze podataka i jedinstvenoga modela podataka, odnosno postavlja se pitanje kako razriješiti probleme razvoja i postići integralnost informacijskoga sustava? To je moguće uz pomoć triju strategija:

- *Uspostavlja se jedinstveni šifrirski sustav*
- *Kontinuirano se unapređuju i koriste CASE alati*

- *Koriste se semantički bogate metode za modeliranje podataka. Razvoj informacijskoga sustava oslanja se svim programskim proizvodom na jedinstveni model podataka.*

Semantički¹² bogate metode (metode mrežne prezentacije znanja) dovode do znanjem bogatih modela podataka, točnije do modela informacija.

Prve od tih metoda bile su semantičke mreže i danas postoje u većemu broju različitih oblika. **Semantička mreža** predstavlja znanje sadržano u pojmovima skupova objekta (čvorova, engl. *node*) i binarnih veza među čvorovima (engl. *direct labeled edges*). Po definiciji **baza znanja** je skup objekata i skup odnosa među objektima. Izmjena baze znanja predstavlja umetanje ili brisanje objekata i veza među njima.

Semantičke mreže se razvijaju od 1968. godine (Brodie, 1984.). Najvažniji intelektualni alati semantičkih mreža su organizacijske osi, odnosno apstrakcije prema kojima se znanje može rasporediti u model. Uočeni su osnovni temeljni aksiomi (pristupi, misli, metode) za organizaciju znanja i to: klasifikacija, agregacija, generalizacija i različite vrste veza (asocijacija) te operacije nad definiranim konceptima.

Danas postoje različite metode za modeliranje podataka. Ideje baštine od semantičkih mreža te su i same manje ili više semantički bogate. S obzirom na količinu znanja koje se može ugraditi u model, odnosno s obzirom na koncepte koji omogućavaju semantički bogatiju prezentaciju podataka, modeli podataka mogu se dijeliti u generacije.

Modeli podataka (tipovi podataka) I. generacije su:

Svaki programski jezik je zaseban model podataka, a podaci se modeliraju preko koncepta kojima dani jezik raspolaže, kao što su na primjer: pointer, integer, real, matrica, stack i dr.

Modeli podataka II. generacije su:

- Funkcionalni model podataka
- Hijerarhijski model podataka
- Mrežni model podataka (Bachman, 1969.)
- Klasični relacijski model podataka (RM) (Codd, 1970.)
- Warnierovi dijagrami

Ovi modeli sadrže koncepte za prezentaciju podataka kao: stablo, set, relacija, i dr. Oni nisu dovoljno semantički bogati i imaju samo "atomsku" semantiku.

¹² Semantika – 1. *lingv.* Grana lingvistike koja proučava pojedine riječi, njihove oblike i grupacije kao nositelje određena značenja te kao sredstva za označavanje predmeta, pojava i odnosa u materijalnom i duhovnom svijetu; utvrđuje glavne procese koji vode do promjena u značenju riječi 2. *log. a.* opća oznaka za teoriju istinitosti logičkih rečenica i nizove **b. mat.** dio formalnoga sustava koji svakom iskazanom elementu logičkoga računa pridružuje jedan od znakova koji pripadaju svijetu označenih (signifikanta) (Anić, 2003.)

Modeli podataka III. generacije su:

- Model entiteti-veze (autor: Chen)
- Prošireni relacijski model – RM/T (autor: Codd)
- SDM-IBM (autor: IBM)
- Model podataka semantičkih mreža (autor: Roussopoulos i Mylopoulos)
- Semantički model podataka - SDM (autor: Hammer i McLeod)
- Petrijeve mreže (autor: Reti di Petri)
- Semantic Association Model - SAM (autor: Stanlay)
- D-Graph Model (autor: Weber)
- Palmer (autor: Palmer)
- Diam II (autor: Senko)

Ovi modeli imaju koncepte generalizacije i agregacije te, pored "atomske", omogućuju ugradnju "molekularne" semantike u reprezentaciju. Dobro modeliraju poslovni sustav, korisnički su razumljivi, ali danas uglavnom bez razvijene softverske podrške.

Premda su ovi modeli različiti u detaljima, skoro svi posjeduju koncepte pomoću kojih je moguće obuhvatiti više informacijskih zahtjeva i bolje opisati entitete poslovnoga sustava.

4.2 Osnovni pojmovi o modeliranju podataka

Uvest ćemo neke osnovne pojmove vezane uz modeliranje podataka, uz minimalno ponavljanje ranije definiranih pojmova u knjizi.

