Strukturna sistem analiza i strukturno projektovanje

Strukturna sistem analiza je metodologija funkcionalne dekompozicije sistema, pri kojoj analitičar analizira funkcije od interesa identifikujući i dekomponujući tokove podataka izmedju funkcija i analizirajući podfunkcije do nivoa elementarnih funkcija.

Sredstva funkcionalne analize su :

1. Dijagram toka podataka (DTP) prikazuje, uz minimum detalja, tokove podataka izmedju procesa obrade, izvorišta i odredišta podataka i internih skladišta podataka (datoteka, baza podataka).

2. Sredstva za predstavljanje logike procesa, kao, npr. tabele odlučivanja, drveta odlučivanja, prirodni jezik, pseudo kod, dijagrami toka programa, služe za dokumentovanje tj. opis potrebnog programa (softvera), ali i za projektovanje.

3. Rečnik podataka za model funkcija predstavlja struktuirani skup definicija i veza objekata informacionog sistema. On sadrži detaljne specifikacije funkcija, tokova podataka, struktura podataka i samih podataka.

1 Dijagram toka podataka

Dijagram toka podataka je mreža funkcija pomoću kojih se vrši transformacija ulaznih podataka u izlazne podatke. Predstavlja se simbolima:

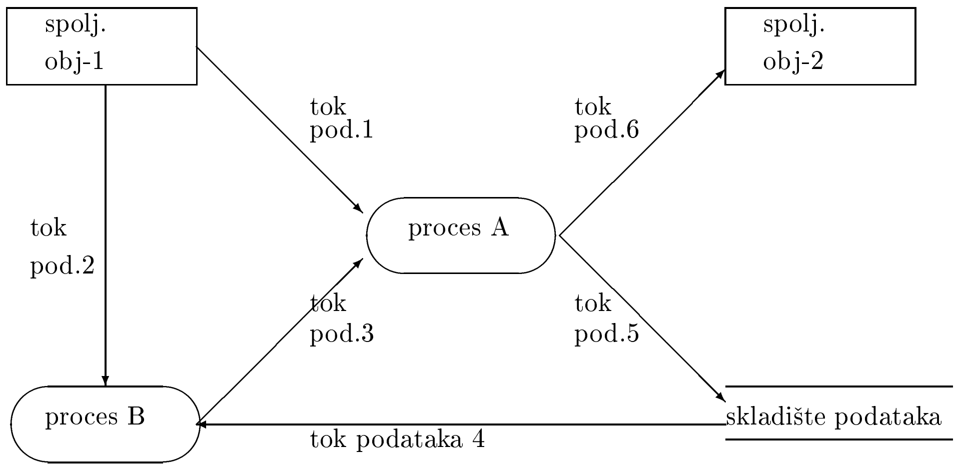
1. krugovi - procesi, funkcije; to su transformacije ulaznog toka (tokova) podataka u izlazni tok (tokove).

2. usmerene grane imenovane podacima koji po njima teku - tokovi podataka, veza;

3. dve linije normalne na tok podataka - skupovi, skladišta podataka, datoteke, baze podataka, tabele, spiskovi, kartoteke, evidencije;

4. pravougaonici - izvorišta i odredišta podataka (interfejsi) - lica, organizacione celine, tj. proizvodjači i potrošači informacija.

Svaki element na DTP ima jednoznačno ime, a ako se pojavljuju više puta (datoteke i interfejsi) označavaju se sa X (slika 5).



slika 5.

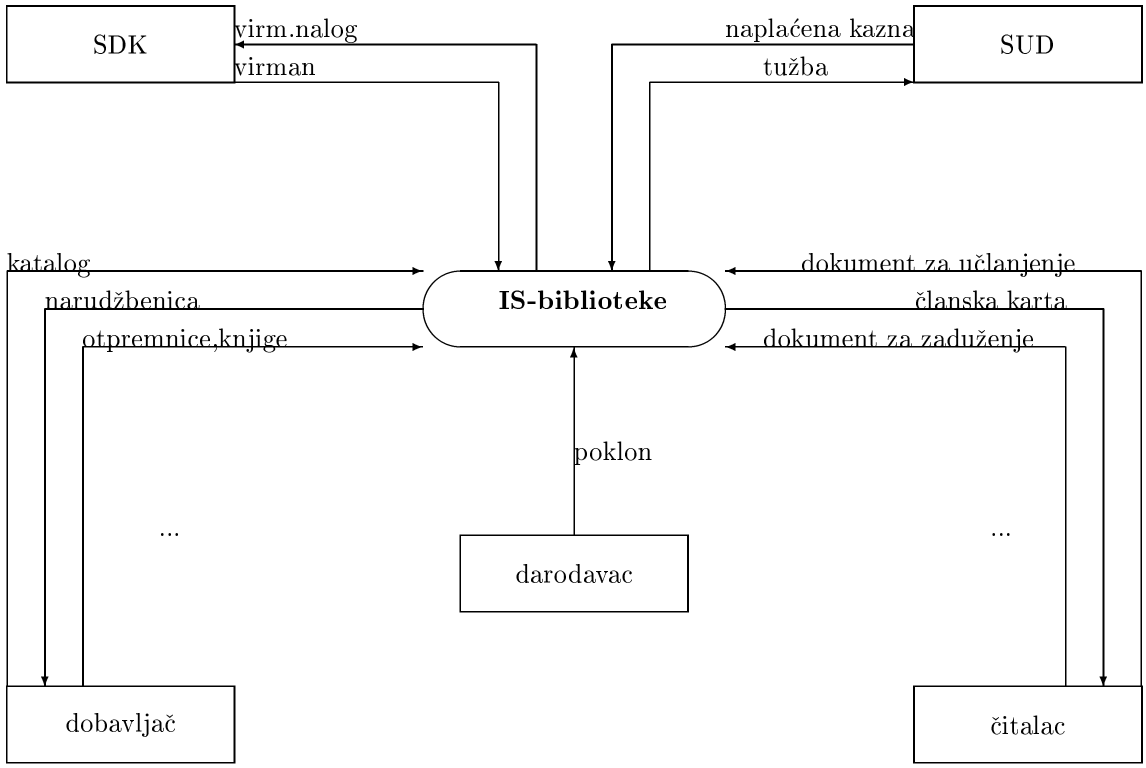
Većina sistema, zbog svoje složenosti, zahteva hijerarhijski opis tj. podelu na podsisteme na više nivoa apstrakcije. DTP se dekomponuju tako da se funkcija na jednom nivou predstavlja DTP-om na sledećem nivou. Pritom se polazi od tzv. dijagrama konteksta -koji postavlja granice sistema i prikazuje samo ulaze i izlaze i uvodi osnovna imena. Srednji nivoi prikazuju podelu sistema na smislene delove koji su još uvek prekrupni za detaljnu analizu. Primitivne funkcije na najnižem nivou su sasvim razumljive i ne moraju se dalje dekomponovati.

