

IMRE PETKOVIĆ

**INFORMACIONI SISTEMI I BAZE
PODATAKA**

SKRIPTA

SUBOTICA, 2001.

SADRŽAJ

1. UVOD.....	4
2. KARAKTERISTIKE POSLOVANJA SAVREMENIH PREDUZEĆA.....	6
2.1. DINAMIKA POSLOVANJA.....	6
2.2. UTICAJ POSLOVANJA NA HIJERARHIJSKU STRUKTURU PRDEUZEĆA.....	8
2.3. POSLOVANJE, KOMUNIKACIJE I INFORMACIONI SISTEM.....	10
3. POJAM SISTEMA I NJEGOVO FORMULISANJE.....	12
3.1. DEFINICIJA I POJAM SISTEMA.....	12
3.2. GRANICE I OKRUŽENJE SISTEMA.....	15
3.3. STRUKTURIRANJE – PODELA SISTEMA.....	17
3.4. DVOSTRUKI POGLED NA SISTEM.....	18
4. INFORMACIONI SISTEM – PROŠIRENI SADRŽAJ STAROG POJMA.....	20
4.1. POJMOVI I SHVATANJA.....	20
4.2. POJAM INFORMACIONOG SISTEMA – SAVREMENO SHVATANJE.....	23
4.2.1. Podatak, informacija, znanje.....	24
4.2.2. Informacione aktivnosti i informacioni događaji.....	27
4.2.3. Resursi potrebni za informacione sisteme.....	29
4.2.4. Korisnici informacionih sitema.....	29
4.2.5. Standardi i procedure.....	31
4.3. PROJEKCIJE INFORMACIONOG SISTEMA.....	32
4.4. NIVOI INFORMACIONOG SISTEMA.....	33
5. TEHNIČKA SREDSTVA – HARDVER I SOFTVER.....	38
5.1. HARDVER.....	40
5.1.1. Arhitekture računara.....	40
5.1.1.1. Masivna paralelna obrada – MPP.....	40
5.1.1.2. Simetrični multiprocesorski sistemi – SMP.....	41
5.1.1.3. Vektorske arhitekture.....	42
5.1.1.4. Mainframe računari.....	43
5.1.1.5. Specifične arhitekture.....	45
5.1.2. Osnovne ploče.....	46
5.1.3. Procesori.....	48
5.1.4. Memorije.....	50
5.1.5. Multimedija.....	53
5.1.6. I/O uređaji.....	57
5.1.7. Komunikacije.....	62
5.2. SOFTVER.....	65
5.2.1. Operativni sistemi.....	66
5.2.2. Uslužni programi.....	67
5.2.3. Aplikativni programi.....	68
6. RAZVOJ INFORMACIONOG SISTEMA.....	71
6.1. TEHNOLOGIJA RAZVOJA.....	73
6.2. PRINCIPI RAZVOJA INFORMACIONOG SISTEMA.....	76
6.2.1. Opšti principi razvoja.....	76
6.2.1.1. Trostruko planiranje i tronivovsko projektovanje.....	76
6.2.1.2. Odnos analize i projektovanja.....	78
6.2.1.3. Principi strukturiranja i iteracije.....	79

6.2.1.4.	Ispoštovanje zahteva korisnika	80
6.2.1.5.	Modeliranje, apstrakcija i modularizacija	81
6.2.1.6.	Standardizacija, dokumentacija i kontrola kvaliteta	82
6.3.	METODOLOGIJA RAZVOJA INFORMACIONOG SISTEMA	84
6.3.1.	Osnovna koncepcija	85
6.3.2.	Struktura i terminologija	86
6.3.3.	Način predstavljanja	86
6.3.4.	Sistem kriterijuma	87
6.3.4.1.	Vernost preslikavanja stvarnosti	88
6.3.4.2.	Razumljivost	88
6.3.4.3.	Jednoznačnost	89
6.3.4.4.	Minimalnost	89
6.3.4.5.	Celovitost	90
6.3.4.6.	Harmonija kriterijuma	90
6.3.5.	Algoritam projektovanja	91
6.3.6.	Rukovanje dokumentacijom	93
6.3.7.	Tehnička podrška	94
6.3.8.	Meta standardi	95
7.	PROJEKCIJA PODATAKA INFORMACIONOG SISTEMA	96
7.1.	O SAZNANJU	96
7.2.	ENTITET	97
7.3.	OBELEŽJE	97
7.4.	POVEZNIK	99
7.5.	EKSTERNA STRUKTURA TIPA ENTITETA	100
7.5.1.	Semantika kardinaliteta tipa poveznika	101
7.5.1.1.	Kardinalitet grupe M:N	101
7.5.1.2.	Kardinalitet grupe 1:N	104
7.5.1.3.	Kardinalitet grupe 1:1	106
7.5.2.	Rekurzivne veze	108
7.5.3.	Slabi tip entiteta	108
7.5.4.	Identifikaciono zavisian tip entiteta	109
7.5.5.	Veza izmedju tipova poveznika	110
7.5.6.	Tranzitivna konstrukcija tipova poveznika	111
7.6.	INTERNA STRUKTURA TIPA ENTITETA	112
7.6.1.	Zavisnosti izmedju obeležja	112
7.6.2.	Normalne forme	115
7.6.3.	Eksterna struktura tipa entiteta na osnovu normalizovane interne strukture	119
7.7.	MODEL PODATAKA, BAZA PODATAKA	119
7.8.	NAČINI OBRADJE SAZNANJA	121
8.	PROJEKCIJA OBRADJE INFORMACIONOG SISTEMA	124
8.1.	POJAVNI OBLICI SAZNANJA	124
8.2.	ULAZ, IZLAZ I MEDJUREZULTAT	126
8.3.	BAZA PODATAKA I PROJEKCIJA OBRADJE	128
8.4.	KONCEPCIJA NAVIGACIJE, UPITI I IZVEŠTAJI	131
8.5.	PROJEKTOVANJE OBRADJE	134
8.5.1.	Modeliranje obrade	135
8.5.2.	Projektovanje obrade na srednjem nivou	136
8.5.2.1.	Graf za rukovanje podataka	136
8.5.2.2.	Dijagram toka podataka	138
8.5.3.	Makro-dizajn	142
8.5.4.	Mikro-dizajn	147
8.5.4.1.	Tabele odlučivanja	147
8.5.4.2.	Grafovi precedencije	149
8.5.5.	Konzistencija obrade	150

9.	PROJEKCIJA OKRUŽENJA INFORMACIONOG SISTEMA	153
9.1.	KORISNIČKO OKRUŽENJE.....	153
9.1.1.	<i>Psihologija korisnika</i>	153
9.1.2.	<i>Psihologija rada</i>	154
9.2.	SISTEM STANDARDA.....	156
9.3.	DOKUMENTACIJA INFORMACIONOG SISTEMA	157
10.	LITERATURA	159

1. UVOD

“Danas je već opšte prihvaćen stav, da je svet sastavljen iz tri komponente: materije, energije i informacije. Takav prilaz svetu potvrđuju kako živi organizmi, tako i organizovani sistemi koje je stvorio čovek. Bez informacija organizovani sistemi ne mogu opstati. Današnji sistemi, međutim, pored toga što su organizovani, i održavaju svoju organizovanost. Objašnjenje takvog ponašanja sistema leži u neprekidnom izvlačenju informacija iz spoljašnjeg sveta o pojavama i procesima koji se u njemu odigravaju. Stabilnost sistema, koja ima presudan značaj za ocenu radne sposobnosti sistema, se procenjuje razmatranjem dinamike procesa koji se odvijaju u sistemima.” (A.J.Lerner).

Informacija se danas svugde u svetu smatra resursom koji igra ključnu ulogu u životu i poslovanju organizacija svih vrsta i veličina. Strateški značaj informacija priznaju svi savremeni stručnjaci. Važnost i neophodnost kao atributi uz pojam informacija ističu ulogu otkrivanja, organizovanja, memorisanja i rukovanja saznanja (podataka – informacija). Informacije se mogu obezbediti i pomoću računara uz pomoć softverskog produkta (proizvoda) koji se zove informacioni sistem. Razvoj informacionog sistema je složen i mukotrpan poduhvat koji pored zavidnog nivoa stručnosti na polju informatike traži i druge sposobnosti: otvorenost prema problemima drugih, agilnost i istrajnost u otkrivanju i rešavanju problema i, svakako, komunikativnost. O činionicima (elementima) informacionog sistema je pisano puno od početka sedamdesetih godina kad je krenulo intenzivno istraživanje ove oblasti. Globalni pregled informacionog sistema i njegovih elemenata, njihove osobine i njihov razvoj, ponašanje sistema kao celine se retko obrađuje u radovima. Ovde je načinjen pokušaj da se prikaže informacioni sistem i njegovi činioци (njihov razvoj) u okviru opšte (statičke) strukture, a isto tako i dinamika veza između elemenata.

Materijal je prvenstveno pripremljen za studente na drugoj godini Više tehničke škole Odseka za informatiku koji slušaju predmet »Informacioni sistemi i baze podataka« u trećem semestru sa fondom časova 2+2. Zbog ograničenog fonda časova mnogo toga se nije našlo u ovom materijalu. Prikaz prilaza i principa projektovanja informacionog sistema su bili prvobitni ciljevi u pripremi ovog materijala, koji je pretežno teoretskog karaktera. Na vežbama studenti rešavaju konkretne primere i zadatke na konkretnim tehničkim sredstvima, a na osnovu metoda sa predavanja.

Želeo bih izneti svoje subjektivno mišljenje budućim informatičarima da su samo u tom slučaju birali odgovarajući smer (užu struku) za sebe, ako im je uža oblast interesovanja ujedno i

svakidašnji hobi. I još nešto na kraju, želeo bih da im informatika ne bude samo zanimanje, već i životni poziv!

U beleškama sigurno postoje manje i/ili veće greške. Broj grešaka se smanjuje sa brojem pregleda (čitanja), ali se u potpunosti, za ovakvo kratko vreme, sve greške ne mogu eliminisati. Sve informacije u vezi poboljšanja kvaliteta materijala su dobrodošle.

mr Imre Petković, dipl. el. ing.

peti@vts.su.ac.yu

Subotica, 20. 04. 2001.

2. KARAKTERISTIKE POSLOVANJA SAVREMENIH PREDUZEĆA

2.1. DINAMIKA POSLOVANJA

Poslovanje današnjih preduzeća je izrazito teško. Moglo bi se reći da je današnje poslovanje teže, dinamičnije i rizičnije nego ikada. Ubrzani tehnološki razvoj, globalizacija svetskog tržišta i sve intenzivnije korišćenje Interneta naprosto tera poslovne ljude, a pogotovu upravljače, članove poslovnog organa organizacija, da pronadju nove ideje, nove logike, nove pristupe u poslovanju. Zadatak upravljačko-poslovnog organa jeste da obezbedi efikasno funkcionisanje preduzeća. U tom cilju je neophodno obezbediti uslove u kojima će organizacija biti u stanju da neprekidno poboljša kvalitet svojih proizvoda i/ili usluga uz neprestano smanjenje troškova poslovanja.

Postati i ostati konkurentan na tržištu je slogan i moto savremenih organizacija. To ujedno znači, da organizacije moraju obezbediti neprestanu kontrolu kvaliteta svojih proizvoda i usluga, a da bi mogli suditi o svom položaju, moraju posmatrati i svet oko sebe. Neminovno je ažurno praćenje konkurenata, dobavljača, kooperanata, zakonskih propisa i regulativa i, naravno, potrošača. Validirano je i opšte prihvaćeno tvrdjenje, da budućnost preduzeća određuju najviše potrošači, odnosno tržište. Realizacija čak i najgenijalnijih ideja je osuđena na propast, ako je tržište ne prihvata. Za obezbeđivanje potrebnih informacija o uslovima na tržištu, o granskim (srodnim) preduzećima, o komitentima se troše sve veća sredstva, jer su poslovodni organi uvideli značaj i neophodnost tih informacija. Obim tih relevantnih informacija raste zajedno sa razvojem preduzeća, odnosno proširenjem asortimana, ili, generalno proširenjem oblasti (grane) funkcionisanja. Relevantne informacije se mogu obezbediti jedino intenzivnijom komunikacijom između organizacije i okoline i između organizacionih celina unutar preduzeća.

Poslovanje današnjih preduzeća određuju prvenstveno promene koje se odigravaju u spoljašnjem i unutrašnjem okruženju. Sposobnost reagovanja ili brzine reakcije firmi na promene određuje njihovu efikasnost i uspešno poslovanje. Mobilnost i dinamika u poslovanju predstavljaju potreban uslov za uspeh današnjih preduzeća. To iziskuje kvalitetne i brze komunikacije, kao i efikasnu i brzu automatsku obradu podataka. U cilju razvoja poslovanja se vrši analiza proizvodnje: da li dobro funkcioniše sve u proizvodnom lancu, da li se i šta može poboljšati efikasnost ili nešto u kvalitetu? Da li bi se mogli proizvoditi ili usluge poboljšati sa nečim? Kako bi se mogli osvojiti nova tržišta odnosno potrošači? Odakle kupovati sirovine, gde proizvoditi, na kojem tržištu prodavati, razmišljajući globalno u geografskim lokacijama, pa čak

i u kontinentima u cilju minimiziranja cene koštanja proizvoda i u maksimiziranju prodajne cene uz pomno praćenje kvaliteta proizvoda.

Sa tog aspekta se posmatra i informacioni sistem u organizaciji: da li efikasno prati poslovne procese i da li potpomaže te procese? Da li se može informacioni sistem brzo prilagoditi novonastalim uslovima uz prihvatljive troškove, i uopšte, da li se informacije koriste za strateško, taktičko i svakodnevno odlučivanje u poslovanju? Da li su informacije koje obezbeđuje informacioni sistem dovoljne, odgovarajuće i tačne i da li su pravovremene? U današnjim uslovima se informacioni sistem sve više smatra integralnim delom samog poslovanja nego li samo kao sredstvo koje služi kao ispomoć poslovanju. Od poslovanja preduzeća se očekuje da:

- a) neprkidno povećava stepen efikasnosti svog funkcionisanja uz pomoć optimalnog angažovanja svojih resursa i da na taj način postigne željene rezultate,
- b) postane aktivan činilac na globalizovanom svetskom tržištu izgradnjom bilateralnih i multilateralnih veza, što podrazumeva realizaciju efikasne i brze komunikacije između organizacija, kao i između organizacija i saradnika i
- c) obezbedi asortiman proizvoda (robe) i/ili usluga koji je atraktivan i tražen na tržištu i da obezbedi visok kvalitet svojih proizvoda i/ili usluga.

U cilju da bi organizacija mogla ispuniti gore navedena očekivanja, ona po pravilu formuliše sledeće zahteve prema informacionom sistemu:

- 1) da doprinosi povećanju stepena efikasnosti funkcionisanja preduzeća,
- 2) da se neprestano prilagođava dinamičnim uslovima tržišta, kao i brzim promenama u samoj organizaciji,
- 3) da obezbedi prihvatanje raznorodnih i nekompatibilnih (multimedijalnih) informacija (podataka) koje stižu na najrazličitiji način, posredstvom povezanih hardverskih i softverskih resursa različitih generacija do preduzeća kao što su: LAN, Intranet, WAN, Internet, EDI, WAP itd., i
- 4) da smanjuje troškove poslovanja i razvoja¹.

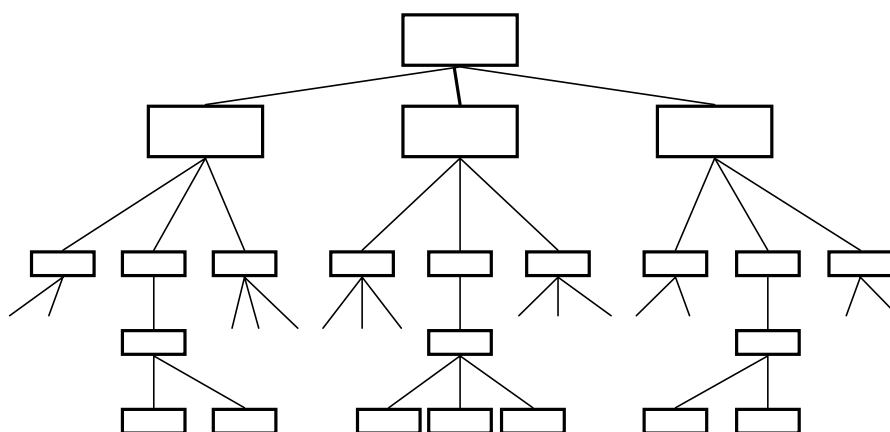
¹ Izvor: [22.], strana: 152.

2.2. UTICAJ POSLOVANJA NA HIJERARHIJSKU STRUKTURU PRDEUZEĆA

Dinamika razvoja se iz dana u dan povećava u svim oblastima nauke, tehnike, tehnologije i poslovnog života. Težnja za primenom novih tehnologija u proizvodnom procesu, stalne promene u uslovima privredjivanja i novi zahtevi potrošača neprestano formulišu nove izazove za organizacije. Odgovor ili blagovremeno i valjano reagovanje na ove izazove traži napuštanje klasične hijerarhijske organizacione strukture preduzeća.

Klasična hijerarhijska organizacija sa više nivoa rukovodjenja polazi od primene principa podele rada po funkcionalnim celinama, a unutar toga prema sadržaju srodnih poslova, sa grupisanjem, tj. razlikovanjem stepena stručnosti. Organizacija rukovodjenja i odgovornosti zasniva se na odnosu pretpostavljeni – izvršilac. Ekonomisti za klasičnu organizaciju kažu da se zasniva na linijskoj organizaciji i hijerarhiji nadležnosti unutar svakog poslovnog područja, sa krutom podelom nadležnosti i da dovodi do ostvarenja “nula sume rezultata”. To, naravno, vodi do opšte stagnacije u privredi. Klasična organizacija se slikovito prikazuje pomoću piramide sa većim brojem nivoa rukovodjenja (Sl. 1.1.). Takva struktura je suviše inertna da bi mogla zadovoljiti zahteve potrošača i uslove poslovanja, jer:

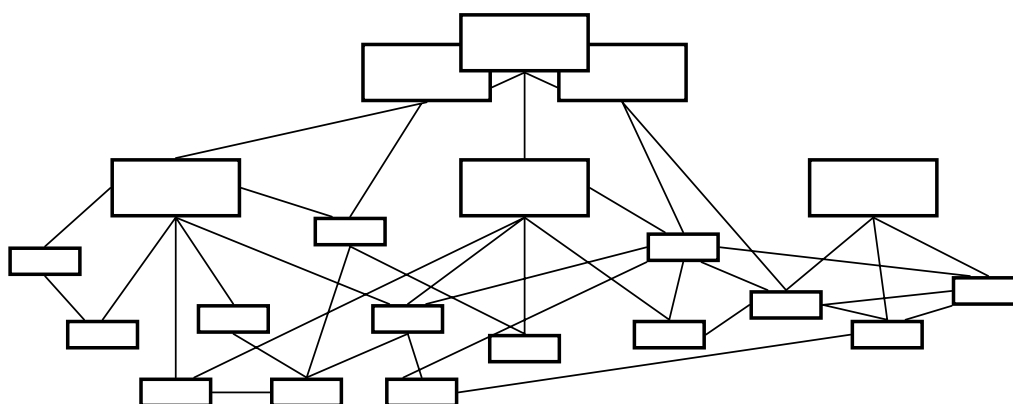
1. velik broj nivoa rukovodjenja u hijerarhijskoj strukturi može da izobličava informacije, a svakako sperčava brzu propagaciju informacija, a time i adekvatne i valjane reakcije,
2. poslovi i procesi se odvijaju u više organizacionih celina čija je saradnja, zbog hijerarhijske organizacije, otežana.



Sl. 2.1.

U cilju prevazilaženja nedostataka klasične hijerarhijske organizacije pronadjene su nove organizacione forme sa malim brojem nivoa rukovodjenja: “piramida” organizacije je sabijena. Suština ovih organizacija se ogleda u izraženim horizontalnim vezama izmedju organizacionih

jedinica (nasuprot hijerarhijskoj organizaciji, gde su samo vertikalne povezanosti bile prisutne), kako je to prikazano na Sl. 1.2. Od novih oblika organizacije su prihvaćene projektna i matrična



Sl. 2.2.

organizacija.

Pod projektnom organizacijom se podrazumeva oblikovanje ekipe za izvršenje određenog zadatka, tj. njihovog izdvajanja iz postojeće organizacione strukture. Projektna organizacija je orijentisana na ostvarenje postavljenog cilja. Razrada ostvarenja cilja je u sistematskom utvrđivanju zadataka nužnih za ostvarenje cilja, pri čemu se za svaki zadatak odredi rok završetka i dodeljuju se odgovarajući izvršioci. Za svaki zadatak se procenjuju troškovi, kao i odgovornost izvršilaca za ostvarene troškove. Rukovodilac projekta je odgovoran za realizaciju projekta: posebno za nastale troškove i dobit. On koordinira radom na projektu i vrši dinamičko uskladjene termina u izvršavanju zadataka.

Matrična organizacija je takav sistem organizacije koji povezuje klasičnu hijerarhijsku (linijsku) organizaciju sa programskim zadacima, tj. zadacima koji proističu iz trenutno aktivnih projekata. Matrični oblik organizacije je prikladan kada se u preduzeću istovremeno realizuje veći broj relativno malih projekata sa relativno kratkim vremenskim rokovima završetka. Kod matrične organizacije se formalno ne uspostavlja poseban organizacioni oblik izdvajanjem iz postojeće organizacione strukture kao kod projektne organizacije.

Nove organizacione forme se ponekad zovu i heterarhijske strukture u kojima su pored naglašenih horizontalnih veza između organizacionih jedinica prisutne i vertikalne povezanosti, samo je broj hijerarhijskih nivoa drastično smanjen u odnosu na klasičnu, hijerarhijsku organizaciju preduzeća. U savremenim organizacijama je, dakle, broj upravljačkog (rukovodećeg) nivoa znatno smanjen, jer se samo na taj način može sa njima lako i efikasno rukovati. Preglednost organizacije, njeno efikasno funkcionisanje i lako upravljanje sa organizacijom obezbeđuju opstanak u konkurentskoj borbi za zadovoljavanje potrebe potrošača.

Uobičajeno je da su u savremenoj organizaciji prisutni samo tri nivoa u organizacionoj strukturi. Top menadžment (CEO – Chief Executive Officer), koji predstavlja “vrh piramide” ili prvi nivo, na kojem se određuje strategija preduzeća, uslovi i način rada, kao i odgovarajući resursi. Drugi nivo je nivo srednjih rukovodilaca koji realizuju strategiju preduzeća organizacijom poslova u timovima (na projektima): sinhronizuju poslove članova tima, kontrolišu izvršene poslove i motivišu svoje saradnike – izvršioce.

Treba istaći novi trend u realizaciji poslova u novim heterarhijski organizovanim preduzećima, a to je intenzivno angažovanje spoljnih saradnika za tačno određene i dobro definisane zadatke, delove nekog većeg projekta. Oglašavanje tih dobro specificiranih zadataka se danas realizuje preko Interneta, šta više, sve aktivnosti se u zadnje vreme realizuju preko pomenutog medijuma: prijava pojedinaca (budućih spoljnih saradnika) na oglašavane zadatke, ugovaranje posla, slanje detaljne specifikacije za ugovoreni zadatak, slanje rešenja zadataka ili uradjenog posla od strane izvršioca kao i plaćanje.

2.3. POSLOVANJE, KOMUNIKACIJE I INFORMACIONI SISTEM

Poslovanje svake savremene organizacije je danas nerazdvojno povezano sa komunikacijama. Samo svakidašnje funkcionisanje organizacije je bukvalno uslovljeno od strane efikasnog funkcionisanja komunikacionog i informacionog sistema. Preduzeće, naime, mora neprestano da razmenjuje podatke sa okolinom da bi moglo da opstane. Informacije (podaci) o tržištu, konkurenciji, novim proizvodima, ostalim granskim organizacijama i organizacionim jedinicama stižu u preduzeće preko različitih komunikacionih kanala. Obrada tih podataka i priprema esencijalnih, obradjenih podataka u što kraćem vremenskom intervalu za potrebe upravljača je zadatak informacionog sistema. Oblik generisanog izveštaja od strane informacionog sistema mora biti prilagodjen grupi korisnika kome je on namenjen.

Rad komunikacionog sistema u celini se ne može zamisliti bez računara, odnosno računarske tehnike. Osnovne funkcije u klasičnoj i bežičnoj telefoniji realizuju računari: uspostavljanje veze, evidencije poziva, obračun telefonskih i raznih ostalih usluga, itd. Funkcionisanjem manjih mreža računara na istoj geografskoj lokaciji (LAN), srednjih mreža računara postavljenih na bliskim geografskim lokacijama (Intranet), kao i velikih mreža računara (WAN) kao što je Internet upravljaju takodje računari.

Razvoj komunikacija, znači, zavisi i od razvojnog nivoa računarske tehnike. Kao konkretni rezultati čisto komunikacionog razvoja mogu se naznačiti samo sledeća tri: digitalizacija mreža,

bežični prenos podataka i prenos podataka pomoću širokog spektra. Prve dve sfere su poznate, a za treću treba samo malo objašnjenja: pomoću širokog spektra se realizuje brzi prenos velike količine podataka. Prenos zvuka, slike, animacije i filmova se vrši na ovaj način.

Sve ostalo je zajednički razvoj komunikacije i računarske tehnike: primanje i slanje poruka sa računara, mobilnog računara ili mobilnog telefona (E-mail), EDI – razmena poslovnih informacija direktno izmedju računarskih sistema bez tvrde kopije tj. papira, WWW, Gopher ili WAP aplikacija na mobilnim telefonima, interaktivna razmena podataka: IRC, prenos glasa i slike preko Interneta u okviru telekonferencije, kao i druge mogućnosti kao Telnet, Finger, Ping itd.

Za zadovoljavanje pojedinačnih zahteva korisnika je razvijeno više različitih tipova usluga kao što su teleshopping, teleteaching, teleworkig, telecooperation, e-business, e-banking, e-hospital itd. Broj razvijenih mrežnih i Internet aplikacija je iz dana u dan veći. Bez ovih vidova prenosa podataka poslovanje je danas nezamislivo.

Aplikacije koje se razvijaju u okviru gore navedenih usluga su delimično specifične, zbog prikaza rezultata obrade na Internetu, a većim delom su klasične, jer obrada je ista bez obzira na način prikazivanja rezultata. Razvoj aplikacije, odnosno informacionog sistema, a naročito onaj deo pre realizacije (programiranja) ne treba da pretrpi nikakve izmene iako se usluga (obrada, informacioni sistem) pripremi kao Web aplikacija. Upravo je to zadatak ovog kursa, da upozna zainteresovane kako se izvodi razvoj informacionog sistema.

3. POJAM SISTEMA I NJEGOVO FORMULISANJE

Čovek već hiljadama godina formuliše pojmove za označavanje pojava, događaja i objekata iz okolnog realnog sveta. Čovek je, međutim, društveno biće¹ i zato treba da koristi iste pojmove, odnosno isti jezik da bi se mogao sporazumeti sa drugim ljudima. Pojam »sistema« se danas sve češće koristi u različitim sredinama. Često se taj pojam koristi i neadekvatno. Reč sistem je postala omiljena u poslednje vreme: ona je sad u modi, jer upućuje na odvažnost lica koje ga koristi i/ili posla kojeg karakteriše.

Nepobitna je činjenica da je svaka stvar na određenom nivou kompleksna. Zato i nije čudno što se reč sistem (uglavnom eksplicitno, a ponekad samo implicitno) često pojavljuje u današnjem jeziku. Pošto se reč sistem može primeniti za svaku složenu stvar, određivanje samog pojma sistema je jako teško, moglo bi se reći skoro nemoguće. Za izučavanje projektovanja informacionih sistema je, međutim, neophodno da poznajemo tu najopštiju filozofsko-metodološku kategoriju: sistem.

3.1. DEFINICIJA I POJAM SISTEMA

Postoji više definicija za određivanje suštine sistema. Neke su neobično kratke, ali ipak ukazuju na bitne osobine sistema. Prvu definiciju sistema dao je kanadski biolog Ludvig von Bertalanffy u svome radu General System Theory objavljenom pedesetih godina, u kojem kaže:

“Sistem je kompleks elemenata koji su u medjusobnoj interakciji”².

U radovima B. Langefors je šestdesetih godina razradio generalne metode projektovanja informacionih sistema na bazi formalne teorije, pa se u njima takodje pojavljuje opšta definicija sistema koja glasi:

“Sistem je skup objekata koje nazivamo delovima, koji su medjusobno povezani na neki način.”³.

Ni u novije vreme nema “konkretnijih” definicija pojma sistema od prethodno navedenih, sem možda pojave pojma okruženja sistema, što ukazuje na to, da se tako opšti pojam ne može previše precizno opisati. Tako B. Halassy kaže:

¹ “Bez mnogo čega čovek može da bude, samo ne bez čoveka” – L. Bjorne u [16.]

² Izvor: [22.], strana: 26.

³ Izvor: [4.], strana: 173.

“Sistem je organizovani skup (njegovih) medjusobno povezanih činilaca”⁴.

M. Raffai svoje vidjenje sistema iskazuje na sledeći način:

“Sistem je skup medjusobno zavisnih elemenata organizovanih u određenu strukturu, pri čemu su elementi pored medjusobne zavisnosti usaglašeni i toj strukturi”⁵.

B.Djordjević u monografiji Vodoprivredni sistemi navodi čak tri definicije sistema u cilju što boljeg osvetljavanja ili dočaravanja suštine odnosno pojma sistema. U nastavku slede njegove definicije:

“Sistem je neka struktura u koju se u nekom vremenu uvodi materija, energija i informacija, a iz kojeg se u nekom drugom vremenu dobija neki izlaz u vidu materije, energije i informacije”⁶.

“Sistem je skup objekata (elemenata) ujedinjenih nekim oblikom interakcija. Kod realnih sistema elementi predstavljaju objekte za neku transformaciju resursa, dok su veze kanali za razmenu mase, energije i informacija.”⁷

“Sistem predstavlja celovit kompleks elemenata koji se nalaze u nekim medjusobnim zavisnostima, doprinoseći kroz njih nekoj zajedničkoj svrsi. Okružen je okolinom sa kojom je u stalnoj interakciji.”⁸

Rezimirajući na osnovu prethodnih definicija treba istaći da je potrebno formulisati tako uopšteni opis (definiciju) sistema koji verno odslikava svaki sistem (da sadrži sve opšte karakteristike svih sistema) tj. da ne isključuje nijednu stvar koja se može smatrati sistemom. U tom slučaju, međjutim, dobijemo premalo informacija o sistemima – samo njegove bitne karakteristike:

- sistem je skup elemenata izdvojen kao celina iz svih postojećih elemenata realnog sveta,
- interakcija između elemenata sistema je neminovna, da bi se postigli zajednički ciljevi,
- sistem je u kontinualnoj vezi sa svojom okolinom i

⁴ Izvor: [11.], strana: 16.

⁵ Izvor: [22.], strana: 26.

⁶ Izvor: [6.], strana:

⁷ Izvor: [6.], strana:

⁸ Izvor: [6.], strana:

- u sistemu se odvija transformacija i razmena tri osnovne kategorije ili veličine sa okruženjem i to: materije, energije i informacije.

Ne znamo, međutim, ništa konkretnije ni o elementima ni o njihovim međusobnim vezama ni o interakciji između sistema i njegove okoline. Dalje detaljisanje, naime, bi suzilo pojam samog sistema, odnosno tako bismo neke sisteme eliminisali već na početku: u trenutku prihvatanja definicije o sistemu.

Naša definicija sistema će nastojati da obuhvati sve četiri bitne karakteristike sistema, odnosno biće malo bogatija od svih prethodnih pojedinačnih definicija:

Sistem je organizovani skup (organizovana celina) činilaca (elemenata) od kojih se svaki element (činilac) nalazi u određenoj medjuzavisnosti (u određenoj strukturi) sa ostalim činiocima (elementima) te celine i tako ostvaruje svoju pripadnost toj celini. Sistem postoji radi ostvarenja nekog cilja i u stalnoj je interakciji sa svojim okruženjem.

Pošto elementi sistema mogu biti jako različiti što se karakteristike, ponašanja i naravi tiče, a i veze između elemenata kao i interakcije sa okolinom su u realnom svetu raznovrsne, pojavile su se jako različite sistematizacije (klasifikacije) samih sistema u prošlosti [Djordjević] [Raffai M. IRF]. Sa tačke gledišta projektovanja informacionih sistema prethodno pomenute klasifikacije su od sporednog značaja. Treba se, međutim, osvrnuti na pojam strukture sistema.

Na osnovu gore navedenih definicija se može zaključiti da pojam struktura obuhvata elemente sistema i njihove medjuzavisnosti. **Struktura sistema** ima svoje statičko i dinamičko obeležje. **Statički prilaz strukturi** sistema podrazumeva određivanje elemenata sistema, njihovih međusobnih veza u nekom konkretnom (statičkom) stanju sistema, kao i uzročno-posledične veze (logiku) između sastavnih delova sistema. **Dinamički pristup strukturi** sistema definiše aktivnosti odnosno funkcije elemenata ili sastavnih delova sistema koje treba da se realizuju da bi se zacrtani cilj postigao.

Nas će ubuduće interesovati samo sistemi koji predstavljaju rezultat ili proizvod kolektivnog razmišljanja više ljudi⁹, koji su zainteresovani u ostvarenju nekog konkretnog cilja. Takvi sistemi se moraju manje ili više zasnivati na odgovarajućim dogovorima ili konvencijama.

Pojam sistema nije apsolutni pojam. Svaki individuum može, a nekad mora da određuje, da definiše za sebe sistem. Šta više ista osoba, u zavisnosti od njegovog interesovanja, može i/ili mora formirati raznovrsne sisteme o istom predmetu (istoj stvari). Ilustracije radi razmislimo šta

⁹ “Da bi se razumno postupalo, nedovoljan je jedan um.” F.I. Dostojevskij u [16.]

predstavlja sistem za korisnika-vlasnika, auto-limara, auto-električara, auto-lakirera i auto-mehaničara ako posmatraju putničko vozilo!

Princip relativnosti za sisteme Langefors definiše na sledeći način:

“Svaki sistem koji je podvrgnut uticaju svoje sredine je podsistem nekog većeg sistema i svaki deo sistema je potencijalno neki sistem”¹⁰.

Medjusobni uticaj (akcije i reakcije) izmedju posmatranog sistema i njegovog okruženja znači možemo smatrati kao interakcija posmatranog sa drugim sistemima. Tada sistemi koji su u korelaciji predstavljaju delove nekog većeg sistema (supersistema). Sa druge strane, pošto nema nekog ograničenja u dekompoziciji sistema, svaki deo posmatranog sistema se može smatrati sistemom.

3.2. GRANICE I OKRUŽENJE SISTEMA

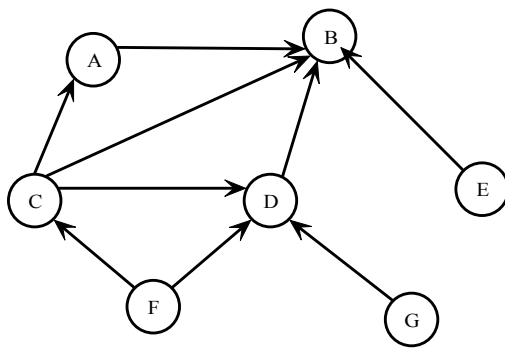
U cilju analize sistema, prvo ga treba definisati ili odrediti njegove granice. Odredjivanje granice sistema je usko povezano sa metodom opisivanja sistema. Postoje dve fundamentalne metode za opisivanje sistema:

- grafičke i
- algebarske metode.

Algebarske metode opisivanja se najčešće primenjuju u tehnici u oblasti automatskog upravljanja. Grafičke metode opisivanja sistema se takodje razlikuju u humanim i prirodno-tehničkim naukama.

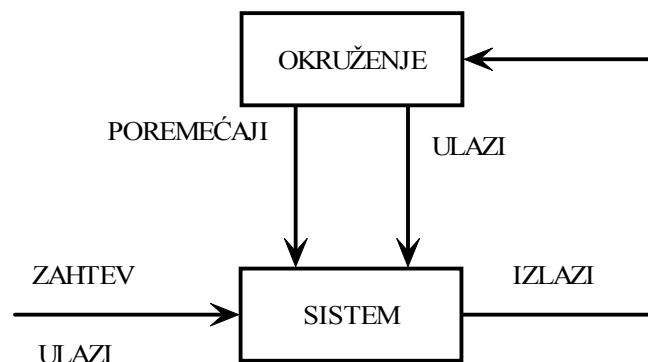
Formalni opis sistema u tehničkim naukama polazi od definicije samog sistema, tačnije od prve i druge bitne karakteristike sistema, tj. da se on sastoji od medjusobno povezanih elemenata (slika 2.1.). To je u stvari usmereni graf sa dijagramskim prikazom relacija prethodjenja i članova prethodjenja (ili pak relacija sledjivanja i članova sledjivanja). Taj način predstavljanja naglašava unutrašnju statičku strukturu sistema. Za ilustraciju interakcije izmedju sistema i njegovog okruženja koristi se grafički prikaz posebnog tipa, koji se nalazi na slici 2.2.

¹⁰ Izvor: [4.], strana: 174.



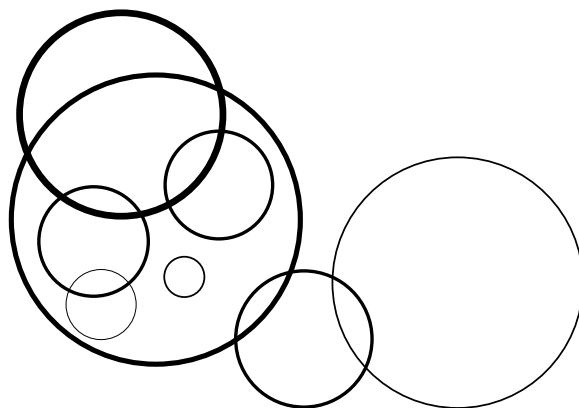
Slika 3.1.

U humanim naukama i u informatici se sistemi obično prikazuju kao krugovi. Kružnica naime deli našu celokupnu realnost na dva dela: na deo koji je za nas od posebnog značaja – to je naš sistem – i na deo koji ne spada u žižu našeg interesovanja, ali je svakako u interakciji sa našim sistemom – a to je naravno okruženje. U životu je sve povezano sa svačim, međjutim, celu našu stvarnost je nemoguće (pa čak da je i moguće – nema puno smisla) obuhvatiti. Upravo zato veštački treba povući granicu i na taj način označiti sistem – deo stvarnosti koji nas najviše zaokupljuje. Pre razvoja samog sistema mora da se sprovede analiza, a prvi korak analize je sâmo raspoznavanje sistema ili prepoznavanje činioca (elemenata) sistema. Povlačenjem granice sistema se vrši označavanje objekta razvoja ili same suštine sistema. Sistemi mogu biti u različitim međusobnim vezama (slika 2.3.).



Slika 3.2.

Odredjivanjem granice sistema treba označiti koji su ti elementi i koje su njihove međusobne veze koje pripadaju dotičnom sistemu, a takodje treba potvrditi koji su ti elementi (sa pripadajućim vezama) koji ne spadaju u posmatrani sistem.



Slika 3.3.

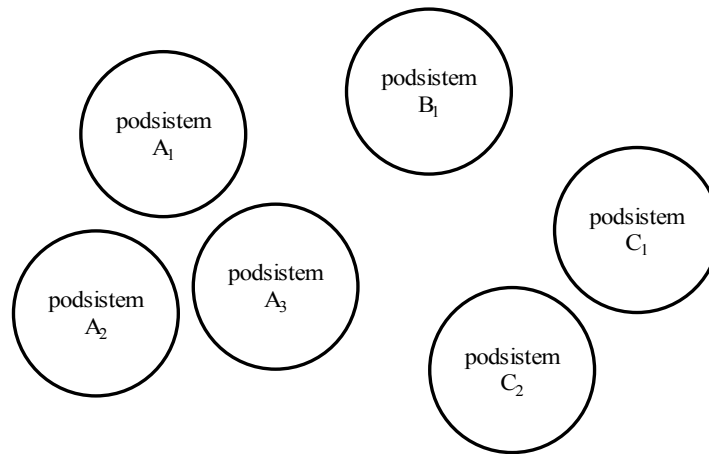
Granice sistema treba striktno i jednoznačno označiti! To je izrazito težak i suptilan zadatak, ali dobro povučene (definisane) granice stvaraju pogodnu osnovu za analizu sistema. Činioce okruženja ne smemo u potpunosti odbaciti, jer smo povlačenjem granice sami veštački prekinuli neke postojeće veze između činioca na različitim stranama granice. Preko tih veza između elemenata na različitim stranama granice će se delimično realizovati interakcije samog sistema sa okolinom.

Na prethodnim stranama je već bilo zaključeno da je sistem relativan pojam, drugim rečima da svako može odrediti sistem za sebe i to samovoljno. Postavlja se pitanje da li je pod takvim uslovima moguće formulisati neku objektivnu meru, definiciju ili preporuke za određivanje granice sistema? Gde se objektivno nalaze granice sistema niko striktno ne može da kaže: granice sistema zavise od suštine samog sistema.

3.3. STRUKTURIRANJE – PODELA SISTEMA

Princip relativnosti sistema jednoznačno formuliše stav da je deo sistema takodje sistem. Sistemi su po pravilu dosta (moglo bi se reći: previše) kompleksni da bi se mogli obraditi u celini i zato se dele na manje celine: podsisteme. Taj princip su poznavali već i Rimljani, o čemu svedoči njihova uzrečica: „Divide et impera!“. Treba, međutim, odmah naglasiti da podela sistema na delove ne sme rezultovati međusobno nezavisne podsisteme (delove sistema). Primer dobro strukturiranog sistema se nalazi na slici 2.3., ako krugove smatramo podsistemima nekog posmatranog sistema (ili delovima nekoliko sistema). Zavisnost između delova sistema (podsistema), se znači simbolički predstavlja preklapanjem krugova. Neodgovarajuća podela sistema se nalazi na slici 2.4. na kojoj su prikazani međusobno nezavisni podsistemi tri različita sistema. Sa druge strane o podeli sistema se može razmišljati samo nakon određivanja samog

sistema. Odredjivanje sistema se sastoji od upoznavanja i shvatanja sistema i njegovih ciljeva, a to znači od označavanja elemenata sistema i definisanja njegovog unutrašnjeg i spoljašnjeg okruženja, tj. od povlačenja granice sistema.



Slika 3.4.

Danas još ne postoji objektivna metodologija za podjelu sistema na delove koja bi generalno odredila kako treba podeliti neki (bilo koji) sistem na podsisteme. Evidentno je, međutim, to da se strukturiranje sistema može realizovati samo nakon jednoznačnog definisanja sistema, tj. odredjivanja:

- cilja (zbog čega u stavri funkcioniše sistem),
- sadržaja (označavanjem njegovih činioca),
- granice (klasifikacijom označenih elemenata ili činioca) i
- unutrašnjeg i spoljašnjeg okruženja sistema.

3.4. DVOSTRUKI POGLED NA SISTEM

Dvostruki pogled na sisteme se može naći i u starijoj stručnoj literaturi. Odavno je izrečena misao da postoje

- primarni i
- sekundarni sistemi.

Ova podjela je formulisana ne na osnovu važnosti sistema, već po izvornosti. U nekoj proizvodnoj organizaciji, naime, proizvodnja predstavlja primarni sistem, a sama evidencija (saznanje) o njoj sekundarni. Bez primarnog sistema, sekundarni i ne može egzistirati: bez

proizvodnje ne bi bilo ni evidencije o njoj. Druga je situacija, međutim, u osiguravajućim zavodima, bankama i statističkim zavodima. Tu primarni sistem predstavljaju statistike (o proizvodnji pojedinih preduzeća, o broju saobraćajnih nezgoda i prosečnoj šteti, o kretanju ponude i potražnje novca, o kursevima između valuta, itd.), a sekundarni proizvodnja, ili sastavljanje raznih tipova izveštaja.

Saznanje (spoznaja) o realnosti je svakako samo jedna slika u nama o stvarnosti koju mi formulišemo-stvaramo o realnom svetu. Stvarnost je dakle prvobitni (primarni) momenat, a saznanje o njoj samo sekundarni.

Realni sistem (proizvodnja u proizvodnim organizacijama ili statističke informacije u osiguravajućim i statističkim zavodima) je onaj o kojem želimo voditi izvesna saznanja i zovemo ga primarni sistem, a samo sakupljanje i rukovanje saznanjima o primarnom sistemu što realizuje informacioni sistem smatramo sekundarnim sistemom. Umesto pojma realni sistem se u zadnje vreme sve više koristi “moderniji” izraz i skraćenica: University of Discourse (UoD), u smislu “o čemu se govori” ili “o čemu je reč”. Novi pojam koji je sad u modi suštinski gledano ništa bolje ne određuje stvar za koju je uveden od starog, dobrog pojma, realnog sistema.

4. INFORMACIONI SISTEM – PROŠIRENI SADRŽAJ STAROG POJMA

4.1. POJMOVI I SHVATANJA

Danas je korišćenje računara za izvršavanje raznoraznih poslova svakako neminovnost, a pored toga je postalo i moda. Ko nema računar, ili ne zna koristiti računar, nije više u “savremenom trendu”. To je sasvim logično i prihvatljivo. Postoje, međutim, razna pogrešna shvatanja u vezi informatike i računarske tehnike koja proističu uglavnom iz neznanja. Danas naime najveći broj korisnika – početnika računara smatra sebe informatičarem, a kao potvrdu za to navodi koliko pisama je napisao pomoću softvera Word2000, koliko crteža je sastavio u Paintbrush Picture okruženju i koliko je slao i primao E-mail-ova u zadnjih mesec dana. Sve se više čini da su se pojmovi cilj, način i sredstva za realizaciju (informacionog) sistema izmešali: ako naime neki informacioni sistem (aplikacija) sporo radi ili ne radi besprekorno kupuju snažniju mašinu i/ili nabavljaju najnovije verzije raznih softvera, umesto da se okrenu samoj aplikaciji, odnosno da analiziraju rešenje zadatka ili problema (možda baš u tom grmu leži zec!).

Treba naglasiti činjenicu da stručnjak koji je vešt i iskusan u korišćenju računara i raznih softvera nije ujedno (ili automatski) i informatičar! Nenadmašni u korišćenju računara i računarskog softvera se ne mogu smatrati informatičarem iz sledećeg prostog razloga: problemi u informatici nisu (samo ili pre svega) tehničke prirode! Primena računara, znači, nije ujedno i informatika. U informatici najveći problem predstavljaju brojnost i kompleksnost činilaca ili elemenata informacionih sistema.

Pojam informatike je definisan na nekoliko načina do danas. Ovde ćemo navesti jedan sažeti, kraći i jedan opširniji opis pojma informatike. Po opširnijoj definiciji **informatika se bavi raznim modalitetima (načinima) strujanja, prenosa informacija, metodima njene obrade i korišćenja, njenim delovanjima na produktivnost i efektivnost (efikasnost) njenom primenom za praćenje, kontrolu i upravljanje, kao i ulogom informacija u razvoju privrede i društva**¹. Po ovom opisu informatika predstavlja jako opširan pojam: obuhvata integraciju informacija (podataka) u sistem, informatička sredstva, načine obrade informacija od trenutka nastanka informacija (tačnije podataka) do njene isporuke do krajnjeg korisnika, pa sve do raznoraznih uticaja informacija na privredne i društvene subjekte. Po ovom shvatanju informatika predstavlja interdisciplinarni pojam, koja u sebi sadrži telekomunikacije, računarsku tehniku, primenu računarske tehnike, primenu mikroelektronike i robotike, organizaciju,

¹ Izvor: [20.], strana: 8.

upravljanje, kao i odgovarajuće oblasti privrednih i društvenih nauka. Ta definicija je preopširna za naše potrebe.

Po sažetoj definiciji **informatika je nauka za upoznavanje, efikasno i svrsishodno organizovanje, sredjivanje i obradu saznanja (podataka ili informacija)**. Informatičarem se može smatrati stručnjak koji je vešt u toj oblasti².

Informatičar u procesu shvatanja, predstavljanja i obrade realnog sveta i njegovog informatičkog razradjivanja uvek treba da ima u vidu sledeće tri suštine, kao i njihove medjusobne veze:

1. objektivna stvarnost (realni sistem – RS),
2. slika stvarnosti na osnovu poznatih saznanja o njoj (informacioni sistem – IS) i
3. skup sredstava (hardvera, softvera, lifeware-a i orgvera, tj. sistema sredstava – SS).

Tu se radi o tri sistema koji se delimično preklapaju, ali su u isto vreme i relativno nezavisni (slika 4.1.). Te tri suštine moraju biti medjusobno uskladjene, odnosno moraju biti u što većoj harmoniji. Taj stav se danas sve više gubi iz vida. Današnji je trend da se kupuju što snažnije mašine (računari) i najnovije verzije softvera.



Slika 4.1.

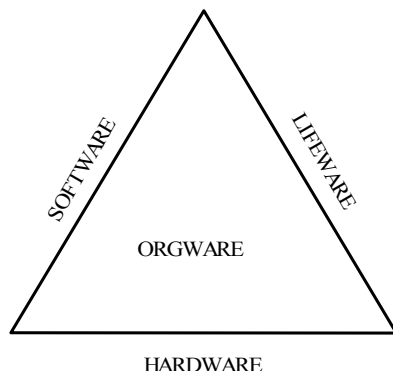
Sistem sredstava se može prikazati i grafički³, kao što se to nalazi na slici 4.2. Grafički prikaz dobro odslikava sintezu ili medjusobnu povezanost te četiri komponente sistema sredstava.

Hardver (hardware) čine materijalni sadržaji sistema sredstava: računar sa svim ulaznim, izlaznim i ulazno-izlaznim kao i uređajima za prenos podataka na daljinu, dodatni uređaji za sakupljanje, prenos, predobradu, memorisanje i distribuciju podataka. Jednom rečju hardver je „celokupno gvoždje”, ili „sve što se može opipati”. Hardver računara se intenzivno razvija. Naše aplikacije koriste jedva nekoliko procenata ukupnog hardvera, no današnji softveri su jako

² Izvor: [11.], strana: 35.

³ Izvor: [6.], strana:

zahtevni prema performansama hardvera i hardver se samo upravo zbog njih i razvija. Prema današnjim procenama razvoj hardvera prednjači oko 10-15 godina ispred razvoja softvera.



Slika 4.2.

Softver obuhvata sve programsko-algoritamske sadržaje: operativni(e) sistem(e), uslužne programe, jezičke procesore (programske jezike), sisteme za rukovanje bazama podataka, editore tekstova i tabela kao i raznorazne softverske produkte za rešavanje opštih problema na svim mogućim poljima ljudske aktivnosti. Razvoj softvera je na prvi pogled brži i očigledniji od razvoja hardvera u zadnje vreme. Primena softvera, međutim, kaska za njegovim razvojem (razvojem hardvera) čak 20 godina po nekim procenama. Neki softveri za rukovanje bazama podataka naime toliko su kompleksni, da ih laici skoro ni pokrenuti ne znaju, a i najsposobniji projektanti i programeri mogu upoznati i razumno (u praksi) primeniti svega nekoliko procenata njihovih mogućnosti.

Lifeware sistema sredstava predstavlja kadrove neophodne za izvršavanje poslova na svim segmentima u toku životnog ciklusa informacionog sistema. Kadrovi su danas dobro pripremljeni za korišćenje računara i najnovijih softvera, ali imaju pogrešan stav (i shvatanja) prema razvoju informacionih sistema.

Orgver sistema sredstava predstavlja njegovo organizaciono ustrojstvo. Povezuje prethodno opisane tri komponente u organizacionu strukturu sa ciljem da se ispune ciljevi informacionog sistema zacrtani u zahtevima naručioca. Orgver je u stvari metod organizacije saznanja.

U cilju besprekornog funkcionisanja informacionog sistema sve četiri komponente sistema sredstava moraju biti na visokom nivou harmonije, odnosno međusobno dobro uskladjene i u potpunosti kompatibilne. Zapostavljanje bilo koje komponente rezultuje u drastičnom smanjenju efektivnosti informacionog sistema bez obzira na kvalitet i uskladenost ostale tri komponente.

4.2. POJAM INFORMACIONOG SISTEMA – SAVREMENO SHVATANJE

Kako samo ime kaže informacioni sistem je takodje sistem, dakle i za njega važe sve zakonitosti koje su otkrivene i dokazane u naučnoj oblasti teorije sistema. Treba, međjutim, naglasiti da pojam informacioni sistem nije ekvivalentan pojmu sistem informacija. Sistem informacija označava samo sredjivanje, odnosno organizaciju informacija, a informacioni sistem obuhvata i razne druge elemente sasvim različite naravi.

Pošto je naš prvenstveni cilj upoznavanje sa suštinom informacionih sistema kao i sa suštinom njihove metodologije razvoja, red je da se na početku definiše pojam informacionog sistema.