Metoda za modeliranje podataka (engl. *Data modelling method*) je definirani postupak nalaženja i prikazivanja informacijskih objekata i njihovog međusobnog odnosa.

Primjenom procesa modeliranja podataka zasnovanog na nekoj metodi dobiva se kao rezultat model podataka. Modeliranje je proces razvoja modela. Model nastaje procesom apstrakcije u kome se prvo biraju relevantni koncepti koje reprezentacija treba sadržati, a zatim se svakome konceptu pridružuju relevantne osobine (atributi) koje se žele prikazati u okviru modela.

Model podataka sustava je pojednostavljena reprezentacija o relevantnim karakteristikama sustava preko skupa entiteta (objekata), veza među entitetima i atributa entiteta i agregacija entiteta. Model podataka treba istovremeno sadržavati i same podatke i njihovu interpretaciju, odnosno treba predstavljati strukturirani skup informacija o prošlosti i sadašnjosti sustava.

Osnovni zadatak istraživanja u području modeliranja podataka je naći koncepte u okviru metode pomoću kojih će se izgraditi model koji predstavlja naše znanje o sustavu.

Koncepti su ideje (simboli, znakovi, intelektualno sredstvo, aksiomi, zamisli, uzorci) koji predstavljaju klasu odabranih pojmova iz poslovnoga sustava. Neki koncepti imaju grafički prikaz u obliku simbola. Dobiveni je model podataka, sa stajališta statičke strukture, skup povezanih koncepata (slika) koji predstavlja cijeli sustav.

Općenito se na području modela podataka došlo do saznanja da se svaki model podataka sastoji iz triju dijelova i to:

- **Strukture** (engl. *structure*)
- **Ograničenja** (engl. *constraints*)
- **Operatora** (engl. *operators*)

Ponekad se struktura i ograničenja nazivaju strukturnom komponentom modela podataka jer zajedno omogućuju opis statičkih svojstava poslovnoga sustava u model. Dinamika u modelu podataka predstavlja se pomoću operatora.

Pomoću **strukture** podataka, podaci o pojmovima poslovnoga sustava spajaju se u međusobno povezane grupe i tako se znanje o grupiranju i interakciji pojmova iz poslovnoga sustava prenosi u model podataka. Na primjer, grupa podataka će biti recepti, druga grupa podataka će biti pacijenti, a treća lijekovi. Lijekove ćemo povezati s receptima, ali ne i s pacijentima.

Pomoću **ograničenja** na podatke u strukturi podataka unosi se dalje znanje o interakciji koncepata iz poslovnoga sustava. Na primjer, što je zabranjeno, ili u kojoj količini je dopušteno, pa možemo ograničiti da jedan recept može propisati samo jedan lijek, ali da pacijent može dobiti više recepata.

Pomoću **operatora** nad podacima, uvodi se skup operacija nad strukturom podataka, koje omogućuju dinamiku podataka o konceptima analogno dinamici stvari u poslovnome sustavu. Na primjer, operacija bi bila upis novoga recepta ili složena operacija pretrage pacijenata koji su primili ove godine bilo koji antibiotski lijek tvrtke „Pliva“.

Model podataka (engl. *Data model*) je skup međusobno povezanih podataka koji opisuju entitete, veze i attribute poslovnoga sustava. On je reprezentacija skupa podataka koji se modelom interpretiraju preko aspekta: strukture, ograničenja i operatora.

Struktura modela podataka je skup entiteta i veza koji interpretira podatke klasificirajući ih u tipove entiteta i tipove veza među tipovima entiteta, zajedno s relevantnim svojstvima tipova entiteta.

Ograničenja modela podataka su koncepti strukture modela, a omogućuju daljnju interpretaciju podataka razdvajajući dopuštenu od zabranjenih stanja skupa podataka preko: dopuštenih podataka u okviru jednoga tipa entiteta, dopuštenih vrijednosti podataka pojedinoga svojstva tipa entiteta, dopuštenih povezivanja među tipovima entiteta.

Operatori modela podataka čine skup koncepata koji omogućuju interpretaciju dinamičkih karakteristika skupa podataka. Koncept strukture i ograničenja reprezentira statička svojstva poslovnoga sustava, a operatori omogućuju izmjenu stanja podataka u bazi podataka u skladu s promjenom stanja u poslovnome sustavu.

Struktura modela podataka gradi se od temeljnih koncepata: entitet, veza i atribut.

Entitet (engl. *Entity*) sustava je neki njegov realni ili konceptualni element, to je neka posebnost što u poslovnome sustavu postoji i jasno se razlikuje od drugih entiteta. Na primjer: muškarci, narudžbe, organizacijske jedinice i dr.

