

I. FELJA, A. PAVIĆ, E. RIFATI, Z. SKOČIR, M. ZORIĆ

ELEKTROTEHNIČKI FAKULTET
ZAGREB, Unska 3

REFERAT

ORGANIZACIJA I OBRADA PODATAKA U TELEKOMUNIKACIJSKOJ MREŽI

ORGANISATION AND DATA PROCESSING IN THE TELECOMMUNICATION NETWORK

SADRŽAJ - U radu su izloženi različiti aspekti obrade podataka u telekomunikacijskoj mreži obzirom na kvalitetu programske podrške te adekvatnu organizaciju baze podataka.

ABSTRACT - In this paper different aspects of data processing in telecommunication network in terms of software support quality and adequate data base organisation are presented.

1. UVOD

Razvoj suvremenih telekomunikacija karakterizira sve veća primjena elektroničke tehnologije, uvođenje procesorskog upravljanja s upisanim programima na svim nivoima telekomunikacijske mreže i proširenje usluga na obradu podataka. Time se omogućava izgradnja jedinstvenih integriranih sistema u kojima su funkcionalne mreže raznih namjena sekundarne mreže primarnog telekomunikacijskog sistema. Ostvaruje se mogućnost da mreže računala za potrebe različitih informacijskih sistema budu sekundarne mreže primarne telekomunikacijske mreže. Na taj će se način definitivno srušiti klasične barijere između računala i telekomunikacija. Funkcija telekomunikacija u takvoj mreži jest obrada, usmjeravanje i prijenos informacija; dakle ostvarivanje mogućnosti povezivanja centara (računala) u kojima se vrši obrada velikih količina informacija. Distribucija resursa (programa, podataka, računarske opreme itd.) između nekoliko sistema argumentirana je ekonomskim i organizacijskim razlozima, kao i većom pouzdanošću sistema.

Postoji nekoliko nivoa obrade podataka u integriranoj komunikacijskoj mreži. Prvi nivo predstavlja obrada podataka za upravljanje komutacijskih centara i operativno vođenje mreže. Drugi nivo predstavlja obrada podataka u informacijskom sistemu za potrebe telekomunikacijske mreže [2]. Treći nivo obrade podataka predstavlja obrada u mrežama računala za potrebe različitih informacijskih sistema.

Kompleksnost integrirane telekomunikacijske mreže, kako na nivou vođenja i upravljanja, tako i na nivou pružanja novih kvalitetnijih usluga mnogobrojnim korisnicima, ukazuje na potrebu za odgovarajućim pristupom organizaciji obrade podataka na svim nivoima. Nužan preduvjet efikasne organizacije obrade podataka je razvoj pouzdane programske podrške i adekvatna organizacija baze podataka prilagođena procesu i funkcijama sistema.

U ovom radu, na temelju višegodišnjeg istraživanja na ovom području [5] [6], istaknut će se postupci izgradnje efikasne programske podrške kao i metode kojima će se ona pratiti i povećavati, te prikazati problemi vezani uz organizaciju baza podataka.

2. EFIKASNOST PROGRAMSKE PODRŠKE

Programska podrška specifičan je proizvod, karakteriziran djelovanjem na nekom skupu (ulaznih) podataka u cilju njegove transformacije u neki drugi skup (izlaznih) podataka. Kvaliteta programske podrške određena je stoga prvenstveno efikasnošću njezinog djelovanja. Osnovni aspekti efikasnosti djelovanja su učinak (efektivnost) programske podrške, te njezina prilagodjenost čovjeku i računalu (djelotvornost) sa kojima u interakciji djeluje.

Zanimljiv je raspored troškova u životnom ciklusu programske podrške [7]. Troškovi testiranja dosežu i do 50% troškova proizvodnje, a što je posebno zanimljivo, troškovi korištenja i održavanja prelaze troškove proizvodnje.

Ovo nas upućuje na neispravnosti programske podrške kao glavni izvor neefikasnosti djelovanja. Dva su pristupa aktualna u rješavanju ovog problema, a to su tzv. analitički i konstruktivni pristup [8].