Informacioni sistem je organizovani skup sledećih činilaca ili elemenata kao i njihovih medjuveza:

1. podataka (informacija),
2. sa njima povezanih informacionih događaja,
3. nad njima izvršenih aktivnosti,
4. potrebni resursi vezani za prethodna tri elementa, kao i za razvoj i funkcionisanje informacionog sistema
5. korisnika informacija (podataka) i
6. standarda i procedura koji upravljaju napred navedenim činilcima.⁴

Pojmove korišćene u prethodnoj definiciji treba i detaljnije objasniti u daljem tekstu, ali pre toga analizirajmo gornju formulaciju pojma informacionog sistema. Ono što se na prvi pogled može zaključiti jeste taj momenat, da informacioni sistem definisan gornjim izrazom spada u porodicu inhomogenih sistema. Tvrdnja je u potpunosti osnovana, jer kako elementi tako i njihove medjusobne veze su raznorodne, odnosno različite. Korisnici, recimo, isto tako spadaju u informacioni sistem kao i hardver, softver, događaji, podaci ili konvencije (standardi).

Pridevi odnosno atributi koji stoje uz izraz informacioni sistem su često pogrešni, neodgovarajući, a ponekad predstavljaju tautologiju. Atribut integralni se često koristi uz izraz informacioni sistem, a ponekad se ukombinuje još i pridev kompleksni, tako da fraza ima oblik: „kompleksni integralni informacioni sistem”. Pošto je informacioni sistem inhomogeni sistem, on je već »ab ovo« ili a priori kompleksan, dakle atribut kompleksan predstavlja tautologiju. Sa druge strane atribut integralni takodje predstavlja tautologiju, jer je svaki sistem integralan. Na osnovu definicije iz teorije sistema znamo da je sistem organizovani (znači integralni) skup

⁴ Izvor: [11.], strana: 43.

činilaca (elemenata). Formulacija oblika »upravljački informacioni sistem« je možda još nezgodnija fraza, mada se ona često koristi i na engleskom govornom području (MIS – Management Information System). Upravljački informacioni sistem u stvari i ne postoji sam za sebe. Informacije potrebne za rukovodioce ili upravljače (menadžere) potiču iz osnovnih informacija, u stvari predstavljaju neku esenciju ili izvod iz osnovnih informacija. Sve informacije koje mogu biti potrebne za upravljače za donošenje bilo kojih odluka nalaze se u nekim segmentima (podsistemima) »sveobuhvatnog kompleksnog integralnog informacionog sistema«, kojeg treba osloviti prosto i jednostavno kao informacioni sistem.

4.2.1. Podatak, informacija, znanje

Pojmovi kao što su podatak, informacija i znanje su objašnjeni na više mesta u stručnoj literaturi. Svugde u objašnjenjima postoji semantička razlika između podatka i informacije.

Najprostiji prilaz u objašnjenju pojma podatak naglašava da je on zapis neke konkretne činjenice na proizvoljnom medijumu (papiru, magnetskom medijumu, itd.) nezavisno od njegove dalje sudbine, tj. nezavisno od toga da li će biti upotrebljen za donošenje poslovnih odluka i/ili upravljanje ili ne. Ako se podatak upotrebi za upravljanje, tj. u procesu odlučivanja, on se transformiše u informaciju. Zapis zadržava svojstvo podatka sve dok postoji verovatnoća da će se na neki način transformisati u informaciju, a posle gubi to svoje svojstvo⁵.

Takvo shvatanje pojma podatka i informacije u današnje doba već ne može da opstane. Opšteprihvaćen je danas stav u vezi pojma podatka, da je on konkretizacija pojave (ili pojavnog oblika) informacije, grupa simbola koja se može obraditi na računaru, ali sa suženijim značenjem i manjim značajem od informacije. Informacija je bogatija u semantičkom smislu od podatka. Podatak je neprotumačeno saznanje ili neprotumačena informacija koja se može protumačiti. Informacija je, prema tome protumačeni podatak. Podaci nose u sebi informaciju, ali količina informacionog sadržaja podatka je takodje relativan pojam, jer u velikoj meri zavisi od primaoca podatka (informacija). Isti podatak, naime za jednog primaoca ostaje samo podatak (znači on ga nije mogao protumačiti), a za drugog primaoca postaje informacija (ako za njega predstavlja novo saznanje).

Podatak je, znači, takav neprotumačeni niz simbola ili znakova koji opisuje karakteristike (obeležja) stanja objekata ili uopšte promena u realnom sistemu sa ciljem kasnijeg korišćenja i/ili obrade, a obuhvata se u formi pogodnoj za memorisanje i prenos.

Istraživači koji se bave teorijom informacija se ni do današnjeg dana nisu mogli da se usaglase u vezi precize definicije informacije. O tome svedoče sledeće formulacije definicije informacije.

B. Langefors ima sledeći stav:

“Informacijom se smatra svako znanje, poruka koja se može iskoristiti za neko dokazivanje ili omogućuje donošenje odluka.”⁶

U Webster’s Dictionary-ju se takodje nalazi definicija informacije:

Informacija je znanje, inteligencija, vest, odnosno njihova predaja ili primanje.

Interesantnu formulaciju je dao H. Rittel, norveški istraživač:

“Informacija je takva aktivnost koja rezultuje u izmeni nečijeg znanja.”⁷

Novi momenat u sledećoj definiciji je taj da se informacijom smatra sredjeni niz već poznatih ili postojećih podataka:

“Informacija nije ništa drugo, već nesredjeni niz podataka ili činjenica dovedenih pomoću izvesnih procesa na upotrebljivu formu.”⁸

Slično mišljenje o pojmu informacije ima i E. Noszkay:

“Informacija nije generalno novo saznanje, nego takvo novo odnosno otkriveno saznanje koje nastaje na osnovu sistematizacije, uporedjivanja, analize i vrednovanja već poznatih saznanja (podataka, činjenica)”⁹

Informacija je protumačeni podatak, iskaz, saznanje, saopštenje, vest, koja može da posluži za smanjenje neizvesnosti, za donošenje odluka, za upravljanje i uvek predstavlja kvalitet više od podatka i ostalih gore navedenih pojmova, jer angažuje svog primaoca: za razmišljanje, za neki izbor, za neku aktivnost itd.

Do samog tumačenja saznanja može se doći apsolviranjem sledeća tri koraka ili sledeće tri aktivnosti:

⁵ Izvor: [6.], strana:

⁶ Izvor: [22.], strana: 13.

⁷ Izvor: [22.], strana: 13.

⁸ J Frates u [22.], strana: 13.

⁹ Izvor: [20.], strana: 9.

1. čulno primanje ili percepcija saznanja,
2. umno primanje ili identifikacija sadržaja saznanja i
3. shvatanje – razumevanje.

U slučaju da se desi makar i jedna “greška” u prethodna tri koraka ili izostane bilo koji korak, do aktivnosti tumačenja saznanja ne može doći. Percepcija ukazuje na ulogu naših čulnih organa u procesu pretvaranja podataka u informacije.

Za hendikepirane ljude, pretežno na polju sluha i vida su, uslovi stvaranja ili nastajanja informacija otežani.

Identifikacija sadržaja saznanja se teže objašnjava od percepcije, ali nam je verovatno poznata situacija iz svakodnevnog života kad u pauzi za doručak grickajući kreker ili sendvič, uz to čitajući novine priupitamo našeg sagovornika: “Šta si ono rekao?”. Percepcija saznanja se očito odigrala, međutim, identifikacija njegovog sadržaja nije. Naša pažnja je bila usmerena na vesti iz novina. U prvom trenutku kada shvatimo da nam se sagovornik obratio, otpočinje “rekonstrukcija” percepcije, da bi, nakon neuspele rekonstrukcije sledilo gore navedeno pitanje za ponavljanje sadržaja izrečene misli. Drugi karakterističan primer je obračun struje: nikad u životu nisam imao u ruci obračunski list za potrošenu električnu energiju sa kojeg sam mogao saznati koliko za potrošača košta jedan kilovat-čas električne energije! Taj podatak nisam mogao izdejsstvovati ni kada sam bio šef sektora Informatike u Elektrodistribuciji “SUBOTICA” u Subotici! Cena jedinice električne energije se i tada, a i sad može dobiti agregiranjem (izračunavanjem) podataka sa obračunskog lista, ali to ni ja nisam nikada znao kako! Primetimo da je fizička organizacija i prikazivanje saznanja bitno utiču na identifikaciju saznanja: suštinska saznanja treba prikazati na upadljivim mestima i na način koji se odmah, na prvi pogled primeti.

Shvatanje ili razumevanje saznanja predstavlja treći, zadnji korak pre njegovog protumačenja. Saznanje se najčešće formuliše u vidu vesti ili iskaza. Iskaz se može shvatiti samo u tom slučaju ako primaoc vesti razume svaku njenu reč, tj. kad generator iskaza i primaoc “govore istim jezikom”.

Protumačenje saznanja je u stvari povezivanje novodobijenog podatka ili iskaza sa ranijim saznanjima. Ako ne postoje ranija saznanja sa kojima bi se mogao povezati podatak, informacija naravno neće nastati. Neće se roditi informacija ni tada, ako se već povezivanje jednom dogodilo. Informacija nastaje u trenucima kada razmišljamo o samom sadržaju podatka, kad donosimo neki sud u vezi tog iskaza, kada pronadjemo mesto tog podatka u nekoj ili nekim klasifikacijama, kada formiramo neku misao, kada formulišemo neki zaključak.

Upravo iz prethodnih razloga na računaru uvek smeštamo “samo” podatke, koje posle, po potrebi i obradimo. Informacija se radja samo u glavama ljudi, a obrada podataka na računaru je

samo onda odgovarajuća, ako obezbeđuje takve nizove podataka da ljudima olakšava i ubrzava proces njihovog protumačenja.

Saznanja o realnom svetu vezana za njegove pojave i procese kroz iskustva i učenje omogućuju formulisanje zaključaka, analizu uzročno-posledičnih veza, apstrahiranje – izvlačenje bitnih crta – atributa, kao i sposobnost sticanja novih saznanja i označavaju sistem na višem nivou apstarkcije koji se naziva znanjem. **Znanje je, prema tome, pojamovna slika realnosti u svesti čoveka o stvarima, objektima, činjenicama, pojavama i procesima zajedno sa svim njihovim međusobnim i uzročno-posledičnim vezama.**

4.2.2. Informacione aktivnosti i informacioni događaji

Aktivnosti i događaji su usko povezani pojmovi i zato se najčešće zajedno analiziraju i objašnjavaju u okviru predstavljanja informacionih sistema. Aktivnosti vezane za informacije u nekom društvu mogu biti izrazito različite, no, pošto nas prvenstveno informacioni sistemi interesuju, zadovoljićemo se užom, specifičnijom definicijom:

Informaciona aktivnost je organizovana celina operacija za rukovanje i stvaranje podataka uz neophodno prisustvo upravljačkih operacija koje usaglašavaju prethodne dve vrste operacija i odredjuju njihov konkretan redosled izvršavanja.

Operacije za rukovanje podataka ne stvaraju novo saznanje. Tu spadaju operacije unosa, modifikacije, brisanja, kopiranja, spašavanja, sortiranja, ispisa, itd. podataka. Unos i modifikacija podataka nas možda kao izrazi ubedjuju kao da je reč o novim saznanjima, međjutim, to nije tako, jer se unositi može samo ono što već postoji. Zato je možda pogodnije da ta dva pojma izrazimo rečima obuhvat (umesto unosa) podataka i sprovođenje izmene (umesto modifikacija) podataka.

Operacije za stvaranje podataka obezbeđuju nova saznanja. Primer za ovu vrstu operacije je recimo izračunavanje vrednosti stavke računa prostim množenjem količine stavke računa sa jediničnom cenom proizvoda.

Upravljačke operacije povezuju operacije za rukovanje i stvaranje podataka u konkretnu strukturu za pojedine informacione aktivnosti i upravo zahvaljujući njima možemo tvrditi da aktivnost predstavlja organizovanu celinu.

Informacioni događaj je vremenski trenutak, odnosno momenat koji pokreće ili završava neku informacionu aktivnost.

Informacione događaje formuliše sam korisnik navodjenjem svojih zahteva prema informacionom sistemu. Zahtev korisnika pokreće informacioni događaj, a on aktivira izvršavanje odgovarajuće informacione aktivnosti. Pošto se aktivnost može uspešno ili neuspešno završiti mogu postojati više informacionih događaja koji završavaju neku informacionu aktivnost. Isto tako, informaciona aktivnost može da se pokreće od strane više događaja. Na osnovu toga se može zaključiti da veze izmedju informacionih aktivnosti i događaja predstavljaju beskonačnu mrežu.

Događaj je povezan sa “relevantnim promenama” sistema, gde pod pojmom relevantne promene podrazumevamo takve promene koje su bitne za korisnika. Relevantne promene u sistemu mogu biti sledeće:

1. životni ciklus pojava ili entiteta u sistemu obuhvata sledeće promene: radjanje, promene i nestajanje (a promene tipa izmena podataka postojećeg kupca, radjanje ili pojava novog kupca, ili kad postojeći kupac izgubi status kupca rezultuju u informacionim događajima)
2. vrlo je važan faktor vremena koji takodje može izazvati informacione događaje¹⁰: porez na promet se obračunava svakih D (petnaest) dana na primer.
3. promena stanja isto može biti generator događaja: ako stanje na zalihi robe padne ispod neke minimalne vrednosti, to može biti znak za pokretanje informacionog događaja.

Skup sličnih aktivnosti čine funkciju, a srodne funkcije sačinjavaju podsisteme od kojih se sastoji sam sistem.

Bitno je ukazati na još jednu činjenicu: dobro isprojektovani informacioni sistem, bez obzira što to danas jako neverovatno zvuči, može i ručno egzistirati, funkcionisati odnosno izvršavati. Teži se naravno da stepen automatizacije u obradi celokupne mreže informacionih aktivnosti bude što je moguće veći. Nije sporedna stvar, naime gde se automatska obrada pojavljuje, a gde napušta lanac, tačnije mrežu aktivnosti. Od toga zavisi udeo automatske obrade podataka u celokupnoj obradi. Od toga zavisi u kolikoj meri računarska obrada olakšava rukovanje podacima.

¹⁰ Tempora mutantur et nos mutamus in illis. – Vremena se menjaju i svi se mi menjamo zajedno sa vremenom.

4.2.3. Resursi potrebni za informacione sisteme

Rasprostranjeno je mišljenje da između pojmova sistem i resursi potrebni za njegovu izgradnju i funkcionisanje postoji suštinski ista veza kao između cilja i orudja koja omogućuju postizanje tog cilja. Već je odavno poznata činjenica da cilj blagoslovljava orudja (sredstvo), tj. da cilj predstavlja primarni a orudja sekundarni pojam. Danas je često sredstvo važnije od cilja što predstavlja veliki problem, jer se time razvoj sistema zapostavlja. U širem smislu podatak, informacioni događaj, informaciona aktivnost, korisnik, pa čak i standardi predstavljaju ne samo elemente, već i resurse informacionih sistema. **Resursi** potrebni za informacione sisteme u užem smislu **obuhvataju samo vreme, novac i tehnička sredstva (hardver i softver)**. Važno je istaći da elementi resursa ne mogu se na proizvoljan način konvertovati jedan u drugi.

Uzmimo prvo novac, pošto se on u zadnje vreme sve više fetišizuje. Ne treba preceniti značaj ovog elementa. Istina je, da se bez novca ne može obezbediti ni jedan činilac informacionog sistema, a kamo li projektovanje, realizacija i funkcionisanje. Dupliranje novca, međutim, neće znatno skratiti vreme potrebno za projektovanje, jer otkrivanje, shvatanje i analiza realnog sveta i njegov opis (izrada projekta informacionog sistema) traži vremena i naravno umešnost od ljudi, koji nemaju “TURBO” dugme za brže shvatanje realne stvarnosti.

Često se slabi projekti informacionih sistema objašnjavaju nedostatkom odgovarajućih sredstava. To bi odgovaralo mom izgovoru kad bih izjavio da je ovaj rad zato tako slab i površan, pošto mi je računar bezbroj puta neočekivano “padao” za vreme pisanja, najverovatnije zbog nekompatibilnih čipova operativne memorije. Ili, kao kontrapunk prethodne izjave: mogao sam i sto puta kvalitetniji rad sastaviti da sam imao na raspolaganju besprekornu mašinu.

Resursi su element(i) informacionih sistema, a informacioni sistem je organizovani skup činilaca, što ukazuje na potrebu sagledavanja svih elemenata informacionih sistema i usaglašavanje potrebnih sredstava sa svih ostalim činilcima.

4.2.4. Korisnici informacionih sistema

Svaki činilac naveden u definiciji informacionog sistema u realnosti predstavlja složen sistem za sebe. Podaci se obrađuju zahvaljujući informacionim aktivnostima koje se pokreću na uticaj odgovarajućih zadataka, što pokazuje na organizovanost elemenata informacionih sistema (njegovi činilci su međusobno povezani). Primetimo, međutim, i to da su do sada prikazani

elementi uvek obuhvatili veći skup objekata, na šta pokazuje i pluralia tantum tj. množina u nabrojanju u okviru definicije informacionog sistema (podaci, događaji, aktivnosti). Podaci su znači povezani sa aktivnostima, ali i između samih podataka postoje medjuveze, kao i između aktivnosti, a na isti način i između događaja.

Korisnici informacionih sistema takodje predstavljaju svojstveni sistem i zato će definicija korisnika obuhvatiti sve ljude i organizacije koje, na bilo koji način, dolaze u kontakt sa informacionim sistemom:

Korisnici su ti ljudi (ili grupe ljudi – organizacione jedinice) koji su na neki način povezani sa saznanjima (podacima, informacijama), odnosno informacionim sistemom.

Pošto se ljudi mogu pojaviti u različitim ulogama u vezi nekog informacionog sistema ili sa tačke gledišta korišćenja saznanja, informacija, podataka obradjenih pomoću informacionih sistema, moramo zaključiti da pojam korisnika ne možemo prihvatiti za apsolutnog činioca.

Krajnji korisnici su oni ljudi zbog kojih je sistem realizovan, ili kojima je informacioni sistem namenjen. U osiguravajućem zavodu su shodno osnovnoj aktivnosti upravo osiguranici krajnji korisnici. Godišnji finansijski izveštaj u osiguravajućem zavodu se, međjutim, priprema za akcionare, i za taj izveštaj su oni krajnji korisnici (korisnik je relativan pojam).

Aplikativni korisnici su oni ljudi koji su u neposrednom kontaktu sa saznanjima u informacionom sistemu, a eventualno i sa krajnjim korisnicima. Radnik osiguravajućeg zavoda koji sklapa ugovore o osiguranju se može smatrati aplikativnim korisnikom, bez obzira na to da možda nikada nije i neće raditi na računaru (sa tačke gledišta platne liste on je naravno krajnji korisnik).

Ljudi na razvoju informacionih sistema su takodje korisnici. Oni sreću iste informacije kao prethodno opisane dve grupe korisnika, samo sa drugog aspekta. Oni moraju u nekim trenucima “odigrati” ulogu krajnjih i aplikativnih korisnika u cilju da bi realizovali jedan stvarno dobar informacioni sistem.

Rukovodilac je specifičan korisnik: upravljanje nad sredstvima je njemu provereno, svojim autoritetom može povoljno uticati na kvalitet saradnje radnika organizacije sa članovima razvojnog tima, a može i da skicira viziju razvoja organizacije u cilju ostvarenja što boljeg sistema.

Organizacione jedinice se takodje mogu smatrati korisnikom u širem smislu reči. Ako je neki konkretan radnik osiguravajućeg zavoda koji sklapa ugovore o osiguranju odsutan, zamenjuje ga drugi sa istom kvalifikacijom, znanjem i iskustvom iz iste organizacione jedinice, jer je ona zadužena za obavljanje poslova u vezi sklapanja ugovora o osiguranju.

4.2.5. Standardi i procedure

Pojam standarda neki doživljavaju u negativnom, a neki u pozitivnom smislu. Oni koji veruju da standardi mogu jedino ograničavati slobodu stvaranja detaljnim opisima svih mogućih zabrana, verovatno nisu previše ushićeni kad čuju reč standard. Drugi, pak, koji pod pojmom standarda podrazumevaju skup korisnih vodilja, opšte prihvaćenih principa i zakonom propisanih regulativa, standard smatraju za pogodno pomoćno sredstvo pri radu. Standard, svakako, može biti i jedno i drugo, pa čak jedno i drugo istovremeno, jer pod standardom treba podrazumevati posebne konvencije vezane za svaki pojedinačni činilac informacionog sistema. Te konvencije su i međusobno povezane (sve je povezano sa svačim i u životu), odnosno predstavljaju neku strukturu, a to drugim rečima znači da standardi predstavljaju specifičan sistem (standarda) unutar informacionog sistema.

Definicija standarda se može formulisati na sledeći način:

Standard je dogovor, konvencija ili propis koji se odnosi na neki činilac (element) informacionog sistema.

Odomaćene su neke konvencije vezane za neke podatke kao i za neke “procesе” u obradi podataka. Uzmimo naše osiguravajuće društvo i razmislimo malo o njegovom (ali i o bilo čijem) knjigovodstvu. Zakonom je propisano koje se podaci moraju obuhvatiti knjigovodstvenom evidencijom i izveštajima, na koji način da se prezentuju ti podaci, kojim procedurama (kako) se dolazi do traženih podataka i kada te podatke treba prezentovati (rokovi za predaju periodičnih izveštaja). Upravo se iz tog razloga na Zapadu obično govori o standardima i procedurama (ta dva pojma zajedno – povezano koriste u praksi).

Od resursa potrebnih za informacione sisteme je na prvi pogled najlakše odrediti standarde za orudja (tehnička sredstva) ili sredstva za obradu podataka: za hardver i softver. U principu su tehnička sredstva standardizovana, ali u stvarnosti nisu. Verovatno se ne bi pojavila tako često na ekranu moga računara poruka “General Failure ...”, da su u njega bili ugrađeni (svojevremeno i kod učestalih proširivanja – taj računar imam već pet-šest godina) standardni delovi. Za njega su samo tvrdili da je PC kompatibilan... Ni sa softverom nije ništa bolja situacija: moji crteži koji su izrađeni pomoću Corel 2.0 ni pomoću trikova se ne mogu “uvući” u Corel 7.0. Šta više, po pravilu novije verzije softvera jedva liče na starije verzije koje zamenjuju. Pošto se proizvođači hardvera i softvera ne mogu međusobno sinhronizovati što se standarda tiče, korisnici bi rešenje trebali tražiti u izgradnji sopstvenog unutrašnjeg standarda za tehnička sredstva. Ostala dva elementa informatičkih sredstava, vreme i novac se još teže standardizuju. Oni su i međusobno povezani, a i pojedinačno se teško procenjuju: potrebno vreme i vrednost izvršenog posla.

Vrednost izvršenog razvoja informacionog sistema ne može se proceniti samo na osnovu broja linija izvornog koda (kao što su to nekad na Zapadu radili), jer razvoj nije samo programiranje, a sa druge strane vrednost razvoja bi trebali da odrede i važnost saznanja, kao i upotrebna vrednost pruženih informacija (što opet zavisi i od izvora primarnih-prvobitnih informacija, a ne samo od razvijenog informacionog sistema).

Korisnički standardi takodje treba da postoje. Osnovu korisničkih standarda čine “zajednički jezik” ili zajednički skup pojmova. Standardom treba odrediti “sadržaj” korišćenih pojmova. Za krajnjeg i aplikativnog korisnika standardi jednoznačno određuju šta, kada, na koji način realizuje informacioni sistem i/ili korisnički interfejs.

Na osnovu prethodno izloženih vidjenja o ulozi standarda, moguće je zaključiti, da standardi imaju trostruku ulogu u toku razvoja i korišćenja informacionih sistema:

1. upućuju korisnike (naročito u toku razvoja informacionog sistema, i to analitičare, projektante i programere) šta kako treba raditi,
2. ugrađuju se u informacione sisteme kao ograničenja (standardizovani podaci se prihvataju samo onda, ako su propisanih oblika, sadržaja, itd.) i
3. obaveštavaju korisnike (šta, gde i u kojem obliku treba tražiti).

Poznate i priznate naučne discipline ili nauke imaju davno izradjene i stabilne standarde, odnosno sistem standarda. U informatici se tek nedavno počelo ozbiljnije razmišljati o potrebi izrade sistema standarda, možda je to razlog, da za sada nju neki ne priznaju za nauku.

4.3.PROJEKCIJE INFORMACIONOG SISTEMA

Informacioni sistem je nehomogeni sistem konkretnih i apstraktnih elemenata. U konkretne činioce spadaju fizički egzistirajuće stvari kao što su tehnička sredstva, standardi i korisnici, a u apstraktne elemente ubrajamo intelektualne proizvode ljudskog razmišljanja koji služe za verno odslikavanje fizičke stvarnosti, a to su događaj, aktivnost i podatak. Konkretni elementi postoje, a apstraktne elemente treba smisliti, projektovati, formulisati i izraditi, a takodje i otkriti i opisati njihove medjusobne veze. Na osnovu prethodno izloženog se već naziru dve projekcije informacionog sistema, medjutim, pošto su podatak sa jedne strane i događaj i aktivnost sa druge strane jako različite prirode, razlikuju se sledeće tri projekcije informacionog sistema:

1. projekcija podataka,
2. projekcija obrade i
3. projekcija okruženja.

Treba konstatovati da su saznanja i pojmovi u nekoj konkretnoj organizaciji poprilično stabilni. Česte korekcije i dogradnje su pretežno posledica nestručno izvedenih razvoja nego nestabilnosti podataka (pojmovi). Projekcija podataka je relativno objektivna. Podatak je u velikoj meri nezavisan od ostalih činilaca informacionog sistema. Važno je istaći da je podatak u velikom stepenu nezavisan od obrade, sa kojom je možda u najtesnijoj vezi.

Projekcija obrade je u odnosu na projekciju podataka relativno nestabilna i subjektivna i, uopšteno govoreći nema takav stepen nezavisnosti od ostalih elemenata informacionog sistema kao što ima projekcija podataka. Projekcija obrade je relativno nestabilna iz razloga što korisnik može izmišljati razne vrste izveštaja i sopstvene događaje bez izmene projekcije podataka. Obrada, dakle, u odnosu na podatak može biti višestruka. Promenljivost u obradi, međutim, ne uslovljava i promenu strukture podataka, jer **logička nezavisnost podataka kao princip propisuje da promene u obradi podataka ne smeju uticati na strukturu podataka i obrnuto.**

Projekcija okruženja obuhvata korisnike, resurse za realizaciju informacionog sistema (vreme, novac, hardver i softver) i standarde, tj. realno postojeće stvari, a ne abstrakcije. Subjektivne želje korisnika, objektivne mogućnosti hardvera i softvera kao i kvazi objektivni standardi sprečavaju analitičara da realizuje po njemu najbolji mogući informacioni sistem. Promena okruženja u vremenu je neminovna pojava, ali to ne (bi) sme(lo) da izazove promenu strukture podataka, jer **princip fizičke nezavisnosti podataka formuliše stav da promene u sredstvima za obradu ne smeju uticati na strukturu podataka i obrnuto.**

4.4. NIVOI INFORMACIONOG SISTEMA

Projekciju podataka, obrade i okruženja možemo shvatiti kao horizontalnu podelu činilaca informacionog sistema. Elementi ovih projekcija nisu nezavisni jer predstavljaju činioce (informacionog) sistema. Kako postoje povezanosti između dva elementa iz različite projekcije tako postoje i međusobne veze između činilaca iz iste projekcije. Elementi projekcije podataka i obrade nastaju kao rezultat intelektualnog napora na raznim nivoima abstrakcije. Projekcija okruženja, pošto označava fizički postojeće resurse, poseduje samo jedan abstrakcioni nivo: fizički. Ostale dve projekcije pored fizičkog, imaju još dva abstrakciona nivoa, tako da se informacioni sistem posmatra i realizuje na sledeća tri abstrakciona nivoa:

1. pojmovnom (konceptualnom),
2. logičkom i

3. fizičkom nivou.

Konceptualni nivo obuhvata **sadržaj** ili **suštinu** dela realnog sistema koji se želi opisati pomoću informacionog sistema, a sâmo **rešenje** se predstavlja na logičkom i fizičkom nivou. Pomenuta tri nivoa informacionog sistema se mogu i moraju (ili morali bi) razlikovati pri razvoju samog sistema. Neispoštovanje ova tri abstrakciona nivoa može dovesti do sledeća dva tipa problema u kojima se mešaju suština i rešenje. Pri razvoju se najčešće predstavlja slika stvarnosti koja odgovara korisniku-naručiocu, a ta slika je po pravilu izvitoperena u odnosu na realni sistem. Sa druge strane razvojni tim pri izradi projekta je najčešće opterećen sa ograničenim mogućnostima sistema za rukovanje i obradu podataka, znači i rešenje će odražavati ta ograničenja, a ne stvarnu sliku dela realnog sveta. Gore opisani problemi su objektivni i naravno neizbežni. Naručioc je zainteresovan da njegovi zahtevi budu ispoštovani pri razvoju, a razvojni tim ne može zaobići ograničenja hardvera i softvera računara. Sve je to u savršenom redu, jedina je manjkavost ta, da projektanti pripremaju samo jedan plan u kojem su i babe i žabe, tj. i suština i rešenje. A zašto je to loše? Suština i koncepcija nekog poduhvata ili posla se znatno sporije menjaju nego računari ili softveri za obradu podataka. U slučaju da postiji samo jedan projekat u koji su ugrađena i ograničenja po zahtevu korisnika i ograničenja hardvera i softvera, kupovina novih računara ili prelazak na novi softver će iziskovati ponavljanje celog razvojnog posla iz početka.

Ispoštovanje tri nivoa informacionog sistema znači sastavljanje tri plana (na svakom nivou po jedan). Na konceptualnom nivou se predstavlja verna slika realnog sistema, na logičkom slika o stvarnosti koja odgovara korisniku, a na fizičkom slika koja može da se realizuje na računaru, a odgovara korisnikovom vidjenju stvarnosti. Niži nivoi informacionog sistema sve više izobličavaju pravu sliku stvarnosti, što je sasvim logično i razumljivo.

U **razvoju projekcije podataka** prvi je zadatak analitičara da upozna pojmove koje korisnik upotrebljava i da otkrije njihovu semantiku. Pojmovi u realnom svetu nisu nezavisni, shodno tome drugi zadatak analitičara je da ih organizuje, da sačini njihovu strukturu (da dokumentuje njihove međusobne zavisnosti), tj. da izgradi pojamovnu strukturu podataka. **Pojmovna struktura podataka** (pojamovni nivo strukture podataka) je beskompromisna, verna i objektivna slika stvarnosti. Ili, drugim rečima: **konceptualni plan podataka** je struktura podataka koja realni sistem odslikava bez ijedne greške.

Na zahtev korisnika – naručioca projektant svesno izobličava, »pokvari« pravu sliku stvarnosti. Uticaj ili zahteve korisnika možemo smatrati subjektivnim ili kvazi-subjektivnim ograničenjima kojima valja izaći u susret. Ponekad analitičar i nesvesno izobličava sliku realnog sistema. To se događa u slučajevima kada on ne obuhvata sve elemente ili činioce informacionih

sistema, ili kada ne može doći do prave objektivne slike realnog sistema, a o tome on sam ne zna. **Logička struktura podataka** sadrži izmenjenu pojmovnu strukturu podataka shodno definisanim ograničenjima okoline (uglavnom korisnika), ili zbog pogrešne vizije projektanta o realnom sistemu. **Logički plan podataka** je, prema tome, struktura podataka koja odražava izmenjenu sliku stvarnosti usled subjektivnih zahteva naručioca ili pogrešne percepcije (pogrešnog shvatanja) realnog sistema od strane projektanta.

U toku razvoja informacionog sistema analitičari moraju biti svesni svih ograničenja sredstava za obradu podataka (hardvera) i softvera za rukovanje i obradu podataka. Razvojni tim mora biti spreman na kompromise u vezi smeštanja i predstavljanja podataka na računaru. **Fizička struktura podataka** obuhvata način predstavljanja i smeštaja podataka na memorijskom medijumu računara i često se naziva **fizički plan podataka**.

Planove podataka na sva tri nivoa abstrakcije obavezno treba izraditi, i to u jedino mogućem i logičkom redosledu: prvo se realizuje konceptualni plan, zatim logički plan, a na kraju fizički plan podataka.

U okviru razvoja projekcije obrade na početku projektant analizira događaje i sa njima neposredno povezane aktivnosti u realnom sistemu. Događaje u realnom sistemu prate informacioni događaji koji izazivaju ili pokreću informacione aktivnosti. Opisani informacioni događaji zajedno sa pratećim informacionim aktivnostima predstavljaju samu formulaciju zadatka (šta treba uraditi) i smatraju se suštinom **pojmovnog nivoa obrade podataka**. **Konceptualni plan obrade** sadrži same opise zadataka (informacione događaje i aktivnosti) koje treba izvršiti u okviru obrade podataka informacionog sistema.

Jednoznačno formulisani zadatak se može rešiti na više načina. Principijelno rešenje same obrade (odnosno kako treba rešiti zadatak), tj. organizacija operacija u prolaze ili procedure predstavlja **plan obrade na logičkom nivou**. **Logički plan obrade** se razradjuje od strane projektanata imajući u vidu kvazi-objektivne, kvazi-subjektivne i subjektivne zahteve korisnika, ograničenja softvera za rukovanje i obradu podataka, a na sve to utiču, naravno, i navike samih projektanata. Jedna formulacija zadataka obrade, tj. jedan konceptualni plan obrade omogućuje realizaciju velikog broja alternativa, odnosno principijelnih rešenja.

Plan obrade na fizičkom nivou sadrži konkretno rešenje samih zadataka obrade koje je zavisno od sredstava (hardvera) na kojima se obrada odnosno zadatak realizuje. Tu spadaju organizacija maski za unos podataka i izveštaja, definisanje modula programa i fizičkih izvršnih celina, itd.

Nužno je izraditi planove obrade podataka sva tri nivoa abstrakcije s tim da je redosled izrade planova obrade, kao i kod projekcije podataka, striktno odredjen. Za analitičara sistema

predstavlja obavezu da prvo formuliše same zadatke u okviru obrade (pojmovni nivo), zatim u sledećem koraku da izradi principijelno rešenje (logički nivo) shodno zahtevima korisnika i ograničenjima sistema za rukovanje podataka, a na kraju da konkretno definiše rešenje zadataka, odnosno obradu (fizički nivo).

U projektnim dokumentacijama se najčešće ne razlikuju i ne razdvajaju ta tri abstraktna nivoa projekcije podataka i obrade. Čak i najpoznatije metodologije predlažu izradu samo dva plana: logičkog i fizičkog. Pojam konceptualnog nivoa je poznat od 1975. godine, kada je objavljena ANSI/SPARC arhitektura baze podataka, međutim, u najvećem broju slučajeva se pogrešno poistovećuje sa logičkim nivoom. Gore su opisane sadržaji planova podataka i obrade na sva tri nivoa abstrakcije. Plan na nižem nivou abstrakcije se dobija preslikavanjem plana na višem nivou. Dobra projektna dokumentacija informacionog sistema sadrži pored svih nabrojanih planova i opise ograničenja, razmišljanja o alternativama i razloge o izabranoj verziji za svako konkretno preslikavanje. Drugim rečima u dokumentaciji projektant treba da opiše kako se odlučio za izabrano preslikavanje između planova različitog nivoa, tj. koje su bile ograničavajuće okolnosti koje su bile alternative usled tih ograničenja i, najzad zašto se odlučio za izabrano preslikavanje.

Mobilnost u poslovanju preduzeća se može unaprediti jedino smanjivanjem vremena razvoja informacionog sistema i njegovog prilagođavanja novonastalim uslovima. U sredinama sa savremenim shvatanjem informatike i prilagođavanje promenama smatraju razvojem sistema. To znači da i promenama pri modifikaciji samog sistema poklanjaju istaknutu pažnju. Prilagođavanje sistema se u takvim firmama realizuje u okviru change management-a.

Razvoj sistema je fleksibilniji i brži ako se projekat informacionog sistema ostvaruje na tri abstrakciona nivoa: Planovi na pojedinim nivoima se brže izrađuju, kontrolišu i verifikuju, jer se u njima ne mešaju elementi realnog sistema sa zahtevima korisnika i ograničenja sredstava za obradu podataka. Promene se pri ovom pristupu mogu ekspeditivnije i efikasnije sprovesti na veliko zadovoljstvo korisnika – naručioca. U slučaju da se menja samo hardverska platforma ili softver za rukovanje i obradu podataka, planovi na pojmovnom i logičkom nivou ne treba da pretrpe nikakve promene: treba menjati samo fizičke planove na osnovu planova na logičkom nivou, prilagođavajući ih novom hardveru i/ili softveru. To znači da će se menjati i deo dokumentacije o preslikavanjima, alternativama, izboru varijanti i razlozima koji su doveli do tog konkretnog izbora. Ako je reč o promenama zahteva korisnika, onda su nužne izmene na logičkom i fizičkom nivou, i samo konceptualni nivo ostaje netaknut. Pri promenama zahteva korisnika menjaju se i pojmovno – logičko i logičko – fizičko preslikavanje.

U slučaju klasičnog projektovanja informacionog sistema, kada se izrađuje samo jedan plan, svaka promena izaziva kompletno reprojektovanje informacionog sistema ispočetka, od upoznavanja realnog sistema. Razlog za to je razumljiv: taj jedan jedini postojeći plan sadrži opis realnog sistema izmešan sa svim željama i vizijama naručioca i svim ograničenjima hardvera i softvera. Po pravilu u planu se i ne pojavljuju eksplicitno ograničenja, nego samo rešenje, a o uzrocima zašto je rešenje baš takvo, ne može se naći ni jedna reč. U klasičnom projektu (dokumentaciji) nema opisa objektivne, neizobličene stvarnosti. To je razlog da se pri bilo kakvoj izmeni deo stvarnosti koji je obuhvaćen informacionim sistemom mora ponovo upoznati. A to je možda i najkompleksniji poduhvat pri razvoju informacionog sistema, koji najviše traži i vremena. Tako tronivovsko planiranje može poboljšati efikasnost pri razvoju i reinžinjeringu informacionog sistema, a samim tim omogućiti veću mobilnost u poslovanju preduzeća.

5. TEHNIČKA SREDSTVA – HARDVER I SOFTVER

Osnovna pokretačka snaga razvoja uopšte je profit, odnosno novac¹ i slobodno tržište, tj. tržišno poslovanje. U cilju ostvarivanja profita organizacije su prinudjene da daju kvalitet više od ostalih, i da to na odgovarajući način i reklamiraju. Bezbroj je takvih primera iz prakse kada je bolji proizvod zapostavljen iz razloga što nije imao odgovarajuću podršku marketinga. Današnje savremene kompanije svoju uspešnost u poslovanju mogu zahvaliti naučno zasnovanim “instinktima”, svojim futurolozima, odnosno predviđanju budućnosti. Nije dovoljno proizvoditi jednostavno kurentne i kvalitetne proizvode, nego treba predvideti ukus i zahteve potrošača, šta više, upravljati njihovim ukusom i zahtevima. Pogoditi njihove tajne i latentne želje, ponuditi im i ono, na šta oni još nisu ni mislili. Prepoznatljivi je trend u zadnje doba da se proizvodi “obogaćuju” izvesnim uslugama u cilju povećanja komocije potrošača i za ostvarivanje većeg profita proizvođača.

Opstanak i razvoj kompanija zavisi od njihove informisanosti, tj umešnosti u neprekidnom izvlačenju informacija iz unutrašnjeg i spoljašnjeg okruženja, i od brzine reagovanja i prilagođavanja novonastalim uslovima. U cilju što bolje informisanosti organizacije zahtevaju sve veće količine informacija u što kraćem vremenskom intervalu i inteligentne softvere koji će na osnovu raspoloživih informacija što brže savetovati rukovodstvo u donošenju presudnih odluka. Za realizaciju mobilnosti uopšteno, znači, traže se sve snažnije mašine sa sve inteligentnijim softverskim rešenjima.

Tendencija razvoja hardvera i softvera odgovara upravo ovim, gore formulisanim zahtevima. Treba, medjutim ukazati na dve dobro razgovetne linije razvoja kako hardvera tako i softvera. Što se hardvera tiče, sa jedne strane konstruišu se snažni računari otporni na greške (fault-tolerant), kod kojih je bitan činilac jedino throughput, tj. brzina obrade, a sa druge strane, teži se minimizaciji komponenata i potrošnje, i pojednostavljenju u održavanju i korišćenju računara. Prva linija razvoja hardvera obuhvata statičke (stone) računare koje u poslovnoj sferi povezuju u lokalne mreže (preko kojih je moguć pristup regionalnim i globalnoj svetskoj mreži – Internetu), razne servere u mrežama raznih nivoa, front-end (komunikacione) i back-end (database engine) računare, kao i razne superkompjutere u cilju njihovih korišćenja u istraživanjima i kompleksnijem matematičkom modeliranju. Druga linija razvoja podržava prenosiva, mobilna sredstva kao što su laptopi, notebook-ci, džepni i ručni PC-i (lične sekretarice) i mobilni telefoni. U ovu grupu spadaju i PC-ji predviđeni za korišćenje Internet usluga: NC (NetworkComputer) i NetPC, koji, istina nisu minijaturene izvedbe kao što su

mobilna sredstva pominjana u prethodnoj rečenici, ali su pojednostavljene izgradnje sa ograničenim mogućnostima (nemaju hard disk, floppy disk, CD-ROM, DVD-ROM).

Softver se razvija takodje u dva generalna pravca, bez obzira da li je reč o operativnim sistemima, uslužnim programima ili aplikativnim programima. Za prvu liniju razvoja hardvera koja obuhvata standardne ili "nemobilne" računare projektovani i proizvedeni softver se neprekidno usložnjava. Treba primetiti, međjutim, da u okviru iste težnje postoji jedna mala ali razumljiva kontradikcija: usložnjavanje softvera ne znači dodatno opterećenje i za korisnike tih softvera. Korisnički interfejs tih kompleksnih softvera se uprošćava, ili tačnije stepen složenosti korišćenja ne prati dinamiku usložnjavanja samih softvera. Drugi trend u razvoju softvera prati mobilna sredstva koja oskudevaju u svim resursima zbog samih karaktera tih uređaja: u procesoru, operativnoj memoriji, eksternim memorijama, ulaznim, izlaznim jedinicama, itd. Druga linija razvoja softvera diktira uprošćavanje svih softverskih produkata na mobilnim sredstvima, zadržavajući samo njihove najbitnije funkcije, smanjujući time njihove zahteve za računarskim resursima. Postoji još jedna svojstvena crta za računarski softver koja ga razlikuje od hardvera: postoje besplatni softveri sa punom funkcionalnošću (završeni, kompletni funkcionalni, istestirani), a takav pojam na nivou harvera naravno ne postoji.

Medjusobni uticaj hardvera i softvera je poznata činjenica. Oni se medjusobno eksituju na razvojnem putu, odnosno nova dostignuća jednog od aktera pokreće inicijativu za razvoj druge oblasti. Nove mogućnosti hardvera, kao što su brži procesori, brže operativne i eksterne memorije i veće brzine prenosa u računarskim, ili uopšte telekomunikacionim mrežama omogućavaju izvodjenje kompleksnijih softvera, a novi nivoi softvera traže snažniji hadver za istu brzinu izvodjenja kao što su to obavile starije verzije softvera. U zadnje vreme je svim akterima u informacionoj tehnologiji jednoznačno da je softverska industrija u velikom zaostatku u odnosu na stepen razvoja hardvera i komunikacija. Softverska kriza se, dakle, i dalje nastavlja. Ta kriza, koja je nastala grubo pre trideset godina bila je dvaput smanjena, ublažena odnosno ukroćena: krajem sedamdesetih godina uvođenjem strukturiranog prgramiranja i projektovanja, a krajem osamdesetih godina osmišljavanjem odnosno obnavljanjem i razradom ideje iz kasnijih šezdesetih godina, objektnog programiranja i razvoja informacionih sistema. U današnje vreme se još ne nazire nova tehnologija/paradigma u softverskoj industriji koja bi mogla biti podstrek za razvoj nove generacije hardvera.

Komunikacija (posredna) preko sredstva veze se realizuje dvojako: pomoću hardvera i softvera. Značajna su istraživanja na polju iznalaženja odgovarajućeg medijuma za prenos

¹ Many makes the World's go around.

signala koja su urodila plodom tako da danas imamo više medijuma na raspolaganju za komunikacije. O njima će kasnije biti više reči.

5.1. HARDVER

Ako mobilnost posmatramo u širem smislu reči, tj. bržu reakciju na promene okoline, onda prva linija razvoja hardvera takodje spada u unapredjenje mobilizacije, a ako pojam mobilnosti shvatamo u užem smislu reči, tj. razvoj lakih prenosivih računara, samo razvoj hardvera druge vrste potpomaže mobilnost. Prihvatajući pojam mobilnosti u širem smislu reči ukazaćemo na savremene tendencije u razvoju arhitekture računara uopšte (tj. ne samo PC-ja) i razvoju PC računara u cilju povećanja throughput-a ili smanjenja vremena obrade.

5.1.1. Arhitekture računara

Standardna konfiguracija PC računara je danas već poznata svim srednjoškolcima zahvaljujući nastavnom planu i programu. Razvoj performansi PC računara sledi tendencije razvoja velikih računara (mainframe) i superkompjutera od pre dvadeset – trideset godina. Logično je očekivati da će u cilju povećanja performansi PC računara proizvođači predložiti ideje koje su već poznate kod računara starije, ali različite arhitekture od PC-ja: masivna paralelna obrada (MPP – Massively Parallel Processors), simetrični multiprocesorski sistemi (SMP- Symmetric Multiprocessors), vektorske arhitekture i mainframe.

5.1.1.1. Masivna paralelna obrada – MPP

Ova arhitektura obuhvata kategoriju superkompjutera koja se gradi od više stotina ili više hiljada čvorova (računara), sa posebno projektovanom mrežom velike prenosne brzine za njihovo povezivanje. Ova filozofija gradnje se primenjuje i za znatno manji broj čvorova-računara bez specijalno projektovane mreže za povezivanje čvorova, i tada se obučno zove klasterka mreža (Cluster of Workstations – COW).

U čvorovima MPP arhitekture se obično nalaze RISC procesori sa sopstvenom memorijom i I/O uređajima, tako da je proširenje takve arhitekture relativno jednostavno i realizuje se prostim dodavanjem novih čvorova (mreža za povezivanje čvorova mora obezbediti mogućnost proširenja). Ova arhitektura pri obradi podataka optimizuje sam proces obrade zahvaljujući softverskoj biblioteci (koju obezbeđuje proizvođač) uključujući sve čvorove u obradu. Zbog velikog broja čvorova fizička povezanost između svih čvorova naravno ne postoji, ali se to može softverski rešiti u okviru rutina operativnog sistema koji upravlja radom cele mreže i cela ova problematika ostaje transparentna za programera i korisnika.

Lokalni memorijski moduli u pojedinim čvorovima se po pravilu ne obuhvataju u zajednički memorijski prostor. Uobičajeno je da se pristup nekog procesora memoriji nekog drugog procesora realizuje preko poruka, a ne preko naredbi za memorijske operacije. Ta realizacija je znači bez udaljenog pristupa memoriji (No Remote Memory Access – NORMA) i ima tu prednost da ne iziskuje hardversku podršku za očuvanje konzistencije sadržaja memorije.

Operativni sistemi u čvorovima su najčešće praktično samo jezgra operativnih sistema sa realizacijom osnovnih lokalnih zadataka, i u tom slučaju njihov rad koordinira operativni sistem centralnog čvora (hosta), koji brine o distribuciji samih zadataka u obradi. NORMA ima za posledicu da je pristup udaljenoj memoriji izrazito spor, zato obrada može biti efikasna samo u tom slučaju ako je strukturiranje zadatka odgovara strukturi MPP-a. U praksi se najčešće koriste takvi razvojni alati koji omogućuju prilagođavanje strukture zadatka strukturi same arhitekture, pre svega nova verzija Fortrana (HPF) koja ima naredbe i za označavanje delova koda za paralelno izvršavanje. Operativni sistemi u čvorovima i na hostu su po pravilu iz porodice Unix.

Više informacija o MPP arhitekturi se može pronaći na Internet adresama www.top500.org (podaci o najboljim superkomjuterima), www.computer.org/parascope/ (časopis IEEE-a za superkompjutere) i www.sandia.gov/ASCI/ASCI-links.htm (Američka laboratorija, najveći korisnik superkompjuteru).

5.1.1.2.Simetrični multiprocesorski sistemi – SMP

Simetrični multiprocesorski sistemi broje od dva do nekoliko desetina procesora, tj. predstavljaju manje sisteme od MPP. Procesori su tu potpuno isti i ravnopravni u korišćenju svih zajedničkih resursa: operativne memorije i I/O uređaja. Procesorima se upravlja pomoću istog operativnog sistema, a zbog korišćenja zajedničkih resursa neophodno je obezbediti sinhronizaciju u njihovom korišćenju. Za komunikaciju između procesora se koristi zajednički

memorijski prostor. Pored softverske sinhronizacije zajedničkog memorijskog prostora, neophodno je i hardverska sinhronizacija sadržaja “privatnih” cache memorija pojedinih procesora. Upravljačka jedinica cache memorije svakog procesora prati svaki transfer na memorijskom busu (bus snooping) i po potrebi (ako se neka bus operacija/transfer odnosi na dotični cache) menja status cache memorije.

Povećanje performanse obrade programa (smanjivanje vremena obrade) u SMP arhitekturi ograničava procenat adresnog prostora programa koji se mora striktno sekvencijalno obradljivati u ukupnom adresnom prostoru programa. Prethodni iskaz se često pominje kao Amdahlov zakon.

Ako se broj procesora poveća na nekoliko desetina u SMP sistemima, zajednički resursi, naročito bus sistem postaje usko grlo. U tim konfiguracijama se primenjuju četiri adresna busa za pristup memoriji. SMP arhitektura je osetljiva na hardverske i softverske greške, pošto su kritični resursi za obradu zajednički (shared) i ispad bilo kojeg od njih onemogućava rad velikog broja komponenata. Programiranje u SMP sistemima je mnogo jednostavnije nego u slučaju MPP arhitekture.

Često se primenjuju hibridna rešenja kombinovanjem MPP i SMP arhitekture, kao što je to slučaj kod SP porodice računara proizvođača IBM, gde su čvorovi MPP sistema u stvari SMP arhitekture. Ako struktura zadatka odgovara MPP strukturi, povećanje performanse SMP čvorova se odražava na ukupne performanse hibridnog sistema, međutim, problemi koji zahtevaju intenzivnu komunikaciju između čvorova su invarijantni na povećanje performanse čvorova, i njihova brzina obrade je zavisna od brzine komunikacione mreže između čvorova.

5.1.1.3. Vektorske arhitekture

Vektorske arhitekture računara se grade od sredine sedamdesetih godina i predstavljaju višeprocesorske sisteme. Kod ove vrste računara se pomoću specijalizovanog hardvera obezbeđuje paralelno izvršavanje elementarnih operacija nad velikom količinom podataka. Svaki procesor ima obično osam vektor-registara sa brojem od 64 do 256 elemenata za memorisanje vrednosti u pokretnom zarezu. Tipični problemi vektorskih arhitektura su operacije sa matricama i vektorima iz linearne algebre, odakle im i ime potiče. Pošto se kod algebarskih operacija javlja obično više vektorskih operacija u nizu za izvršavanje, pogodno je primeniti olančavanje vektor-registara. Tako elementi rezultatnog vektora iz prve vektorske operacije se neposredno šalju u vektorski pipeline druge vektorske operacije, ne čekajući pojavljivanje

celokupnog rezultata prve operacije. Za nesmetan i efikasan rad vektorskog pipelinea neophodno je izgraditi memoriju iz više memorijskih modula, da bi svaki element pipelinea mogao da se obrati drugom memorijskom modulu u cilju ubrzanja obrade. Kod vektorskih računara zbog velikog broja memorijskog modula ne primenjuje se cache memorija izmedju procesora i operativne memorije. Procesori vektorske arhitekture su CISC procesori i imaju dodatne jedinice specifične namene u cilju izvršavanja pomoćnih aktivnosti za ubrzanje izvršavanja vektorskih operacija.

Pošto su specijalne izgradnje, vektorske arhitekture su efikasne samo za probleme vezane za matrice i vektore, i upravo zbog toga programi se moraju prilagoditi njihovoj arhitekturi, s tim da prilagodjavanje nije tako kompleksno kao kod MPP arhitekture. Svakako treba istaći, da je izvršavanje nevektorske, tj. skalarne obrade na vektorskim računarima su svakako skuplje nego na bilo kakvom drugom računaru.