Veza (engl. *Relationship*) je koncept koji predstavlja neku interakciju među entitetima u sustavu, odnosno predstavlja znanje o njihovoj povezanosti. Na primjer: brak je veza između muškaraca i žena.

Atribut entiteta (engl. Attribute) je neko svojstvo entiteta. Na primjer: ime radnika, datum prispjeca narudžbe, faktor uvećanja, brzina vozila, šifra općine i dr.

U modelu podataka, kao i u nizu drugih modela (matematičkome sustavu jednadžbi putanje rakete, geografskoj karti teritorija, nacrtu broda, shemi električnih strujnih krugova pojačala) koji predstavljaju neki sustav, ne opisuje se potpuni skup znanja o sustavu, već se obavlja odabir relevantnih karakteristika sustava te se u model uključuju samo relevantni podaci.

Relevantni podatak (do koga nam je stalo, od interesa) je onaj podatak za koga smo zainteresirani te ulazi u model sustava, odnosno onaj koji pri analizi podataka smatramo važnim s aspekta namjene informacijskoga sustava.

4.3 Apstrakcija podataka

Kako savladati (opisati) složene sustave? Sustavi se opisuju i predstavljaju modelom sustava. Model se gradi pomoću pravila koje propisuje metoda. Metoda je proizvod sastavljen od intelektualnih alata (konceptata) pomoću kojih se gradi model sustava. Tako na primjer metodu EV možemo zamisliti kao stolarsku radionicu u kojoj ima niza alata (pila, blanja, čekić,...). Stolar odabire pojedine fizičke alate primjenjuje ih gradeći proizvod, na primjer stolicu. Slično će projektant IS birati koncepte u metodi (intelektualne alate) i graditi model.

Intelektualni alati u metodama za modeliranje podataka zasnovani su na principu apstrakcije.

Apstrakcija (engl. *abstraction*) je kontrolirano uključivanje detalja u cjelinu, pri čemu u mislima istovremeno poima ideja detalja cjeline i veze među njima.

Apstrakcija se sastoji od dvaju kretanja različitih smjerova i to: apstrakcija k općem (skraćeno apstrakcija) i apstrakcija k pojedinačnom (skraćeno detaljizacija). Ova dva smjera skraćeno možemo zapisati kao: apstrakcija (detaljizacija).

Apstrakcija se još naziva i „skrivanje“ detalja, ili „izvlačenje“ općih karakteristika. **Apstrakcija k općem** je spoznajni proces u kome se od uočenoga pojedinačnog elementa zaključi o postojanju skupa koji sadržava više sličnih pojedinačnih elemenata. Na primjer: Na dokumentu Osobna iskaznica je u polju Ime napisan podatak „Marko“. Apstrakcija je sljedeći niz zaključaka: Marko je ime pojedinoga studenta. Ima još takvih podataka i ima još pojedinih studenata. Svi oni predstavljaju natpojam Student. Dakle skup Student je apstraktan naziv za niz pojedinačnih studenata. Ovdje je koncept Student apstraktan koncept u odnosu na osobu Marka koja je konkretna. Pored toga je korištena metoda „apstrakcije“ u ovome zaključivanju koja povezuje konkretno i pojedinačno s apstraktnim i u svjesnom umu izgrađenome pojmu Student. Valja naglasiti da je ljudskom umu to prirodan način funkcioniranja.

Inverzan postupak od apstrakcije je detaljiziranje. **Detaljiziranje** je spoznajni proces u kome se za uočeni skup pita od kojih se elemenata skup sastoji i potom uzrokuje jedan element skupa i on dovede u svjesni um te analiziraju svojstva odabranog elementa. Na primjer: Ako u modelu imamo apstraktan koncept Student koji je projektant IS ranije oblikovao, onda se možemo pitati, od kojih se detalja taj skup sastoji? Kada „uđemo“ u taj skup naći ćemo osobu imena Marko. Tada promatramo samo tu osobu i njezina ostala svojstva (adresu stanovanja, datum rođenja, ...). Korištena je metoda detaljiziranja pomoću koje smo od općega došli do pojedinačnog. Naglasimo da je naziv „detaljiziranje“ samo naziv

smjera kretanja metode apstrakcije. Također se u jednome zaključivanju i pojednava i detaljizira i to po nekoliko puta krećući se prema gore i nazad.