Analitički pristup je usmjeren na razvoj i usavršavanje metoda za otkrivanje i korekciju neispravnosti gotovih programa, dok konstruktivni pristup pridaje više pažnje metodama koje već u toku dizajna i implementacije programa osiguravaju njegovu ispravnost. Rješenje problema neispravnosti programa je svakako u odgovarajućoj kombinaciji ova dva pristupe.

Uz neispravnosti ugrađene u programe, drugi značajan izvor neefikasnosti u fazi korištenja, predstavljaju zahtjevi za izmjenama programske podrške, koji proizlaze iz naglog razvoja i proširenja područja primjene. Rješenje ovog problema jest u pridavanju odgovarajuće pažnje parametrima programa koji povećavaju njegovu fleksibilnost, a to su strukturiranost programa, te njegova pogodnost za proširenja (zalihost u funkcijama i memorijskim zahtjevima).

Veliki dio programske podrške u okviru telekomunikacijske mreže radi u interakciji sa čovjekom, zbog čega posebnu pažnju treba pridati karakteristikama prilagodjenosti čovjeku a to su u prvom redu razumljivost te komunikativnost programske podrške [9].

I na kraju, programska podrška mora biti optimalno uskladjena sa karakteristikama sklopovske osnove mreže (potrošnja vremena i memorije), u cilju zadovoljavanja strogih zahtjeva rada u realnom vremenu.

3. NEKI ASPEKTI PROIZVODNJE PROGRAMSKE PODRŠKE

U vrlo razvijena tehnologija računala osnova je za novi i vrlo dinamičan razvoj u području organizacije proizvodnje programske podrške i obrade podataka. Očito je da se s jedne strane broj korisnika naglo povećava i da se među korisnike ubrajaju praktički sve kategorije kadrova po strukama i stupnju obrazovanja, a koji računala i mrežu upotrebljavaju u svom poslu od jednostavnih matematičkih aplikacija do projektiranja, masovne obrade podataka i upravljanja procesima. Programski sistem je u eksploataciji izložen savladavanju različitih stupnjeva problema, različitim zahtjevima i dinamičnim prilagodjenjima trenutnoj aplikaciji korisnika. S druge strane tehnologija pruža sve veće i veće mogućnosti. Nove koncepcije u izradi računala, velike brzine rada, nove vrste memorija, višeprocorski sistemi, mreže računala, multiprogramiranje i distribuirani sistemi rezultati su takvog napretka. Proizvodnja programske podrške mora se prilagoditi i toj strani sistema na kojem se osniva i izvršava svoje funkcije i tako pružiti korisniku mogućnost bolje eksploatacije i kroz to veću ekonomičnost u korištenju relativno skupih tehnoloških dostignuća. Proces izgradnje programske podrške ima dakako i svoje organizacione i ekonomske zakonitosti koje obuhvaćaju postupke u organizaciji rada da bi se postigli željeni efekti, efikasnost, ekonomičnost i kvaliteta.

Pretakanje određenog problema ili zadatka u program kroz filter pravila strukturiranja programa i podataka omogućuje nam sačuvanje logike strukture i postupnost u rješavanju. Proces kojim se vrši dekompozicija problema u niz al-

goritama različit je od primjene do primjene, ali u osnovi može se svaki problem formalno opisati i funkcionalno razraditi i onda dekomponirati. Zatim slijedi formiranje algoritama te razrada strukture i opis podataka. Takav proces izgradnje programske podrške daje nam sada mogućnost da kroz formalni opis algoritma provjerimo njihovu ispravnost, logičnost, funkcionalnost. U proces kreiranja uključujemo sva znanja iz područja koje obuhvaća problem, kao i metode rješavanja karakteristične za proizvodnju programske podrške. Algoritme i module treba grupirati i tipizirati, standardizirati njihove funkcije i način baratanja podacima ili ih proširiti funkcionalno zbog određenih zahtjeva kao što su testiranje, međusobna neovisnost i elastičnost cijele programske podrške na moguće promjene.