Tokovi podataka funkcije na dijagramu moraju odgovarati ukupnim ulazima i izlazima na dijagramu kojim se ta funkcija predstavlja na sledećem nivou apstrakcije. Logička I veza izmedju tokova označava se sa *, ekskluzivno ILI sa + a elipsom izmedju obuhvaćenih tokova predstavlja se ILI veza izmedju tih tokova.

Sa crtanjem DTP obično se polazi od izlaza ka ulazu (prema specifikaciji), ali je u upotrebi i crtanje od celine sistema (crne kutije).

Skup DTP koji opisuju postojeće procese, tokove i skladišta podataka je skup fizičkih DTP. Oni opisuju kompletan postojeći sistem, kako njegove suštinske poslove i funkcije, tako i one koji su posledica postojeće organizacije rada. Zadatak sistem analitičara je da iz skupa fizičkih DTP izdvoji one elemente koji odražavaju samo suštinske poslove sistema. Skup takvih DTP je skup logičkih DTP.

Jedan primer DTP je dijagram konteksta bibliotečkog informacionog sistema:



slika 6. DTP1 - dijagram konteksta 1

2 Sredstva za predstavljanje logike procesa - tabele odlučivanja

Tabele odlučivanja su jezik tj. sredstvo za definisanje, analizu i rešavanje problema upravljanja. Tabelom se prikazuju veze izmedju uslova nekog problema i akcija koje treba preduzeti u svakom slučaju definisanom stanjem uslova. Ideja potiče od tablica istinitosnih vrednosti iz logike. Tabele odlučivanja se mogu primeniti za opis problema nezavisno od oblasti; ovde će se koristiti za opis logike procesa informacionih sistema.

Format tabele odlučivanja je

Naziv tabele		Identifikator pravila			
		P1	P2	...	Pk
C1 . . . Cn	AND (uslovi)	ORR (ostvareni uslovi)			
A1 . . . Am	AND (akcije)	(izabrane akcije)			

Primer tabele odlučivanja:

		1	2	3	4	5	6	7	8
C1:	mesečna plata manja od 100 dinara	Y	Y	Y	Y	N	N	N	N
C2:	redovan na poslu	Y	Y	N	N	Y	Y	N	N
C3:	efikasan	Y	N	Y	N	Y	N	Y	N
A1:	dati normalnu povišicu	X	X	X	X	X	X	X	X
A2:	dati dodatnu povišicu	X	X	X					

Ova tabela se može i skraćeno zapisati:

C1:	...	Y	Y	Y	N
C2:	...	Y	N	N	-
C3:	...	-	Y	N	-
A1:	...	X	X	X	X
A2:	...	X	X		

ili, upotrebom ELSE,

				ELSE
C1:	...	Y	Y	-
C2:	...	Y	N	-
C3:	...	-	Y	-
A1:	...	X	X	X
A2:	...	X	X	

Za razliku od prethodnih (prostih) postoje i tabele odlučivanja sa proširenim sadržajem, i mešovite tabele odlučivanja, u kojima se uslovi tj. akcije proširuju i na deo tabele sa realizovanim uslovima tj. izabranim akcijama. Na primer,

Tip transakcije	Prijem	Nabavka	Višak	Manjak
Dodati na zalihe	X		X	
Oduzeti od zaliha				X
...			...	

Medju akcijama u tabeli odlučivanja može se naći i akcija prelaza na drugu tabelu, pa i na istu tabelu (rekurzivno).

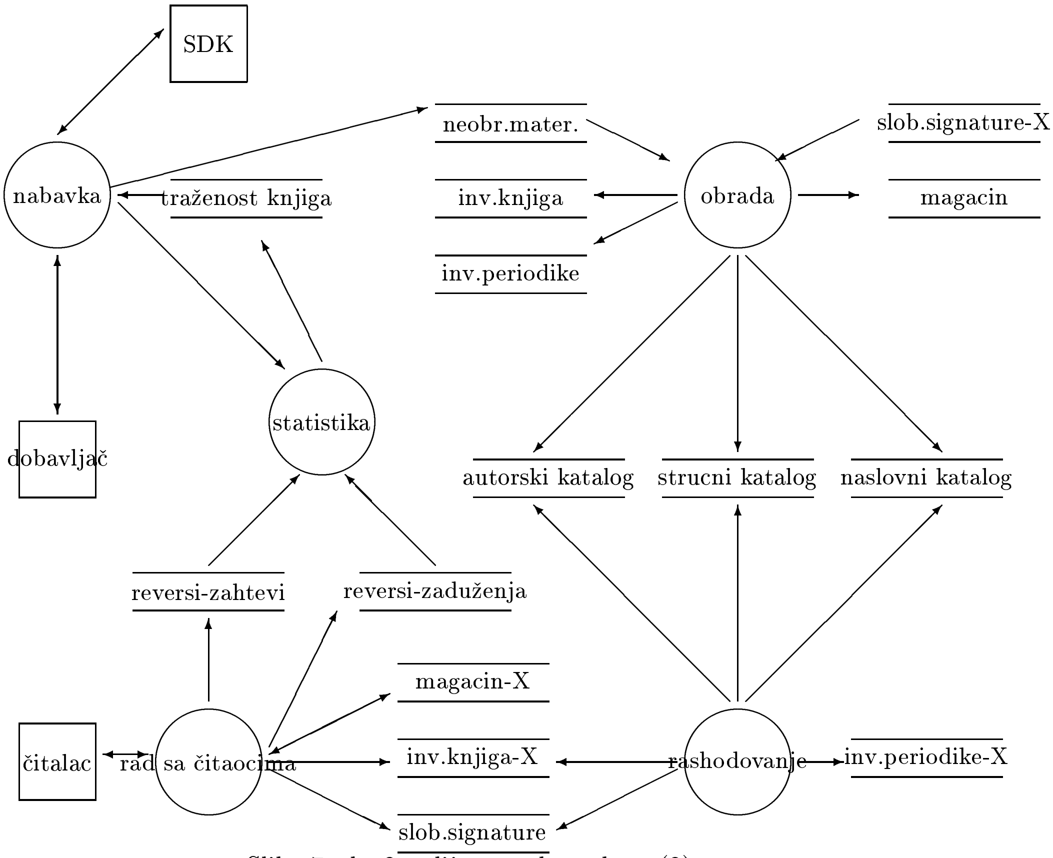
U formiranju tabele odlučivanja postoje dva pristupa. Jedan je kada je problem sasvim definisan, pa se ispišu svi uslovi i akcije, a zatim, kombinovanjem uslova i akcija kompletiraju se pravila. Drugi je pristup formiranju u fazi razmišljanja o problemu, kada se svaki iskaz o sistemu beleži kao jedno pravilo, a zatim se prestrukturiranjem i sintezom iskaza dobije tabela.

