

VISOKA TEHNIČKA ŠKOLA STRUKOVNIH STUDIJA
KRAGUJEVAC

Skripta iz predmeta

PROJEKTOVANJE INFORMACIONIH SISTEMA

Dr.Miroljub Banković, prof.

Kragujevac, 2008.

SADRŽAJ

OSNOVI TEORIJE SISTEMA	3
DEFINICIJE SISTEMA:	3
KRITERIJUMI ZA RAZVRSTAVANJE SISTEMA:	4
PODSISTEMI I INTERAKCIJE.....	5
POJAM INFORMATIKE, ENTROPIJE, INFORMACIJA, PODATAKA I INFORMACIONOG SISTEMA.....	7
OSNOVI TEORIJE UPRAVLJANJA	8
OSNOVI TEORIJE INFORMACIJA I KOMUNIKACIJA.....	9
KLASIFIKACIJE INFORMACIONIH SISTEMA.....	10
POLAZNE OSNOVE ZA PROJEKTOVANJE INFORMACIONIH SISTEMA.....	11
ORGANIZACIJA INFORMACIONIH SISTEMA	12
STRUKTURA INFORMACIONOG SISTEMA.....	12
PROJEKTNI PRISTUP IZGRADNJI INFORMACIONIH SISTEMA.....	13
PROJEKTNI ZADATAK	13
IDEJNI PROJEKAT	13
GLAVNI PROJEKAT	14
PROJEKAT IZVEDENOG STANJA.....	14
ANALIZA SISTEMA	15
TEHNIKE SNIMANJA POSTOJEĆEG STANJA.....	15
ANALIZA SISTEMA POMOĆU MATRICA	16
METODI ZA PREDSTAVLJANJE TOKA OBRADE.....	16
MODELOVANJE PROCESA	17
Hijerarhijska dekompozicija poslovnog sistema (stablo procesa)	18
Hijerarhijska struktura (dijagram zavisnosti) poslovnih procesa:	18
Izrada dijagrama toka podataka (po SSA metodu)	20
ANALIZA PODATAKA.....	25
<i>Modelovanje entiteta</i>	<i>25</i>
<i>Modelovanje podataka</i>	<i>30</i>
Komponente modela podataka	32
Vrste modela podataka	34
KLASIČNI METODI PROJEKTOVANJA INFORMACIONIH SISTEMA.....	47
SDM (SYSTEM DEVELOPMENT METHOD).....	48
SSA (STRUCTURED SYSTEM ANALYSIS) - METOD STRUKTURNE ANALIZE SISTEMA.....	49
IMPLEMENTACIJA INFORMACIONOG SISTEMA.....	55
TESTIRANJE INFORMACIONOG SISTEMA	55
KREIRANJE I ADMINISTRACIJA BAZE PODATAKA	55
UVOĐENJE INFORMACIONOG SISTEMA U EKSPLOATACIJU	55
<i>Metodi uvođenja</i>	<i>55</i>
<i>Izbor informatičke opreme.....</i>	<i>56</i>
Faktori koji utiču na izbor opreme.....	56
Metodi za izbor opreme.....	57
LITERATURA	60

OSNOVI TEORIJE SISTEMA

Analitički pristup

Izdvajanje iz celine pojedinih delova koji se zatim zasebno proučavaju.

Osnovna obeležja:

- svi predmeti i pojave (fenomeni) posmatraju se kao skup nezavisnih elemenata koje treba izučavati,
- odnosi između elemenata objašnjavaju se uzročno-posledičnim vezama
- predmeti i pojave opisuju se krutim statičkim definicijama, pravilima ili materijalnim strukturama.

Sistemski pristup

Svi fenomeni (predmeti ili pojave) posmatraju se kao sistemi, odnosno celine koje se ne mogu rastaviti na svoje elemente, a da se pri tome ne izgube osnovna svojstva celine.

Osnovne faze:

- razmišljanje o svrsi i ciljevima i
- razmišljanje o funkcijama kojima se ostvaruju ciljevi.

Predmet proučavanja opšte teorije sistema

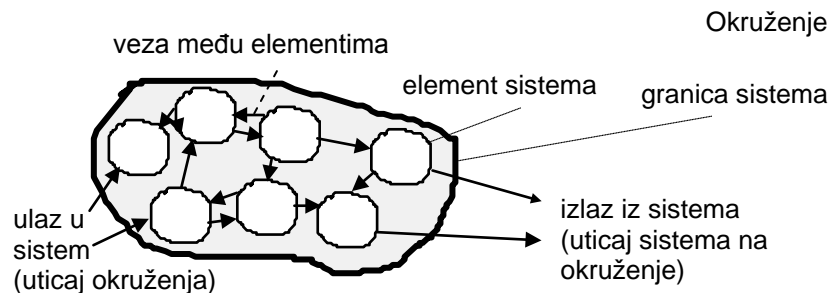
Opšta teorija sistema je naučna disciplina koja se bavi izučavanjem sistema i zakonitosti koje u njima vladaju.

Osnovne postavke opšte teorije sistema

- celina predstavlja više nego zbir delova;
- celina određuje prirodu delova;
- delovi se ne mogu shvatiti ako se posmatraju nezavisno od celine;
- između delova postoje dinamične veze i međuzavisnosti

Definicije sistema:

- Izdvojena funkcionalna celina koja je sastavljena od skupa objekata, njihovih utvrđenih svojstava (atributa) i skupa relacija koje povezuju objekte, kao i od svojstava relacija.
- Uređena celina koju sačinjavaju elementi (objekti, pojmovi ili činjenice) između kojih postoje ili mogu da se uspostave bilo kakvi odnosi.
- Kompleksni dinamički entitet identifikovan u određenom vremenu i prostoru, koji se sastoji iz određenog broja povezanih delova ili podsistema i ima mogućnost adaptacije na zahteve okruženja.



Struktura sistema i veze (među elementima i sa okruženjem)

Elementi sistema - delovi sistema koji sa svojim uzajamnim vezama čine strukturu sistema

Veze sistema - dva elementa sistema su povezana ako jedan od njih deluje na drugi. Veze (sprege) koje se uspostavljaju među elementima mogu biti: materijalne, energetske ili informacione. Osnovni tipovi veza među elementima sistema su:

- jednostrana veza (uzročno-posledična)
- povratna veza (posredna i neposredna) i
- uporedna veza.

Ponašanje sistema- definiše se kao određena transformacija njegovih ulaznih veličina u izlazne veličine.

Osnovne karakteristike sistema:

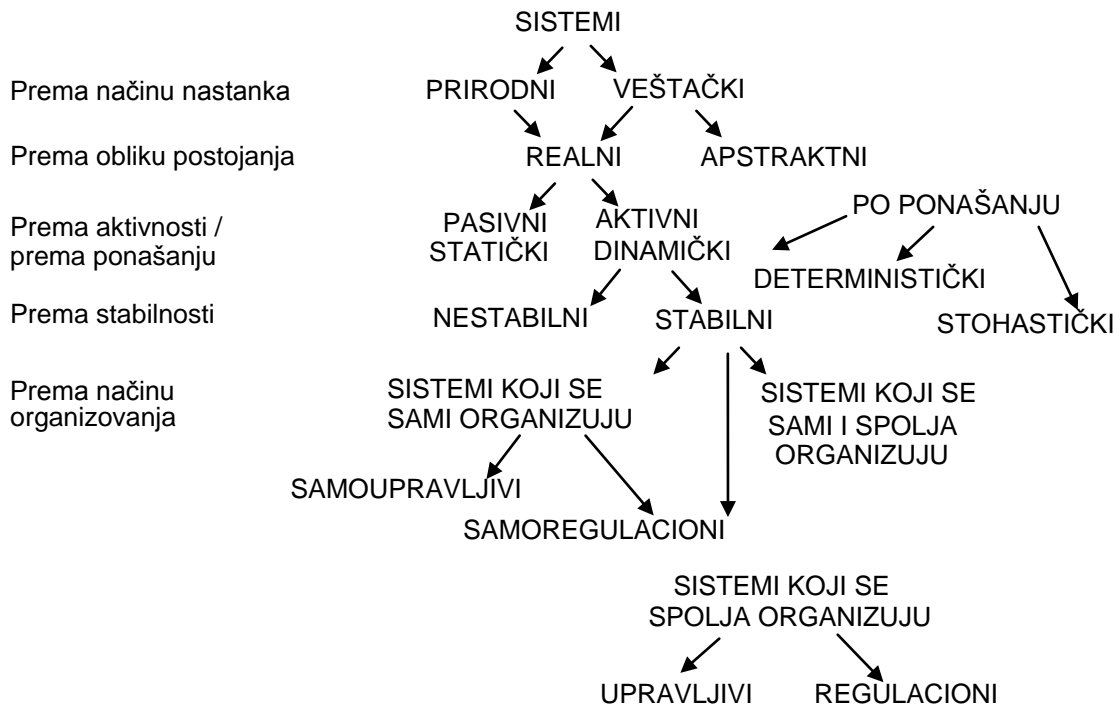
- jednoznačna identifikacija
- kompleksnost
- dinamičnost
- sistem postoji kao entitet
- zavisnost
- stabilnost
- adaptibilnost

Osnovne postavke opšte teorije sistema:

- Svrhovitost, odn. ciljna usmerenost
- Celovitost odn. *holizam*
- Sistem čine uzajamno povezani elementi, specijalizovani za obavljanje različitih funkcija.
- Hijerarhija
- Ulazi i izlazi
- Transformacija
- Energija
- Entropija
- Ekvifinalnost
- Sinergija
- Cilj sistema postiže se procesom regulacije.

Kriterijumi za razvrstavanje sistema:

- način nastanka
- oblik postojanja
- aktivnosti
- predvidivost ponašanja
- stabilnost
- način organizovanja
- povezanost s okruženjem
- unutrašnji odnosi
- stepen otvorenosti
- kompleksnost



Klasifikacija sistema

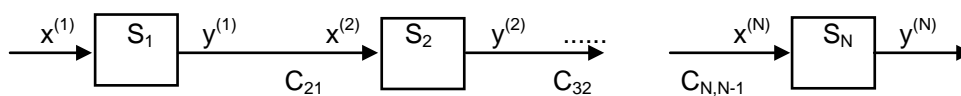
Kibernetički sistemi su sistemi koji su:

- upravljani (vođeni ka ostvarenju cilja funkcionisanja);
- dinamički (inače bi upravljanje bilo besmisleno);
- kompleksni (inače ne bi bilo neophodno upravljati njima);
- probabilistički (inače bi upravljanje bilo nepotrebno);
- sa povratnom spregom (inače bi upravljanje bilo isuviše teško).

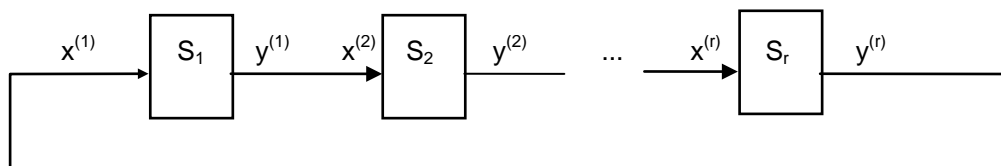
Podsistemi i interakcije

Povezivanjem više podsistema u okviru sistema može se formirati:

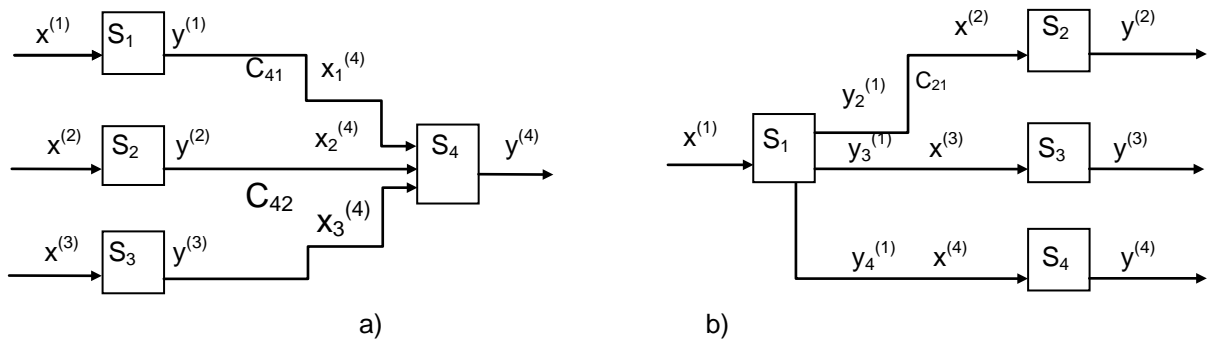
- otvoreni lanac;
- zatvoreni lanac
- razgranati lanac



Otvoreni lanac povezivanja podsistema

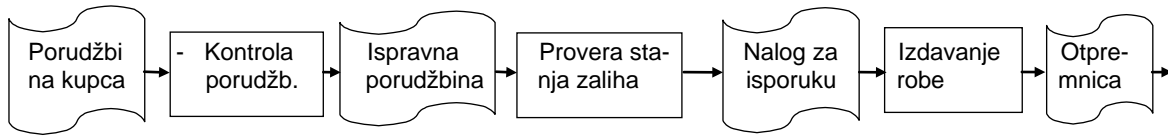


Zatvoreni lanac povezivanja podsistema

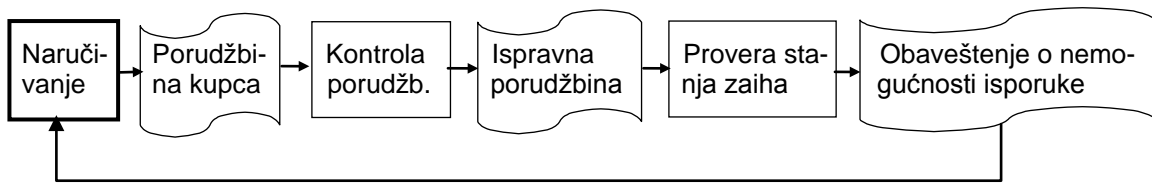


Razgranati lanac - a) grananje ulaza, b) grananje izlaza

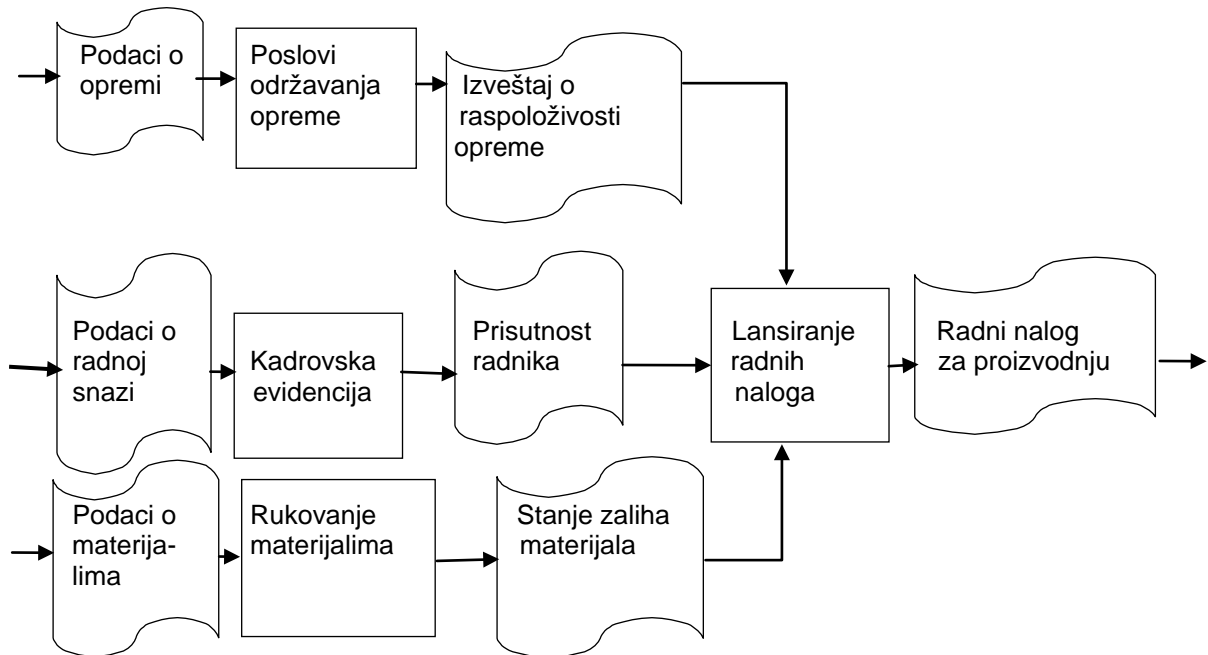
Primeri



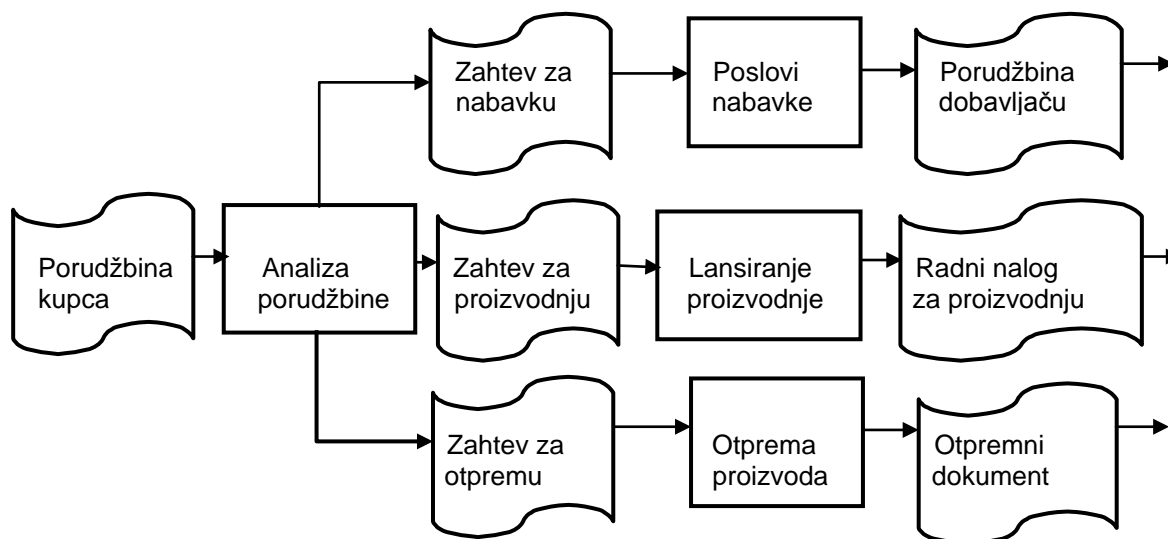
Otvoren lanac povezivanja



Zatvoren lanac povezivanja



Grananje ulaza



Grananje izlaza

POJAM INFORMATIKE, ENTROPIJE, INFORMACIJA, PODATAKA I INFORMACIONOG SISTEMA

Informatika je nauka o sistematskoj i racionalnoj obradi informacija kao nosilaca ljudskog znanja i komunikacija u tehničkom, ekonomskom i društvenom kontekstu, prvenstveno pomoću automatskih mašina

Entropija izražava zakon po kome se realni sistem ne može vratiti u stanje u kome je jednom bio bez izvesnog gubitka energije.

Entropija izražava prirodnu težnju svakog realnog sistema da dođe u stanje najveće verovatnoće, koje odgovara potpunom haosu koji bi natao zbog njegove totalne neorganizovanosti.

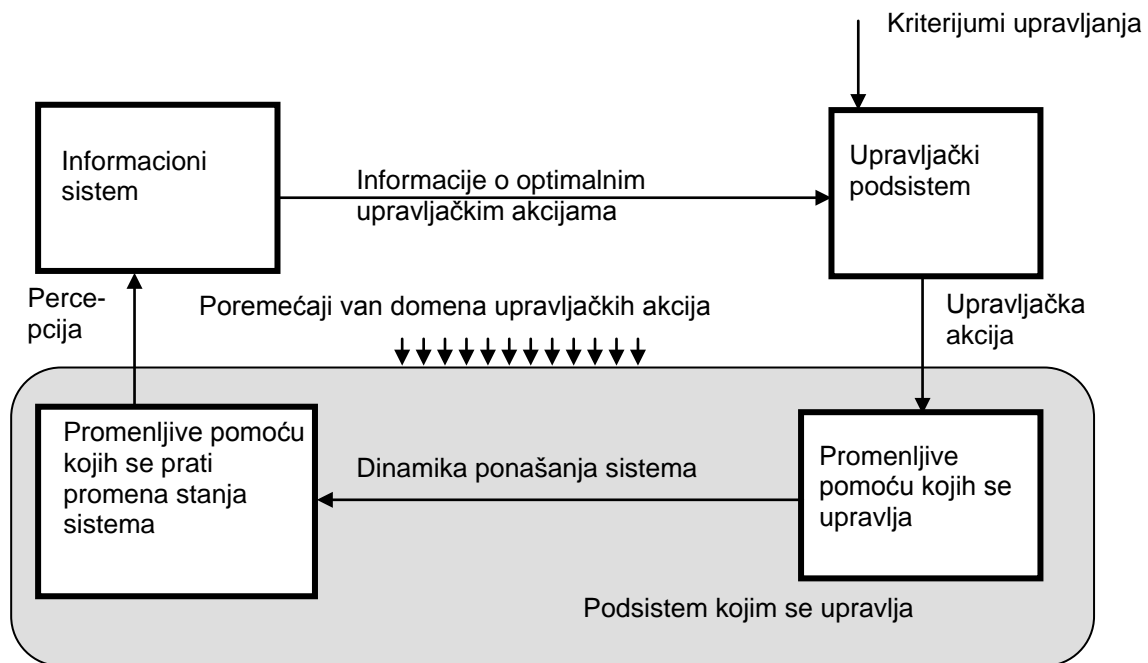
Informacija je, suprotno od entropije, mera za red odnosno organizovanost sistema. Po Šenonu, informacija se može posmatrati kao negativna vrednost entropije - negentropija. Zaštita od porasta entropije sastoji se u prikupljanju podataka koji opisuju sistem i njihovom transformisanju u informacije koje se koriste za kontrolu i ograničavanje rasta entropije. Postupci upotrebe informacija za kontrolu entropije odn. porast organizovanosti sistema predstavljaju u kibernetici upravljanje u širem smislu.

Podaci su registrovane činjenice, oznake ili zapažanja nastala u toku nekog procesa. Pojam podataka vezan je za fizičke simbole koji mogu da se beleže (registruju), čuvaju, prenose i obrađuju.

Podatak se može definisati i kao kodirana predstava o nekoj činjenici iz realnog sveta ili kao zapis o realnom svetu, koji služi za formiranje informacija. S druge strane, informacija je protumačeni podatak, odnosno inkrement znanja koji se može izvući iz podatka.

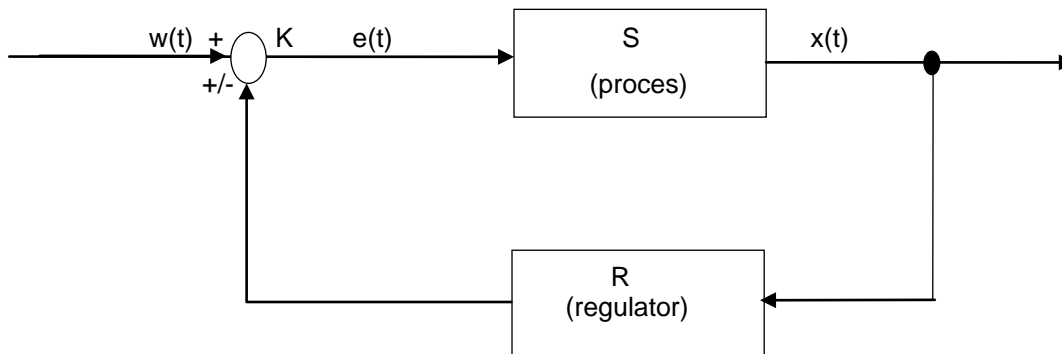
Informacioni sistem se, polazeći od opšte teorije sistema i sistemskog pristupa, može definisati kao *sređeni skup metoda, procesa i operacija za prikupljanje, čuvanje, obradu, prenošenje i distribuciju podataka u okviru jedne organizacije, uključujući i opremu koja se u te svrhe koristi i ljude koji se tim sktivnostima bave*. Informacioni sistem je *model* realnog - poslovnog sistema. Četiri osnovne komponente informacionog sistema su:

- operacije nad podacima;
- metodi i tehnologije obrade podataka;
- sistemska analiza i
- tehnike modelovanja.