Dalje informacije o vektorskim arhitekturama se mogu pronaći na sledećim Internet adresama: www.cray.com , pošto najpoznatiji proizvođač superkompjutera ima na proizvodnoj paleti i računare vektorske arhitekture, www.nec.com , www.fujitsu.com i www.hitachi.com .

5.1.1.4.Mainframe računari

Mainframe računari predstavljaju snažne i brze računare koji su predviđeni za centralizovanu obradu podataka, za istovremeno usluživanje terminalske mreže od čak i više desetine hiljada terminala bez inteligencije. Mada su stručnjaci pre deset godina nagovestili potpuno napuštanje centralizovane obrade i izumiranje mainframe računara, oni su i danas prisutni na tržištu uz tendenciju rasta prodaje. Ova arhitektura podržava filozofiju transakcije kako na softverskom tako i na hardverskom nivou. Transakcija je skup aktivnosti ili operacija koje se izvršavaju u nizu. Skup aktivnosti obuhvaćen u okviru transakcije se sprovodi u celosti, ili, ako se transakcija nije mogla izvršiti u celosti, zadržava se stanje od pre početka izvršenja. Transakciona obrada uvek obezbeđuje konzistentno stanje podataka. Na nivou hardvera obezbeđuje se zamena za deo konfiguracije koji se pokvari, kao što je to kod procesora IBM S/90, unutar kojeg se u slučaju otkaza ili greške procesora, zadatak preuzima “rezervni” procesor i završava obradu.

U mainframe računarima se primenjuju klasični CISC procesori sa izrazito velikim brojem instrukcija različite dužine zbog binarne kompatibilnosti ovih računara sa računarima izradjenim od pre dvadeset-trideset godina. Broj procesora retko prelazi 14, a rade u simetričnom načinu

rada (SMP). Samo u trenutku inicijalizacije se određuju uloge samih (inače potpuno identičnih) procesora: neki će od njih izvršavati uloge CPU-a, neki će izvršiti I/O operacije (SAP – System Assist Processor), a neki će biti u “rezervi” (SPU – Spare Processing Unit). Unutar procesora postoje dve jedinice za izvršavanje instrukcija koje rade paralelno i posle svakog realizovanog koraka izvršenja instrukcije vrši se kontrola unutar istog procesora. U slučaju različitog stanja jedinice za izvršavanje, korak obrade se ponavlja u istom procesoru. Ako i ponovna kontrola javlja grešku, obrada se predaje drugom (rezervnom) procesoru kojeg CPU-i zajedničko određuju. Sve je to transparentno u odnosu na obradu i korisnika.

Uobičajeno je da procesori kompleksnije instrukcije realizuju pomoću mikrokoda ili mikroprograma. Procesori mainframe računara imaju “medjunivo” između instrukcija i mikrokoda, a to je “milikod”. Pojedine milikod instrukcije obuhvataju veći broj mikrokoda, i grubo odgovaraju makroima assembly jezika. Procesorski moduli se naravno mogu menjati u toku rada.

Zahvaljujući specijalnoj konstrukciji i gradnji pouzdanost procesora mainframe računara se ne može uporediti sa pouzdanošću procesora opšte namene koji se mogu komercijalno nabaviti (CTOS – Commercial Off-The-Shelf). Za povećanje pouzdanosti rada memorijski moduli se takodje multipliciraju uz neprestanu kontrolu rada pojedinih čipova. Neispravni moduli memorije se isključuju iz obrade i njihovu ulogu preuzimaju rezervni čipovi. Takodje se vrši kontrola rada cache memorije i njihova zamena ako neispravno rade. Kontrola pojedinih delova arhitekture se obuhvata pomoću specijalno razvijenog zajedničkog monitor programa, tako da je prosečno vreme između dva uzastopna zastoja (MTBF – Mean Time Between Failure), ili neplanirana zaustavljanja (otkaza) je zavidno dugačko, a može da iznosi i deset godina. Rekonfiguracija hardvera i operativnog sistema može da se obalja u toku rada, te zbog toga ne treba zaustaviti rad sistema.

U cilju zaštite i bezbednosti podataka IBM S/90 može imati koprocesor za kriptografiju, a može da se proširuje sa vektorskim jedinicama za izvršavanje operacije sa matricama i vektorima. Druge informacije o mainframe računarima se mogu pronaći na WEB stranicama njihovih malobrojnih proizvođača: www.ibm.com , njihov najveći proizvođač, a www.amdahl.com www.hitachi.com i www.fujitsu.com su najpoznatiji proizvođači IBM-kompatibilnih (PCM – Plug-Compatible Manufacturer) mainframe računara.

5.1.1.5.Specifične arhitekture

Prvi predstavnik specifične arhitekture je hibridna konfiguracija realizovana na ELTE Univerzitetu u Budimpešti koji nosi ime PMS1 (Poor Man's Supercomputer). Po osnovnoj arhitekturi to je klasterska mreža – COW sa 128 čvorova, a u svakom čvoru se nalazi AMD-ov K6 mikroprocesor radne učestanosti 450 MHz, 128 MB operativne memorije i 2,1 GB disk kapaciteta. Arhitektura računara PMS1, međutim, poseduje pored klasičnog Ethernet prstena za povezivanje čvorova (što je karakteristično za COW-mreže) i posebno projektovanu mrežu velike prenosne brzine za povezivanje čvorova kao što ima MPP arhitektura. Računar se za sada sastoji od samo 32 čvora u ukupnoj vrednosti od oko 20.000 US\$, a može da realizuje 16% od performanse Cray superkompjuteru vrednog nekoliko miliona dolara. Čvorovi PMC1 arhitekture su sastavljeni od klasičnih PC delova na kojima radi Linux operativni sistem. Programiranje ove arhitekture je jako prosto na Fortran i na C jeziku. Komunikacijom preko mreže između čvorova upravlja poseban procesor koji radi po specijalno projektovanom protokolu. Dalje informacije se mogu naći na adresi <http://xxx.lanl.gov/abs/hep-lat/9912059> .

Komisija za atomsku energiju Francuske (CEA) je naručio superkompjuter brzine 5 teraflopsa (pet biliona operacija u kliznom zarezu u sekundi) od proizvođača Compaq, koji će biti sedam puta snažniji od najbržeg sadašnjeg računara u Evropi. Računar će biti MPP arhitekture sa čvorovima SMP izgradnje. Čvorovi će biti superkompjuteri SMP arhitekture izgradjeni od 64-bitnih Alpha (RISC) procesora tipa Alpha Server SC Supercomputer. Operativni sistem na računarima će biti Tru64 Unix, a mreža za povezivanje čvorova se posebno projektuje za ovu naručenu konfiguraciju. Računar će posedovati više od 2000 procesora, a treba da proradi krajem 2001. godine (www.compaq.com).

IBM je objavio podatke o projektu Blue Gene superkompjuteru koji će biti izgradjen za pet godina (do 2005. godine), sadržaće više od milion procesora i imaće brzinu obrade od 1 petafropa (10^{15} flopa). Računar će biti izgradjen po novoj SMASH (Simple, Many and Self-Healing) arhitekturi koja će drastično smanjiti broj operacija za izvršavanje za pojedine procesore (time će smanjiti dimenzije i disipacije pojedinih procesora), u velikoj meri će povećati paralelni rad procesora (broj simultanih, paralelnih niti u obradi biće 8 miliona – za razliku od današnjeg 5000) i biće samostabilizirajuća konfiguracija koja će moći da “zaceli svoje rane”, tj. da udalji delove konfiguracije u kvaru bez potrebe spoljašnje intervencije od strane stručnjaka za održavanje (www.ibm.com).

5.1.2. Osnovne ploče

Nakon upoznavanja sa najčešće korišćenim arhitekturama računara vraćamo se PC računarima, jer su oni najmasovniji u primeni kao osnove za razvoju informacionih sistema i u obradi podataka uopšte.

Kod izbora računara se najčešće ističu samo zahtevi u vezi takta i/ili tipa procesora, kapacitet memorije, tip i/ili performanse grafičke kartice i veličina adresnog prostora (kapacitet) hard diska. O osnovnoj ploči malo ko brine, mada najčešće ona uslovljava dalja proširenja i obezbeđuje stabilnost sistema.

Opšta tendencija je danas da se povećava stepen integracije osnovnih ploča: sve veći broj kontrolera se integriše na osnovnu ploču: IDE (Integrated Device Equipment) i EIDE (Enhanced Integrated Device Equipment) kontroleri za diskove, audio, video i mrežne kartice integrisane u jedan ili nekoliko čipova, modemi itd. Kod ovakvog rešenja ne pojavljuju se problemi kompatibilnosti u radu ovih jedinica i osnovne ploče.

Radom osnovne ploče upravlja porodica električnih krugova koja se u svakodnevnom životu zove čip set. Svaki čip set ima svoje specifičnosti i obično svaki novorazvijeni mikroprocesor zahteva novi, svojstven čip set. Najveći proizvođači upravljačkog čip seta za osnovne ploče su Intel (www.intel.com), Via (www.viahardware.com) i Ali (www.ali.com). Najpoznatiji i najčešće korišćeni čip setovi su sledeći: Intel 82440BX, Intel 82440ZX, Intel 82810E (i810E), Intel 82815E (i815E), Camino (i820), i840, i850 (za Pentium IV), Via Apollo KX133, Via Apollo Pro 133, Via Apollo Pro Plus, Ali Aladdin TNT2 GFX Pro, AMD 750.

Osnovna ploča obezbeđuje sat realnog vremena, osnovni clock ili osnovnu frekvenciju rada na kojoj radi magistrala. Brzinu procesora određuje frekvencija FSB-a (Front Side Bus – magistrala), što predstavlja ulaznu frekvenciju za procesor i multiplikator u samome procesoru. Standardne frekvencije FSB-a su 66, 100 i 133 MHz, a obično je moguće podesiti i druge frekvencije na skoro svim osnovnim pločama u granicama između 66 i 150 (200) MHz. Podešavanje FSB frekvencije može se realizovati kontinualno, pomoću BIOS setup-a ili u koracima uz pomoć jumpera, DIP-prekidača ili BIOS setup-a. Multiplikator u procesoru se takodje može menjati pomoću za to predviđenih jumpera, DIP-prekidača i BIOS setup-a.

Ploče sadrže podnožje za procesora koja su danas najčešće tipa Slot 1, Socket 370, Socket 423 – za Pentium IV (Intel), ili Socket A (AMD Athlon) i Socket 462 (AMD Duron i Thunderbird). Neke osnovne ploče imaju dva različita tipa procesorska podnožja (Acorp 6VIA96A – Slot 1 i Socket 370). Napon napajanja procesora se na svim osnovnim pločama može menjati ručno (jumper ili BIOS setup) ili automatski, pošto podnožja mogu prihvatiti razne procesore sa različitim naponima napajanja.

Slotovi za razna proširenja na osnovnoj ploči pokazuju šta podržava osnovna ploča: ISA (Industry Standard Architecture), EISA (Extended Industry Standard Architecture) i VLB (VESA Local Bus) slotovi predstavljaju prošlost, no proizvođači obično ostavljaju jedan ISA slot radi kompatibilnosti za starije delove konfiguracije. PCI (Peripheral Component Interconnect) slotovi, čija je sabirnica 32-bitna (kao VLB), a frekvencija rada je od 20 do 33 MHz su dominantni na današnjim osnovnim pločama (njihov je broj danas od 3 do 6). Skoro svaka kartica za proširenje se može nabaviti u PCI izvedbi. Za aplikacije koje intenzivno koriste grafiku PCI sabirnica nije pogodna zbog ograničene brzine u odnosu na zahteve grafike, zato je njima razvijena AGP (Accelerated Graphics Port) sabirnica sa tri brzine rada (AGP, AGP2x i AGP4x), preko koje video kontroler može neposredno pristupiti operativnoj memoriji. Osnovne ploče sa najnovijim čip-setovima sadrže AMR (Audio/Modem Riser) priključak za pristup audio-kontroleru i modemu koji su integrisani delimično ili u potpunosti na osnovnu ploču. Takodje je moguće već naći rešenja za bežični prenos podataka između PC-ja i njegovih ulaznih i izlaznih jedinica po IrDA (Infrared Data Association) standardu ili bluetooth tehnologiji.

Disk kontroleri su prisutni na svim osnovnim pločama, no postoje i specijalni kontroleri koji obezbeđuju veću stabilnost, sigurnost i prenosnu brzinu podataka sa, za sada najvažnijeg eksternog memorijskog medijuma. SCSI (Small Computer System Interface) nije predviđen samo za diskove, nego za sve ostale delove konfiguracije PC-računara koji su pripremljeni za takvu vrstu kontrolera. U cilju smanjenja rizika od gubljenja podataka na diskovima, primenjuje se RAID (Redundant Array of Inexpensive/Independent Disks) tehnologija pomoću istoimenog kontrolera, koji pored veće sigurnosti obezbeđuje i brži pristup podacima smeštajući neparne blokove podataka na jedan, a parne blokove na drugi disk (u slučaju postojanja dve jedinice diska). Prenos podataka sa diskova se danas obavlja po UDMA (Ultra Direct Memory Access) porodici standarda: UDMA33, UDMA66 i UDMA100 koji obezbeđuju prenosne brzine od 33, 66 i 100 Mbyte/s. U novije vreme sve više dobije prednost brzi serijski prenos podataka preko tankog kabla. Za takvu realizaciju postoje dva standarda: USB i IEEE 1394 (Firewire). USB (Universal Serial Bus) obezbeđuje brz serijski prenos podataka (1,5 Mbit/s ili 12 Mbit/s – najnovija verzija USB 2.0 predviđ 40 puta veće prenosne brzine), napajanje uređaja (do 0,5 A) i mogućnost povezivanja maksimalno 127 uređaja. IEEE 1394 obezbeđuje slične karakteristike prenosa: brzinu od 200Mbit/s, maksimalan broj uređaja je 64, a napajanje uređaja je obezbeđeno.

Na osnovnoj ploči se nalazi BIOS (Basic Input Output System) čip koji sadrži programe za testiranje stanja konfiguracije i pokretanje rada računara. Neki proizvođači osnovnih ploča stavljaju čak dva BIOS ROM čipa na ploču. U slučaju punjenja nove verzije BIOS-a ne sme biti

prekida rada sistema (računara), ako se to ipak dogodi računar se više neće »probuditi«. Za takve situacije je od velike koristi drugi BIOS čip, koji bez problema preuzima ulogu i obezbeđuje dalje nesmetano korišćenje PC-ja. Nove verzije BIOS-a se mogu pronaći na Web-stranama njihovih proizvođača (www.award.com i www.ami.com).

Savremene osnovne ploče poseduju priključak (priključke) za ventilator. Procesori su oni delovi konfiguracije koji se najintenzivnije greju, tj. imaju najveću disipaciju energije. To je, naravno, posledica velikog stepena integracije u tehnologiji izrade procesora, a i drugih integrisanih krugova. Elektronika koja kontroliše temperaturu pojedinih delova konfiguracije na osnovnoj ploči može i da upravlja radom (brzinom okretanja) ventilatora. Idealna temperatura delova računara se nalazi između 30 i 45 °C.

Osnovne ploče koje odgovaraju ATX standardu imaju mogućnost da pojedine delove konfiguracije nakon podešljivog intervala vremena isključe iz aktivnog rada, štedeći tako energiju i produžavajući radni vek odgovarajuće jedinice. Načini aktiviziranja tako isključenih jedinica se po pravilu može definisati u BIOS-u.

Osnovne ploče su glavna tema na sledećim Web adresama:

www.bxboards.com ,
www.motherboards.com i
www.slota.com .

Najpoznatiji proizvođači obezbeđuju pristup dokumentaciji svojih osnovnih ploča na adresama:

www.abit.com.tw/english/download/manual.htm ,
www.asus.com/Products/Motherboard/manual.html ,
www.epox.com/support/database.html ,
www.support.intel.com/support/motherboards ,
www.soltek.com.tw/English/download/manuals.htm ,
www.soyousa.com/manuals.html ,
www.supermicro.com/PRODUCT/UsersManual.htm i
www.zida.com/support/manual.htm .

5.1.3. Procesori

Davno je prošlo vreme kada su procesori izvršavali mašinske instrukcije samo pomoću hardvera. Složenije instrukcije se danas u svim procesorima izvršavaju pomoću mikroprograma.

Procesori po arhitekturi se mnogo razlikuju: sasvim su drugačije izvedbe procesori mainframe računara od procesore za PC-je. Generalno važi podjela procesora na CISC (Complex Instruction Set Computer) i RISC (Reduced Instruction Set Computer) procesore. CISC procesori imaju veliki broj instrukcija od kojih mnoge izvršavaju više funkcija (takvi su i procesori u PC računarima). RISC procesori imaju znatno manji broj instrukcija koje se tako mogu hardverski izvršavati, zato su brži i efikasniji od prethodne vrste, ali su i znatno skuplji.

Konvencionalni, 32-bitni mikroprocesor PC računara je povezan sa drugim delovima konfiguracije pomoću sledeće tri magistrale: adresne, upravljačke i magistrale za podatke. Savremena adresna magistrala (Address Bus) je širine 36 bita, tako da je moguće adresirati ukupan adresni prostor od 64 Gbyte-a. Upravljačka magistrala (Control Bus) služi za prenos upravljačkih signala između aktivnog elementa obrade (procesora) i drugih delova hardvera. Ova magistrala zavisi od konstrukcije samog procesora. Magistrala za podatke (Data Bus) je danas širine 64 bita, a prenosna brzina između procesora i memorije (naime procesor se može obraćati samo operativnoj memoriji) se meri u Gbyte/s.

Kvalitet procesora mnogi određuju samo na osnovu njegove radne frekvencije. Takva procena nije u potpunosti merodavna, jer brzina rada (obrade podataka) zavisi i od drugih karakteristika (od širine i brzine adresne magistrale i magistrale za podatke). Mogućnosti procesora se procenjuju na osnovu sledećih podataka: broj, kapacitet i namena njegovih registara, broj jedinica za izvršavanje celobrojnih operacija i operacija kliznog zareza – matematički koprocessor ili FPU (Floating Point Unit), broj jedinica za dekodiranje i izvršavanje instrukcija, konstrukcija jedinice za izvršavanje instrukcije (a. superskalarna izgradnja – kad više izvršnih jedinica paralelno rade na izvršavanju različitih instrukcija, kao neki interni multiprocesorski rad, b. pipeline – superpipeline princip rada – izvršavanje instrukcije se realizuje u delovima, a broj delova određuje dubinu pipeline-a), nivoi, kapaciteti i brzine keš (cache) memorije integrisane na procesorskom čipu, kao i set ili skup mašinskih instrukcija koji procesor može izvršiti. Tip podnožja, naponski nivo procesorskog jezgra i tehnologija izrade (realizacija, tj. materijal i debljina povezivanja tranzistora unutar procesora) predstavljaju sekundarne pokazatelje koje informišu o disipaciji i mogućnosti primene procesora na osnovnim pločama.

Najzapaženiji novi 32-bitni procesori u 2000-toj godini su bili AMD Duron (700 MHz), Athlon i Thunderbird (1,2GHz), Intel Pentium 4 (1,5GHz) i IBM Power4 sa novom arhitekturom: sa dva procesorska jezgra i sa tri nivoa cache memorije integrisane na procesorskom čipu.

Pojavili su se i 64-bitni procesori, za sada ne za široko tržište: Intel Itanium IA-64 i AMD Sledgehammer. Svi ovi procesori koji su predviđeni za rad u PC računarima su hardverski kompatibilni sa važećim standardnim setom instrukcija koji ima oznaku x86.

Novu filozofiju u proizvodnji procesora je otvorila novi proizvođač mikroprocesora Transmeta sa svojim Crusoe procesorima. Softverski procesor je u potpunosti kompatibilan sa današnjim PC procesorima, međutim Crusoe radi sa VLIW (Very Large Instruction Word) instrukcijama dužine 128 bita, a konvertovanje x86 instrukcija na VLIW vrši pomoću Code Morphing Software-a. Time se smanjuje kompleksnost arhitekture samog procesora, broj potrebnih tranzistora za realizaciju čipa, kao i disipacija.

Na Univerzitetu u Mačesteru, u Engleskoj, su razvili ARM7, prvi asinhroni mikroprocesor o kojem se malo zna što se same arhitekture tiče, ali je izvesno, zbog asinhronog rada, da će taj čip znatno manje energije trošiti od njegovih sinhronih savremenika. Tranzistori u ARM7 se uključuju samo ako imaju korisni signal na svojim ulazima.

Za realizaciju aplikacija u realnom vremenu (naročito u obradi slike, zvuka i prenosa signala u bežičnoj, mobilnoj komunikaciji) u zadnjih deset godina se isključivo koriste DSP (Digital Signal Processor) procesori. Najveći proizvođač DSP procesora je kompanija Texas Instruments.

Podrobnije o procesorima je moguće naći puno informacija na sledećim adresama:

www.intel.com ,

www.amd.com ,

www.cyrix.com ,

www.ibm.com

www.transmeta.com ,

www.arm.com i

www.ti.com .

5.1.4. Memorije

U svakom računarskom sistemu postoji više tipova memorije, u stvari cela memorijska hijerarhija. Već i registri procesora predstavljaju privremenu memoriju iako njih niko ne smatra memorijom. Memorijska hijerarhija se sastoji od cache memorije, operativne memorije i memorijskog prostora na diskovima u klasičnom shvatanju. U novije vreme su se pojavile i flash memorije koje svakako spadaju u memorijsku hijerarhiju, a tu su i ostali uređaji za memorisanje

podataka: floppy uredjaji, CD-ROM, DVD-ROM i najnoviji FVD-ROM uredjaji, a za arhiviranje velikih količina podataka i danas su najviše korišćeni traka uredjaji. Uredjaji magnetne trake se danas retko sreću u konfiguracijama PC računara, ali u velikim mainframe sistemima još uvek nema za njih bolje, pouzdanije zamene.

Registri predstavljaju najviši nivo memorijske hijerarhije, nalaze se u samom procesoru, a brzina rada odgovara brzini procesora. Njihov broj i kapacitet su veoma mali i služe samo za privremeno memorisanje podataka (medjurezultata i rezultata obrade).

Cache (keš) memorije predstavljaju sledeći nivo u memorijskoj hijerarhiji. Ove memorije su nešto sporije od brzine rada registara, ali rade na približno istim frekvencijama kao i procesor. Njihov kapacitet je veći od ukupnog kapaciteta registara i predstavljaju skup medijum za (uglavnom privremeno) memorisanje podataka. Danas se koristi maksimalno tri nivoa cache memorije. L1 je oznaka za prvi nivo cache-a, koji se ugrađuje na mikroprocesorski čip i tada radi na frekvenciji procesora. L1 cache kod Intela ima kapacitet od 32 Kbyte-a, a kod AMD-a 128 Kbyte-a. Drugi nivo cache-a ima oznaku L2, i u zadnje vreme se i on ugrađuje u sam procesor. Karakteristične veličine L2 cache-a kod Intela su 128 Kbyte, a kod AMD-a 256 Kbyte-a. Kod novijih rešenja se pojavljuje i treći nivo cache memorije sa oznakom L3, koji se nalazi na čip setu, na osnovnoj ploči (čip set Mamba, proizvođača Micron), ili čak unutar procesora, tj. na procesorskom čipu (IBM Power4).

Operativna memorija predstavlja treći nivo u memorijskoj hijerarhiji, a brzina rada strujnih krugova operativne memorije je grubo za red veličine (deset puta) sporija od brzine rada procesora. Operativna memorija svih, pa i PC računara sadrži RAM (Random Access Memory) tipove memorije. Njih ima više vrsta, najstarija je DRAM (Dinamyc Random Access Memory) memorija koja se dugo koristila kao operativna memorija zbog jednostavne tehnologije izrade, dosta velikog kapaciteta i jeftine cene. DRAM je izradjen od FET tranzistora i kondenzatora, relativno je spor (do 18 MHz frekvencije) i neminovno je često obnavljati sadržaj memorijske ćelije zbog pražnjenja naelektrisanja kondenzatora. Druga vrsta RAM memorije je EDORAM (Extended Data Out RAM), koji se i danas koristi u konfiguracijama i može da postigne radnu frekvenciju od 50 MHz. Danas se najviše koristi SDRAM (Synchronous DRAM) memorija koja istina zahteva komplikovaniji kontroler (zbog podele memorije na dva memorijska bank-a), ali je njena radna frekvencija 100 MHz, odnosno u najnovije vreme 133 MHz. Proizvođač memorije Rambus nudi tip operativne memorije DRDRAM (Direct Rambus DRAM) ili RDRAM (Rambus DRAM) koji je jako brz, ali je i preskup u odnosu na (znatno) sporije SDRAM memorije. Frekvencija rada RDRAM memorije je 800 MHz, a najnovija verzija radi na 1066 MHz (!). Najveći proizvođač memorije Micron, kao odgovor na Rambus memorije, se pojavio na tržištu

DDR SDRAM (Double Data Rate SDRAM) tipovima memorije koji su jedva skuplji od običnih SDRAM (SDR SDRAM – Single Data Rate SDRAM) memorija, a dvostruko su brži. Istraživači IBM-a eksperimentišu na izradi statičkih RAM memorijama na nivou dimenzije atoma. Na univerzitetu Keele u Engleskoj su patentirali realizaciju statičke memorije kapaciteta 86 Gbyte/cm², što znači da na kartici dimenzije 85x55x15 mm stane 10800 Gbyte (10Tbyte) podataka po ceni od oko 50 US\$. Za potrebe grafičkih kartica je razvijena specifična VRAM (Video RAM) memorija čiji svaki bit sadrži podatak o jednoj tački slike ekrana. Princip rada DRAM-a i VRAM-a je jako sličan, razlika je u njihovoj izgradnji: VRAM sadrži dva porta za podatke (Dual Port RAM), jedan za upis i jedan za čitanje i tako je moguće istovremeno upisivati (modifikovati) i iščitavati njegov sadržaj.

Tehnologija smeštanja podataka na magnetni disk se duže vreme ne menja. Povećanje gustine zapisivanja se postiže uglavnom povećanjem brzine obrtaja medijuma, odnosno ploča disk paketa sa početnih 3000 na 5400, 7200 i 15000 (Seagate CheetahX15, gde je brzina prenosa podataka 48 Mbyte/s). Razvijena je i nova glava za čitanje i pisanje magnetnog diska koja koristi tehnologiju GMR i obezbeđuje prenosnu brzinu od 128 Mbyte/s. Ona se predviđa za primenu uređaja sa velikim brojem obrtaja ploča (www.readrite.com i www.guzik.com). Traganje za novim tehnologijama u cilju povećanja gustine pisanja se naravno nastavlja, tako da se svaka 3-4 meseca se pojavljuje proizvod, disk većeg kapaciteta. Posle Seagate-a (60 GB) i Maxtora (60 GB) pojavio se IBM Deskstar 75GXP disk sa pet ploča i 75 GB kapaciteta. Kompanije Lucent, Bell Laboratories i Imation rade na zajedničkom projektu holografskih izmeljivih diskova kapaciteta 125 GB (u prodaji bi se našli za dve – 2002. – godine). IBM je obavestio javnost o novoj tehnologiji smeštanja podataka na memorijski medijum (disk), koja će omogućiti proizvodnju disk uređaja kapaciteta reda terabajta. “Proces predstavlja novi pristup sa 3D samoformirajućim strukturama na molekularnom nivou”²- izjavio je predstavnik IBM-a.

Standardna CD ploča ima kapacitet od 650 Mbyte-a, DVD ploča 4,7 Gbyte-a (5,2 Gbyte-a – Imation), a nova tehnologija nazvana FMD – ROM njujorške kompanije Constellation 3D koristi ploče kapaciteta od čak 140 Gbyte-a. FMD – ROM ploča memoriše podatke u najviše 100 slojeva i omogućava istovremeno čitanje podataka sa više slojeva. Uređaj će po procenama koštati 100, a jedna ploča 2 US\$.

Magnetna traka kao medijum za arhiviranje podataka je i danas aktuelna. Istina ne za kućne PC konfiguracije, nego za veće računarske sisteme. IBM i Tandberg intenzivno rade na usavršavanju tehnologije zapisivanja podataka na magnetne trake. Kao rezultat ovog razvojnog posla, pojavile su se magnetne kasete standardnih dimenzija sa kapacitetom od 100 Gbyte-a

² PC Press br. 58., jul-avgust 2000.

Informacije u vezi memorijskih medijuma i uređaja se mogu pronaći i na sledećim Internet adresama:

www.microm.com ,
www.rambus.com ,
www.cmruk.com/cmrKHD.html ,
www.kingston.com ,
www.hitachi.com ,
www.samsung.com ,
www.tandberg.com ,
www.seagate.com ,
www.lucent.com ,
www.maxtor.com ,
www.immation.com ,
www.ibm.com .

5.1.5. Multimedija

Multimedija obuhvata sliku, zvuk i grafičke primene kao i sve uređaje koji su vezani za njihovu obradu (razvoj i reprodukciju).

Monitor računara sadrži minimalni broj strujnih krugova u cilju lakšeg proširivanja. Upravljačka jedinica monitora se smešta na posebnu, tzv. grafičku karticu. Na grafičkoj kartici su se menjali razni kontroleri kao što su MDA (Monochrome Display Adapter, dvobojni sa kapacitetom sopstvene memorije od 4Kbyte-a i rezolucijom od 80x25 karaktera – jedan karakter je ostvaren pomoću 9x14 tačaka-pixela), CGA (Color Graphics Adapter sa memorijom od 16 Kbyte-a, sa grafičkim i tekstualnim načinom rada: u tekstualnom načinu je rezolucija 80x25 karaktera – jedan karakter je sastavljen od 8x8 pixela, a u grafičkom načinu sa dve rezolucije i to 600x200 pixela u dve boje ili 320x200 u četiri boje), Hercules (sa tekstualnim i grafičkim radom i rezolucijama: u tekstualnom načinu 80x25 karaktera – jedan karakter predstavljen pomoću 9x14 pixela, a u grafičkom načinu 720x348 tačaka) monohromatski kontroler, EGA (Enhanced Graphic Adapter sa kapacitetom memorije od 256 Kbyte-a i rezolucijom od 80x43 karaktera – jedan karakter sačinjen od 8x14 pixela – u tekstualnom načinu i 640x350 tačaka i 16 boja u grafičkom načinu rada), VGA i SVGA.

VGA (Video Graphics Array) je razvio IBM za svoje PS/2 računare. Tekstualni način rada je sličan, tačnije malo je unapredjan u odnosu na EGA standard, a u grafičkom načinu rada

ostvaruje rezoluciju 640x480 i 16 boja. Svaka VGA kartica ima sopstveni BIOS (pomoću njega se može podešavati, emulirati rad bilo kojeg drugog kontrolera).

SVGA (Super VGA) je najšire rasprostranjen standard na PC tržištu. U okviru ovog standarda su definisane konkretne vrednosti rezolucije ekrana, broj (dubina) boja i shodno tome treba da se obezbedi potrebna memorija na grafičkoj kartici (tabela 5.1.5.-1.). Predstavljanje boja pomoću 16 bita se često naziva Hi-Color, a primena dubine boja od 24 bita ima naziv True-Color.

Tabela 5.1.5.-1. Kapacitet memorije grafičke kartice u funkciji rezolucije i dubine boja

Dubina boja (u bitovima)	4	8	16	24
Broj boja (nijansi)	16	256	65536	$16,7 \cdot 10^6$
Rezolucija ekrana				
640x480	256 KB	512 KB	1 MB	1 MB
800x600	256 KB	512 KB	1 MB	1,5 MB
1024x768	512 KB	1MB	2 MB	2,5 MB
1280x1024	1 MB	1,5 MB	3 MB	4 MB

Centralni čip na grafičkoj kartici je kolo za ubrzanje ili akcelerator, koji se u današnje vreme izborio za naziv procesora (GPU – Graphical Processing Unit). U zadnje vreme su bile favorizovane grafičke kartice sa akceleratorima Voodoo porodica (porodice Voodoo 1 do Voodoo 5) kompanije 3dfx, a od sredine 2000. godine grafičke kartice za probirljive kupce može da krasi jedino jedan od grafičkih procesora porodice GeForce, proizvođača NVidia. Postoje naravno i druga rešenja (proizvoda) GPU-a (Savage 2000 kompanije S3, G800 proizvođača Matrox, itd), ali i odnos cena/performansa ukazuje na superiornost Nvidie (brzina i tehnologija takodje). Problematika u vezi grafičkih kartica i akceleratora je pristupačna i na Internetu.

Proizvođači GPU-a:

www.s3.com ,
www.nvidia.com ,
www.matrox.com ,
www.3dfx.com ,
www.ati.com ,

Proizvođači grafički kartica:

www.diamond.com ,
www.elsa.com ,
www.ct.com ,
www.asus.com ,
www.microstar.com .

Prvi monitori kućnih računara su bili televizori, no njihov kvalitet izrade i izobličenja usled dvostrukog pretvaranja signala (slika sa računara je pretvarana u visokofrekvencijski signal koji je još jednom konvertovan u cilju prikazivanja na ekranu) su ih ubrzo eliminisali kao deo konfiguracije računara. Grafičke aplikacije traže grafičke kartice i monitore velike rezolucije. Monohromni monitori danas već predstavljaju istoriju, ali isti princip rada se primenjuje i kod monitora u boji: katodna cev sa modifikovanom anodom i skretnim elektrodama po redovima osvetljava ekran. Tri elektronska topa (RGB) skeniraju unutrašnju stranu ekrana color monitora i to tačku po tačku. Tačke koje se skeniraju (osvetljavaju) pomoću mlaza elektrona se sastoje od tri manje tačkice smeštene u obliku trougla (delta konstrukcija katodne cevi), ili se smeštaju tri vertikalne crte (i jedna tamna crta) jedna do druge u jednoj liniji (in-line konstrukcija katodne cevi). Prikazivanje (iscrtavanje) slike se može obaviti pomoću poluslika (prvo se skeniraju neparni redovi ekrana, a posle parni) i tada se za prikazivanje slike na monitoru kaže da je interlaced. Kod novijih monitora slika na ekranu se u celosti prikazuje u jednom prolazu (non-interlaced), a broj slika u sekundi iznosi oko 70. Danas se koriste ravni, in-line monitori bez vibracije slike (non-interlaced) u multisync (koji mogu menjati frekvenciju prikazivanja slike) izvedbi. Postoje dva široko prihvaćena standarda za klasične monitore: MPRII i DMPS. MPRII je izradjen u Švedskoj i ograničava jačinu zračenja elektromagnetskih talasa u dva frekvencijska opsega, od 5 Hz do 2 kHz, i od 2 kHz do 400 kHz. DMPS definiše četiri radna rešima za monitore sa definisanim maksimalnim vrednostima potrošnje električne energije.

Ekрани sa tečnim kristalima (LCD – Liquid Crystal Display) se pre svega primenjuju u mobilnim računarima, ali imaju primene i u klasičnim PC konfiguracijama. Dva osnovna tipa LCD ekrana postoje na tržištu: DSTN i TFT. DSTN (Dual-Scan Twisted Nematic) ekran je spor u iscrtavanju slike na ekran (za prikazivanje slike treba oko 300 ms, tj. maksimalni broj prikazanih slika u sekundi na ekranu je 3!) i primenjuje se samo kod jeftinijih mobilnih (laptop, notebook, palmtop) računara. Kod DSTN ekrana upravljački vodovi služe za preraspodelu molekula tečnog kristala, upravo je zato vreme odziva dugačko. TFT (Thin Film Transistor) koristi FET (Field Effect Transistor) tranzistore za upravljanje stanja molekula tečnog kristala. Ovde upravljački vodovi brzo mogu otvarati ili zatvarati FET-ove, zato je i iscrtavanje slike znatno brže (ispod 25 ms, tj. broj prikazanih slika u sekundi na ekranu je 40!).

Upoređenjem klasičnih sa LCD monitorima može se zaključiti da je prednost na strani LCD monitora ako posmatramo težinu, dimenziju ekrana (efektivna veličina ekrana kod klasičnih monitora je manja za 1 do 1,5 cola), potrošnju, fizičke dimenzije, osvežavanje slike (kod LCD ekrana slika se uopšte ne osvežava) i elektromagnetsko zračenje (nema zračenja).

Karakteristike oba tipa ekrana su iste što se rezolucije i broja boja (nijansi) tiče, a klasični monitori su u prednosti ako uporedimo njihovu cenu, intenzitet slike i kontrast slike.

Ponude najpoznatijih proizvođača monitora se mogu prelistati na sledećim adresama:

www.acer.com ,

www.belinea.com ,

www.hansol-us.com ,

www.hitachi.com ,

www.nec.com ,

www.panasonic.com ,

www.sony-cp.com i

www.viewsonic.com .

Audio kartice su oživele najpre ili pre svega video igre koje se izvršavaju (igraju) na računaru. Pošto se podatak memoriše u digitalnom obliku, neophodno je pretvarati analogni signal (zvuk) u digitalni u fazi akvizicije (smeštanja na memorijski medijum), odnosno digitalni u analogni u procesu reprodukcije. Najpoznatije su audio kartice porodice Sound Blaster kompanije Creative Technology (Sound Blaster, Sound Blaster Pro, Sound Blaster 16, Sound Blaster AWE 32, Sound Blaster AWE 64, Sound Blaster AudioPCI 64V, itd.). Stariji tipovi su predviđeni za priključak na ISA port (sabirnicu), a savremeni su pripremljeni za PCI. Proizvođači audio kartica su u dosta velikom broju prisutni na tržištu, sa širokom gamom proizvoda. Kompanije nude velik broj zvučnih kartica različitog kvaliteta i cena. Tako Terratec nudi vrhunske kartice tipa EWS 88 MT, a takodje i kartice srednjeg kvaliteta EWS 64 XL. Delovi (jedinice) sistema za obradu zvuka se delimično mogu integrisati na osnovnu ploču (upravljačka jedinica je svakako tamo) kako propisuje standard za Audio Codec interfejs iz 1997. godine (AC'97). AMR (Audio/Modem Riser) je standard koji proširuje sistem AC'97 i opisuje AMR karticu (na kojoj se nalazi modem i priključak modema) i njegov priključak na ISA ili PCI magistralu.

U obradi slike i zvuka se danas koriste isključivo specijalizovani procesori za obradu digitalnih signala (DSP – Digital Signal Processor) prvenstveno zbog potrebe obrade u realnom vremenu. Zbog ograničenog broja funkcija, tj. operacija DSP procesori su optimizirani za izvršavanje tih funkcija, a ujedno i rasterećuju mikroprocesor PC-ja od izvršavanja tih operacija.

Informacije o audio karticama se mogu naći na sledećim adresama:

www.creativelabs.com ,

www.diamondmm.com ,

www.digitalaudio.com ,
www.dtk.com.tw ,
www.gravis.com i
www.terratec.de .

5.1.6. I/O uređjaji

Pre dvadeset godina sam u jednoj knjizi pročitao mišljenje autora da su do tada u Americi već sve i svašta povezali na računar bilo kao ulazni ili izlazni uređaj (nisu valjda ni na plug zaboravili). Bila je to naravno samo šala, ali u svakoj šali ima istine. Danas je, svakako, situacija »naprednija«, no nas će ovde interesovati samo skup uređaja koji se standardno koristi kod PC računara kao ulazni, izlazni ili ulazno – izlazni modul konfiguracije.

Ulazni uređjaji sakupljaju informacije iz bliže ili dalje okoline PC računara, preko njih može okolina uticati na rad računara. U ulazne uređjaje spadaju tastatura, miš, trackball, digitajzer (tabla za digitalizaciju), skener, mikrofoni i kamera da samo najpoznatije pomenemo.

Izlazni uređjaji omogućuju da računar deluje na svoju okolinu, a među njih spadaju monitor, štampač, ploter i zvučnici (slušalice).

Ulazno-izlazni uređjaji mogu primiti ulazne informacije (signale), a takodje mogu slati izlaze. Među njih spadaju monitori osetljivi na dodir (touch screen), modemi i mrežne kartice.

Tastatura je inteligentni deo konfiguracije, jer njenim radom upravlja mikroprocesor, a prenos podataka prema računaru je serijski. Postoje XT i AT tastature (način kodovanja je drugačiji), i mada su XT računari nestali iz upotrebe, neke današnje tastature još uvek podržavaju oba načina kodovanja.

Miš je nezamenljivi deo hardvera zbog grafičkih operativnih sistema i softvera. Najrasprostranjeniji tip miša je optomehanički sa kuglicom jer je najprostiji što se konstrukcije tiče, a i najjeftiniji. Optoelektrični miševi su već prisutni na tržištu i može se očekivati njihov prodor zbog robusnosti i pouzdanijeg rada (nemaju pokretne delove), istina, cena im je još malo visoka. Postoje i druge vrste realizacije miša koji rade na osnovu piezoelektričnog efekta ili pomoću ultrazvuka, ali su ovakvi miševi prvenstveno zbog njihove cene malo traženi na tržištu. Mada miš može komunicirati i preko RS232 priključka, danas su skoro isključivo traženi uređjaji sa PS/2 priključkom (tako se oslobađa jedan serijski port). Najveći broj miša ima dva tastera (levi i desni), a u zadnje vreme sve češće je prisutan i treći (novi softveri podržavaju i treći taster miša dodeljujući mu razne funkcije).

U novije vreme u cilju pojednostavljenja identifikacije korisnika i onemogućavanja neovlašćenog pristupa na tastaturu ili na miš se smešta čitač otisaka prstiju. To je minijaturni skener koji snimak prsta ili samo nekoliko karakterističnih tačaka snimka (od 40 do 60 – na osnovu kojih se ne može rekonstruisati kompletan otisak) šalje preko USB ili paralelnog porta u računar u cilju identifikacije osobe koja želi koristiti računar. Na tom principu funkcioniše tastatura Cherry G81-12000, odnosno neki modeli miša proizvodnje Siemens i Logitech.

Trackball je ulazni uređaj koji ima sve funkcije miša i koristi se umesto miša u uslovima kada nema dovoljno prostora za manevrisanje sa mišem. Trackball je miš “okrenut na leđa” kod kojeg se ne pomera uređaj nego se samo kuglica okreće. Radi na optomehničkom principu i ima po pravilu dva tastera.

Tabla za digitalizaciju se najviše primenjuje kao korisno pomagalo kod CAD programa. Izrađuje se od specijalne plastike sa induktivnim senzorima na poledjini. Senzori su povezani u matricu, a broj senzora određuje osetljivost ili stepen (finoću) razlaganja. Izrađuju se u raznim dimenzijama i olakšavaju (ubrzavaju) projektovanje pomoću raznih CAD programa.

Skener služi za pretvaranje arhiviranog materijala (izvornog dokumenta na papiru) u elektronski oblik. Skener pravi sliku dokumenta, tj. vrši digitalizaciju slike o dokumentu. Senzor kod digitalizacije je CCD (Charge Coupled Device) koji se sastoji od puno elemenata. Elementi CCD-a su fotometri koji na izlazu daju analogni signal (napon) koji se pomoću ADC (Analog-Digital Converter) pretvarača preinači u digitalni signal. Skeneri mogu biti ručni ili stoni, skeniranje mogu realizovati u jednom ili više prolaza. Digitalizaciju pomoću skenera je moguće izvršiti na četiri načina: a) Line art, b) Halftoning, c) Grayscale i d) True Colour. Danas su u korišćenju najviše stoni skeneri koji digitalizaciju slike vrše u jednom prolazu, a koriste Grayscale ili True Colour načinu u zavisnosti da li su crno – beli ili kolor skeneri u pitanju. Optičko razlaganje direktno utiče na kvalitet slike. Savremeni skeneri imaju optičko razlaganje 600x1200 dpi (Dot per Inch) ili ppi (Pixel per Inch). Poboljšanje razlaganja se vrši interpolacijom (uz primenu neke matematičke metode za interpolaciju) bilo hardverski ili softverski. Moć razlaganja skenera generalno određuje i njegovu cenu. Skeneri mogu kreirati više tipova slika (više tipova datoteka), medju kojima su najčešći: BMP (Bitmap), TIFF (Tagged Image File Format), PCX (PC Paintbrush) i JPEG (JPG). JPEG je format slike koji iziskuje najmanji memorijski prostor za smeštanje na disk uz zadržavanje kvaliteta slike (dubina boje je True Colour, tj. 24 bita). Često se proizvođači skenera zajedno sa uređajem isporučuju i neki OCR (Optical Character Recognition) program za prepoznavanje teksta. Skeniranjem teksta, propuštajući kroz OCR softver se može dobiti tekstualni fajl, koji zauzima mnogo manji memorijski prostor od skeniranog oblika, a i tekstualne modifikacije su moguće.

Mikrofon, a pogotovu kamera (naročito WEB – kamera) su se raširili pojavom Interneta i lansiranjem kvalitetnih softvera za obradu zvuka i slike. Oni služe uglavnom za kućnu raznodu, za organizovanje video – konferencija, za daljinsku nastavu (tele-teaching), za video – telefoniju, itd.

O monitorima je već bilo reči, tako da bi ovde samo podvukli, da monitori predstavljaju izlazne ili ulazno – izlazne uređaje, u slučaju da su osetljivi na dodir bili kakvog ili specijalnog tela (specijalna olovka). Istraživanja su intenzivna na polju izrade supertankih, savitljivih monitora.

Rezultati obrade se još i dan danas najčešće odštampaju na tzv. trdjoj kopiji ili papiru. Štampati se može pomoću raznih vrsta štampača koji rade na drugačijim principima štampanja, daju različit kvalitet, uz određene specifične troškove.

Karakter – štampač ima valjak na kojem se nalazi po jedan red od svakog znaka na plaštu valjka. U jednom momentu se odštampaju isti znaci u jednom redu udarima čekića, pa se valjak pomera, okrene sledeći potreban simbol naspram čekića i odštampaju se na odgovarajućim pozicijama ti simboli na hartiji. Ako su svi znaci u redu odštampani vrši se pomeranje hartije za jedan red (line feed) i proces odštampavanja se ponavlja: napr. odštampavaju se slova a (ili A) svugde gde su potrebni u tekućem redu, pa slova b (ili B) itd. Taj tip štampača nije rasprostranjen zbog njegove sporosti i zbog velike buke pri štampanju.

Linijski štampač je poboljšan karakter – štampač koji umesto valjka ima diskove širine jednog znaka, a na plaštu ima ugravirane sve simbole koji se mogu štampati. Broj diskova je jednak širini štampanja u znacima. Pozicioniranje svih diskova je pojedinačno i sinhronizovano pomoću elektronike i mehanike, tako da se istovremenim udarom čekića odštampa ceo red odjednom. Relativno je spor, apsolutno bučan i ima kompleksni mehanizam za sinhronizaciju.

Matrični štampač formira otisak (znakove) pomoću malih iglica (prema broju iglica se dele na devetoiglične i dvadesetčetvoroiglične matrične štampače). Obojena traka (ribbon) koja stvara otisak na hartiji je u kutiji i nema početka, ni kraja. Ti štampači su dugo bili korišćeni, a i danas su još u primeni zbog robustnosti, i jeftine cene. Otisak im nije baš kvalitetan, a brzina takodje nije njihova jača strana.

Ink – Jet štampači rade na dva principa: kod prvog tipa štampača postoji neprekidna cirkulacija mastila, što zahteva neprekidnu potrošnju energije za cirkulaciju. Po drugom principu mastilo se aktivira samo po potrebi. Ink – jet štampači koji realizuju prvi princip rada nisu prihvaćeni na tržištu. Postoje, međjutim dve konstrukcije štampača sa mastilom koji realizuju drugi princip rada: bubblejet i piezoelektrični štampači. Bubblejet štampači rade pomoću mehurića od mastila, a mehuriće obezbeđuje grejno telo u glavi štampača koje greje mastilo na

300 °C. Usled visoke temperature i mehurića nastaje pritisak u glavi štampača koji izbacuje mastilo. Piezoelektrični ink – jet štampači obezbeđuju potreban pritisak za izbacivanje mastila iz glave pomoću sićušnih piezo – keramičkih ploča koji se iskrivljuju na primljeni električni impuls. Ink – jet štampači mogu dati brz, kvalitetan otisak, rade prilično tiho i u boji, ali su za sada troškovi štampanja po strani veliki.

Laserski štampači su danas najčešći delovi konfiguracije PC računara. Služe za štampanje cele strane u jednom prolazu. Valjak je prevučen poluprovodnikom od selena koji se pomoću laserskog zraka slabog intenziteta naelektriše. U naelektrisanim tačkama valjak privlači pigmente (obično crne) boje iz tonerske kutije, to predaje prijanjajućoj hartiji. Na hartiju se zapeku zalepljeni pigmenti na 200 °C. Razlaganje se kreće od 300 do 600 dpi, a brzina štampanja je od 4 do 20 listova u minutu. Postoje i laserski štampači u boji koji su za sada još preskupi za kućnu upotrebu.

Termički štampači spadaju u najjeftinije ili najskuplje uređaje za stvaranje tvrde kopije u zavisnosti od primenjene tehnologije stvaranja otisaka. Jeftini termički štampači imaju termoelemente u glavi i koriste specijalan papir koji je osetljiv na temperaturu. Otisak nastaje u procesu u kojem zagrejani termoelement glave dodiruje papir. Termoosetljivi papir nije dovoljno stabilan, brzo propada na uticaj svetlosti. Skuplji modeli termičkih štampača ne zahtevaju papir osetljiv na temperaturu, i otisak stvaraju zagrevanjem delova trake prevučene mešavinom mastila i specijalnog voska. Mešavina sa trake se topi usled zagrejanih termoelemenata glave i zalepi se za papir. Jeftiniji modeli su sa pokretnom glavom, a skuplji imaju nepokretnu glavu sastavljenu od više hiljada termoelemenata (otpornika). Primenu ih i u štampanju u boji (najpogodniji su za štampanje u četiri boje).

Sublimacioni štampači su u stvari termički štampači u kojima se otisak stvara pomoću čvrstog materijala koji se na uticaj temperature direktno pretvara u gas. Zbog specifičnog procesa štampanja i specijalnog, polimerizovanog papira, štampanje sublimacionim štampačem je izrazito skupo i neverovatno dugo traje. Istina otisak je prvoklasan, kao fotografija, a slika A4 formata u True Color verziji (dubina boje 24 bita, tj. sa 16,7 miliona nijansi) se radja za otprilike 10 minuta.

Ploteri su mašine za crtanje i imaju najveću primenu u kombinaciji sa CAD aplikacijama (softverima). Veličina ctreža je standardizovana (A4, A3, A2, A1 i A0) i prema tome se na jedan način i klasifikuju ploteri. U softveru za rukovanje radom plotera je moguće saopštiti koje su debljine pera prisutne u ploteru, a sam softver vrši optimalizaciju rada za što brže iscrtavanje crteža. Danas su ulogu plotera na dimenziji lista A4 u potpunosti preuzeli laserski štampači.

Zvučnici i slušalice su danas neophodni, pre svega omladini, pošto su multimedijalne aplikacije dominantne na tržištu, a tu su i muzičke CD – ploče koje su dosta omiljene naročito kod mlađeg sveta, o DVD pločama i da ne govorimo.

Modemi su uređaji koji omogućuju povezivanje dva računara preko postojećih PTT linija. Računar barata sa digitalnim signalima koji se jako izobličavaju na PTT linijama, zato ih treba pretvoriti u (modulisani) analogni signal na predaji (modulator), a na prijemu modulirani analogni signal pretvoriti u digitalni (demodulator). Pošto računar čas šalje čas prima podatke sa PTT linije (parice), modem obavlja sve aktivnosti zaštite (modulacije) signala pre slanja i pretvaranje (demodulacija) u binarni oblik nakon prijema signala odakle mu i ime potiče. Modulacija (zaštita) signala može biti amplitudna, frekvencijska i fazna. Zvanična oznaka modema je DCE (Data Circuit – Terminating Equipment). Postoje unutrašnji modemi koji se priključuju na PCI (ranije ISA) slot ili magistralu: to su u stvari modemske kartice koje se smeštaju u kućište računara. Spoljašnji modemi su uređaji koji su smešteni u posebno kućište i priključuju se za serijski port računara. Danas se najviše koriste modemi brzine 56 Kbit/s.