Prevodjenje tabela odlučivanja u dijagram toka programa ili programski jezik vrši se jednostavno i pravolinijski, po pravilima, pri čemu se mogu primeniti razne metode optimizacije.

3 Rečnik podataka

Ako se za rečnik podataka koristi specijalizovani softver, omogućena je provera konzistentnosti definicija. Rečnik podataka se zadaje sadržajem skupa formulara, koji predstavlja dokumentaciju - opis IS, tj. bazu podataka o sistemu koji se projektuje i samom projektu. Na primer, za opis funkcije se može koristiti formular sa poljima: naziv, referenca (broj sa dijagrama), kraći naziv, opis, naziv nadredjene funkcije, tokovi podataka na ulazu, tokovi podataka na izlazu, evidencije koje se koriste, standardni procesi koje koristi, itd. Opis toka može se zadati formularom sa poljima naziv, šifra, opis, ka, od, struktura podataka,..., opis skladišta može da uklјči opis evidencije, opis podatka, opis grupe podataka, itd. Opis podatka, funkcije, toka (njegova definicija) ne sme biti cirkularna (posredno ili neposredno). Rečnik sadrži i pojmove koji se ne definišu (bilo zbog svoje krajnje opštosti, npr. "supstanca", bilo zbog svoje krajnje specifičnosti, npr. "dekadna cifra". Za rečnik podataka obično se gradi procesor koji može da generiše hijerarhijski listing termina (... u relaciji je ... je ...). Procesor rečnika podataka, u opštem slučaju, može da:

1. prihvata definicije sa ulaza, o tokovima podataka, elementima podataka, skladištima i procesima;
2. koristi odgovarajuće konvencije u definiciji, za oznaku sekvence, disjunkcije, ekskluzivne disjunkcije, iteracije, selekcije;
3. omogući lako ažuriranje definicija;
4. vrši osnovnu proveru konzistentnosti informacija (dupliranje imena, cirkularnost definicije, sintaksnu ispravnost definicije u skladu sa usvojenom konvencijom;
5. proizvede kao izlaz listing definicija u azbučnom redu;
6. proveri medjuzavisnosti definicija;
7. crta i održava DTP.



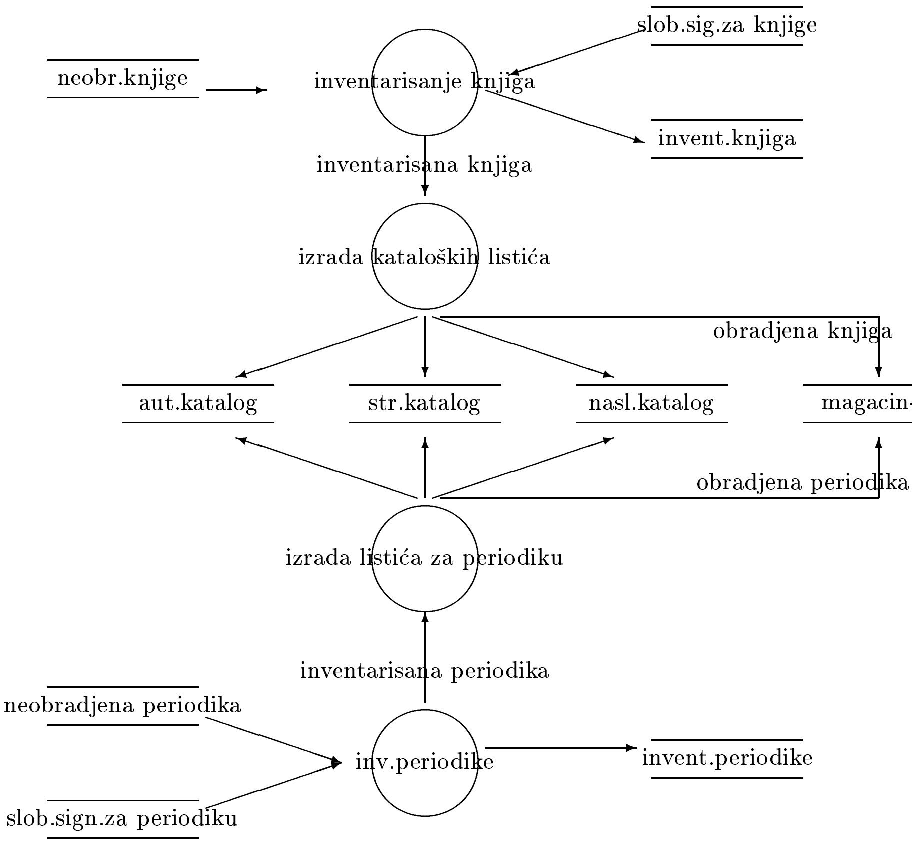
Slika 7: dtp2 – dijagram konteksta (2)

4 Primena SSA

Pošto se u SSA došlo do detaljnih dijagrama, opisa logike procesa i rečnika podataka, u rečniku podataka detaljno su opisani procesi, skladišta i tokovi podataka. Do sadržaja i strukture skladišta (koji su bitni za projektovanje baze podataka) dolazi se snimanjem originalnih dokumenata, obrazaca i neformalnih zapisa koji se u postojećem sistemu koriste. Analizom postojećeg stanja može se doći do predloga za strukturu procesa i skladišta podataka novog informacionog sistema. Kao rezultat ove aktivnosti dobija se novi informacioni sistem zadat novim skupom DTP.

Proces izvodjenja SSA prikazaćemo na primeru dijagrama toka podataka bibliotečkog informacionog sistema (prema diplomskom radu Zorice Pavlović, "Projektovanje baze podataka bibliotečkog fonda", Beograd 1993). Polazni dijagram (slika 6) razradjuje se u dijagram konteksta na sledećem nivou (slika 7, dtp2 - dijagram konteksta (2)).