Upravljanje poslovnim sistemom posredstvom informacionog sistema

OSNOVI TEORIJE UPRAVLJANJA



Elementarni regulacioni proces

Proces (S) - objekat regulisanja

Regulator (R) - deo za regulisanje koji obuhvata sve elemente procesa regulisanja

Ulazni signal $w(t)$ - uticaj okruženja (referentni ulaz, vodeća veličina);

Izlazni signal $x(t)$ - uticaj procesa na okruženje (upravljana veličina);

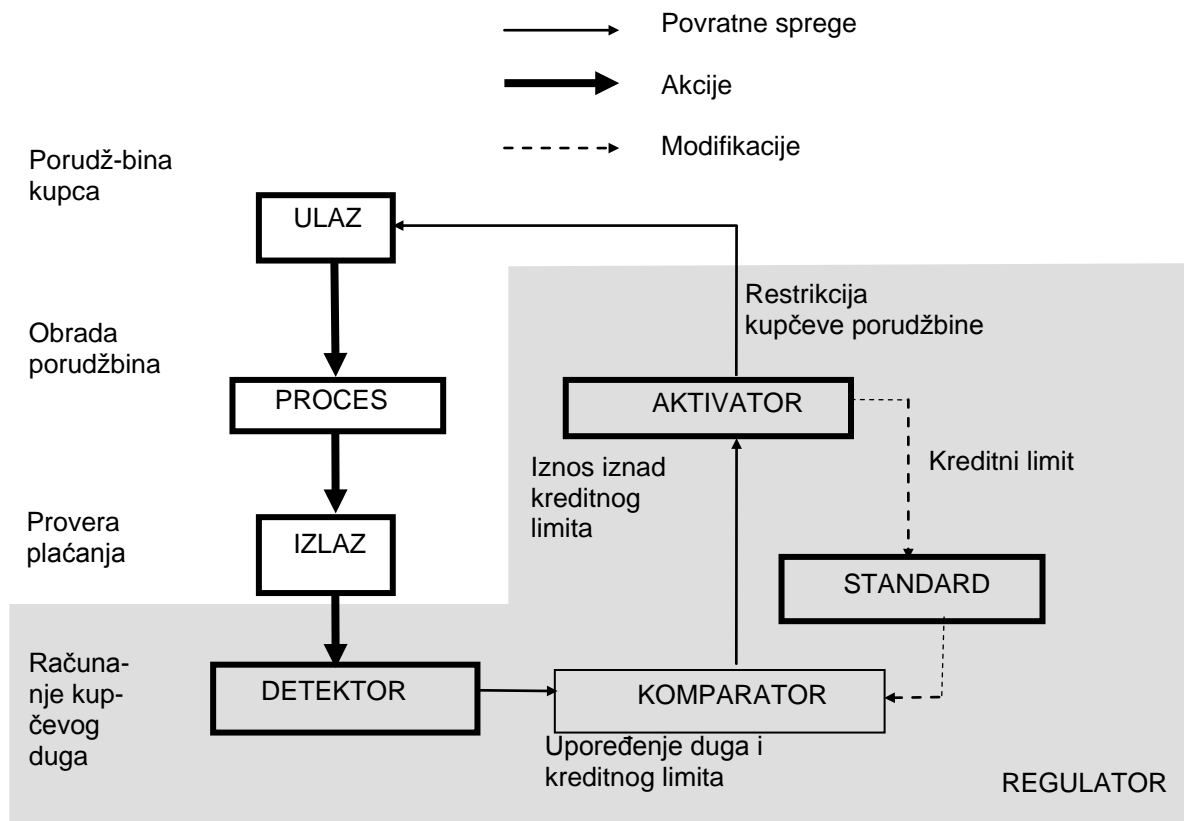
Signal povratne sprege $y(t)$;

Upravljačka akcija (odstupanje) $e(t) = w(t) \pm y(t)$;

Komparator K - generiše upravljačku akciju

Negativna povratna sprega je povratna sprega kojom se smanjuje uticaj ulaznog dejstva na izlaznu veličinu.

Pozitivna povratna sprega je povratna sprega kojom se povećava uticaj ulaznog dejstva na izlaznu veličinu.



Primer regulisanja primenom povratne sprege

- Objekat regulisanja (s): Obrada i kontrola porudžbine kupca
- Regulator (R): Detektor, komparator, standard, aktivator
- Ulazni signal (w): porudžbina kupca
- Izlazni signal (x): Kupčev dug
- Signal povratne sprege (y): Iznos iznad kreditnog limita
- Upravljački signal (e): Ograničenje porudžbine

OSNOVI TEORIJE INFORMACIJA I KOMUNIKACIJA

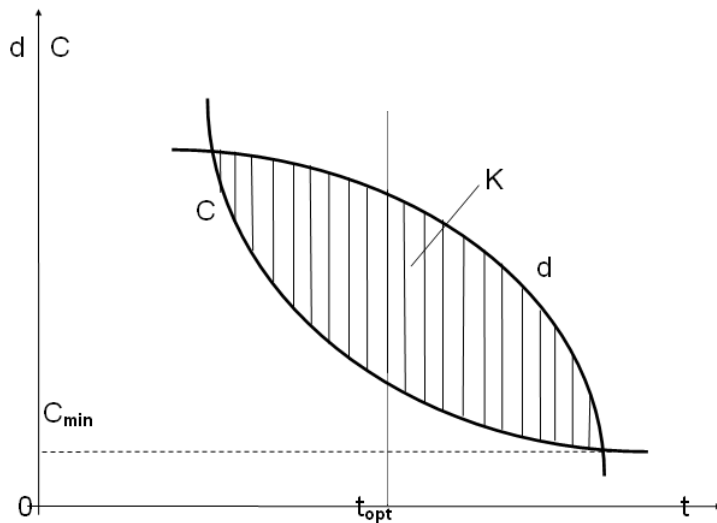
Vrste informacija

- Naučno-tehničke informacije / poslovne informacije
- Interne (informacije iz sistema) / eksterne informacije (iz okruženja)
- Informacije zasnovane na podacima iz prošlosti / informacije o tekućem funkcionisanju / informacije o budućem funkcionisanju ili događajima
- Rekurentne (periodične) / nerekurentne (povremene)
- Informacije *opšteg značaja* (zakoni, propisi,...) / informacije *na nivou poslovnog sistema* (završni račun,...) / informacije *na nivou funkcije* (radni nalog, faktura...) / *individualne informacije* (plata pojedinca...).
- Operativne / taktičke / strateške informacije

Faktori kvaliteta informacija:

- tačnost;
- blagovremenost (aktuelnost);
- relevantnost;
- sažetost (informacija treba da je što kraća);
- svrsishodnost;
- razumljivost;

- ekonomičnost;
- iniciranje određene upravljačke akcije



t - vreme generisanja informacije; C - cena generisanja; d - doprinos informacije; t_{opt} – najpogodnije vreme za korišćenje informacije; C_{min} - najmanja cena informacije; K - oblast korisnosti informacije

KLASIFIKACIJE INFORMACIONIH SISTEMA

Prema vrsti obrade:

Informacioni sistemi zasnovani na paketskoj (batch) obradi, na interaktivnom radu, radu u realnom vremenu i sl.

Prema vrsti usluga koje pružaju:

- **sistemi za usluge obrade podataka opšte namene** (korisnici obezbeđuju programe i podatke i definišu informacije prema svojim potrebama);
- **sistemi za čuvanje i pretraživanje podataka** (imaju zadatak da čuvaju velike količine podataka u vidu dokumenata iz pojedinih oblasti i da na korisnički upit pronalaze tražene podatke);
- **sistemi za komutaciju poruka** (zadatak im je prenos poruka od izvora do korisnika preko transmisionih linija);
- **sistemi za upravljanje procesima** (u procesnoj proizvodnji - sa senzora i monitora se dobijaju podaci o stanju procesa i na osnovu njih se generišu signali za upravljanje i kontrolu uređaja - regulatora, procesne opreme i mašina);
- **sistemi za kontrolu i upozorenja** (periodičnom obradom i ažuriranjem baze podataka se daju upozorenja i izveštaji za preduzimanje akcija, npr. u upravljanju zalihama izveštaj o kritičnim i nekurentnim zalihama);
- **sistemi za obradu transakcija** (obrađuju unapred definisane i isprogramirane transakcije i stvaraju predviđene informacije, npr. ulaz/izlaz robe iz magacina, fakturisanje, naručivanje itd).

Prema stepenu automatizacije:

- **Neautomatizovani informacioni sistemi** (zasnovani na ručnoj ili mehanografskoj obradi podataka);
- **Sistemi AOP-a** (obezbeđuju prikupljanje, čuvanje i obradu podataka i izradu informacija o pojedinim funkcijama; sastoje se iz aplikacija AOP-a za pojedine funkcije ili organizacione celine; veze među aplikacijama su slabe i ne može se govoriti o jedinstvenom IS);
- **Upravljački informacioni sistemi** (podržavaju donošenje rutinskih odluka, uglavnom na nivou operativnog upravljanja; težište stavljaju na informacije i njihovu upotrebu za donošenje odluka);
- **Izvršni informacioni sistemi** (daju analitičke informacije o tekućem stanju organizacije i projekciji u budućnosti, a namenjene najvišem nivou upravljanja);
- **Sistemi za podršku odlučivanju** (DSS - Decision Support Systems), podržavaju donošenje složenih odluka; njihov razvoj počeo je 70-tih godina i predstavljaju kvalitativni skok u razvoju informacionih sistema).

- **Ekspertni sistemi**, namenjeni rešavanju stručnih problema za određenu specijalizovanu oblast.

POLAZNE OSNOVE ZA PROJEKTOVANJE INFORMACIONIH SISTEMA

Organizacioni sistemi mogu biti:

- **Moralni sistemi** - oni čijim se funkcionisanjem uspostavljaju određene relacije ili odnosi među ljudima (političke partije, verske zajednice, druga udruženja i savezi).
- **Instrumentalni sistemi** - oni čija je namena da posluže kao instrument nekoj ljudskoj delatnosti ili društvenoj aktivnosti (razna naučna, stručna ili umetnička dela).
- **Materijalni sistemi** - oni koji funkcionišu na osnovu materijalnog ulaza u sistem, materijalne transformacije u sistemu i materijalnog izlaza iz sistema. Tu spadaju ekonomski, poslovni ili proizvodni sistemi

Osnovni principi dizajna IS:

- **Informacioni sistem treba definisati iz perspektive najvišeg nivoa rukovođenja**, kako bi davao informacije koje će omogućiti donošenje najvažnijih odluka. IS treba da podržava osnovni cilj poslovnog sistema kao celine kao i parcijalne ciljeve pojedinih njegovih delova.
- **IS treba projektovati prema ključnim procesima** koji učestvuju u okviru kreiranja proizvoda ili usluga, u zavisnosti od konkretnog poslovnog sistema. Time se postiže da IS bude nezavisan od organizacione strukture jer je ona podložna čestim promenama a razvoj IS je dugoročan i skup posao. Zato se IS projektuje prema poslovnim procesima, stalnim za određenu delatnost preduzeća.
- Da bi IS mogao dati informacije raznim korisnicima u okviru preduzeća, **podatke treba tretirati kao sredstvo celog preduzeća**, tj. sistema, a ne kao vlasništvo pojedinih njegovih delova ili aplikacija, kako bi svi korisnicu dobili informacije u potrebnom obliku.

Preduslovi:

- Zbog složenosti i obimnosti posla na projektovanju IS-a, ceo taj posao odn. projektovanje treba podeliti na faze;
- U sve faze projektovanja, uvođenja i korišćenja IS-a treba uključiti rukovodstvo;
- IS treba planirati "sa vrha" a realizovati odn. uvoditi ga "sa dna";
- Prilikom projektovanja koristiti praktičnu i jednostavnu tehnologiju planiranja, uvesti odgovarajuću tehnologiju upravljanja podacima i definisati funkciju za administraciju podataka.

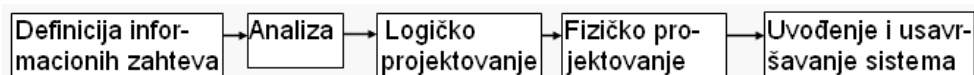
Polazne osnove za projektovanje informacionog sistema:

- Projekat treba da radi mešoviti tim stručnjaka, eksternih (specijalista za pojedine oblasti) i internih (ljudi iz poslovnog sistema).
- Pre formiranja tima treba oformiti koordinaciono telo za kontrolu i praćenje rada na realizaciji projekta; to telo utvrđuje sastav i vrši izbor projektnog tima.
- U tim ulaze (pored eksternih konsultanata za pojedine oblasti) projektanti, analitičari i programeri.

Faze u projektovanju IS:

- strateško planiranje;
- analiza;
- dizajn;
- izgradnja;
- uvođenje sistema u eksploataciju i otklanjanje nedostataka.

Navedene faze se mogu predstaviti i na sledeći način:



KRITERIJUMI KLASIFIKACIJE INFORMACIONIH ZAHTEVA

Kriterijumi klasifikacije informacionih zahteva	Nivoi odlučivanja		
	Operativni	Taktički	Strateški
Mogućnost definisanja informacionih zahteva	moguće unapred	delimično moguće	Ad hoc zahtevi
Strukturisanost inform. zahteva	pretežno	delimično	nestrukturisano
Izvor informacija	interni	interni i eksterni	eksterni (i interni)
Vremenska određenost	sadašnjost i prošlost	sadašnjost	budućnost
Tačnost i pouzdanost	velika	srednja	mala

ORGANIZACIJA INFORMACIONIH SISTEMA

Organizacija u odnosu na strukturu

- **Hijerarhijska** (sa izdiferenciranim nivoima rukovođenja i strukturom koja podrazumeva subordinaciju, slično kao u organizaciji preduzeća);
- **Projektna** (timska - podrazumeva fleksibilno formiranje i funkcionisanje mešovityh stručnih timova u periodu intenzivnog rada na novim projektima IS);
- **Matrična** (uz hijerarhiju uključuje i horizontalne veze, odnosno stalnu saradnju timova iz pojedinih organizacionih celina).

Organizacija u odnosu na komunikaciju

Dominantan je koncept **distribuirane obrade podataka** sa podelom zaduženja između lokalnih centara koji obrađuju podatke pojedinih funkcija i centralne funkcije informatike, opremljene korporacijskim računarom na koji se periodično dostavljaju svodne informacije iz lokalnih centara i zajedničkim funkcijama omogućuje uvid u centralnu bazu podataka, sa informacijama od interesa za ceo poslovni sistem.

Organizacija u odnosu na upravljanje

Kako god fizički bio organizovan, informacioni sistem mora da obezbedi podatke za:

- **operativni**,
- **taktički** i
- **strateški** nivo upravljanja.

STRUKTURA INFORMACIONOG SISTEMA

Hijerarhijska struktura

Dominantna je centralizovana obrada podataka, za koju se podaci samo prikupljaju lokalno i odmah prenose na centralni računar. Razvoj ovakvog informacionog sistema omogućava punu unifikaciju i diktiranje rada krajnjih korisnika, kao i jedinstvenu, uvek ažurnu, bazu podataka, ali je eksploatacija osetljiva na kvarove centralnog računara i usporava se razvoj jer ga radi centralna ekipa.

Decentralizovana struktura

Dovodi do stvaranja "informativnih ostrva", jer se informativni sistem razvija parcijalno, u okviru pojedinih posistema odn. poslovnih funkcija. Lokalni računski centri se najčešće baziraju na heterogenoj opremi, različitim standardima u obradi podataka i pokrivaju samo lokalnu problematiku.

Matrična struktura

Obezbeđuje ne samo vertikalnu, već i horizontalnu komunikaciju, balansirajući detaljnost i poslovanju primeren razvoj lokalnih aplikacija AOP-a sa potrebom da se, povezivanjem centara i razmenom podataka između njih stvore uslovi da se ukupni informativni sistem preduzeća može izgraditi na Bottom-Up način, od informacija za operativni nivo odlučivanja, preko taktičkog do zajedničkog, strateškog nivoa odlučivanja (upravljanja). Ovakav koncept odgovara distribuiranoj obradi informacija

PROJEKTNI PRISTUP IZGRADNJI INFORMACIONIH SISTEMA

Projektovanje informativnih sistema svodi se na izradu kompletne tehničke dokumentacije kojom se definiše cilj informativnog sistema, daju se osnovne karakteristike, sastav i uzajamne veze između komponenata i njegovo funkcionisanje.

Sve aktivnosti vezane za projektovanje informativnog sistema odlikavaju se kroz sledeće dokumente koji se stvaraju u pojedinim fazama razvoja informativnog sistema:

- projektni zadatak;
- idejni projekat;
- glavni projekat i
- projekat izvedenog stanja.

Projektni zadatak

Projektnim zadatkom se naručuje jedan projekat informativnog sistema i sa njim se jasno definiše zadatak projektnog tima. **Projektni zadatak sadrži:**

- naslovnu stranu (identifikacija naručioca, naziv projekta za koji se daje zadatak);
- osnovne vezne podatke (inicijativna dokumentacija, nosilac istraživanja i izrade projekta, zahtevani rok - period izrade i implementacije);
- potpise predstavnika naručioca i izvršioca (overa zadatka);
- uvodne napomene (kratak opis postojećeg sistema);
- uzroke i inicijative (uočene problemske tačke u postojećem sistemu);
- ciljeve čije postizanje se očekuje izradom projekta;
- predmet i granice istraživanja (šta se radi a šta ne);
- period istraživanja;
- sadržaj projekta (sa što više detalja);
- obavezu izrade projektnih zadataka za npr. građevinski projekat, instalacije, obuku);
- organizaciju, oblik i način prezentacije rezultata projektovanja;
- način verifikovanja rezultata projektovanja;
- event. način regulisanja odnosa naručioca i izvođača (kod eksternih projekatana);
- očekivanu dinamiku izrade projekta (faze, termini);
- datum i mesto izdavanja projektnog zadatka.

Idejni projekat

Idejnim projektom se vrši projektovanje logičke strukture informativnog sistema. Njime se daje logika prikupljanja, obrade i izdavanja informacija kao i logička struktura tehničko - tehnoloških resursa potrebnih za realizaciju projektovanog sistema. Ovaj projekat se naziva i investicioni, jer se njime predviđaju sredstva potrebna za realizaciju IS. **Idejni projekat sadrži:**

- kratak opis poslovnog sistema za koji se projektuje IS (delatnost, kratak istorijat poslovnog sistema, organizaciona struktura);
- prikaz metoda koje se koriste u projektovanju informativnog sistema;

- identifikaciju procesa u poslovnom sistemu (matrice proces -funkcija, prethođenje procesa, struktura procesa, dijagrami zavisnosti procesa);
- identifikaciju relevantnih entiteta (entiteti, atributi, odnosi među entitetima, dijagram zavisnosti entiteta);
- strukturu / arhitekturu informacionog sistema za sve nivoe rukovođenja (globalna organizacija podataka, organizacija prikupljanja podataka, način obrade podataka i asortiman informacija sa njihovim sadržajem);
- specifikaciju potrebne opreme za realizaciju projekta (hardver, sistemski softver, infrastruktura - objekti, komunikacione linije i sl);
- plan obezbeđenja i obuke potrebnih kadrova za realizaciju IS;
- plan daljih aktivnosti sa rokovima i zaduženjima.

Glavni projekat

Predstavlja konkretizaciju onog rešenja informacionog sistema koje je dato u idejnom projektu, tako da sadži sledeće elemente:

- specifikaciju opreme na kojoj će se razvijati informacioni sistem;
- logičku organizaciju baze podataka;
- fizičku organizaciju baze podataka;
- način prikupljanja podataka;
- način održavanja baze podataka;
- način dobijanja svake od informacija;
- definisanu proceduru čuvanja podataka i baze podataka;
- definisan način prelaska na novi informacioni sistem;
- operativni plan aktivnosti za sledeću fazu.

Projekat izvedenog stanja

Ovaj projekat treba da sadži konkretna rešenja svih elemenata i aktivnosti informacionog sistema.

Projekat izvedenog stanja sadži:

- specifikaciju i opis datoteka;
- bazu podataka;
- specifikaciju programskih zadataka;
- uslove za održavanje baze podataka;
- specifikaciju i listinge svih programa za izvođenje samog procesa obrade podataka;
- šifarnike resursa čije šifre predstavljaju ključeve u bazi podataka;
- scenario (prikaz) obrade tj. izvođenja aplikacija AOP-a;
- specifikaciju i razmeštaj opreme;
- potrebna uputstva (za rad korisnika, održavanje baze podataka i "By pass" procedure).

ANALIZA SISTEMA

Definicija	Analiza	Moguće rešenje	Razrada	Usvajanje
Šta	Zašto	Šta bi trebalo da bude	Zašto bi moralo da bude	<div style="border: 1px solid black; padding: 5px; text-align: center;"> Da li rešenje odgovara svim zahtevima </div>
Gde	Zašto tamo	Gde treba da bude	Zašto tamo	
Kada	Zašto tada	Kada treba da bude	Zašto tada	
Ko	Zašto on/ona	Ko bi mogao da bude	Zašto on/ona	
Kako	Zašto tako	Kako bi trebalo	Zašto tako	

Pet ključnih reči u sistemskoj analizi (engl. 5W - Five Words: What, Where, When, Who, Why).

Predmet istraživanja analize sistema

- istorijski aspekt organizacije
- poslovne operacije
- informacioni tokovi
- postojeće metode, procedure i tehnička sredstva koja se u sistemu koriste
- dokumentacija
- delatnost, odnosno glavni poslovi organizacije i njihov obim
- postojeći sistem šifriranja
- ljudski faktor (radna snaga)
- kontrolne tačke u sistemu
- budući (očekivani) zahtevi sistema.

Tehnike snimanja postojećeg stanja

- Anketiranje
- Intervjuisanje
- Posmatranje
- Upoređivanja i pretpostavke
- Čitanje dokumentacije
- Merenje
- Uzorkovanje
- Simulacija
- Verifikacija
- Dokumentovanje

Izvođenje analize postojećeg stanja poznaje dva pristupa:

- Da se analiza vrši **od nižih ka višim nivoima** u hijerarhiji poslovnog sistema (**Bottom Up** - ovaj način pristupa analizi pogodan je utoliko što omogućava da se uoče neracionalnosti i nedostaci koji se kriju na višim nivoima. Npr., pri osnivanju nekog preduzeća se viši organizacijski nivoi formiraju u cilju boljeg odvijanja osnovnih poslovnih procesa. Međutim, povećavanje broja organizacionih nivoa u strukturi preduzeća uslovljava duži put kretanja dokumentacije i informacija, a to znači i duže vreme i veće troškove i uopšte veću neefikasnost u radu. Pristup odozdo na gore omogućava da se uoče takvi nedostaci);

- Analiza **od viših ka nižim nivoima (Top Down)** - ovaj pristup analizi se češće koristi. On započinje analizu globalne strukture sistema, tj. analizira sistem kao celinu koristeći kao polaz istorijski aspekt organizacije, planove, organizacionu i funkcionalnu strukturu, da bi prešao na analizu pojedinih delova sistema sa njihovim poslovima i podacima koje koriste. Proces raščlanjavanja organizacije se nastavlja do nivoa radnih mesta i poslova koji se u okviru njih obavljaju. U fazi raščlanjavanja analiziraju se i informacioni tokovi na svakom nivou. Analiza tokova informacija treba da pruži odgovor na pitanje opravdanosti kretanja i zadržavanja dokumentacije u pojedinim funkcijama - često se otkriva da su pojedini tokovi nepotrebni ili neopravdano dugo traju).

Analiza sistema pomoću matrica

Najčešće se koriste **dvodimenzionalne matrice** za analizu odnosa sledećih parova elemenata poslovnog sistema:

- organizacione jedinice - funkcije;
- organizacione jedinice - procesi;
- organizacione jedinice - osnovni podaci;
- dokumenti - podaci;
- izveštaji - ulazni podaci;
- ulazni dokumenti - izlazni dokumenti;
- izveštaji - datoteke;
- funkcije - datoteke;
- programi - podaci;
- izveštaji - programi
itd.

Org.jed. \ Proces	OJ-1	OJ-2	...	OJ-m
Proces-1	X		...	X
Proces-2	X		...	
...
Proces-n		X	...	X

Primer matrice Proces/Organizaciona jedinica (x – proces se obavlja u organizacionoj jedinici)

Metodi za predstavljanje toka obrade

Algoritam podrazumeva skup pravila koja određuju proces transformacije ulaznih u izlazne podatke, odnosno kojima se zadaje redosled operacija koje treba izvršiti nad ulaznim (polaznim) podacima da bi se dobilo traženo rešenje koje je potpuno određeno izlaznim podacima.

Najvažnije karakteristike algoritma su:

- **determinisanost** (jednoznačnost propisanog redosleda operacija, koji ne dozvoljava njegovo proizvoljno tumačenje);
- **masovnost** odn. *univerzalnost* (pogodnost za rešavanje većine ili svih zadataka istog tipa);
- **diskretnost** (mogućnost razlaganja algoritma na elementarne operacije);
- **rezultativnost** (mogućnost da se rešenje dobije posle konačnog broja koraka);
- **invarijantnost u odnosu na izvršioce** (algoritam ne zavisi od osobe ili tipa mašine koja ga izvršava).

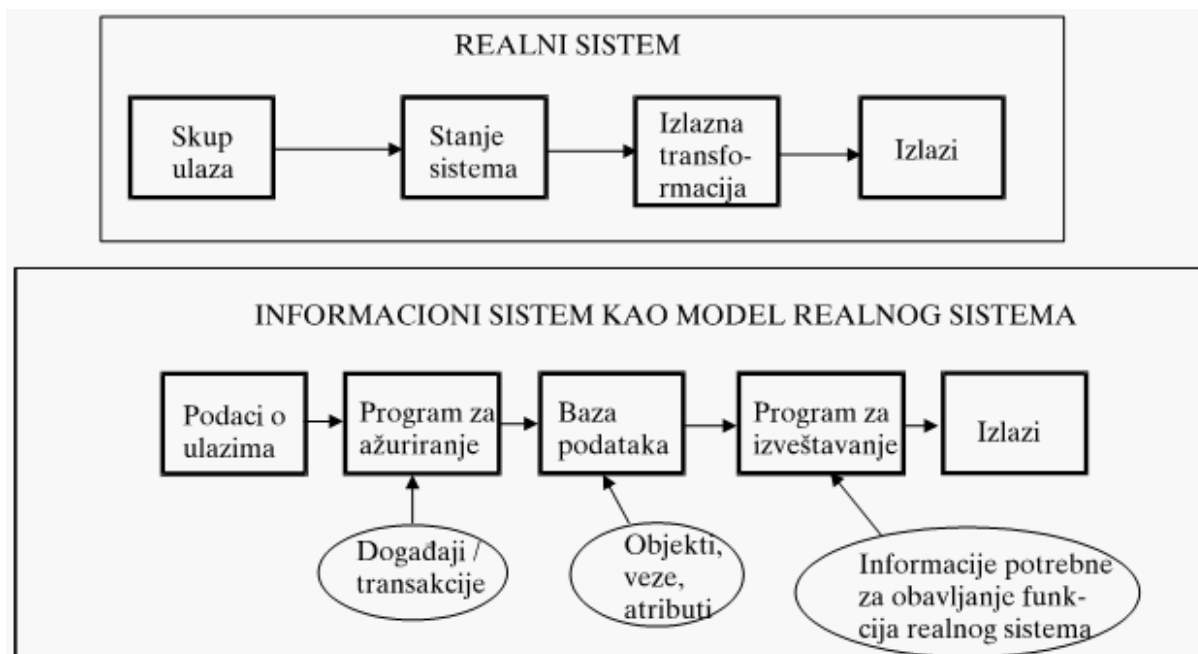
Metodi za prikaz algoritma koji je razumljiv za čoveka:

- opis algoritma na prirodnom jeziku (verbalni model);
- dijagram toka podataka (Flow Chart);
- stablo odlučivanja;
- tabela odlučivanja;

- pseudo kod;
- Nassi - Schneidermann dijagram itd.