Mrežne kartice su nastale kao potreba da se pojedini računari na istoj geografskoj lokaciji povezuju. Povezivanje računara omogućuje zajedničko korišćenje istih, skupih hardverskih i softverskih resursa od strane više računara, istovremeno korišćenje zajedničke baze podataka, višestruko smeštanje (arhiviranje) Pojam topografija mreže označava fizički raspored računara na datoj geografskoj lokaciji, a topologija označava način povezivanja računara. Ožičene realizacije računarskih mreža se mogu podeliti u dve velike grupe: a) ostvarivanje veza između računara je tipa tačka – tačka (point to point topology) i b) veza između računara se ostvaruje distribucijom poruke svim računarima mreže (multipoint topology). Prva vrsta mreže se u praksi najčešće izvodi u obliku zvezde, prstena ili, eventualno stabla. Kod prve vrste mreže računar koji dobije poruku kontroliše da li je ona njemu adresirana, ako jeste on je obradi, ako nije, prosledjuje sledećem računaru koji je sa njim u fizičkoj vezi. Kod druge vrste (multipoint) mreže se najčešće primenjuje sistem sabirnice ili prstena Kod ove, druge vrste topologije mreže svi su računari priključeni na isti kanal za slanje poruke, pa poslatu poruku primaju svi računari, kontrolišu adresu i poruku obradjuje samo adresirani računar, ili u slučaju grupne adrese (multicasting) svi računari adresirane grupe. Fizičko povezivanje računara u mrežu bez obzira na topologiju se vrši pomoću tri vrste provodnika: a) upredenog voda (sa zaštitom – STP tj. Shielded Twisted Pair, ili bez zaštite – UTP, tj. Unshielded Twisted Pair), b) koaksijalnog kabla i c) optičkog kabla. Mrežne kartice omogućuju priključak na realizovanu fizičku mrežu. Za LAN (Local Area Network) mreže se koriste UTP vodovi ili koaksijalni kablovi. Najnovije mrežne kartice imaju priključak i za UTP vod (oznaka je RJ-45) i za tanki koaksijalni kabl (oznaka je

BNC – Bayonet-Neil-Counselman). Za premošćavanje većih udaljenosti se primenjuju debeli koaksijalni kablovi, koji imaju manje slabljenje, no oni se direktno retko priključuju na računar. Mrežne kartice ostvaruju samo fizički priključak, a ostalo realizuje odgovarajući softver za komunikaciju.

5.1.7. Komunikacije

Komunikacije predstavljaju oblast u kojoj su hardver i softver podjednako prisutni i ne mogu jedan bez drugog. Savremena komunikacija traži velike brzine prenosa informacija i veliku širinu opsega za prenos. Pošto se brzina prenosa ne može beskonačno povećavati, veći prenos se ostvaruje povećanjem širine opsega. Generalno mreža velike brzine prenosa (velike širine opsega) može da stigne do stanova preko provodnika ili bežičnim putem. Svi stanovi se napajaju električnom energijom, i to je upravo prvi medijum koji može da se takmiči za primat u realizaciji komunikacione mreže. Drugi mogući medijum je telefonska mreža, a treći mreža kablovske televizije. Za bežične komunikacije prenosni medijum je vazduh.

Komunikacija preko energetske vodove (PLC – PowerLine Communication) je najviše izložena smetnjama, međutim, u zadnje vreme su u Nemačkoj postigli izuzetno dobre performanse prenosa sa najnovijom generacijom interfejsa za energetske vodove. Brzina prenosa je 2 Mbit/s, što predstavlja solidnu brzinu za pristup Internetu. Adapteri (interfejsi) PLC komunikacije bi se mogli smestiti u obične konektore za struju (ili još bolje u napojnu jedinicu samog računara) i time bi bio omogućen pristup za prenos podataka preko energetske mreže.

Telefonska mreža je realizovana u cilju prenosa govora, no ona se danas sve češće koristi za prenos podataka, jer je znatno sposobnija od onog za šta su je prvobitno predvideli. Standardizacija ISDN (Integrated Services Digital Network) usluga preko PTT parica je uradjena 1984. godine, i ona kao osnovnu uslugu obezbeđuje dva kanala brzine 64 Kbit/s za prenos. Ako se parica koristi samo za digitalni prenos signala (tj. ne služi istovremeno i za telefoniranje) onda se zove digitalna pretplatnička linija, u oznaci DSL (Digital Subscriber Line), i u svetu ima puno varijanti DSL – a . Najpoznatija vrsta DSL – a se označava sa ADSL, a radi se o asimetričnom DSL – u, sa brzinom slanja od 9 Mbit/s i primanja od 1 Mbit/s u okolini od 5 km od telefonske centrale. Najbrža varijanta je VDSL (Very high-rate DSL) kod koje je brzina slanja 25 Mbit/s, a primanja 3 Mbit/s, ako telefonska centrala nije udaljenija od 300 m. DSL usluge još nisu rasprostranjene kod nas, ali se očekuje njihov pohod.

Koaksijalni kabel kablovske televizije je simpatičan medijum i svakako ozbiljnije deluje od proste parice telefonskog kabla. Problemi sa kablovskom mrežom su sledeći: pojačavači su “jednosmerni” i svi pretplatnici dobijaju program sa istog kabla. Za ciljeve pružanja Internet usluga “jednosmernost” mreže treba ukloniti i mrežu pretvoriti u zvezdastu, tj. da svaki pretplatnik bude direktno vezan za centar. U pogodnim tačkama (da ne bi trebalo vući nekoliko hiljada bakarnih parica) treba obezbediti prelazak na optički kabel, što svakako poskupljuje rešenje. Problem sa ovakvom mrežom je i taj, da se ne može garantovati širina prenosa, jer ona zavisi od broja aktivnih korisnika koji u tom trenutku komuniciraju. Očekivane brzine slanja su 0,5 – 2,5 Mbit/s, a primanja 1,5 do 3 Mbit/s.

Prenos podataka preko satelita je sledeći po brzini, a obavlja se preko vazdušnog medijuma. Radna frekvencija je reda veličine gigaherca ($10^9 - 10^{10}$ Hz). Za obične građane bežičnim putem, preko satelita se vrši samo primanje informacija, jer je cena predajnika za sada previsoka. U 2000. toj godini je krenuo eksperiment u cilju zamene skupih komunikacionih satelita sa malim i jeftinim satelitima. Kompanija Aerospace Corp. je lansirala »picosat« mase 20 dkg, dimenzije 10x8x2,5 cm. Predmetni satelit je ispunio očekivanja, međutim, stručnjaci smatraju da će budućnost komunikacija obezbediti nešto veći sateliti, tzv. »nanosat« – i u oblasti prenosa podataka preko vazdušnog medijuma.

Optički kablovi rade na frekvenciji koja je grubo stohiljada puta veća od radne frekvencije uređaja za bežični prenos, odnosno iznosi $10^{14} - 10^{15}$ Hz. Pošto je na većim frekvencijama moguće prenositi veću količinu podataka, optički kabel ima primat u brzini prenosa. Problemi nastaju pri konverziji električnih signala u optički: teško je i skupo obezbediti odgovarajuća električna kola tako velike brzine za pretvaranje električnih u svetlosni signal u cilju prenosa preko optičkog kabla.. Upravo je zato značajan proizvod Lucent Lambda Wavestar, pretvarač električnih u optički signal koji je 25 % jeftiniji od prosečne cene sličnih konvertora. Zadnje godine prošlog, dvadesetog veka patentiran je novi pretvarač nazvan opto – čip koji je razvijen na univerzitetima u Vašingtonu i Južnoj Karolini koji ima brzinu pretvaranja signala 100 Gbyte/s. Najbrži prenos podataka, odnosno najbržu komunikaciju do sada je ostvarila kompanija Lucent Bell na optičkom kablju dužine 300 km, a brzina je 3,28 Tbit/s (terabita u sekundi). Ta brzina je ostvarena frekvencijskim multipleksom na 82 različite frekvencije, na svakoj frekvenciji brzina je bila 40 Gbit/s.

Pošto je komunikacija izmedju računara kompleksan problem, zato se po preporuci ISO standarda (naziv preporuke je OSI – Open System Interconnect), komunikacija ili njena realizacija se deli na nivoe, slojeve (layer) po OSI referentnom modelu (arhitekturi). U okviru

komunikacije između dva računara u stvari podatke razmenjuju pojedini slojevi istog nivoa na dva računara koji razmenjuju podatke. Slojevi po OSI arhitekturi su sledeći:

1. Fizički sloj
2. Sloj linka podataka
3. Mrežni sloj
4. Transportni sloj
5. Sloj sesije
6. Prezentacioni sloj i
7. Aplikacioni sloj

Fizički sloj brine o bit – serijskom prenosu (signalizacija, modulacija i sinhronizacija bita) i u preporukama su propisi vezani za električne i mehaničke osobine fizičkog interfejsa za realizaciju prenosa bitova kroz provodnike. Sloj linka podataka OSI preporuka sadrži proceduralne i funkcionalne preporuke za uspostavljanje, održavanje i raskid veze između dva računara u mreži. Mrežni sloj, po OSI standardima, obavlja pristup računara mreži, kontrolu greške i protoka podataka i za rutiranje poruke do odredišta. Transportni sloj vrši multipleksiranje, segmentiranje i skupljanje dugačkih poruka u kraće mrežne poruke i obezbeđuje mrežne adrese za niži, mrežni sloj. Sloj sesije vrši kontrolu dijaloga i obezbeđuje ispitne tačke (check point) koje mogu poslužiti kao polazna osnova za nastavak sesije posle neočekivanog raskida veze (sesije). Prezentacioni sloj obezbeđuje premošćavanje razlika u prikazu informacija između aplikacija na različitim računarima. Ovaj sloj vrši transformaciju podataka (konverziju kodova, adaptaciju slike), strukturiranje, enkripciju i deskripciju, kao i kompresiju i dekompresiju podataka. Aplikacioni sloj brine o korisniku, njegovom uređaju i aplikaciji (kontrola štampača, ekrana računara itd.).

Pri prenosu analognih signala preko fizičkog medijuma širina opsega je najvažnija odlika, a u slučaju digitalnog prenosa bitna je brzina prenosa, tj. broj prenešenih bitova u jedinici vremena (bit/s). Često se za opis digitalnog prenosa koristi jedinica baud, što označava broj prenešenih promena u jedinici vremena. U cilju povećanja brzine prenosa podataka vrši se podela postojećih provodnika za prenos. Jedna mogućnost podele je multipleksiranje ili istovremeno prenošenje više veza (komunikacionih kanala) kroz isti fizički medijum. Poznata su frekvencijsko i vremensko multipleksiranje. Kod vremenskog multipleksa »fizički« kanal veće širine opsega se vremenski deli na više »vremenskih« kanala, a frekvencijski multipleks veću širinu opsega fizičkog kanala deli na više frekvencijskih kanala na taj način da se prenos realizuje pomoću simultano emitovanih informacija modulisanih na različite noseće frekvencije.

Frekvencijsko multipleksiranje se koristi pre svega za analogne signale, a vremenski multipleks za digitalne. Moguće je prenosni medijum bolje iskoristiti i bez multipleksiranja. Podelom poruke na manje delove, pakete, se može obezbediti prenos poruke u više delova sa posebnim rutiranjem pojedinog paketa. Smer protoka podataka (ili bitova) kroz provodnike može biti različit, shodno tome postoje sledeće tri vrste prenosa podataka:

1. Jednosmerni (simplex),
2. Naizmenično dvosmerni ili poludupleks (half-duplex) i
3. Pravi dvosmerni (full-duplex).

Po mišljenju mnogih stručnjaka i za prenos podataka preko optičkog kabla važi Moore – ov zakon, tj. da se brzina komunikacije udvostručuje svakih osamnaest meseci.

5.2. SOFTVER

Poznata je činjenica da je razvoj računarske tehnike u velikom zaostatku ako ga uporedimo sa razvojem komunikacija u zadnjih dvadeset – trideset godina. U sedamdesetim godinama brzi računar je izvršavao instrukcije za 100 ns (u proseku), danas to čini za 1 ns. Odnos brzine je dakle 100, što je zavidno ubrzanje u obradi. Uporedimo li brzine digitalnog prenosa (komunikacije) za isto doba (u sedamdesetim godinama brzina je bila 56 kbit/s, danas je 10 Gbit/s) dobijemo odnos stopetdesethiljada! Ako uporedimo brzine prenosa kroz optički medijum (1985. godine – 50 Mbit/s, 1998. godine 10 Gbit/s) odnos je svega 200, no tu se već oseća ograničenje usled sporosti elektronike za pretvaranje električnog u optički signal (Lucent Bell je komunicirao u 2000. godini preko optičkog kabla brzinom od 3,2 Tbit/s).

Neki stručnjaci smatraju da je raskorak između razvijenosti softvera i hardvera veći od zaostatka brzine hardvera od komunikacija.

Razvoj softvera je omogućio prodor i primenu računara u svim oblastima ljudske aktivnosti. Današnji softveri su sve kompleksniji, traže brže procesore, operativne memorije većeg kapaciteta, ogroman disk prostor, kvalitetne grafičke kartice, itd. Konkurentska borba za primat u pojedinim oblastima tera softverske kuće da što pre objave svoje proizvode, što negativno utiče na njihov kvalitet. Testiranja softvera se ne mogu temeljito izvršiti, pa su bagovi prisutni u svim realizacijama. Tendencija usložnjavanja je primetna u svim oblastima primene softvera, a postoji istovremena težnja za uprošćenjem korisničkog interfejsa u cilju omogućavanja lakše i jednostavnije primene. Jedino mobilni uređaji vape za

pojednostavljenijim softverima koji traže manji prostor na eksternoj memoriji, slabiji procesor i manju memoriju, jer oni oskudevaju u svim resursima.

5.2.1. Operativni sistemi

Ako se zadržimo na klasi PC i mobilnih računara onda su najčešće korišćeni operativni sistemi Windows, Windows NT, Linux, OS2/Warp i Solaris za PC računare i Windows CE (Windows for Pocket PC), Epoc, BeOS, Linux i PalmOS za mobilne računare. Najrasprostranjeniji mrežni operativni sistem za personalne računare je od kraja osamdesetih godina Novell NetWare.

Tendencija stalnog obogaćenja usluga koje nude operativni sistemi je permanentno prisutna, tako da su današnje verzije operativnih sistema za PC računare sposobne za upravljanje mrežom, tj. podržavaju mrežno okruženje rada (povezivanje računara u mrežu). Windows je prvobitno projektovan za šestnaestobitne procesore, a tek posle je preradjen i sada važi kao 32 – bitni operativni sistem. Windows NT je, kako neki kažu Microsoftov prvi pravi 32 – bitni OS i time se objašnjava njegova stabilnost. Linux je najveći konkurent Windowsu (i NT – u) na polju operativnih sistema, prednost mu je u ceni, brzini, složenosti, stabilnosti i skalabilnosti, a za sada jedini ozbiljan nedostatak je u broju postojećih aplikacija razvijenih za njega.

Teško je proceniti rasprostranjenost operativnih sistema na mobilnim računarima. Dva lidera na tom polju su Windows CE (i njegova poboljšana verzija Windows for Pocket PC) i Epoc. Funkcionisanje Windows CE OS – a je skoro identična kao starijeg i većeg brata, a upravo je to i bio cilj kod njegovog razvoja. Poboljšanja u brzini rada, kao i modifikacije u grafičkim elementima su prisutni u novoj verziji za Pocket PC – je. Epoc je stabilan operativni sistem specijalno projektovan za mobilne aparate. Računari kompanije Psion su snabdeveni ovim OS – om, ali Epoc se primenjuje i u drugim mobilnim uređajima, kao što su telefoni (smartphone) i komunikatori. Mada se Epoc razlikuje od Microsoftovog operativnog sistema za mobilne računare saradnja između njih je solidna (razmena podataka sa stonim PC računarima se obavlja brzo i efikasno).

U razvoju mrežnih operativnih sistema Novell je (u Novell NetWare 4.0) 1994. godine lansirao upotrebu direktorijumskog servisa NDS (NetWare Directory Service, kasnije nazvanog kao Novell Directory Service), koji je omogućio administriranje svih mogućih delova (objekata) u (lokalnoj) računarskoj mreži (serveri, štampači, diskovi itd.). Kasnije je NDS “izrastao” iz Novell NetWare mrežnog operativnog sistema, tačnije napisan (prilagodjen) i za druge mrežne

OS – e. Na kraju 2000. godine direktorijumski servis (NDS) je funkcionisao na sledećim OS – ima: NetWare, Windows NT, Windows 2000, Linux, Solaris 6, 7 i 8. U međuvremenu je NDS prilagodjen opšte prihvaćenom novom transportnom protokolu LDAP (Lightweight Directory Access Protocol) koji je koristio TCP/IP, a i podeljen na dve oblasti: eDirectory za primene na Internetu i Corporate Edition koji označava klasični direktorijumski servis. To je upravo oblast (NDS) koja je u početku, a delimično i sada pripada i operativnom sistemu i uslužnim ili pomoćnim programima.

5.2.2. Uslužni programi

Pojam uslužnog programa pokriva sve one softverske proizvode koji realizuju potrebne i korisne aktivnosti, a koje operativni sistemi ne obezbeđuju. Lako je bilo nabrajati takve (uslužne) programe na početku masovnijeg korišćenja PC računara. U to doba je DOS bio operativni sistem, a bilo je svega dva poznata uslužna programa: PC Tools i Norton Utilities. Danas je riskantno izjaviti koji je program uslužni iz više razloga. Prvo postoje različita mišljenja o samom pojmu uslužnog programa, a drugo, neki softveri na jednom OS – u predstavljaju uslužne programe, a na drugom OS – u su ugrađeni kao funkcija operativnog sistema. Banalni primer za to su softveri za zaštitu od napada virusa i za otklanjanje virusa.

Osvajanje tržišta koje se interesuje za e – commerce rešenja predstavlja cilj za vodeće i eminentne softverske kuće danas, jer po mnogim mišljenjima to je ključ opstanka na tržištu. Takva rešenja nude mnogi, medju njima su IBM, Compaq, Sun, Oracle, Netscape, Computer Associates, Novell, itd. Interesantno je kako je Novell izgradio svoju strategiju. Prvo je smislio direktorijumski servis (NDS) kao funkciju NetWare – a, kasnije je pripremio taj servis za druge mrežne OS – e, još kasnije ga je podelio na dva smera (eDirectory i Corporate Edition), nadogradio svoje usluge i na kraju je sve to obuhvatio u celinu pod nazivom One Net. Ova vizija predstavlja grupu uslužnih servisa NSS (Net Services Software): Net Management Services (upravlja objektima – korisnici, uređaji, aplikacije, itd.), Net Content Services (usluge u vezi raspodele i memorisanja informacija na mrežama) i Net Portal Services (servis za identifikaciju i e – commerce). Pojednim servisima pripadaju razni produkti od kojih je jedan od najpoznatijih ZENworks (Zero Effort Networks). ZENworks obavlja administraciju svih radnih stanica i svih servera na mreži u vezi hardverske konfiguracije i instaliranog softvera, a ima i ugrađenu zaštitu od virusa.

5.2.3. Aplikativni programi

Aplikativni programi se pripremaju pomoću programskih jezika. Njih ima danas dosta na tržištu i svi su sposobni za rukovanje bazama podataka. Savremeni programski jezici su objektno orijentisani i mogu raditi u file server, client/server ili multi – tier arhitekturi (okruženju). Svaka aplikacija koja radi sa bazom podataka ima tri razgovetne funkcionalne celine

1. Fizička obrada podataka (Data processing),
2. Realizacija poslovne logike (Business logic) i
3. Kontakt sa korisnikom (User Interface).

Neposredna (fizička) obrada podataka podrazumeva otvaranje, zatvaranje i indeksiranje datoteka, upis novih, modifikacija i brisanje postojećih podataka, rešavanje konfliktnih situacija pri istovremenom pristupu, itd.

Poslovna logika obezbeđuje pravilno funkcionisanje aplikacije: brine o zaštiti podataka (kontrolom i zabranom pristupa za neovlašćene korisnike), o kontroli integriteta podataka (*integriteta domena* – tip podatka, dužina podatka i uslov ; *integriteta entiteta* koji je vezan za pojam ključa – jedinstvenost ključa: kompletan ključ ili delovi ključa ne smeju imati nepoznatu vrednost, tj. da su NOT NULL ; *referencijalnog integriteta* – koji služi kao međjurelaciono ograničenje – primer: ne može se uneti ocena konkretnog studenta u tabelu OCENA, ako se on ne nalazi u tabeli STUDENT), o grupisanju operacija u obradi, o upravljanju transakcione obrade, itd.

Savremene aplikacije komuniciraju sa korisnikom na specifičan, ali za nas već uobičajen način: preko prozora. Korisnički interfejs (meniji, maske, help – prozori, itd.) mora biti razumljiv, lako shvatljiv, pregledan i simpatičan.

U file server arhitekturi se sve funkcije realizuju na jednom računaru u slučaju da nema potrebe za višekorisničkim radom. File server arhitektura na mreži računara u slučaju korišćenja iste aplikacije od strane više korisnika pokazuje ozbiljne nedostatke: svaki korisnik u toku obrade traži i dobije podatke preko mreže sa servera i na sopstvenom računaru ih obrađuje. Količina podataka za obradu može biti i pozamašna za više korisnika i tako mreža može lako postati usko grlo u obradi. Tu arhitekturu koriste dBase, FoxPro, Clipper, Paradox, Visual Basic, Access, itd. Često se ova arhitektura zove i jednoslojna (single – tier).

U client/server arhitekturi se aplikacija deli na dva dela, i ponekad se ta arhitektura zove dvoslojna (two – tier). Na serveru postoji softver koji obavlja sve poslove u vezi održavanja baze

podataka, kao i većinu poslovne logike (deo poslovne logike kao što su kontrola integriteta podataka, i transakciona obrada) pomoću triggera i memorisanih procedura. Taj softver se često naziva server podataka ili server baze podataka (database server). Na računaru korisnika se implementira preostali deo poslovne logike, i kompletna problematika razmene podataka ili interakcije sa korisnikom i to najčešće u grafičkoj formi – GUI (Graphical User Interface). Obrada se u potpunosti obavlja na serveru, a preko mreže putuju samo zahtevi korisnika formulisani na SQL (Structured Query Language) jeziku u jednom (od korisnika prema serveru) i rezultati obrade kao odgovori na zahteve u drugom (od servera prema korisniku) smeru. Na prethodno opisan način rade sledeći serveri podataka: Microsoft SQL Server, Oracle, Informix, Sybase, MySQL, InterBase, itd.

Višeslojna (multi – tier) arhitektura obrade podataka pretpostavlja podelu aplikacije na više delova. Najčešća višeslojna arhitektura obrade je ona najprostija, sa tri sloja. Tu se na serveru nalaze podaci i softver za održavanje baze podataka, na srednjem nivou (middle – tier) programi koji ostvaruju poslovnu logiku (tačnije deo poslovne logike koji ne može da se ugradi u logičku strukturu podataka, tj. u održavanje baze podataka), a na strani korisnika (klijenta) su samo programi za realizaciju GUI – a (zato se često takav klijent zove “mršav” – thin client).

Microsoft je sredinom 1999. godine lansirao svoju novu .Net ideju za razvoj aplikacija na svim svojim programskim jezicima koji su obuhvaćeni u grupi Visual Studio (Visual Basic, Visual C ++, Visual InterDev, Visual J++, Visual FoxPro). Prva verzija razvojne platforme zamišljene .Net tehnologije se pojavila na tržištu u julu 2000. godine pod imenom .Net Framework. .Net Framework treba da bude temelj za buduće verzije Visual Studio.NET porodice. Današnji efikasni programski jezici kao što je Visual C++ traže širok dijapazon pažnje od programera za menedžiranje sitnih detalja (“ručno” oslobađanje memorije i ostalih resursa sistema), što ruši nivo efikasnosti u programiranju. Nova .Net platforma obezbeđuje stabilnost, brzo funkcionisanje programa i kratko vreme implementacije programa (programiranje samo poslovne logike).

Prva novina je ta, da .Net Framework obezbeđuje zajednički runtime modul za sve programe Visual Studio – a, smanjujući time razlike između programskih jezika i stvaranjem jednog zajedničkog (runtime) jezgra za aplikacije. Druga je novina u unifikaciji programskih klasa stvaranjem zajedničke biblioteke ili zajedničkog API (Application Programming Interface – zajedničko ime rutina koje se mogu pozivati iz Windows aplikacija) sloja u svim programskim jezicima. To će omogućiti “prelaznost” ili nasledjivanje objekata između programskih jezika, stvaranje zajedničke mehanizme za upravljanje greškama, zajednički debugging, itd. Treća novina .NET Frameworka je ASP+ (Active Server Pages+) koja obezbeđuje okruženje za brzo i

jednostavno ostvarenje (implementaciju) Web aplikacija. COM (Component Object Model) koncept projektovanja ostaje i dalje aktuelan za razvoj aplikacija sa velikim stepenom automatizacije, jer .NET Framework preuzima brigu od programera u vezi registracije, opisa interfejsa i referenciranja komponenti. COM je inače standard koji propisuje način definisanja interfejsa i metoda objekata za zajedničko korišćenje. Mrežno korišćenje COM objekata obezbeđuje DCOM (Distributed COM). Microsoft predviđa uvođenje COM+ tehnologije koja bi obuhvatila COM, MTS (Microsoft Transaction Server) i DCOM, čime bi postao glavni servis na srednjem sloju (middle – tier) arhitekture aplikacije.

Nova .NET tehnologija za izradu softvera kompanije Microsoft vredna je pažnje jer se u nju ulažu veliki napor i velika novčana sredstva, a želimo li priznati, ili ne, još uvek je Microsoft taj koji kroji budućnost i diktira tempo u razvoju softvera.

6. RAZVOJ INFORMACIONOG SISTEMA

U prethodnim izlaganjima je istaknuto da informacioni sistem spada u porodicu inhomogenih sistema, tj. da se sastoji od raznorodnih činilaca ili elemenata. Otkrivanje interaktivnih veza između raznorodnih činilaca obično predstavlja veliku poteškoću, i zbog toga se tačke gledišta sa kojih se prilazi informacionom sistemu menjaju: neki (pogrešno) sredstva za obradu (tehnička sredstva) smatraju primarnim elementima, drugi (takodje greše, zbog varijabilnih zahteva korisnika) procedure koje treba realizovati, a treći (s pravom, na današnjem nivou naših saznanja) tvrde da podaci, odnosno pojmovi predstavljaju primarne činioce.

Sa druge strane poznato nam je da je informacioni sistem sekundarni, što se izvornosti tiče, iza primarnog sistema fizičke realnosti. Informacioni sistem, naime samo odslikava stvarnost, a slika stvarnosti (informacioni sistem) kao bilo koja druga slika može biti i višestruka. Šta više slika o stvarnosti (koja je možda i jedinstvena – jednostruka) može imati različite pojavne oblike (saznanja u nečijoj glavi, na izvornim dokumentima i na računaru, recimo). Tako se, prilikom realizacije informacionog sistema, javlja problem uopštavanja, tj. apstrakcije i preslikavanja iz realnog u informacioni sistem.

Razvoj informacionog sistema podrazumeva primenu raznih konkretnih principa, metoda, tehnika, potupaka i sredstava u toku perioda razvoja sistema, ili životnog ciklusa informacionog sistema. Pojmovi životni ciklus informacionog sistema i razvoj informacionog sistema se naravno razlikuju. Njihovo naizmenično korišćenje može da izazove rasprave na teoretskom i praktičnom polju primene informacionih sistema. Da bi ukazali na povod i razloge izjednačavanja ova dva pojma na početku ovog pasusa, krenimo sa objašnjenjem šta se podrazumeva pod ovim pojmovima. Životni ciklus informacionog sistema se sastoji iz tri faze:

1. izgradnja,
2. funkcionisanje i
3. prilagođavanje novim uslovima.

Drugi pojam, razvoj informacionih sistema obuhvata sledeće aktivnosti:

1. analiza realnog sistema,
2. projektovanje informacionog sistema,
3. realizacija sistema – programiranje,
4. funkcionisanje – nadzor,
5. prilagođavanje novim potrebama – change management.

Pošto izgradnja (prva faza životnog ciklusa informacionog sistema) u stvari pokriva analizu, projektovanje i realizaciju kao aktivnosti razvoja, ekvivalentnost pojmova je očigledna. Postavlja

se samo još jedno pitanje teoretskog karaktera: da li se funkcionisanje sistema može smatrati razvojem? Odgovor je svakako negativan u slučaju da se posmatra samo funkcionisanje, međutim, ako se istovremeno sa funkcionisanjem vrši i nadzor samog funkcionisanja, onda je to aktivnost razvoja.

Pod pojmom paradigme (principa) razvoja informacionog sistema se podrazumevaju takvi opšte važeći principi, i načini prilaza i razmišljanja o razvoju tj. analizi, projektovanju i realizaciji sistema koji omogućuju pogodno uopštavanje i verno preslikavanje odgovarajućih svojstva i zakonitosti realnog sistema u informacioni sistem.

Korisnik – naručilac traži visok kvalitet u toku izrade (korisna, aktuelna dokumentacija, kvalitetni radni sastanci razvojnog tima i naručilaca), funkcionisanja (da informacioni sistem obavlja zacrtane funkcije striktno po formulisanju zahteva korisnika, da funkcioniše besprekorno i da bude user friendly, odnosno lak za korišćenje, razumljiv i da pomoću on line help – a pomaže rad korisnika) i održavanju (naročito da bude lako prilagodljiv novim uslovima, tj. da change management traje što je moguće kraće). Sa tog aspekta se može pratiti istorijski razvoj same tehnologije razvoja informacionih sistema, što je opisano sa više detalja u daljem tekstu u ovoj glavi.

U prvoj tački ovog poglavlja biće reči o tehnologiji razvoja informacionih sistema od pojave prvih računara do danas, sa dodatnim osvrtom na neke predikcije u vezi tehnologije razvoja sistema u budućnosti. Nezahvalno je dati predviđanja naročito u ovoj, najdinamičnijoj naučno – tehničkoj oblasti, jer, kako G.Winberg kaže: najveći problem u vezi predviđanja budućnosti je taj, što ništa nije ni nalik na ono što je za nas danas poznato.¹ Zato su date samo grube smernice moguće evolucije razvoja informacionog sistema.

Razvoj sistema može biti brz i efikasan kada svi članovi razvojnog tima imaju identičnu i čistu sliku o tome šta je uopšte informacioni sistem, koji su njegovi elementi, kako i u kom redosledu se oni otkrivaju, kakav je redosled aktivnosti pri razvoju sistema, kako se odnose analiza i projektovanje informacionog sistema, kako se predstavlja (dokumentuje) sistem, koji su kriterijumi po kojima se vrši kontrola kvaliteta i na koji način. O prethodno nabrajanim stvarima će biti reči u preostalom delu ovog poglavlja.

Druga tačka opisuje opšte principe razvoja koje treba ispoštovati u vezi planiranja, projektovanja i analize, o načinu podele sistema na manje delove (u cilju smanjivanja kompleksnosti) i o iterativnom postupku razvoja. Biće reči i o tome, da pri razvoju sistema neophodna moć apstrakcije od strane analitičara pri izgradnje modela sistema, a da se ni u jednom trenutku u toku razvoja ne sme zaboraviti na činjenicu, da se informacioni sistem

¹ Izvor: [22.], strana: 674.

priprema za korisnika kome treba predati istestirani (kontrolisani) bogato dokumentovan sistem sa standardnom terminologijom.

U okviru treće tačke ovog poglavlja ukazaćemo na razliku između pojmova metod i metodologija razvoja informacionih sistema. Ta dva pojma se u informatici koriste (pogrešno) kao sinonimi. Metodologija je metod nad metodama, jer ispituje strukturu i način izgradnje pojedinih metoda.

6.1. TEHNOLOGIJA RAZVOJA

Na početku računarske ere pojam softvera je obuhvatao operativne sisteme, programske jezike, sisteme za rukovanje podacima (metode pristupa za nekoliko vrsta organizacije podataka) i pomoćne (uslužne) programe. Proizvodnja softvera je bila povezana sa proizvodnjom hardvera, i računari su obavezno prodavani zajedno sa softverom. Softver je projektovan i realizovan samo za konkretan tip računara, a na drugim računarima nije mogao ni funkcionisati nijedan tada projektovan i realizovani softver. Informacioni sistemi su u to vreme bili razvijeni uglavnom samo za preduzeća sa najvećom akumulacijom: za banke i osiguravajuća društva. Razlog za to je jednostavan, druga preduzeća nisu bila dovoljno bogata i/ili kreditno sposobna da kupuju računare. Informacioni sistemi koji su u to doba razvijeni bili su “krojeni po meri”, odnosno striktno po zahtevima konkretnog kupca – naručioca računara u stilu “ključ za bravu”. Kasnije su preduzeća osnovala sopstvene timove za razvoj, funkcionisanje i održavanje informacionih sistema.

Proizvodnja računara u većim serijama, a naročito posle pojave mikroprocesora na početku sedamdesetih godina, je dovelo do pada cena hardverskih komponenata. Drastično povećanje broja korisnika računara je donelo i raznovrsne nove zahteve definisane od strane tih korisnika koji su vapili za novim programskim rešenjima. Za zadovoljavanje raznorodnih zahteva korisnika u sedamdesetim godinama su se pojavile softverske kuće nezavisne od proizvođača hardvera računara. Ti softverski centri su krenuli u razvoj takvih programskih proizvoda koji su bili dosta uopšteni (opšti) da su zadovoljili očekivanja velikog broj korisnika.

Inženjerski prilaz razvoju softvera je prvi put proklamovao Fritz Bauer još 1968. godine u Garmišu na konferenciji koju je organizovao Naučni komitet NATO – a. Od tada se koriste pojmovi kao što su tehnologija razvoja softvera (informacionog sistema) i softver inženjering. Time je istaknuto, a kasnije je i prihvaćeno i odmaćeno shvatanje, da je softverski proizvod

sličan, ili ga u toku razvoja (“proizvodnje”) treba tretirati kao bilo koji drugi “inženjerski” proizvod iz privrede.

“Teorijske osnove industrijskog razvoja softverskih proizvoda opisao je sredinom osamdesetih Brad Cox. On razvoj složenog sistema softverskog proizvoda definiše kao proces povezivanja manje složenih gotovih objekata i komponenti. Do ovakvog načina industrijskog razvoja dolazi se evolucijom iz zanatskog razvoja, koji i dalje postoji, ali prestaje biti dominantan način.”²

Softverski inženjering je razvojni proces u kojem se, uz strogu primenu inženjeringa, naučnih i matematičkih principa i metoda, vrši ekonomična proizvodnja kvalitetnog softvera u cilju efikasnog rešavanja problema iz realnog sistema pomoću računara.

U stručnoj literaturi se ponegde može naći klasifikacija softver inženjeringa sa tačke gledišta oblasti njegove primene, tj. od karaktera dela realnog sistema za razvoj, koja razlikuje:³

1. sistem inženjering (system engineering),
2. informacioni inženjering (information engineering) i
3. projektovanje baza znanja (knowledge engineering).

Kao dokaz da gornji pojmovi još nisu do kraja raščišćeni i generalno i semantički usaglašeni navešćemo karakteristike sistemskog inženjeringa, koje pokazuju da je sistemski inženjering globalniji pojam od softverskog inženjeringa:

“Opštiji pojam, od pojma softverski inženjering, je sistemski inženjering. On obuhvata aktivnosti specificiranja, projektovanja, primene, provere, instaliranja i održavanja sistema kao celine. Sistemski inženjering podrazumeva i niz pratećih komponenti koje nisu strogo informatičke, a koje omogućuju funkcionisanje sistema. Da bi se realizovao proces sistemskog inženjeringa kao niz interdisciplinarnih aktivnosti, potrebno je angažovati niz timova sa različitim znanjima i ulogama, koji će zajednički definisati sistem. Kada je sistem jednom definisan, ne preporučuje se ulazak u redizajn sistema bez preke potrebe, jer su sve kasnije izmene izuzetno komplikovane i skupe.”⁴

Softverski inženjering sa svojim “podvarijantama” pokriva projektovanje informacionog sistema u stilu industrijskog razvojnog procesa. Varijante softver inženjeringa po gornjoj klasifikaciji se razlikuju po kompleksnosti, načinu i tehnikama u izvršenju svojih zadataka, tj obradi podataka. Razvoj informacionih sistema je, međutim, faličan bez postojanja standarda. A čini se da je upravo razrada standarda predstavlja nerešiv problem za jednu takvu dinamičku oblast. Industrijski standardi, istina, postoje, ali njih diktira najčešće samo jedna softverska kuća,

² Dr Pere Tumbas: Softver inženjering, udžbenik u rukopisu pripremljen za štampu.

³ Izvor: [22.], strane: 157/158.

⁴ Dr Pere Tumbas: Softver inženjering, udžbenik u rukopisu pripremljen za štampu.

ili mala grupa proizvođača softvera u pojedinim oblastima primene na osnovu kojih bi trebalo doneti zvanične međunarodne standarde koje su do sada uvek neprihvatljivo kasno donošene.

U razvoju tehnologije razvoja informacionog sistema najveću pomoć je pružala računarska podrška, naročito u memorisanju i sredjivanju podataka nastalih u pojedinim koracima realizacije projektovanja informacionog sistema. Ti podaci se smeštaju u takozvanu razvojnu bazu podataka (repository), koja se do pre pet – šest godina zvala kao rečnik podataka (data dictionary). Repository može da rukuje više verzija projekta, i da brine o kontroli pristupa pojedinim verzijama projekta od strane pojedinih članova razvojnog tima po privilegijama. U zadnje vreme se pojavljuju softverski alati koji delimično ili u potpunosti potpomažu razvoj informacionih sistema i zovu se zajedničkim imenom CASE alati. Reč CASE skraćenica sa dva značenja: Computer – Aided Software Engineering, tj. inženjersko projektovanje softvera (i informacionih sistema) pomoću računara i Computer – Aided Systems Engineering ili inženjersko projektovanje sistema uz pomoć računara (računarska podrška inženjerskom projektovanju sistema).

Da provirimo sada malo u budućnost kroz predviđanja (proricanja) futurologa koji imaju svoja vidjenja u vezi tendencije pariteta hardvera i softvera, kao i pariteta ponude i potražnje softverskih produkata:

“Odnos cena i performansi računara smanjen je za proteklih 20 godina milion puta, a prema Murovom zakonu (Moor) predviđa se isti trend i za dva nastupajuća perioda od po 20 godina, nakon čega se očekuje da će doći do zasićenja i ovaj odnos će biti znatno umanjen. Kao posledice Murovog zakona, ustanovljeni su osnovni softverski zakoni, koji opisuju trendove razvoja softvera:

- softver se širi kao gas, sa stopom rasta od 33,9% godišnje,
- softver će se širiti sve dok ne bude ograničen Murovim zakonom,
- rast softvera omogućava održavanje Murovog zakona o rastu hardvera,
- rast softvera je ograničen jedino ljudskim ambicijama i mogućnostima i
- softver je "nepoderiv" proizvod, što omogućava njegovo korišćenje neograničeno dugo (iz čega je proizašao i čuveni problem 2000. godine, jer niko nije očekivao da će se softverska rešenja tako dugo održati u produkciji).

Najznačajnije posledice ovih zakona su da će razvoj softvera predstavljati najvažniju industrijsku granu i u 21. veku i da će se softverska kriza i dalje nastaviti.”⁵

⁵ Dr Pere Tumbas: Softver inženjering, udžbenik u rukopisu pripremljen za štampu.

6.2.PRINCIPI RAZVOJA INFORMACIONOG SISTEMA

U cilju efikasnijeg i bržeg razvoja informacionog sistema neminovno je da svi članovi razvojnog tima budu načisto sa principima razvoja sistema. Po velikom broju stručnjaka iz informatike principi razvoja informacionog sistema se mogu klasifikovati u tri grupe:

1. opšti principi,
2. principi u realizaciji (koraka) razvoja (često pominjani kao modeli razvoja) i
3. principi prilaza razvoju (u literaturi poznati kao filozofije razvoja).

U ovom kursu ćemo se ograničiti samo na opšte principe razvoja informacionog sistema, modeli i filozofije razvoja predstavljaju predmet naprednog kursa.

6.2.1. Opšti principi razvoja

U okviru opštih principa razvoja informacionih sistema potrebno je dogovoriti se u krugu razvojnog tima o opštim vodiljama, gledištima i normama, o načinu saradnje i donošenja odluka za vreme zajedničkog rada na razvoju sistema.

6.2.1.1. *Trostruko planiranje i tronivovsko projektovanje*

Razvoj informacionog sistema predstavlja projekat kojeg treba ostvariti, realizovati. Planiranje razvoja sistema je znači ekvivalentan sa planiranjem projekta, s tim, da razvoj informacionog sistema ima svoje specifične odlike, kao uostalom i projekti druge vrste. Pošto planiranje projekta spada u oblast projekt menedžmenta, planiranje razvoja informacionog sistema (kao kod planiranja bilo kojeg drugog projekta), shodno principima projekt menedžmenta obuhvata tri vrste plana:

1. globalni plan (plan),
2. detaljni plan (design) i
3. izvršni plan ili plan tempiranja poslova (schedule, project-plan).

Pošto projekt menedžment predstavlja posebnu naučnu oblast u fazi intenzivnog razvoja, sve pojedinosti u vezi ovih planova nema smisla iznositi na ovom mestu. One koji se interesuju u menedžiranju razvoja informacionog sistema upućujemo na odgovarajuću literaturu [3.] i [21.], a

tu ćemo se ograničiti na kratak opis ovih planova. Umešnost u izradi ovih planova (naročito globalnog plana), istina, može imati sudbonosnu ulogu kod prihvatanja ponude (ako je u pitanju raspisani konkurs ili tender) od strane naručioca.

Pre početka aktivnosti na razvoju informacionog sistema se vrši preliminarna i gruba procena samoga razvoja u okviru globalnog plana. Tu se opisuje cilj ili hijerarhija ciljeva, globalni sadržaj i gruba struktura budućeg informacionog sistema, hardversko – softverska okolina funkcionisanja, kao i očekivani troškovi i vreme razvoja sistema.

Detaljni plan se može sastaviti samo nakon temeljitijeg upoznavanja realnog sistema, za koji se razvija informacioni sistem. Na osnovu raspoloživih informacija o podacima, događajima i aktivnostima u realnom sistemu, moguće je dati tačniju procenu o svim potrebnim resursima: vremenu, novcu (uključujući dodatno potreban hardver, softver, komunikacije itd.). U prethodnoj rečenici naglasak je na reči “tačniju”, jer apsolutno tačna procena, naravno ne postoji. S tim u vezi treba napomenuti, da u informatičko razvijenijim sredinama sasvim je prihvaćen pojam kliznog planiranja (naročito što se vremena i novca tiče), u okviru kojeg se više puta vrši revidiranje potrebnog vremena i ukupnih troškova razvoja. Te revizije se priključuju detaljnom planu kao aneksi ili proširenja.

Projektom se danas smatra takav niz aktivnosti koji je usmeren na jedinstveni i kompleksan zadatak (za neki tim ili organizaciju) koji je ograničen vremenski i troškovno, i ima jasno definisani cilj koji treba da postigne.⁶ U okviru razvoja informacionog sistema treba predvideti odgovarajuće resurse i njih sinhronizovati, odnosno tempirati posao. Upravo to treba da sadrži izvršni plan razvoja informacionog sistema. U procesu izrade ovog trećeg plana ili plana tempiranja poslova moraju učestvovati stručnjaci raznih profila: ekonomisti, pravnici, informatičari i tehnička lica sa iskustvom u računarskoj tehnici i telekomunikacijama.

U procesu projektovanja razlikuju se tri vrste projekcije informacionog sistema: (a) podataka, (b) obrade i (c) okruženja i tri apstrakciona nivoa: (a) pojmovni – konceptualni, (b) logički i (c) fizički.

Projektovanje (design – process) ili proces projektovanja je niz aktivnosti u kojem se za odgovarajući informacioni sistem određuju elementi i u okviru kojeg se ti činioči (elementi) organizuju u tri projekcije i tri apstrakciona nivoa i, najzad, kada se ti elementi usaglašavaju.⁷

Projektovanje je samo deo razvoja informacionog sistema (tačnije deo izgradnje sistema), sam razvoj obuhvata i razne druge aktivnosti.

⁶ Izvor: [8.], strana: 14.

⁷ Izvor: [11.], strana: 122.

Projekat (design-product) je krajnji rezultat procesa projektovanja. Pored projekta postoje i drugi produkti razvoja kao što su računarski programi, tehnička i korisnička dokumentacija informacionog sistema itd.

6.2.1.2. Odnos analize i projektovanja

Odnos analize sistema i projektovanja informacionog sistema i dan danas izaziva polemike medju stručnjacima ove naučne grane, informatike. Mnogi stručnjaci tvrde da analiza nema nikakve veze sa projektovanjem. To potvrđuju razne metode i filozofije razvoja sistema u kojima su faza analize i faza projektovanja potpuno odvojene, i realizuju se i vremenski odvojeno: prvo analiza, a posle projektovanje. Na drugoj strani su stručnjaci koji smatraju da su analiza i projektovanje nerazdvojni. Razumljivo je da i ova druga strana ima svoje osnovane argumente: kod izbora najbolje od više alternativa u projektovanju neminovno je njihovo upoređenje (analiza). Znači li to da su svi u pravu? Neverovatno zvuči, ali je odgovor na prethodno pitanje potvrđan!

U toku razvoja informacionog sistema izvršavaju se zaista dve vrste analize, jer analiza može da predstavlja celu jednu fazu razvoja informacionog sistema, a može da bude samo deo jedne faze, jedna aktivnost (kod izbora alternative u fazi projektovanja).

Impuls za pokretanje razvoja može biti neki nastali problem u funkcionisanju informacionog sistema, a takodje i neka nova ideja za poboljšanje preslikavanja realnog sistema u informacioni sistem. U prvom slučaju vrši se analiza nastale situacije ili otkrivanje činjenica (fact finding) – uzroka problema, a u drugoj situaciji se vrši procena, vrednovanje (evaluation) nove ideje. Šta god da se desi, nakon impulsa za pokretanje razvoja nužno je tačno formulisati cilj ili ciljeve (hijerarhiju ciljeva) razvoja. Faza razvoja koja ispituje valjanost postavljenog cilja i mogućnost i svrsishodnost realizacije naziva se analiza sistema.

Analiza informacionog sistema se može izvršiti na dva nivoa:

1. na apstraktnom nivou i
2. na konkretnom nivou.

Na apstraktnom nivou se formulišu koraci analize koji se izvršavaju u fazi realizacije same analize na konkretnom nivou, a ti koraci na apstraktnom nivou su sledeći:

1. određivanje cilja, ili hijerarhije ciljeva informacionog sistema,
2. određivanje sadržaja sistema (sadržaj je skriven u pojmovima koji se koriste u delu realnog sveta za koji se priprema informacioni sistem – cilj i sadržaj su usko povezani),

3. određivanje strukture informacionog sistema (koji su elementi sistema i kako su međusobno povezani),
4. određivanje dinamike sistema, odnosno dinamike odnosa između elemenata, pri čemu se ne smeju ispustiti iz vida izuzeci u funkcionisanju ili dinamici, i
5. otkrivanje i obuhvatanje smernica planiranog razvoja dela realnog sveta u buduću informacioni sistem u okviru prethodna četiri koraka analize.

Analiza informacionog sistema na konkretnom nivou vrši otkrivanje ili upoznavanje realnog sistema na osnovu gore navedenih koraka na apstraktnom nivou.

Posle ove gore opisane faze razvoja, analiza i projektovanje informacionih sistema su nerazdvojno isprepleteni.

Prilaz sistemu prilikom analize i projektovanja je svojstven i različit. Opšte je prihvaćen stav da na početku razvoja, u fazi analize se polazi od celine i dekompozicijom se približava detaljima i elementima (informacionog) sistema. To je takozvani top – down način prilaza sistemu. Kod projektovanja je smer obrnut: od elementarnih činilaca se izgrađuje celina, tj. informacioni sistem. To se zove bottom – up način realizacije.

6.2.1.3. *Principi strukturiranja i iteracije*

Poznata je činjenica, da je za upoznavanje složenih stvari neophodna njihova podela na sastavne elemente, delove. Kod razvoja iole većeg sistema nemoguće je odjenom obuhvatiti sve stvari, tj. sve pojmove, događaje i aktivnosti koje su obuhvaćeni tim sistemom. Dokazano je, međutim, da način strukturiranja sistema, tj. njegova podela na delove, ima uticaja i na uspešnost funkcionisanja budućeg informacionog sistema kao celine. Principi strukturiranja sistema su sledeći:

1. geografski,
2. po sredstvima za obradu (po tehničkim sredstvima),
3. organizacioni,
4. funkcionalni i
5. po podacima.

Geografski princip strukturiranja označava višestruku realizaciju istih funkcija na različitim geografskim lokacijama i ne predstavlja prihvatljivu soluciju na polju podele sistema. Opasnost u geografskoj podeli leži u samom načinu realizacije: ako nije reč o distribuiranoj obradi sa različitih lokacija gubi se celina samog sistema.

Podela sistema po tehničkim sredstvima (po hardveru i softveru) nema opravdanja, jer se hardver i softver svakako češće menjaju nego sam informacioni sistem (ili bi barem tako trebalo da bude). Strukturiranje po toj podeli bi iziskivalo inoviranje podele (prestrukturiranje) celog sistema kod zamene računara i korišćenog softvera.

Organizacioni princip podele sistema ima svoje pozitivne argumente, ali relativna nepostojanost organizacionih celina (problemi u preduzeću se najčešće rešavaju reorganizacijom) ukazuje na manjkavost i nepodobnost u primeni ovog principa strukturiranja.

Danas zadnja dva principa predstavljaju osnovu za podelu sistema. Pod funkcijom treba podrazumevati izvršavanje odgovarajućih aktivnosti u vezi nekog zadatka. Pošto se funkcije mogu češće menjati od samih pojmova (podataka) u sistemu, tj. događaji i aktivnosti su nestabilniji, a svakako su zavisni od podataka, najcelishodnije je strukturiranje izvršiti po podacima.

Upoznavanje činjenica realnog sistema se vrši iterativno uz stalnu korekciju prethodno otkrivenih i dokumentovanih elemenata (činilaca) sistema. U analizi sistema se bazira na iterativnom upoznavanju fizičke realnosti u cilju što tačnijeg i preciznijeg upoznavanja. Nije to drugačije ni u fazi projektovanja: izradjen projekat, ili još bolje: više njegovih alternativa se analiziraju, i u jednom iterativnom postupku se stalno poboljšavaju.

6.2.1.4. *Ispoštovanje zahteva korisnika*

Pri razvoju informacionog sistema nikad ne treba izgubiti iz vida zbog čega i kome se sistem realizuje. Korisnik-naručilac ima svoje formulisane zahteve koje po mogućstvu treba ispoštovati. Bitno je da korisnik što češće bude pitan ili prisutan u procesu razvoja (uglavom pri projektovanju i realizaciji) da bi se izjasnio u vezi elemenata vezanih za njegove zahteve.

Naručilac ne može, ne zna definisati sve zahteve, jer ne poznaje sve činioce informacionog sistema, zato mu treba pružiti pomoć i predložiti alternative. Realizovani informacioni sistem treba da bude korisnički orijentisan (sistem menija i maske za unos da budu pregledni i laki za korišćenje, i da ima case sensitive help i pogodnu dokumentaciju), da bude tolerantan (sistem treba da dopušta korisniku da pogreši u toku korišćenja, a posle ili odmah ili kasnije da koriguje svoje greške), da bude pouzdan i performantan.

Članovi razvojnog tima, međjutim, moraju biti obazrivi sa zahtevima korisnika. Ne treba sve zahteve bezuslovno i bez prethodne analize prihvatiti. Razvoj ne treba da podržava takve zahteve koji vode do degradiranja performansi u funkcionisanju informacionog sistema. U stvari

zahtevi korisnika su često kontradiktorni i upravo odluke u toku razvoja treba da omoguće najbolji kompromis za budući sistem.

6.2.1.5. Modeliranje, apstrakcija i modularizacija

Projektovanje informacionog sistema, kao faza razvoja, odnosno izgradnje sistema se danas sve češće naziva modeliranjem. Svi analitičari i projektanti koji žele biti u trendu, ostavili su projektovanje i “prešli su” na modeliranje, jer je to sad moderno, a novi pojam je ubedljiviji od starijeg. Shodno tome se projekti nazivaju modelima, međutim, modelima nazvani projekti time nisu postali ništa bolji.

Elementi informacionih sistema kao što su podaci, informacioni događaji i aktivnosti su apstrakcije. Šta više, pri razvoju sistema se ti činioci formulišu i organizuju u tri projekcije i tri apstrakciona nivoa s tim da se opisuju kako uzroci tako i sama preslikavanja izmedju apstraktnih nivoa. Apstrakcija označava umešnost analitičara da otkriva jednu opštu crtu na osnovu više konkretnih, specifičnih⁸. Projektovanje se ne može zamisliti bez apstrakcije kao ni bez ponovnih primena dobrih rešenja (šablona) iz prethodnih projekata. Dobar informatičar izrađuje varijante (prototipove), analizira ih, upoređuje i na kraju vrednuje pojedine alternative.

Po novom značenju pojma modelom se mogu smatrati samo oni projekti informacionih sistema koji odgovaraju unapred zacrtanim kriterijuma kvaliteta. Samo modeliranje je proces izgradnje takvog projekta koji optimalno odgovara propisanim kriterijuma. Pri modeliranju, znači, je prisutna i analiza, jer treba uporediti izrađene varijante sa kriterijumima kvaliteta i, po potrebi, inicirati novi korak iteracije za poboljšanje alternativa. Svaki projekat nije automatski i model (ako ne odgovara kriterijumima), ali svaki model je, naravno projekat. O kriterijumima kvaliteta biće više reči u tački 6.3.4.

Pod pojmom modularizacije se podrazumeva strukturiranje sistema na manje celine, module i definiciju međusobnih veza tih celina u cilju kasnije, eventualno višestruke primene tih modula u istom ili drugim informacionim sistemima.