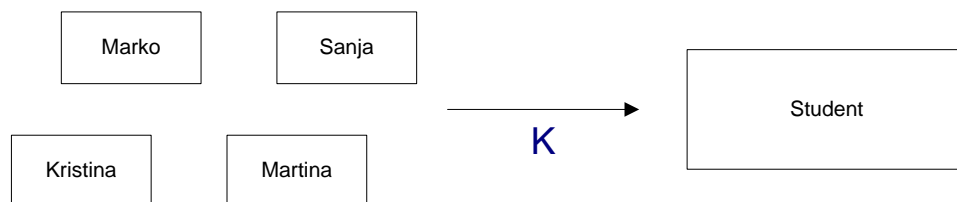
Postoji više vrsta apstrakcije. Nama su za gradnju metoda za modeliranje podataka od interesa neke osnovne vrste apstrakcije (s detaljizacijom¹³) i to:

- Klasifikacija (uzorkovanje)
- Generalizacija (specijalizacija)
- Agregacija (dekompozicija)
- Veza (veza)

Pored apstrakcija postoje i druge ideje, principi i definicije od kojih se grade metode.

4.3.1 Klasifikacija

Klasifikacija (ili tipizacija) (engl. *classification*) je apstrakcija u kojoj se skup sličnih objekata predstavlja jednom klasom objekata.

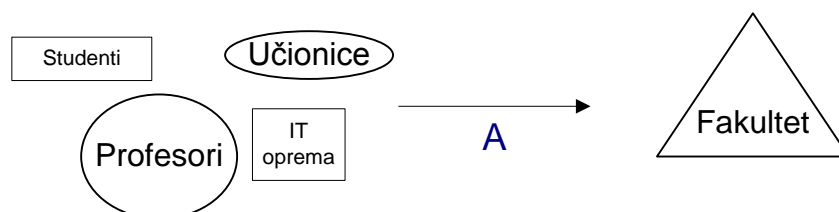


Slika 4.2 Klasifikacijska apstrakcija

Slični objekti su oni objekti koji imaju iste atribute, koji mogu stupiti u iste veze s drugim objektima i na koje se mogu primijeniti iste operacije.

4.3.2 Agregacija

Agregacija (engl. *aggregation*) je apstrakcija gdje se skup tipova entiteta i njihovih veza (te svojstava entiteta i veza) predstavlja novim izvedenim tipom entiteta.

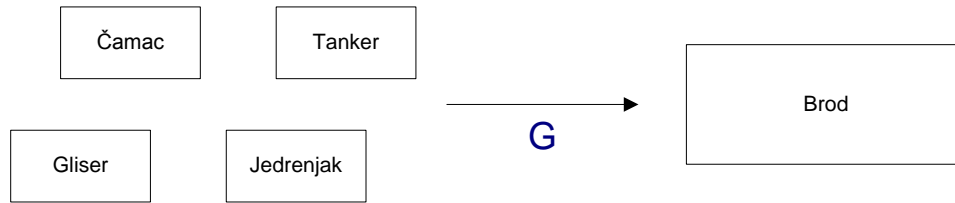


Slika 4.3 Agregacijska apstrakcija

¹³ podrazumijeva se da svaka vrsta apstrakcije ima oba smjera

4.3.3 Generalizacija

Generalizacija (engl. *generalisation*) je apstrakcija gdje se skup djelomično sličnih tipova entiteta tretira (predstavlja) kao novi izvedeni tip entiteta na višoj razini, općenitiji, generički tip, **nadtip**.



Slika 4.4 Generalizacijska apstrakcija

Djelomično slični tipovi entiteta su oni koji imaju jedan broj istih (zajedničkih) atributa, tipova veza s drugim entitetima i operacija.

Agregirani i generalizirani tip entiteta mogu se dalje agregirati i generalizirati u nove tipove entiteta.

5 METODA ENTITETI - VEZE

5.1 Uvod u metodu ENTITETI - VEZE

Metoda entiteti – veze poznata pod imenom model entiteti – veze (engl. *Entity-relationship model*, skraćeno ER) je prvi put objavljena u Chenovu članku (Chen 1976.). Metoda ER je jedan od prvih semantički bogatijih metoda za modeliranje podataka. ER model je opisan u većini knjiga koje se bave razvojem informacijskih sustava i bazama podataka i na gotovo svim svjetskim sveučilištima se uči ER modeliranje, a većina praktičara oblikovanja baze podataka u informacijskim središtima koristi metodu ER.

Dr. Peter Pin-Shan Chen je američki znanstvenik koji se bavi informatikom i profesor je informatike na fakultetu Louisiana State University (wikipedija, 2010.). Poznat je po tome što je razvio metodu entiteti – veze.