Modelovanje procesa

Model procesa opisuje osnovne aktivnosti, poslove i operacije u poslovnom sistemu.



Veza realnog (poslovnog) sistema i informacionog sistema

Model procesa je intelektualno sredstvo za opisivanje dinamike sistema, dejstva ulaza na stanje i izlazne informacije preko programa nad definisanim modelom podataka.

Poslovni proces je:

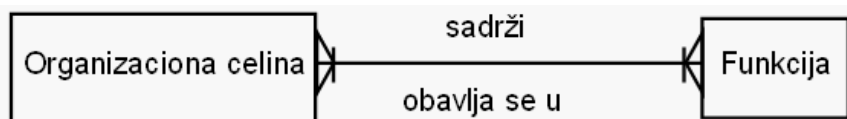
- Skup aktivnosti kojima se informacioni ulaz (podaci) transformiše u upotrebljiv informacioni izlaz.
- Grupa logički povezanih aktivnosti za upravljanje poslovnim resursima.

Skup logički povezanih poslovnih procesa predstavlja **poslovnu funkciju**

Funkcije i procesi su nezavisni od organizacije tj. organizacione strukture.

Model procesa opisuje niz elementarnih aktivnosti tj. poslova i operacija kao i tok podataka između njih.

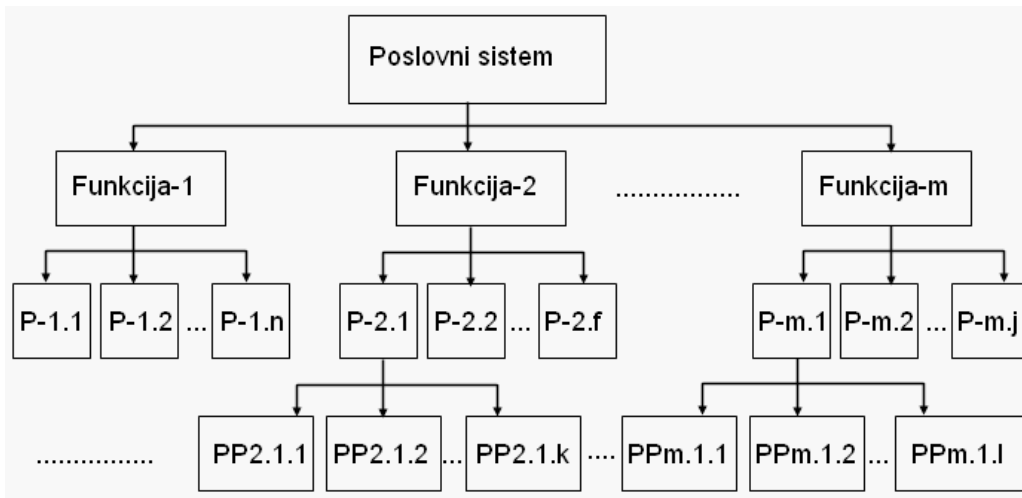
Analiza procesa i analiza podataka se međusobno prepliću i dopunjavaju, tako da ih je u praksi projektovanja teško razdvojiti, već se izvode istovremeno.



Jedna organizaciona celina sadrži jednu ili više funkcija.

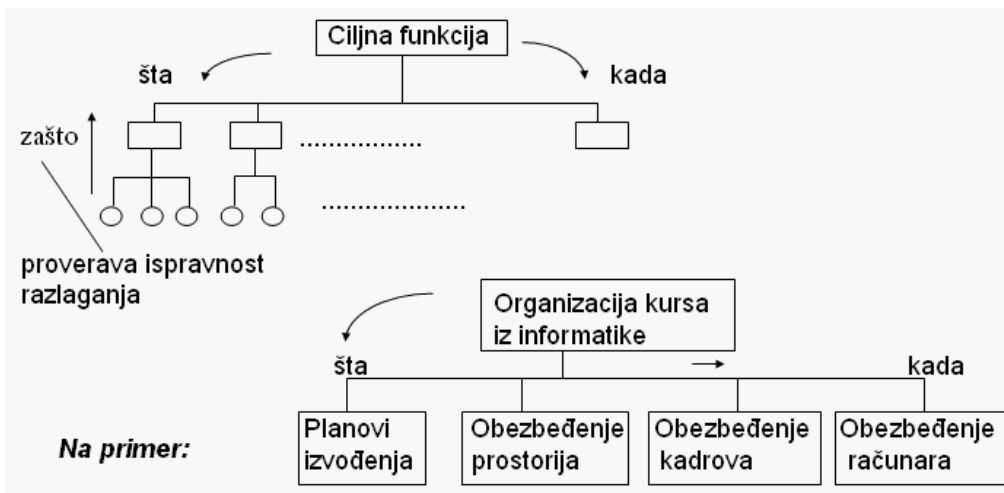
Jedna funkcija može da se obavlja u jednoj ili više organizacionih celina.

Hijerarhijska dekompozicija poslovnog sistema (stablo procesa)



P – proces, PP - potproces

Hijerahijska struktura (dijagram zavisnosti) poslovnih procesa:

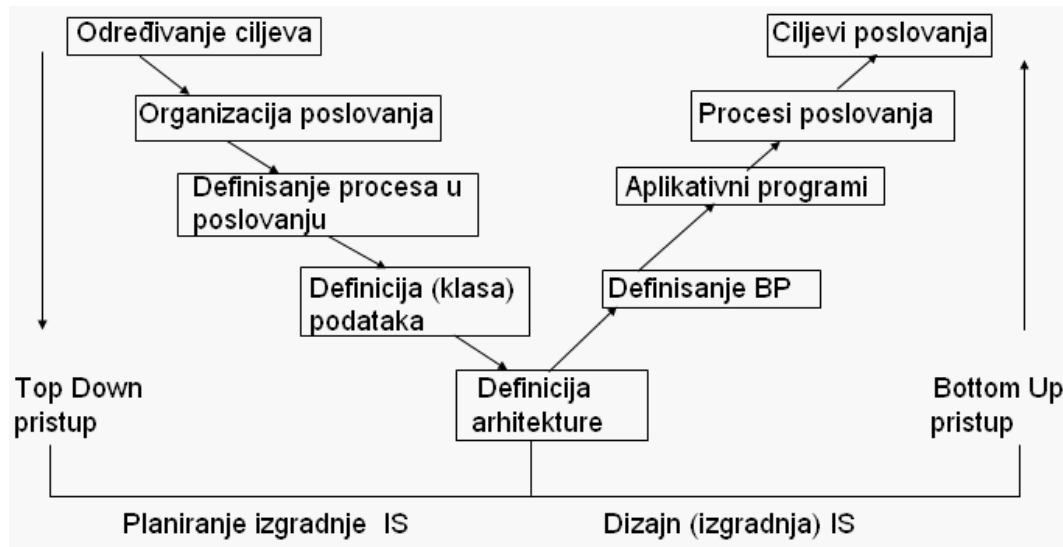


Matrica proces/organizaciona celina (primer)

Organizac. celina / Proces	Plan	Služba nabavke	Služba proizvodnje	Služba finansija	Služba prodaje	Služba održavanja opreme
Uraditi planove	O, I	U	U		U	
Ugovoriti nabavku		O, I		U		
Uraditi tehničku dokumentaciju			O, I			
Proizvesti proizvode	U		O, I			
Isporučiti proizvode kupcu				U	O, I	
Fakturisati proizvode					O, I	

O - odgovornost, I - izvršavanje neke funkcije, U - učestvovanje u izvođenju procesa

Oznaka "1" u nekoj ćeliji tabele znači da proces u odgovarajućoj koloni prethodi procesu u odgovarajućem redu.

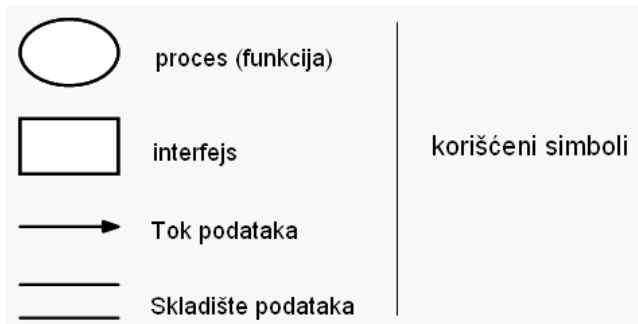


Primena Top-Down / Bottom – Up principa u razvoju

Izrada dijagrama toka podataka (po SSA metodu)

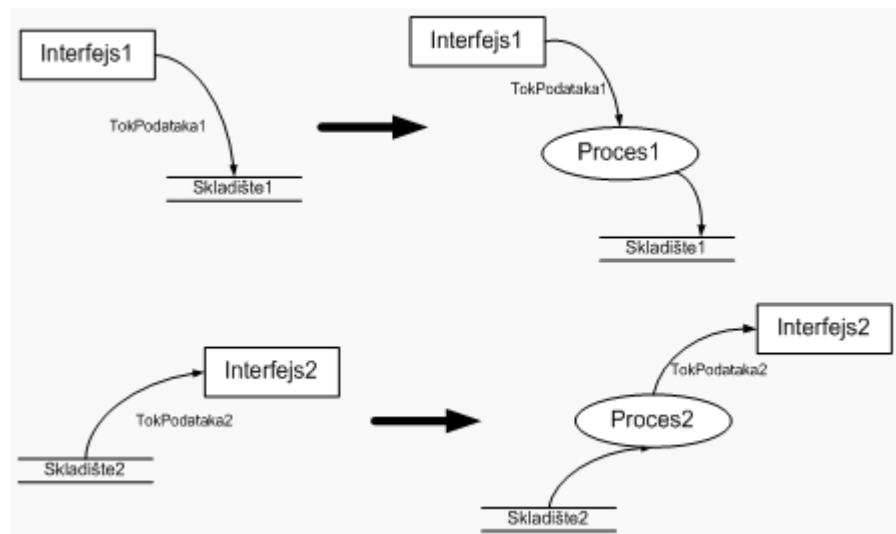
DTP predstavlja model sistema koji sadrži četiri osnovne komponente – koncepta za prikaz IS:

- funkcije, odn. procesi obrade podataka
- tokovi podataka,
- skladišta podataka i
- interfejsi.



Pravila za izradu dijagrama tokova podataka

1. Tok podataka mora da ima izvor i odredište. Izvor ili odredište može biti bilo koja druga komponenta (interfejs, proces, skladište podataka), ali za jedan tok bilo izvor bilo odredište (ili oba) mora biti proces (tokovima se ne mogu neposredno povezati dva interfejsa, dva skladišta ili skladište i interfejs)



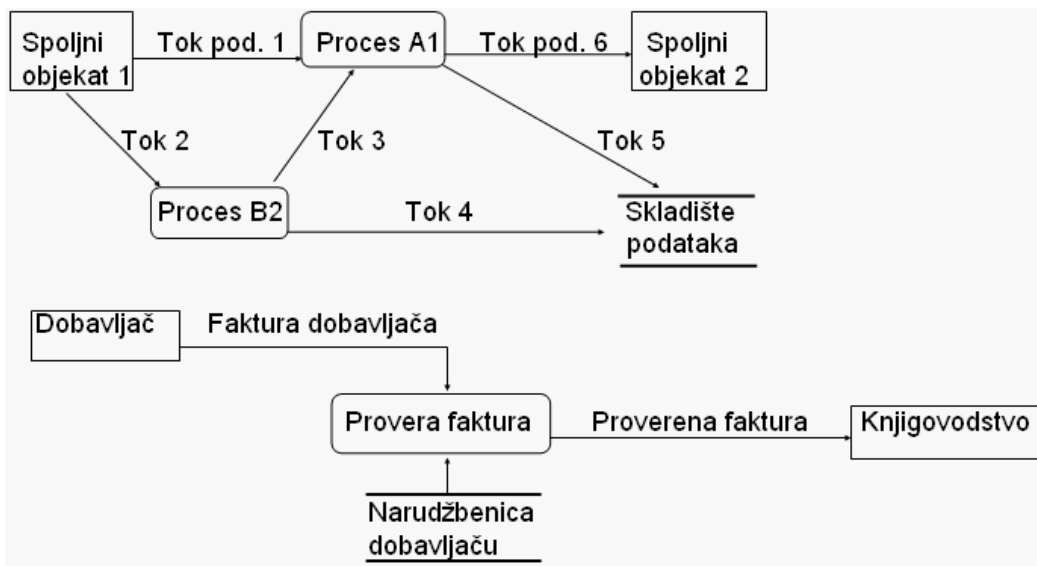
2. Svaki tok podataka mora imati ime, izuzev tokova koji idu od odn. ka skladištima podataka (tada se podrazumeva da takav tok nosi kompletan sadržaj skladišta podataka);



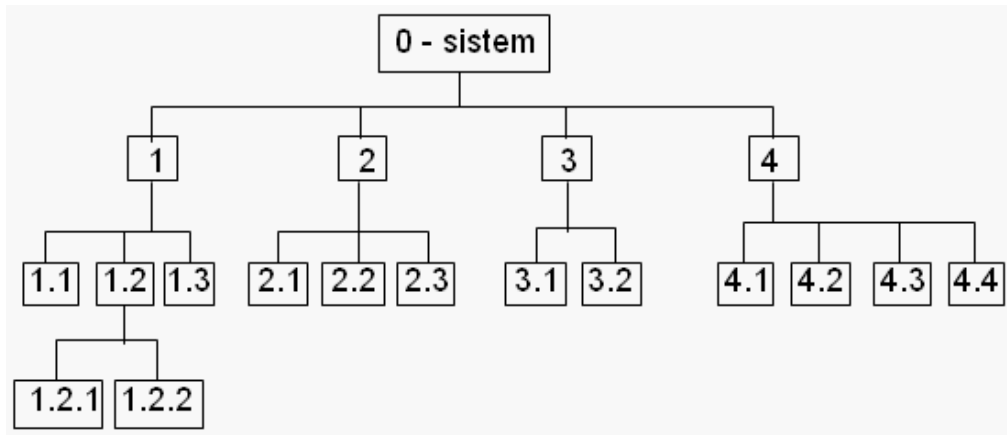
3. **Tok podataka se može granati** ili se daju dva toka sa istim imenom koja imaju isti izvor a različita odredišta;
4. **Svaki proces mora imati ime**, a poželjna je i referentna oznaka koja iskazuje hijerarhijsku pripadnost procesa pri dekompoziciji

5. **Svaki proces treba da ima bar jedan ulazni i bar jedan izlazni tok podataka** (proces bez ulaza stvara izlaz niizčega, a proces bez izlaza je nesvrsishodan);
6. **Skladište podataka može biti bez ulaznog toka** (podrazumeva se da se stvara i ažurira u nekom drugom sistemu) ili bez izlaznog toka (posmatrani sistem ga samo formira a koristi se u nekom drugom sistemu);
7. **Svaki interfejs mora imati bar jedan ulazni ili izlazni tok podataka**, inače bi bio izolovan od sistema;
8. **Dozvoljava se višestruko ponavljanje skladišta i interfejsa**, da bi se minimiziralo presecanje tokova podataka.

Primer dijagrama tokova podataka



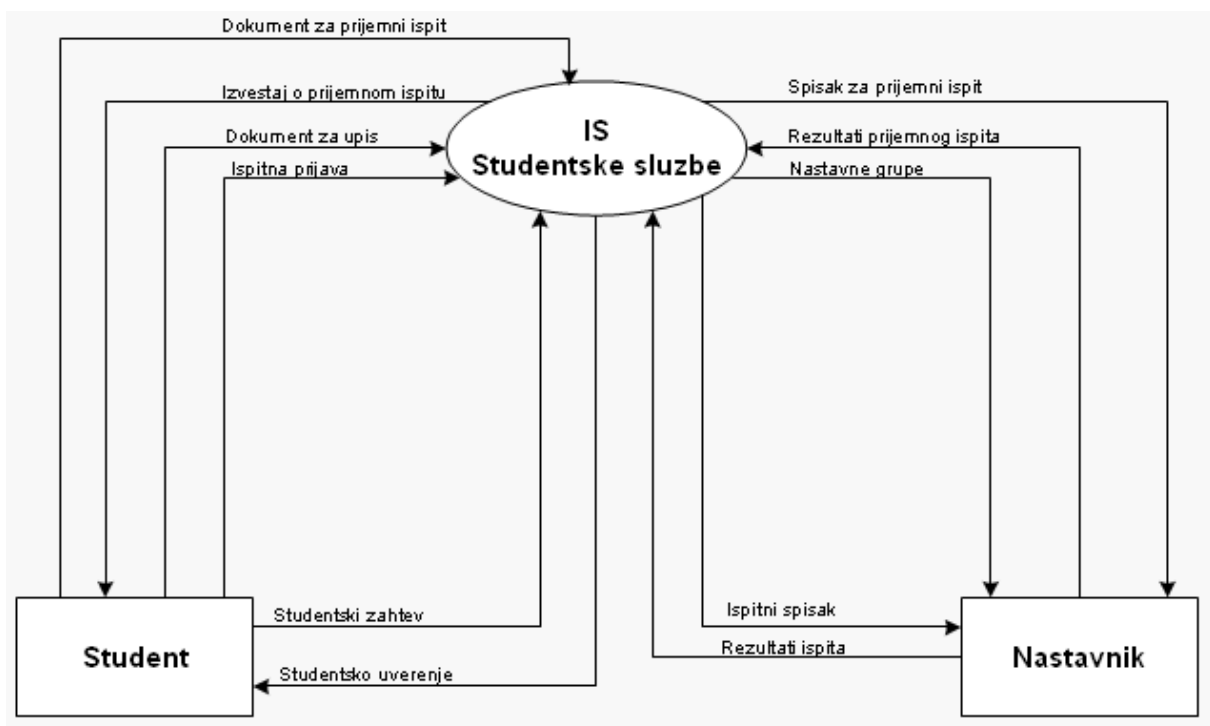
Dijagram dekompozicije sistema



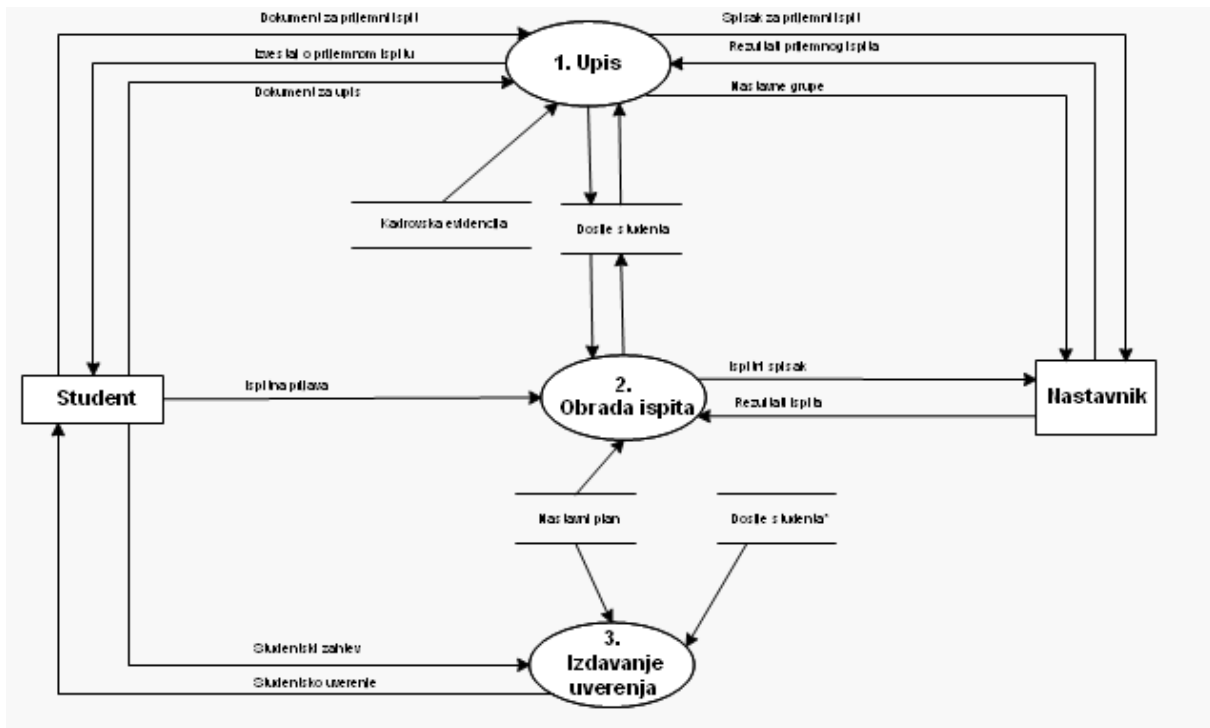
Pravila dekompozicije sistema

- Na najvišem (nultom) nivou je **dijagram konteksta** koji sadrži jedan proces koji predstavlja ceo sistem, interfejsa (objekte van sistema) i odgovarajuće tokove podataka između sistema i interfejsa (eksterne tokove podataka).
- Na prvom nivou dekompozicije je **Root dijagram - dijagram prvog nivoa** (dekompozicija dijagrama konteksta na osnovne procese, numerisane sa "1", "2",...).
- Proces koji se dalje ne dekomponuje su **elementarni** odn. **primitivni procesi**. Za njih se daje opis - specifikacija logike njihovog odvijanja, odnosno mini-specifikacija sistema.
- Pored procesa, mogu se dekomponovati i **tokovi i skladišta podataka**.
- **Pravilo balansa tokova**: ulazni i izlazni tokovi na celokupnom dijagramu toka podataka dobijenom dekompozicijom nekog procesa moraju biti ekvivalentni sa ulaznim i izlaznim tokovima tog procesa prikazanim na dijagramu višeg nivoa.
- **Skladišta podataka** od nivoa prvog pojavljivanja moraju se pojavljivati na svim nižim nivoima.
- Jedan dijagram toka ne bi trebalo da sadrži previše (recimo maksimalno desetak) procesa - ako procesa ima mnogo, verovatno je preskočen neki nivo dekompozicije.