Jedna od njegovih funkcija (procesa) - OBRADA BIBLIOTEČKOG MATER-



Slika 8: dtp3 – proces OBRADA (1)

IJALA, prikazana je dijagramom dtp3 - proces OBRADA (1) (slika 8), koji karakteriše istovetnost ulaza i izlaza ove funkcije sa celokupnošću njenih ulaza i izlaza sa prethodnog dijagrama konteksta (dtp2 (2)).

Tako se dolazi do primitivnih procesa svakog procesa. Na primer, za proces OBRADA (bibliotečkog materijala), primitivni procesi (koji se dalje ne dekomponuju i ne prikazuju novim dijagramima) su procesi

- INVENTARISANJE KNJIGA
- IZRADA KATALOŠKIH LISTIĆA (za autorski, naslovni, stručni katalog)
- INVENTARISANJE PERIODIKE
- IZRADA KATALOŠKIH LISTIĆA ZA PERIODIKU.

Sledeći korak u SSA je opis skladišta podataka. Analiza originalnih obrazaca, dokumenata i neformalnih zapisa rezultuje identifikovanjem sledećeg sadržaja i strukture skladišta:

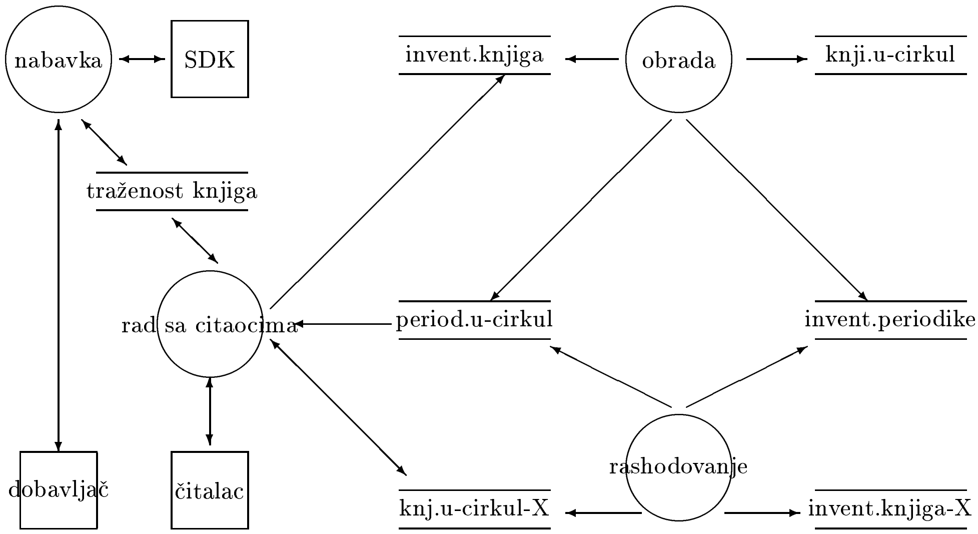
- KATALOG-DOBAVLJAČA: <naziv-dobavljača, adresa-dobavljača, {br.tel.}, {naslov, {autor}, izdanje, god.izdanja, izdavač, cena}>
- ...
- INVENTAR-KNJIGA: <inv-br, datum-inv, signatura, naslov, {autor}, tom, izdanje, god.izdanja, naziv-izdavača, mesto-izdavanja, veličina, vrsta- poveza, {UDK-signatura}>
- AUTORSKI-KATALOG: {[KATALOŠKI LISTIĆ ZA KNJIGU-AUTORA — KATALOŠKI LISTIĆ ZA PERIODIKU-AUTORA]}
- KATALOŠKI-LISTIĆ-ZA-KNJIGU-AUTORA: <autor, naslov, {ime-autora, vrsta- autora}, {signatura}, {inv.broj}, mesto-izdavanja, naziv-izdavača, god.izdanja, izdanje, br.strana, veličina, {UDK-signatura}, pismo, {prevodilac}>

Analizom predstavljenog sistema, dobija se predlog novog, računarski zasnovanog informacionog sistema. Predstavljaju se samo automatizovani procesi i računarska skladišta podataka (mada su neophodna i fizička). Najveća korist od ovog postupka je brzina pretraživanja, bez dupliranja informacija (npr. u katalozima). Tako se sve informacije o knjizi smeštaju u skladište KNJIGA-U-CIRKULACIJI, pri čemu katalozi postaju nepotrebni. Vrši se i izmena nekih postupaka, npr. kod zaduživanja, gde promena redosleda operacija provere i izdavanja reversa čini revers-zahtev nepoterbnim. Dodavanje dva atributa skladištu KNJIGA-U-CIRKULACIJI - broj-zaduženja i borj-traženja čini proces STATISTIKA nepotrebnim. Skladišta NEOBRADJENA-KNJIGA i NEOBRADJENA-PERIODIKA ostaju fizička (neračunarska) skladišta i ne predstavljaju se novim DTP.

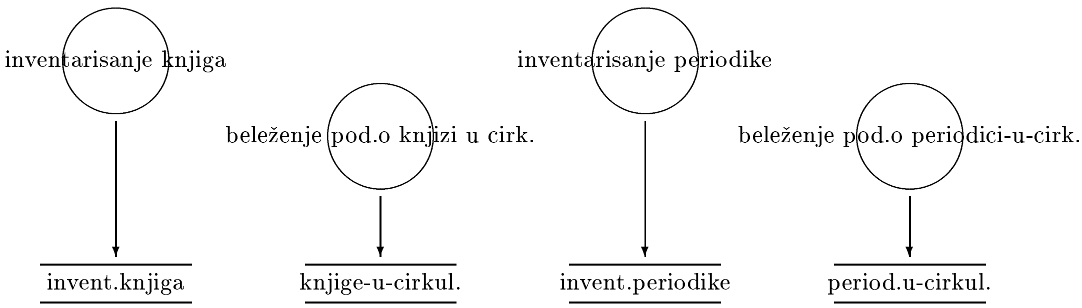
Ovakvom analizom dobija se novi skup dijagrama toka podataka, npr. dtp1' - dijagram konteksta (slika 9), i dtp2' - novi dijagram procesa OBRADA (bibliotečkog materijala) - slika 10.

5 Logičko projektovanje baze podataka

Opis strukture informacionog sistema (obično) sugeriše uvodjenje baze podataka za čuvanje podataka opisanih skladištima podataka. Nakon izbora modela podataka, treba pristupiti logičkom projektovanju baze. Opis skladišta podataka daje sve informacije koje baza podataka treba da sadrži. Medjutim, skladišta su složeni



Slika 9: dtp1' – novi dijagram konteksta (1)



Slika 10. dtp2' – novi dijagram procesa OBRADA (1)

entiteti, i njihovo neposredno predstavljanje relacijama u relacionom modelu dovelo bi do nenormalizovanog modela.