Peter Chen je rođen na Tajvanu gdje je završio studij elektrotehnike. Dobiva stipendiju za Harvard i odlazi na postdiplomski studij računarstva. U Las Vegasu je 1975. godine na “1st International Conference on Very Large Databases” predstavio prvi rad na temu ER metode. Ovaj rad smatra se jednim od najutjecajnijih radova u informatičkoj povijesti.

Reagirajući na Chenov rad Codd (utemeljitelj relacijske metode) šalje dugo pismo uredniku ACM Transaction on Database Systems-a i kritizira velikim zamjerkama na ER metodu. Codd je 1979. godine predložio novu verziju relacijskoga modela u koju je uveo neke koncepte preuzete iz ER modela. Tijekom devedesetih godina su Codd i “Data Consulting Group” nekoliko puta pozvali Chena da, uz Codd, bude glavni govornik na njihovim simpozijima u Londonu što dokazuje opću prihvaćenost ER modela. Peter Chen je redovito glavni govornik na raznim međunarodnim konferencijama.

Petera Chena su mnogi pitali kako je došao na ideju za ER model. Promišljajući o tome, došao je do zaključka da je njegovo kinesko podrijetlo imalo utjecaja. I kineski znakovi i ER model pokušavaju modelirati svijet, koristeći grafičke simbole za predstavljanje entiteta iz stvarnoga svijeta.



Slika 5.1 Peter Pin-Shan Chen

Metoda entiteti - veze (skraćeno **EV**) je grafički prikaz međusobno povezanih grupa podataka promatranoga sustava. EV je semantički bogata metoda za modeliranje podataka jer raspolaze ljudski bliskim konceptima. EV se odlikuje prirodnošću opisa a njezini koncepti su bliski korisniku, pa je shema modela podataka laka za razumijevanje i komunikaciju korisnika i projektanta.

Kao praktično sredstvo EV model podataka je uskoro dobio niz komercijalnih softverskih produkata DBMS (*Database Management System*) (Chaing i Barger, 1980.; Iossiphidis, 1980.; Munz, 1980.; Esculier i Glorieux, 1980.; Zhang i Mendelzon, 1983.; Nakata i Yamazaki, 1983.; Ford, 1985.; Goldberg, 1986.; Junet, 1987., i drugi).

Razvojem EV-a nastalo je niz varijanti polazne metode i ona je postala jedna od najčešće korištenih metoda. Tako se EV koristi u specijaliziranim metodologijama: CASE*Method, MIRIS, SSADM, IEM, i dr.

Za modeliranje podataka postoje i često korištene standardizirane notacije kao što su ISO UML (class diagrams) i IEEE IDEF1x.

REFERENCES

- (Alagić, 1984.) S. Alagić, "Relacijske baze podataka", Svjetlost, Sarajevo, 1984.
- (Anić, 2003.) V. Anić, „Veliki rječnik hrvatskoga jezika“, Novi liber, Zagreb, 2003.
- (Anić, Goldstein, 1999.) V. Anić, I. Goldstein, I. "Rječnik stranih riječi", Novi Liber, Zagreb, 1999.
- (Anić, Rončević, Cikota, Goldstein, Jojić, Matasović, Pranjković, 2002.) V. Anić, D. B. Rončević, LJ. Cikota, I. Goldstein, LJ. Jojić, R. Matasović, I. Pranjković, "Hrvatski enciklopedijski rječnik", Novi Liber, Zagreb, 2002.
- (Atzeni, 1983.) P. Atzeni i E. Carboni, "INCOD (A System for Interactive Conceptual Design) Revisited After the Implementation of a Prototype", vidi (Davis, 1983.).
- (Avison, 1995.) D.E. Avison and G. Fitzgerald, "Information System Development: Methodologies, Techniques and Tools", McGraw-Hill, London, 1995.
- (Bachman, 1969.) C. W. Bachman, "Data Structure Diagrams", Database 1, 2, pp. 4-10, 1969.
- (Barker, 1990.) Richard Barker, "CASE*Method Entity Relationship Modelling - ORACLE", Addison-Wesley P.C., Wokingham, England, 1990.
- (Beizer, 1990.) B. Beizer, "Software Testing Techinques"; Boston, Inetrnational Thompson Computer Press, 1990.
- (Beizer, 1995.) B. Beizer, "Black Box Testing", New York, Yohn Wiley & Sons, Inc., 1995.
- (Blaha, 2010.) M. Blaha, "Patterns of Data Modeling", CRC Press, NY, 2010.
- (Boggs, 2003.) W. Boggs, M. Boggs, "UML sa Rational Rose-om 2002", Kompjuter Biblioteka, Čačak, 2003., p.736
- (Brodie, 1984.) M. L. Brodie, J. Mylopoulos, J. W. Schmidt, "On Conceptual Modelling", Springer-Verlag, NY, 1984.
- (CARNet, 2010.) Hrvatska akademska i istraživačka mreža CARNet, tvrtka Pro Leksis d.o.o., (2010.), *Proleksis enciklopedija - prva hrvatska opća i nacionalna online enciklopedija*: <http://enciklopedija.carnet.hr/login.aspx>
- (Chaing, 1980.) T. C. Chaing, R. F. Barger, "A Database Management System With an E-R Conceptual Model", 1980, vidi (Chen, 1980).
- (Chan, 1980.) E.P.F. Chan i F.H. Lochovsky, "A Graphical Database Design Aid Using the Entity-Relationship Diagram", (vidi Chen, 1979.), str. 303-318.
- (Chen, 1976.) P. P. Chen, "The Entity-Relationship Model-Towards a Unified View of Data", ACM TODS, Vol.1, No.1, 1976.
- (Chen, 1980.) P. P. Chen, "Entity-Relationship Approach to System Analysis and Design", North-Holland, Amsterdam, 1980.
- (Chen, 1983.) P.P. Chen (Editor), "Entity-Relationship Approach to Software Engineering, Anaheim, North-Holland, Amsterdam, the Netherlands, 1983.
- (Chen, 2001.) I. J. Chen, "Planning for ERP systems: analysis and future trend", Bussines Process Management Journal, 7, pp.374-386, 2001.
- (Coad, 1991.) Coad P., Yourdon E., "Object-Oriented Analysis", Yourdon Press, New Jersey, 1991.
- (Database eLearning, 2010.) <http://db.grussell.org/section002.html>