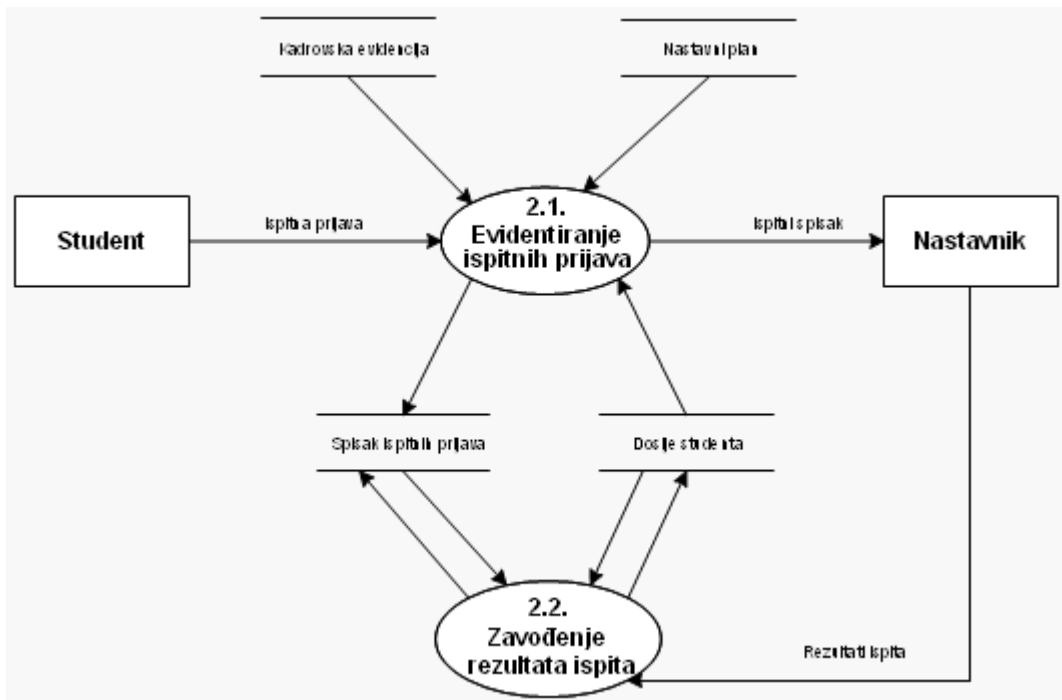
Primer dijagrama konteksta



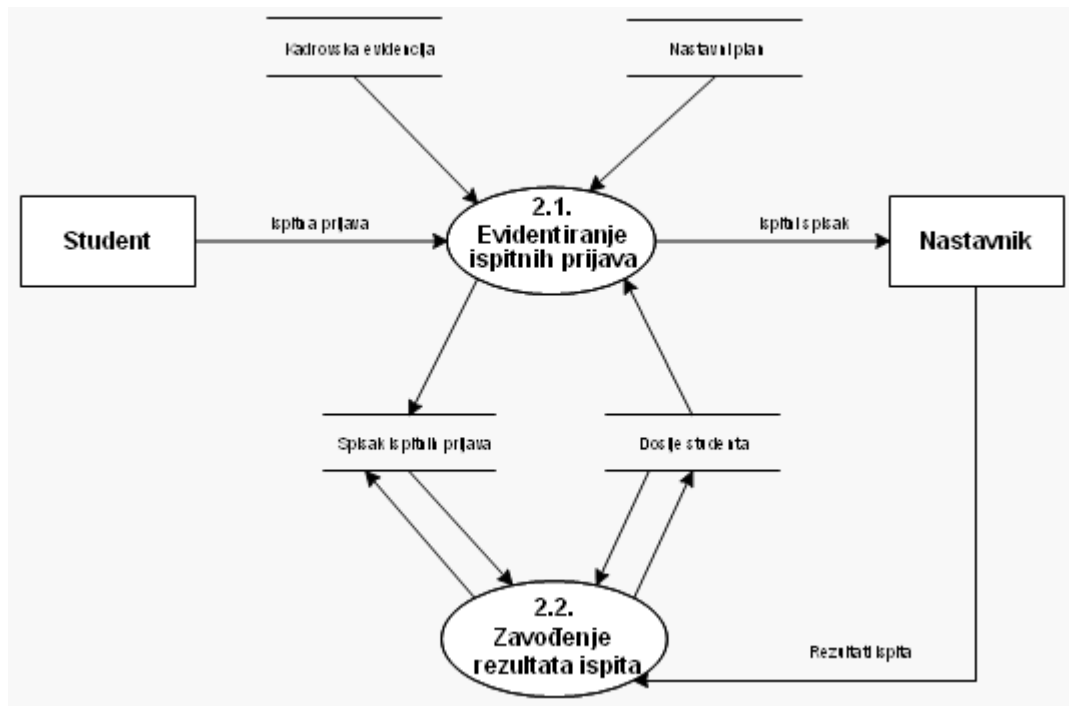
Primer root dijagrama



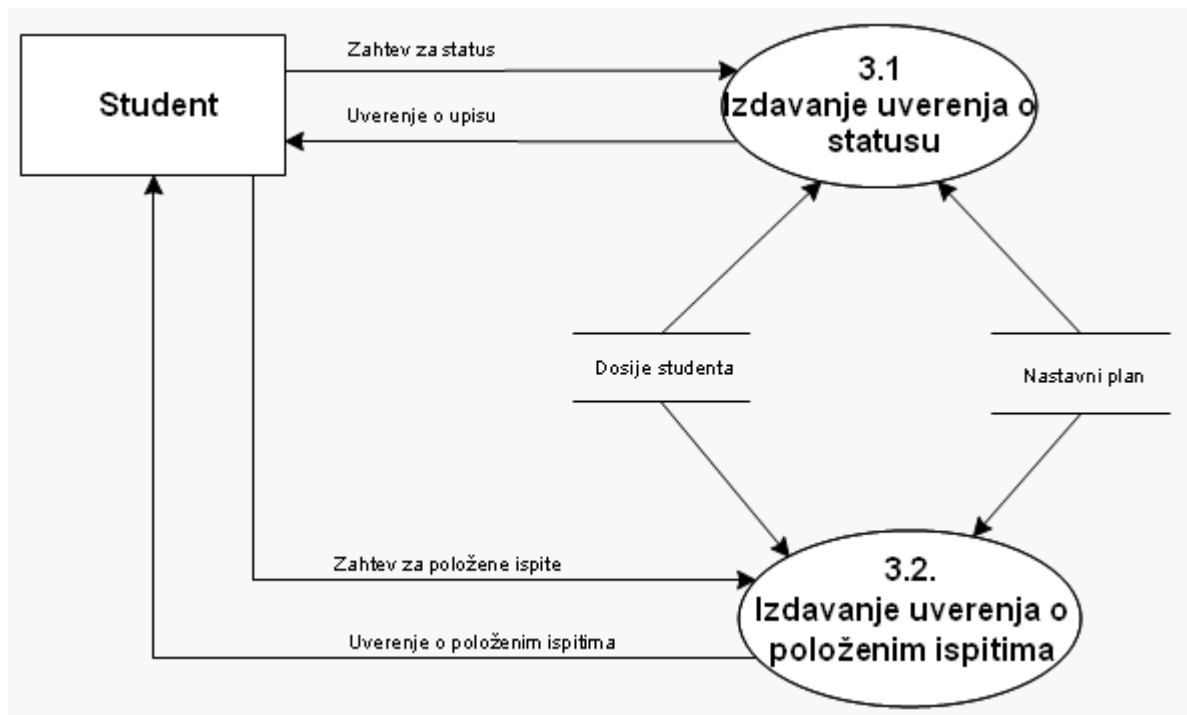
Primer dekompozicije procesa 1 (UPIS) – primitivni procesi



Primer dekompozicije procesa 2 (OBRADA ISPITA) – primitivni procesi



Primer dekompozicije procesa 3 (IZDAVANJE UVERENJA) – primitivni procesi



Analiza podataka

Analiza podataka omogućava lakše i bolje kreiranje i korišćenje baze podataka. Krajnji rezultat analize podataka treba da bude **model podataka** koji predstavlja osnovu za dizajn baze podataka.

Model podataka je intelektualno sredstvo za prikazivanje objekata sistema, njihovih atributa i međusobnih veza. On prikazuje stanje sistema u jednom trenutku vremena.

Baza podataka je *kolekcija međusobno povezanih podataka koja modeluje objekte, veze objekata i attribute objekata posmatranog sistema.*

Baza podataka je osnova na kojoj počiva informacioni sistem - fundamentalna i sporo promenljiva karakteristika sistema. Projektovanje informacionih sistema se zasniva na bazi podataka koja mora da predstavlja adekvatan model stanja realnog sistema.

Analiza podataka treba da omogući:

- da dizajn baze podataka bude stabilan i manje podložan promenama, čime se snižavaju troškovi;
- da opisi podataka budu kompletni i konzistentni, nezavisni od softvera;
- veću produktivnost u razvoju aplikacija i operativnoj obradi podataka.

Najviše korišćeni modeli podataka su relacioni modeli, s obzirom da postoje komercijalni **sistemi za upravljanje bazom podataka** (SUBP) koji su na njima zasnovani. Proces dolaženja do relacionog modela je poznat kao **relaciona analiza podataka** i odvija se u dve osnovne faze:

- formiranje modela entiteta;
- formiranje modela podataka.

Modelovanje entiteta

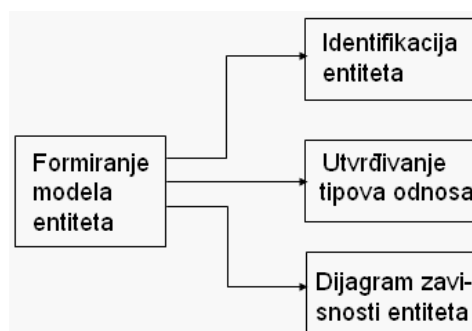
Entiteti predstavljaju neku posebnost, nešto o čemu informacioni sistem treba da sakuplja podatke.

Entiteti su poslovni resursi, odn. objekti posmatranja o kojima u informacionom sistemu treba prikupljati, memorisati i obrađivati podatke, sa ciljem izrade informacija.

Entiteti mogu biti realni, apstraktni, mogu biti predmeti (mašina, proizvod i sl), osobe ili događaji (npr. kupovina ili prodaja).

U fazi modelovanja entiteta vrši se sagledavanje načina odvijanja procesa i definisanje svih elemenata, tj. resursa ili entiteta stvarnog poslovanja. Rezultati se prikazuju preko modela entiteta.

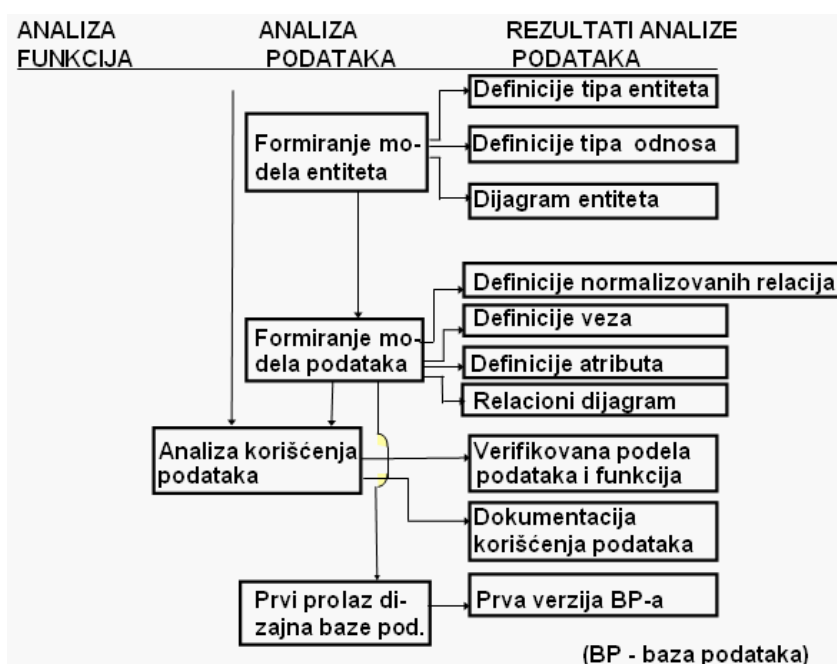
Model entiteta predstavlja viši nivo sagledavanja poslovnih podataka. Veoma je podesan za analizu i diskusiju sa korisnikom.



Koraci u formiranju modela entiteta

Faza razvoja	Poslovi	Oblasti
Definicija projekta	Formiranje modela entiteta	Analiza podataka
Definicija zahteva i potreba		
Eksterni dizajn	Formiranje modela podataka	
	Analiza korišćenja podataka	
	Eksterni dizajn baze podataka	Dizajn baze podataka
Interni dizajn	Interni dizajn baze podataka	
Implementacija		

Osnovni koraci u relacionoj analizi



Analiza podataka i njeni rezultati

Skup entiteta koji imaju neke zajedničke osobine zove se **klasa** ili **tip entiteta**.

Konkretni entitet u nekom tipu entiteta naziva se **pojavljivanje** ili **nastupajući element entiteta** (svako individualno pojavljivanje nekog tipa entiteta u stvarnom svetu).

Skup entiteta obuhvata sva pojavljivanja entiteta istog tipa, npr.:

Tip entiteta	Nastupajući element	Skup entiteta
Radnik	Radiša Radišić	Svi radnici
Automobil	KG=456-39	Svi automobili

Atributi (obeležja) karakterišu ili predstavljaju svojstva jednog entiteta.

U formalnom obliku entitet se prikazuje kao $E(a_1, a_2, a_3, \dots, a_n)$, gde su $a_1, a_2, a_3, \dots, a_n$ atributi. Za svaki nastupajući element, atributi imaju konkretne vrednosti i tako opisuju entitete.

Identifikator entiteta je atribut koji ima različitu vrednost za svaki nastupajući element, tako da jednoznačno identifikuje taj nastupajući element.

Pri utvrđivanju i identifikaciji tipova entiteta treba voditi računa o:

- poslovnim pravilima pripadnosti i
- životnom ciklusu entiteta.

Pravila pripadnosti definišu šta jeste a šta nije entitet određenog tipa. Npr kupac kod maloprodajnih preduzeća je fizičko lice, dok je kod proizvodnih ili veleprodajnih preduzeća kupac drugo preduzeće.

Životni ciklus entiteta određuje kada neki objekat posmatranja postaje element nekog skupa entiteta odn. kad prestaje to da bude, odnosno određuje nastanak, promene ili prestanak postojanja entiteta. Npr. za entitet MAŠINA životni ciklus može biti nabavka - popravke - rashodovanje.

Podtip (podskup) **entiteta** kao pojam obuhvata skupove koji imaju neke zajedničke osobine koje omogućavaju njihovo izdvajanje u podskupove a imaju i neke zajedničke osobine sa ostalima, pa su i tipovi entiteta, npr.:

Tip: STUDENT
Podtip: STUDENT INFORMATIKE

Način identifikovanja tipova entiteta

- Identifikovati osnovne tipove entiteta koji su karakteristični za konkretnu oblast poslovanja. To su entiteti koji su neophodni za poslovanje, karakterišu ga, pa se ponekad zovu i **ključni entiteti**.
- Za identifikovani tip entiteta utvrditi da li postoje podtipovi, tj. tipove podeliti na podtipove (npr. za tip MAŠINA podtip bi bio PRESA, STRUG i sl).
- Proveriti da li se za svaki tip entiteta može uvesti više nivoa (npr. za podtip PRESA može se uvesti nivo velike / srednje / male prese, zatim podela po proizvođačima itd).
- Sagledati kako entiteti komuniciraju jedan sa drugim, jer se u tom slučaju mogu otkriti novi tipovi entiteta (npr. iz odnosa entiteta PROIZVOD i SKLADIŠTE može se izvesti entitet STANJE ZALIHA).
- Istražiti ključne događaje u životnom ciklusu entiteta - u životnom ciklusu jednog tipa entiteta neki događaj može prouzrokovati nastajanje nekog drugog tipa entiteta (npr. događaj naručivanja u životnom ciklusu entiteta UGOVOR stvara tip entiteta PORUDŽBINA; realizacija porudžbine stvara tip entiteta FAKTURA itd).
- Voditi istorijske podatke o životnom ciklusu entiteta.

Definisanje odnosa među entitetima

Entiteti se međusobno nalaze u odnosima koje treba precizno definisati. Ti odnosi (ili veze odn. relacije) predstavljaju združivanja entiteta, koja se mogu predstaviti kao $R(E_1, E_2, \dots, E_m, a_1, a_2, \dots, a_n)$, gde je R oznaka relacije, $E_i, i = 1, 2, \dots, m$ su tipovi entiteta, a $a_j, j = 1, 2, \dots, n$ su atributi relacije.

Primeri relacija:

- Student POLAŽE ispit;
- Kupac ISPOSTAVLJA porudžbinu;
- Dobavljač ISPORUČUJE proizvod;

.....

Kao i kod entiteta, i kod njihovih odnosa mogu se razlikovati tipovi, skupovi i elementi skupova odnosa.

Tip odnosa opisuje sve elemente koji odgovaraju datoj definiciji odnosa.

Skup odnosa sadrži sva pojedinačna pojavljivanja odnosa koji spadaju u neki tip. Npr:

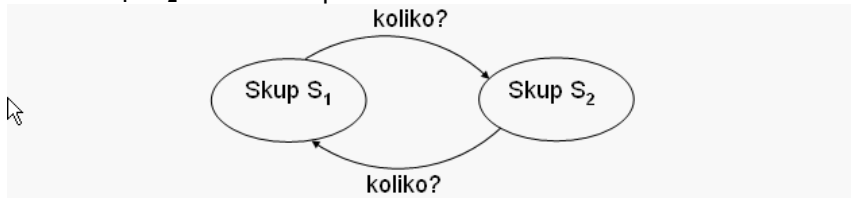
Tip odnosa	Nastupajući element	Skup odnosa
Student POLAŽE ispit	Jovan Jović polaže statiku	Polaganje statike
Čovek VOZI motorno vozilo	Rambo vozi kamion	Voženje kamiona

Prilikom utvrđivanja odnosa (veza) treba definisati:

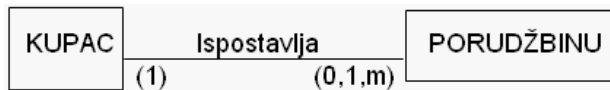
- ime koje određuje značenje odnosa;
- oznaku za kvantifikaciju.

Oznaka za kvantifikaciju podrazumeva kvantitativno određivanje odnosa u oba smera, tj. definiše broj pojavljivanja entiteta sa jedne strane za svaki nastupajući element sa druge strane odnosa. To znači da se utvrđuje koliko elemenata jednog skupa entiteta odgovara jednom elementu drugog skupa i obrnuto.

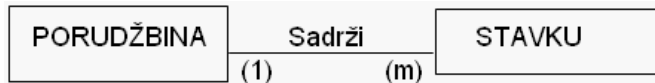
Kvantifikovanje odnosa može da se definiše kao dozvoljena kombinacija 0, 1 i m. Na primeru skupova entiteta S_1 i S_2 to se može prikazati šemom:



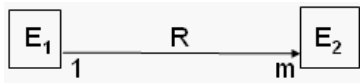
Ilustracija: odnos entiteta KUPAC i PORUDŽBINA. Jedan kupac ISPOSTAVLJA nijednu (0), jednu (1) ili više (m - many) porudžbina, ali se jedna porudžbina ODNOSI samo na jednog kupca:



Slično, porudžbina SADRŽI više stavki, ali se jedna stavka porudžbine SADRŽI u samo jednoj porudžbini:



U opštem slučaju, ova zadnja vrsta odnosa se može predstaviti kao:

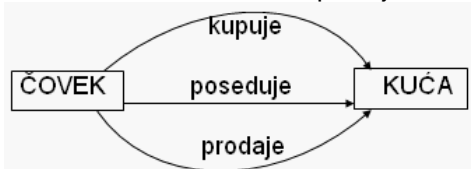


R označava naziv odnosa, E_1 i E_2 označavaju povezane entitete, a oznaka za kvantifikaciju je 1:m.

Veza može da se kvantifikuje kao **1:1** (jedan prema jedan), **1:m** (jedan prema više) i **m:m** (više prema više - sa obe strane može biti više nastupajućih elemenata). Moguć je i uslovni odnos tipa **1:c** (c - conditional odn. 0 ili 1 - jedan sa jedne strane, nijedan ili jedan sa druge strane).

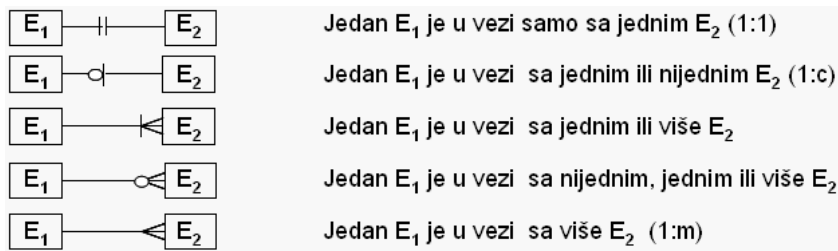


Između dva entiteta može postojati više tipova odnosa, na primer:



Uobičajeno se odnos tipa **1:m** predstavlja usmerenom linijom (\longrightarrow), sa strelicom na "m" strani, **m:m** "dvosmernom" linijom (\longleftrightarrow), **1:1** linijom bez strelica, **1:c** linijom bez strelica, sa oznakom (0,1) na "c" strani ($\text{---}(0,1)$).

U novijoj literaturi nalaze se i ovakve oznake:



Dijagram zavisnosti entiteta

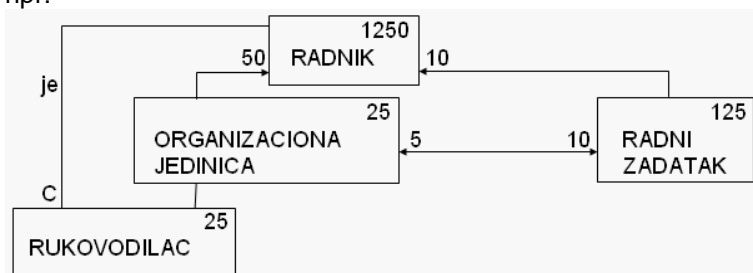
Model entiteta treba da sadrži tri sledeće komponente:

- definicije tipova entiteta;
- definicije odnosa među entitetima;
- dijagram zavisnosti entiteta.

Dijagram zavisnosti entiteta slikovito prikazuje do čega se došlo u definiciji tipova entiteta i odnosa među njima. U dijagramu zavisnosti entiteta svaki tip entiteta se prikazuje pravougaonikom u koji se upisuje ime entiteta. Međusobni odnosi se prikazuju pravim linijama koje spajaju po dva pravougaonika. Nazivi odnosa se mogu upisati iznad linije.

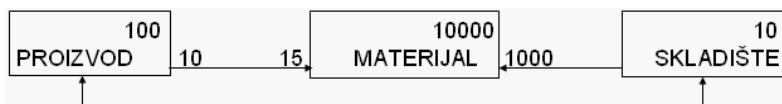
Veze se mogu predstavljati i linijama bez strelica ako se na krajevima daju slovne odn. bročane oznake (1, c, m) ili se koriste već navedene oznake iz novije literature. Prosečan broj nastupajućih elemenata nekog tipa entiteta može se upisati u gornji desni ugao oznake (pravougaonika) tog entiteta. Za određivanje te vrednosti treba uzeti prosečno trajanje životnog ciklusa nastupajućih elemenata određenog tipa entiteta u nekim vremenskim razmacima.

Oznaka za kvantifikaciju može se upisati pored odgovarajuće strelice u dijagramu zavisnosti entiteta, npr:



Redundantni odnosi među entitetima

Redundantnim se smatra onaj odnos između dva entiteta koji se može predstaviti kombinacijom drugih, već utvrđenih odnosa. To znači da ako se u dijagramu zavisnosti entiteta neki odnos može izostaviti a da se ista informacija može dobiti iz već postojećih odnosa, taj odnos je redundantan. Na primer:

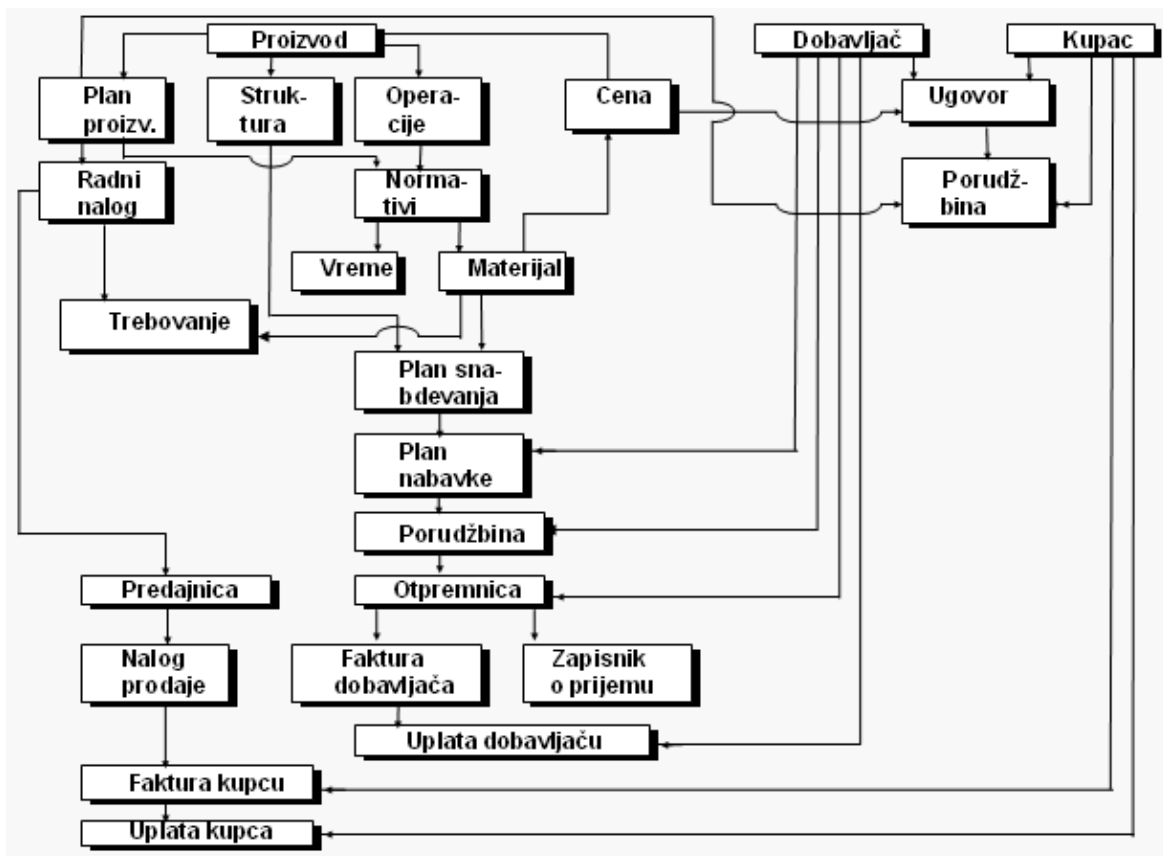


Veza (odnos) proizvoda i skladišta može se dobiti direktno ili preko entiteta MATERIJAL. Tu, dakle, postoji **stvarna redundancija** koju treba otkloniti.

Pored stvarne redundancije, postoji i **prividna redundancija**, veza koja se ne može drugačije predstaviti, jer ima drugo značenje od značenja koje se može dobiti iz ostalih postojećih odnosa na dijagramu zavisnosti entiteta. Tu vezu ne treba uklanjati. Na primer:



Primer dijagrama zavisnosti entiteta



Modelovanje podataka

Model podataka je strukturisani skup informacija o prošlosti i sadašnjosti sistema, potreban da bi se pod dejstvom budućih poznatih ulaza odredili budući izlazi iz sistema.