Komponente predstavljaju veće celine od modula i obuhvataju konkretne aktivnosti unutar neke funkcije, ili, pak, celu funkciju. Danas se sve više podržava (tačnije favorizuje) projektovanje na osnovu komponenata (elemenata) ranije razvijenih aplikacija. Trend je, za sada samo u razvoju i postoje mnogobrojni problemi u realizaciji takvog projektovanja zbog nekompatibilnosti različitih alata (na kojima su te komponente razvijene) i distribucije/korišćenja

⁸ In pluribus unum.

tih rešenja na različitim platformama. Za premošćavanje tih poteškoća su upravo uloženi naponi od strane softverske kuće Microsoft u vidu uvođenja nove koncepcije sa oznakom .Net Framework. O toj zajedničkoj razvojnoj filozofiji (sa zajedničkim API – Application Programming Interface – om, ASP+ i COM+ tehnologijom) je bilo reči u prethodnoj glavi. Mogućnost ponovnog korišćenja jednom izradjenih i testiranih komponenata bi znatno ubrzao izgradnju informacionih sistema. Više informacija o komponentnom razvoju se mogu naći u literaturi [Raffai], [Tumbas] i na Web adresi www.microsoft.com .

6.2.1.6. *Standardizacija, dokumentacija i kontrola kvaliteta*

U cilju koristoljivosti i prenosivosti realizovanog rešenja nužna je primena standardizovanih rešenja uz pomoć standardnih sredstava i tehnologija. Primedba se prvenstveno odnosi na izbor hardverskih komponenti, izbor mreže računara, protokola komunikacija i sistemskih softvera. Prethodno pomenuti zahtevi, kao što se vidi, odnose se pretežno na projekciju okruženja informacionog sistema i na način realizacije (programiranja) sistema.

O ulozi standarda je bilo već reči u prethodnim izlaganjima. Tu bi samo ukazali na nove momente u vezi pojma dokumentacija. Projekat informacionog sistema se priprema u pisanoj formi u obliku tekstova, listi, dijagrama i crteža što se obično zove dokumentacija. Projektna dokumentacija sadrži najbolju, krajnju varijantu projekta informacionog sistema, a poželjno je da joj se i ostale alternative razvoja projekta priključuju. Projektna dokumentacija nije ničija privatna stvar (privatna sfera). Ona služi za:

1. komunikaciju izmedju članova projektantskog (razvojnog) tima,
2. procenu produktivnosti članova razvojnog tima od strane rukovodioca,
3. informisanje (novih) korisnika (članova razvojnog tima), tj. i za komunikaciju sa naručiocem,
4. etalon,
5. daljnu ili ponovnu analizu rešenja,
6. revidiranje potrebnih resursa za razvoj sistema.

U ranijim razvojnim metodama su razdvojili fazu projektovanja od faze specifikacije. Shodno tome dokumentacija kao proizvod projektovanja je obuhvatila opšte, globalne činioce informacionog sistema, a specifikacija je sadržala detalje koji su se uglavnom odnosili na komunikaciju izmedju projektanta/analitičara i programera/dizajnera programa. U novije vreme zahvaljujući primeni baza podataka u obradi podataka, prethodno opisana podela činilaca na

globalne – važne i detaljne – manje važne ne smatra se modernom. Umesto toga se posmatraju sledeća tri pojma: zadatak, rešenje i sredstva (tj. pojmovni/konceptualni, logički i fizički nivo). Danas informacioni sistem se ne dokumentuje i specificira, već se za njega pripremaju odgovarajući projekti, i to: za projekciju podataka, obrade i okruženja. Shodno prirodi ove tri projekcije za okruženje se priprema samo jedan projekat, a za projekciju podataka i obrade tri posebna projekta: za pojmovni logički i fizički nivo (pojmovna, logička i fizička struktura podataka i obrade).

Dokumentacija pored činjenica treba da sadrži i uzroke datog oblika rešenja (zašto je rešenje baš takvo kako u dokumentaciji stoji). Između tri apstrakciona nivoa projekcije podataka i obrade postoje odgovarajuća preslikavanja. Dokumentacija treba da opiše zbog kojih razloga je došlo do izobličenja/preslikavanja između dva apstrakciona nivoa, tj uzroke preslikavanja.

Dokumentacija sadrži saznanja ili informacije o konkretnom razvoju informacionog sistema, dokumentovanje se realizuje shodno nekim događajima koje prate odgovarajuće aktivnosti, postoje i korisnici dokumentacije, koji možda i sami vrše dokumentovanje pomoću nekih konkretnih sredstava. Znači dokumentacija informacionog sistema je takodje sistem koji sadrži informacije o informacionom sistemu (podatke o podacima ili saznanja o saznanjima) i zato se često naziva metabazom podataka.

Dobra dokumentacija predstavlja najbolju osnovu za kontrolu kvaliteta isprojektovanog informacionog sistema. Inženjerski prilaz razvoju sistema, međjutim, nalaže da se projekat informacionog sistema posmatra kao bilo koji drugi industrijski proizvod, sastavljen od delova ili sastavnica. U proizvodnji sastavnica je usvojen standardizovani proces izgradnje, koji obuhvata pripremu, samu proizvodnju i kontrolu kvaliteta. Upravo ovaj opšteprihvaćen algoritam izgradnje je prihvaćen i u razvoju informacionih sistema: ceo razvoj se razbija na tačno definisane faze i korake razvoja. Tačno se zna šta obuhvataju faze i koraci razvoja i kako se oslanjaju na prethodne aktivnosti (korake i faze koji se vremenski moraju izvršiti pre dotičnih faza i koraka). Određeni su i kriterijumi za kontrolu kvaliteta rezultata pojedinih faza i koraka i oni se obavezno i kontrolišu nakon svake završene faze i koraka. Tako se u projektovanju informacionog sistema razlikuje tzv. fazni test i sistemski test.

Razvoj sistema kao što znamo obuhvata i realizaciju informacionog sistema. Sami softverski produkti se posebno testiraju. Test za procenu kontrole kvaliteta realizovanog softvera obuhvata sledeće dve aktivnosti:

1. verifikacija i
2. validacija.

Verifikacija softvera je niz aktivnosti u okviru kojeg se kontroliše funkcionalnost, tj. da li rade sve “grane” ili mogućnosti u programima bez greške (bez “pada”). Validacija softvera obuhvata procenu i kontrolu da li softver radi ono, samo ono i sve ono što je definisano u projektu i da li je to ono što je korisnik želeo. U cilju verifikacije već se u procesu projektovanja pripremaju, tačnije sastavljaju ili konstruišu test podaci koji obuhvataju sve “kritične” tačke u funkcionisanju sistema. Do današnjeg dana ne postoji ni jedan ozbiljniji softverski proizvod koji je stoprocentno testiran.

Verifikacija se realizuje na dva nivoa, shodno tome razlikujemo:

1. modul test i
2. integracioni test.

Integracioni test koristi već istestirane module i kontroliše saradnju izmedju modula (programa). Integracioni test može da otkrije nepravilnosti u definisanim iterfejsima izmedju modula i/ili u njihovim realizacijama u programima. Interfejsi sadrže upravljačke podatke i parametre informacionog (neupravljačkog) tipa. Shodno tome je preporučljivo integracioni test izvršiti u dva koraka:

1. top – down integracija i
2. bottom – up integracija.

Validacija softvera posmatra “celinu”, a ne pojedine izvršne jedinice ili module i kontroliše da li softver realizuje one funkcije koje su zacrtane u projektu, sa jedne strane, i da li to odgovara korisnicima – naručiocima, sa druge strane. Validacija softvera je testiranje pomoću “živih” podataka. Postoje dve vrste validacije:

1. alfa test i
2. beta test.

Alfa testom se naziva prvo testiranje karakteristika softvera koji izvršava razvojni tim, a beta test rade korisnici u njima specifičnim (hardversko-softverskim) uslovima.

6.3. METODOLOGIJA RAZVOJA INFORMACIONOG SISTEMA

Informacioni sistem se sastoji od velikog broja raznorodnih činilaca. Raznolikost njegovih elemenata i kompleksnost veze izmedju činilaca ukazuje na potrebu postojanja konkretne metode po kojoj bi se realizovao njegov razvoj. Po mnogim mišljenjima informacioni sistem je previše komplikovan, da bi se mogao obraditi bez konkretnog metoda.

Metod razvoja informacionih sistema obezbeđuje organizovanu celinu konkretnih tehnika i procedura za realizaciju informacionog sistema. Pošto se metodi sastoje od organizovanih činilaca (tehnika i procedura), i sami se mogu smatrati sistemima. Metodologija se bavi izgradnjom i analizom pojedinih metoda, što znači da je pojam metodologija na višem nivou apstrakcije od pojma metoda.

U svakom razvojnom procesu, nezavisno od primenjene metode za razvoj, se pojavljuju dve nerazdvojne, isprepletene niti: sadržajna (šta) i upravljačka (kako). U konkretnim metodama te dve niti se obično ne razdvajaju.

Metodi razvoja informacionih sistema (kao sistemi) treba da obuhvataju sledeće elemente (činioce):

1. osnovna koncepcija,
2. struktura i terminologija,
3. način predstavljanja,
4. sistem kriterijuma,
5. algoritam planiranja,
6. rukovanje dokumentacijom i
7. tehnička podrška.

6.3.1. Osnovna koncepcija

Koncepcija je ozbiljno smišljen način pristupa problemu. Osnovna koncepcija je apstraktna slika o predmetu razvoja, u našem slučaju o informacionom sistemu. Priča o osnovnoj koncepciji počinje pitanjem: šta je informacioni sistem. Pojam informacionog sistema je definisan u četvrtoj glavi, gde se kaže da je informacioni sistem organizovani skup njegovih elemenata kao i njihovih medjuveza. U te elemente spadaju podaci (informacije), sa njima povezani informacioni događaji, nad njima izvršene aktivnosti, potrebni resursi vezani za prethodna tri elementa, kao i za razvoj i funkcionisanje informacionog sistema, korisnik podataka i standardi i procedure koji upravljaju napred navedenim činiocima.

Neobično je važno, da svi članovi razvojnog tima (još bolje ako svi učesnici razvoja) prihvataju (ili priznaju) istu (zajedničku) osnovnu koncepciju.

6.3.2. Struktura i terminologija

Informacioni sistem je organizovana celina nehomogenih činilaca. Obuhvata projekciju podataka, obrade i okoline (gde spadaju i korisnici i potrebni resursi – vreme, novac i tehnička sredstva – za razvoj i funkcionisanje informacionog sistema). Podaci i obrade se mogu definisati na tri nivoa: pojmovnom, logičkom i fizičkom. Projekcije su na svakom od ova tri nivoa su i medjusobno povezane. Podaci su vezani za obrade, a podaci i obrade su povezani sa korisnicima. I unutar projekcija na istom nivou postoje medjuzavisnosti: podaci su i medjusobno povezani, a obrade su povezane sa aktivnostima koje ih slede i koje ih prethode.

Na osnovu prethodnog opisa se može zaključiti da je (unutrašnja) struktura informacionog sistema jako komplikovana, i da ima oblik mreže.

Struktura sistema je osnovan i smišljen način posmatranja činioca sistema i njihovih medjusobnih veza. Za apstraktnu sliku strukture informacionog sistema se ponekad koristi naziv (pojam) metamodel. Svaki konkretan informacioni sistem ima svoj plan koji sadrži konkretne elemente i njihove medjuzavisnosti; taj plan (projekat) se zove model informacionog sistema.

Činioci informacionog sistema i veze između njih moraju dobiti konkretna imena, da bi se o njima moglo raspravljati. Skup pojmova (tačnije njihovih konkretnih imena) koji se koristi za opisivanje tipičnih elemenata informacionog sistema i njihovih tipičnih medjuzavisnosti zove se terminologija. Pošto nazivi elemenata (pojmovi) i veza pripadaju pojmovnom (konceptualnom), logičkom ili fizičkom nivou, možemo zaključiti da je terminologija strukturirana. Terminologija treba da je jedinstvena za vreme razvoja i funkcionisanja informacionog sistema.

6.3.3. Način predstavljanja

Projekat informacionog sistema je i sam informacioni sistem, jer odgovara definiciji informacionog sistema formulisane u četvrtoj glavi. Projekat sadrži podatke (informacije), u okviru projektovanja (razvoja) postoje događaji i aktivnosti, za realizaciju su potrebni odgovarajući resursi, itd. Ako se za apstraktnu sliku strukture informacionog sistema koristi naziv metamodel, logično je organizovanom skupu činilaca u vezi projekta informacionog sistema dodeliti naziv meta informacioni sistem.

Samo formulisanje i opisivanje projekta informacionog sistema određuje njegov način predstavljanja. Na osnovu načina predstavljanja se razlikuju tri tipa projektovanja:

1. jezički orijentisano projektovanje,
2. projektovanje pomoću standardizovanih obrazaca (formulara) i
3. projektovanje pomoću unapred definisanih simbola.

Po uzoru na programske jezike na početku sedamdesetih godina su formulisali metajezik sa sintaktičkim i semantičkim pravilima, koji je služio za potrebe projektovanja, odnosno za razvoj informacionih sistema. Formulisanje delova projekta je bilo prilično naporno na tom metajeziku, zato je predložen novi prilaz u projektovanju na sredini sedamdesetih godina. Projektovanje se izvršavalo ispunjavanjem unapred definisanih standardizovanih obrazaca. Taj prilaz je ubrzao opis delova projekta, no i takvo projektovanje je napušteno zahvaljujući CAD (Computer Aided Design) programima koji su omogućili opis projekta pomoću unapred definisanih simbola.

Neophodno je uvideti da način predstavljanja nije ništa drugo, već jedan jezik, bez obzira na konkretne načine predstavljanja, i kao svaki drugi jezik mora imati jedinstvenu terminologiju, a takodje i sintaktička i semantička pravila.

Kao zaključak bi trebalo istaći da u razvoju informacionog sistema neophodno imati stabilnu sliku o strukturi informacionog sistema kao i objektivnu i preciznu terminologiju. Konkretno načine predstavljanja je najpogodnije zajednički koristiti menjajući ih po potrebi, jer tamo gde je jedan način slab ili suviše štur, tamo je baš drugi način predstavljanja jak i (semantički) bogat.

6.3.4. Sistem kriterijuma

U toku projektovanja informacionog sistema projektant, tačnije projektantski tim treba da ima pred sobom stabilnu vodilju: osnovnu koncepciju o informacionom sistemu, osnovnu strukturu izgradnje i terminologiju. Nije, međutim, dovoljno ukazati samo na sadržaj i način predstavljanja projekta. Moraju postojati i kriterijumi vrednovanja koji će poslužiti kao ispomoć pri projektovanju ili pri kontroli »kvaliteta« izradjenog projekta.

Projektant, bez obzira što je to u suprotnosti današnjoj praksi, treba da pripremi više alternativa projekta informacionog sistema (barem dve – tri). Projekat informacionog sistema je, kao što znamo, delimično apstraktna slika realne stvarnosti i nastaje kao rezultat intelektualnog napora razvojnog tima. Informacioni sistem, kao rešenje nije jedinstveno, čak i isti tim ne bi dvaput na isti način realizovao projekat po istom zadatku. Razradjene alternative su, međutim,

dragocene jer se mogu vrednovati po odgovarajućim kriterijumima za procenu kvaliteta. Nakon vrednovanja pojedinih alternativa projekta, može se desiti da je neka kombinacija predloženih rešenja najprihvatljivija za realizaciju.

Sistem kriterijuma je skup objektivnih i medjusobno povezanih merila za procenu valjanosti projektnih alternativa, odnosno projekata, koji je prihvaćen od strane svih članova razvojnog tima.

Sistem kriterijuma obuhvata velik broj konkretnih kriterijuma (po nekima čak više stotina) koji su medjusobno povezani i svesno organizovani, i upravo je to razlog zbog čega se mogu smatrati sistemom. Od velikog broja konkretnih kriterijuma ovde će biti nabrajano samo njih pet.

6.3.4.1. *Vernost preslikavanja stvarnosti*

Činioci u realnom svetu treba da se preslikaju zajedno sa njihovim medjusobnim vezama u informacioni sistem bez izobličenja.

Ovaj kriterijum se ne može formalno (automatski – pomoću računara – na osnovu matematičkih metoda ili semantike) kontrolisati.

Pogrešiti po ovom kriterijumu je moguće na dva načina: ako je projektant suviše precizan u preslikavanju, ili, pak, suprotno: ako je suviše neprecizan. Predvideti jedno najviše obrazovanje (jednu grupu podataka za opis šta, gde, kada, itd. je završio kao najviše obrazovanje) za radnike predstavlja izobličenje realnosti za one koji imaju više diploma istog (najvišeg) ranga. Vodjenje prethodnog radnog mesta ne znači nužno evidentiranje svih dvadeset prethodnih radnih mesta za pojedine radnike sa svim podacima kompanija (od kojih su možda mnoge i nestale, tj. prestale da funkcionišu).

6.3.4.2. *Razumljivost*

Razumljivost se može formulisati na apstraktnom i konkretnom nivou. Na apstraktnom nivou ona znači, da se pojmovi projektovanja (razvoja) informacionog sistema koriste pravilno. Na konkretnom nivou takodje mogu nastati problemi u vezi razumljivosti ako se nazivi neadekvatno koriste.

Preporučljivo je dodeliti razumljive nazive za delove sistema, obrade i podatke. Grešiti po ovom kriterijumu je moguće davanjem previše kratkih i nerazumljivih naziva za apstraktne ili intelektualne »proizvode« u projektu, a takodje i dodelom previše dugačkih (»pričljivih«) naziva.

6.3.4.3. Jednoznačnost

Jednoznačnost kao kriterijum je povezan sa razumljivošću i znači korišćenje uvek istog, odgovarajućeg naziva za isti pojam na apstraktnom i na konkretnom nivou. To drugim rečima znači: treba se boriti protiv homonima i sinonima.

Homonimi se javljaju u tom slučaju kada se za različite pojmove (suštine) koriste isti nazivi. Sinonimi nastaju tada, kada se ista suština (pojam) označava različitim nazivima.

Kriterijum jednoznačnosti nije ispoštovan u slučaju da se nepotrebno koriste prekratki ili predugački nazivi za delove sistema, obrade ili podatke. U osiguravajućim zavodima postoje razni tipovi osiguranja, kao i tipovi vozila. Korišćenjem samo naziva tip (umesto tip vozila i tip osiguranja) javlja se problematika homonima. Takodje u osiguravajućem zavodu postoji životno osiguranje, osiguranje imovine, osiguranje motornih vozila, itd. Dodelom naziva lični broj osiguranika, lični broj vlasnika imovine, lični broj vlasnika vozila, itd., projektant nepotrebno, generiše sinonime.

Kontrola jednoznačnosti se može izvršiti pomoću matematičkih i semantičkih metoda, ali analiza homonima i sinonima traži primenu i drugih iz navedenih šest kriterijuma.

6.3.4.4. Minimalnost

Složeni realni sistem se ne može predstaviti pomoću jednostavnog informacionog sistema. Ne sme se tražiti minimalnost po svaku cenu: bitni elementi fizičke realnosti se ne smeju ispustiti iz vida u cilju da bi se postigla minimalnost projekta informacionog sistema.

Kriterijum minimalnosti se poštuje u tom slučaju ako se isključuju (»zaobidju«) preklapanja, odnosno redundancije u projektovanju. Projektanti ne vole redundanciju, jer nije uvek jednoznačno na prvi pogled da li su time povredili kriterijum redundantnosti ili možda jednoznačnosti. Ako je na više mesta dat samo naziv podatka (obeležja) »tip«, nije izvesno da li se radi o homonimima ili, možda, redundanciji.

Projekat informacionog sistema ima tri projekcije i tri apstraktna nivoa: konceptualni, logički i fizički. Na pojmovnom nivou treba težiti apsolutnoj minimalnosti, odnosno isključiti sva moguća preklapanja. Na logičkom i fizičkom nivou, zbog primenjenog softvera za rukovanje podacima i računara slabijih performansi ponekad je redundancija »nužno zlo« i predstavlja jednu od mogućnosti za »fino podešavanje« funcionisanja sistema.

Redundancija se može ustanoviti pomoću matematičkih metoda, no otkrivene redundancije se moraju ispitivati da li se možda radi o homonimima. Nakon otkrivanja homonima redundancije se lako eliminiše.

6.3.4.5. Celovitost

Kriterijum celovitosti je složeniji od prethodnog, jer se ne može automatizovati njegova kontrola. Ne postoji pomoćno sredstvo koje bi moglo ukazati na odsutne elemente u projektu koji su u realnom sistemu prisutni.

Ako projekat nije celovit, to znači da ne predstavlja vernu sliku stvarnosti, odnosno grešiti po ovom kriterijumu u pravcu »manjkavosti« nije moguće, a da se ne greši o kriterijumu vernost preslikavanja stvarnosti. Kako se može grešiti u »drugom pravcu«? Od viška ne boli glava kaže poslovice, to, međjutim, ne važi za informacioni sistem. Ako projekat sadrži ponavljanja (»celovitiji od celovitog«), onda su povredjeni (nisu podržani) kriterijumi minimalnosti i (eventualno) jednoznačnosti.

6.3.4.6. Harmonija kriterijuma

Gore objašnjeni kriterijumi, kako je to u njihovim opisima podvučeno, su medjusobno tesno povezani. Projekat informacionog sistema jedino tada može biti optimalan, ako odgovara svim opisanim kriterijumima. Redosled ispitivanja ili kontrole projekta po kriterijumima nije slučajan. U prvom »prolazu« se rešenje analizira po jednoznačnosti, razumljivosti i vernosti preslikavanja.

Najstrožije se kontroliše kriterijum jednoznačnosti. Tu se javljaju sinonimi i homonimi koje svakako treba eliminisati imajući u vidu i kriterijum razumljivosti, da bi se na kraju kontrolisala vernost preslikavanja. Nakon eliminisanih sinonima i homonima se projekat ispituje po minimalnosti i celovitosti uz održavanje vernosti preslikavanja.

Sinonimi označavaju skrivenu redundanciju. Korišćenje različitih naziva za iste pojmove može rezultovati zaključak da projekat zadovoljava kriterijum minimalnosti, što nije tačno zbog postojećih (neotkrivenih) sinonima. Sinonimi mogu dovesti do inkonzistentnih stanja u podacima. Ne znajući koji nazivi se koriste za konkretni pojam (napr. ako tip vozila i vrsta vozila označavaju isti pojam), različiti nazivi podataka (obeležja) mogu biti punjeni sa drugačijim vrednostima (podacima, konkretizacijama, tj. tip vozila može primiti različit sadržaj od vrste vozila).

Očigledna redundancija u projektu može biti i homonim. Uklanjanjem takve redundancije projekat neće odgovarati kriterijumu celovitosti. Zato se često kaže da je pojava homonima u stvari prividna redundancija.

Upravo se iz tih razloga vrši prvo eliminisanje sinonima i otklanjanje homonima uz kriterijum razumljivosti i vernosti preslikavanja, a tek posle minimalnost i celovitost.

6.3.5. Algoritam projektovanja

Proces razvoja informacionog sistema pripada u delokrug projekt menedžmenta i sastoji se iz više faza. Jedna od faza razvoja informacionog sistema je projektovanje.

Algoritam projektovanja označava grupu aktivnosti u okviru koje projektant razradjuje jedan za drugim elemente (činioce) modela informacionog sistema i te činioce usaglašava. Formulacija u prethodnoj rečenici u vezi izgradnje »modela informacionog sistema« ukazuje na primenu sistema kriterijuma za vrednovanje pri razradi činilaca sistema.

Algoritam projektovanja može biti striktno propisan (kao kod metoda korišćenih u sedamdesetim godinama) ili fleksibilan (kao kod najviše korišćenih današnjih automatizovanih metoda). Šta više, današnji računarom podržani automatizovani metodi razvoja informacionog sistema imaju toliko fleksibilne algoritme projektovanja, da se oni skoro i ne mogu smatrati algoritmom.

Algoritam projektovanja BISAD (Business Information Systems Analysis and Design) metoda je strogo propisao korake u projektovanju i formulisanju rešenja. Kruti algoritmi ne dopuštaju da kreativnost projektanta dominira pri realizaciji projektnog zadatka.

Pristup projektovanju (algoritam projektovanja) pod nazivom »Pokušaj i promašaj« (trial and error) u stvari i nije algoritam. Projekat (najčešće samo jedna obrada) se sastavlja na osnovu prvih razmišljanja i ideja, na osnovu čega se pomoću generatora aplikacija odmah realizuju programi. Gotov informacioni sistem (odnosno samo jedan njegov deo) se odmah testira,

vrednuju se dobijeni rezultati obrade i u slučaju promašaja, na osnovu prvobitne ideje i iskustva stečenih u analizi pogrešnih rezultata (promašaja) sastavlja se novi projekat i celokupan ciklus se ponavlja. Ovaj pristup je koristan, istina ne kao algoritam projektovanja, nego kod fizičke realizacije (programiranja projekta) za optimiziranje (tuning) obrade.

Pristup projektovanju kojeg poznajemo pod nazivom »Metod pretprojekta« (pilot – project) obuhvata pogodno izabrani deo informacionog sistema (podsystem) i to obradjuje do »kraja«, tj. realizuje do fizičkog nivoa. Taj proces se naziva izradom prototipa (prototyping). Problemi nastaju ako projektanti nisu svesni da se celokupan sistem ne može ostvariti isprojektovanjem njegovih delova. Na konceptualnom (pojmovnom) nivou je neophodno obuhvatiti kompletan projekat informacionog sistema i vrednovati ga po kriterijumima za procenu kvaliteta. Na logičkom i fizičkom nivou delovi sistema mogu biti ostvareni privremeno i na različitim nivoima realizacije. Prototip ne može poslužiti kao konačno rešenje, ali njegova izrada je dragocena u smislu stečenih iskustava.

U prethodnim izlaganjima je istaknuto da su projektovanje i analiza usko povezane aktivnosti. Algoritam projektovanja mora obuhvatiti i projektovanje i analizu, a takodje i iteraciju. Nakon isprojektovanog sistema mora slediti analiza, koja na osnovu kriterijuma za procenu vrši vrednovanje dobijenog rešenja, a iteracije omogućuju novi ciklus projektovanja i analize: tako se dobijaju različite alternative projekta informacionog sistema.

Prilikom projektovanja, na početku se uvek postavlja pitanje: odakle početi, koja da bude fiksna polazna tačka. U procesu razvoja metoda za projektovanje razni algoritmi su imali sledeće polazne tačke:

1. izlazi (izveštaji),
2. ulazi,
3. obrade, tj. funkcije i
4. podaci.

Kruti algoritmi za projektovanje su uvek striktno odredili polaznu tačku rada, fleksibilniji to, naravno, ne čine. Na početku sedamdesetih godina projektovanje je krenulo od izlaza i algoritam je služio za otkrivanje potrebnih ulaznih podataka i odgovarajućih obrada za dobijanje zahtevanih izveštaja. Zbog varijabilnih zahteva za izlaznim podacima ulazni podaci su postali polazište za projektovanje: od njih je sve zavisilo, njihovo poznavanje je omogućilo, preko raznovrsnih obrada, iskazivanje bilo kakvih rezultata. Od sredine sedamdesetih godina su se koristili strukturirano programiranje i projektovanje. U okviru strukturiranog dizajna sistema crtali su se strukturni dijagrami događaja i aktivnosti, odnosno obrade. Tako su obrade postale polazna tačka u projektovanju. Tu se prvi put javio pojam crne kutije (black box). Na višim

nivoima hijerarhije je bila poznata samo suština glavnih funkcija, a sadržaj ne. Nepoznati sadržaj crne kutije je osvetljen na nižim nivoima u hijerarhijskoj strukturi. Zbog promenljivih zahteva za raznolikim obradama, kao polazna tačka projektovanja postao je najstabilniji element informacionog sistema, tako da danas kao polazna osnova za projektovanje služe podaci, odnosno projektovanje baze podataka (database design).

Delovi informacionog sistema mogu biti jako različite prirode i zato, možda treba menjati polazne osnove, odnosno algoritme projektovanja. Proces (algoritam) projektovanja se mora oslanjati na valjanu i objektivnu strukturu informacionog sistema i da obezbeđuje analizu rešenja (projekta) po kriterijumima za procenu kvaliteta.

6.3.6. Rukovanje dokumentacijom

U toku razvoja informacionog sistema pripremaju se razne dokumentacije. Ako se ograničimo samo na dokumentaciju projekta informacionog sistema, i njih može biti više (tu se misli na alternative rešenja projekta).

Dokumentacija informacionog sistema je od suštinskog značaja za sudbinu informacionog sistema u celini i za projektantski tim, jer:

1. služi kao osnova za komunikaciju između članova razvojnog tima u toku procesu projektovanja (razvoja),
2. predstavlja izvor informacija za nove korisnike (prevažodno za nove članove projektantskog tima),
3. treba da služi kao standard za elemente informacionog sistema i njihovih medjuveza na apstraktnom i konkretnom nivou, za pojmove i procedure koje se koriste u projektu, i
4. omogućava analizu ostvarenog rešenja na osnovu sistema kriterijuma.

Sistem kriterijuma sadrži kontradiktorne zahteve, što se lako može zaključiti pažljivim čitanjem njihovog opisa. Nije jednom zauvek dat redosled kriterijuma za kontrolu kvaliteta, a ako se prioritet kriterijuma izmeni nužno je izvršiti ponovnu analizu kvaliteta projekta na osnovu projektne dokumentacije.

Mnogi smatraju (a o tome je već i bilo reči) da dokumentacija pored činjenica treba da sadrži i razloge za rešenja korišćena ili prihvaćena u projektu. Dokumentacija predstavlja takodje sistem, jer sadrži podatke o projektu informacionog sistema, u toku realizacije dokumentacije javljaju se događaji koji izazivaju izvršavanje potrebnih aktivnosti – po izvesnim standardima, obrada (stvaranje) dokumentacije se izvršava pomoću odgovarajućih tehničkih sredstava,

neophodni su i resursi (vreme, novac) za njeno ostvarenje, a potencijalnih i faktičkih korisnika ima mnogo. Pošto dokumentacija sadrži saznanja o saznanjima (podatke o podacima), ona predstavlja meta bazu podataka.

Rukovanje dokumentacijom obuhvata efikasno i organizovano smeštanje podataka o informacionom sistemu, tj. optimalno rukovanje podacima i ostalim elementima dokumentacije kao sistema. Dokumentacija treba da odslikava osnovnu koncepciju i strukturu informacionog sistema, da koristi prihvaćeni(e) način(e) predstavljanja, da sadrži ideje koje odgovaraju propisanim kriterijumima, da sledi algoritam projektovanja, jednom rečju da bude u saglasnosti sa ostalim elementima metodologije projektovanja informacionog sistema.

Sa druge strane, dokumentacija – nezavisno od sadržaja plana – poseduje sopstveni život. O tome svedoče situacije iz prakse kada se »sitne« i »neznatne« promene sprovedene u funkcionisanju sistema ne uvode u dokumentaciju, ili kada se poboljšanja projektanta, predložena i predstavljena u projektnoj dokumentaciji nikad ne sprovedu u informacionom sistemu.

Zadatak rukovanja dokumentacijom bi bio da obezbedi harmoničan i jednoznačan odnos izmedju isprojektovanih, uvedenih (realizovanih) i promenjenih činilaca informacionog sistema, tačnije, izmedju činilaca u izgradnji ili funkcionisanju sa jedne, i njihove slike sa druge strane.

6.3.7. Tehnička podrška

Na današnjem stepenu razvoja računarske tehnike i informatike sasvim je prirodna pojava da se razvoj (projektovanje) informacionih sistema realizuje uz pomoć računara, i da se podaci vezani za projekat informacionog sistema obradjuju i evidentiraju na računaru.

Softveri koji na bilo kakav način podržavaju projektovanje informacionih sistema danas se nazivaju CASE (Computer Aided Systems Engineering) alatima. Broj takvih softverskih proizvoda je velik, no najveći procenat tih alata realizuje samo deo projekta informacionog sistema na računaru. Treba istaći činjenicu da su ti CASE softveri vrlo simpatični, rade sa grafičkim interfejsom i prozorima u Windows okruženju, da imaju ikonice pored raznoraznih menija i funkcionišu vrtoglavom brzinom (na snažnim računarima). Kod vrednovanja i izbora CASE alata, međjutim, treba se usredsrediti na osnovnu koncepciju i strukturu koju alat podržava (razlikuje li tri osnovne projekcije i tri nivoa projektovanja i da li ima odgovarajuće

strukture na tim nivoima), na korišćenu terminologiju i način predstavljanja (da li su simboli pogodni za predstavljanje realnog sistema), na sistem kriterijuma pomoću kojih kontroliše rad na projektu i sam projekat, na algoritam projektovanja i način dokumentacije.

Kompletan razvoj informacionog sistema pomoću računara danas još ne postoji, najverovatnije zbog toga, što se pažnja više posvećuje tehničkoj podršci softvera (korisničkim prozorima, ikonama) nego sadržaju, odnosno usštini samog zadatka.

6.3.8. Meta standardi

O standardima je već bilo reči kod opisa pojma informacionog sistema (u tačkama 4.2., odnosno 4.2.5.). Standardi bez epiteta (atributa) se odnose na korisničke standarde, i oni predstavljaju jednog od pet činioca informacionog sistema. To su prihvaćeni sporazumi koji se odnose na korisničke podatke, događaje, aktivnosti, tehnička sredstva, itd.

Postoje, međutim, kao što znamo i podaci, događaji, aktivnosti, tehnička sredstva, itd. koji se odnose na razvoj, tačnije projektovanje informacionih sistema, a koje propisuju ili upravljaju razvojni standardi. Te standarde, pošto se odnose na saznanja o saznanjima, često nazivaju meta standardima.

Meta standardi ne spadaju u činioce metode (tj. u metodologiju) projektovanja informacionih sistema, nego nekako obuhvataju ili se odnose na sve napred nabrajane činioce (elemente) metodologije.

U razvoju informacionih sistema ne postoje opšte prihvaćene konvencije, odnosno meta standardi za svih napred opisanih elemenata metodologije projektovanja, i zbog toga na ovom polju vlada veliki haos.

7. PROJEKCIJA PODATAKA INFORMACIONOG SISTEMA

Informacioni sistemi se razvijaju da bi zadovoljili potrebe organizacija različitih profila, kao i pojedinaca, a iza tih zahteva i potreba, u pozadini uvek stoje ljudi i njihova težnja da što brže i jednostavnije dodju do njima važnih podataka, informacija. Podaci i ljudi spadaju u činioce informacionih sistema, i kako su oni medjusobno povezani, tako su i sa ostalim elementima informacionog sistema u vezi.

Saznanja u okviru nekog konkretnog informacionog sistema i sami predstavljaju sistem. Sistem podataka (projekcija podataka) je i sam jako složen. Od saznanja do podatka takodje treba stići, isto tako od podatka do informacije. Podatak, koji će za neke postati i informacija predstavlja osnovu za komunikaciju. Podatak ima razne svoje aspekte. Njega treba fizički smestiti na magnetske medijume računara, podaci su medjusobno i logički povezani, a svakako služe za stvaranje slike o realnom sistemu.

7.1.O SAZNANJU

Ljudsko biće neprekidno skuplja saznanja iz okolnog sveta pomoću svojih čulnih organa. Nepobitno je da i osmeh, prijatan miris i ukusno jelo može biti jednoznačno i važno saznanje, informacija, no ipak se za komunikaciju između ljudi najviše koristi jezik. Kao što već znamo podatak je saznanje koje se može protumačiti, ali koje još nije protumačeno. Protumačeni podatak, ako nosi novo saznanje za lice koje ga je protumačio, postaće informacija.

U prirodnom govoru (jeziku) svaki prost iskaz (prosta rečenica) ima subjekat i predikat. U slučaju da fali subjekat ili predikat, on se podrazumeva, odnosno može se zaključiti na osnovu prethodnih iskaza. Subjektat određuje o čemu želimo nešto izjaviti, a predikat sam iskaz u vezi subjekta rečenice.

U informacionom sistemu saznanja se formalizuju. Osnovni koncepti ili pojmovi koji se u sistemu podataka na pojmovnom (konceptualnom) nivou informacionog sistema koriste, jesu: entitet, obeležje i poveznik.

7.2. ENTITET

Pojave realnog sveta koje želimo opisati informacionim sistemom na konceptualnom nivou apstrakcije nazivamo entitetima. Tačnije entitet predstavlja sliku neke pojave realnog sistema na pojmovnom nivou apstrakcije. Entitet je nešto što se može jednoznačno identifikovati, on je “objekat posmatranja” ili “jedinica posmatranja”. Taj termin se može koristiti za svaki realni subjekat, objekat, događaj, pojavu ili apstraktni pojam.

Čovek uvek klasifikuje pojave realnog sveta. Klase pojava realnog sveta koje želimo opisati pomoću informacionog sistema na pojmovnom nivou nazivamo tipom entiteta. U stvari predstavljanje klase pojava realnog sistema na konceptualnom nivou se naziva tipom entiteta.

Pojave realnog sveta čovek uvrštava u klase putem apstrakcije koja je uvek zavisna od ciljeva samog sistema ili od namere čoveka. Od trenutka konstruisanja nekog apstraktnog tipa entiteta, njemu pripadajuće konkretne entitete nazivamo konkretizacijama ili pojavama tipa entiteta. Konkretizacije tipa entiteta su u stvari vrednosti iz (vremenski zavisnog – promenljivog) skupa pojava pripadajućeg tipa entiteta. Tip entiteta predstavlja pojam na višem nivou apstrakcije od pojave tipa entiteta.

Pretpostavimo, u cilju ilustracije gore izloženih pojmova, da želimo voditi podatke o studentima Zoranu i Dejanu, prema tome oni su entiteti (objekti posmatranja) za nas. Odredimo li apstraktni tip entiteta STUDENT, od tog trenutka Zoran i Dejan gube svojstvo entiteta i postaju konkretizacije ili pojave tipa entiteta STUDENT.

7.3. OBELEŽJE

Pojave realnog sistema imaju osobine pomoću kojih se mogu opisati, klase pojava takodje. Slika osobine klase pojava realnog sveta na pojmovnom nivou abstrakcije se naziva obeležje (atribut). Drugim rečima osobine tipova entiteta na pojmovnom nivou se nazivaju obeležjima (atributima). Obeležje je takodje apstrakcija, jer čovek dodeljuje pojmovima nazive. Sve pojave (konkretizacije) istog tipa entiteta imaju barem jednu zajedničku osobinu (obeležje), na osnovu koje su svrstane u isti tip entiteta.

Kao ilustrativni primer navedimo moguća obeležja za apstraktni tip entiteta VOZILO: registarski_broj, broj_šasijske, tip_vozila, boja, itd.

Svakom obeležju odgovara jedan skup mogućih vrednosti (pojava ili konkretizacija) koje pomenuto obeležje u konkretnim slučajevima može imati. Taj skup vrednosti se zove domen

ovbeležja. Slično kao kod entiteta, ovde je obeležje apstraktni, a vrednost obeležja konkretni pojam.

Entiteti, konkretizacije entiteta, obeležja i vrednosti obeležja služe za preslikavanje pojava i njenih osobina iz fizičke realnosti na konceptualni nivo predstavljanja informacionog sistema. Treba, međutim, istaći da su entitet i obeležje relativni pojmovi. Za boju po pravilu kažemo da je obeležje (napr. boja_automobila), i to s pravom, međutim u fabrici boja i lakova “boja” treba da bude opisana pomoću raznih informacija (od kojih osnovnih boja je sastavljena i sa kojim procentima, koliki je procenat razredjivača u sastavu, kojom tehnologijom treba nanositi tu boju, itd.), tj. tu je boja entitet.

Obeležje ili grupa obeležja koja za svaku pojavu entiteta datog tipa uzima različite vrednosti nazivamo identifikatorom ili ključem (kandidatom za ključ). Retko se nadje neka prirodna osobina – obeležje čije su konkretizacije (vrednosti) različite za svaku pojavu datog tipa entiteta. U tom slučaju se obično “veštački” uvodi jedno obeležje koje će imati različite vrednosti za svaku pojavu posmatranog tipa entiteta, ili se grupisanjem nekoliko atributa (obeležja) postiže da će njihove konkretizacije biti različite za svaku pojavu datog entiteta.

Na osnovu prethodnog razmišljanja se može zaključiti da postoje dva vrste identifikacije:

1. nominativna i
2. deskriptivna.

Kod nominativne identifikacije postoji jedno pogodno izabrano obeležje čija je vrednost različita za sve konkretizacije posmatranog tipa entiteta. Primer: broj_narudžbe koji jednoznačno identifikuje svaku narudžbu, ili navodjenjem jedinstvenog matičnog broja građana (JMBG) za identifikaciju lica.

Deskriptivna identifikacija se dobije kada se uključuje toliki broj obeležja u primarni ključ, da on jednoznačno identifikuje svaku pojavu tipa entiteta. Primer: deskriptivna identifikacija lica (građana) se realizuje pomoću identifikatora sastavljenog od sledećih obeležja: ime, prezime, datum_rodjenja, mesto_rodjenja, ime_oca, adresa.

U praksi se može desiti da jedan tip entiteta ima više takvih obeležja koja jednoznačno određuju njegove konkretizacije, tj. ima više kandidata za ključ. U tom slučaju se jedan od kandidata bira za ključ (primarni ključ).

7.4. POVEZNIK

U prirodnom govoru retko koristimo kompletne proste iskaze (celovite proste rečenice). Rečenice ili iskaze najčešće povezujemo. Danas je već svim informatičarima poznato da saznanje ili podatak generalno ima tri svoje projekcije:

1. ekstenzionalnu,
2. intenzionalnu i
3. transverzijalnu.

Shodno tome ekstenzija tipa entiteta odgovara skupu njegove konkretizacije (pojave). To znači da ekstenziju tipa entiteta dobijemo nabrojanjem svih njegovih pojava.

Intenziju tipa entiteta predstavlja skup njegovih obeležja. Intenzija tipa entiteta se dobije navodjenjem svih njegovih obeležja.

Transverzija predstavlja povezanost ili razmenu podataka izmedju pojava (entiteta) ili grupa pojava (tipova entiteta). Transverzija se može definisati na dva nivoa apstrakcije.

Povezanosti izmedju entiteta na pojavnom nivou ili odnosi izmedju pojava tipova entiteta nazivaju se vezama (to su konkretne veze) ili poveznicama. Generički odnosi izmedju tipova entiteta nazivaju se tipovima poveznika.

Najvažnija crta tipa poveznika je njegova kardinalnost. Tipovi poveznika se mogu podeliti u tri grupe, što ujedno i znači da dva tipa entiteta mogu biti u sledećim odnosima: 1:1, 1:N i M:N. Primer za kardinalnost veze 1:1 je veza izmedju vozila i obaveznog osiguranja, tj. da jedno vozilo može da ima samo jedno obavezno osiguranje, a takodje jedno osiguranje se može odnositi samo na jedno vozilo. Kardinalnost tipa 1:N je prisutna u povezanosti izmedju lica (vlasnika) i vozila, tj. da jedno vozilo može imati samo jednog vlasnika, a jedno fizičko lice može imati više vozila. Primer za tip kardinalnosti M:N je veza izmedju vozila i saobraćajnih nezgoda, tj. u jednoj saobraćajnoj nezgodi mogu učestvovati više vozila, a da jedno vozilo može »preživeti« ili učestvovati u više saobraćajnih nezgoda (naravno, i nadajmo se, ne istovremeno).¹

Postoje mišljenja da se svaki tip poveznika, bez obzira na njihovu kardinalnost, mora opisati podacima (saznanjima). Treba, medjutim, imati na umu i činjenicu da sve pojave koje se žele opisati saznanjima nazivaju entitetima, tj. ako se veza opiše podacima postaće entitet (primer robe, narudžbe i stavke narudžbe), ili kako se još naziva entitet-poveznik.

Ima preporuka sa anglosaksonskog govornog područja da povezanost označava (veže) dve suštine u zavisnosti od toga sa koje strane se posmatra ta veza, pa prema tome treba dodeliti dva

¹ Primeri uzeti iz [11.]

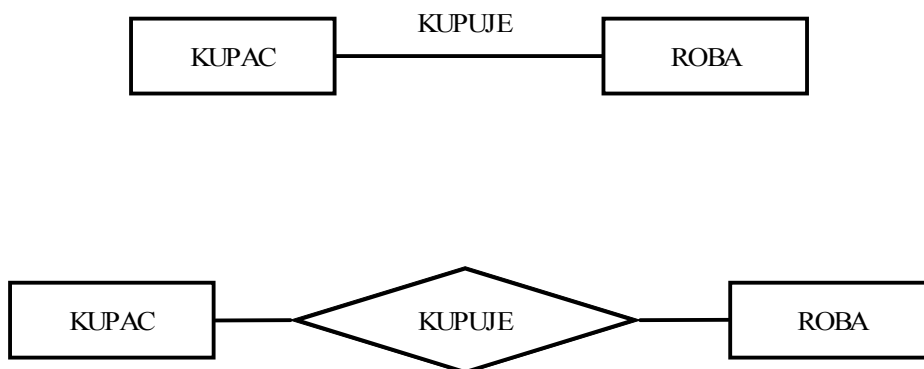
naziva za istu fizičku vezu. Povezanost je jedna bez obzira što povezuje dva (tipa) entiteta i zato je pogrešno posmatrati sa dve različite strane i dodeliti joj dva naziva.

Za povezanost tipa kardinalnosti M:N se kaže da ne predstavlja pravu vezu između dva entiteta koje povezuje, tj. ti entiteti nisu u vezi u pravom smislu reči. Dva entiteta su samo tada povezani ako se identifikator (ključ) jednog entiteta može pronaći u drugom entitetu. Tip poveznika kardinalnosti M:N nije poželjan i zato ga treba ubacivanjem novog entiteta–poveznika razbiti na dva tipa poveznika kardinalnosti 1:N.

Poveznik ili povezanost je pojam koji se najteže shvata, međutim veze predstavljaju ključ za efikasno memorisanje i rukovanje povezanih saznanja.

7.5.EKSTERNA STRUKTURA TIPa ENTITETA

Eksterna struktura tipa entiteta nosi informaciju o entitetima i njihovim međusobnim vezama. Često se za nju koristi i pojam (naziv) logički model podataka, tako da za konstruisanje eksterne strukture tipa entiteta stručnjaci ponekad kažu prosto da je to modeliranje podataka (na konceptualnom – pojmovnom ili logičkom nivou). Logički model podataka se realizuje u grafičkoj formi. Entitet se na logičkom modelu podataka prikazuje pravougaonikom, a za



Sl. 7.1.

poveznik postoje dve predstave, kako je to prikazano na sl. 7.1. Temeljitiom analizom veze između tipova entiteta se lako može uvideti da pored kardinalnosti postoji i njena druga osobina opcionalnost veze, koja opisuje prirodu egzistencije entiteta i njemu pripadajuće veze, ili drugim rečima međusobnu uslovljenost poveznika i entiteta. Tu se prati uslovljenost između postojanja entiteta i egzistencije makar jedne jedine njegove veze. To znači da opcionalnost veze određuje

da li može egzistirati entitet bez postojanja barem jedne konkretne realizacije tipa poveznika. Za poveznik na sl.7.1. KUPUJE ova formulacija može da glasi: da li može egzistirati konkretan kupac, koji do sada nije kupio nijednu robu, ili da li može da postoji pojava entiteta roba (tj. neka konkretna roba) koju do sada nije kupio nijedan kupac. Te dve osobine se mogu opisati i pratiti ažuriranjem minimalne i maksimalne kardinalnosti, kao što je objašnjeno u sledećoj tački.

7.5.1. Semantika kardinaliteta tipa poveznika

Podrobnijom analizom opštih osobina veza izmedju različitih tipova entiteta, a takodje izmedju različitih pojava entiteta istog tipa, može se doći do opšteg opisa kardinaliteta tipa poveznika koji ima sledeći oblik:

$$R (E_1 (a_2 , b_2) : E_2 (a_1 , b_1))$$

U prethodnom izrazu R označava naziv tipa poveznika, E_1 i E_2 su nazivi entiteta, a_1 i a_2 su minimalni, a b_1 i b_2 su maksimalni kardinaliteti tipa poveznika R.

U slučaju kad se govori o kardinalitetu poveznika, kao u tački 7.4., obično se podrazumevaju maksimalni kardinaliteti. Shodno maksimalnim kardinalitetima (kardinalitetima grupe 1:1, 1:N i M:N) upoznatih kod definisanja poveznika vrši će se analiza mogućih kombinacija tipa i kardinalnosti, tj. minimalne i maksimalne kardinalnosti veza.

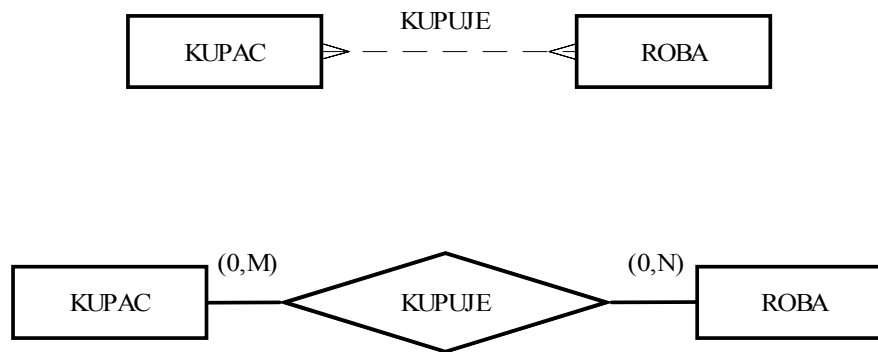
7.5.1.1. Kardinalitet grupe M:N

Kod ove grupe kardinaliteta su maksimalni kardinaliteti fiksni i poznati: $b_1 = N$ i $b_2 = M$. Minimalni kardinaliteti se slobodno menjaju i mogu uzimati bilo koje vrednosti. Četiri osnovna slučaja se analiziraju u ovoj grupi kardinaliteta kao što sledi u daljem tekstu.

$$1. \quad \mathbf{a_1=0 \text{ i } a_2=0} \quad [R (E_1 (0 , M) : E_2 (0 , N))]$$

Minimalni kardinaliteti ukazuju na činjenicu da u obe klase mogu postojati entiteti koji nisu povezani sa bilo kojim entitetom druge klase, ali, naravno, nije isključen ni taj slučaj kada su svi entiteti posmatrana dva tipa entiteta (posmatranih klasa) međusobno povezani. Grafička predstava ove vrste veze je prikazana na sl. 7.2. Entiteti klase E_1 (KUPAC) ne moraju biti povezani ni sa jednim entitetom, mogu biti povezani sa jednim ili sa više entiteta klase E_2 (ROBA). To važi i obratno, tj. entiteti klase E_2

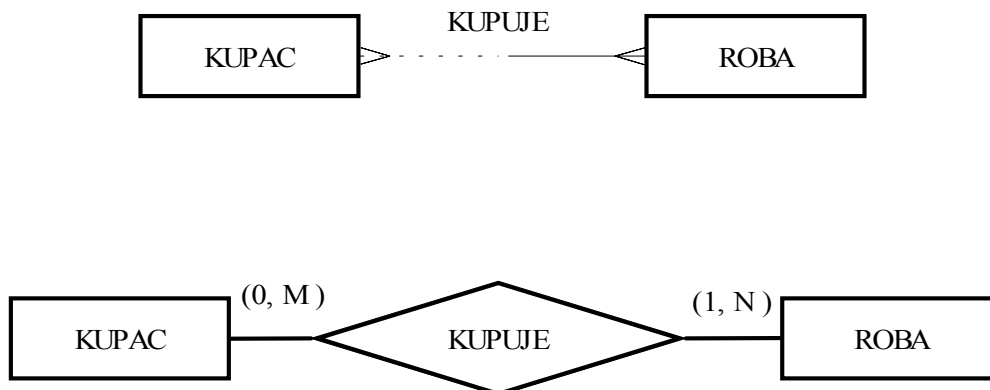
(ROBA) mogu biti povezani sa jednim (ali ne moraju ni sa jednim) ili više entiteta klase E_1 (KUPAC).



Sl. 7.2.

2. $a_1=1$ i $a_2=0$ [R ($E_1(0, M):E_2(1, N)$)]

Formulacija ove veze je sedeća: svi entiteti tipa entiteta E_2 moraju biti povezani bar sa jednim entitetom klase E_1 . To znači da je egzistencija entiteta u klasi E_2 je uslovljena njihovom povezanošću sa barem jednim entitetom iz tipa entiteta E_1 . Entiteti iz tipa entiteta E_1 ne moraju biti povezani ni sa jednim entitetom, mogu biti povezani sa jednim ili sa više entiteta klase E_2 . Grafička predstava ove veze se nalazi na sl. 7.3.