- (Davis, 1983.) C.G. Davis, S. Jajodia, P. A-B. Ng. i R.T. Yeh (Editors), "Entity-Relationship Approach to Software Engineering", North-Holland, Amsterdam, the Netherlands, 1983.
- (ER, 1997.) Conceptual Modeling – ER '97, Proceedings of 16th International Conference on Conceptual Modeling, Ed. Embley, D. W., Goldstein R. C., Los Angeles, 1997., p. 1-477
- (Esculier, 1980.) C. Esculier, A. M. Glorieux, "The Sirious-Delta Distributed DBMS", vidi (Chen, 1980.), 1980.
- (Fertalj, 2006.) K. Fertalj, V. Mornar, i dr. "Komparativna analiza programske potpore informacijskim sustavima u Hrvatskoj", 5. svibnja 2006., <<http://www.zpr.fer.hr/projekti/erp/Dokumenti/ERP-HR.pdf>>, 28. siječnja 2007.
- (Ford, 1985.) R.M. Ford, "A Data Manager Using Entity- Relationship", PC Tech Journal for IBM Personal Computer Users, Vol.3, N0.10, Octobar 1985.
- (Giorgio, 1987.) B. Giorgio i E. Antonio, "Extending the Entity - Relationship Approach for Dynamic Modeling Purposes", vidi (Spaccapietra, 1987.).
- (Goldberg, 1986.) E. Goldberg, "Excelerator aids Straids adoption", Comuter World, 21.april 1986.
- (Hak, 2010.) http://www.hak.hr/media/35177/osobna_iskaznica.jpg (12.04.2010.)
- (Hammer, 1980.) Hammer, McLeod, "Semantis Data Model", TODS ACM, 1980.
- (Hernandez, 1997.) M. J. Hernandez, "Database Design for Mere Mortals", Addison-Wesley Developers Press, New York, 1997., p. 440
- (IBM, 1985.) IBM, "System Development Method (SDM) - System Design, IBM, 1985.
- (Iossiphidis, 1980.) J. Iossiphidis, "A Translator to Convert the DDL of ERM to the DDL of System 2000", vidi (Chen, 1980.).
- (Ivasic-Kos, 2010.) Ivasic-Kos, M., Poscic, P., & Pavlic, M. (2010, July). Definition of descriptors for semantic image interpretation. In Visual Information Processing (EUVIP), 2010 2nd European Workshop on (pp. 214-219). IEEE.
- (Jajodia, 1983.) S. Jajodia i dr., "On Universal and Representative Instances for Inconsistent Databases", vidi (Davis, 1983.).
- (Junet, 1987.) M. Junet, "Design and Implementation of an Extended Entity - Relationship Data Base Management System (ECRINS/86 ", vidi (Spaccapietra, 1987.).
- (Jakupović, 2010.) Jakupović, A., Pavlić, M., & Ivašić-Kos, M. (2010, September). Estimation of the Complexity of Business Sectors Covered by ERP Solutions. In Proceedings of the 21st Central European Conference on Information and Intelligent Systems.
- (Jakupović, 2010.) Jakupovic, A., Pavlic, M., & Poscic, P. (2010, September). Estimation of the Size of Business Sectors Covered by ERP Solutions. In Computing in the Global Information Technology (ICCGI), 2010 Fifth International Multi-Conference on (pp. 60-65). IEEE.
- (Kangassalo, 1991.) Hannu Kangassalo (Editor), "Entity - Relationship Approach: The Core of Conceptual Modelling, Proceedings of the Ninth International Conference on the Entity - Relationship Approach", Lausanne, Switzerland, 8.-10. October, 1990., North - Holland, Amsterdam, the Netherlands, 1991.
- (Kellner, 1999.) M. I. Kellner, R. J. Madachy, D. M. Raffo, "Software processes simulation modelling: Why? What? How? ", J.Syst.Software, 46, 91-105, 1999.
- (Kiš, 2002.) M. Kiš, "Englesko-hrvatski i hrvatsko-engleski INFORMATIČKI RJEČNIK", Naklada Ljevak, Zagreb, 2002.