Model podataka je polazna osnova za projektovanje baze podataka i njene realizacije pomoću konkretnog SUBP-a.

Karakteristike modela podataka - model podataka treba da bude:

- semantički bogat konceptima (treba da se što više znanja o realnom sistemu unese u bazu podataka na eksplicitan, prirodan, konzistentan i nedvosmislen način - da se baza podataka učini što "inteligentnijom").
- relativan (treba da podrži relativizam različitih pogleda na realni sistem, i to kako iz različitih uglova posmatranja na istom nivou apstrakcije tako i na različitim nivoima apstrakcije).
- fizički nezavisan (treba da se zasniva na konceptima bliskim realnom sistemu, a ne na konceptima bliskim računaru kao što su zapis, pokazivač i sl.).
- pogodan za dalje projektovanje i realizaciju (to znači da su koncepti ograničeni i operatori tako definisani da omogućavaju formalno projektovanje logičke i fizičke baze podataka).
- takav da ga je moguće realizovati preko nekog sistema za upravljanje bazom podataka.

U pogledu **pristupa modelovanju**, koristi se:

- Direktno modelovanje na osnovu poznavanja sistema i zakonitosti koje u njemu vladaju, odnosno na osnovu poznavanja objekata sistema, njihovih atributa i veza (tzv. dugoročni modeli);
- Identifikacija na bazi poznavanja ulaza i izlaza informacionog sistema (tzv. kratkoročni modeli); pri modelovanju na bazi poznavanja ulaza i izlaza mogu se koristiti Warnierovi dijagrami (inače sredstvo za projektovanje programa) i kao model podataka.

- Analiza i sinteza (postupku analize odgovara pristup od vrha na dole - Top Down, a postupku sinteze pristup odozdo na gore - Bottom Up).

Definicije:

Šema modela podataka je skup klasa ili kategorija, njihovih osobina i veza između njih.

Baza podataka je skup podataka strukturisanih na način opisan u šemi. To je skup svih pojavljivanja entiteta navedenih tipova (kategorija), skup svih njihovih osobina i skup veza između njih.

Jezik za definiciju podataka (Data Definition Language - DDL) predstavlja skup generičkih pravila **G** pomoću kojih je moguće definisati (opisati) šemu.

Model podataka je jedinstveni način predstavljanja skupa podataka i njihove interpretacije preko strukture podataka, skupa ograničenja i skupa operatora.

Model podataka čine tri osnovne komponente:

- struktura;
- ograničenja;
- operatori.

Osnovni problem u modelovanju (pa i u formiranju modela podataka) je savlađivanje složenosti u opisu sistema. Složenost se savlađuje hijerarhijskim opisom sistema, što znači da se na višim nivoima strukture daje jasniji opšti opis sistema, a na nižim nivoima detaljniji opis. Takav pristup u modelovanju naziva se apstrakcija.

Apstrakcija je kontrolisano uključivanje detalja u opis sistema, izostavljanje detalja u pojedinim fazama (na pojedinim nivoima apstrakcije) i predstavljanje samo zajedničkih opštih osobina pojedinih koncepata. Apstrakcijom se zanemaruju nebitne osobine realnog sistema a takođe i nebitne veze, tako da se u model unose samo one opšte, bitne jer model treba da odslika realni sistem.

Apstrakcija se ostvaruje postupcima agregacije i generalizacije.

Agregacija je tip apstrakcije u kojoj se skup objekata i njihovih veza posmatra kao objekat višeg nivoa.

Generalizacija je tip apstrakcije u kojoj se skup sličnih objekata tretira kao generički objekat.

Inverzni postupak od generalizacije je **specijalizacija**.

Postupkom generalizacije se skup objekata iste vrste predstavlja odgovarajućim tipom entiteta (npr. radnik, direktor, sekretarica, inženjer,... se agregiraju u tip entiteta ZAPOSLENI). Osobine entiteta opisuju se atributima a atributi uzimaju vrednosti iz odgovarajućih domena, pri čemu se domen definiše kao homogeni skup podataka. Atributi se, s druge strane, agregiraju u tip entiteta (npr. Pera Perić, Lepa Lukić, Jova Jović,... čine atribut IME, Užička 13, Pavlova 23, Glavna 116,... čine atribut ADRESU, itd). Dakle, tip entiteta se dobija generalizacijom skupa pojedinačnih entiteta iste vrste ili agregacijom skupa atributa.

Generalizacija se može koristiti za dalje uopštavanje i stvaranje tipova opštijeg (višeg) nivoa a agregacija za grupisanje postojećih tipova entiteta u nove tipove. Ukoliko se agregiraju atributi različitih objekata, takva agregacija se može tretirati kao stvaranje "veze" između tih objekata.

I agregacija i generalizacija imaju jednostavnu matematičku interpretaciju i opisuju se preko **relacija**. Matematička relacija se definiše kao skup koji opisuje korespondenciju (agregaciju) dva ili više skupova. Korespondencija između dva skupa, S_1 i S_2 , naziva se binarna relacija i opisuju se skupom parova, " s_1, s_2 " koji su u korespondenciji.

Relacija se definiše kao $R \subseteq S_1 \times S_2 = \{ \langle s_1, s_2 \rangle, s_1 \in S_1, s_2 \in S_2 \}$

Na sličan način se definiše relacija n-tog stepena: $R \subseteq S_1 \times S_2 \times \dots \times S_n = \{ \langle s_1, s_2, \dots, s_n \rangle, s_1 \in S_1, s_2 \in S_2, \dots, s_n \in S_n \}$, pa se kaže da je relacija skup uređenih n-torki.

1. **DOMEN** je homogeni skup podataka:

- intenzija domena odnosno imenovani domen se naziva atribut,
- ekstenzija domena odgovara vrednostima atributa.

2. **ENTITET** je relacija odnosno agregacija vrednosti nekih atributa:

- ekstenzija ove relacije (agregacije) je skup entiteta,
- intenzija ove relacije (agregacije) je tip entiteta.

3. **TIPOVI ENTITETA** se dalje mogu uz pomoć relacije agregirati u nove tipove entiteta.

4. TIPOVI ENTITETA se dalje mogu generalizovati u nove tipove entiteta na višem nivou, pa se tipovi na nižem nivou posmatraju kao ekstenzija a tip na višem nivou kao intenzija odgovarajućeg skupa, odnosno **relacije**.
5. VEZA je relacija u kojoj pojedini atributi pripadaju različitim objektima realnog sistema:
 - ekstenzija ove relacije je skup veze,
 - intenzija ove relacije je tip veze.
6. TIPOVI VEZA se mogu dalje opisivati na višim nivoima apstrakcije preko njihove generalizacije, npr. veza PARTNERSTVO je generalizacija veza *poslovno partnerstvo*, *bračno partnerstvo* itd.

Kao i svaki skup, relacija se može opisati na dva načina:

- preko svoje ekstenzije odn. skupa elemenata, npr. $X = \{1, 3, 5, 7, 9\}$ - skup prirodnih neparnih brojeva od 1 do 10;
- preko svoje intenzije odn. komprehenzije (definisanjem karakteristične zajedničke osobine skupa elemenata, npr. $X = \{x \mid x \text{ je neparan ceo broj manji od } 10\}$)

Komponente modela podataka

Struktura podataka

U opštem slučaju, pod strukturom neke celine podrazumevaju se elementi koji tu celinu čine kao i veze koje između tih elemenata postoje. Osnovni elementi strukture modela podataka su **entiteti**.

Objekti / entiteti imaju svoje karakteristične osobine koje se zovu **atributi**. Formalno se prikazuju kao $E = E(a_1, a_2, \dots, a_n)$, gde je E ime entiteta a $a_i, i = 1, 2, \dots, n$ atributi.

Svaki od atributa u konkretnom slučaju ima svoju vrednost. Te vrednosti atributi uzimaju iz odgovarajućeg **domena**. Zato se i kaže da je domen skup veličina iz koga atributi uzimaju konkretne vrednosti i da je atribut imenovani domen. Jedan atribut može da ima samo jedan domen, ali može da bude više različitih atributa zadato nad istim domenom.

Atributi se identifikuju na osnovu:

- zahteva korisnika;
- poslovne dokumentacije;
- dokumentacije informacionog sistema.

Svaki entitet može imati proizvoljan broj atributa, dok obrnuto ne važi - određeni atribut može pripadati samo jednom entitetu.

Svako pojavljivanje entiteta ima vrednosti za sve attribute. Može se dogoditi da neka pojavljivanja ostanu bez vrednosti atributa, pa kažemo da imaju nultu vrednost (npr. nepopunjena radna mesta po organizacionoj šemi imaju nulte vrednosti atributa).

Jedan ili više atributa koji jednoznačno određuju svaki entitet u klasi entiteta predstavlja **ključ klase entiteta**. **Ključ** je atribut (ili skup atributa) koji je jednoznačan i nepromenljiv. Zato kriterijumi za izbor ključa treba da budu: jednoznačnost, nepromenljivost i raspoloživost.

Između elemenata strukture postoje određene **veze** odn. **relacije**. Dve klase entiteta mogu biti povezane relacijama. Veza odn. relacija je određeno združivanje među entitetima i predstavlja se kao

$R(E_1, E_2, \dots, E_m, a_1, a_2, \dots, a_n)$, gde su: $E_i, i = 1, 2, \dots, m$ klasa entiteta,

$a_j, j = 1, 2, \dots, n$ atributi relacija (i relacije imaju svoja obeležja - attribute).

Skup svih relacija nad istim klasama entiteta zove se **klasa** ili **vrsta relacije**.

Osobine preslikavanja definišu se preko **kardinalnosti preslikavanja**.

Za dva skupa, S_1 i S_2 , **kardinalnost preslikavanja** (asocijacija $S_1 \rightarrow S_2$) predstavlja broj elemenata skupa S_2 koji se mogu pridružiti jednom elementu skupa S_1 .

Razlikuju se sledeće vrste asocijacija:

- Prosta asocijacija (tip 1), gde jednom elementu skupa S_1 odgovara samo jedan element skupa S_2 ;
- Uslovna asocijacija ili parcijalno preslikavanje (tip C), gde jednom elementu skupa S_1 odgovara jedan ili nijedan element skupa S_2 ;

- Složena ili višeznačna asocijacija (tip M), gde jednom elementu skupa S_1 odgovara nijedan, jedan ili više elemenata skupa S_2 .

Ista podela važi i za inverzna preslikavanja, tj. i ona mogu biti tipa 1, C ili M, tako da se veze između atributa prikazuju kao 1:1, 1:C, C:C ili M:M.

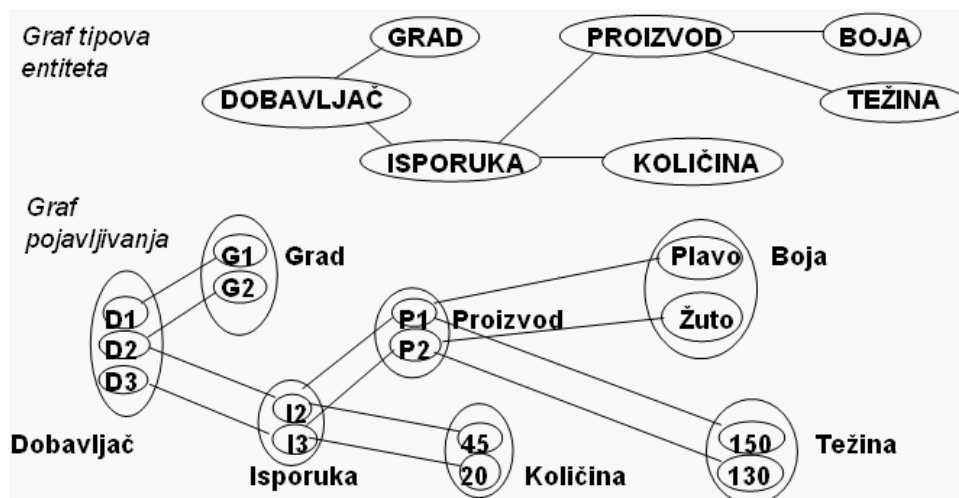
Predstavljanje strukture podataka preko tabele:

RADNIK	Mat. broj	Ime i prezime	Org. celina	Pogon	Kvalifikacija
	23859	Petar Petrović	102300	1	maš.tehničar
	31310	Laza Lazarević	104100	1	VK brusač
	28965	Milanka Marković	123460	2	VK strugar
	34333	Stevan Jović	123460	1	maš.inženjer
	29856	Jovan Jovanović	158000	1	KV zavarivač

agregacija atributa u entitete
generalizacija skupova vrednosti u attribute i entiteta u tip entiteta

Predstavljanje strukture podataka preko grafa:

Čvorovi grafa predstavljaju objekte, odnosno entitete ili attribute, a grane grafa predstavljaju veze između entiteta i/ili atributa.



Ograničenja modela podataka

Predstavljaju svojstva entiteta, atributa i veza koja omogućavaju da se što adekvatnije opiše realni sistem i sa druge strane očuva integritet baze podataka, tj. ona omogućavaju da SUBP održava samo dozvoljeno stanje baze podataka. Zato se za njih ponekad susreće i naziv **pravila integriteta**.

Ograničenja modela podataka predstavljaju logičke granice postavljene kako na pojedine podatke tako i na njihove međusobne veze. Zato treba definisati skup ograničenja kojima se iskazuju dozvoljena stanja sistema i dozvoljeni načini prelaska iz stanja u stanje.

- **Ograničenja prve vrste** mogu biti *pravila kojima se specificira dopustivo stanje atributa*. Mogu da se odnose na ograničenja vrednosti atributa (npr. fizičko stanje zaliha u skladištu ne može biti negativno). S obzirom da atributi uzimaju vrednosti iz domena, ograničenja podataka zavise od ograničenja domena. Međutim, ograničenja mogu da zavise i od vrednosti podataka.
- **Druga vrsta ograničenja** iskazuje *dozvoljena stanja prelaza* i predstavlja *ograničenja veza* u vidu minimalnog i maksimalnog broja veza. Na primer, u vezi "RADI NA" između entiteta RADNIK i POSAO može se navesti koliko najmanje i najviše radnika radi na jednom poslu. Ova ograničenja nazivaju se i *kardinalnost veze*.

Ograničenja mogu biti sastavni deo strukture - u tom slučaju se radi o **inherentnim (implicitnim) ograničenjima**, ali mogu biti i **eksplicitna** (ne mogu se ugraditi u strukturu već se moraju eksplicitno izraziti).

Ograničenja u odnosu na vrstu veze lakše se ugrađuju kao inherentna ograničenja u model podataka prikazan grafom nego u model prikazan tabelom. Na primer, veze tipa 1:m i m:m između dva objekta mogu se obuhvatiti grafom odnosno samom strukturom, dok se u tabelarnom prikazu tabele koje ih sadrže moraju razložiti na posebne tabele.

Inherentna ograničenja ne zahtevaju posebno eksplicitno iskazivanje, već se ugrađuju u samu strukturu.

Sa druge strane, ako su ograničenja eksplicitno iskazana, lakše je prilagoditi model realnom sistemu, jer se dozvoljava određena sloboda u strukturi.

Operatori

Operatori su komponente modela podataka kojima se iskazuje dinamičnost realnog sistema, a time i prelazak baze podataka iz jednog stanja u drugo. Da bi se izvršila bilo koja operacija nad podacima u bazi podataka, mora prvo da se izvrši selekcija dela baze podataka nad kojim će te operacije da se izvrše.

U zavisnosti od selekcije dela baze podataka, razlikuju se sledeće vrste operatora:

- **navigacioni** (ako selekcija odabira pri svakom pozivu samo jedan entitet ili vezu (zapis, n-torku) na taj način što sledi logički put (navigaciju) kroz strukturu baze podataka, odgovarajući operatori se nazivaju navigacioni operatori).
- **specifikacioni** (biranje (selekcija) može da se specificira logičkom pozicijom, što znači da podaci mogu da se izdvoje prema logičkoj poziciji u šemi odnosno tabeli ili grafu, tako da možemo da izdvojimo npr. prvi ili zadnji red u tabeli ili grafu:
 - na osnovu veza između podataka;
 - na osnovu tekućih vrednosti.
- **kombinovani operatori** (kombinacija navigacionih i specifikacionih).

Ako selekcija odabira grupu objekata (entiteta) ili veza pri jednom pozivu i od njih formira (specificira) nove strukture baze podataka, odgovarajući operatori se nazivaju **specifikacioni operatori**. Operatori specifikacije u sistemu definišu novu strukturu (podmodel) koji treba izvući iz datog modela. Operatori definišu novu tabelu iz skupa tabela koje su pod dejstvom operatora.

Navigacionim i specifikacionim operatorima se izdvajaju pojedinačni podaci ili manji delovi baze podataka, nad kojima se mogu izvršiti operacije kao što su:

- ažuriranje;
- sažimanje baze podataka;
- izdvajanje vrednosti koje nisu eksplicitno sadržane u bazi podataka (sume, proseci, minimalne i maksimalne vrednosti i sl).
- izvršavanje radnih procedura za očuvanje integriteta baze podataka i sl.

Vrste modela podataka

Prva generacija:

- funkcionalni,
- hijerarhijski,
- mrežni,
- klasični relacioni modeli
- Warnierovi dijagrami.

Po opštim karakteristikama, to su:

- semantički nedovoljno bogati modeli, bez molekularne semantike;
- relativno fizički zavisni;
- realizovani pomoću komercijalnih sistema za upravljanje bazom podataka.

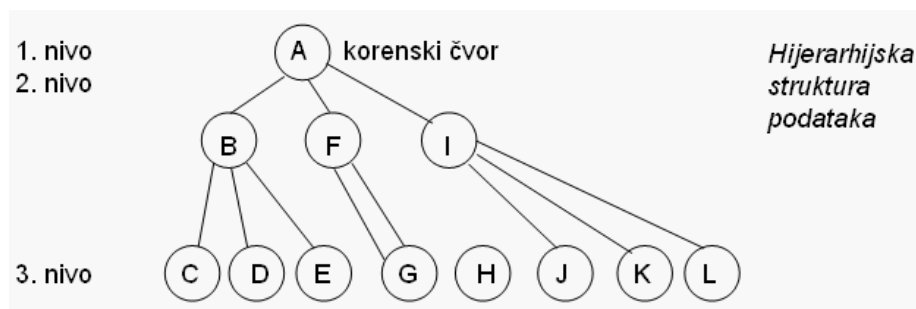
Druga generacija:

- model objekti - veze,
- prošireni relacioni model,
- semantičke mreže,
- SDM model.

Njihove opšte karakteristike su da su semantički bogati, sa molekularnom semantikom, fizički nezavisni i da još nisu realizovani pomoću komercijalnih sistema za upravljanje bazom podataka.

Hijerarhijski model podataka

Hijerarhijski model podataka koristi se od 60-tih godina. Podaci se organizuju u vidu hijerarhijske strukture koja podseća na stablo, sastavljene od čvorova i grana. Hijerarhijsko stablo je u stvari "prevrnuto stablo", kod koga je "koren" na vrhu a grana se naniže.



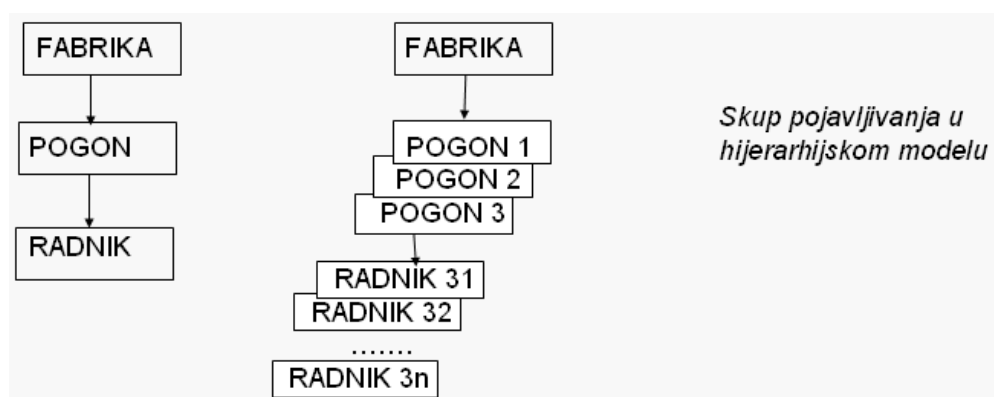
Čvor na vrhu je *korenski (Root) čvor* i to je nezavisni čvor a zavisni čvorovi su na nižim nivoima.

U hijerarhijskom modelu podataka čvorovi predstavljaju skupove funkcionalnih *segmenata* (povezanih elemenata podataka) ili *zapisa*, kojima se opisuju objekti sistema (zapisi se koriste u logičkoj, a segmenti u fizičkoj strukturi). Segment na korenskom čvoru naziva se *korenski segment*. Svi drugi segmenti svrstani su u hijerarhijske nivoe, sa vezama između segmenata na susednim nivoima.

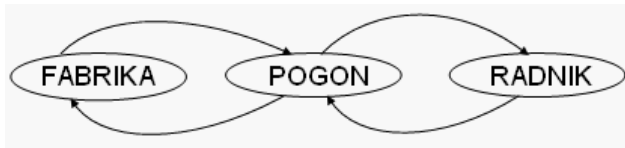
Ako se posmatraju dva povezana segmenta na susednim nivoima, segment na višem nivou naziva se "*roditelj*" a za njega vezani segmenti nižeg nivoa su "*deca*". Veza segmenta tipa "roditelj" sa segmentima tipa "deca" je 1 : m.

Pristup svakom segmentu izuzev korenskom segmentu vrši se preko segmenata "roditelja" - pristup svakom segmentu je jedinstven, odozgo na dole niz hijerarhiju stabla (npr. do segmenta K na prethodnoj šemi se dolazi isključivo putanjom A-I-K). Pristupni putevi su tačno određeni.

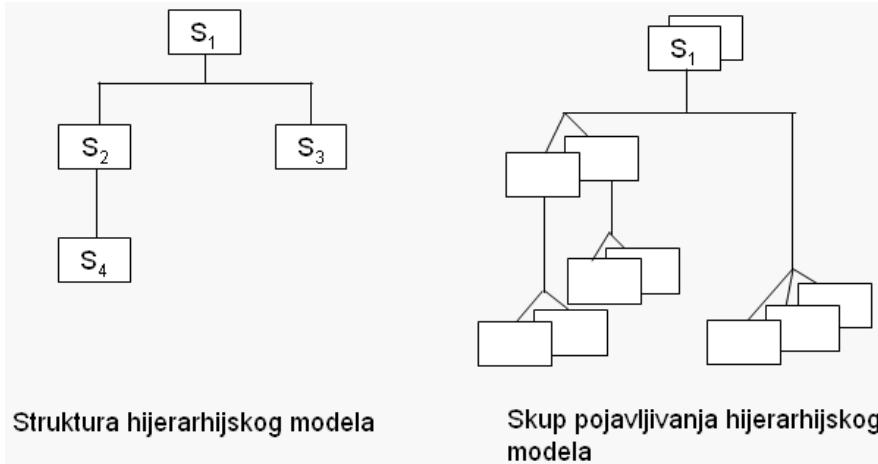
Kako su podaci organizovani ilustruje sledeća šema, na primeru entiteta FABRIKA, POGON i RADNIK.



Odnosi između navedenih entiteta mogu se prikazati kao:



Preslikavanje FABRIKA - POGON i POGON - RADNIK je tipa m (višestruko) a u obrnutom smeru tipa 1, tj. POGON - FABRIKA i RADNIK - POGON je jednostruko preslikavanje; to je veza 1 : m (roditelji - deca).



Dobre strane:

- korisnici lako shvataju hijerarhijsku strukturu kao prirodan opis realnog sistema, pa se i razumevanje sistema lako postiže;
- omogućavaju postepeno povećavanje detaljnosti u razradi sistema;
- postoji standardizovana navigacija;
- postoji sistem za upravljanje bazom podataka (SUBP) koji je baziran na hijerarhijskom modelu (npr. IBM-ovi sistemi IMS - Information Management System i DL/I - Data Language 1)

Nedostaci:

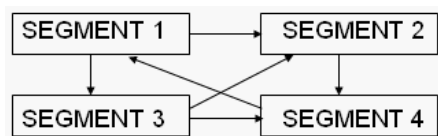
- do zapisa na nižem hijerarhijskom nivou može se doći samo preko svih zapisa na višim hijerarhijskim nivoima;
- česta pojava redundantnih podataka;
- anomalije u operaciji ažuriranja (pri dodavanju i brisanju podataka).

Hijerarhijski model podataka se sve ređe koristi.

Mrežni model podataka

Ovaj model predstavlja skup međusobno povezanih podataka u kome preslikavanje između segmenata principijelno može biti bilo kog tipa. Mrežni model se zasniva na orijentisanim grafovima opšteg tipa čiji čvorovi u stvari predstavljaju klase entiteta, pri čemu je klasa entiteta skup nastupajućih elemenata, a linije grafa predstavljaju odgovarajuće veze (npr. izlazak studenta na ispit - polaganje predmeta).

Grafički se mrežni model može predstaviti kao:



Segmenti (zapisi) su prikazani pravougaonicima a linije sa strelicama prikazuju veze među segmentima. Jedan isti objekat može biti i nadređeni i podređeni, za razliku od hijerarhijskog, u kome je samo korenski objekat nezavisan (nadređen svima ostalima) a ostali su uređeni po hijerarhijskim nivoima.