Kao i kod dekompozicije početnih zahtjeva i problematike i u obrnutom procesu sastavljanja programa iz mozaika algoritama modula može se ovisno o specifičnoj primjeni poći različitim putevima. Klasično sklapanje velikih programskih paketa s krutim unutrašnjim strukturama jedan je od njih, ali je očito da novi zahtjevi i nova tehnologija otvara i nove mogućnosti. Programska podrška s dinamičnom strukturom bila bi prilagodjena korisniku s jedne i sistemu s druge strane. Preko povoljno odabranog programskog ulaznog jezika visokog nivoa kreirala bi se prema potrebi i zahtjevima složene strukture programa na osnovama samoizgradnje. Takve bi strukture imale i određene slobode, unutarnju nedeterminiranost pa i sposobnost učenja i prilagodjenja. Da bi se omogućila ovakva kvaliteta očito je treba predvidjeti i organizirati u cijelom postupku proizvodnje programske podrške kroz strukturiranje i organizaciju podataka, dekompoziciju funkcija, izbor, tipiziranje i proširenje funkcija algoritama. Karakteristika funkcioniranja novih programskih sistema bila bi bolje prilagodjenost i veća otvorenost prema korisniku, njegovim sposobnostima i problemima kada on po raznoj stručnosti ne ulazi u same strukture programiranja, obrade podataka i cjelokupna znanja koja su primjenjena u rješavanju ili prilagodjenju tehničkoj osnovi sistema.

4. GREŠKE U PROGRAMSKOJ PODRŠCI I NJIHOVO OTKRIVANJE

Iskustva pokazuju da i sistemi programske podrške kojima je posvećena najveća pažnja u proizvodnji, ipak sačinjavaju greške. Greške koje se pojavljuju mogu se klasificirati na više načina i mi ćemo se zadovoljiti slijedećom podjelom:

1. nepotpuna ili pogrešna specifikacija
2. namjerno odstupanje od specifikacije
3. nepridržavanje utvrdjenih standarda pri programiranju
4. pogrešan pristup podacima
5. pogrešna logika odlučivanja ili pogrešan slijed
6. pogrešni aritmetički proračuni
7. pogreške u vremenskom tempiranju
8. pogrešno korištenje prekidnih rutina
9. pogrešne vrijednosti konstanti ili podataka
10. neodgovarajuća dokumentacija

Iz navedenog je vidljivo da ne postoji jedan jedini uzrok pojavljivanja grešaka u programskoj podršci, te se može zaključiti da, barem u današnjem trenutku, ne postoji jedan aparat ili način koji bi otkrivao sve tipove grešaka. Pri ocjeni ispravnosti programske podrške koriste se stoga različite metode koje se mogu klasificirati u dvije kategorije - verificiranje i testiranje.

Kod verificiranja ispravnosti razlikuju se dva pristupa - statički i konstruktivni. U statičkom pristupu predpostavlja se da su program i njegova specifikacija poznati. Pristupa se izvodenju matematičkih dokaza kako bi se pokazalo da logičko ponašanje programa odgovara specifikaciji. Konstruktivni pristup naglašava nužnost korektnog razvoja programske podrške. No oba pristupa se oslanjaju na sposobnost programera da uoči neka dovoljno istaknuta nepromjenljiva svojstva programa, bilo da su ona dana unaprijed ili se podrazumijevaju. Razvoj metoda verifikacije programske podrške je od velike važnosti za cjelokupno pod-

ručje definiranja, razvoja, testiranja i održavanja programske podrške, no u ovom trenutku je još nedovoljno razvijeno da bi predstavljalo ekonomično i pouzdano rješenje.

Pri testiranju programa, program P se izvodi na skupu ulaznih vrijednosti X, a izlazne vrijednosti Y se kontroliraju obzirom na očekivane. Testiranje ispravnosti programa podrazumijeva postojanje vanjskog mehanizma M (često čovjek) koji može ocijeniti ispravnost test parova (X, Y). M može uključivati ručne proračune, tabelarne usporedbe, ili usporedbu skupa (X, Y) sa specifikacijama ulaza i izlaza. Moguće je i korištenje test parova (X, Y) u kojima Y nema značenje izlaznog rezultata programa već predstavlja odabrani medjurezultat.