Jedan mogući postupak za logičko projektovanje baze na osnovu skladišta podataka je primena neke verzije modela entiteta i odnosa (npr. PMEO - proširenog modela entiteta i odnosa) za dobijanje skupa i strukture dobro identifikovanih entiteta i njihovih odnosa, koji se zatim precizno definisanim postupkom mogu jednoznačno preslikati u koncepte relacionog modela (relacije, primarne, strane ključeve). Proces "izdvajanja" entiteta i njihovih odnosa iz skladišta podataka je iskustveni proces, vezan za veštinu i praksu, kao i za niz semantičkih informacija iz realnog sveta. Na primer, proces

OBRADA (bibliotečkog materijala)

koristi sledeće informacije: iz skladišta

INVENTAR KNJIGA

izdvaja se entitet

KNJIGA (inventarisana knjiga).

Iz redosleda procedura koje se primenjuju na novu knjigu, inventarisanje pa kataloška obrada, sledi da je entitet

KNJIGA-U-CIRKULACIJI (dobijen posle kataloške obrade)

specijalizacija entiteta

KNJIGA.

Takodje, podaci o knjizi koja se rashoduje brišu se iz skladišta

KNJIGE-U-CIRKULACIJI,

ali se red u knjizi inventara ne briše, tj.

kardinalnost specijalizacije KNJIGA -> KNJIGA-U-CIRKULACIJI je (0,1).

Za proces Nabavka značajno je, na primer, da se otpremnice numerišu u okviru jednog dobavljača, tj.

otpremnica je slabi entitet koji zavisi od samostalnog entiteta **dobavljač**.

Da bi se sistematizovao proces projektovanja, prvo se grade modeli entiteta i odnosa za procese iz DTP na najnižem nivou detaljnosti, a zatim se ti modeli integrišu u jedan model. Integracija podmodela ide istim pravcem ali suprotnim smerom od onog kojim je išla dekompozicija DTP-a u SSA. Integracija podmodela podrazumeva prepoznavanje "sličnih" entiteta (npr. entitet "knjiga u cirkulaciji" u podmodelima OBRADA i RAD-SA-ČITAOCIMA). Takve entitete treba spojiti u jedan, eventualnim neutralisanjem njihovih razlika, npr. uvodjenjem novih atributa. Primer PMEO za proces OBRADA (bibliotečkog materijala) prikazan je na slici 11.

Relaciona baza podataka treba da obezbedi minimalnost ponavljanja podataka i mogućnost regenerisanja skladišta podataka. Stoga su pravila prevodjenja sa modela entiteta i odnosa na relacioni model sledeća:

- regularni entitet se prevodi u relaciju sa svim njegovim atributima
- slabi entitet se prevodi u relaciju koja sadrži sve njegove atribute i primarni ključ entiteta za koji je vezan
- agregirani entitet se prevodi u relaciju koja sadrži sve njegove atribute i primarne ključeve komponenata
- podtip nekog generičkog tipa entiteta se prevodi u relaciju koja sadrži njegove specifične atribute i primarni ključ odgovarajućeg generičkog entiteta
- odnos se prevodi u relaciju koja sadrži primarne ključeve entiteta koji grade odnos.
- Ako se jedan tip entiteta preslikava na drugi preslikavanjem kardinalnosti (1,1), onda mu se kao atribut dodeljuje primarni ključ drugog entiteta. Odnos izmedju njih ne treba prevoditi u posebnu relaciju.

Neke od (nekoliko desetina) relacija koje se dobiju ovim postupkom transformacije PMEO u relacioni model su:

- RED-INV-KNJIGE(inv.br.*, datum-inv., sign.br., tom, način-nabavke, šif.knjige)
- KNJIGA(šif.knjige*, naslov, izdanje, god.izdanja, veličina, vrsta- poveza, vrsta, šif.izdavača)
- KNJIGA-U-CIRK(šif.knjige*, br.strana, pismo)
- AUTOR(šif.aut*, ime, vrsta)
- KNJ-AUT(šif.knjige, šif.aut), itd.

6 Strukturno projektovanje informacionog sistema

(Structured analysis and system specification, Tom DeMarco, Prentice Hall, 1979.)

Ciljevi faze projektovanja informacionog sistema su, globalno, trojaki:

1. smanjiti troškove održavanja informacionog sistema; ovde se misli na modifikacije i ispravke programa posle isporuke sistema kao gotovog proizvoda; procenjuje se da se do 60% sredstava koja se potrše na razvoj IS utroše upravo u ovoj fazi;
2. smanjiti troškove testiranja IS; ovde se misli na modifikacije i ispravke u programima pre isporuke sistema; procenjuje se da se više od 20% sredstava utroši u ovoj fazi;
3. proizvesti dokumentaciju IS kao prateći proizvod faze projektovanja.

Na smanjenje troškova testiranja i održavanja sistema utiču sledeći faktori:

1. mali moduli;
2. nezavisni moduli;
3. moduli koji se ponašaju kao crne kutije: imaju ulaz, izlaz i pozive drugih modula, što povećava čitljivost programa;
4. konceptualno modeliranje – sistem treba da bude modeliran na prirodi problema koji treba da se reši;
5. izoliranje detalja od koncepcije sistema u projektu i implementaciji.

Prethodni faktori, mada utiču na povećanje fleksibilnosti sistema, obično, u izvesnoj meri, smanjuju efikasnost sistema (povećavaju potrošnju vremena i prostora). Procenjuje se da metoda strukturnog projektovanja proizvodi do 10% smanjenja efikasnosti sistema. Medjutim, pri sadašnjem odnosu cene memorije, procesora i ljudskog rada, efikasnost sistema je daleko manje važna od lakoće testiranja i održavanja.

Strukturno projektovanje je pristup projektovanju koji rezultuje malim, nezavisnim modulima sa ponašanjem crne kutije, uredjenim u hijerarhiju koja predstavlja konceptualni model poslovne oblasti; to je strategija projektovanja pouzdanih sistema koje je moguće održavati.