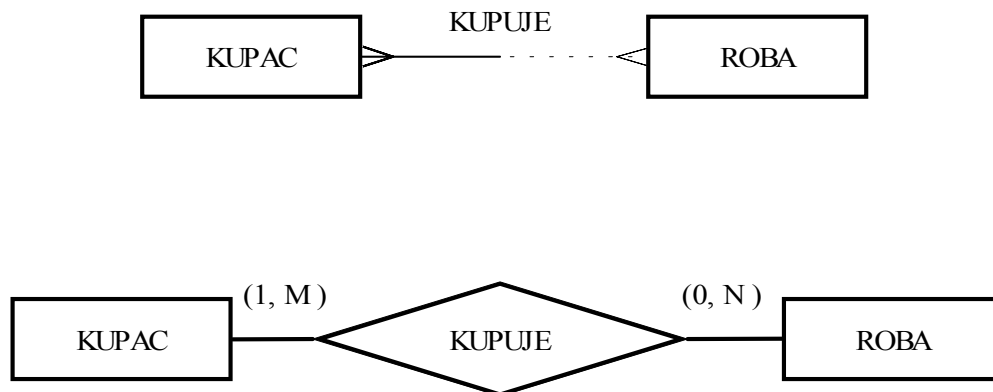


Sl. 7.3.

3. $a_1=0$ i $a_2=1$ [R ($E_1(1, M):E_2(0, N)$)]

Ova veze se može opisati na sledeći način: svi entiteti tipa entiteta E_1 moraju biti povezani bar sa jednim entitetom klase E_2 . To znači da je egzistencija entiteta u klasi E_1 je uslovljena njihovom povezanošću sa barem jednim entitetom iz klase entiteta E_2 . Entiteti iz tipa entiteta E_2 ne moraju biti povezani ni sa jednim entitetom, mogu biti

povezani sa jednim ili sa više entiteta klase E_1 . Grafička predstava ove veze se nalazi

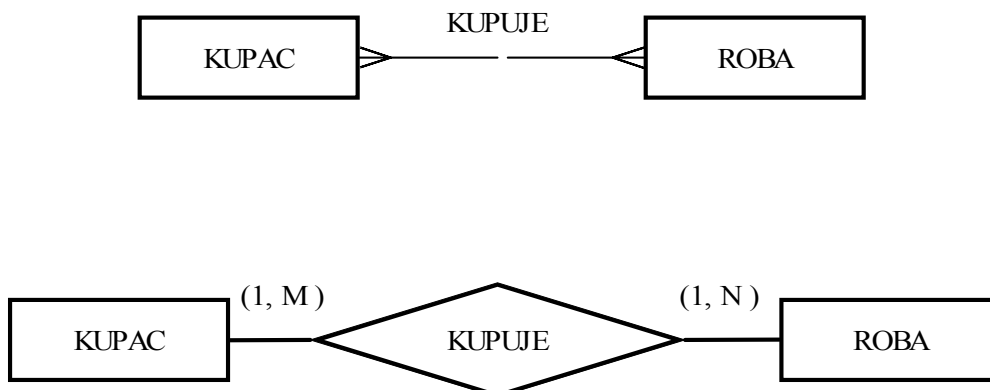


Sl. 7.4.

na sl. 7.4.

4. $a_1 = 1$ i $a_2 = 1$ $[R(E_1(1, M):E_2(1, N))]$

Ovaj četvrti tip veze grupe »više prema više« ima sledeće semantičko značenje. Svaki entitet jednog tipa entiteta mora biti povezan sa bar jednim entitetom druge klase. Egzistencija entiteta u obe klase uslovljena je njihovom povezanošću sa bar jednim entitetom druge klase, što predstavlja izuzetno strogo ograničenje. Konkretno, svi entiteti tipa entiteta E_1 moraju biti povezani bar sa jednim (a, naravno, mogu i sa više) entitetom klase E_2 , a takodje svi entiteti klase E_2 moraju biti povezani bar sa jednim (a, naravno, mogu i sa više) entitetom tipa entiteta E_2 . Grafička predstava ove veze se nalazi na sl. 7.5.



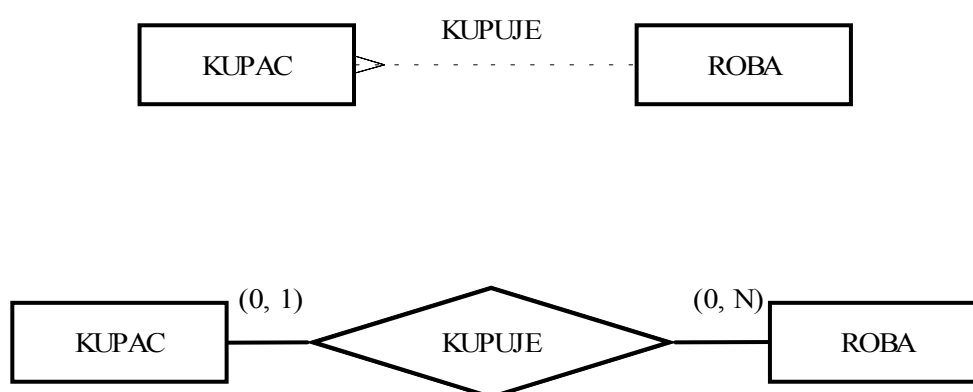
Sl. 7.5.

7.5.1.2. Kardinalitet grupe 1:N

Za ovu grupu kardinaliteta važi da su maksimalni kardinaliteti $b_1 = N$ i $b_2 = 1$. Kombinacija minimalnih kardinaliteta i u ovom slučaju daje četiri različite veze.

1. $a_1 = 0$ i $a_2 = 0$ [R ($E_1 (0, 1) : E_2 (0, N)$)]

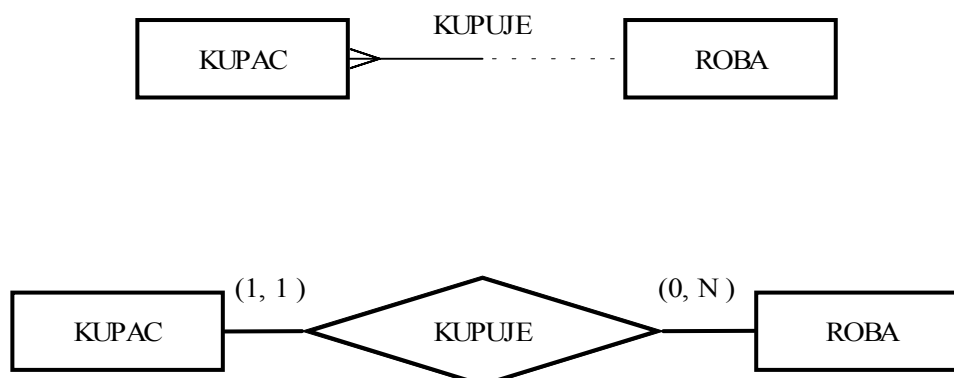
U ovom slučaju svaki entitet tipa entiteta E_1 ne mora biti povezan ni sa jednim, a može biti povezan sa najviše jednim entitetom iz klase E_2 . Entiteti klase E_2 ne moraju biti povezani ni sa jednim entitetom, mogu biti povezani sa jednim ili sa više entiteta klase E_1 . Na jeziku grafike ova veza je formulisana na sl 7.6.



Sl. 7.6.

2. $a_1 = 0$ i $a_2 = 1$ [R ($E_1 (1, 1) : E_2 (0, N)$)]

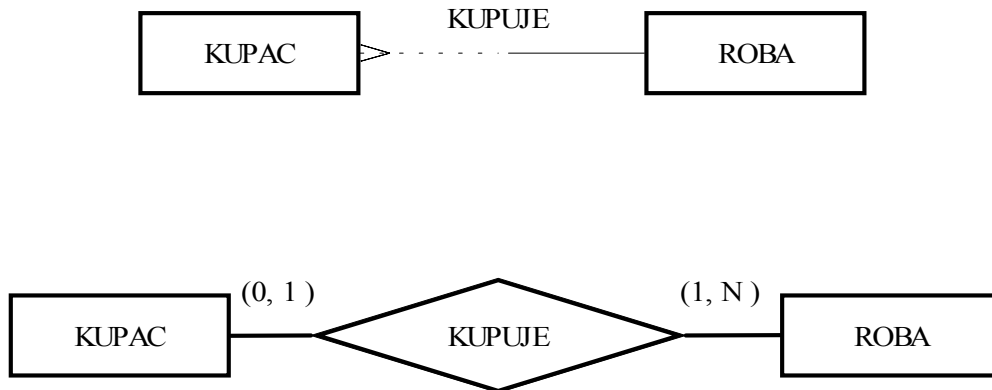
Za ovu vezu je karakteristično, da svaki entitet klase E_1 mora biti povezan sa jednim i samo jednim entitetom iz klase entiteta E_2 , a entiteti tipa entiteta E_2 ne moraju biti povezani ni sa jednim entitetom, mogu biti povezani sa jednim ili sa više entiteta klase E_1 . U grafičkoj formi ova veza je predstavljena na sl 7.7.



Sl. 7.7.

3. $a_1=1$ i $a_2=0$ $[R(E_1(0,1):E_2(1,N))]$

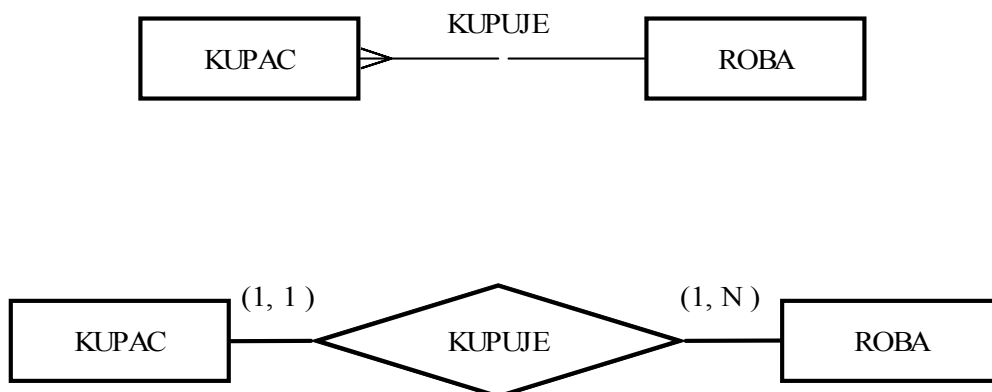
Semantika ove veze je sledeća. Svaki entitet tipa entiteta E_1 ne mora biti povezan ni sa jednim, a može biti povezan sa najviše jednim entitetom iz klase E_2 , a istovremeno svaki entiteti klase E_2 mora biti povezan sa barem jednim, a može biti povezan sa većim brojem entiteta klase E_1 . Kao što se vidi sada su entiteti klase E_2 egzistencijalno zavisni od postojanja veze sa entitetima klase E_1 . Odgovarajući grafik je na sl. 7.8.



Sl. 7.8.

4. $a_1=1$ i $a_2=1$ $[R(E_1(1,1):E_2(1,N))]$

Ova vrsta veze se rečima može opisati na sledeći način. Svaki entitet klase E_1 mora biti povezan sa jednim i samo jednim entitetom klase E_2 , a entiteti tipa entiteta E_2 moraju biti povezani bar sa jednim (a mogu i sa više) entitetom klase E_1 . Ovde su entiteti obe povezane klase su egzistencijalno uslovljeni postojanjem barem jedne konkretne veze sa entitetima druge klase. Ovu formulaciju predstavlja grafik 7.9.



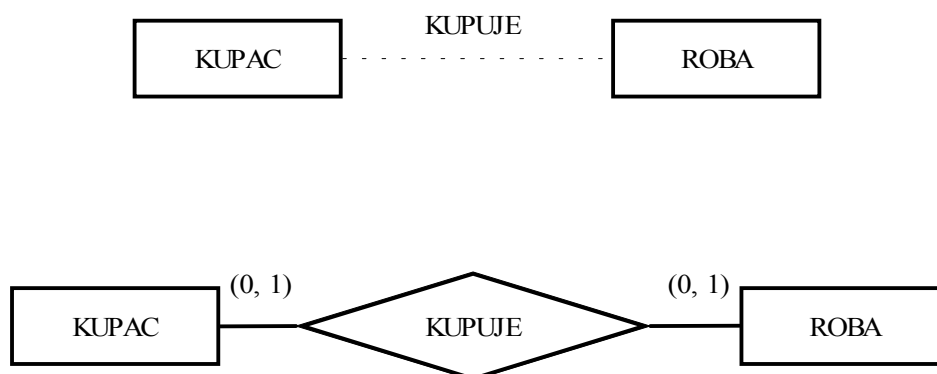
Sl. 7.9.

7.5.1.3. Kardinalitet grupe 1:1

Kod ove grupe kardinaliteta su maksimalni kardinaliteti isti i imaju vrednost $b_1 = b_2 = 1$, i kao kod prethodne dve grupe kardinaliteta kombinacije dva minimalna kardinaliteta rezultuju četiri moguće konkretne veze.

$$1. \mathbf{a_1=0 \text{ i } a_2=0} \quad [R(E_1(0,1):E_2(0,1))]$$

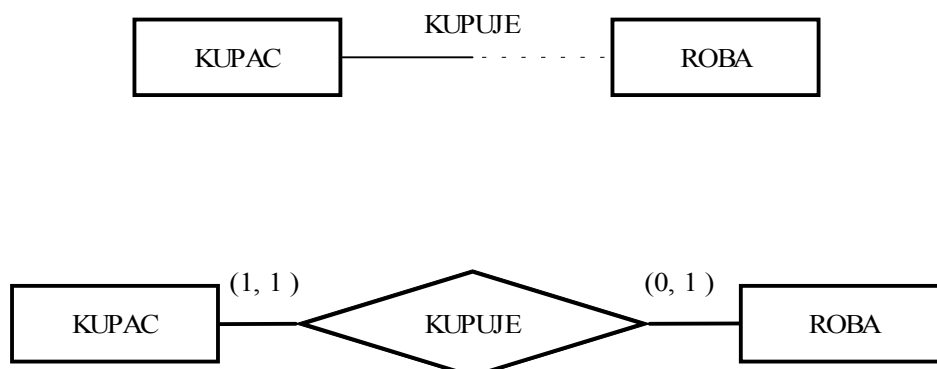
Ova veza predstavlja situaciju kada svaki entitet jedne klase može biti povezan sa najviše jednim entitetom druge klase. Drugim rečima svaki entitet tipa entiteta E_1 ne mora biti povezan ni sa jednim, a može biti povezan sa najviše jednim entitetom iz klase E_2 , a to važi i za entitete klase E_2 , koji ne moraju biti povezani ni sa jednim, a mogu biti povezani sa najviše jednim entitetom iz klase E_1 . Slika sl. 7.10. prikazuje ovu vezu.



Sl. 7.10.

$$2. \mathbf{a_1=0 \text{ i } a_2=1} \quad [R(E_1(1,1):E_2(0,1))]$$

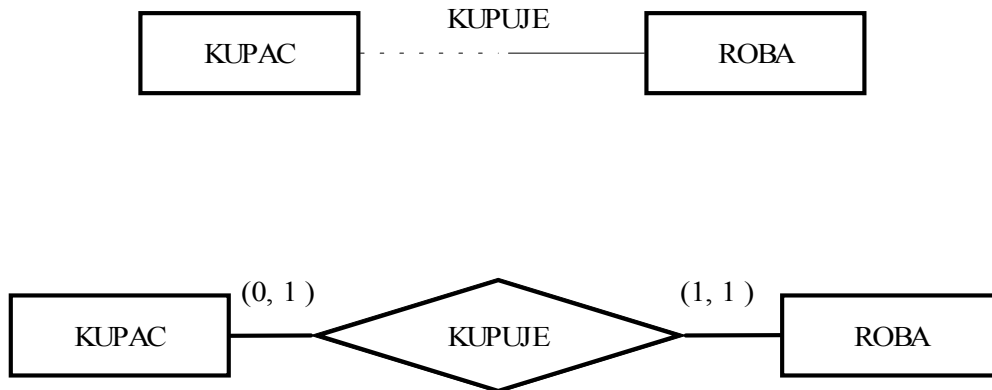
Svaki entitet klase E_1 za ovu vezu mora biti povezan sa jednim i samo jednim entitetom iz klase entiteta E_2 , a konkretni entiteti tipa entiteta E_2 ne moraju, a mogu biti povezani sa najviše jednim entitetom klase E_1 . Crtež na sl. 7.11. prikazuje ovu vezu.



Sl. 7.11.

3. $a_1=1$ i $a_2=0$ $[R(E_1(0,1):E_2(1,1))]$

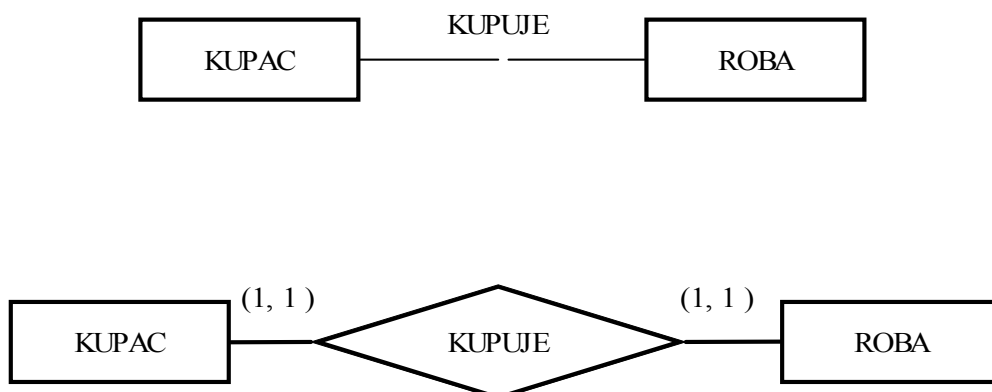
Semantiku ove veze je moguće formulisati na sledeći način. Svaki entitet tipa entiteta E_1 ne mora biti povezan ni sa jednim, a može biti povezan sa najviše jednim entitetom iz klase E_2 , a istovremeno svaki entiteti klase E_2 mora biti povezan sa barem jednim i samo jednim entitetom iz klase E_1 . Egzistencijalna zavisnost entiteta klase E_2 od postojanja veze sa entitetima klase E_1 je očigledna. Odgovarajući grafik je na sl. 7.12.



Sl. 7.12.

4. $a_1=1$ i $a_2=1$ $[R(E_1(1,1):E_2(1,1))]$

Poslednja vrsta veze u ovoj grupi se rečima može opisati na sledeći način. Svaki entitet klase E_1 mora biti povezan sa jednim i samo jednim entitetom klase E_2 , a isto to važi i za entitete klase E_2 , tj. svaki entitet klase E_2 mora biti povezani bar sa jednim i samo jednim entitetom klase E_1 . Ovde je prisutna egzistencijalna uslovljenost svih entiteta obe povezane klase od postojanja barem jedne konkretne veze sa entitetima druge klase. Slikovito predstavljanje ove veze se nalazi na sl. 7.13.



Sl. 7.13.

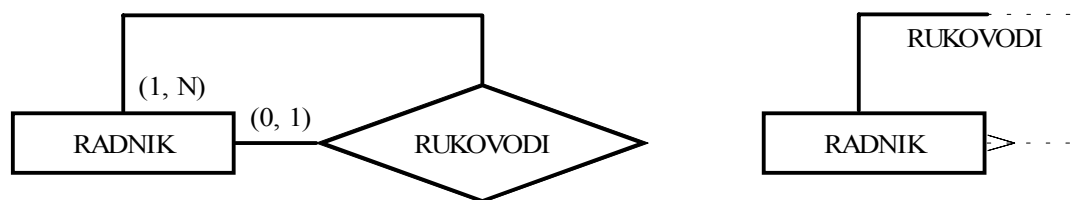
7.5.2. Rekurzivne veze

Odnosi izmedju različitih entiteta istoga tipa se na konceptualnom nivou apstrakcije nazivaju rekurzivnim vezama. Apstrakcijom konkretnih odnosa izmedju entiteta istoga tipa se dolazi do pojma rekurzivnog tipa poveznika. Kod rekurzivnog tipa poveznika (ako se upoređuje sa »klasičnim tipom poveznika«) isti tip entiteta predstavlja i prvu i drugu klasu entiteta, pri čemu jedna klasa obično predstavlja podskup druge klase. Kardinaliteti rekurzivnog tipa poveznika mogu uzimati sve moguće vrednosti upoznatih u tački 7.5.1.

Kao primer za rekurzivni tip poveznika može da posluži tip entiteta radnik i rekurzivni tip poveznika Rukovodi sa kardinalitetima:

Rukovodi (Radnik - "podređeni" (0,1) : Radnik - "nadređeni" (1, N))

Grafička predstava rekurzivnog tipa poveznika Rukovodi je na sl. 7.14.



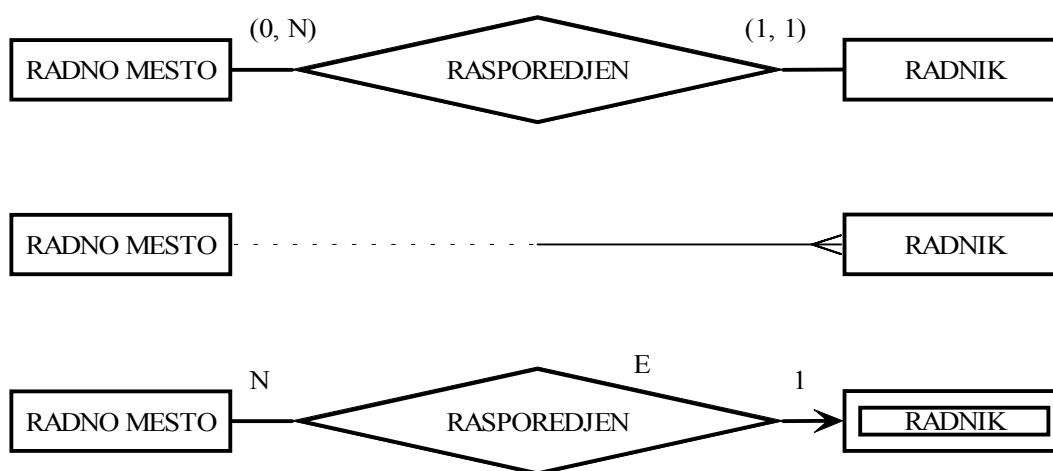
Sl. 7.14.

7.5.3. Slabi tip entiteta

Na dijagramima tipova entiteta i tipova poveznika (ER – dijagrami po engleskim rečima Entity i Relationship) ponekad se zadaju samo maksimalne vrednosti kardinaliteta. Na takvim crtežima egzistencijalno zavisani tip entiteta (onaj koji ima minimalnu kardinalnost 1) se označava dvostrukim pravougaonikom i naziva se slabi tip entiteta. Linija koja povezuje tip poveznika (koji ima oznaku E za egzistencijalnu zavisnost) sa slabim tipom entiteta je usmerena ka slabom tipu entiteta. Često se i tip poveznika koji povezuje slabi tip entiteta sa drugim tipom entiteta naziva slabim tipom poveznika. Egzistencijalno nezavisni tipovi entiteta i njihovi tipovi poveznika se smatraju regularnim i ponekad se to posebno ističe u cilju razlikovanja od slabih tipova entiteta i poveznika. Takav primer ilustruje sl. 7.15. na nama svim poznatim "jezicima" grafičke predstave.

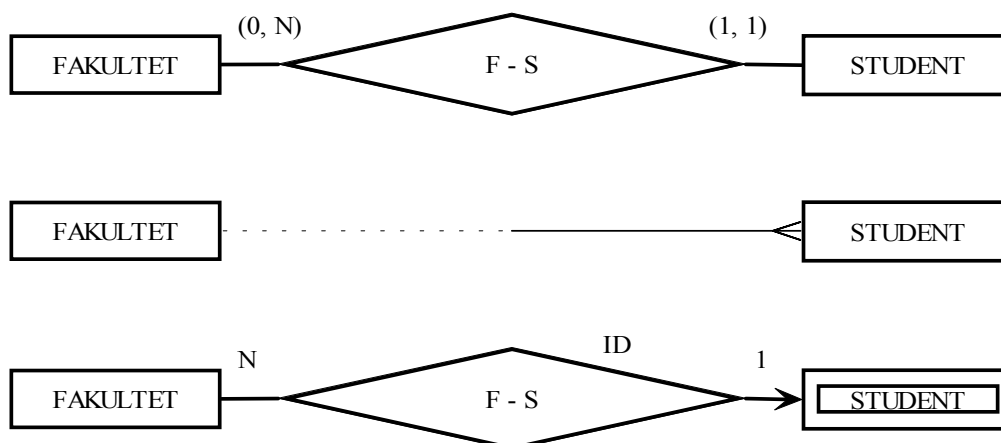
7.5.4. Identifikaciono zavisan tip entiteta

Identifikaciono zavisan tip entiteta predstavlja specijalan slučaj slabog tipa entiteta. Ako slabi tip entiteta nema svoj (jedinstveni) identifikator, nego se identifikuje pomoću identifikatora egzistencijalno nezavisnog entiteta sa kojim je u neposrednoj vezi, onda se on zove identifikaciono zavisni tip entiteta. Pojave tipa identifikaciono zavisnog entiteta se ne mogu jednoznačno identifikovati pomoću vrednosti nekog svog obeležja, već im treba pridružiti identifikator regularnog tipa entiteta sa kojim ga povezuje identifikaciono zavisan tip poveznika. Identifikaciona zavisnost uvek znači i egzistencijalnu zavisnost, a to obrnuto ne mora automatski važiti.



Sl. 7.15.

Pretpostavimo primer studenta koji ima za svoju identifikaciju broj indeksa na svom matičnom fakultetu, no to ga ne može identifikovati na nivou države. Na nivou države njegov identifikator može biti broj indeksa i identifikator (šifra) matičnog fakulteta koji je jedinstven na nivou države (sl. 7.16.).



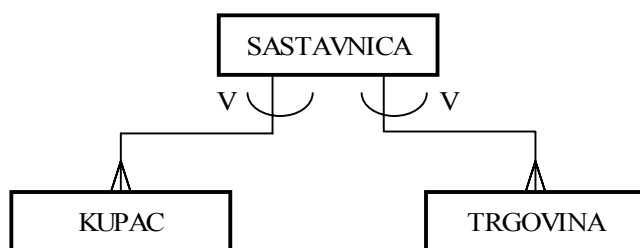
Sl. 7.16.

7.5.5. Veza izmedju tipova poveznika

Pored osobine tipa poveznika da ima minimalnu i maksimalnu kardinalnost, treba imati na umu da tipovi poveznika mogu i medjusobno biti povezani. Može se desiti da dva ili više tipova poveznika jedan drugog medjusobno isključuju. Veza izmedju tipova poveznika može biti:

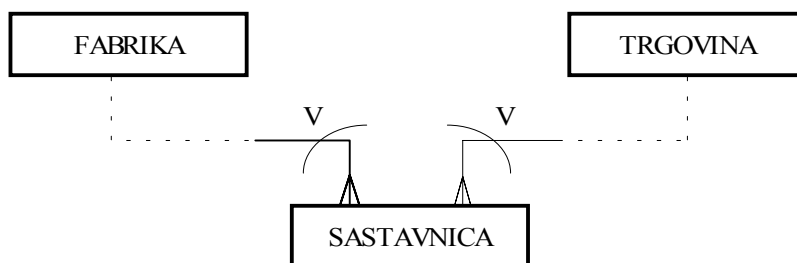
1. isključiva podređena i
2. isključiva nadređena veza.

Za ilustraciju isključive podređene veze može da posluži sledeća formulacija realizacije (prodaje) sastavnih delova proizvoda jedne fabrike. Fabrika konkretnu sastavnicu (deo proizvoda, poluproizvod ili gotov proizvod) prodaje direktno kupcu (preko veleprodaje ili maloprodaje) ili trgovini (za daljnju preprodaju). Taj primer je prikazan na sl. 7.17.



Sl. 7.17.

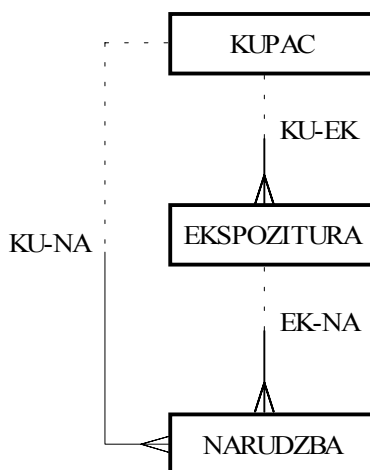
Za ilustraciju druge vrste veze izmedju tipova poveznika posmatrajmo situaciju preduzeća koji samo slapa (a ne proizvodi) sastavnice. Praćenjem izvora nabavke sastavnica može se doći do sledeće konstrukcije tipova entiteta i poveznika u slučaju da znamo da preduzeće sastavnice nabavlja ili direktno iz fabrike ili u trgovini (sl. 7.18.).



Sl. 7.18.

7.5.6. Tranzitivna konstrukcija tipova poveznika

Pretpostavimo situaciju koja je svakidašnja u realnom svetu: medju kupcima neke fabrike postoje i takvi koji imaju svoje ekspoziture (velike trgovačke kompanije sa lancem robnih kuća), tj. konkretna prodajna mesta, koja pored centrale mogu podnositi narudžbe (tj. naručiti robe) fabrici, odnosno proizvođaču. Grafik sa tipovima entiteta i poveznika se za pomenuti primer nalazi na sl.7.19.



Sl. 7.19.

U ovom slučaju treba ispitati da li tranzitivna (“premošćena”) konstrukcija ima semantičko značenje ili ne. Ako nema, onda predstavlja redundanciju i tranzitivni tip poveznika treba ukinuti (udaljiti).

Pretpostavljajući jednu konkretizaciju tipa entiteta KUPAC, skup konkretizacija tipa entiteta NARUDZBA, koji po tipu poveznika KU-NA odgovara konkretnom kupcu označimo sa S_1 . Ako za istog konkretnog kupca potražimo sve konkretizacije tipa entiteta EKSPozITURA po tipu obeležja KU-EK, skup konkretnih narudžbi koji odgovara svim konkretnim ekspoziturama izabranog kupca označimo sa S_2 . Ako su dva skupa ekvivalentna, onda je tranzitivni tip poveznika (KU-NA) suvišan za izabranu konkretizaciju tipa entiteta KUPAC. Ako prethodni uslov (da su skupovi S_1 i S_2 jednaki) važi za sve moguće konkretizacije tipa entiteta KUPAC tranzitivni tip poveznika KU-NA treba ukinuti.

7.6. INTERNA STRUKTURA TIP A ENTITETA

Odredjivanje eksterne strukture entiteta se zasniva na otkrivenim entitetima i njihovim medjusobnim vezama i to najčešće na samom početku analize realnog sistema (u toku upoznavanja sa pojmovima i pojavama koje želimo pratiti i opisati obeležjima – podacima). Pošto se analiza ili upoznavanje realnog sistema vrši polazeći od celine, izgradnja eksterne strukture entiteta se zove i kao top-down pristup izgradnje logičkog modela podataka.

Upoznavajući detalje, projektanti se upoznaju i sa obeležjima u realnom i informacionom sistemu. Ispitivanjem medjuzavisnosti izmedju obeležja imajući u vidu opšte prihvaćene preporuke o grupisanju tih obeležja (normalizacija) moguće je doći do samih entiteta i njihovih medjusobnih veza. Logički model podataka se može izgraditi i izgradnjom interne strukture entiteta, tj. bottom-up načinom izgradnje.

Logički model podataka dobijen izgradnjom eksterne strukture entiteta treba da se slaže sa onim koji se dobije na osnovu normalizacije (interne strukture entiteta) jer se odnose na isti informacioni sistem. Razlike (ako postoje) ukazuju na potrebu revizije i izrade konačnog i preciznog logičkog modela podataka.

7.6.1. Zavisnosti izmedju obeležja

Naj poznatija zavisnost izmedju obeležja je funkcionalna zavisnost koja može biti potpuna ili nepotpuna. Za nas će biti važna još jedna, tzv. tranzitivna zavisnost, jer pomoću ove tri zavisnosti se može realizovati treća normalna forma, prihvatljiva organizacija interne strukture entiteta. Obeležja organizovana u grupe koje su u trećoj normalnoj formi mogu se smatrati entitetima.

Obeležje entiteta E_1 , u oznaci Y je u funkcionalnoj zavisnosti (FD – functional dependence) od obeležja X istog entiteta samo u tom slučaju ako se u svim pojavama tipa entiteta uz sve vrednosti obeležja X pojavljuje samo jedna vrednost obeležja Y . Izraz oblika $f: X \rightarrow Y$, ili, skraćeno $X \rightarrow Y$ predstavlja funkcionalnu zavisnost, a taj izraz se rečima može formulisati na sledeći način: X funkcionalno određuje Y , ili, Y funkcionalno zavisi od X . U opštem slučaju X i Y predstavljaju skupove obeležja. Funkcionalna zavisnost se može opisati i na drugi način: svakom elementu domena X (svakoj konkretizaciji obeležja X) se može pridružiti najviše jedan elemenat iz domena Y (jedna konkretizacija obeležja Y).

Funkcionalna zavisnost u obrnutom smeru, u opštem slučaju označava nezavisnost, odnosno u slučaju $X \rightarrow Y$ važi da $Y \not\rightarrow X$. Primetimo da funkcionalna zavisnost definiše hijerarhiju između obeležja.

Primer: $\text{Matični_broj_studenta} \rightarrow \text{Ime_i_prezime_studenta}$, gde između konkretizacija (vrednosti domena) ova dva obeležja postoji veza tipa N:1. Veći broj studenata sa istim imenom i prezimenom imaju različite matične brojeve (N), a svakom matičnom broju odgovara samo jedno ime studenta (1).

Funkcionalna zavisnost može biti jaka, kada svakoj »levoj strani«, tj. X-u mora postojati »desna strana«, tj. Y, ili slaba, kada postoje X-evi za koje ne postoje Y-i.

Primer: funkcionalna zavisnost $\text{Matični_broj_studenta} \rightarrow \text{Ime_i_prezime_studenta}$ je jaka, jer svakom matičnom broju studenta uvek pripada ime i prezime studenta, a $\text{Matični_broj_studenta} \rightarrow \text{Vrsta_primljenog_kredita}$ predstavlja slabu funkcionalnu zavisnost, pošto ima studenata koji ne primaju nikakav kredit.

Funkcionalna zavisnost može biti potpuna (full dependence) ili nepotpuna (partial dependence). Ako se na levoj strani zavisnosti nalazi obeležje, a ne skup obeležja, za zavisnost se kaže da je elementarna, u suprotnom se radi o složenoj zavisnosti. Klasifikacija funkcionalnih zavisnosti na potpune i nepotpune se primenjuje samo na složene zavisnosti. Potpuna je složena funkcionalna zavisnost u tom slučaju kad desna strana (obeležje na desnoj strani) zavisi od kompletne leve strane (celog skupa obeležja na levoj strani). To znači da odstranjivanjem bilo kojeg obeležja iz skupa obeležja na levoj strani kod potpune zavisnosti uništimo funkcionalnu zavisnost. Ako izvlačenjem bilo kojeg obeležja iz skupa obeležja na levoj strani složene funkcionalne zavisnosti automatski ne dolazi i do uništavanja funkcionalne zavisnosti, onda se radi o nepotpunoj funkcionalnoj zavisnosti.

Primer: posmatrajmo entitet POLOŽENI_ISPITI, i dve funkcionalne zavisnosti u okviru njega:

$\text{POLOŽENI_ISPITI} (\text{Matični_broj_studenta} + \text{Šifra_predmeta}, \text{ocena}, \dots, \text{Naziv_predmeta})$

$\text{Matični_broj_studenta} + \text{Šifra_predmeta} \rightarrow \text{ocena}$

$\text{Matični_broj_studenta} + \text{Šifra_predmeta} \rightarrow \text{Naziv_predmeta}$

Očigledno je da je prva funkcionalna zavisnost potpuna, pošto ni prvo ni drugo obeležje na levoj strani ponaosob ne određuju desnu stranu. Druga funkcionalna zavisnost je nepotpuna, jer izbacivanjem matičnog broja studenta zavisnost ostaje dalje na snazi pošto šifra predmeta određuje naziv predmeta.

Semantički funkcionalna zavisnost $f: X \rightarrow Y$ predstavlja vremenski promenljivu funkciju. To znači da se leva strana (domen) i desna strana (kodomen) funkcionalne zavisnosti menjaju u

vremenu, odnosno da u različitim trenucima, istoj vrednosti iz domena X , f može pridružiti različite vrednosti iz domena Y . Kao primer za prethodnu konstataciju može da posluži funkcionalna zavisnost $REG_BR_VOZILA \rightarrow BR_OBAVEZNIH_SERVISA$ u kojoj se pridružena vrednost levoj strani menja u vremenu: broj obavljenih obaveznih servisa raste u toku životnog ciklusa automobila.

Funkcionalna zavisnost predstavlja intenzionalno ograničenje, jer se definiše nad skupom obeležja. Jedini način za utvrđivanje funkcionalnih zavisnosti je pažljiva analiza realnog sveta i pojmova u njemu, tj. analiza osobina entiteta, formulisanje njegovih obeležja i njihovih domena.

Obeležje ili grupa obeležja koja za svaku konkretizaciju posmatranog tipa entiteta uzima različite vrednosti nazivamo identifikatorom ili ključem, eventualno kandidatom za ključ. Identifikator treba da je minimalne konstrukcije. To drugim rečima znači, da ne postoji nijedan pravi podskup kandidata za ključ koji sam može biti kandidat za ključ (sve zavisnosti u kojima figuriše ključ-identifikator moraju biti potpune).

Ako u nekom tipu entiteta postoji opisno, neidentifikujuće obeležje koje je u drugom tipu entiteta identifikator, tj. ključ, onda ga zovemo spoljnim ključem.

Ako se ključ sastoji od jednog obeležja, onda se zove prost ključ. U slučaju da se ključ sastoji od više spoljnih ključeva, zove se složeni ključ. Ako se ključ sastoji od jednog ili više spoljnih ključeva i od barem jednog obeležja koje pripada tipu entiteta onda se naziva hijerarhijski ključ.

Tranzitivna zavisnost se definiše na sledeći način. Označimo skup obeležja (atributa) nekog tipa entiteta sa A , i neka postoje skupovi obeležja X i Y ($X, Y \subseteq A$) i $X \rightarrow Y$. Kaže se da Y tranzitivno zavisi od X , ako postoji takav skup obeležja $Z \subseteq A$, da $X \rightarrow Z$ i $Z \rightarrow Y$, ali tako da ne važi $Z \rightarrow X$ i $Y \rightarrow Z$.

Jednostavnija formulacija tranzitivne zavisnosti je sledeća. Neključno obeležje Y posmatranog tipa entiteta tranzitivno zavisi od ključa tipa entiteta X samo u tom slučaju ako ga (obeležje Y) funkcionalno određuje i neko neključno obeležje Z , koje je, naravno, funkcionalno zavisno od ključa tipa entiteta X .

Dekompozicija je proces u kojem se entitet sa nepogodnom internom strukturom deli (dekomponuje) na dva ili više entiteta pogodnije ili bolje unutrašnje strukture.

Neka skup obeležja tipa entiteta E_1 bude A , i neka postoje podskupovi $X \subseteq A$ i $Y \subseteq A$ tako, da je $X \cup Y = A$. Kaže se da X i Y predstavljaju dekompozicioni par u tipu entiteta, ako iz egzistencije (postojanja) dve konkretizacije tipa entiteta, recimo za pojave t_i i t_j ($t_i = \text{ext}(E_1)$ i $t_j = \text{ext}(E_1)$), to su redovi t_i i t_j u tabeli 7.1.) za koje je ispunjen uslov $t_i = t_j$ na preseku dva skupa

$X \cap Y$, sledi da postoji takva konkretizacija tipa entiteta (red t) za koju $t=t_i$ na skupu X , a $t=t_j$ na skupu Y . Definicija se može formalno napisati na sledeći način:

Ako $\forall t_i, t_j$ takvi da $t_i(X \cap Y) = t_j(X \cap Y)$ i $\exists t$ takav da $t(X) = t_i(X)$ i $t(Y) = t_j(Y)$,

onda X i Y predstavljaju dekompozicioni par u tipu entiteta E_1 .

Primer: Neka je $X = \{A_1, A_2, A_3, A_4\}$ i $Y = \{A_3, A_4, A_5, A_6\}$, tj $X \cap Y = \{A_3, A_4\}$, a entitet E_1 ima ekstenzije prikazane u sledećoj tabeli 7.1.

Tabela 7.1.

	A₁	A₂	A₃	A₄	A₅	A₆
t₁	a	b	c	d	e	f
t₂	g	f	c	d	h	e
t₃	a	b	c	d	h	e
t₄	.	.	.	.	.	.

Na osnovu prikazane ekstenzije u tabeli za tip entiteta E_1 se može zaključiti da X i Y predstavljaju dekompozicioni par, jer

1. $t_1(X \cap Y) = t_2(X \cap Y) = \{c, d\}$ i
2. $t_3(X) = t_1(X)$ i $t_3(Y) = t_2(Y)$

7.6.2. Normalne forme

Normalne forme odražavaju stanje (kvalitet) unutrašnje strukture tipa entiteta. Postoji veći broj normalnih formi, koji se označavaju u obliku xNF, u kojem se na mestu x-a nalazi broj, a NF potiče od skraćenice engleskih reči Normal Form. Da bi ukazali na jedini izuzetak u označavanju dajemo spisak svih normalnih formi od najslabije do najkvalitetnije: 0NF, 1NF, 2NF, 3NF, BCNF (Boyce-Codd-ova normalna forma), 4NF i 5NF.

Normalizacija je nastala kao matematički metod za poboljšanje interne strukture entiteta, ali je neminovno ukazati i na potrebu primene paralelne semantičke analize (greške u shvatanju i razumevanju fizičke realnosti i njenom strukturiranju) uz izvršavanje koraka normalizacije. Normalizacija ili izgradnja optimalne unutrašnje strukture tipova entiteta neminovno utiče i na spoljašnju strukturu tipova entiteta, a sama izgradnja logičke strukture podataka se može realizovati metodom dekompozicije i metodom sinteze. U izlaganjima se prikazuje metoda dekompozicije kao sredstvo za realizaciju normalizacije.

Tip entiteta sadrži ponavljajuća saznanja ako sadrži takva obeležja koja nisu funkcionalno zavisna od identifikatora tipa entiteta.

Tip entiteta je u nenormalizovanoj formi (0NF) ako u njemu postoji bar jedno takvo neključno obeležje koje nije funkcionalno zavisno od njegovog identifikatora.

Primer: STUDENT (Br_indeksa, Ime_i_prz, ..., Šifra_pr, Naziv_pr, Datum_ispita, Ocena). Tip entiteta student je prikazan i grafički na sl. 7.20. Obeležja šifra predmeta (Šifra_pr), naziv predmeta (Naziv_pr, datum ispita i ocena u nekim redovima (pojavama tipa entiteta) mogu uzimati više vrednosti. Identifikator tipa entiteta STUDENT je Br_indeksa.

STUDENT

Br indeksa	Ime i prz	...	Šifra pr	Naziv pr	Datum ispita	Ocena
E494	Marko Marković		112	Matematika	12.09.2000.	9
			147	Fizika	15.09.2000.	7
			174	Elektrotehnika	05.10.2000.	6
E495	Petar Petrović		174	Elektrotehnika	14.07.2000.	8
			112	Matematika	15.09.2000.	8
...	...		...	...	...	...

Sl. 7.20.

Tip entiteta se sa sigurnošću nalazi barem u prvoj normalnoj formi (1NF) ako svako neključno obeležje funkcionalno zavisno od identifikatora (u svakoj pojavi tipa entiteta za svaku konkretizaciju obeležja stoji tačno jedna vrednost).

U cilju dovodjenja tipa entiteta na 1NF, nužno je iz tipa entiteta odstraniti sva obeležja koja nisu funkcionalno zavisna od identifikatora (imaju višestruke vrednosti). Prvobitni tip entiteta se dekomponuje na dva nova tipa entiteta: u prvom ostaju obeležja bez višestrukih vrednosti, a ona sa višestrukim vrednostima se smeštaju u drugi tip entiteta dopunjujući ih sa identifikatorom prvobitnog tipa entiteta. Tu dekompoziciju prikazuje sl. 7.21.

STUDENT

Br indeksa	Ime i prz	...
E494	Marko Marković	
E495	Petar Petrović	
...	...	

POLOŽENI ISPITI

Br indeksa	Šifra pr	Naziv pr	Datum ispita	Ocena
E494	112	Matematika	12.09.2000.	9
E494	147	Fizika	15.09.2000.	7
E494	174	Elektrotehnika	05.10.2000.	6
E495	174	Elektrotehnika	14.07.2000.	8
E495	112	Matematika	15.09.2000.	8
...	...	...	...	...

Sl. 7.21.

Novi tip entiteta je dobio naziv POLOŽENI_ISPITI i ima složeni identifikator (ključ) sastavljen iz dva obeležja, a ona su: Br_indeksa i Šifra_pr. Identifikator prvobitnog tipa entiteta se nije menjao.

Tip entiteta je minimalno u drugoj normalnoj formi (2NF) ako se nalazi u 1NF i ako je svako neključno obeležje potpuno funkcionalno zavisno od identifikatora.

Na osnovu prethodne definicije se mogu izvući dva zaključka:

1. ako tip entiteta ima prost identifikator onda je sigurno u 2NF i
2. ako tip entiteta nema neključna (neidentifikujuća) obeležja, takodje je sigurno, da se nalazi u 2NF.

U procesu dovodjenja tipa entiteta na 2NF prvo se otkrivaju nepotpune funkcionalne zavisnosti da bi se posle te zavisnosti uklonile. Algoritam dovodjenja tipa entiteta na 2NF se sastoji od tri koraka:

1. iz identifikatora tipa entiteta se izdvajaju ona obeležja (grupe obeležja) koja i sama funkcionalno određuju neidentifikujuća (neključna) obeležja,
2. obeležja izabrana u tački 1. sa njima pripadajućim neključnim obeležjima (koja su od njih funkcionalno zavisna) se izdvajaju u posebne tipove entiteta, a
3. neključna obeležja koja su u potpunoj funkcionalnoj zavisnosti od (celokupnog) identifikatora u prvobitnom tipu entiteta, ostaju na svom starom mestu, tj. u prvobitnom tipu entiteta.

STUDENT

Br indeksa	Ime i prz	...
E494	Marko Marković	
E495	Petar Petrović	
...	...	

PREDMET

Šifra_pr	Naziv_pr
112	Matematika
147	Fizika
174	Elektrotehnika
174	Elektrotehnika
112	Matematika
...	...

POLOŽENI ISPITI

Br indeksa	Šifra_pr	Datum ispita	Ocena
E494	112	12.09.2000.	9
E494	147	15.09.2000.	7
E494	174	05.10.2000.	6
E495	174	14.07.2000.	8
E495	112	15.09.2000.	8
...	...	...	...

Pošto tip entiteta STUDENT ima prost identifikator, dovodjenje na 2NF se može prikazati na tipu entitete POLOŽENI_ISPITI. U tom tipu entiteta deo identifikatora Šifra_pr, odnosno šifra predmeta funkcionalno određuje Naziv_pr, ili naziv predmeta, što znači da njih treba izdvojiti u novi tip entiteta PREDMET. Tipovi entiteta u drugoj normalnoj formi se nalaze na sl. 7. 22.

Tip entiteta se garantovano nalazi bar u trećoj normalnoj formi (3NF), ako je u 2NF i ako nijedno neidentifikujuće (neključno) obeležje nije tranzitivno zavisno od identifikatora tipa entiteta.

O unutrašnjoj strukturi tipa entiteta se može tvrditi da su u njoj sva opisna (neidentifikujuća) obeležja zavisna od identifikatora (ključa – 1NF), i to potpuno funkcionalno zavisna od (celokupnog složenog) ključa (2NF), a da su funkcionalno nezavisna od ostalih opisnih obeležja (3NF).

Dovodjenje tipa entiteta u treću normalnu formu se sastoji od izdvajanja obeležja koja su tranzitivno zavisna od identifikatora u novi tip entiteta. Tako prvobitni tip entiteta neće sadržati tranzitivno zavisna obeležja. Novi tipovi entiteta osim tranzitivno zavisnih obeležja u prvobitnom tipu entiteta, sadržaće i ona opisna obeležja iz prvobitnog tipa entiteta koja ih funkcionalno određuju.

Primer za dovodjenje tipa entiteta na 3NF je uzet iz oblasti trgovine. Konkretno se radi o evidenciji narudžbe. Tip entiteta NARUDŽBA koji je u drugoj normalnoj formi, se dekompozicijom pretvara u 3NF (sl. 7.23).

NARUDŽBA

Br_narudžbe	Šifra_kupca	Naziv_kupca	Datum_nar
17494	112	Carnex	12.09.2000.
15944	147	Pionir	15.09.2000.
13675	174	Sever	05.10.2000.
19395	174	Sever	14.07.2000.
25495	112	Carnex	15.09.2000.
...	...		...

NARUDŽBA 1

Br_narudžbe	Šifra_kupca	Datum_nar
17494	112	12.09.2000.
15944	147	15.09.2000.
13675	174	05.10.2000.
19395	174	14.07.2000.
25495	112	15.09.2000.
...	...	...

KUPAC

Šifra_kupca	Naziv_kupca
112	Carnex
147	Pionir
174	Sever
...	

Proces normalizacije (dovodjenje unutrašnje strukture entiteta na sve bolje normalne forme) je realizovan metodom dekompozicije, podelom prvobitnog na dva ili više nova entiteta. Normalizacija omogućuje postizanje sve savršenije interne strukture entiteta.

Dekompozicija se ostvaruje projekcijom (odstranjivanjem) obeležja koja nisu funkcionalno zavisna od identifikatora (1NF), koja su nepotpuno (delimično) zavisna od složenog identifikatora (2NF), ili, pak zavise i od drugih opisnih (neključnih) obeležja pored postojanja njihove funkcionalne zavisnosti od identifikatora (3NF) u nove tipove entiteta. Postavlja se, međjutim, specijalan zahtev za dekompoziciju: mora predstavljati reverzibilan proces, odnosno da se na osnovu novodobijenih tipova entiteta uvek može rekonstruisati prvobitna struktura. Takva dekompozicija se zove dekompozicija bez gubitaka (non-loss decomposition).

Aktivnost suprotna projekciji je spoj. Uslov spajanja dva tipa entiteta je postojanje istog obeležja u oba tipa entiteta. Uobičajeno je da se od spoja takodje traži reverzibilnost, tj. da realizuje novi spajani tip entiteta bez gubitaka informacija u njima (non-loss join).

7.6.3. Eksterna struktura tipa entiteta na osnovu normalizovane interne strukture

Normalizacija, odnosno izgradnja optimalne interne strukture tipa entiteta utiče i na njegovu eksternu strukturu. Na osnovu tipa entiteta u 3NF se lako može konstruisati logički model podataka (transverzije ili eksterna struktura entiteta) ako imamo u vidu činjenice izložene u prethodnim izlaganjima:

1. spoljni ključevi određuju nadredjene tipove entiteta (tip poveznika ima kardinalnost 1:N – tip entiteta u kojem je obeležje spoljni ključ na strani N, a tip entiteta u kojem je isto obeležje identifikator na strani 1),
2. složeni ključevi se sastoje od spoljnih ključeva i
3. hijerarhijski ključ sadrži barem jedan spoljni ključ.

7.7.MODEL PODATAKA, BAZA PODATAKA

Saznanje obuhvata konkretne osobine (konkretizacije obeležja) o konkretnim pojavama (entiteti) i njihovim konkretnim međusobnim odnosima (poveznici). Zadatak informacionog

sistema je da sakuplja, smešta, rukuje i prikaže konkretne podatke o konkretnim osobinama i vezama između konkretnih entiteta (pojave tipova entiteta).

Zadatak informacionog sistema je i taj da sadrži informaciju ili odslikava realni sistem na pojmovnom nivou apstrakcije: o tipovima entiteta (klasama entiteta), njihovim osobinama (obeležjima) i generičkom odnosu između tipova entiteta (tipovima poveznika).

Model podataka je organizovan skup tipova entiteta, obeležja i tipova poveznika ka i njihovih ograničenja. Skup je organizovan, jer se detaljno analiziraju intenzije tipova entiteta (skup obeležja tipova entiteta) i transverzije (tipovi poveznika između tipova entiteta). To su unutrašnja i spoljašnja struktura tipa entiteta, koje impliciraju razna (strukturalna i opisna) ograničenja.

Podmodel podataka je relativno samostalan organizovani skup tipova entiteta, obeležja i tipova poveznika i njihovih međusobnih veza i ograničenja. Po sistemskom pristupu prvo treba da se definiše, oformi, izgradi model podataka, a tek posle da se odrede odgovarajući delovi, tj. podmodeli.

Baza podataka sadrži skup konkretizacija (pojava) tipova entiteta, pojava (konkretizacija) obeležja i pojava tipova poveznika organizovan po modelu podataka. U cilju realizacije baze podataka prvo treba izgraditi njenu apstraktnu sliku, tj. model podataka da bi se konkretne stvari – pojave mogle generalno srediti.