Najveći problem u testiranju programa je kako odabrati skup ulaza kojim će se izvršiti testiranje. Iscrpno testiranje, definirano na taj način da se program testira obzirom na sve moguće ulaze, ili tako da svi logički putevi budu testirani, očito predstavlja pouzdano ali praktički neprimjenljivo rješenje. Zbog toga se postavlja pitanje da li postoji način testiranja programa koji bi bio ekvivalentan iscrpnom testiranju. Osnovna postavka je da se ulazni skup može rastaviti u konačan broj ekvivalentnih klasa te da se testiranje određene klase može zamijeniti testiranjem jednog predstavnika. Prisutne su i druge strategije izbora test podataka, zasada najvećim dijelom bazirane na analizi strukture programa, npr. metoda prekrivanja grana. Navedena metoda zahtijeva takav izbor test sekvenci da svaka grana u programu mora biti predjena barem jednom. Metoda, kao i niz sličnih otkriva niz grešaka, ali se i dalje može reći da nije poznata.

Zbog navedenih prednosti i mana koje verifikiranje i testiranje imaju u ovom trenutku, poboljšanje ispravnosti programske podrške može se najekonomičnije postići izgradnjom sistema za ocjenu programske podrške. Pod sistemom za ocjenu programske podrške podrazumijeva se niz programa koji kontroliraju da li program posjeduje određena svojstva. Ta svojstva mogu biti ispravnost sintakse programa, ispravnost upravljačke strukture programa, ispravnost veza među modulima, itd. Sigurno je da ako program ima određena svojstva neće sadržavati veće greške. Sistemom za ocjenu programske podrške vrši se analiza konstrukcije sistema, testiranje te parcijalno utvrđivanje valjanosti. Parcijalno utvrđivanje valjanosti je proces utvrđivanja valjanosti programa do određenog nivoa pouzdanosti.

Spomenuti skup programa može na primjer sadržavati programe za analizu koda, kontrolu strukture programa, kontrolu ispravnosti veza među modulima, kontrolu ispravnog slijeda izvođenja operacija, programe za kontrolu ponašanja programa pri izvođenju, programe za automatsko generiranje testova itd.

U pogledu programske podrške u komutacijskim čvorovima telekomunikacijske mreže treba istaknuti neke specifičnosti. Komutacijski čvor se može promatrati kao diskretni automat. Općenito, diskretni automat se koristi za opis specifikacije i realizacije komutacijskog čvora. Sistem koji obradjuje mnogo poziva se dekomponira u višestruki virtualni automat koji obradjuje jedan poziv. Funkcija komutacije se opisuje kao ponašanje virtualnog automata sa dijagramom prijelaza stanja. Navedeni pristup donosi bitno novu kvalitetu u specificiranju sistema, realizaciji, dekompoziciji na module, a također i novi pristup u testiranju programske podrške. Naime, rezultati iz teorije diskretnih automata mogu se primijeniti na generiranje test sekvenci te minimizaciju broja neophodnih testova.

5. ORGANIZACIJA I SPREMANJE PODATAKA

Analize u svijetu pokazuju da broj spremljenih podataka u računarskim sistemima na memoriji s direktnim pristupom drastično raste. Razumljivo je, usporedo s povećanjem sistema za obradu podataka kao i sve raznolikijim obradama, da jedan od ključnih problema postaje organizacija, spremanje i pristup podacima potrebnim za različite obrade. Zato se postupcima organizacije i spremanja podataka posvećuje sve veća pažnja kako bi se cijena po spremljenom podatku snizila.

Suvremeni pristup organizaciji i spremanju podataka ide za tim da se podaci za različite obrade zajednički organiziraju i spremne. Struktura koja odgovara takvoj koncepciji organizacije podataka je baza podataka.

Bazu podataka [1] možemo definirati kao skup međusobno povezanih podataka spremljenih bez nepotrebne zalihosti s ciljem da na optimalan način posluže u raznim primjenama. Podaci su spremljeni nezavisno do programa koji ih koriste, zajedničkim pristupom dodaju se novi te mijenjaju i premještaju postojeći.

Možemo uočiti da baza podataka svojom strukturom i podacima predstavlja preslikan stvaran proces, za koje se i obrade realiziraju, na način pogodan za računarsku obradu. Sve veze koje postoje među entitetima u stvarnom procesu moraju se na adekvatan način preslikati u bazi podataka.