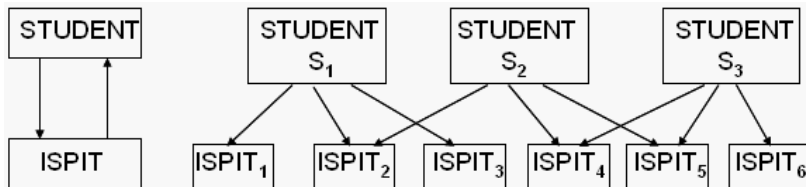
Kod mrežnog modela se pojam nadređenih i podređenih segmenata širi.

U mrežnim modelima se uvodi pojam **seta**, preko koga se iskazuju veze između segmenata.

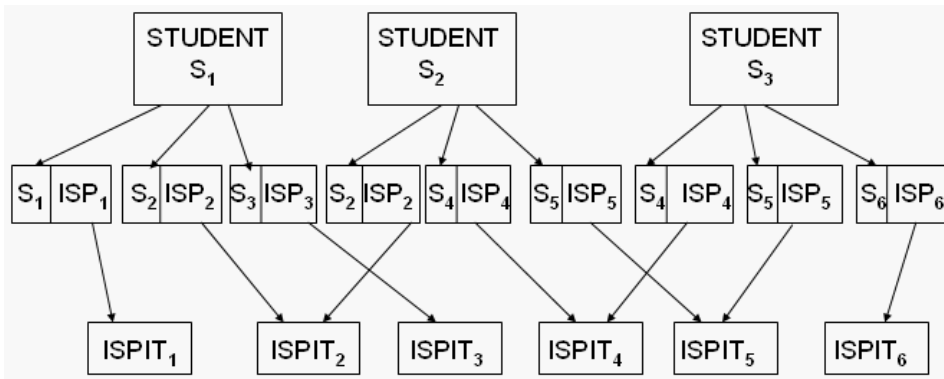
Set je takav tip veze između dva segmenta kod koga jednom pojavljivanju jednog segmenta odgovara jedno ili više pojavljivanja drugog segmenta. Svaki segment ima svoje ime koje označava tip entiteta a usmerena linija (strelica) označava vezu među segmentima.

Segment od koga polazi usmerena linija naziva se "vlasnik" (Owner) seta, a segment na koji pokazuje strelica je "član" (Member) seta. Na taj način tip seta ustvari predstavlja logičku vezu 1:m.

Primer: veza STUDENT - ISPIT, tipa m:m (jedan student može da polaže više ispita, na jedan ispit izlazi više studenata):



Veze tipa m:m su nepoželjne i teško ih je kontrolisati, pa se radi njihove eliminacije uvodi i treći tip segmenata, tzv. "vezni" (X) segment. Preko veznih segmenata se veze tipa m:m prevode u vezu m:1 (posmatrano od veznog segmenta) pa u vezu 1:m, kao na sledećoj šemi:



Dobre strane:

- izbegavanje redundantnosti podataka;
- izbegnute su anomalije pri ažuriranju podataka (način realizacije preslikavanja m : m je takav da se izbegavaju problemi koji su se javljali kod hijerarhijskog modela);
- gotovo uvek se može postići saglasnost strukture upita i strukture modela;
- niz komercijalnih sistema za upravljanje bazom podataka (npr. DMS 1100 korporacije Univac, TOTAL korporacije Cincom itd).

Nedostaci:

- pogodniji su za realizaciju na računaru, nego za opis realnog sistema (bliži računaru nego realnom sistemu);
- teža navigacija i aplikativno programiranje, jer aplikativni programi moraju da "poznaju" logičku organizaciju baze podataka kako bi moglo da se pristupi odgovarajućim setovima i segmentima;
- mrežne baze podataka koje su bazirane na mrežnom modelu se sastoje iz velikog broja segmenata – zapisa koji sadrže malu količinu informacija i veliki broj pokazivača prema drugim skupovima zapisa, što mrežni model čini složenim za implementaciju i loše utiče na performanse mrežne BP

Relacioni model podataka

Osnovne karakteristike:

- omogućava vrlo jednostavan i prirodan prikaz podataka, jer se model predstavlja skupom tabela, a tabele su prikaz strukture koji je najprihvatljiviji za komunikaciju sa korisnikom, pre svega zbog svoje preglednosti i jednostavnosti;

- moguća je formalna matematička interpretacija modela; tabela se može definisati kao matematička relacija.

Zbog mogućnosti da se formalno matematički interpretira celokupna teorija modela podataka, projektovanje logičke baze zasniva se najčešće na ovom modelu.

U relacionom modelu podaci se predstavljaju skupom **tabela** odn. **relacija** čiji je sadržaj predstavljen specifičnom optimalnom formom.

Podaci su na osnovu semantičkih svojstava svrstani u **n-torke**, npr. entitet RADNIK (matični broj, ime i prezime, kvalifikacija, godina zaposlenja,...) je jedna uređena n-torka.

U opštem slučaju, n-torke se posmatraju kao uređeni skupovi unutar kojih podaci stoje u tačno određenim odnosima. Definicija tih odnosa naziva se **relacija**. Dakle, kod relacionih modela podaci se daju relacijama odn. tabelama a model se prikazuje kao skup tabela. Na primer, relacija RADNIK može da izgleda kao:

Mat.br.	Ime i prezime	Pol	Kvalifikac.	God.zap.	God.rođ.
38181	Pera Perić	M	KV glodač	1975.	1950.
48506	Jovan Jovičić	M	gitarista	1980.	1935.
50333	Milan Panić	M	maš.inženjer	1982.	1955.
56891	Slavica Trifunović	Ž	dipl.ekon.	1982.	1958
...

Atributi za svaki red u relaciji (tabeli) imaju određene vrednosti, koje uzimaju iz odgovarajućih **domena**, tako da su kolone u tabeli ustvari domeni odgovarajućih atributa.

Broj domena nad kojima je relacija definisana odn. broj atributa relacije ili broj kolona odgovarajuće tabele definiše red relacije. Tako razlikujemo relacije prvog reda ili **unarne relacije** (jedan atribut), relacije drugog reda (**binarne relacije** - dva atributa) ili relacije n-tog reda odn. **n-arne relacije** (n atributa).

Razlikuju se i:

- nastupajući elementi relacije (pojedini element ili slog sadrži vrednost za svaki od atributa i to je red relacije ili n-torka relacije);
- ključ (atribut ili grupa atributa odn. složeni ključ koji ima različite vrednosti za svaki nastupajući element relacije - red tabele; u tabeli na prethodnoj slici to je matični broj).

Ključ relacije je primarni atribut ili grupa atributa (složeni ključ koji ima različite vrednosti za svaki nastupajući element relacije, odn. red tabele; u prethodnoj tabeli to je matični broj). Ključ relacije odn. **primarni ključ** se bira između svih potencijalnih ključeva ili kandidata za ključ, a potencijalni ključevi su oni atributi ili grupe atributa koji imaju različite vrednosti za svaki nastupajući element.

Karakteristike:

- ne postoje jednaki redovi u tabelama;
- poredak redova nije važan;
- poredak kolona nije važan; pri tome svaka kolona ima svoje ime.
- sve vrednosti u tabeli imaju "atomske" karakter tj. ne mogu se razbiti na komponente bez gubljenja informacija.

Zbog iznetih karakteristika, relacioni model podataka je najčešće korišćen.

Postupak stvaranja relacionog modela:

- prevođenje dijagrama zavisnosti entiteta u inicijalni relacioni dijagram;
- pridruživanje ili alociranje svakog identifikovanog atributa odgovarajućoj relaciji;
- pregled i provera relacionog dijagrama;
- kvantifikovanje modela podataka;
- kompletiranje definicije relacije veza i atributa.

Primer: Pojednostavljena aplikacija upita o statusu porudžbina.

Polaz: *pretpostavljena ekranska slika*

Broj porudžbine	xxx	Datum	xx xx xx
Šifra kupca	xxxx	Naziv kupca	xxxxxxxxxxxxxxxxxxxx
Broj proizvoda	xxxx	Isporučeni proizvod	xxxx

Identifikovani tipovi entiteta: KUPAC, PROIZVOD, PORUDŽBINA (o njima su potrebni podaci).

Dijagram zavisnosti entiteta:



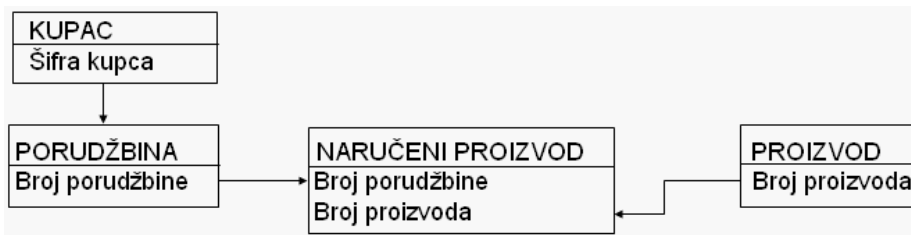
U optimalnoj formi će se javiti relacije:

- PORUDŽBINA (Broj porudžbine, datum, šifra kupca);
- PROIZVOD (Broj proizvoda, naziv proizvoda, jedinična cena proizvoda);
- NARUČENI PROIZVOD (Broj porudžbine, Broj proizvoda, naručena količina, isporučena količina);
- KUPAC (Šifra kupca, ime kupca).

U grafičkom prikazu, relacije se, kao i tipovi entiteta, predstavljaju pravougaonicima u koje se upisuje ime i primarni ključ. Veze se prikazuju na sledeći način:

- veze tipa 1:1 i 1:c se predstavljaju linijama koje spajaju pravougaonike;
- veza tipa 1:m se predstavlja linijom sa strelicom u onu stranu gde se nalazi "m" tačka veze;
- veza tipa m:m između relacija ne postoji.

Relacioni dijagram za prethodni primer:



Pravila za formiranje inicijalnog relacionog dijagrama na bazi dijagrama zavisnosti entiteta:

- svaki tip entiteta se predstavlja relacijom čije ime je isto kao i ime entiteta; identifikator entiteta postaje primarni ključ relacije;
- odnosi 1:1, 1:c i 1:m se predstavljaju vezama koje se kvantifikuju kao u dijagramu zavisnosti entiteta;
- svaki odnos m:m iz dijagrama zavisnosti entiteta se predstavlja novom relacijom čiji je primarni ključ sastavljen od identifikatora onih entiteta koji učestvuju u odnosima. Veza između nove relacije i svake od relacija iz kojih je izvedena je tipa 1:m.

Ovako dobijeni dijagram se zove inicijalni relacioni dijagram. On ne sadrži druge atribute sem primarnih ključeva.

Pridruživanje atributa odgovarajućim relacijama iz inicijalnog relacionog dijagrama (normalizacija)

Kada se atributi definišu, pridružuju se odgovarajućim relacijama. To pridruživanje (alociranje) je uspešno izvedeno ako svaki posmatrani atribut zavisi od:

- ključa,
- celog ključa,
- ničeg osim od ključa izabrane relacije.

To znači da su relacije **normalizovane** u odnosu na atribute. Ako to za neki atribut nije slučaj, treba formirati novu relaciju pa je uneti u dijagram. Normalizovana odn. normalna forma je optimalna forma atributa.

Normalizacija je proces koji pretvara inicijalni set relacija u optimalni skup koji se zove i relacioni model.

Cilj normalizacije je da se eliminiše redundancija (radi pojednostavljenja ažuriranja) i da se optimizuje mogućnost proširivanja.

Sam proces se sastoji od transformisanja inicijalne relacije u više drugih relacija, pri čemu svaki atribut u svakoj relaciji zavisi samo od ključa.

Standardni koraci u procesu normalizacije su:

Prva normalna forma (1NF): Ukloniti attribute koji ne zavise od ključa, tj. mogu imati višestruke vrednosti za datu vrednost ključa i formirati novu relaciju čiji ključ će biti sastavljen (konkatiniran) od jednog ili više njenih atributa sa ključem originalne relacije

Prva normalna forma ukida ponavljanje podataka u nenormalizovanim datotekama odn. segmentima baze, izdvajanjem podataka koji se ponavljaju u posebne datoteke/tabele koje ostaju u vezi sa osnovnom datotekom preko veznog podatka. Takođe se slogovi varijabilne dužine prevode u slogove fiksne dužine.

Druga normalna forma (2NF): Ukloniti attribute koji ne zavise od celog ključa i formirati novu relaciju čiji ključ će biti minimalni deo ključa originalne relacije i od koga će zavisiti svaki od uklonjenih atributa

Druga normalna forma zadovoljava uslove 1NF i uvodi punu funkcionalnu zavisnost svih neključnih podataka od primarnog ključa.

Treća normalna forma (3NF): Ukloniti attribute koji zavise od drugih atributa u relaciji izuzev od ključa i formirati novu relaciju čiji ključ će biti onaj atribut u originalnoj relaciji od koga zavise svi uklonjeni atributi

Treća normalna forma zadovoljava uslove 2NF i ukida bilo kakvu međuzavisnost neključnih podataka.

4. i 5. normalna forma (ređe se koriste, i to u slučajevima kada 3NF ne može da završi normalizaciju; 4NF je značajna za situacije gde ključni podatak ima višeznačne zavisnosti, koje se mogu do kraja eliminisati tek svođenjem na 5NF).

Primer:

- Sirovi (nenormalizovani) podaci:

Zona	Račun	Ime i adresa	Telefon	Saldo	Transakcije			
					broj	datum	tip	iznos
15	36895	Milan Jović, Beograd	511-231	10050,-	129	11.05	F	200,-
15	36895	Milan Jović, Beograd	511-231	10050,-	149	14.05	F	150,-
15	36895	Milan Jović, Beograd	511-231	10050,-	233	17.06	P	250,-
15	36895	Milan Jović, Beograd	511-231	10050,-	431	21.07	P	100,-
17	29433	Lazar Panić, Kraljevo	62-041	6590,-	211	11.04	F	320,-
17	29433	Lazar Panić, Kraljevo	62-041	6590,-	217	17.04	F	35,-
17	29433	Lazar Panić, Kraljevo	62-041	6590,-	114	19.04	P	256,-

1. normalna forma (1NF):

Zona	Račun	Ime i adresa	Telefon	Saldo	* Računi *
15	36895	Milan Jović, Beograd	511-231	10050,-	
17	29433	Lazar Panić, Kraljevo	62-041	6590,-	

Račun	broj	datum	tip	iznos	* Transakcije *
36895	129	11.05	F	200,-	
36895	149	14.05	F	150,-	
36895	233	17.06	P	250,-	
36895	431	21.07	P	100,-	
29433	211	11.04	F	320,-	
29433	217	17.04	F	35,-	
29433	114	19.04	P	256,-	

- 2. normalna forma (2NF):

Zona	Račun	* Veza zona - račun *	Račun	Broj trans.	* Veza račun - broj transakcije *
15	36895		29433	211	
17	29433		29433	217	
			29433	114	
			36895	129	
			36895	149	
			36895	233	
			36895	431	

* Detalji računa *

Zona	Račun	Ime i adresa	Telefon	Saldo
15	36895	Milan Jović, Beograd	511-231	10050,-
17	29433	Lazar Panić, Kraljevo	62-041	6590,-

* Detalji transakcija *

Račun	broj	datum	tip	iznos
36895	129	11.05	F	200,-
36895	149	14.05	F	150,-
36895	233	17.06	P	250,-
36895	431	21.07	P	100,-
29433	211	11.04	F	320,-
29433	217	17.04	F	35,-
29433	114	19.04	P	256,-

- 3. normalna forma (3NF):

Zona	Račun	* Veza zona - račun *	Račun	Broj trans.	* Veza račun - broj transakcije *
15	36895		29433	211	
17	29433		29433	217	
			29433	114	
			36895	129	
			36895	149	
			36895	233	
			36895	431	

* Detalji računa *

Račun	Ime i adresa
36895	Milan Jović, Beograd
29433	Lazar Panić, Kraljevo

* Detalji transakcija *

Račun	broj	datum	tip	iznos
36895	129	11.05	F	200,-
36895	149	14.05	F	150,-
36895	233	17.06	P	250,-
36895	431	21.07	P	100,-
29433	211	11.04	F	320,-
29433	217	17.04	F	35,-
29433	114	19.04	P	256,-

* Detalji imena i adrese *

Saldo	Ime i adresa	Telefon
10050,-	Lazar Panić, Kraljevo	62-041
6590,-	Milan Jović, Beograd	511-231

Pregled i provera relacionog dijagrama

Kada su svi atributi iz poslovnog sistema identifikovani i pridodati odgovarajućim relacijama, treba još jednom izvršiti proveru modela. Pri tome može da se desi da normalizovane relacije sadrže attribute sa nultim vrednostima. To se dešava onda kada atribut trenutno nema vrednost ili nema vrednost za određeni period. Tada treba formirati podtip entiteta, pa se formira nova relacija ili se dve relacije spajaju u jednu, gde će neki atribut imati nultu vrednost.

Takođe se mogu pojaviti relacije koje osim primarnog ključa ne sadrže druge atribute. Takve relacije treba ukloniti iz modela, ali pri tome treba voditi računa da li takve relacije predstavljaju odnose tipa m:m. Takve relacije ne treba uklanjati i ako sadrže samo komponente primarnog ključa, jer nisu redundantne.

Proveru modela treba izvršiti u smislu otkrivanja i uklanjanja redundantnih veza.

Kvantifikovanje modela podataka

Kvantifikovanje modela podataka se izvodi na isti način kao i kvantifikovanje modela entiteta.

Kompletiranje definicija relacija, veza i atributa

Osim samog relacionog dijagrama koji predstavlja jednu komponentu modela podataka, treba formirati i odgovarajuću dokumentaciju koja sadrži tačnu definiciju relacija, veza i atributa.

Redosled odvijanja aktivnosti pri formiranju modela podataka:

- Formirati relacije za svaki tip entiteta i za svaki odnos m:m i to dokumentovati listom entitet / relacija;
- Dodeliti atribute odgovarajućim relacijama;
- Identifikovati primarne, alternativne i strane ključeve za svaku relaciju;
- Normalizovati svaku relaciju;
- Proveriti atribute sa nultim vrednostima i, ako je potrebno, kreirati nove relacije;
- Napraviti listu atributa, odn. uneti atribute u odgovarajuće obrasce;
- Nacrtati relacioni dijagram;
- Ukloniti redundantne veze;
- Dokumentovati veze, tj. uneti ih u odgovarajuće obrasce;
- Proveriti i obezbediti međusobnu konzistentnost relacionog dijagrama, definicija veza i strukture ključa.

Dobre strane relacionog modela podataka:

- jednostavnost i laka shvatljivost (pristupačan je za krajnjeg korisnika);
- ima dobru teorijsku osnovu;
- široka primena usled razvijenih komercijalnih sistema za upravljanje bazom podataka.

Osnovni koncepti relacionog modela su struktura, ograničenja (pravila integriteta) i operatori:

Koncepti strukture	Pravila integriteta	Operatori
<ul style="list-style-type: none"> • Domen • Atribut <ul style="list-style-type: none"> - kandidat za ključ - primarni ključ - alternativni ključ - spoljni ključ • Relacija: <ul style="list-style-type: none"> - bazna - izvedena - poluizvedena - n-torka 	<ul style="list-style-type: none"> • Interna (integritet entiteta); • Eksterna (referencijalni integritet) 	<ul style="list-style-type: none"> • Unija • Presek • Diferencija • Dekartov proizvod

Model koji sadrži sve koncepte date u ovoj tabeli je **potpuni relacioni model**. Relacioni model koji podržava samo koncepte strukture i pravila integriteta a nema skup operatora naziva se **tabelarni** odn. **semirelacioni model** podataka.

Prošireni relacioni model podataka

U periodu 1970. - 1979. razvijen je čitav niz modela podataka, sa sve složenijim i semantički sve bogatijim konceptima, razvijajući se u pravcu 2. generacije modela podataka.

Dr E.F. Codd je 1979. god. neke od novih koncepata ugradio u svoj prošireni relacioni model podataka.

Novi elementi u proširenom relacionom modelu su:

- Uključivanje *nula vrednosti* i definisanje novog skupa operatora;
- Uvođenje *ključeva - surogata* da bi se napravila razlika između entiteta i njihovih sintaksnih identifikatora (imena); svi primarni i spoljni ključevi su surogati;
- Uvođenje različitih vrsta domena i relacija sa ciljem da se obuhvate novi semantički koncepti koji su se u međuvremenu pojavili u drugim modelima; npr. E - domen (E - relacija) je domen svih mogućih vrednosti surogata – unarna relacija.

Model objekti-veze (ER model)

Ovaj model je razvio Piter Čen sa MIT-a pod nazivom ERA (Entity - Relationship - Attribute: model objekti - veze - atributi, ili ER: model objekti-veze).

ER modeli su:

- jednostavni;
- pregledni;
- laki za primenu i komuniciranje;
- bogati semantikom;
- nezavisni ni od jednog konkretnog SUBP;
- laki za prevođenje u bilo koji od ranije opisanih modela (hijerarhijski, mrežni ili relacioni).

Za grafički prikaz modela se koristi ER dijagram. Dijagramom se prikazuju objekti (klase objekata), odnosno entiteti, njihovi atributi i veze među entitetima.

Koriste se sledeći standardni simboli:



Simboli se spajaju linijama, tj.



Klasa objekata A je u vezi sa klasom objekata B.

Postoji i **prošireni model objekti veze**, kod koga se prikazuju samo **binarne veze**. To znači da svaki tip veze između dva tipa objekata, E_1 i E_2 definiše dva tipa preslikavanja, tj. $E_1 \rightarrow E_2$ i inverzno, $E_2 \rightarrow E_1$.

Jedna od bitnih karakteristika veza između objekata je **kardinalnost preslikavanja**. Kardinalnost preslikavanja E_1 u E_2 definiše se parom (DG, GG) gde je DG donja granica i predstavlja minimum tj. najmanji broj pojavljivanja tipa objekta E_2 za jedno pojavljivanje E_1 , dok je GG gornja granica i predstavlja najveći broj pojavljivanja tipa objekta E_2 za jedno pojavljivanje E_1 .

Donja granica može imati vrednost 0, 1 ili neki ceo broj veći od 1.

Gornja granica (GG) može imati vrednost 1, neki ceo broj veći od 1 ili nepoznat ceo broj veći od 1, koji se označava sa M.

U jednom preslikavanju mora biti $DG < GG$.



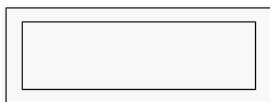
Primer: kupovina avionske karte

Pored navedenih koncepata, u ER modelu se uvodi i:

- vrsta (klasa) slabog entiteta;
- vrsta (klasa) specijalne veze.

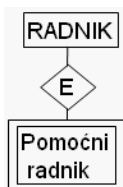
Klasa slabog entiteta je vrsta entiteta koji je na neki način zavisian od drugog entiteta i pojavljivanje tog "slabog" entiteta nema neko značenje, tj. ne predstavlja ni jedan objekat od interesa u realnom sistemu, već dobija značenje kad se poveže sa nekim drugim, tzv. nadređenim entitetom.

Za slabi entitet se koristi simbol:



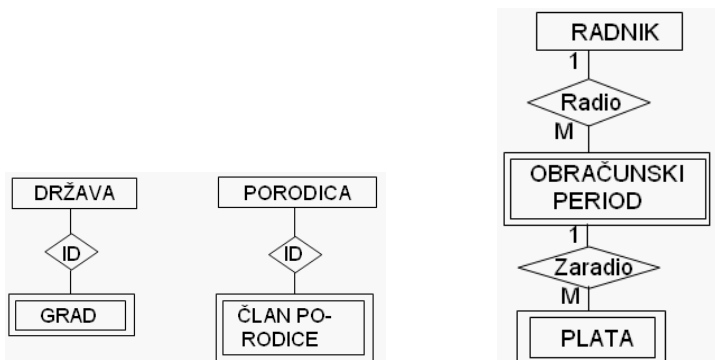
Zavisnost slabog entiteta se opisuje preko tipa specijalne veze. Postoji više vrsta specijalne veze:

- *egzistencijalna zavisnost (E)* javlja se u onim slučajevima kada postojanje entiteta zavisi od nekog drugog entiteta. Na primer:



Entitet POMOĆNI RADNIK je egzistencijalno zavisian od entiteta RADNIK

- *identifikaciona zavisnost (ID)* se javlja u slučaju kada se klasa entiteta ne može identifikovati skupom njegovih atributa, već se mora koristiti atribut identifikator nekog drugog entiteta. Na primer, entitet GRAD sa atributom IME. Da bi ovaj entitet bio pravilno identifikovan, ponekad je potrebno da bude vezan za državu, kao npr. entitet ČLAN PORODICE za entitet PORODICA.



I egzistencijalna i identifikaciona zavisnost znače da slabi entitet ne može da postoji bez nadređenog entiteta, tj. entiteta koji ga identifikuje.

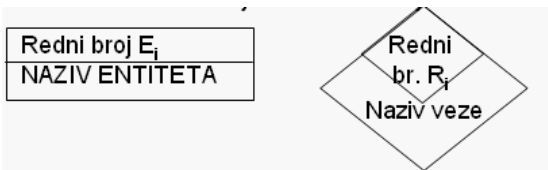
Pored egzistencijalnih i identifikacionih zavisnosti, uvodi se i pojam *kombinovane veze*, a to je slučaj kad postoje istovremeno obe navedene zavisnosti.

Uopšte, uvođenje slabih entiteta predstavlja određenu apstrakciju (apstrakcija - kontrolisano uključivanje detalja da bi se predstavila opšta svojstva objekata).