Korisnika informacionog sistema najčešće interesuje samo specifičan pogled na podatke (na primer spisak kupaca koji su kupili robu X).

Pogled (view) je apstraktna projekcija modela podataka pomoću kojeg se iz baze podataka kao celine označavaju saznanja koja interesuju korisnika. Samo označavanje je trojake prirode: intenzionalno, transverzijalno i ekstenzionalno. Korisnika, naime, interesuju samo izvesna konkretna obeležja (intenzija) nekih konkretnih tipova entiteta i njihove međusobne veze (transverzija) i to samo za neke konkretne pojave (ekstenzija).

Pogled može biti realan kada korisnik ima isti pogled na elemente modela podataka kao i projektant (na primer pogled na narudžbe od prethodnog dana sa nekim podacima kupaca).

Pogled je virtualan kada korisnik u svom zahtevu generiše entitet koji ne postoji u modelu podataka (na primer generisati pojave tipa entiteta kupac-roba za robu X, tako da se prikaže kupac i ukupna naručena količina od robe X za neki određeni period vremena).

Pogled je samo jedan deo (podskup) od celokupnog saznanja (od svih podataka) koji korisnik želi videti, i koji može da sadrži i virtualne elemente (nepostojeće tipove entiteta). Podmodel je takav realni deo celokupnog modela podataka koji je potreban za generisanje pogleda definisanog od strane korisnika.

7.8. NAČINI OBRADE SAZNANJA

Saznanja se danas (ako izuzmemo multimediju, tj. grafiku, sliku, zvuk, itd) obradjuju na dva osnovna načina:

1. činjenični (faktografski) način obrade saznanja i
2. tekstualni način obrade saznanja.

U vezi faktografskog načina obrade saznanja treba se malo vratiti nazad u istoriji. Stari Grci su “činjenice koje treba zabeležiti” zvali rekordima. Još i danas se za skup bitnih saznanja u vezi značajnih činjenica i/ili događaja od opšteg interesa koristi izraz rekord i to ne samo u našem jeziku. Ista reč se koristi u engleskom jeziku i za niz vezanih podataka na nekom računuu.

Rekordi (slogovi, zapisi) se o pojavama ili stvarima koji pripadaju istoj klasi stvari smeštaju u isti fajl (dosije ili regulator), tako da je jedna stavka je u stvari jedan rekord (slog). Rekord (slog) se sastoji od polja, a za sadržj polja se koristi pojam ili izraz podatak.

Činjenični način obrade saznanja se danas zove obrada podataka. Obrada podataka se može realizovati na dva načina:

1. klasičnom organizacijom datoteka i
2. pomoću baze podataka.

Klasična organizacija podataka, odnosno obrada na osnovu nje je prvenstveno ekstenzionalna. Nema povezanosti između tipova entiteta i njihovih pojava. Fizičko smeštanje podataka fajla je bilo najvažnije, o sadržaju, smislu, značaju i značenju podataka u fajlovima malo je ko brinuo.

Baza podataka je u stvari skup konkretizacija (pojava) tipova entiteta, pojava (konkretizacija) obeležja i pojava tipova poveznika organizovan po modelu podataka ili organizovani skup fajlova. Obrada podataka na osnovu baze podataka ne isključuje klasičnu obradu datoteka ili vertikalno kretanje u tabeli prilikom obrade (ekstenzionalnu projekciju), ali je prvenstveno transverzijalnog karaktera (više je krasi horizontalno kretanje između tabela).

Tekstualni način obrade saznanja se često meša sa pojmovima obrada teksta i editiranje teksta. Editor teksta ne može da se koristi za rukovanje podacima i za stvaranje novih podataka. Kao što znamo tekstualni fajlovi nemaju stavke – rekorde, editor ne poznaje tipove entiteta i pojave tipova entiteta, obeležja i domene, a ne mogu postojati ni tipovi poveznika i njihove pojave.

Tekstualni način obrade saznanja obuhvata dve moguće realizacije:

1. sisteme za obradu tekstova i
2. sisteme za obradu tabela.

Sistemi za obradu tekstova mogu prihvatiti “predobradjene” tekstove i memorišu naslov, autora, mesto, datum nastanka, ključne reči, itd. Poseduju dodatno proširenje u obliku rečnika koji se ovde zove tezaurus, a zasniva se na deskriptorima. Korisnik pri unosu testa može označiti koje ključne reči treba zapamtiti (odnosno njihovo mesto u tekstu), a deskriptor služi za pamćenje obeležja kojem pomenuta ključna reč pripada. Time u stvari pomoću tezaurusa sa deskriptorima možemo pronaći sve konkretizacije izabranog obeležja.

Pojedini dokumenti se u sistemima za obradu tekstova mogu, na osnovu – recimo – ključnih reči, eksplicitno grupisati u “fajl”, koje sistem zajedno rukuje i obradjuje. Time je ostvaren tip entiteta (“fajl”) i pojava tipa entiteta (dokument).

Sistemi za obradu tekstova, ipak, ne predstavljaju sistem za rukovanje bazom podataka jer nisu zasnovani na modelu podataka i ne mogu obradu realizovati na osnovu veza koje postoje između “fajlova”.

Banka podataka je skuppogodno organizovanih i celishodno smeštenih tekstualnih datoteka (dokumenata) kojima se može pristupiti pomoću obezbedjenih sistema za obradu tekstova. Povezivanje ovih dokumenata (datoteka) po sadržaju nije moguće, ali je tehnika njihove obrade ista.

Sistemi za obradu tabela se ponekad nazivaju (pogrešno) sistemima za rukovanje bazom podataka, ali to ne odgovara stvarnosti. Greška potiče možda od toga što se na logičkom nivou entiteta (pojmovni nivo) odgovara pojam relacije ili tabele. No, u klasičnoj tabeli nema obeležja – bilo koji sadržaj može da se smesti u bilo koju kolonu, bez kontrole se dopušta upis proizvoljne vrednosti u proizvoljnu kolonu. U klasičnim tabelama ne postoje entiteti, i, prema tome, ne mogu postojati ni njihove medjusobne veze.

Može se postaviti pitanje kako se mogu integrisati u bazu podataka slike, grafike, zvučni zapisi, ili saznanja obradjena na tekstualni način. Definicija modela podataka i baze podataka pominju samo tipove i pojave tipova entiteta, obeležja i njihove konkretizacije i tipove i pojave tipova poveznika, a o samom načinu predstavljanja i memorisanja ništa ne kažu. Različiti načini predstavljanja osnovnih koncepata (tipa entiteta, obeležja i tipa poveznika) su, znači, invarijantni za model podataka i bazu podataka. Na sadržajni ili pojmovni nivo projekcije podataka ne utiču različiti načini predstavljanja njegovih osnovnih koncepata. Multimedijalni zapis, konkretnije slika studenta postaće tip entiteta (SLIKA_STUDENTA) koji će biti povezan pomoću tipa poveznika sa odgovarajućim tipom entiteta (STUDENT). Student može imati više konkretnih slika, što znači da tip entiteta SLIKA_STUDENTA ima više konkretizacija ili pojava od kojih na osnovu identifikatora ili ključa (obeležje) se može izabrati najpogodnija.

Tehničko rešenje problema već nije jednoznačno. Rešenja krojena po meri su obično nepogodna, jer je struktura podataka kod njih, po pravilu, vezana na fizičkom nivou, proširenja nisu moguća, a rukovanje podacima je zacementirano u programima. Drugim rečima ni logička ni fizička nezavisnost podataka nisu ispoštovane.

Fizička (tehnička) rešenja problematike različitih načina predstavljanja osnovnih koncepata projekcije podataka se mogu realizovati i pomoću savremenih sistema za rukovanje bazama podataka (SRBP), koji su opšti (generalized) sistemi za rukovanje podacima. Imaju unapred pripremljene rutine koje su vezane za fizičku strukturu predstavljanja i memorisanje podataka, a za sadržaj i logičku strukturu ne. Zato se sadržaj i logička struktura mogu proširivati bez ograničenja i u toku funkcionisanja informacionog sistema. Upravo iz pomenutih razloga je SRBP izrazito složen sistem (po mnogima je kompleksniji od operativnog sistema). Predstavljanje i memorisanje grafike, zvuka, slike, itd. se puno razlikuje od predstavljanja i memorisanja običnog podatka, a sva nova predstavljanja i načina memorisanja bi trebalo sprovesti u već postojećim, glomaznim i kompleksnim sistemima za rukovanje bazama podataka. Pravih SRBP, koji rade na nivou multimedije tako da i fizičko rukovanje grafikom, zvukom, itd. rešavaju sami je vrlo malo.

8. PROJEKCIJA OBRADJE INFORMACIONOG SISTEMA

Projekcija podataka i projekcija obrade informacionog sistema su sasvim drugačije prirode. Projekcija podataka se može samostalno posmatrati, izgraditi i analizirati, a projekcija obrade uvek uslovljava postojanje projekcije podataka. Bez podataka ne mogu se definisati obrade. Podaci predstavljaju predmet obrade, a samim tim nju (obradu) i određuju (kanališu, ograničavaju). Traženje prosečnog izleda (prosečne slike) studenata se ne može definisati analogno recimo prosečnom uspehu studenata, jer je sabiranje (ili traženje proseka) video zapisa nedefinisano (za sada).

Obrade i same predstavljaju sistem unutar informacionog sistema, a sastoje se od činilaca na različitim nivoima apstrakcije od elementarnih operacija do funkcija i delova sistema (podsistema). Elementi obrade su dvostruko povezani sa podacima na sva tri nivoa apstrakcije, jer se informacione aktivnosti (prvi činilac projekcije obrade) odnose na podatke, a i drugi činilac obrade (informacioni događaj) se materijalizuje u vidu podataka (nova adresa boravišta studenta opisuje ulazni informacioni događaj). Sistem (projekcija) obrade se, dakle mora analizirati zajedno sa sistemom (projekcijom) podataka na sva tri nivoa apstrakcije. Uobičajeno je da se projekcija obrade i projekcija podataka posmatraju zajedno samo na srednjem nivou apstrakcije – na nivou dijagrama toka podataka (DTP, ili DFD – Data Flow Diagram). Na višem nivou (dela sistema, podsistema ili grupa funkcija) i na nižem nivou (nivo elementarnih operacija) se obično zaboravlja na nerazdvojivu vezu između sistema obrade i sistema podataka.

8.1. POJAVNI OBLICI SAZNANJA

Smeštanje podataka na magnetski medijum (fizički nivo) izvršava operativni sistem. Njegov je zadatak da obezbedi podatke i u slučaju traženja ili pretraživanja. Ako se želi, moguće je zaobići sistem za rukovanje podacima (data management system), odnosno metoda pristupa u korisničkim programima, ali tada se program mora direktno obratiti operativnom sistemu za uslugu i sam mora voditi računa o načinu smeštanja podataka. Ta praksa je bila veoma kratkog veka na početku računarske ere.

Aplikacioni programi danas koriste metode pristupa za smeštanje i dobijanje nazad memorisanih podataka. Sistemi za rukovanje podacima (metode pristupa) u saradnji sa operativnim sistemom rešavaju taj zadatak koji je transparentan za korisnika.

Ako su podaci informacionog sistema distribuirani, odnosno geografski dislocirani, neophodno je uključiti u obradu i sistem za prenos podataka (data communication system). Njega po potrebi angažuje sistem za rukovanje podacima u cilju realizacije smeštanja i dobijanja nazad onih podataka koji se nalaze na računarima izvan lokacije ili mesta upita (memorisanja ili traženja podataka).

Početkom osamdesetih godina pojavile su se relacione baze podataka. Sistemi za rukovanje bazama podataka po relacionom principu su preuzeli brigu preslikavanja između logičkog i fizičkog modela kao i samu fizičku realizaciju baza podataka. Zahvaljujući tome, projektovanje koje je do tada bilo zaokupljeno uglavnom fizičkom realizacijom baza podataka, od tada se moglo usredsrediti na logičko projektovanje baza podataka. U zadnje vreme se sve češće pristupa projektovanju pojmovnog ili konceptualnog modela podataka jer se shvatilo da baza podataka formulisana na logičkom nivou sadrži ograničenja koja su posledica zahteva korisnika pa i sredstava za obradu. Relacioni pristup omogućava izvršenje normalizacije, ali nema instrumentaciju da natera projektanta za njeno izvršavanje. Osnova relacionog pristupa, relacija predstavlja logičku celinu podataka na logičkom nivou apstrakcije. Model podataka (konceptualni nivo) se može realizovati pomoću raznih (hijerarhijskog, mrežnog ili relacionog) sistema za rukovanje podacima (logički nivo).

Opis modela po odgovarajućim strukturalnim, semantičkim (sadržajnim) i sintaktičkim (formalnim) pravilima naziva se šema.

Šema, znači, nije isto što i model, nego je njegova prezentacija. Isti model u zavisnosti od prihvaćenih (dogovorenih) pravila predstavljanja može imati više šema.

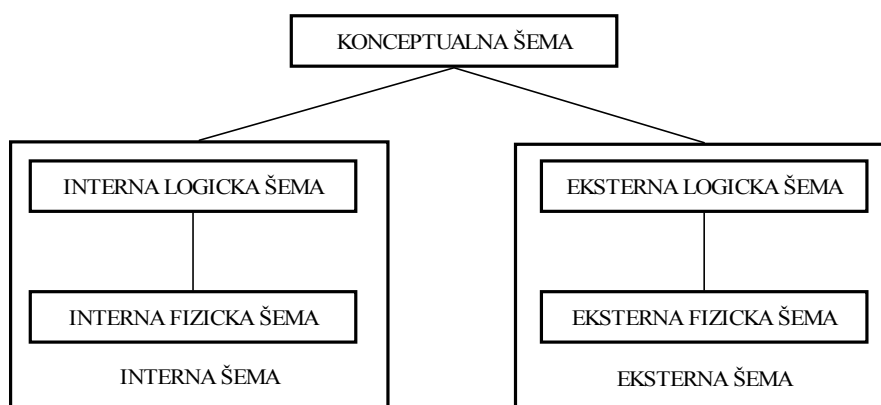
Po ANSI/SPARC preporuci za arhitekturu opisa baze podataka treba pripremiti tri vrste šeme. Organizaciju podataka na računaru opisuje interna šema. Podatke u realnom svetu (na izvornim dokumentima i glavi projektanta) opisuje eksterna šema.

Inerna i eksterna šema moraju biti usaglašeni. To usaglašavanje, tj. vezu između interne i eksterne šeme obezbeđuje konceptualna šema. Na primer podatak na izvornom dokumentu (narudžbenica) pod nazivom »naručena količina« se u bazi podataka pojavljuje kao »naruckol«. Tu vezu (i sve takve i slične ostale veze) treba da sadrži pojmovna šema. U konceptualnoj šemi se mogu naći takvi podaci koji se ne pojavljuju u bazi podataka (izvedeni podaci se ne memorišu u bazi podataka, ali na izvornim dokumentima se pojavljuju). Isto tako egzistiraju takvi podaci na računaru (u bazi podataka) koji se na izvornim dokumentima ne vode (indeksne trukture, pokazivači, unutrašnji identifikatori).

Konceptualna šema je takav opis modela podataka (apstrakcije ili koncepata na pojmovnom nivou) koji odgovara propisanim strukturalnim, semantičkim i sintaktičkim pravilima.

Eksterni i interni model imaju dva nivoa apstrakcije . Sadržaj predstavlja logički nivo, a forma (raspoređjivanje podataka) fizički.

Eksterna šema je opis eksternog modela podataka koji ima dva nivoa apstrakcije, shodno tome se deli i eksterna šema na eksternu logičku i fizičku šemu. Eksterna šema se oslanja na konceptualnu šemu.



Sl. 8.1

Interna šema je takav opis internog modela podataka podeljenog na logički i fizički nivo, koji se oslanja na konceptualnu šemu.

Po prethodnim rečenicama se može zaključiti da postoje dve eksterne i dve interne šeme po strukturi koja se nalazi na sl. 8.1.

Dva aspekta interne šeme odgovaraju logičkom i fizičkom modelu podataka (logičkom i fizičkom projektu baze podataka).

8.2.ULAZ, IZLAZ I MEDJUREZULTAT

U cilju rukovanja podacima smeštenih na računaru neophodno je da čovek (korisnik) komunicira sa mašinom (računarem).

Računarske komunikacije se mogu odvijati na dva načina:

1. u off-line modu – korisnik nije u neposrednoj vezi sa računarem koji obavlja obradu – unos podataka se vrši na sekundarne nosioce podataka (off-line obuhvat podataka) i

2. u on-line modu – korisnik je u neposrednoj vezi sa računarem koji obavlja obradu podataka.

Predmet obrade podataka se naziva ulaz, a rezultat obrade izlaz. Pošto i ulaz i izlaz predstavljaju podatke, govori se o ulaznim i izlaznim podacima (a ne informacijama).

Medjurezultat je takav rezultat (izlaz) neke aktivnosti (operaciji) u unutrašnjosti obrade koji se ne pojavljuje kao rezultat cele obrade, već predstavlja predmet (ulaz) druge (drugih) aktivnosti u okviru posmatrane obrade.

Obrada podataka se može realizovati na dva načina:

1. batch ili paketna obrada (batch processing) se može predstaviti sledećim izrazom:

ulaz1 – obrada1 – medjurezultat1 – obrada2 – medjurezultat2 – ...

U ovom slučaju se skupljaju podaci za obradu u jednu “kutiju” (paket) i odgovarajući niz operacija se primenjuje na ceo skup sakupljenih podataka. Izlaz (medjurezultat) se smešta u neku privremenu datoteku da bi se novi niz operacija mogao primeniti na ceo paket u sledećem koraku. Tako se vrši fakturisanje komunalnih usluga i potrošnje energenata za stanovništvo.

2. transakciona obrada (transaction oriented processing)

Tu se sve operacije kompletne obrade primenjuju na svaki poseban ulaz, tj. na svaku stavku ulaza. Kod obrade bankarskih promena kompletno knjiženje se sprovodi bez zadržavanja za svakog štedišu i to “trenutno”.

U računarskoj tehnici se često vrši zadržavanje procesa obrade. Obrada se najčešće prekida na sledeća dva mesta u obradi:

1. na ulaznoj strani. Kod zadržavanja obrade na ulazu, operacije obrade se nakon učitavanja i kontrole ulaza (ulaznih podataka) prekidaju. Na primer ulazni podaci – stanja pročitanih strujomera – nakon unosa dospevaju u datoteku promena. Takva zadržana obrada, ako se izvršava u paketnom (batch) načinu, pored samog signala za start i nema drugi ulaz. Datoteka promena je njen interni ulaz.
2. na izlaznoj strani. Nakon završene obrade nema uvek smisla odmah štampati rezultat. Razloga za to ima više. Prvi razlog, recimo, može biti da se obrada obavlja u delovima zbog prevelikog broja stavaka ulaza (u slučaju pada sistema, obrada se mora pokrenuti ispočetka; ako računar nema funkciju checkpoint-restart bolje je obradu izvršavati na manjem broju stavaka iz paketa). Drugi razlog može biti da je izlaz obrade potreban na drugim geografskim lokacijama (prenos magnetskog medijuma je znatno jednostavniji i jeftiniji od prenosa istampanih rezultata na papiru. U tim slučajevima se izlazni

podaci obrade smeštaju po krajnjem fizičkom izgledu u prelaznu datoteku – print file (print – image file).

Model podataka ne sme da sadrži ulazne, izlazne i podatke medjurezultata. Razlozi su sledeći:

1. ti podaci se, po pravilu, preklapaju,
2. najčešće ne predstavljaju suštine na pojmovnom nivou i
3. takvi podaci nisu normalizovani (sadrže podatke više vezanih tipova entiteta).

8.3.BAZA PODATAKA I PROJEKCIJA OBRADE

Baza podataka je skup saznanja (podataka) koji se organizuje na osnovu pojmovnog modela podataka (konceptualne šeme), a memoriše se na računaru. Kartoteke u bibliotekama – mada predstavljaju organizovani skup saznanja – se ne smatraju bazama podataka.

U informacionom sistemu se razlikuju dve vrste podataka, s obzirom na njihovu promenljivost ili stabilnost (nepromenljivost) u vremenu:

1. Podaci koji se odnose na pojave (tipove entiteta) realnog sistema nazivaju se matičnim podacima.

Pojam matičnih podataka je najbliži pojmu evidencije. Često se, naime smatra da su podaci o pojavama ili objektima o kojima treba voditi evidenciju predstavljaju matične podatke.

Za pojave realnog sistema se kaže da su relativno stabilne. Ta formulacija, međutim, ukazuje na to, da se stvari (pojave, objekti) ipak menjaju (nastaju, menjaju se i nestaju) u realnom svetu. Njihova odslikavanja u informacionom sistemu – entiteti – se takodje radjaju, menjaju se i odumiru. Za te promene se u informatici koristi pojam životni ciklus entiteta (entity life-cycle).

2. Podaci o događajima (transakcioni podaci ili samo prosto: promene) se odnose na promene osobina (obeležja – atributa) relativno stabilnih pojava realnog sistema.

Matični i transakcioni podaci su u dvostrukoj medjusobnoj povezanosti, koja se može formulisati na sledeći način:

1. te dve vrste podataka se preklapaju: kod promene imena studenta, na primer, mora se zadati i matični podatak ime, za koje se promena odnosi, kao i podatak promene: novo, promenjeno ime studenta,

2. postoje takvi događaji čiji se podaci smeštju u bazu podataka (podaci ne izazivaju samo izmene opisa neke druge pojave, već se i memorišu), tj. postaju matični podaci. Kada opisni podaci događaja postaju matični podaci, nastaje tip entiteta koji nema celokupan životni ciklus (nostrifikacija diplome): nedostaje modifikacija.

Pojam modifikacije se na konceptualnom nivou razlikuje od pojma ispravke grešaka, bez obzira što se tehnički, u praksi na isti način realizuju. Na pojmovnom nivou, naime, nije svejedno da li se podatak, recimo ime studenta, modifikuje zbog lošeg kucanja pri unosu, ili zbog toga što se ime studenta zaista promenilo.

Promene se najčešće pojavljuju samo na izvornim dokumentima, dok matični podaci se skoro isključivo memorišu na računaru.

Priroda obrade podataka se izmenila tokom godina računarske ere. Ulazni podaci se danas neposredno ne prebacuju (nakon izvesnih i/ili potrebnih obrada) na izlaz, već se memorišu negde u matičnim podacima. Odande ih treba uzeti u slučaju dalje obrade. Tako formula

ulaz – obrada – izlaz

se pretvara u par formula

ulaz – obrada1 – baza podataka

i

baza podataka – obrada2 – izlaz.

Prva formula (lanac) predstavlja ažuriranje (održavanje – maintenance), što pokriva dodavanje (obuhvat novih entiteta), aktualizaciju (modifikaciju) i brisanje. Ove operacije nemaju prave izlazne podatke.

Druga formula (lanac) predstavlja dve vrste obrade:

1. (selektivno) pretraživanje, koje može da pruži i (novu) informaciju (u odgovoru na upit koji studenti nisu upisali godinu se pojavljuje i njihov broj) ako se ono pametno formuliše i
2. stvaranje novih saznanja, što se može postići na dva načina:
 - pomoću unapred definisanih (rutinskih) operacija za stvaranje novih saznanja (računanje izvedenih informacija na osnovu postojećih – tražimo prosečnu ocenu konkretnog studenta za konkretnu godinu studija),
 - obradom na slučajno postavljeni upit.

Činjenica je da ažuriranje usvljeno upitom, tj. da podrazumeva prethodni upit ili pretraživanje, ali namera korisnika je sasvim drugačija nego kod prostog traženja ili pretraživanja.

Obrada podataka (rukovanje podacima i stvaranje novih saznanja) se može izvršiti na sledeća dva načina:

1. proceduralna – programer pisanjem naredbi izvornog koda programa definiše obradu,
2. deklarativna – softver za rukovanje bazom podataka sadrži takve obrade u uopštenom, parametrizovanom obliku. Za izvršenje deklarativne obrade treba zadati odgovarajuće tehničke podatke (parametre) u vezi podataka koje želimo obraditi.

Podaci u bazi podataka su međusobno povezani. Može se desiti da korekcija (modifikacija) jednog podatka povlači za sobom modifikaciju drugog ili drugih podataka. Ta pojava je poznata pod imenom automatsko prosledjivanje (propagation). U tom slučaju na konceptualnom nivou obrade jednu operaciju sledi neka druga operacija čije neizvršavanje dovodi bazu podataka u nekonzistentno stanje. Kao primer možemo navesti situaciju kada se, na primer, memoriše trenutna prosečna ocena za studenta, koja se ažurira prilikom svakog unosa ocene položenog ispita. Ako student poništi neki ispit, ažuriranje trenutne prosečne ocene je obavezno, jer u protivnom će podaci u bazi podataka postati nekonzistentni.

Ako ne postoji automatsko prosledjivanje, programer mora sprovoditi sve vezane obrade koje su prisutne na pojmovnom nivou. U novijim sistemima za rukovanje bazama podataka može se deklarirati automatska propagacija za brisanje i modifikaciju naznačavanjem opcionalnosti tipa poveznika (minimalni kardinalitet 0 označava da je veza opcionalna (tip poveznika opcionalan), a vrednost minimalnog kardinaliteta 1 pokazuje na obaveznu vezu. Na sl.8.2. se nalaze dva “školska” primera u vezi potrebe propagacije.



Sl. 8.2.

Za oba primera važi konstatacija da je promena identifikatora u nadređenom tipu entiteta neminovno povlači za sobom sprovođenje modifikacije u svim podređenim tipovima entiteta i to do nivoa na kojem tipovi entiteta sadrže promenjeni identifikator. Promena ključa (vrednosti identifikatora) se ne preporučuje u praksi (strogo zabranjena)!

Brisanje konkretnog studenta (nadređeni tip entiteta – STUDENT) iz evidencije u prvom primeru povlači sa sobom i brisanje svih njegovih ocena (podređeni tip entiteta – OCENA) iz evidencije, jer je za tip entiteta OCENA označeni tip poveznika IMA obavezna!

U drugom primeru se javljaju studenti – sportisti za Elektrijadu za veći broj timova (fudbal, košarka, rukomet, šah, stoni tenis, itd), ali tako da svaki student može biti član samo jednog tima. Organizator može da odluči o ukidanju nekog konkretnog tima, ali student se ne briše iz evidencije (putovaće na Elektrijadu kao navijač). Brisanjem konkretnog tima (nadređeni tip entiteta – TIM) ne iziskuje brisanje studenata (podređeni tip entiteta – STUDENT) – njegovih članova (već se samo briše vrednost spoljnog ključa u tipu entiteta STUDENT) jer je za podređeni tip entiteta veza opcionalna (nije obavezna). Treba samo još istaći da se primer razlikuje od prakse organizacije Elektrijade.

8.4.KONCEPCIJA NAVIGACIJE, UPITI I IZVEŠTAJI

Pojam navigacije se pojavio u toku sedamdesetih godina u okviru mrežne organizacije baza podataka. Na samom početku navigacija je bila koncepcija na fizičkom nivou apstrakcije (pamtile su se fizičke adrese u bazi podataka), danas se, međutim, navigacija formuliše na logičkom nivou. Navigacija danas označava transversalna kretanja u bazi podataka shodno

STUDENT

Br_indeksa	Ime_i_prz	...
E494	Marko Marković	
E495	Petar Petrović	
...	...	

PREDMET

Šifra_pr	Naziv_pr
112	Matematika
147	Fizika
174	Elektrotehnika
...	...

POLOŽENI_ISPITI

Br_indeksa	Šifra_pr	Datum_ispita	Ocena
E494	112	12.09.2000.	9
E494	147	15.09.2000.	7
E494	174	05.10.2000.	6
E495	174	14.07.2000.	8
E495	112	15.09.2000.	8
...	...	...	...

Sl. 8.3.

tipovima poveznika u modelu podataka.

Za sastavljanje liste položenih ispita za sve studente prvo se pronadje prvi student iz tipa entiteta STUDENT (sl. 8.3.), i na osnovu njegovog identifikatora, a to je obeležje Br_indeksa, prelazi se u tip entiteta POLOŽENI_ISPITI (u ovom tipu entiteta Br_indeksa je spoljni ključ) i pronadje se prva ocena iz položenih ispita (ako ih ima). Dobijanje konkretnog naziva predmeta

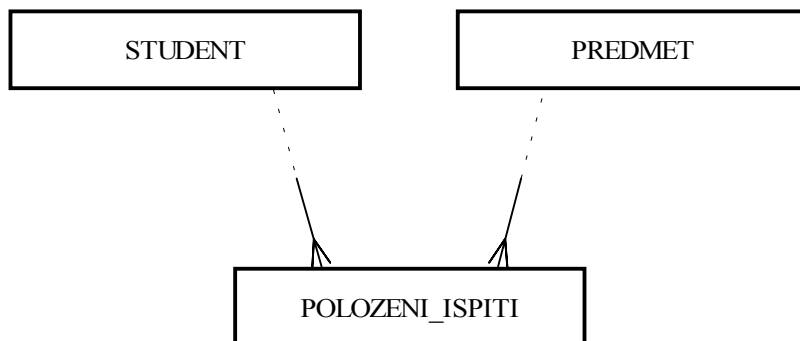
za kojeg je pronadjena ocena u tipu entiteta POLOŽENI_ISPITI treba zapamtiti vrednost spoljnog ključa Šifra_pr i preći u tip entiteta PREDMET (gde je Šifra_pr identifikator) u cilju nalaženja naziva predmeta. Nakon dobijenog naziva predmeta treba se vratiti na onu pojavu tipa entiteta POLOŽENI_ISPITI sa koje je tip entiteta napušten. Ista je situacija i nakon pronalaska zadnje ocene i njoj pripadajućeg naziva predmeta, kada se treba vratiti u tip entiteta STUDENT: treba znati tačku (pojavu tipa entiteta STUDENT) sa koje je tip entiteta (u ovom slučaju STUDENT) napušten. Pre napuštanja svakog tipa entiteta (osim poslednjeg u lancu obrade) treba zapamtiti tačku (pojavu tipa entiteta) sa koje je tip entiteta napušten. To je potreban uslov za uspešno izvršenje obrade.

Navigacija kroz tipove entiteta se jednostavno može formulisati pomoću deklarativnog SQL (Structured Query Language – strukturirani upitni jezik) jezika. SQL, kao što i njegovo ime kaže se koristi za definisanje upita, a takozvani generator izveštaja (report writer) za generisanje i formiranje izveštaja. SQL i report writer su deklarativne alternative za do tada korišćene proceduralne obrade.

Upit se deklarise sastavljanjem SELECT rečenice (naredbe) SQL jezika ili popunjavanjem forme (unapred definisanog ekrana). Select rečenicom treba definisati koje podatke, odakle, pod kojim uslovima, primenom kojih prostih funkcija, u kojem redosledu, u kakvoj formi itd. na kom izlaznom uredjaju (ekran/štampan) treba da prikaže sistem za rukovanje bazama podataka.

Generator izveštaja upit realizuje na isti način kao što to radi upitni jezik, ali postoje suštinske razlike u prikazivanju rezultata. Report writeri obezbeđuju tabelarnu organizaciju rezultata pretraživanja sa višestepenim medjuzbirevima, a i parametrizaciju samih tabela.

Pretpostavimo da treba rešiti zadatak (upit) za listanje položenih ispita zajedno sa ocenama za nekog konkretnog studenta pomoću SELECT naredbe na osnovu strukture koja se nalazi na sl. 8.3., čiji je model podataka prikazan na sl. 8.4.



Sl. 8.4.

SELECT rečenica je ekstenzionalne prirode, jer vrši izbor (selektuje) deklariranih pojava tipa entiteta iz celog skupa pojave tipa entiteta, ali ima ugrađenu intenzionalnu dimenziju, a to je projekcija. Projekcija je konceptualna operacija i označava izbor potrebnih (za nas važnih) osobina (obeležja, atributa) od svih mogućih za tip ili tipove entiteta.

Primer: za listanje ocena konkretnog studenta (sa brojem indeksa 495) za tip entiteta POLOŽENI_ISPITI se može formulisati upit oblika

```
SELECT Br_indeksa, Šifra_pr, Ocena
FROM položeni_ispiti
WHERE Br_indeksa=495;
```

SELECT rečenica, znači bira pojave tipova entiteta POLOŽENI_ISPITI (operacija nad ekstenzijom) po odgovarajućem uslovu (WHERE), i to prikazivanjem vrednosti samo nekih od obeležja (operacija nad intenzijom).

Upit bez transverzije nije moguć ako treba konsultovati više od jednog tipa entiteta u cilju dobijanja rezultata. Pomoću transverzije se realizuje navigacija između tipova entiteta (odnosno prelazak sa jednog na drugi tip entiteta). Transverzija se može ostvariti na tri načina:

1. višestrukim selektom,
2. kombinovanim selektom i
3. pomoću join (spajanje) operacije.

Višestruki selekt označava izvršenje više jednostavnih SELECT rečenica u kojima se “napada” samo jedan tip entiteta. Na primer posle izvršavanja gore naznačenog upita, vršimo upit u tip entiteta STUDENT sa vrednošću Br_indeksa, da bi saznali njegovo ime, a posle bi redom vršili upit na osnovu vrednosti obeležja Šifra_pr u tip entiteta PREDMET u cilju prikazivanja naziva predmeta (Naziv_pr).

Kombinovani selekt obuhvata sva tri tipa entiteta u okviru jedne rečenice bez korišćenja operacije spajanja (join):

```
SELECT Br_indeksa, Ime_i_prz, Naziv_pr, Ocena
FROM student, položeni_ispiti, predmet
WHERE Br_indeksa IN položeni_ispiti=495 AND
      Br_indeksa IN položeni_ispiti=Br_indeksa IN student AND
      Šifra_pr IN položeni_ispiti=Šifra_pr IN predmet;
```

Sintakse pojedinih SQL jezika se razlikuju, ali je suština ista. Kod treće vrste realizacije transverzije vrši se spajanje dva ili više tipova entiteta po zajedničkom obeležju. Primer bi mogao biti formiranje novog tipa entiteta POLOŽENI_ISPITI_SA_NAZIVOM_PREDMETA pomoću join operacije, koji bi pored obeležja u tipu entiteta POLOŽENI_ISPITI sadržao i nazive odgovarajućih predmeta kod svake pojave tip entiteta. Join operacija je izrazito spora jer podrazumeva operacije paketnog (batch) tipa: pojava svakog tipa entiteta jednog tipa entiteta se proširuje sa pojavom dodatnog obeležja ako su vrednosti zajedničkog (spajajućeg) obeležja iste.

U toku obrade treba obratiti pažnju na smer navigacije: ako je taj smer od nadređenog prema podređenom, biće više selektovanih pojava tipa entiteta (zbog kardinalnosti 1:N), a u obrnutom smer samo jedna pojava.

Transakciona obrada transakciju spovodi kroz celu bazu podataka za svaku stavku. Na primer kod generisanja liste položenih ispita za sve studente na početku se nalazi prvi student tipu entiteta STUDENT, pamćenjem vrednosti njegovog Br_indeksa se prelazi na tip entiteta POLOŽENI_ISPITI se dobije veći broj pojava entiteta (ako je položio više ispita), stane se kod prvog položenog ispita i pamćenjem Šifra_pr se prelazi na tip entiteta PREDMET u cilju dobijanja Naziv_pr, pa se vraća u tip entiteta POLOŽENI_ISPITI, prelazi se na sledeći položen ispit studenta, pa opet u PREDMET,...Sve dok se ne obrade podaci za jednog studenta, onda se tek prelazi na sledeći student (sledeću pojavu tipa entiteta STUDENT). U slučaju kad je smer navigacije od nadređenog prema podređenom (STUDENT → POLOŽENI_ISPITI) organizuje se ciklus za obradu (u cilju obrade svake selektovane stavke) svakog položenog ispita (ta obrada je u stvari nalaženje naziva predmeta).

Stvaranje novog tipa entiteta POLOŽENI_ISPITI_SA_NAZIVOM_PREDMETA pomoću join operacije stvaramo “predobradjen” paket sa dosta velike potrošnje vremena. Ciklus se ni ovde ne može izbeći, ali je filozofija drugačija od transakcione obrade.

8.5.PROJEKTOVANJE OBRADJE

Proces obrade predstavlja lanac događaja i aktivnosti. Uzrok izvršavanja neke aktivnosti je određen događaj, a završetak aktivnosti može značiti više različitih događaja. Povezanost aktivnosti i događaja je takva, da neke aktivnosti mogu izazvati više različitih događaja, a završetak aktivnosti može označavati početak više različitih događaja. Aktivnosti i događaji u opštem slučaju obrazuju mrežnu strukturu. Aktivnosti se odnose na podatke, a događaji izazivaju aktivnosti i/ili nastaju kao posledice izvršene aktivnosti, ali se i sami mogu opisati

podacima. Obradu generiše ulaz (ulazni događaj), a niz aktivnosti i događaja (obrada) se završava izlazom (izlaznim događajem). Projektant ima zadatak da na osnovu poznatog ulaza, izlaza i modela podataka organizuje takav logički redosled izvršavanja aktivnosti koji ukazuje i na vremenski redosled njihove egzekucije. Za vreme izvršenja obrade vrši se propagacija podataka sa ulaza u bazu podataka, a posle iz baze na izlaz. Ta propagacija se najčešće opisuje samo na srednjem nivou apstrakcije. Problem u prethodnoj rečenici predstavlja reč samo. Da se obrada prati na srednjem nivou apstrakcije to još ne bi bio problem, ali da se prati samo na njemu to već jeste. U daljem tekstu će se prikazati sva tri nivoa apstrakcije na polju obrade podataka.

8.5.1. Modeliranje obrade

Osnova modeliranja u projekciji obrade je spoznaja da se i informacioni događaji i aktivnosti (operacije) mogu tipizirati. U vezi događaja sam pojam tipiziranja se lako shvata, pošto se događaj može smatrati pojavom entiteta. Narudžba (tip entiteta) je, recimo, jedna pojava tipa događaja “narudžba-događaj”. Pojava jedne narudžbe (tip entiteta) označava događaj koji izaziva izvršavanje neke obrade, aktivnosti ili operacije

Informacione aktivnosti su u stvari operacije koje obezbeđuju funkcionisanje sistema. Pošto je predmet operacije, kao i rezultat događaj (pojava nekog tipa entiteta) može se definisati tip operacije na sledeći način. Operacije koje imaju iste ulazne i izlazne tipove događaja (ulazne i izlazne tipove entiteta) spadaju u isti (zajednički) tip operacije.

Za tipove operacija i same operacije važe klasifikacije koje su date u tački 8.2. za obradu podataka.

Funkcija je skup konkretnih operacija (konkretna obrada) za realizaciju izlaza na osnovu konkretnog korisničkog zahteva. Funkcija generiše izlaz u formi koja omogućuje njeno tumačenje od strane naručioca (korisnika). Funkcije koje imaju iste tipove entiteta u ulaznim, izlaznim i matičnim podacima kao i medjurezultatima pripadaju istom tipu funkcije.

Tipovi funkcija, tipovi operacija koji njih (tipove funkcija) sačinjavaju i tipovi prelaznih događaja koji tipove operacija povezuju unutar tipova funkcija, a i tipove funkcija međusobno povezuju, zajedno čine funkcionalni model (konceptualni model operacija) informacionog sistema.

Funkcionalni model se može ostvariti na više načina. Struktura operacija koja prikazuje način realizacije funkcionalnog modela naziva se logička struktura operacija ili logički model operacija.

Konkretno izvršavanje operacija zavisi od sredstava (hardvera i softvera) na kojima sistem funkcioniše. Struktura operacija koja konkretizuje izvršavanje operacije (niza operacija) na nekom hardversko – softverskom okruženju (zajedno sa svim hardverskim i softverskim specifičnostima) naziva se fizička struktura operacija ili fizički model operacija.

Funkcionalni model je nestabilniji od konceptualnog modela podataka, jer je pojam funkcije odnosno tipa funkcije vezan sa izlazom određenim od strane korisnika – naručioca informacionog sistema.

Pošto su veze između tipova funkcija u stvari podaci (tipovi prelaznih događaja, tj. tipovi entiteta), njih treba smatrati prvobitnim (podatke). Funkcionalni model, međutim može povratno uticati na model podataka preko izlaza definisanih od strane korisnika. Novi zahtevi korisnika u izveštajima mogu izazvati proširenje modela podataka.

8.5.2. Projektovanje obrade na srednjem nivou

U sedamdesetim godinama za prikazivanje obrade informacionog sistema koristile su se dve vrste dijagrama: sistemski dijagram i programski dijagrami (flow chart). Flow chart-i se danas, u vreme kada su jezici četvrte generacije najrasprostranjenije ne koriste, a umesto sistemskih dijagrama se primenjuju naprednije i izražajnije tehnike. O njima će biti reči u ovoj tački.

8.5.2.1. Graf za rukovanje podataka

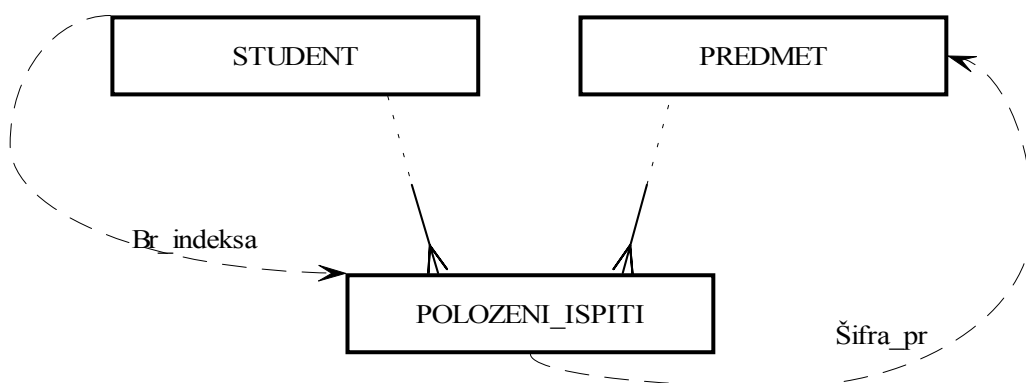
Top – down način pristupa u savremenom projektovanju (razvoju) uopšte, a takodje u projektovanju obrade potiče iz vremena strukturiranog programiranja i dizajna. Ta koncepcija je, po principima strukturalnog dizajna usko povezana sa pojmom crne kutije koja karakteriše odnos celine prema svojem delu. Na početku se zna šta predstavlja celinu, a ne znaju se detalji te celine. Iz tog razloga se celina na početku prikazuje kao crna kutija, koja se tek posle postepenog upoznavanja detalja rasvetljava: neki detalji postaju beli ili sivi, a pojavljuju se nove crne kutije manjih dimenzija. Te manje crne kutije se rasvetljavaju u sledećem prolazu, pri čemu se otkriva upravljanje, tačnije pogodna struktura upravljačkih operacija. Obrada podataka se predviđa u najnižem nivou hijerarhije.

Po današnjem shvatanju upravljanje je određeno logičkim redosledom izvršavanja operacija nad podacima. Model podataka sadrži sve tipove entiteta, njihova obeležja i sve tipove poveznika izmedju tipova entiteta. Artikulisani informacioni zahtev korisnika (ako je kompletan) definiše i ulaz(e) i izlaz(e), na osnovu čega se mogu odrediti ulazni entitet(i) i izlazni entitet(i), a model podataka (organizacija tipova entiteta u šemi – fizičkoj predstavi modela podataka) diktira redosled obrade pojedinih tipova entiteta uz pomoć tipova poveznika. Makrostruktura obrade po tom informacionom zahtevu je time i određena i predstavlja osnovu za konkretan računarski program, te je zgodno to odmah i skicirati (program suit – skeleton programa).

Činioci informacionog sistema se moraju usaglašavati, kako unutar iste grupe (projekcije podataka, obrade i okruženja), tako i izmedju elemenata koji pripadaju različitim grupama činilaca. Homogeno usaglašavanje činilaca unutar projekcije podataka se realizuje putem modeliranja podataka. Samo modeliranje podataka se može izvršiti nezavisno od elemenata ostalih činilaca projekcije obrade i okruženja zbog postojanosti i stabilnosti logičke veze izmedju podataka. U nekoj obrazovnoj ustanovi će sigurno postojati nastavnici, učenici, zgrade, nastavne oblasti (predmeti), ocene, itd. čije se medjusobne zavisnosti neće zavisiti od načina same obrade podataka. Model obrade se ne može realizovati bez poznavanja i nezavisno od izgradjenog modela podataka.

Na osnovu razgovetno formulisanog informacionog zahteva se mogu napisati i računske operacije koje vode do željenih rezultata. Poznavajući potrebne računske operacije i mesta “nastanka” podataka (zahvaljujući modelu podataka) lako je skicu programa obogatiti sa mestima ili trenucima izvršavanja računskih operacija. Svim ovim aktivnostima u vezi izrade program suita je osnova jedan graf koji se zove graf za rukovanje podataka.

Graf za rukovanje podataka je pogodan deo modela podataka koji predstavlja osnovu za artikulisanje odgovora na informacioni(e) zahtev(e) korisnika dopunjen usmerenim linijama za navigaciju pri obradi. Model podataka sa sl. 8.4. preradjen u graf za rukovanje podataka se nalazi na sl. 8.5.



Sl. 8.5.

8.5.2.2. Dijagram toka podataka

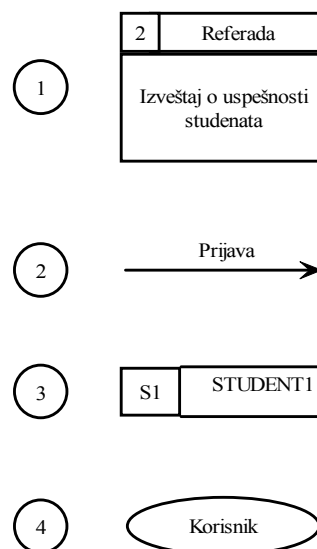
Drugo korisno sredstvo u projektovanju obrade je dijagram toka podataka (DTP ili DFD – Data Flow Diagram). DTP sadrži glavne aktivnosti i veće jedinice podataka (ponekad samo tipove entiteta). Predstavlja vizuelnu predstavu na srednjem nivou abstrakcije lanca

ulaz – obrada1 – baza_podataka – obrada2 – izlaz

Dijagram toka podataka ukazuje na upravljačke operacije i na operacije za rukovanje podacima. Operacije za stvaranje novih saznanja na ovom dijagramu nisu predviđene.

Simboli dijagrama toka podataka se nalaze na sl. 8.6.:

1. Proces
2. Tok podataka
3. Skladište podataka
4. Spoljni entitet

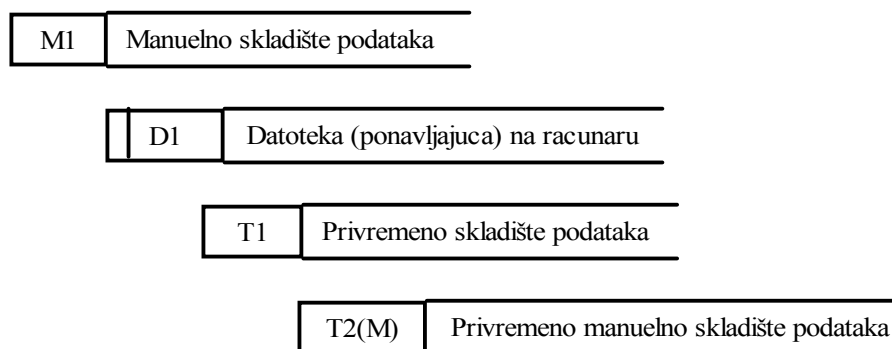


Sl. 8.6

Proces se predstavlja pomoću pravougaonika i realizuje aktivnosti sistema za obradu podataka. Proces ima svoje ulazne podatke (ulazne tokove podataka) koje transformiše u izlaze (izlazne tokove). Pravougaonik za predstavljanje procesa se deli na tri dela. U gornjem levom ćošku se nalazi identifikator konkretnog procesa. To je prosto redni broj i samovoljno se dodeljuje procesima (uobičajeno je da se brojevi dodeljuju procesima crtajući ih od gore prema dole i s leva na desno na papiru), tj. ne označavanju nikakav redosled ni crtanja ni izvršavanja. U gornjem desnom delu se smešta naziv organizacione jedinice koja je nadležna za izvršavanje tog procesa (eventualno naziv radnog mesta, konkretnog izvršioca procesa). Izvršioci i organizacije se smeštaju kod prvog predstavljanja (prve skice) DTP-a (tada se DTP ponekad naziva i fizički dijagram toka podataka i može imati prikaze i o toku materijala) radi bržeg snalaženja na dijagramu, a kasnije se ne moraju prikazivati. Naziv procesa se nalazi u centralnom delu pravougaonika i treba da označava neku aktivnost. Sadržaj procesa se može sažeto opisati u prilogu dijagrama toka podataka.

Tok podataka na DTP-u se označava strelicama i prikazuje tok podataka između elemenata na dijagramu, pri čemu smer kretanja određen pokazivanjem vrha strelice. Strelice se mogu zamisliti kao kanali za prenos podataka i zato se u nekim izvorima zovu cevovod za podatke (data pipeline). Tokovi podataka treba da dobiju (po mogućstvu) jedinstvena i opisna imena. Naziv toka podataka se ne mora navesti u nekim očiglednim situacijama, kao što je ažuriranje podataka u skladištu podataka. Kod opisa postojećih sistema često se za nazive tokova podataka koriste postojeći dokumenti koji su prisutni (koji se koriste) u cirkulaciji dokumentacije preduzeća.

Skladište podataka predstavlja privremeno ili trajno mesto smeštanja podataka. Skladišta mogu biti manuelna (kartoteke, registri) a mogu se nalaziti na računaru (klasične datoteke ili tabele u bazi podataka). Posebno se mogu označavati i privremene (temporary) datoteke shodno oznakama na sl. 8.7.



Sl. 8.7.

Spoljni entitet je u stvari spoljni element koji ima neki uticaj na posmatrani sistem. Aktivni uticaj spoljnog entiteta na analizirani sistem se ogleda u ulaznim podacima pomoću kojih spoljni element formuliše svoj uticaj. Spoljni entitet je pasivan ako predstavlja primaoca izlaznih podataka posmatranog sistema. Spoljni element na istom crtežu može biti i aktivan i pasivan (pa čak i »istovremeno«). Prikaz istog spoljnog entiteta na više mesta u okviru istog dijagrama toka podataka može pojednostaviti crtež, tj. povećati njegovu preglednost. U tim slučajevima se u sam simbol spoljnog elementa ucrtava kosa linija.

Elementi DTP-a su međusobno povezani preko tokova podataka, ali nisu sve veze dozvoljene. Tabela 8.1. sadrži kvalifikaciju veze između elemenata dijagrama toka podataka, pri čemu vrednost elementa tabele »DA« označava mogućnost povezivanja dva elementa, a »NE« ukazuje na zabranu povezivanja između dva elementa.

Tabela 8.1.

	Spoljni entitet	Proces	Skladište
Spoljni entitet	NE	DA	NE
Proces	DA	DA	DA
Skladište	NE	DA	NE

Dijagram toka podataka se crta u više nivoa. Pregledni dijagram koji obuhvata celokupan informacioni sistem se naziva dijagram konteksta i smatra se nultim nivoom. Broj procesa na jednom DTP-a može biti od 7 do 10. Dijagram konteksta sadrži samo jedan proces koji predstavlja celokupan informacioni sistem.

Dijagram konteksta je polazna osnova za hijerarhijsku dekompoziciju DTP-a. Na nižim hijerarhijskim nivoima se razradjuju procesi sa višeg nivoa hijerarhije. Proces na hijerarhijskom nivou 1 su praktično podsistemi (glavne grupe procesa) posmatranog sistema. Na prvom nivou obično još nema skladišta podataka (sem u izuzetnim slučajevima, kada barem dva podsistema koriste isto skladište). »Procesi« na nivou 1 DTP-a, koji označavaju podsisteme se numerišu brojevima 1, 2, 3, itd. Na sledećem nivou dekompozicije prilikom razlaganja podsistema br. 1., procesi dobijaju numeraciju 1.1., 1.2., 1.3., itd.

Prilikom crtanja DTP nižih nivoa, zahvaljujući dubljoj analizi funkcionisanja sistema može se desiti da se otkrije greška ili nepreciznost u predstavi na višim nivoima DTP-a. Tada je neminovno usaglašavanje pogrešnih gornjih nivoa DTP-a sa činjeničnim stanjem. Dijagrami na nižem nivou koji predstavljaju razrade konkretnih procesa na višem nivou hijerarhije moraju imati iste ulazne i izlazne tokove podataka kao što je proces na višem nivou koji se detaljiše. Tok podataka se ne može računati. U slučaju da se različite kopije istog dokumenta šalju različitim primaocima (skladištima, a posebno spoljnim entitetima), primaoci (spoljni elementi) se moraju posebno prikazivati. Situacija je slična, ako se neki dokument sastoji od više kopija, a izvor mu je različit. DTP ne sadrži simbol odluke, jer se odluka smatra procesom ili delom procesa. Dijagrami toka podataka moraju biti pregledni bez nepotrebnih detalja: glavni procesi treba da su vodilje, na njih treba obratiti pažnju i kod prikazivanja.