Definicije:

1. Strategija projektovanja je skup metoda za izvodjenje, evaluaciju, poboljšanje i dokumentovanje projekta.
2. Projektovanje odozgo-naniže (top-down) je način projektovanja, ne strategija, koji se sastoji od hijerarhije modula, svaki sa jednom ulaznom i jednom izlaznom tačkom.
3. Strukturno projektovanje je strategija projektovanja na odozgo-naniže-način, kojom se proizvode projekti sistema koji se lako testiraju i održavaju. Metode izvodjenja ove strategije su **analiza transformacije** i **analiza transakcije**, tehnike evaluacije i poboljšanja su **kopčanje**, **kohezija**, **pakovanje** i **heuristike**, dok je sredstvo dokumentovanja – strukturni dijagram.

Informacioni sistem ima proceduralne i hijerarhijske karakteristike. Proceduralne karakteristike definišu redosled aktivnosti (npr. x pre y), dok hijerarhijske karakteristike definišu rang komponenti (modula). Oba skupa karakteristika moraju biti odrednjena; pitanje je kako.

Ako se počne sa odredjivanjem proceduralnih karakteristika, dobiće se projektovanje odozdo-naviše.

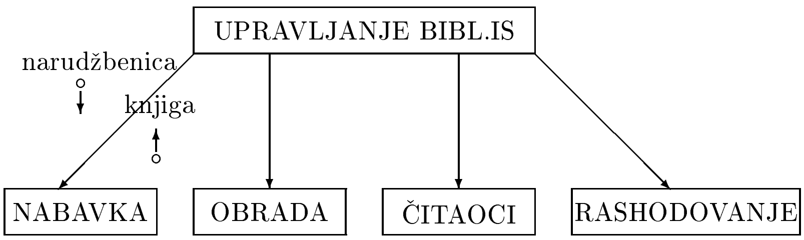
Strukturno projektovanje počinje hijerarhijskim karakteristikama, postavljanjem pitanja u redosledu značaja, npr.

1. – Koja je priroda sistema?
   – Upravlja radom biblioteke.

2. – Koji su najvažniji aspekti tog posla?
   – Nabavka bibliotečkog materijala, obrada, rad sa čitaocima, rashodovanje.

3. ...

Strukturnim dijagramom, prva dva nivoa hijerarhije predstavljena su na slici 12.



Slika 12: Prva dva nivoa hijerarhije strukturnog dijagrama

Sledeća iteracija projektovanja proizvodi sledeći nivo hijerarhije (slika 13).

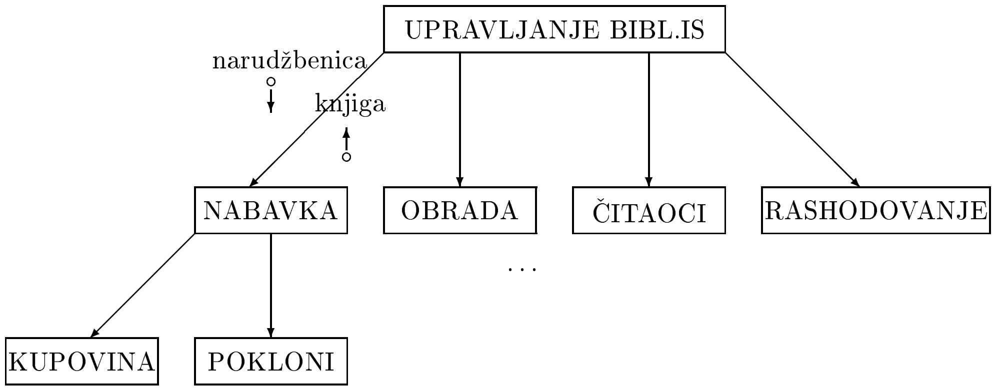
Strukturni dijagram se sastoji od:

1. modula (u pravougaoniku sa imenom); modul je imenovani, ograničeni blok iskaza;

2. veza, predstavljenih vektorima koji spajaju 2 modula; vezza obično znači da jedan modul poziva drugi;

3. kopča: to su elementi podataka koji se kreću od jednog modula do drugog; predstavljaju se kratkim strelicama sa kružnim krajem; strelica je usmerena prema modulu ka kome se kreće element podataka.

Skup veza iz jednog modula pokazuje da postoji bar po jedan poziv unutar tog modula svakom podredjenom modulu, ali se ne zna koliko ih je, u kom su redosledu i koliko se često koriste; ta proceduralna informacija se ne prikazuje na strukturnom dijagramu. Strukturni dijagram prikazuje samo particioniranje sistema i interfejse, slično DTP, sa kojim, pored sličnosti, ima i razlika.



Slika 13: Prva tri nivoa hijerarhije strukturnog dijagrama

6.1 Kopče

Kopče ukazuju na kvalitet projektovanja, mereći medjuzavisnost modula. Kopčanje treba minimizovati, jer se inače izmenama u jednom modulu narušava funkcionalnost drugog.

Nedozvoljene kopče su GOTO iz jednog modula u drugi, kao i obraćanje podacima u drugom modulu iz prvog modula.

Tip podataka koji se kreću medju modulima (tip kopči)može biti izračunljiv (na primer za indeksiranje) ili prekidački (switch) na osnovu kojih se donose odluke. Prvi tip podataka je poželjniji za kopčanje.

Količina podataka kopčanja je obrnuto srazmerana kvalitetu projekta.

Smer nekih kopči je od interesa za kvalitet projekta. Na primer, prekidači usmereni nadole su manje poželjni od onih usmerenih nagore, jer narušavaju integritet modula koji ih prima, upravljajući njima "odozgo", a da moduli kojima se na taj način upravlja nemaju "razumevanja" o tom upravljanju.

6.2 Kohezija

Kohezija (modula) je mera jačine asocijacije elemenata unutar modula. Visoko kohezivni modul je kolekcija iskaza i podataka koje treba tretirati kao celinu zato što su veoma tesno povezani. Svaki pokušaj njihove podele proizveo bi samo povećanje kopči i smanjenje čitljivosti.

Modul ima prihvatljivu koheziju ako izvršava tačni jedan zadatak ili nekoliko zadataka povezanih u[otrbom istih podataka. Modul ima neprihvatljivu koheziju kad izvršava nepovezane zadatke, ili povezane samo slabom racionalizacijom, vremenskom zavisnošću (zadaci se izvršavaju istovremeno) ili redosledom (jedan za drugim). Prihvatljivost se često ogleda u imenu modula (npr. upis u stariju godinu/ažuriranje evidencija; studiranje/glavni program). Ime modula je je ime celog drveta u čijem je korenu taj modul

Kohezija i kopče su u tesnoj vezi. Puno kopči, kao i nanižne prekidačke kopče, ukazuje na slabu koheziju modula.