Da bi baza podataka bila efikasno organizirana potrebno je da ispuni slijedeće zahtjeve:

- sposobnost definiranja različitih i mnogostukih veza među podacima
- zadovoljavanje kriterija odgovora
- minimalna cijena po pohranjenom podatku
- minimalna ili kontrolirana redundancija
- sposobnost pronalazača traženih podataka za standardne i nestandardne zahtjeve
- cjelovitost baze podataka
- tajnost i ispravnost baze podataka
- kompatibilnost s prošlim stanjem
- kompatibilnost s budućim stanjem
- sposobnost prestrukturiranja fizičke organizacije podataka u cilju povećanja efikasnosti sistema
- jednostavnost opisa složenih struktura.

Me ulazeći detaljnije u navedene zahtjeve, istaknut ćemo samo najbitnije činjenice. Za ispunjenje navedenih zahtjeva bitan preduvjet su fizička i logička nezavisnost podataka. Zbog čestih promjena u procesu koji se preslikava u bazu podataka, potrebno je vršiti određene izmjene bilo u organizaciji podataka bilo u aplikacionim programima. Nužno je naglasiti da postoje tri nivoa opisa podataka. Jedan je opis koji definira aplikacioni programer za potrebu svoje aplikacije. Drugi nivo je globalni logički opis podataka u koji su uključeni svi podaci i logičke veze među njima za potrebe svih aplikacija. Opis podataka na prvom nivou je derivacija opisa na drugom nivou. I konačno, treći nivo opisa podataka je fizički opis podataka, dakle opis njihovog stvarnog pohranjivanja na mediju za spremanje podataka. Logička nezavisnost podataka znači da se opis globalne logičke strukture podataka može promijeniti bez mijenjanja aplikacionih programa. Fizička nezavisnost podataka omogućuje promjenu fizičkog položaja i organizacije podataka bez mijenjanja logičkog opisa podataka i aplikacionih programa. Zadovoljavanje kriterija odgovora je bitan preduvjet efikasnom radu. Pri tom je ovaj zahtjev daleko kritičniji u sistemima koji rade u realnom vremenu jer je vrijeme potrebno za dohvat podataka daleko kraće. Baza podataka bez redundancije bila bi idealna. Međutim, postojeća tehnička rješenja zahtijevaju za efikasan rad određeni nivo redundancije pa je razumljiviji zahtjev za kontroliranom redundancijom. Rad sistema mora biti takav da osigura u svakom trenutku dostup do svih podataka (cjelovitost), da podaci budu ispravni te da se sprječeni reautorizirani pristup podacima.

Danas se razvijaju posebni sistemi za organizaciju, održavanje i korištenje baze podataka (Data-Base Management Systems - DBMS). Do sad nije razvijen DBMS koji bi optimalno zadovoljio sve zahtjeve na bazu podataka i omogućio ostvarivanje najrazličitijih veza među podacima.

Najveća razlika između pojedinih DBMS-ima je u mogućnosti opisivanja veza među podacima. Postoje sistemi u kojima se jednostavno opisuje struktura stabla, postoje specijalizirani DBMS-ovi za velike baze podataka organizirane kao mreža podataka. Oni dozvoljavaju mnogo ulaza u sistem, čine se smanjuju vremena pretraživanja, dohвата i obrade podataka. Dalje razlike između DBMS-ima

su u količini glavne memorije zauzete programskom podrškom za održavanje rada, mogućnostima zaštite tajnosti i cjelovitosti podataka te u mogućnostima spašavanja baze ukoliko dodje do sklopovske ili programske pogreške. Osim toga, neki DBMS-i služe za održavanje obrade u realnom vremenu. Za optimalno strukturiranje baze podataka bitna pretpostavka je valjano izvršen postupak analize sistema za koji se informacijski sistem razvija. Osim osnovnih entiteta, pridjeljenih obilježja i veza među entitetima, ovom analizom moguće je dobiti čitav niz parametara bitnih za odabiranje pravilne organizacije baze podataka i odgovarajućeg DBMS-a. Na kraju treba istaći da izbor odgovarajućeg DBMS ovisi o različitim faktorima. Katkada bolje rješenje ograničava već postojeća sklopovska konfiguracija. Isti DBMS često nije kompatibilan s različitim računalima. Potrebno se uvijek truditi da s danim sredstvima odabiremo onaj DBMS koji će nam omogućiti što efikasniji rad sistema uz što nižu cijenu potrebnu za održavanje postojeće baze podataka.