Pored pomenutih koncepata, u poslednje vreme se uvodi i pojam *mešovita klasa entitet - veza* i njome se predstavlja veza između relacija (veza). To se predstavlja simbolom:

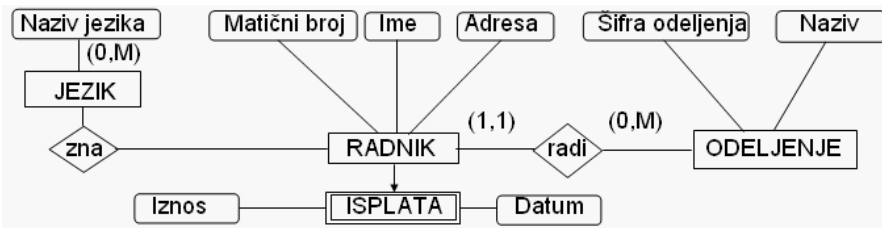


Ova apstrakcija, tj. uključivanje ovakvog detalja, predstavlja agregaciju. Skup entiteta i njihovih veza se posmatra kao jedan entitet na višem nivou apstrakcije. S obzirom da takav entitet sadrži i entitet i veze, on se naziva mešovita klasa (tip) entitet - veza, pa se grafički tako i predstavlja - kao na slici:



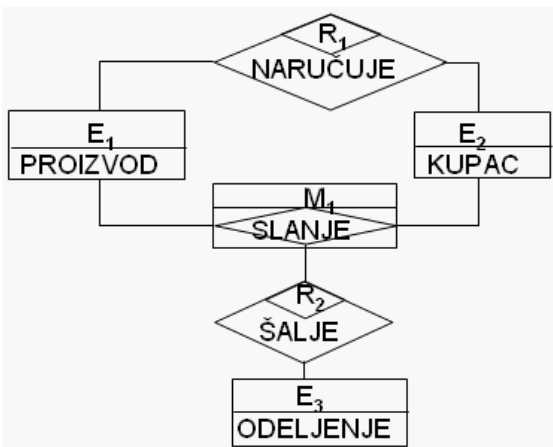
U gornjem delu oznake entiteta ili veze daje se njihov redni broj. Ti redni brojevi uzimaju se iz tabela u kojima se radi preglednosti daju nazivi entiteta i veza sa odgovarajućim atributima.

Primer ER modela sa slabim entitetom - slučaj isplate radnika:



Ovde su upisane DG i GG, tako da je kod veze "radi" DG = 1 i GG = 1, što znači da jednom pojavljivanju entiteta RADNIK odgovara najmanje jedno i najviše jedno pojavljivanje enteta ODELJENJE, tako da je veza (1,1). Jedan radnik radi u jednom odeljenju. Obrnuto, za slučaj ODELJENJE --> RADNIK, DG = 0 i GG = M (u jednom odeljenju može najmanje da nema radnika (0) ili da najviše radi mnogo radnika (M)). Takođe, ovde je dat i slabi entitet ISPLATA. Ovaj entitet sa atributima "Datum" i "Iznos" sam nema nekog značenja, ali kada se pojavljivanja entiteta ISPLATE daju za svako pojavljivanje entiteta RADNIK, tada dobija značenje.

Primer dijagrama objekti - veze sa oznakama:



Entiteti KUPAC i PROIZVOD su vezani sa dva tipa veza: "Naručuje" i "Šalje" (R_1 i R_2). SLANJE se dalje može tretirati kao mešoviti tip veze (entitet - veza) i povezati sa tipom entiteta ODELJENJE.

Metodologija modelovanja:

Prva faza - ANALIZA ZAHTEVA (identifikacija kolona, odnosno atributa - karakteristika entiteta). To podrazumeva jasno i precizno definisanje zahteva za informacijama. Zahtevi se lako definišu na osnovu dobro urađene analize funkcija realnog sistema.

Druga faza - NALAŽENJE GLOBALNOG MODELA PODATAKA (grupisanje kolona u entitete - definisanje tipova entiteta, veza i atributa). Entiteti, veze i njihovi atributi se prikazuju grafički. Dobijeni globalni model služi za komunikaciju sa korisnikom. Nakon toga se dodaju i drugi semantički detalji i crta se konačni dijagram.

Treća faza - IDENTIFIKACIJA PRIMARNIH KLJUČEVA (za svaki entitet treba ispitati da li neka od kolona ima jedinstvenu vrednost za svaki nastupajući element tog entiteta - ako postoji, proglašiti je za primarni ključ).

Četvrta faza - IDENTIFIKACIJA STRANIH KLJUČEVA (strani ključevi su kolone čije vrednosti povezuju entitet sa primarnim ključem nekog drugog entiteta. Identifikacija stranih ključeva se sastoji u poređenju primarnih ključeva i kolona kod drugih entiteta).

Analiza veza među entitetima sa aspekta kardinalnosti

Tipovi veza mogu biti 1:1, 1:c, 1: m i m:m.

Ako se uoči veza tipa m:m, treba je eliminisati uvođenjem novog entiteta koji je u vezi tipa 1:m sa oba entiteta koji su bili u vezi tipa m:m. Taj postupak znači normalizaciju modela podataka, po pravilima:

- 1. pravilo: samo atomske vrednosti (nijedna kolona da ne sadrži složene vrednosti niti da predstavlja grupu koja se ponavlja; npr. ako u tabeli treba imati podatke o svim vrstama isplata radniku ili o brojevima telefona komitenta, bolje je napraviti novu tabelu od ključa i po jednog broja telefona / vrste isplate).
- 2. pravilo: zavisnost od celog primarnog ključa (npr. u tabeli koja sadrži ključ = broj indeksa + šifra predmeta i atribut *ime studenta*, ime studenta zavisi samo od broja dosijea a ne i od šifre predmeta, pa ga treba eliminisati).
- 3. pravilo: zavisnost samo od primarnog ključa (kolone u tabeli treba da zavise samo od primarnog ključa; npr. tabela koja sadrži: šifru predmeta, naziv predmeta, šifru studijske grupe i naziv studijske grupe treba da se podeli u 2 tabele).

Prevođenje ER modela u druge

Da bi ER modeli mogli da se realizuju odn. da se projektuje baza podataka, moraju da se prevedu na neki od komercijalnih modela tj. na one modele za koje postoje komercijalni SUBP (hijerarhijski, mrežni ili relacioni).

Ako se želi da se ER model realizuje pomoću relacionog SUBP, mora se transformisati u skup tabela (relacija) koje su u trećoj odn. četvrtoj normalnoj formi. Za takva prevođenja postoje utvrđeni postupci, kao što su:

- Čenov postupak;
- postupci analize relacija;
- postupci sinteze relacija i dr.

Čenov postupak sastoji se u sledećem:

- definicija funkcionalne zavisnosti između atributa;
- normalizacija tipova entiteta;
- normalizacija tipova veza.

Zbog mogućnosti prevođenja u druge modele, kaže se da ER model predstavlja generalizaciju tih modela.

Prevođenje ER modela u relacioni model

Prevođenje ER modela u relacioni model izvodi se na sledeći način:

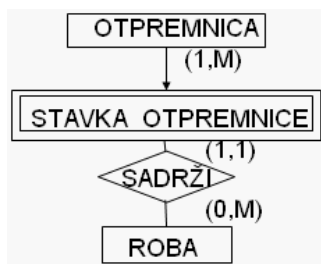
- Deo strukture ER modela, odnosno DOV (Dijagram Objekti Veze) se predstavlja relacionom šemom.

- Ograničenja, operacije i deo strukture ER modela se predstavljaju operacijama za očuvanje integriteta definisanja nad relacionim modelom, a implementiraju korišćenjem sredstava odabranog relacionog SUBP.

Postupak prevođenja

- Pravila za objekte (entitete):
 1. Svaki objekat iz DOV postaje šema relacije. Ime tipa entiteta postaje ime šeme relacije. Obeležja objekta su obeležja šeme relacije. Za osnovne objekte identifikator objekta (entiteta) postaje primarni ključ šeme relacije.
 2. Svaki "slab" objekat takođe postaje šema relacije. Ime tipa objekta postaje ime šeme relacije. Obeležja objekta su obeležja šeme relacije. Identifikator nadređenog objekta postaje jedno od obeležja šeme relacije koja odgovara "slabom" objektu. Identifikator slabog objekta čine identifikator nadređenog objekta i obeležja "slabog" objekta koja jedinstveno identifikuju pojavljivanje "slabog" objekta u okviru pojavljivanja njemu nadređenog objekta.
 3. Objekat nadtip (generalizovani tip objekta) postaje šema relacije. Ime nadtipa postaje ime šeme relacije. Obeležja nadtipa su obeležja šeme relacije. Identifikator nadtipa postaje ključ šeme relacije.
 4. Objekat podtip takođe postaje šema relacije. Ime podtipa postaje ime šeme relacije. Obeležja podtipa su obeležja šeme relacije. Identifikator nadtipa predstavlja ključ šeme relacije.
- Pravila za prevođenje veza među objektima
 1. Veze tipa (1:1)(0:M) ne postaju posebne šeme relacije, već identifikator objekta za koji je GG=M postaje obeležje objekta za koji je GG=1.

Primer:



2. Veze tipa (0:1)(1:1) sa objektima koji su u vezi prevode se u dve šeme relacije (za svaki objekat po jedna šema relacije i identifikator jednog objekta postaje obeležje drugog objekta). Bolje je predstavljanje spoljnjim ključem u šemi relacije objekta sa strane (1:1).

KLASIČNI METODI PROJEKTOVANJA INFORMACIONIH SISTEMA

Najpoznatiji klasični metodi su:

- **BSP** (Business System Planning) metod;
- **SDM** (System Development Method) firme "IBM";
- **SADT** (System Analysis Design Technics);
- **SSA, HIPO** - koriste se za pojedine faze prikaza informacionog sistema.
- **BISAD** (Business Information System Analysis and Design) firme "Honeywell";
- **ISAC** (Information System Analysis of Change);
- **PORGI** (Planning Organization and Implementation);
- **PN** (Petri Nets) - Petrijeve mreže;
- **EPN** (Extended Petri Nets) - proširene Petrijeve mreže.

Ciljevi metoda su:

- Povećanje produktivnosti, što znači **brže** obavljanje procesa rada;
- Zadovoljenje stvarnih potreba korisnika odn. **bolji rad**;

- Postizanje fleksibilnosti rešenja u odnosu na promene, odn. **trajnost**

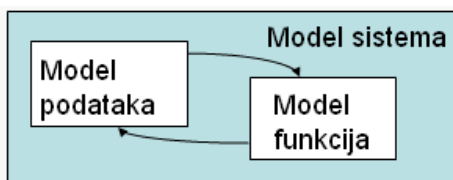
Zajedničke karakteristike metoda:

- Korišćenje pristupa jedinstvenim podacima (to znači da su podaci jedinstveni resurs u okviru nekog sistema, tj. da različiti podsistemi koriste iste podatke);
- Izgradnja IS-a počinje sa vrha, tj. koristi se Top Down pristup u planiranju a Bottom Up u realizaciji;
- Postoji nezavisnost poslovnih zahteva od tehničkih specifikacija, što znači da je modelovanje podataka nezavisno od procesa njihove obrade;
- Usklađena je analiza podataka i procesa;
- Svi metodi koriste fazni pristup;
- Postoji metodološka usklađenost arhitekture i aplikacija;
- Svi metodi koriste standardnu dokumentaciju na računaru;
- Postoji uključenje korisnika u rad.

SDM (SYSTEM DEVELOPMENT METHOD)

Ovaj metod je razvio IBM. Sastoji se od dve osnovne faze:

- modelovanje podataka;
- modelovanje funkcija (procesa).

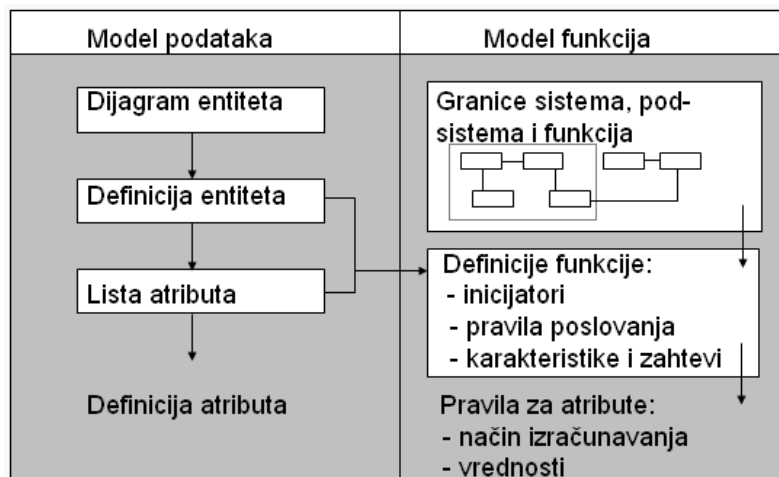


Sistem se posmatra kao jedinstveni model koji se sastoji od modela funkcija i modela podataka.

Model podataka i model procesa se projektuju nezavisno, ili bolje rečeno, model podataka prethodi izradi modela procesa. Nezavisno projektovanje je moguće, jer:

- kada se projektuje model podataka ne uzimaju se u obzir procesi koji će obrađivati bazu podataka;
- mnoge funkcije nisu kritične sa aspekta projektovanja;
- podaci su zajednički za ceo poslovni sistem dok su funkcije više lokalne, tako da model podataka doprinosi integraciji IS-a.

Veze između modela podataka i modela funkcija:



SSA (STRUCTURED SYSTEM ANALYSIS) - metod strukturne analize sistema

Koristi se za analizu postojećeg informacionog sistema i za izgradnju strukturnog modela procesa sistema.

Rezultat primene SSA je funkcionalna specifikacija, tj. za izradu funkcionalne specifikacije potrebno je definisati funkciju odnosno procese.

Funkcija se može definisati kao proces ili skup procesa koji koristi ulaz da bi ga na efikasan način transformisao u izlaz. Svaka funkcija (proces) ima **podfunkcije** (podproces) a ova svoje podfunkcije (podproces), sve do **elementarnih procesa** koji nemaju svoje podproces. Efekat dejstava podprocesa je postizanje ostvarenje funkcije u celini.

U SSA metodu, svaki proces obrade podataka započinje ili se inicira nekim ulaznim tokom podataka, zatim se obrada specificira logički jasno i jednoznačno, tako da se dobijaju jednoznačno definisani rezultati. To znači da SSA metod posmatra informacioni sistem kao funkciju (proces obrade) koja na bazi ulaznih generiše izlazne podatke.

Danas većina CASE alata koji imaju ugrađenu prvu fazu projektovanja IS (EXCELERATOR, IEW, SPAS, ARTIST itd) koristi SSA metod.

Osnovne karakteristike:

- koristi jasnu grafičku prezentaciju, pogodnu za komunikaciju sa korisnikom;
- daje jasan i detaljan opis sistema primenom metoda apstrakcije, tako da se sistem na višim nivoima opisuje uopšteno a na nižim detaljno;
- daje logičku a ne fizičku specifikaciju procesa, što znači da se koristi za prikaz onoga što će sistem pružiti korisniku a ne za ilustraciju kako će sistem biti implementiran.

Osnovni koncepti

Pri obradi podataka koriste se podaci iz baze podataka i ulazni podaci. Model poslovnih procesa se prikazuje dijagramom toka podataka. Dijagram toka podataka obuhvata sledeće koncepte:

- **Funkcije / procesi**

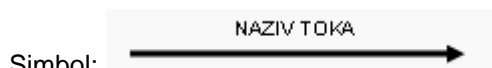
Proces predstavlja transformaciju ulaznih u izlazne tokove podataka. Sinonimi za proces su: rad, aktivnost i posao. Proces na dijagramu toka podataka predstavlja aktivnost obrade podataka, pa može biti u vezi samo sa tokom podataka, od koga dobija podatke ili za koji stvara podatke.



Simbol:

- **Tokovi podataka**

Tok podataka je jedan ili više skupova podataka (agregacija podataka) koji u sistemu ima korisnički jasnu semantiku i koji povezuje ostale komponente sistema. To je vod kroz koji teče informacija poznate strukture i sadržaja. Po svojoj prirodi može biti: dokument, skup dokumenata, elementarni podatak (npr. nabavna cena proizvoda), neformatizovan skup podataka (npr. knjiga, časopis i sl.), bilo koje činjenice (npr. znanje eksperata, razgovor, telefonski poziv,...).

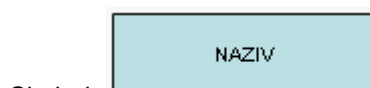


Simbol:

- **Izvori i odredišta podataka (interfejsi)**

Interfejs je eksterni sistem odnosno spoljni objekat koji je povezan sa posmatranim sistemom kao izvor ili odredište podataka (može se javiti i u obe uloge). Procesi u okviru interfejsa nas ne zanimaju. Interfejs može biti osoba (npr. radnik, kupac i sl) ili organizacioni sistem (institucija, poslovna funkcija i sl).

Interfejs modeluje ona područja koja su van domena našeg interesovanja. Sam sistem i veze sa interfejsima predstavljaju (sačinjavaju) **dijagram konteksta sistema**.



Simbol:

- **Skladišta podataka**

Skladište podataka je memorijska lokacija na kojoj su podaci privremeno memorisani. Može biti dokument (npr. faktura), datoteka ili baza podataka, znanje u ljudskoj glavi, kartoteka, biblioteka, skriveni skup podataka (lična beležnica nekog izvršioca) itd.

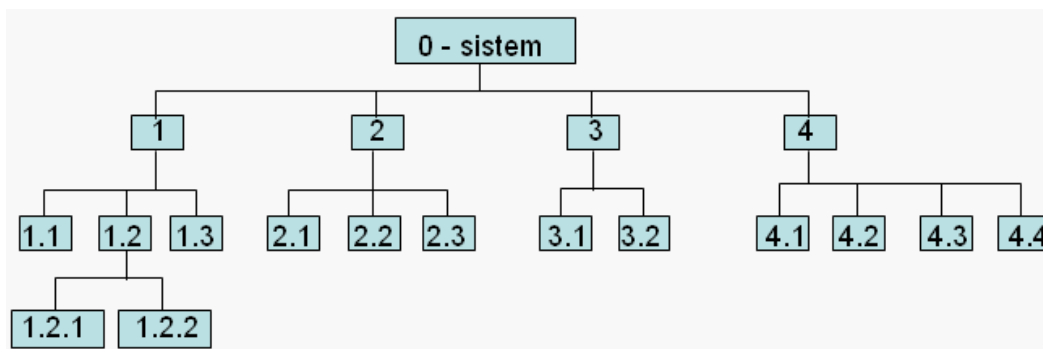


Pravila za izradu dijagrama tokova podataka

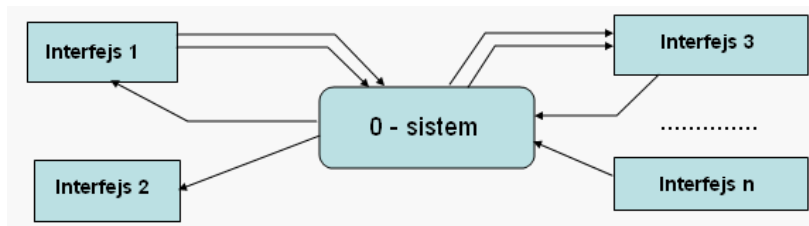
- Tok podataka mora da ima izvor i odredište. Izvor ili odredište može biti bilo koja druga komponenta (interfejs, proces, skladište podataka), ali za jedan tok, bilo izvor bilo odredište (ili oba) mora biti proces;
- Svaki tok podataka mora imati ime, izuzev tokova koji idu od odn. ka skladištima podataka;
- Tok podataka se može granati ili se daju dva toka sa istim imenom koja imaju isti izvor a različita odredišta;
- Svaki proces mora imati ime, a poželjna je i referentna oznaka koja iskazuje hijerarhijsku pripadnost procesa pri dekompoziciji;
- Svaki proces treba da ima bar jedan ulazni i bar jedan izlazni tok podataka (proces bez ulaza stvara izlaz nižčega, a proces bez izlaza je nesvrshodan);
- Skladište podataka može biti bez ulaznog toka (podrazumeva se da se stvara i ažurira u nekom drugom sistemu) ili bez izlaznog toka (posmatrani sistem ga samo formira a koristi se u nekom drugom sistemu);
- Svaki interfejs mora imati bar jedan ulazni ili izlazni tok podataka, inače bi bio izolovan od sistema;
- Da bi se minimiziralo presecanje tokova podataka, dozvoljava se višestruko ponavljanje skladišta i interfejsa.

Pravila za dekompoziciju dijagrama toka podataka

- Na najvišem nivou je **dijagram konteksta** (jedan proces koji predstavlja ceo sistem, interfejsa i odgovarajuće tokove podataka između sistema i interfejsa);
- Na prvom nivou dekompozicije je **dijagram prvog nivoa** (tzv. Root ili korenski dijagram – dekompozicija dijagrama konteksta na osnovne procese);
- Ceo skup hijerarhijski dekomponovanih procesa prikazuje se **dijagramom dekompozicije**;
- Proces koji se dalje ne dekomponuju su **elementarni** odn. **primitivni procesi**. Za njih se daje opis - specifikacija logike njihovog odvijanja, odnosno **mini-specifikacija** sistema;
- Pored procesa, mogu se dekomponovati i tokovi i skladišta podataka. Njihova dekompozicija se prikazuje u **rečniku podataka SSA**, pomoću sintakse za opis strukture podataka;
- **Pravilo balansa tokova**: ulazni i izlazni tokovi na celokupnom dijagramu toka podataka dobijenom dekompozicijom nekog procesa moraju biti ekvivalentni sa ulaznim i izlaznim tokovima tog procesa prikazanim na dijagramu višeg nivoa;
- Jedan dijagram toka ne bi trebalo da sadrži previše (npr. maksimalno desetak) procesa - ako procesa ima mnogo, verovatno je preskočen neki nivo dekompozicije.



Hijerarhijski dijagram dekompozicije sistema



Dijagram konteksta

Kriterijumi dekompozicije

Svrishodnost za analitičara sistema znači semantički jasne razloge za dekompoziciju. Izvodi se od strane analitičara i korisnika, intervjuisanjem. Nema formalizovanih tehnika, već se koristi zdrav razum i ekspertske znanje.

Koherentnost je kriterijum koji izražava kvalitet dekompozicije na osnovu povezanosti delova sistema. Može biti:

- **Interna koherentnost** - označava stepen međusobne povezanosti elemenata unutar procesa.
- **Eksterna koherentnost** - izražava povezanost procesa ("kuplovanje") sa ostalim procesima u okviru sistema. Posmatraju se tokovi podataka od i ka ostalim procesima.
- **Strukturna koherentnost** - proces je strukturno koherentan ako postoji preslikavanje strukture ulaznih tokova u izlazne tokove podataka, tj. ako su svi ulazi nužni za proizvodnju izlaza i ako su svi izlazi iz procesa dovoljni za postizanje cilja procesa. Strukturna koherentnost se iskazuje kao **kongruencija** (usklađenost, pripadnost istoj klasi strukture).

Interna koherentnost može biti:

- **koincidentalna** (podproces i elementi su okupljeni u proces slučajno i ni po kom osnovu nisu povezani, proces ima veliki broj ulaza i izlaza - najniži mogući stepen kohezije; takvo okupljanje u proces je nepoželjno);
- **temporalna** (podproces i elementi su okupljeni da se zajedno izvršavaju zato što im je vreme izvođenja isto; stepen kohezije je vrlo nizak);
- **taksomatska** (podproces i elementi se izvršavaju zajedno zato što pripadaju istoj klasi i među njima postoji neka logička veza, npr. procesi jednog radnog mesta, procesi jedne organizacione jedinice; stepen kohezije je nizak);
- **komunikaciona** (podproces i elementi u sastavu procesa rade sa istom strukturom podataka odn. sa istim skladištem podataka; stepen kohezije je dobar);
- **sekvencijalna** (podproces i elementi u okviru procesa se izvršavaju u nizu jedan za drugim, tako da su izlazni tokovi podataka iz jednog procesa ulazi u drugi proces; stepen kohezije je dobar);
- **funkcionalna** (podproces i elementi su nužni i dovoljni da se ostvari funkcija ukupnog procesa, problematski su orijentisani u postizanju jedne izdiferencirane funkcije kao npr. formiranje porudžbine dobavljaču, imaju jasan cilj a mali broj ulaza i izlaza; stepen kohezije je maksimalan).

Eksterna koherentnost može biti:

- **Sa aspekta sadržaja veze:**
 - **povezanost podacima** (ako je izlaz iz jednog procesa ulaz u drugi proces; ako su svi podaci u ulaznom toku potrebni procesu za njegovo funkcionisanje, u pitanju je parametarska veza - najbolji tip veze, inače se radi o vezi preko zapisa; ako su procesi povezani skladištem podataka tako da mogu da menjaju podatke u njemu, radi se o vezi preko zajedničkog područja podataka - najlošija veza);
 - **povezanost upravljanjem** (proces je povezan upravljanjem ako ulazni tok podataka nosi upravljačke informacije koje određuju kako proces treba da funkcioniše; povezanost se može izraziti putem parametara, zapisa ili zajedničkog područja podataka, kao kod povezanosti podacima);
 - **hibridna povezanost** (istovremena povezanost podacima i upravljanjem sadržana u jednom toku podataka; nepoželjna je);

- **Sa aspekta mehanizma povezivanja:**

- **normalna povezanost** (nema prekida u radu zbog komunikacije sa drugim procesima, već je u procesu skriven tok podataka koji sam na početku rada uzima podatke iz skladišta i na kraju rada proizvodi izlazne tokove podataka);
- **patološka povezanost** (proces je patološki povezan sa drugim procesom ako prekida sa radom da bi prihvatio ulaz ili proizveo izlaz za drugi proces).