- (Kiš, 1993.) M. Kiš, J. Buljan, S. Vuković, O. Anić, "Englesko-hrvatski informatički rječnik s računalnim nazivljem", Školska Knjiga, Zagreb, 1993.
- (Lazarević, 1985.) B. Lazarević, V. Jovanović, P. Dizdarević i M. Vučković, "Projektiranje informacijskih sistema", I i II dio, Naučna knjiga, Beograd, 1985.
- (Lazarević, 2008.) B. Lazarević, Z. Marjanović, N. Aničić, S. Babarogić, "Baze podataka", FON, Beograd, 2008.
- (Lien, 1980.) E. Y. Lien, "Semantic of Entity-Relationship Model", vidi (Chen, 1980.).
- (Linthicum, 2000.) D. S. Linthicum, "Enterprise Application Integration", Addison-Wesley Information Technology Series, Boston, 2000., p. 377
- (Longworth, 1989.) G. Longworth, D. Nicholls, "SSADM MANUEL Techniques and documentation", National Computing Centre, England, 1989.
- (Lusk, 1980.) E. L. Lusk i R. A. Overbeek, "A DML for Entity- Relationship Models", 1980 vidi (Chen, 1980.).
- (Markowitz, 1983. a) V. M. Markowitz i Y. Raz, "ERROL: An Entity- Relationship, Role Oriented, Query Language", vidi (Davis, 1983.).
- (Markowitz, 1983.) V. M. Markowitz i Y. Raz, "A Modified Relational Algebra and its Use in an Entity-Relationship Environment", vidi (Davis, 1983.).
- (Marti, 1983.) R. W. Marti, "Integrating Database and Program Descriptions Using on ER-Data Dictionary", vidi (Davis, 1983.).
- (Martin, 1977.) J. Martin, "Computer Data - Base Organization", Prentice - Hall, New Jersey, 1977.
- (Martin, 1985.) J. Martin, "Banka podataka za krajnjeg korisnika" Informatika i društvo, Zagreb, 1985.
- (Martin, 1985.) J. Martin, & C. McClure, "Diagramming Techniques for Analysts and Programmers". Englewood Cliffs, NJ: Prentice Hall, Inc., 1985.
- (Martin, 1985.) J. Martin, C. McClure, "Action Diagrams: Clearly Structured Program Design", Prentice Hall, Englewood Cliffs, N.J. 1985.
- (Martin, 1986.) J. Martin, "Fourth Generation Languages", Prentice Hall, Vol 1 i Vol 2, 1985.-1986.
- (Melkanoff, 1980.) M. A. Melkanoff, C. Zaniolo, "Decomposition of Relations and Synthesis of Entity Relationship Diagrams", vidi (Chen, 1980.), 1980.
- (Mogin, 2000.) P. Mogin, V. Luković, M. Govedarica, "Principi projektovanja baza podataka", Univerzitet u Novom sadu, Novi sad, 2000., p. 699
- (Munz, 1980.) R. Munz, "Design of WELLSystem", 1980., vidi (Chen, 1980.).
- (Nakata, 1983.) S. Nakata, G. Yamazaki, "A System Based on the E-R Model and its Application to Database-Oriented Tool Generator", 1983., vidi (Davis, 1983.).
- (Navathe, 1983.) S. B. Navathe, A. Cheng, "A Methodology for Database Schema Mapping from Extended Entity-Relationship Models into the Hierarchical Model", 1983., vidi (Davis, 1983.).
- (Neufeldt, 1991.) V. Neufeldt, D. B. Guralnik, "Webster's New World Dictionary", Prentice Hall, New York, 1991.
- (Ng, 1980.) P.A.Ng i J.F. Paul, "A Formal Definition of Entity- Relationship Models", (vidi Chen, 1980.).