Naša dosadašnja iskustva pokazuju da je poželjno katkada razviti vlastitu programsku podršku koja proširuje mogućnosti određenog DBMS-a jer na taj način prilagodimo dotični DBMS opisu naše strukture podataka [4].

6. ZAKLJUČAK

Dosadašnji rad na organizaciji obrada podataka u telekomunikacijskoj mreži rezultirao je razvojem informacijskih sustava primjenljivih u raznim nivoima telekomunikacijske mreže. Prikupljena iskustva predstavljaju podlogu za dalji razvoj, ali uz nužno podizanje kvalitete programske podrške. Da bi se to postiglo biti će potrebno provesti određene organizacione promjene u procesu razvoja programske podrške te razviti odgovarajuće programe za efikasnije testiranje, ocjenu efikasnosti i uopće poboljšanje svih aktivnosti u razvoju programske podrške.

Literatura:

- /1/ J.Martin, Computer Data Base Organization, Prentice Hall, N.J., Englewood Cliffs, 1975.
- /2/ I.Felja, D.Petrović, Z.Skočir, "Organizacija, održavanje i korištenje velikih baza podataka u telekomunikacijskim mrežama", XXII ETAN, Zadar, 1978.
- /3/ I.Felja, A.Pavić, D.Petrović, Z.Skočir, "Obrada podataka u telekomunikacijskoj mreži", XIII JUTEL, Ljubljana, 1979.
- /4/ Š.Dembitz, D.Petrović, "Mreža datoteka", Informatika 79, Bled, 1979.
- /5/ Š.Dembitz, I.Felja, A.Pavić, D.Petrović, Z.Skočir, "Rezultati istraživanja na razvoju informacijskog sistema za praćenje školovanja kadrova, XXIV ETAN, Priština, 1979.
- /6/ Znanstveni projekt, "Etapna realizacija integrirane komunikacijske mreže, Projekt 5 SIZ-a 1 za znanost SR Hrvatske, Godišnji izvještaji 1976-1979.
- /7/ T.Gilb, Software Metrics, Winthrop Publishers Inc., 1977.
- /8/ A.Pavić, Teorija pouzdanosti i kvaliteta programskih sistema, Referat na 1. Savjetovanju o pouzdanosti u elektrotehnici, Opatija, 4-6. ožujka 1981.
- /9/ B.W.Boehm et al., Characteristics of Software Quality, TRW and North Holland Publishing Comp., 1978.
- /10/ N.Wirth, Algorithmen und datenstrukturen, Teubner Studienbücher, 1975.
- /11/ A.W.Dijkstra, A discipline of programming, Prentice-Hall, 1976.
- /12/ Dahl, Dijkstra, Hoare, Structured programming, Academic, Press INC.1972.

- /14/ Tsun S.Chow. "Testing Software Design Modeled by Finite-State Machines" IEEE Trans.on Software Engineering, Vol SE-4 No.3, pp 176-187, May 1978.
- /15/ W.E.Haweden, "Methodology for the Generation of Program Test Data", IEEE Trans. on Comput. Vol C 24 pp 554-560, May 1975.
- /16/ J.B.Goodenough and S.L.Gerhart "Toward a Theory of Test Data Selection, Proc.Int.Conf.on Reliable Software pp 493-510, Los Angeles 1975.
- /17/ C.V.Ramamoorthy and S.F.Ho, "Testing Large Software with Automated Software Evaluation Systems", Proc.Int.Conf. on Reliable Software, pp 382-394, Los Angeles, 1975.
- /18/ H.Kawashima, T.Arima, Y.Shinizu, "Software Structure for Today and Tomorrow", ISS 79 Paris, May 1979.
- /19/ P.Henderson, "Finite State Modeling in Program Development", Proc.Int. Conf. on Reliable Software, pp 221-227, Los Angeles 1975.