6.3 Pakovanje

Pakovanje je tehnika koja se odnosi na efikasnost sistema, i predstavlja oblikovanje projekta prema fizičkom okruženju – računaru, operativnom sistemu, jeziku, unutrašnjoj i spoljašnjoj memoriji, vremenskim ograničenjima. Pakovanje se ostavlja za kraj projektovanja, tako da se prvo konstruiše strukturni dijagram nezavisan od okruženja kojim se maksimizuje kohezija i minimizuje kopčanje, da bi se zatim primenilo pakovanje modula u programe i procese, prema fizičkim ograničenjima (npr. veličina sektora na disku, preklapanje, programski jezik, upotreba COMMON zona, itd). Dakle, pakovanje treba primeniti POSLE uspešnog testiranja. Ed Yourdon: "Lakše je učiniti sistem koji radi efikasnim, nego učiniti da efikasni sistem proradi".

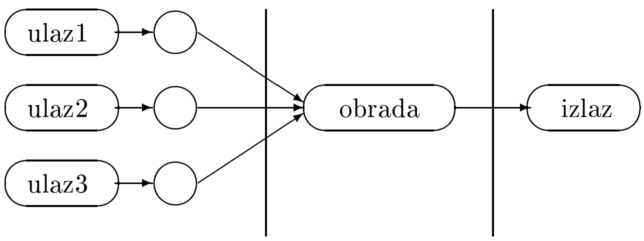
Problemi sa efkasnošću su po pravilu drugačiji od onih koje projektant može unapred da predvidi.

6.4 Strukturni dijagrami i DTP

Izgradnja dijagrama toka podataka i strukturnih dijagrama vodjena je istom filozofijom hijerarhijske dekompozicije. Razlike izmedju ova dva sredstva razlike izmedju namere i metode, tj. u tome što DTP izražava zahtev – ŠTA treba ostvariti, dok strukturni dijagram izražava projekat – KAKO ispuniti zahtev.

Strukturni dijagram se iz DTP izvodi tehnikama analize transformacije i analize transakcije. Ove tehnike ukazuju na to kako početi proces projektovanja.

Analiza transformacije. Ova tehnika se primenjuje na aplikacije čija je struktura predstavljena DTP-om linijska: ulazni tok, centralna obrada, izlazni tok. Takav DTP je prikazan na slici 14.



Slika 14: DTP za analizu transformacije

Primer transformacije DTP u strukturni dijagram prikazan je na slici 15.

Projekat mora da sadrži elemente koje DTP nije razmatrao: tok upravljanja, petlje i odluke, pakovanje, unos prekidačkih kopči. Zato se na dobijene module primenjuje analiza kopčanja i kohezije, koji mogu biti narušeni ovim razmatranjima.

Analiza transformacije, ilustrovana slikom 15, sprovodi se prema sledećim pravilima:

1. Vrh strukturnog dijagrama odgovara centralnom delu obrade iz DTP (koji se ne bavi ni ulazom ni izlazom).
2. Postoji jedan modul (na sledećem nivou) za svaki ulazni strim, jedan za svaki izlazni strim, i jedan za centralnu obradu.
3. Kopča iz modula za ulazni strim (iz prethodnog koraka) odgovara toku podataka u centralnu obradu sa DTP. Kopča prema modulu za izlazni strim odgovara toku podataka iz centralne obrade sa DTP.
4. Kopče ka i od modula centralne obrade su zbir svih kopči primljenih od ulaznih mosula i svih kopči koje treba da se proslede izlaznim modulima.
5. Svakom ulaznom procesu na DTP odgovara par modula – "uzeti" i "preraditi" na strukturnom dijagramu. Svakom izlaznom procesu na DTP odgovara par modula – "preraditi" i "izdati".
6. Kopče podstruktura odgovaraju tokovima podataka u i iz pripadnog procesa.
7. Moduli nižeg nivoa ispod modula obrade odgovaraju (1–1) podredjenim procesima na DTP-skupu.
8. Kopče ka i iz podrdjenih modula izvode se iz tokova podataka u i iz pridruženih procesa.

Analiza transakcije Analiza transakcije primenjuje se na aplikacije koje imaju centar transakcije koja se sastoji od procesa koji se mogu izvršavati paralelno. Odgovarajući DTP je prikazan na slici 16.

Primer transformacije DTP u strukturni dijagram, primenom analize transakcije, prikazan je na slici 17.

Analiza transformacija, ilustrovana slikom 15, sprovodi se prema pravilima za analizu transformacije (1–8), i sledećim pravilima:

9. Modul koji odgovara procesu centra transakcije je i sam centar transakcije, tj. upravlja po jednim podredjenim modulom za svaki paralelni proces na DTP.
10. Svakom podredjenom modulu pod centrom transakcije pridružen je jedan od paralelnih procesa. Ti podredejni moduli čine "nivo transakcije".
11. Pod svakim modulom nivoa transakcije nalazi se skup "modula akcija", po jedan za svaku akciju koja ulazi u sastav procesa (akcije su funkcije čijim se kombinovanjem opisuje proces). Ispod modula akcija je nivo detaljnih modula.

12. Kopčanje na nivou transakcije odgovara toku podataka ka i od skupa paralelnih procesa.

Najčešće aplikacije su kombinovane i uključuju obe tehnike izvodjenja – analizu transformacije i analizu transakcije. Tada se prva primenjuje analiza transformacije. Primer dijagrama toka podataka takve aplikacije prikazan je na slici 18.

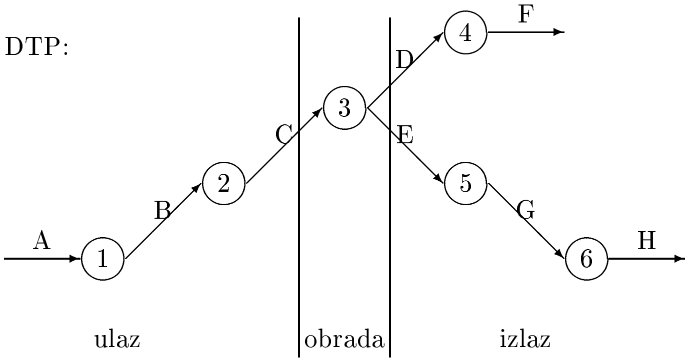
6.5 Implementacija strukturnog projektovanja

Implementaciju projekta moguće je sprovesti na uobičajeni način – programiranjem i testiranjem modula, podsistema, sistema i integracije. Ali, najvažniji interfejsi će se testirati tek na kraju, a rani stadijumi implementacije biće vezani sa detaljima.