- (Ostojić, 2010.) Josip, O., Pavlić, M., & Dovedan, Z. (2010). SYSTEM PROCEDURES AND ERRORS IN CLIENTS DATA SYNCHRONIZATION IN INSURANCE COMPANIES. *Informatologia*, 43(2), 96-104.
- (Papp, 1979.) L. Papp, "A konyvviteli információs rendszer elméleti kérdései", Tankönyvkiadó, Budapest, 1979.
- (Parent, 1985.) C. Parent i S. Spaccapietra, "An Algebra for General Entity-Relationship Model", *IEEE Transaction on Software Engineering*, Volume SE-11, Number 7, pages 634-643, July 1985.
- (Pavlić, 1986.) M. Pavlić, V. Jovanović, P. Dizdarević, Z. Jugo, "Projektiranje informacijskog sustava kalkulacija cijene broda primjenom metode SSA i E-V", III savjetovanje za informatičku djelatnost, Privredna komora Rijeka, Opatija, 1986.
- (Pavlić, 1992.) M. Pavlić, "Apstrakcija metodologije modeliranja na području projektiranja informacijskih sustava - doktorski rad", FOI, Varaždin, 1992.
- (Pavlić, 1996.) M. Pavlić, "Razvoj informacijskih sustava - projektiranje, praktična iskustva", metodologija, Znak, Zagreb, 1996.
- (Pavlić, 1983.) Pavlić, M., Zemljčić, V., Srdoč, A., Ledinko, Z., & Vučić, B. (1983). *Informacijski sustav praćenja pristizanja i ugradnje opreme platforme LABIN*.
- (Sakai, 1980.) H. Sakai, "A Unified Approach to the Logical design of a Hierarchical Data Model", 1980, vidi (Chen, 1980.).
- (Sakai, 1983.) H. Sakai, "Entity-Relationship Approach to Logical Database Design", 1983., vidi (Davis, 1983.).
- (Santos, 1980.) C.S. dos Santos, E.J. Neuhold i A.L. Furtado, "A Date Type Approach to the Entity-Relationship Model", (vidi Chan, 1980.).
- (Schuldt, 1987.) G. Schuldt, "ER - Based Access Modeling", vidi (Spaccapietra, 1987.).
- (Tardieu, 1980.) H. Tardieu, H. Heckenroth, D. Pascot i D. Nanci, "A Method, A Formalism and Tools for Database Design (Three Years of Experimental Practice)", 1980., vidi (Chen, 1980.).
- (Tsichritzis, 1982.) D.C. Tsichritzis , F.H. Lochovsky , "Data Models", Prentice Hall, 1982.
- (Ursprung, 1983.) P. Ursprung i C.A. Zemnder, "HIQUEL: An Interactive Query Language to Define and Use Hierarchies", (vidi Davis, 1983.).
- (Varga, 1994.) Mladen Varga, "Baze podataka - konceptualno, logičko i fizičko modeliranje podataka", DRIP, Zagreb, 1994.
- (Vetter, 1987.) M. Vetter, "Strategy for Data Modelling, Application and Enterprise-wide", John Wiley and sons, Chichester, 1987., p. 344
- (Vujnović, 1995.) R. Vujnović, "SQL i relacijski model podataka" , Znak, Zagreb, 1995.
- (Wikipedia, 2010.) <http://en.wikipedia.org/wiki/Categorization> (09.05.2010.)
- (Wikipedia, 2010.) Wikipedija: http://en.wikipedia.org/wiki/Peter_Chen, 22.04.2010.
- (Wikipedija, 2010.) http://hr.wikipedia.org/wiki/Informacijski_sustavi (19.04.2010.)
- (Yao, 1985.) S. Bing Yao, "Principles of Database Design", Prentice - Hall, New Jersey, 1985.
- (Yourdon, 1979.) E. Yourdon, L. L. Constantine, "Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design", facsimile edn, Prentice Hall, 1979.
- (Zhang, 1983.) Z.Q. Zhang i A.O. Mendelzon, "A Graphical Query Language for Entity-Relationship Databases", 1983., vidi (Davis, 1983.).