Pitanje nivoa razlaganja dijagrama toka podataka je povezano sa pojmom elementarnog procesa. Proces postaje elementaran ako tokovi njegovih podataka postaju izmišljeni, veštački, kakvih u stvarnosti i nema. Nivo razlaganja DTP-a u slučaju reprojekovanja postojećeg informacionog sistema je određeno problemima zbog kojih informacioni sistem ne može efikasno funkcionisati: DTP se tada detaljiše do tog nivoa na kojem se pronadju razlozi lošeg funkcionisanja informacionog sistema. Elementarni procesi se dalje ne razlažu, ali se nalaže

priprema njegovog opisa koji sadrži informacije o tome šta se i kako dešava unutar elementarnog procesa. Taj opis ne sme da bude duži od pola strane A4 formata i mora da se priloži uz dijagram toka podataka (za procese uopšte važi opcionalnost: može, a za elementarne procese je to obaveza).

Konstruisanje dijagrama toka podataka počinje od kontekstnog dijagrama (nulti nivo) sa kojim je u uskoj vezi još nivo 1, na kojem su prikazani podsistemi. Na samom početku projektant treba da obrati pažnju na:

1. granice sistema,
2. spoljne entitete (koji formulišu zahteve i ulazne podatke prema sistemu i koji koriste izveštaje generisane od strane sistema),
3. najvažnije ulazne i izlazne tokove podataka (koji predstavljaju formu interakcije između sistema i njegove okoline) i
4. glavne funkcije (glavne grupe procesa – podsisteme) predmetnog sistema.

Dalja hijerarhijska dekompozicija dijagrama toka podataka iziskuje nove aktivnosti projektanta u cilju efikasnog rada:

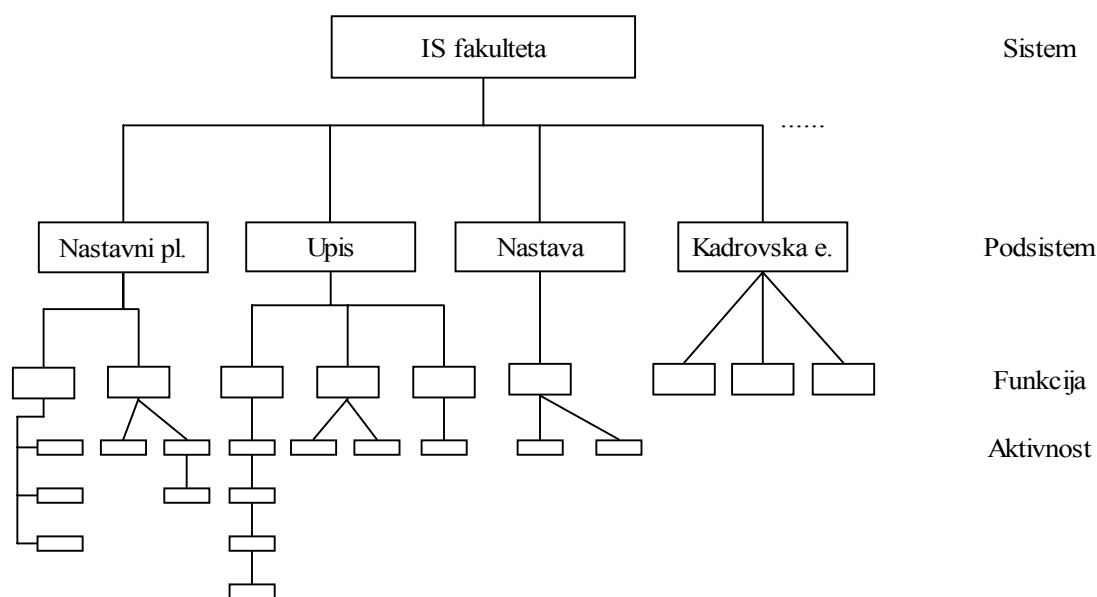
1. sakupljanje svih dokumenata (formulara) u okviru cirkulacije dokumentacije zajedno sa pošaljiocem i primaocem (organizacionim jedinicama),
2. izbor i evidentiranje najvažnijih dokumenata sa tačke gledišta informacionog sistema,
3. sastavljanje dijagrama cirkulacije dokumentacije (sa organizacionim jedinicama i dokumentima koji cirkulišu između njih),
4. definisanje granice sistema na dijagramu cirkulacije dokumentacije,
5. zamena organizacionih jedinica sa procesima i naznačavanjem daljih tokova podataka između njih,
6. verifikacija tako dobijenog dijagrama toka podataka na celovitost, jednoznačnost i tačnost,
7. sprovođenje potrebnih korekcija i
8. hijerarhijska dekompozicija tako dobijenog dijagrama toka podataka po potrebi uz ponavljanje koraka 6., 7. i 8. sve do nivoa elementarnih procesa.

Neophodno je ukazati na važnost tesne saradnje između projekta(n)a i korisnika u toku izgradnje dijagrama cirkulacije dokumentacije i DTP.

8.5.3. Makro-dizajn

Kao što je napred istaknuto, DTP i graf za rukovanje podataka predstavljaju opis i modeliranje obrade na srednjem nivou. Sistemski prilaz razvoju informacionih sistema, medjutim, nalaže top-down pristup u rešavanju zadataka. Prvi korak u makro-projektovanju obrade je odredjivanje podsistema posmatranog informacionog sistema. Podsystemi obuhvataju srodne funkcije, a funkcije se sastoje od više aktivnosti (operacija) gradeći hijerarhijsku strukturu aktivnosti izgradjenu od “elemenata različite složenosti” (aktivnost, funkcija, podsystem i sistem) prikazanu na sl. 8.8.

U cilju objektivne predstave arhitekture obrade (funkcionalnog modela) informacionog sistema najčešće se primenjuje poznata algoritam ili metodološki korak BSP (Business System Planning) metode za planiranje razvoja informacionog sistema pod nazivom “Definisanje arhitekture informacionog sistema”.



Sl. 8.8.

BSP metodologiju je razvio IBM za planiranje razvoja informacionih sistema. Njeno sprovođenje se često naziva i strateškim planiranjem. Ona obuhvata definisanje opšte strukture informacionog sistema, njegovih podsistema sa pripadajućim medjuzavisnostima izmedju njih i plana i prioriteta njegovog (njihovog) faznog uvođenja. O BSP metodologiji postoji obimna literatura, a kratki opisi se mogu pronaći i u nekim radovima koji se bave razvojem informacionih sistema.

Osnovni principi na kojima se zasniva BSP metodologija su i danas aktuelni:

1. "Informacioni sistem treba da podržava smernice razvoja i ciljeve poslovanja organizacije, prenošenjem poslovne strategije na strategiju informacionog sistema, primenom metode strukturiranog pristupa,
2. Informacioni sistem mora zadovoljavati informacione potrebe svih nivou upravljanja, planiranja i kontrole u sklopu poslovnog sistema,
3. Informacioni sistem treba da obezbeđuje konzistentnost informacija,
4. Informacioni sistem mora biti sposoban za dugoročni razvoj nezavisan od organizacionih i upravljačkih promena,
5. Informacioni sistem, koji podržava sve informacione potrebe, mora posedovati mogućnost realizacije po delovima definisane informacione strukture odnosno korak-po-korak ili projekat po projekat."¹

Sama metodologija se sprovodi u više koraka:

1. Pribavljanje saglasnosti za sprovođenje studije (metodologije),
2. Priprema za primenu studije,
3. Održavanje prvog niza sastanaka u vezi primene metodologije,
4. Definisanje poslovnih procesa,
5. Definisanje klase podataka,
6. Alaliza postojećeg informacionog sistema,
7. Sprovođenje intervjua u vezi izvršenih prethodnih aktivnosti (koraka),
8. Analiza rezultata intervjua i preko njega otkrivenih problema, zaključci,
9. Definisanje arhitekture informacionog sistema,
10. Utvrđivanje faza i prioriteta izgradnje informacionog sistema,
11. Razrada plana realizacije informacionog sistema i
12. Izveštavanje i prezentacija BSP projekta.

Sa tačke gledišta određivanja podsistema nekog konkretnog informacionog sistema iz nabrojanih koraka metodologije su interesantni samo četvrti, peti i deveti korak odnosno određivanje poslovnih procesa, klase podataka i strukture (arhitekture) informacionog sistema.

Poslovni proces je organizovani skup logički povezanih aktivnosti i odlučivanja koji se realizuje u cilju upravljanja nekim resursom poslovnog sistema.

BSP metodologija predlaže klasifikaciju procesa u tri kategorije:

1. procesi planiranja i upravljanja,
2. procesi proizvodnje/usluga
3. pomoćni procesi.

¹ Izvor: [7.], strana: 26.

Najvažniji zadatak u definisanju poslovnih procesa je određivanje (identifikovanje) glavnog proizvoda (odnosno usluga ili njihovu grupu) koji se realizuje temeljitom analizom hijerarhije poslovnih ciljeva preduzeća. Otkrivanje procesa se vrši preko sledeće četiri faze životnog ciklusa:

1. planiranje,
2. proizvodnja i tehnologija,
3. praćenje proizvodnje i
4. zatvaranje ciklusa.

Nakon definisanja poslovnih procesa u navedene tri kategorije i četiri faze životnog ciklusa pristupa se analizi i agregaciji (grupisanju) poslovnih procesa u četiri do dvanaest grupa procesa.

Po sistemskom pristupu klasa podataka se može smatrati organizovanim skupom podataka o nekom entitetu. Uopšteno se može reći da skup klasa podataka odgovara projekciji podataka koja je već u prethodnom delu izlaganja definisana. Otkrivanje klasa podataka se realizuje na sledeća dva načina:

1. podaci koji opisuju iste poslovne entitete (pojmovi koji se žele evidentirati i opisivati podacima smatraju se entitetima) spadaju u istu klasu podataka i
2. analizom definisanih poslovnih procesa je moguće doći do logički povezanih podataka koji određuju posebnu klasu.

Na početku treba uzeti u obzir entitete na najvišem nivou generalizacije kao što su:

1. partneri u poslovanju,
2. predmeti poslovanja,
3. obaveze u poslovanju,
4. transakcije u poslovanju i
5. subjekti poslovanja.

Posle toga se entiteti na najvišem nivou generalizacije dele na posebne, manje uopštene kategorije, kao što se partneri u poslovanju dele na kupce i dobavljače, predmeti poslovanja na sirovine, poluproizvode, proizvode, opremu i usluge, obaveze u poslovanju na planove i ugovore, itd.

Definisane klase podataka se uparaju sa prethodno definisanim poslovnim procesima u okviru matrice odnosa poslovni procesi/klase podataka. Proces se u toj matrici smeštaju u redove po redosledu njihovog događanja u životnom ciklusu, a klase u kolone na taj način, da klasa u prvoj koloni bude ona koju kreira prvi proces, klasa u drugoj koju kreira drugi proces itd. Tako se grupisanjem procesa prema generalisanim poslovnim resursima kojima se u njima upravlja dobijaju grupe procesa koji se smatraju podsistemima a i klase podataka koje njima

pripadaju. Grupisanje procesa i klasa podataka rezultuje blokove uglavnom na dijagonali matrice odnosa poslovni procesi/klase podataka, a blokovi se konstruišu na taj način da je prvi proces i prva klasa podataka koju isti kreira predstavlja gornju levu granicu pravougaonika (bloka), a poslednji proces u logičkoj celini i poslednja klasa podataka koja se formira sa njegove strane predstavlja donju desnu granicu informacionog podsistema (bloka).

Mesto BSP metodologije se može videti iz klasifikacije koraka u razvoju informacionih sistema:

1. planiranje razvoja sistema,
2. analiza i dizajn informacionog sistema,
3. implementacija sistema,
4. održavanje sistema i
5. vrednovanje i kontrola sistema.

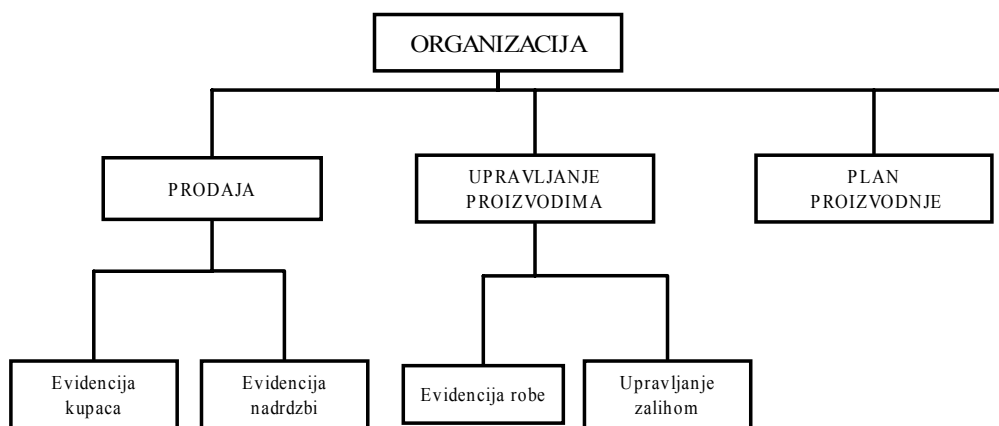
Mnogi stručnjaci smatraju da BSP više spada u oblast project managementa nego u sam razvoj informacionih sistema, tako da postoje druge vrste klasifikacije razvoja informacionih sistema koji ne obuhvataju planiranje razvoja sistema, a time i BSP.

Treba, međutim, istaći da je korak BSP metodologije “Definisanje arhitekture informacionog sistema” jako praktičan i koristan i u oblasti makro-projektovanja obrade. U tom slučaju se konstruiše matrica odnosa funkcija (grupisani procesi)/tip entiteta. Za ilustraciju primene BSP metodologije u određivanju podsistema i hijerarhiji obrade koristićemo primer prikazan u [11.] u tabeli 8.2.

Tabela 8.2

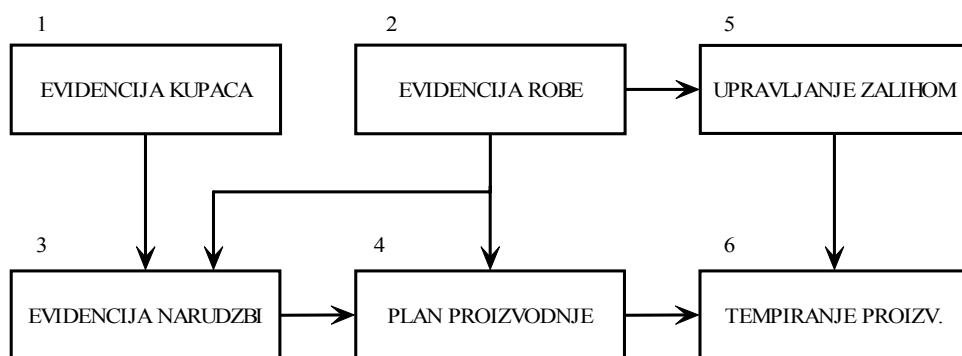
Entiteti (Klase podataka)	K u p a c i	U g o v o r i	N a r u d ž b i	C e n o v n i k	R o b o t o r i	Z a l i h o m
Funkcije (Grupisani procesi)						
Evidencija kupaca	C	C	-	-	-	-
Evidencija narudžbi	C	U	C	U	U	U
Evidencija robe	-	-	U	C	C	U
Upravljanje zalihom	-	-	-	U	C	C
Plan proizvodnje	-	-	U	U	-	-
Tempiranje proizvodnjom	-	-	U	U	-	-

Značenje oznaka u matrici odnosa je sledeće: **C** – formira (kreira, create), **U** – koristi (use). Na osnovu prethodne tabele se jednoznačno može zaključiti kojim grupama funkcija koje klase podataka pripadaju (vezuju). Na osnovu te informacije je moguće predstaviti statičku strukturu (arhitekturu) sistema obrade koja je prikazana na sl. 8.9.



Sl. 8.9.

Informacioni sistem se priprema u cilju korišćenja odnosno funkcionisanja, a o njegovom funkcionisanju statička slika (sl 8.9.) ništa ne kaže. Logiku redosleda obrade odražava dinamika veze izmedju pojedinih obrada koja predstavlja funkcionisanje sistema. Slika sl. 8.10. prikazuje mrežu obrade sa naznačenim redosledom izvršavanja funkcija.



Sl. 8.10.

8.5.4. Mikro-dizajn

Makro-projektovanje služi za određivanje podsistema informacionih sistema, tj. šta treba globalno raditi. Kao rezultat se određuju glavne grupe funkcija (delovi sistema) koje se pojavljuju u posmatranom informacionom sistemu. Taj nivo planiranja projekcije obrade se smatra konceptualnim jer se bavi samo pitanjem šta treba uraditi.

Podsistemi informacionih sistema predstavljaju organizovani skup funkcija. Određivanje i organizovanje funkcija u okviru podsistema predstavlja logički nivo planiranja projekcije obrade, jer opisuje kako treba realizovati glavne grupe funkcija. Tu problematiku analiziraju tehnike za vizuelnu predstavu na srednjem nivou abstrakcije lanca **ulaz – obrada1 – baza_podataka – obrada2 – izlaz**. DTP, recimo, sadrži glavne aktivnosti (funkcije) i veće jedinice podataka (ponekad samo tipove entiteta), a ukazuje na upravljačke operacije i na operacije za rukovanje podacima.

U obradi podataka se koriste tri vrste operacije: za rukovanje podacima, za stvaranje novih podataka i upravljačke operacije. U praksi se najčešće pažnja poklanja samo operacijama za rukovanje podacima: to prejudiciraju grafovi za rukovanje podataka i dijagrami tokova podataka na pomenutom srednjem nivou projektovanja obrade. U cilju postizanja jednoznačnosti i celovitosti upravljačkih operacija neophodno je analizu i projektovanje obrade realizovati na najnižem (mikro) nivou pomoću jedne davno poznate (danas skoro zaboravljene) tehnike, koja se zove tabela odlučivanja. Treća vrsta operacije – stvaranje novih podataka – takodje iziskuje aktivnosti na mikro-nivou analize i dizajna. U cilju temeljitog opisa stvaranja novih podataka najsvrsishodnije je koristiti mreže (grafove) precedencije.

8.5.4.1. Tabele odlučivanja

U izvršavanju pojedinih funkcija obrade podataka mogu nastati složene situacije u uslovima koji upravljaju izvršavanjem pojedinih operacija u okviru funkcije. Uslove koji određuju izvršavanje funkcije nije preporučljivo ugraditi u programe zbog njihove brze promene. Zgodno je takve složene situacije, odnosno uslove koje one definišu formalno definisati i smestiti u tzv. tabele odlučivanja. Tabele odlučivanja predstavljaju pogodno sredstvo za definisanje, analizu i projektovanje rešenja za složene situacije odlučivanja, naročito u situacijama brzih promena uslova privredjivanja. Tabele odlučivanja se, znači, mogu koristiti i u dizajniranju programa, a i u proceduri za pripremanje poslovnih odluka (potpuno nezavisno i od

računara). Tabele odlučivanja mogu biti različite. Tabela 8.3. prikazuje opštu strukturu tabele odlučivanja po [Raffai Maria Informaciorendszer-fejlesztés] i [Branislav Lazarević i o.].

Tabela 8.3.

		Uslovi	Matrica vrednosti
IF	O	U_1	$e_{11} e_{12} \dots e_{1k}$
	D	U_2	$e_{21} e_{22} \dots e_{2k}$
	L	U_3	$e_{31} e_{32} \dots e_{3k}$
	U	.	.
	K	.	.
	A	U_n	$e_{n1} e_{n2} \dots e_{nk}$
THEN	A	A_1	$a_{11} a_{12} \dots a_{1k}$
	K	A_2	$a_{21} a_{22} \dots a_{2k}$
	C	A_3	$a_{31} a_{32} \dots a_{3k}$
	I	.	.
	J	.	.
	E	A_m	$a_{m1} a_{m2} \dots a_{mk}$
		Aktivnosti	Matrica funkcija

Tabela odlučivanja je na tabeli 8.3. označena pojačanom linijom i predstavlja pomoć pri izboru adekvatnog niza aktivnosti u složenim situacijama odlučivanja. Tabela odlučivanja se sastoji od dva dela: od dela odluke (gornji deo tabele) i od dela akcije (donji deo tabele).

U delu odluke svakom uslovu (U_i) se dodeljuju vrednosti u okviru matrice vrednosti. Matrica vrednosti se često puta naziva i označava kao ostvareni uslovi. Uslovi definisani u prvoj četvrtini matrice odlučivanja u drugoj četvrtini matrice imaju simbole (podatke) o svojim realizacijama ili ostvarenjima. Simboli ostvarenih uslova (članovi matrice vrednosti e_{ij}) su najčešće samo Y (yes), N (no) ili – (što znači da uslov ne utiče na rešenje). Treća četvrtina sadrži moguće aktivnosti za izvršavanje, a četvrta četvrtina predstavlja matricu funkcija, koja se ponekad naziva i izabranim akcijama (aktivnostima). Oznake koje se koriste u delu izabranih akcija (elementi matrice funkcije a_{ij}) su X (označava obavezu ili potrebu izvršenja aktivnosti u čijem redu se oznaka X pojavljuje) ili – što znači da ne treba izvršiti akciju.

Školski primer će biti tabela odlučivanja pri određivanju korena kvadratne jednačine. Pretpostavimo opštu kvadratnu jednačinu oblika: $ax^2+bx+c=0$. U tom slučaju tabela odlučivanja za računanje korena jednačine može biti oblika kao u tabeli 8.4.

Tabela 8.4.

$b^2 - 4ac > 0$	Y	N	N
$b^2 - 4ac = 0$	N	Y	N
$b^2 - 4ac < 0$	N	N	Y
$rex = -\frac{b}{2a} \quad imx = \frac{\sqrt{b^2 - 4ac}}{2a}$	X	X	X
$imx1 = 0 \quad imx2 = 0$	X	X	-
$rex1 = rex \quad rex2 = rex$	X	X	X
$rex1 = rex1 + imx \quad rex2 = rex2 - imx$	X	-	-
$imx1 = imx \quad imx2 = -imx$	-	-	X

8.5.4.2. Grafovi precedencije

Za pravilno funkcionisanje obrade podataka upravljačke operacije se moraju pažljivo projektovati ne bi li se neka kombinacija u slučaju složenih uslova ispustila. Tabele odlučivanja pomažu u definisanju, analizi i projektovanju svih mogućih alternativa u složenim situacijama odlučivanja. U operacijama (aktivnostima) stvaranja novih podataka ili kada se formuliše, projektuje izlaz neke funkcije – neophodno je istaći kako se može stvoriti željeni izlaz na osnovu postojećih podataka. Za formalizovani način definisanja i analize stvaranja novih podataka i/ili sastavljanja izveštaja (odgovora na korisnički informacioni zahtev), sa tačke gledišta potrebnih podataka za njihovu realizaciju, koriste se mreže (grafovi) precedencije.

Prekid i modifikacija mreže precedencije funkcija obrade predstavljaju najveći problem u funkcionisanju informacionog sistema kao celine. Danas ne postoji ona klasična “direktna” obrada podataka (ulazni podaci se danas neposredno ne prebacuju – nakon potrebnih obrada – odmah na izlaz), već se ulazni podaci memorišu negde u bazi podataka (kao matični podaci). Formula oblika

ulaz – obrada – izlaz

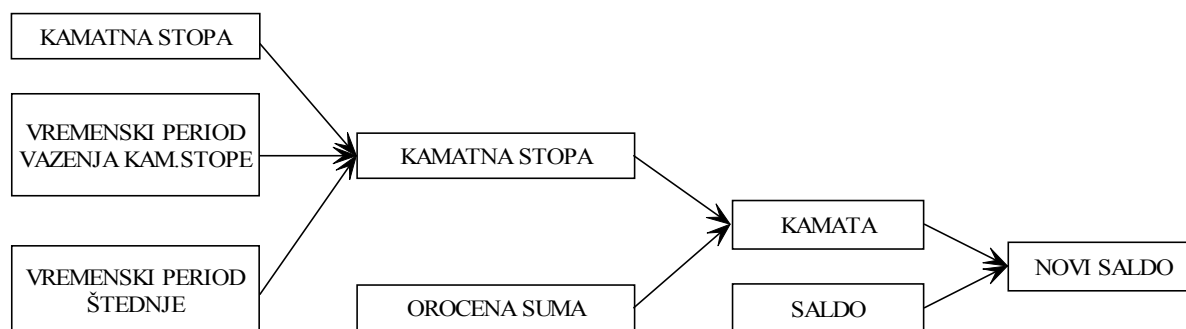
se danas smatra prevaziđenom i umesto nje se koriste sledeće dve:

ulaz – obrada1 – baza podataka

i

baza podataka – obrada2 – izlaz.

Operacije (obrade) funkcija su tako rasplinite u celom informacionom sistemu, a time mreža precedencije postaje još važnija. Mreža precedencije inače predstavlja unutrašnju strukturu funkcionalnog modela ili konceptualnog modela operacija sa tačke gledišta podataka potrebnih za izvršavanje konkretne funkcije. Kao primer na sl. 8.11. je prikazana mreža precedencije za obračun kamate štedišama za oročena sredstva.



Sl. 8.11.

8.5.5. Konzistencija obrade

Analiza i projektovanje obrade se najčešće obavljaju na srednjem nivou pomoću dijagrama toka podataka. DTP odlikava aktivnosti i podatke shodno njihovom redosledu obavljanja. Dijagram konteksta (nulti nivo) prikazuje ceo informacioni sistem kao jednu aktivnost, na prvom nivou se prikazuju grupe srodnih funkcija (podsistemi), a sledeći nivoi već odražavaju detalje – aktivnosti koje se tiču vrednosti obeležja pojedinih tipova entiteta.

Odredjivanje glavnih grupa procesa (glavne funkcije – podsisteme) na osnovu cirkulacije dokumenata, tj. dijagrama toka podataka može biti subjektivno. U cilju objektivnog formulisanja (definisanja) podsistema koristi se makro-projektovanje. U praksi se, u tu svrhu, najčešće primenjuje korak BSP (Business System Planning) metode za planiranje razvoja informacionog sistema pod nazivom “Definisanje arhitekture informacionog sistema”. Time se obezbedjuje uskladenost izmedju makro-dizajna i projektovanja na srednjem nivou abstrakcije obrade.

Nivoi DTP-a koji prikazuju detalje treba da odražavaju ponekad i složene situacije u uslovima koji upravljaju izvršavanjem pojedinih operacija u okviru funkcije. U tim situacijama su od velike pomoći tabele odlučivanja u definisanju, analizi i projektovanju svih mogućih alternativa u složenim situacijama odlučivanja. DTP-i prikazuju i izlaze (izveštaje) informacionog sistema. U sastavljanju odgovora na informacione zahteve korisnika neophodno

je znati kako se može stvoriti željeni izlaz na osnovu postojećih podataka. Mreže precedencije pomažu (na jedan formalizovani način) u definisanju i analizi stvaranja novih podataka i/ili sastavljanja izveštaja, sa tačke gledišta potrebnih podataka za njihovu realizaciju. Time se realizuje konzistencija između mikro-dizajna i projektovanja na srednjem nivou obrade.

DTP prikazuje ulaze, skladišta podataka i izlaze sistema, ali projektant/analitičar nema nikakve informacije o tome, da li se željeni (zahtevani) izlaz može stvoriti na osnovu podataka koji stoje na raspolaganju, sa jedne strane, i sa druge strane, da li su svi ulazni podaci potrebni za dobijanje izlaza, tj. da li skup ulaznih podataka predstavlja neredundantni skup podataka za konstruisanje željenog izveštaja (izlaza). DTP ne sadrži podatke o tome kako se ulazni podaci koriste (i koji su ti podaci, koji se uopšte ne koriste) u procesima, a mreže precedencije ne vode brigu o tome odakle potiču potrebni podaci u transformacionom lancu kreiranja izlaza.

Potreban i dovoljan uslov na polju raspoloživosti podataka za stvaranje izlaza (da li su svi podaci potrebni i da li su dovoljni za generisanje predmetnog izveštaja) mogu obezbediti HIPO (Hierarchy plus Input Process Output) dijagrami. HIPO metodu je razvio IBM sa ciljem da bude sredstvo za dokumentaciju složenih softverskih proizvoda (informacionih sistema). Metoda je top-down orijentacije i obuhvata od opisa glavnih grupa funkcija (podsistema), preko funkcija do nivoa samih procesa obrade koji na osnovu ulaza daju odgovarajuće izlaze. HIPO metoda se sastoji od sledećih komponenti [B.Lazarević i drugi 1986 I deo]:

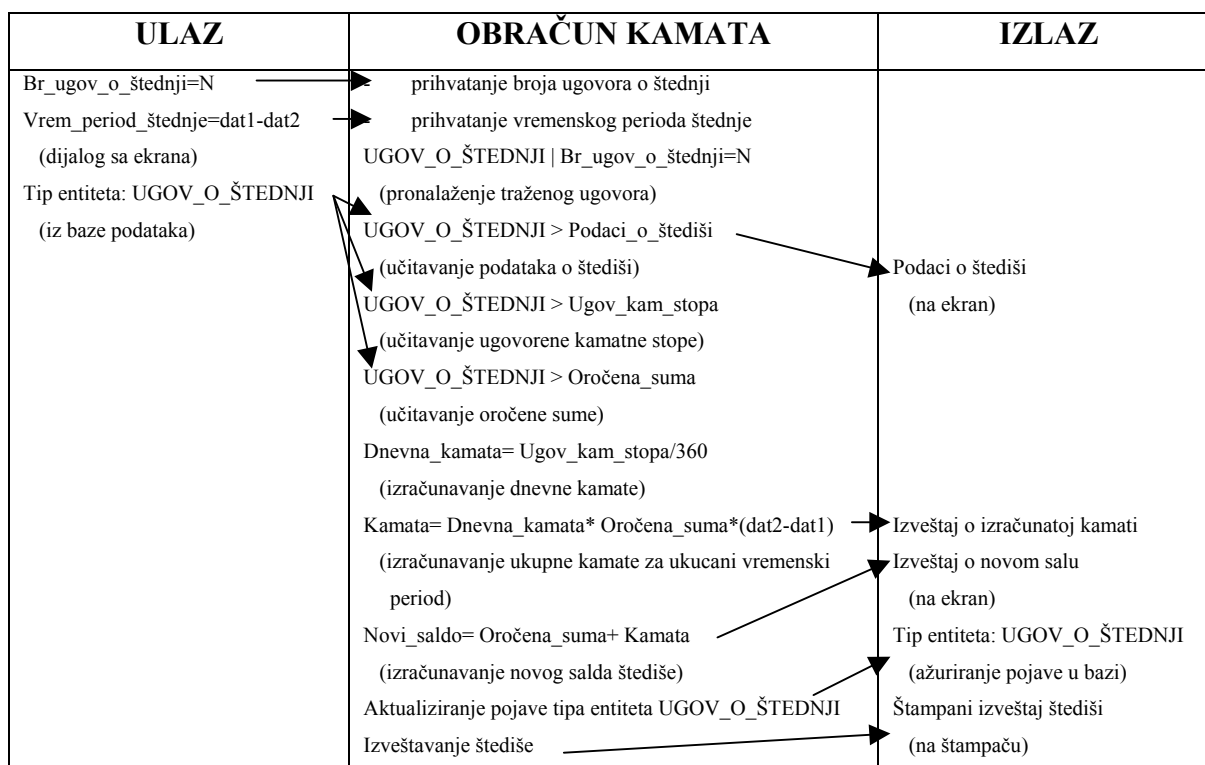
1. Vizuelni sadržaj,
2. Pregledni dijagrami i
3. Detaljni dijagrami.

Vizuelni sadržaj HIPO metode predstavlja prvi nivo dekompozicije samog informacionog sistema. Pošto za određivanje podsistema (a to je upravo prvi nivo dekompozicije informacionog sistema) već opisan korak BSP metode “Definisanje arhitekture informacionog sistema”, sa vizuelnim sadržajem HIPO metode se nećemo upoznati. Sam rezultat primene ili “proizvod” vizuelnog sadržaja je isti kao kod pomenutog koraka BSP metode koji je prikazan na primeru u okviru tabele 8.2. i sl. 8.9.

Pregledni dijagrami predstavljaju opise funkcija na višim nivoima hijerarhije, bez prikaza logike realizacije (odvijanja) operacija (elementarnih procesa). Oni takodje nisu u stanju da daju informacije ili sude o dovoljnosti i/ili neredundantnosti raspoloživih podataka u vezi generisanja izlaza.

Detaljni dijagrami sadrže sve pojedinosti u vezi svake operacije (procesa) na najnižem nivou hijerarhije. U slučaju preglednih dokumenata ulazi i izlazi su obično dokumenti ili tipovi

entiteta. Detaljni dijagrami kao ulaze i izlaze sadrže konkretizacije obeležja nekog tipa entiteta. Primer za detaljni HIPO dijagram je na sl. 8.12.



9. PROJEKCIJA OKRUŽENJA INFORMACIONOG SISTEMA

Projekcija okruženja informacionog sistema, kako je istaknuto na početku izlaganja, obuhvata korisnike, potrebne resurse i standarde. Resursi potrebni za oživotvorenje zamisli ili ideje u vidu informacionog sistema su vreme, novac i tehnička sredstva (hardver i softver). O tehničkim sredstvima (generalno) je bilo reči u poglavlju 5, no hardver i softver se tako brzo menjaju da nema smisla (a ne bi bilo ni moguće sve varijacije raspoloživog hardvera i razne generacije – verzije softvera obraditi) opisati neko konkretno (bilo hardversko ili softversko) rešenje. Vreme i novac spadaju u kategoriju projekt menedžmenta i ne predstavljaju predmet posmatranja u ovom izlaganju.

U okviru projekcije okruženja su tako ostali samo korisnici i standardi. Oni će biti predmet interesovanja u ovoj, relativno kratkoj glavi.

9.1.KORISNIČKO OKRUŽENJE

Projekcija okruženja predstavlja objektivno postojeću sredinu (okolinu) u kojoj treba razvijati, realizovati i koristiti budući informacioni sistem. To važi i za korisnike, odnosno korisničko okruženje. Njega ne treba izmisliti, formulisati i predstaviti, ono postoji.

Korisničko okruženje sačinjavaju krajnji korisnici (end user), aplikativni korisnici, članovi razvojnog tima, rukovodioci, pa čak, u izvesnom smislu i organizacione jedinice kompanije naručioca.

9.1.1. Psihologija korisnika

Informacioni sistem je namenjen krajnjim korisnicima. Njihov zahtev nije uvek jednoznačno formulisani sa jedne strane, sa druge strane krajnji korisnici ne predstavljaju homogenu grupu ljudi: spadaju u različite socijalne i kulturne grupacije (slojeve). Kod sastavljanja upitnika (intervjua) za krajnje korisnike i generisanja izveštaja za njih o tome bi trebalo voditi računa. Razvojni tim treba da bude spreman za komunikaciju sa nehomogenom grupom krajnjih korisnika (naravno ne iz dana u dan i ne sa svakim krajnjim korisnikom

pojedinačno). Izveštaji namenjeni krajnjim korisnicima treba da budu razumljivi njima, a ne projektantu ili analitičaru.

Aplikativni korisnici obavljaju samo konkretne funkcije u okviru informacionog sistema. Oni predstavljaju manju grupu po broju od krajnjih korisnika, a takodje su manje nehomogeni, tj. predstavljaju kompaktniju grupu od krajnjih korisnika. Koriste specifične izraze koji odražavaju žargon oblasti struke iz koje potiču. Njihov život razvojni tim može da olakša jednostavnim i standardizovanim menijima i maskama za unos koji su u saglasnosti sa njihovim nivoima znanja i sa njihovim svakidašnjim poslovima. Aplikativni korisnici treba da dobiju odgovarajuću dokumentaciju o funkcionisanju za njih interesantnih delova informacionog sistema, a po potrebi treba obezbediti i obuku za njih.

Članovi razvojnog tima su takodje ljudi, neki od njih su ljubomorni na svoje znanje, neki su arogantni, a ima i onih koji teže saradjuju sa drugima. Razvijeni sistem neće biti dobar ako projektant i analitičar prvenstveno (ali i ostali članovi razvojnog tima) nemaju sposobnost imaginacije i ne mogu sebe smestiti u ulogu krajnjeg ili aplikativnog korisnika. Dobar proizvod razvoja iziskuje napor neposrednih korisnika i članova tima kroz dogovore, sastanke i usaglašavanja mišljenja.

Rukovodilac može imati veliki udeo u uspešno realizovanom projektu razvoja informacionog sistema. Svojim zalaganjem pokazuje primer svojim saradnicima (ako rukovodioca ne interesuje sudbina razvoja, njegove će potčinjene još manje interesovati). On može da nadzire ispoštovanje dogovorenih rokova za izvršenje zadataka svojih radnika. Razvojni tim treba da izdejstvuje tesnu saradnju i sa njih. Na osnovu napred izloženih se može zaključiti, da je za informatičara jako poželjno znanje iz psihologije i zavidan nivo komunikativne sposobnosti.

9.1.2. Psihologija rada

Psihologija rada na razvoju informacionog sistema spada u delokrug projekt menedžmenta, ali je korisno ukaziti, bar u kratkim crtama, koji su najčešći problemi razvoja. Razvojni tim za informacioni sistem je sastavljen od većeg broja stručnjaka različitog profila. Članovi nisu permanentno angažovani za vreme perioda razvoja: na primer, analitičari i projektanti su aktivniji na početku i u fazi testiranja sistema, a programeri u fazi dizajniranja, kodiranja i testiranja programa. Nakon perioda razvoja (tri važna perioda u životu informacionih sistema su: razvoj, eksploatacija i prilagođavanje novim uslovima – change management) sledi

eksploatacija informacionog sistema, a tu su i neizbežne modifikacije sistema. Korišćenje sistema se često meša sa održavanjem (prilagodjavanjem) sistema. Stručnjaci eksploatacije često koriguju programe da bi sistem prilagodili svojim “navikama”, a to ne spada u njihov delokrug rada. Svaka promena je problem za sebe, i nju treba da analiziraju oni koji imaju predstavu o celini problema (informacionog sistema). Svojstvo da se informacioni sistem menja (prilagodjava se novim zahtevima i uslovima) je objektivna stvarnost, koja takodje spada u psihologiju rada sa sistemom.

Rukovodilac projektnog tima treba da upravlja grupom stručnjaka u okviru koje su pojedinci sasvim različitog interesovanja. Projektante više interesuju korisnici i njihovi zahtevi, tj. funkcionisanje sistema kao celine, a programeri su nestrpljivi i odmah bi pisali programe u cilju da vide rezultate svog rada. U slučaju grešaka pri validaciji sistema, projektanti se žale da programeri ne znaju pisati programe, a programeri da analitičari ne znaju dokumentovati sistem. Ako razvoj projekta realizuje neko specijalizovano preduzeće za projektovanje informacionih sistema, oni su zbog neravnomernog opterećenja pojedinih članova razvojnog tima organizovani po tzv. matričnom tipu organizacije (o njemu je bilo reči u tački 2.2).

Sledeći momenat u psihologiji rada u vezi informacionog sistema je mešanje pojmova u vezi pojedinih funkcija i delokrugova rada grupa u razvojnog timu. Poslovi rukovodioca informatike, administratora sistema i administratora podataka se mešaju.

Rukovodioc informatike (CIO – Chief Information Manager) je pre svega menedžer i bavi se samo makro aspektima problema informatičke oblasti: razrada informatičke koncepcije, obezbeđenje potrebnih resursa za funkcionisanje, grubo tempiranje poslova, itd. Administrator sistema (System Administrator) je zadužen za nadzor funkcionisanja (a ne projektovanja) informacionog sistema (u slučaju velikih kompanija postoje posebni administratori za pojedine podsisteme). Njegovi problemi su delimično menedžerske, a pretežno tehničke prirode koji se tiču eksploatacije i sa njom povezanih oblasti. Administrator podataka (Data Base Administrator) je odgovoran za jedinstveni model podataka (i za jedinstvenu bazu podataka) na nivou kompanije (preduzeća) kao celine. Njegov je zadatak da brine o standardima i (standardnim) procedurama, da ih razvija, da sa tačke gledišta modela podataka obrade u sistemu usaglašava i da brine (održava) o dokumentaciji informacionog sistema.

Dve problematike u vezi informacionog sistema su manje poznate za prosečnog korisnika/naručioca, a to su kontrola kvaliteta (Quality Control) i prilagodjavanje promenama (Change Management). Za kontrolu kvaliteta su od najvećeg značaja propisani ili prihvaćeni standardi, a za efikasno prilagodjavanje informacionog sistema promenama u uslovima privredjivanja je neophodno izvršiti reprojektovanje sistema.

9.2.SISTEM STANDARDA

Medjunarodno prihvaćeni opšti standardi na polju razvoja i korišćenja informacionih sistema ne postoje. Zato preduzeća treba da izgrade svoje unutrašnje standarde. Prihvaćeni standard propisuje korišćenje zajedničkog jezika (u najširem mogućem značenju) na polju: osnovne koncepcije, strukture, terminologije, načina predstavljanja, sistema kriterijuma, algoritma projektovanja, rukovanja dokumentacijom i meta-standarda.

Izgradnja internih standarda ne može biti samovoljna. Prihvaćeni standardi ne mogu izolovano živeti i opstati, jer je informacioni sistem tesno povezan sa spoljašnjim (propisanim – zakonski propisi ili neobaveznim – napr. ISO preporukama) standardima. Tehnička sredstva (softveri i hardver) na kojima se realizuje razvoj, a kasnije eksploatacija sistema takodje mogu nametnuti neke propisane modele ponašanja ili aktivnosti. Korisnici raznih specijalnosti imaju raznolike prilaze istom problemu ili objektu posmatranja: koriste posebnu terminologiju, simboli predstavljanja su drugačiji, itd. Ovakvi heterogeni propisi, konvencije, navike, preporuke i prilazi treba da se sinhronizuju u internim standardima, što je mukotrpan i nezahvalan posao, jer će sinhronizacija naići na otpor u svim oblastima koje obuhvataju standardi.

Standard i sam predstavlja sistem jer se sastoji od medjusobno povezanih elemenata (zakona, preporuka i procedura). O tome je bilo već reči u tački 4.2.5. Standard je i apstrakcija, jer propise je trebalo smisliti i formulisati, a takodje i njihovu konstrukciju (strukturu) odrediti. Postje (ili barem bi trebalo da postoje) korisnički i razvojni (meta) standardi.

Korisnički standardi se odnose na činioce informacionog sistema kao što su podaci, aktivnosti, događaji, tehnička sredstva itd. Meta standardi (o njima je bilo reči u 6.3.8.) predstavljaju osnovu za rad projekt menedžmenta, jer nalažu način razvoja sistema (i njegovu formulaciju). Pored tehnike razvoja meta standardi propisuju metode (tehnike) programiranja, testiranja i predaje-peuzimanja gotovog informacionog sistema, odnosno sve podatke, događaje, aktivnosti, tehnička sredstva, itd. koji se odnose na razvoj informacionih sistema.

U okviru meta standarda se nalaze i standardi za dokumentaciju. Oni treba da formulišu zahteve u vezi sadržaja, strukture i načina predstavljanja (prikazivanja) meta podataka (pošto dokumentacija sadrži podatke koji se odnose na korisničke podatke – tj. saznanja o saznanjima – tako se ona često naziva metabazom podataka). Nije, međjutim, bitno koja tehnička sredstva omogućuju realizaciju dokumentacije (ona se može, na kraju krajeva i ručno uraditi, samo, naravno, nema puno smisla, pošto postoje softveri koji to mogu brže ostvariti). Kvalitet dokumentacije odredjuju sposobnost ljudi i prihvaćenih standarda, a nikako sredstva za njenu realizaciju.

9.3.DOKUMENTACIJA INFORMACIONOG SISTEMA

Standardi se mogu odnositi na sam razvoj informacionog sistema i tada se nazivaju meta standardima, odnosno pored standarda razlikujemo i meta standarde. Slično tom prilazu [Halassy E, I, R] dokumentaciju informacionog sistema sačinjavaju dva glavne vrste dokumenata:

1. projektna dokumentacija i
2. dokumentacija informacionog sistema.

Projektna dokumentacija odslikava proces razvoja informacionog sistema sa svim svojim administrativnim sadržajima, i može biti interesantna za proširivanje iskustva iz projekt menedžmenta.

Dokumentacija informacionog sistema sadrži projekat sistema u standardizovanoj formi (ako su se radnici pri razvoju – izrade – dokumentacije pridržavali nekim prihvaćenim standardima). Ona obuhvata sledeće tri vrste dokumentacije:

1. projektna dokumentacija,
2. korisnička dokumentacija i
3. dokumentacija za održavanje.

Knjiga projektna dokumentacija sadrži opis informacionog sistema u vidu dijagrama, tabela i (šturog, ali razumljivog i objektivnog) verbalnog opisa. Pošto ona predstavlja okosnicu za druge dve dokumentacije, nju treba kao prvu izraditi.

Prva glava projektne dokumentacije treba da sadrži tačno definiciju cilja (ili hijerarhije ciljeva) informacionog sistema, i to konkretno, taksativno. Ako je u pitanju hijerarhija ciljeva, što je najčešće slučaj, treba ukazati i na prioritet (redosled po važnosti) pojedinih podciljeva. Dobro definisan cilj određuje sadržaj sistema, jer su oni nerazdvojni. Sadržaj informacionog sistema se krije u pojmovima koji se koriste u delu realnog sistema za koji se projektuje informacioni sistem. Pojmovi određuju tipove entiteta, a, kako već znamo, oni diktiraju moguće obrade (od kojih korisnik-naručilac bira po svome, odnosno ne mora biti »kupac« za sve). Tu se povlače granice sistema, a na osnovu principa strukturiranja sistema po najstabilnijem elementu – podacima se određuje statička struktura sistema. Gruba dinamika ponašanja sistema je još poželjna ovde, u uvodu (ili prvoj glavi), i pravci budućeg razvoja.

U projektnoj dokumentaciji treba predvideti po jednu glavu za sve tri projekcije informacionog sistema, a to su projekcije:

1. podataka,
2. obrade i
3. okruženja.

Pošto prve dve projekcije predstavljaju u stvari rezultat intelektualnog napora analitičara i projekatanta, njih treba izraditi (podeliti) na sva tri abstrakciona nivoa: na konceptualnom, logičkom i fizičkom nivou. Smanjenje posla se može ostvariti primenom relacionog softvera za rukovanje bazama podataka: oni sami rešavaju pretvaranje logičkog u fizički model podataka i ne dozvoljavaju korekcije od strane programera i projektanta. U toku izrade dokumentacije za dve apstraktne projekcije treba imati na umu ANSI/SPARC preporuke za arhitekturu opisa baze podataka (Sl. 8.1.) i treba pripremiti tri vrste šeme. Ne sme se zaboraviti da se događaji (koji se opisuju podacima) i aktivnosti odigravaju i u svtarnosti, a da je njihovo preslikavanje na računar upravo taj informacioni sistem koji se dokumentuje. Organizaciju podataka na računar kao što znamo opisuje interna, a podatke u realnom svetu eksterna šema. Usaglašavanje između interne i eksterne šeme obezbeđuje konceptualna šema. Pored temeljnog opisa tri apstrakciona nivoa dokumentacija treba da ukaže na uzroke preslikavanja između pojedinih nivoa da bi se dobila celovita slika o tome zašto je došlo do deformisane slike realnog sistema (sveta).

Treća po redu projekcija okoline nema tri abstrakciona nivoa, pošto sadrži opis postojećih činilaca informacionog sistema. Tu se nalazi opis korisnika, potrebnih resursa i standarda. U slučaju promene hardverske ili softverske okoline treba modifikovati samo deo glave koji se odnosi na zamenjeni resurs.

Knjiga korisnička dokumentacija sadrži opis svih potrebnih “informacija” za korisnike informacionog sistema. Uvod je predviđen za opšti tehnički opis sistema koji je koristan za sve aplikativne korisnike. Po jednu glavu treba posvetiti svakom tipu aplikativnog korisnika, gde su opisane sve mogućnosti, akcije, događaji, podaci, izgledi ekrana, izveštaja itd. koji se mogu desiti, dobiti itd. u toku rada i korišćenja sistema, ali samo za tog konkretnog korisnika za koga je glava namenjena.

Knjiga dokumentacija za održavanje služi za stručnjake koji omogućuju ili obezbeđuju eksploataciju informacionog sistema. Ova je složenija od prethodne knjige, jer pored detaljnijeg opisa funkcionisanja sistema sa tačke gledišta aplikativnih korisnika i opisa načina održavanja tih funkcija obuhvata i opise funkcionisanja i održavanja funkcija i podataka sa aspekta celog sistema, a u cilju zadovoljavanja potreba za administratora(e) sistema i administratora baze podataka. Ovi zadnji opisi (za administratora sistema i baze podataka) su kompleksniji od prethodnih (u vezi funkcionisanja i održavanja obrade za aplikativne korisnike) i treba da stoje u posebnim glavama dokumentacije za održavanje.

Bez prihvaćenih standarda i dobro izradjene dokumentacije informacioni sistem se ne može održavati i prilagođavati novonastalim uslovima. Dokumentacija se isporučuje zajedno sa informacionim sistemom i predstavlja vlasništvo naručioca.

10.LITERATURA

1. Baga E.: Delphi másképp..., GDF, Budapest, 1999.
2. Bana I.: Az SSADM rendszerszervezési módszertan, LSI, Budapest, 1994.
3. Bennatan E.M.: SoftverProject Management: A Practitioner's Approach, Second Edition, McGRAW-HILL Book Company Europe, 1995.
4. Čarapić M., Pendić Z., Furht B., Veselinović D.: Osnovi projektovanja informacionih sistema zasnovanih na primeni računara, Tehnička knjiga, Beograd, 1974.
5. Date C.J.: "An Introduction to Database Systems", Vol. 1, Third Edition, Addison-Wesley Publ. Co., Reading, MA, 1981.
6. Đorđević B., Vodoprivredni sistemi
7. Đurković J., Tumbas P.: Analiza i projektovanje informacionih sistema – CASE STUDY, EF Subotica – ALEF Novi Sad, 1997.
8. Görög M.: Általános projektmenedzsment, Aula Kiadó, Budapest, 1996.
9. Halassy B.: Az adatbázis-tervezés alapjai és titkai, IDG, Budapest, 1994.
10. Halassy B.: Az információrendszer alapfogalmai, SZÁMALK, Budapest, 1982.
11. Halassy B.: Ember – Információ – Rendszer, IDG, Budapest, 1996.
12. Kuzmanov S., Mogin P., Petković I., Popović M.: "Automatizacija projektovanja šeme baze podataka", JAHORINA '88.
13. Lazarević B., Jovanović V., Vučković M.: "Projektovanje informacionih sistema" I deo, Naučna knjiga, Beograd, 1986.
14. Lazarević B.: "Prošireni model objekti-veze" interni materijal, Laboratorija za informacione sisteme FON-a, Beograd, april 1990.
15. Lazarević B.: Baze podataka, Radni materijal za interne kurseve, Beograd, 1989.
16. Lerner A.J.: Principi kibernetike, Tehnička knjiga, Beograd, 1970.
17. Marjanović Z.: ORACLE – relacioni sistem za upravljanje bazom podataka, Breza, Beograd, 1990.
18. Mogin P., Luković I., Govedarica M.: Principi projektovanja baza podataka, FTN-STYLOS, Novi Sad, 2000.
19. Mogin P., Luković I.: Principi baza podataka, FTN-STYLOS, Novi Sad, 1996.
20. Noszkay E.: Informatikai és rendszerszervezési alapismeretek, GDF, Budapest, 1994.
21. Purba S., Sawh D., Shah B.: How to Manage a Succesfull Software Project, John Wiley & Sons, 1995.
22. Raffai M.: Információrendszer-fejlesztés, Novodat Kiadó, Győr, 1999.
23. Strukturirana sistemska analiza, IskraDelta, Ljubljana, 1984.
24. Šuc L.: Projektovanje baza podataka, Izobrazevalni center DELTA, Ljubljana, 1985.
25. Szelezsán J.: Adatbázisok, LSI, Budapest, 1996.
26. Ullman J.D.: "Principles of Database Systems", Computer Science Press, 1980.
27. Ullmann J.D., Widom J.: A First Course in Database Systems, Prentice Hall Inc., 1997. (prevod na mađarski jezik, PANEM, Budapest, 1997.).
28. Veljović A.: Modeliranje podataka – Erwin (materijal za kurs), CIT, Beograd, 1998.
29. Veljović A.: Modeliranje procesa – Bpwin (materijal za kurs), CIT, Beograd, 1998.
30. Yourdon E., Constantine L.L.: "Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design", Prentice Hall, Inc. Engl. Cliffs, N.J., 1979.