Alternativa ovakvom načinu implementacije je strukturna: implementacija se sprovodi odozgo-naniže. Programira se i testira vršni modul, zatim moduli drugog nivoa, koji se zatim spajaju sa vršnim i testira vršna struktura. Moduli trećeg nivoa u tom trenutku su samo okviri (sadrže, npr. samo return sa fiksnom vrednošću).

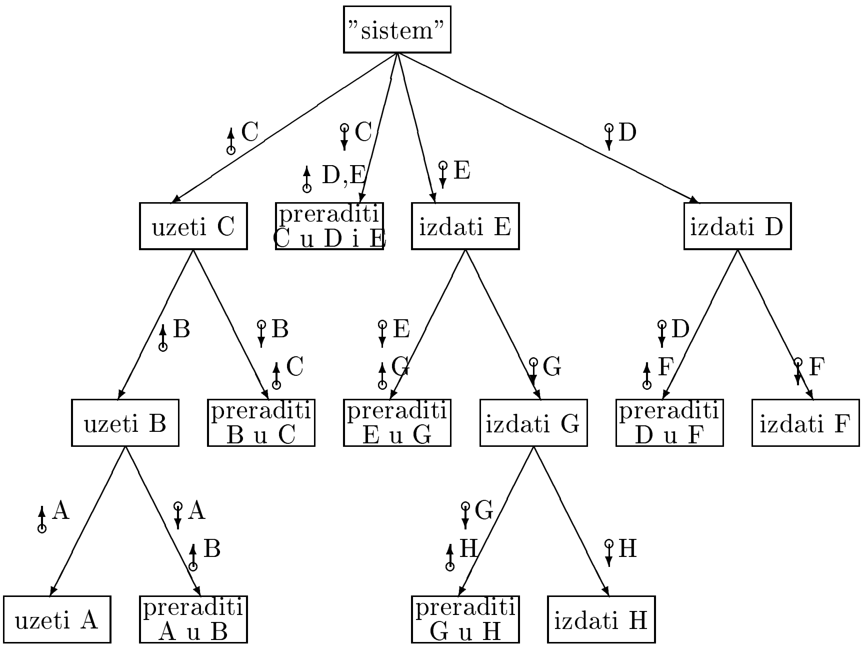
Prednosti strukturne implementaciju su brojne:

- kritični interfejsi se prvi testiraju;
- rane verzije, iako nekompletne, liče na realni sistem dovoljno da pruže korisnu vidljivost;
- nema skupih i komplikovanih testiranja;
- ostvaruje se korisno preklapanje programiranja, testiranja i integracije.

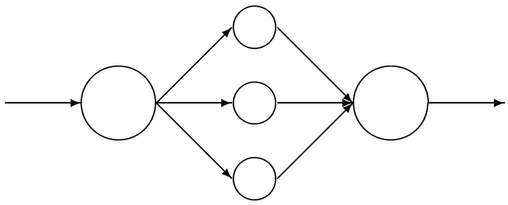


Proces 3 može biti proizvoljne složenosti koja se reflektuje na strukturni dijagram

STRUKTURNI DIJAGRAM:

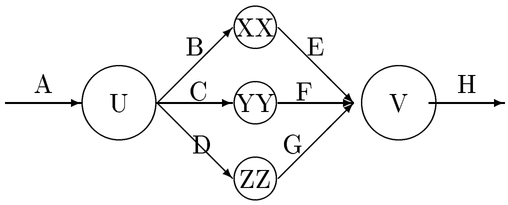


Slika 15: Transformacija DTP u sintaksni dijagram

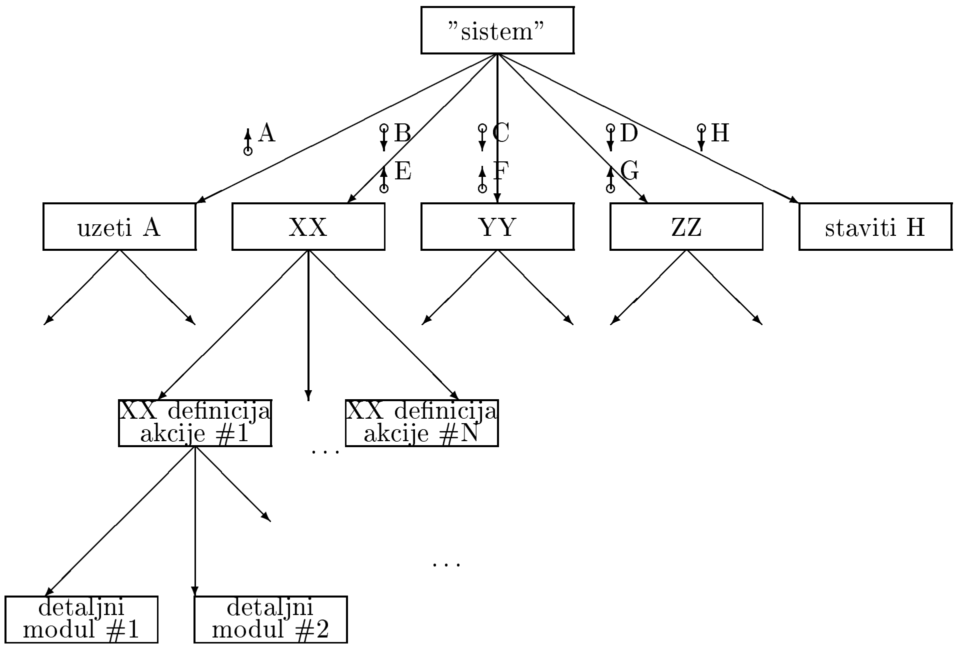


Slika 16: DTP za analizu transakcije

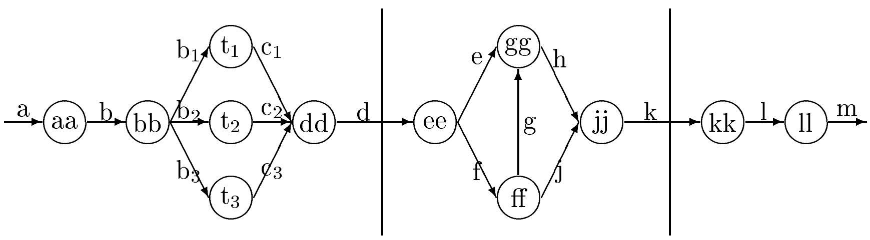
DTP:



STRUKTURNI DIJAGRAM:



Slika 17: Izvodjenje strukturnog dijagrama iz DTP analizom transakcije



Slika 18: DTP kombinovane aplikacije