Rečnik podataka strukturne sistemske analize

Daje opis strukture i sadržaja svih tokova i skladišta podataka. Bez obzira što tok ili skladište podataka mogu predstavljati papirni dokument, niz znakova unet tastaturom, "paket" informacija dobijen TP-vezom, kartoteku ili datoteku, kao logička struktura oni predstavljaju neku kombinaciju polja podataka. Za preciznu definiciju logičke strukture skladišta i tokova i definisanje sintakse rečnika, neophodno je da se najpre definišu svi *koncepti rečnika podataka*.

Polje je elementarna (atomska) struktura koja se dalje ne dekomponuje i koja ima svoju vrednost. Npr., u fakturi polja mogu biti: broj fakture, jedinična cena, količina i sl.

Domen je skup mogućih vrednosti iz koga polje dobija konkretnu vrednost. Domeni mogu biti:

- predefinisani, odn. standardni programsko-jezički domeni (npr. INTEGER, CHARACTER, REAL,...)
- semantički (definišu se posebno, preko svog imena, predefinisano domena i event. ograničenja na mogući skup vrednosti predefinisano domena, npr. **MESEC defined as INTEGER(2) between 1.12**)

Dva polja su semantički slična samo ako su definisana nad istim domenom. Različita polja mogu se povezati nekim operatorom samo ako su definisana nad istim domenom i ako je operator definisan u tom domenu.

Ograničenja u pogledu vrednosti polja odnosno domena mogu biti:

- konstanta sa operatorom poređenja <, >, <=, >=, = (npr. **RADNI_STAZ: INT(2) <= 40**);
- BETWEEN konstanta, konstanta, gde su konstante vrednosti iz datog domena (npr. **MESEC:INTEGER (2) BETWEEN 1,12**);
- IN (lista vrednosti), gde se lista formira od vrednosti iz odgovarajućeg domena (npr. **BOJA: CHAR(6) IN("CRVENA", "ZELENA", "PLAVA")**);
- NOT NULL (za polje nije dozvoljena vrednost nula, npr. **LICNA_KARTA: CHAR(6) NOT NULL**);
- složena, odnosno kombinacije gornjih ograničenja (npr. **RADNI_STAZ: INT(2) <= 40 AND NOT NULL**).

Struktura tokova podataka i skladišta predstavlja neku kompoziciju polja, odnosno konstrukciju čije su komponente polja. Kao komponenta jedne strukture se, pored polja, može javiti i druga definisana struktura.

Konstrukcija kojom se od komponenata gradi struktura može biti:

- agregacija komponenti,
- ekskluzivna specijalizacija (unija) komponenti,
- neekskluzivna specijalizacija (unija) komponenti,
- skup komponenti.

Agregacija komponenti predstavlja složenu strukturu *n* komponenti; vrednost agregacije je *n* - torka u kojoj svaki element ima vrednost odgovarajuće komponente. Npr.

```
PROMET < SIFRA_PROIZVODA,  
          SIFRA_MAGACINA,  
          VRSTA_PROMETA,  
          KOLICINA,  
          CENA  
>
```

Ekskluzivna specijalizacija (unija) komponenti se predstavlja kao lista komponenti u uglastim zagradama (u strukturi se javlja jedna od tih komponenti), npr.

```
PROMET < SIFRA_PROIZVODA,  
          SIFRA_MAGACINA,  
          VRSTA_PROMETA,  
          KOLICINA,  
          [PLANSKA_CENA, NABAVNA_CENA]
```

>

Neekskluzivna specijalizacija (unija) komponenti - lista komponenti u kosim zagradama (jedna, dve ili sve komponente iz liste mogu se pojaviti u strukturi). Npr.

```
PROMET < SIFRA_PROIZVODA,  
          SIFRA_MAGACINA,  
          VRSTA_PROMETA,  
          /KOLICINA_ULAZA, KOLICINA_IZLAZA, STANJE/  
          CENA
```

>

Skup komponenti (tj. skup više vrednosti jedne komponente) - u strukturi komponenta može da se javi više puta. Predstavlja se vitičastim zagradama, npr.

```
PROMET < SIFRA_PROIZVODA,  
          SIFRA_MAGACINA,  
          VRSTA_PROMETA,  
          <{DATUM_PROMENE, KOLICINA}>  
          CENA
```

>

Mini specifikacije - specifikacija logike primitivnih procesa

Primitivni procesi su procesi na najnižem nivou dekompozicije. Oni su po pravilu sekvencijalni, redosled aktivnosti u okviru njih je definisan, pa se za njihovu specifikaciju moraju koristiti neki alati za specifikaciju sekvencijalnih procesa, ili, kako se obično zovu, alati za opis logike procesa.

Postoji čitav skup ovih alata, počev od dijagrama toka programa ("Flowchart"), preko Nassi Shneiderman-dijagrama, tabela odlučivanja, stabla odlučivanja, raznih vrsta strukturalnih jezika i pseudokodova. Najčešće se koristi neka vrsta strukturalnog prirodnog jezika ("Structured English" ili "Strukturalni srpski") ili **pseudokoda**.

Ponekad se pravi razlika između strukturalnog prirodnog jezika i pseudokoda, mada se oni u osnovi baziraju na istim principima. Naime, strukturalni prirodni jezik koristi rečnik nekog prirodnog jezika (srpskog, na primer), a za konstrukciju rečenica i složenijih sklopova koristi strukture strukturalnog programiranja, sekvenciju, selekciju i iteraciju, predstavljajući ove strukture uobičajenim "ključnim rečima" (BEGIN, END, DO WHILE, IF ... THEN.... ELSE i drugim sličnim). Pseudokod je bliži programskim jezicima jer više koristi rečnik (ključne reči) nekog izabranog programskog jezika.

Pseudokod

Zanemarujući eventualne manje razlike između pojmova "strukturalni prirodni jezik" i "pseudokod", možemo reći da je pseudokod strukturisani prirodni jezik, jezik koji koristi rečnik prirodnog jezika, a čije su konstrukcije strukturisane pomoću koncepata strukturalnog programiranja. Prirodni jezik nije pogodno sredstvo za specifikaciju logike procesa zbog svoje nepreciznosti i višeznačnosti i zato ga je neophodno strukturisati.

Osnovne strukture za strukturisanje prirodnog jezika su:

- **Sekvencija** - Aktivnosti u sekvencijalnom bloku se odvijaju po redosledu navođenja. Ponekad je pogodno da se i sekvencijalni blok akcija ograniči sa ključnim rečima BEGIN i END.

Na primer, grubi pseudokod za proces *Evidentiranje kandidata* bi mogao da bude:

```
BEGIN  
Unesi podatke o diplomu;  
Unesi podatke sa svedočanstava;  
Unesi podatke o nagradama;  
Ažuriraj datoteku KANDIDATI_ZA_UPIS;  
END;
```

- **Selekcija** - Redosled aktivnosti zavisi od nekog uslova. Ako je uslov ispunjen, obavlja se jedan, a ako nije drugi blok akcija. Opšti iskaz selekcije je:
IF uslov THEN blok_akcija_1 ELSE blok_akcija_2;
Na primer, IF BROJ POENA > 85 Upiši kandidata ELSE Odbij kandidata;
- **Case struktura** - Ovo je specijalni slučaj selekcije, kada se u zavisnosti od vrednosti jednog parametra može izvršavati više različitih blokova akcija. Opšti iskaz za ovu strukturu je:
CASE parametar OF
vrednost_parametra_1: blok_akcija_1
vrednost_parametra_2: blok_akcija_2
...
vrednost_parametra_n: blok_akcija_n
Na primer,
CASE SEMESTAR OF
1: Stavi_studenta u grupu za prvu godinu
3: Stavi_studenta u grupu za drugu godinu
5: Stavi_studenta u grupu za treću godinu
7: Stavi_studenta u grupu za četvrtu godinu;
- **Iteracija** - Blok akcija se ponavlja dok se neki uslov ne ispuni ili dok se akcije ne obave za sve objekte nekog skupa. Osnovni oblici ove strukture su:
DO WHILE uslov blok_akcija - uslov se ispituje pre svakog izvršenja bloka akcija, moguće je da se blok akcija ne izvrši nijednom,
DO UNTIL uslov blok_akcija - uslov se ispituje na kraju svakog izvršenja bloka akcija, blok akcija se izvršava bar jednom.
FOR EACH naziv_skupa_objekata DO blok_akcija
Ponekad se ključna reč DO zamenjuje sa REPEAT, PERFORM ili LOOP. Uopšte, ključne reči za pseudokod mogu se izabrati kao standard u pojedinim organizacijama, tako da budu bliske rečima programskog jezika koji se koristi.

IMPLEMENTACIJA INFORMACIONOG SISTEMA

Testiranje informacionog sistema

Prvi korak implementacije novog sistema je **testiranje IS-a**. Testiranje treba obaviti u cilju uočavanja i otklanjanja svih grešaka. Sam proces testiranja se može razgraničiti na:

- testiranje programa;
- testiranje cele aplikacije AOP-a.

Testiranje programa obavljaju programeri koji su radili programe, a obavlja se u cilju provere programskih zadataka koje je programeru dao analitičar sistema. Testiranje programa je u tesnoj vezi sa ukupnom softverskom podrškom programu, pre svega sa SUBP, zatim programima za prevođenje i drugim softverom koji omogućava izradu aplikacija.

Testiranje se obavlja u smislu provere upotrebljivosti i ispravnosti izlaznih podataka, provere sadržaja slogova, provere izlaza na ekranu i štampaču, provere veza sa drugim programima i sl.

U okviru *testiranja aplikacije* treba testirati povezanost programa u okviru aplikacija, zahteve korisnika za izmenu podataka i za izlazom, pripremu i distribuciju izlaza, održavanje i kontrolu logičke i fizičke baze podataka, merenje trajanja obrada za stvarni obim podataka i sl.

Kreiranje i administracija baze podataka

Drugi korak implementacije je **kreiranje i administracija baze podataka**. Ovim poslovima bavi se administrator baze podataka. Projektanti informacionog sistema su zaduženi za logički i implementacijski dizajn baze podataka kao i za inicijalno testiranje baze podataka. Testiranje sa stvarnim podacima rade administratori baze podataka.

Administratori BP su stručna lica koja odgovaraju za podatke u jednom preduzeću. Oni odlučuju o tome kako će podaci biti organizovani i kako će im se pristupati. Nakon dizajniranja oni prate ponašanje BP i otklanjaju eventualne greške. U skladu sa uočenim nedostacima administratori BP mogu da sprovedu određene izmene u logičkoj strukturi. Oni vrše oporavak BP nakon oštećenja, održavaju SUBP i vode evidenciju o korišćenju BP. Administratori BP daju ovlašćenja za korišćenje BP ili njenih delova i regulišu lozinke za pristup podacima. Administratori BP mogu eksplicitno da dodeljuju i da oduzimaju ovlašćenja pojedinim korisnicima u odnosu na pojedine resurse.

Administrator BP može da menja fizičku strukturu bez uticaja na logičku strukturu. Kako su korisnici različiti, od izuzetnog je značaja i pitanje osiguranja BP, jer podaci moraju biti pouzdani i bezbedni. Zbog toga moraju da postoje procedure za održavanje i obezbeđenje BP. S obzirom da se BP može oštetiti ili uništiti, mora da se obezbedi periodično kopiranje primarne verzije BP i pravljenje kopija.

Uvođenje informacionog sistema u eksploataciju

Metodi uvođenja

Prelazak na novi sistem se vrši tek pošto je kompletan aplikativni softver do kraja istestiran, baza podataka kreirana a hardver instalisan. Način prelaska na novi sistem zavisi od prirode aplikacija i organizacije rada preduzeća. Od konkretnih uslova zavisi izbor nekog od sledeća četiri metoda uvođenja informacionog sistema:

Direktni start

U metodu direktnog starta, poznatom i kao "metod spaljenih mostova" u datom trenutku se prekida rad na stari način i odmah se startuje sa primenom novog informacionog sistema. Za trenutak prelaska obično se bira period nerada preduzeća (noć, vikend, praznik). Metod direktnog starta je normalan metod uvođenja on line aplikacija, jer se ne isplati držati paralelno stari "ručni" sistem ili batch obradu sa novim on line informacionim sistemom. Njime se minimizira dupliranje rada ali zahteva pažljivo planiranje, testiranje i poklanjanje pune pažnje ispravnosti rada da bi bio kompletno uspešan.

Rizik u metodu direktnog starta leži u potencijalno nedovoljno istestiranom ili nekompletnom novom sistemu, u nedovoljnoj obučenosti korisnika, kao i u nemanju rezultata (izlaza) koje je davao stari sistem da bi se odmah sa njima uporedili rezultati novog sistema i otklonile greške koje izazivaju odstupanja.

Paralelni rad

Po ovom metodu, stari i novi sistem paralelno funkcionišu u određenom vremenskom periodu u kome se upoređuju njihovi izlazni rezultati sa ciljem da budu identični. Stari sistem funkcionise dok se rezultati novog ne stabilizuju (otklanjanje svih grešaka i propusta uhodavanje). Ako se vrši prelaz sa jednog tipa računara na drugi (što je danas čest slučaj), paralelni rad može izazvati usporenje rada, jer se ulazni podaci moraju čuvati na kompatibilnim medijima (npr. magnetskim trakama) i odatle unositi praktično konkurentno na oba računara. Osnovni problem je dupliranje napora korisnika koji u isto vreme treba da održavaju dva informaciona sistema.

Pomereni paralelni rad predstavlja varijantu paralelnog rada u kojoj obrada u novom sistemu kasni za jedan vremenski period iza starog sistema, tako da se u izvesnom broju perioda rezultati novog sistema upoređuju sa već dobijenim rezultatima starog sistema. Ovaj metod, poznat i kao metod pilot-obrade, dozvoljava temeljniju kontrolu novog sistema uz nesmetano funkcionisanje starog. Kad novi sistem bude sasvim spreman, intenzivira se njegova obrada do "hvatanja koraka" sa starim sistemom i onda se stari sistem gasi.

Nedostatak pomećenog paralelnog rada je višestruki intenzitet rada u novom sistemu u fazi postizanja ažurnosti.

Fazno uvođenje

Slično je metodu paralelnog rada, s tim što se u početku samo deo podataka uključuje u paralelni rad, na primer, startuje se novi informacioni sistem praćenja proizvodnje samo za određene proizvode a ostali se uključuju u kasnijim fazama. Kad se postigne stabilan paralelni rad sa svim podacima, stari sistem se gasi.

Slabost ovog metoda je u kašnjenju implementacije, odnosno njenom protezanju na veći broj perioda.

Fazno uvođenje je pogodno u situacijama gde postoji više lokacija na kojima se odvija isti posao - novi sistem se uvodi po grupama lokacija.

Prototipsko uvođenje

Novi sistem se ne uvodi u celosti, već na probnom uzorku, najčešće kroz paralelni rad. Uzorci se mogu uvoditi i fazno.

Izbor informatičke opreme

Faktori koji utiču na izbor opreme

Izboru i raspoređivanju opreme treba posvetiti punu pažnju, jer će od toga zavisiti i uspešnost realizacije novog sistema. Izbor opreme zavisi i od finansijskih mogućnosti. Treba kupiti onu opremu koja će zadovoljiti potrebe preduzeća. Zato je razvijen čitav niz metoda koji omogućavaju pravilan izbor opreme.

Postoje četiri osnovne grupe faktora koje služe kao osnov za izbor opreme:

- **zahtevi koje novi informacioni sistem postavlja u pogledu obrade** (najvažniji faktor - zahtevi se određuju u fazi globalnog projektovanja odn. u fazi izrade idejnog projekta, kojim se definiše rešenje novog IS i sagledava se oblast primene opreme);
- **mogućnosti tehničkih sredstava** (pri izboru opreme važnu ulogu igraju njene tehničke karakteristike - brzina obrade i kapacitet. U tehničke karakteristike najpre se ubrajaju karakteristike procesora, karakteristike unutrašnje i spoljašnje memorije – kapacitet, veličina memorijske lokacije, vreme pristupa i vreme ciklusa. I softver u velikoj meri određuje

mogućnosti primene računara, zatim troškove obrade podataka, kao i stepen u kome neki računarski sistem odgovara potrebama korisnika);

- **podrška proizvođača - dobavljača** (uobičajena je praksa, bar kad su u pitanju veliki računarski sistemi, da se proizvođači ocenjuju pre svega po mestu na listi svetskih proizvođača opreme i po stepenu kompletnosti usluga koje pružaju, posebno u procesima montaže računarskog sistema i ostalih instalacija, održavanja sistema, obuke kadrova i pomoći pri uvođenju sistema);
- **cena opreme** (iz razloga ekonomičnosti, nastoji se da se kupi najjeftinija oprema koja zadovoljava prethodna tri kriterijuma. Pri tome, ako to finansijske mogućnosti dozvoljavaju, ne treba štedeti na kvalitetu, posebno imajući u vidu brz razvoj informacionih tehnologija, zastarevanje, kao i trend pada cena, bar kada je u pitanju hardver. Kvalitet i potencijal IS koji podržava nabavljena oprema treba da bude glavni motiv pri izboru opreme, kao što se vidi na sledećem primeru sa softverom.

Karakteristike softvera	Karakteristike korišćenja
Prost, nerazvijen softver	Mali izdaci za obradu; Komplikovano programiranje; Dugo trajanje obrada
Razvijen softver	Veliki troškovi računara; Prosto programiranje; Kratko vreme obrade

Metodi za izbor opreme

Veliki broj faktora koji se moraju uzeti u obzir pri izboru opreme otežava formalizaciju postupka izbora. Pored velikog broja **kvantitativnih faktora** (brzina procesora, brzina pristupa memoriji, brzina prenosa podataka, brzina U/I uređaja, kapacitet memorija i sl.) postoje i **kvalitativni faktori** (osobine i mogućnosti softvera, podrška proizvođača itd) koji se moraju manje-više subjektivno ocenjavati. Subjektivizam se ne može potpuno eliminisati, pa se metodi za izbor opreme kreću u širokom spektru, od neformalnih do pretežno formalizovanih metoda.

Neformalni metodi

Najpre se formira stručni tim za izbor opreme. Tim formira i definiše skup elemenata za procenu i svakom elementu daje određenu težinu, odnosno dodeljuje mu težinski koeficijent. Na osnovu datih težinskih koeficijenata svih definisanih elemenata vrši se izbor opreme.

Alternativno, konkurentna oprema se najpre svodi na sličnu konfiguraciju a zatim se proučava razvoj ukupnih godišnjih troškova po fazama razvoja IS uz procenu odnosa cena i performansi sistema. Za polaz se obično uzima osnovna konfiguracija koja se po fazama razvoja IS dograđuje. U okviru procene mogućnosti sistema, daju se posebno ocene za hardver, za softver i za proizvođača. Bira se oprema sa najvećom ukupnom ili najvećom prosečnom ocenom.

Metod faktora dobrote

Faktori dobrote odn. mere efikasnosti su kvantitativni pokazatelji (parametri) preko kojih se pokušava da se kvantitativno izraze karakteristike delova opreme ili čitave konfiguracije. Najčešće se koriste:

1. *brzina odvijanja elementarnih operacija* $No = 1/To$, gde je To vreme sabiranja celih brojeva.
2. *širina propusnog opsega memorije* $B = LR/Tc$, gde je LR dužina reči u bitovima a Tc vreme ciklusa za posmatranu memorijsku reč.
3. *efektivnost centralnog procesora* $Ecp = M \cdot Nc$, gde je M kapacitet operativne (glavne) memorije u bitovima, a $Nc = 1/Tc$ broj memorijskih ciklusa u sekundi (Tc - dužina memorijskog ciklusa).
4. *snaga računara* (empirijska formula po Knightu) $Pr = Mf \cdot No$, pri čemu je $Mf = M(Lr - 7)Wp/k$, a $No = 1012/(tc + ti/o)$. Tu je: Mf - memorijski faktor, No - broj operacija/sec, M - kapacitet operativne memorije u rečima, Lr - dužina reči u bitovima, $W = 1$ (za fiksnu dužinu reči u memoriji) odn. 2 (za promenljivu dužinu reči u memoriji), $p = 0,5$ (za naučne proračune) odn. 0,333 (za komercijalne proračune), k - konstanta zavisna od tipa računara, Tc - vreme u

mikrosec potrebno da se obavi milion računskih operacija, **ti/o** - vreme u mikrosec potrebno da se obavi milion ulazno-izlaznih operacija.

5. *faktori dobrote perifernih uređaja:*

- za spoljašnje memorije E_T - maksimalan prenosni faktor (u hiljadama znakova u sekundi);
- za štampač E_p - maksimalan broj odštampanih redova/min, znakova/sec, strana/min,... zavisno od tipa.

Metod prosečne instrukcije

Ovaj metod je pogodan za poređenje procesora. Pri upoređivanju više procesora u obzir se uzimaju brzine izvršavanja operacija, cena rada procesora po vremenskoj jedinici i težinski faktori (značaj) svake instrukcije.

Ako postoji m procesora i n instrukcija za upoređivanje, onda se cena i -te instrukcije na j -tom procesoru izračunava kao $C_{ij} = T_{ij} \cdot C_j$, gde je C_j cena rada j -tog procesora ($j = 1, 2, \dots, m$) a T_{ij} vreme izvršenja i -te instrukcije ($i = 1, 2, \dots, n$) na j -tom procesoru.

Ako bi frekventnost pojave u obradi bila jednaka za sve instrukcije, najbolji procesor bi bio onaj kod koga je zbir cena izvršenja instrukcija minimalan, tj.

$$\min \sum_{i=1}^n C_{ij}$$

S obzirom da nemaju sve instrukcije isti značaj i frekventnost, svakoj instrukciji se dodeljuje težinski faktor W_i , pri čemu je

$$\sum_{i=1}^n W_i = 1.$$

Vrednost težinskih faktora određuje se na osnovu (pretpostavljene) frekventnosti pojave instrukcija u obradi.

Metod standardnih zadataka

Ideja je da se konkurentski računari dovedu u stvarne radne uslove, tj. da rešavaju konkretne zadatke, tipične za poslove firme za koju se biraju, pa da se upoređivanjem ukupnog trajanja obrade utvrdi koji sistem je najprihvatljiviji. Ovaj metod traži da računarske konfiguracije budu približno iste.

Metod ponderisanog rangiranja

Pri oceni ovaj metod uzima u obzir najveći broj relevantnih kvalitativnih i kvantitativnih karakteristika računarskih sistema. Ocenjivanje se odvija kroz sledeće korake: 1) izbor elemenata za procenu; 2) dodeljivanje težinskih faktora elementima za procenu; 3) ocenjivanje; 4) ponderisano usrednjavanje ocena.

Kao elementi za ocenu se mogu uzeti i rezultati ostalih (prethodno navedenih) metoda. Ovaj metod je pogodan, jer uključuje rezultate drugih procena, jeftin je i pogodan za komuniciranje kupca i proizvođača.

Simulacioni metodi

Prethodni metodi upoređuju računare po izabranim kriterijumima ali ne kažu u kome će stepenu "najbolje" rešenje da odgovori zahtevima realnog sistema u kome se vrši automatizacija. Za to se koriste simulacioni metodi. Oni su naročito pogodni za analizu složenih sistema (pre svega po pitanju "uskih grla" - komponenti koje svojim performansama ne zadovoljavaju zahteve) kada nije moguće izvršiti analizu sistema analitičkim metodima. Simulacija se vrši pre nabavke računara, korišćenjem programa - simulatora (matematičko-logičkih modela koji za određene vrednosti parametara daju parcijalna rešenja). Ovi metodi zahtevaju specijalizovana stručna znanja i skupi su, tako da se za sada relativno retko koriste.

Standardna procedura izbora opreme

Nabavka i izbor opreme najčešće prolazi kroz sledeće korake:

1. prethodni širi izbor dobavljača i uspostavljanje prvih kontakata;
2. izrada pregleda tehničkih potreba novog sistema i izbor metoda za procenu opreme:
 - raspisivanje konkursa;
 - dostavljanje pregleda tehničkih potreba učesnicima konkursa;
 - procena rešenja predloženih u ponudama dobavljača;
 - konačan izbor dobavljača opreme.

Literatura

1. Živković D., Projektovanje informacionih sistema - JSD metoda, CET Computer Equipment and Trade, Beograd, 1996.
2. Champine G.A., Distributed Computer Systems - Impact on Management Design and Analysis, North-Holland, Amsterdam, 1980.
3. Clifton H.D., Business Data Systems, Prentice-Hall International (UK) Ltd., 1990.
4. Jauković M., Uvod u informacione sisteme, Tehnička knjiga, Beograd, 1992.
5. Lazarević B, Projektovanje informacionih sistema (skripta), FON, Beograd, 1995. (ili novije)
6. Martin J., Chapman K.K., Leben J., System Application Architecture - Common User Access, Prentice Hall, Englewood Cliffs, New Jersey, 1991.
7. Martin J., Information Engineering II - Planning & Analysis, Prentice Hall, Englewood Cliffs, New Jersey, 1990.
8. Martin J., Information Engineering III - Design & Construction, Prentice Hall, Englewood Cliffs, New Jersey, 1990.
9. Murdick R.G., Ross J.E., Claggett J.R., Information Systems for Modern Management, Prentice Hall, Englewood Cliffs, New Jersey, 1984.
10. Pavlić M., Sistem analiza i modelovanje podataka - projektovanje informacionih sistema, Naučna knjiga, Beograd, 1990.
11. Veljović A., Osnove objektnog modeliranja UML, Kompjuter biblioteka, Čačak, 2004.
12. Veljović A., Projektovanje informacionih sistema, Kompjuter biblioteka, Čačak, 2003.