

FORMIRANJE I PRETRAŽIVANJE
BAZA PODATAKA
U SISTEMU
NAUČNIH I TEHNOLOŠKIH
INFORMACIJA
SRBIJE

redaktor

BRANISLAV LAZAREVIC

M-13 994

MINISTARSTVO ZA NAUKU I TEHNOLOGIJU
REPUBLIKE SRBIJE

FORMIRANJE I PRETRAŽIVANJE BAZA PODATAKA U SISTEMU NAUČNIH I TEHNOLOŠKIH INFORMACIJA SRBIJE

REDAKTOR: BRANISLAV LAZAREVIC

AUTORI:

BENDER MIROSLAV
CVETANOVIĆ SVETLANA
DIKANOVIĆ VIOLETA
ĐURIĆIĆ DRAGAN
IVKOVIĆ MIODRAG
KONEČNI ALEG
KORUNOVIĆ DUŠAN
LAZAREVIC BRANISLAV
MILIVOJEVIĆ LJUBIŠA
NEŠKOVIĆ SINIŠA
PAUNOVIĆ ĐORĐE
SURLA DUŠAN

Izdavač

Ministarstvo za nauku i tehnologiju Republike Srbije

Za izdavača

dr Ratko Uzunović



Recenzenti

Prof. dr Pavle Mogin

Prof. dr Dušan Starčević

Uređivački odbor

dr Đorđe Paunović, dr Branislav Lazarević, dr Dušan Surla

dr Miodrag Ivković, mr Dušan Korunović

Tehnički urednik

Milorad Ladarević

Štampa

Politop-P, Beograd

Tiraž

600 primeraka

Maj 1996.

ISBN 86-7282-001-0

PREDGOVOR

U toku 1991. godine započete su u Srbiji aktivnosti na razvoju novog koncepta Sistema naučnih i tehnoloških informacija (SNTIS). Sistem je preko odgovarajuće računarske mreže trebalo da poveže akademske i druge naučno-istraživačke i dokumentacione organizacije i omogući vezu sa sličnim sistemima i mrežama u svetu. Dva su osnovna razloga uslovila otpočinjanje aktivnosti na razvoju SNTIS: Sistem naučno-tehnoloških informacija Jugoslavije (SNTIJ), koji su zajednički razvijale sve republike prethodne Jugoslavije prestao je da funkcioniše, a već je tada moglo da se predviđi da će i koncepcija i tehnološka osnova na kojima je SNTIJ izgrađen (homogena računarska mreža sa jedinstvenim centralnim "hostom", centralizovani razvoj) biti veoma brzo prevaziđeni.

U Republici Srbiji je 1991. godine usvojena nova koncepcija razvoja SNTIS. SNTIS je koncipiran kao otvoreni sistem u sadržajnom, institucionalnom, tehnološkom i strukturnom smislu. Sadržajno, sistem je trebalo da obuhvati ne samo konvencionalne naučno-tehnološke informacije (bibliografske baze podataka i njima pridružene servise), već i druge tipove baza podataka i druge načine informisanja uopšte, posebno neposrednu razmenu informacija između učesnika u naučno-istraživačkom radu preko različitih servisa koje pružaju savremene računarske mreže. Institucionalno, SNTIS treba da omogući da svaka naučno-istraživačka, razvojna i dokumentaciona organizacija može da postane njegov deo i da sama, u skladu sa opštom koncepcijom, standardima i opštim planovima razvoja, definiše i ostvaruje svoju ulogu u njemu. Tehnološki, SNTIS se razvija kao heterogeni, distribuirani informacioni sistem čije se jedinstvo ostvaruje preko minimuma neophodnih standarda koje svaki deo sistema u svom razvoju mora da poštuje, uz stalno praćenje i implementaciju novih tehnologija u ovoj oblasti. Strukturno, SNTIS je skup većeg broja lokalnih meža i podistema, sa jedinstvenom fizičkom računarsko-komunikacionom mrežom oslonjenom na javni PTT sistem.

Mreža SNTIS podržava sve standardne servise jedne računarske mreže. Poseban značaj i zahtevi postavljeni su pred servis za pretraživanje baza podataka. Ovaj servis je trebalo da obezbedi da se zameni raniji sistem zasnovan na softveru

ATLASS i homogenoj računarskoj mreži VAX- Digital racunara, koji je prvenstveno podržavao bibliotečki podsistem i bio osnova za centralizovani servis za pretraživanje informacija na hostu u Mariboru i razvije novi, baziran na prethodno iznetoj koncepciji. Sistem za pretraživanje informacija u okviru SNTIS razvijan je u okviru projekta, finansiranog od strane Ministarstva za nauku i tehnologiju Republike Srbije. Projekat je realizovan u zajedničkom radu nekoliko naučno-istraživačkih institucija preko sledećih podprojekata:

- Definisanje arhitekture sistema, osnovne metodologije, dogradnja CASE alata, kao i koordinacija izrade celokupnog projekta - Fakultet organizacionih nauka, Beograd;
- Projektovanje i realizacija Jedinstvenog sistema za indeksiranje i pretraživanje dokumenata - Institut "Mihajlo Pupin", Beograd;
- Projektovanje i izgradnja Bibliotečkog podsistema SNTIS - Prirodno-matematički fakultet Novi Sad;
- Projektovanje Sistema za formiranje i pretraživanje baze dokumenata i publikovanje - Univerzitet u Nišu, JUNIS;
- Projektovanje i izgradnja Rečnika podataka SNTIS, Računski centar Univerziteta u Beogradu.

Autori ove monografije su članovi užeg tima koji je radio na projektu. Imajući u vidu zajednički rad i diskusije na mnogobrojnim sastancima tima može se reći da su svi rezultati prikazani u monografiji zajednički. Ipak ističemo posebno autorstvo za pojedina poglavlja:

Poglavlje 1: **Uvod**, Branislav Lazarević.

Poglavlje 2: **Koncepcija, sadašnje stanje i pravci daljeg razvoja sistema naučnih i tehnoloških informacija**, Đorđe Paunović, Miodrag Ivković i Dusan Korunović.

Poglavlje 3: **Karakteristike otvorenih sistema**, Branislav Lazarević i Ljubiša Milivojević.

Poglavlje 4: **Metodologija**, Branislav Lazarević i Sinisa Nešković.

Poglavlje 5: **Arhitektura sistema za pretraživanje informacija u SNTIS**, Branislav Lazarević i Sinisa Nesković.

Poglavlje 6: **Jedinstveni sistem za indeksiranje i pretraživanje dokumenata**, Ljubiša Milivojević, Dragan Đuričić i Aleg Konečni.

Poglavlje 7: **Sistem za formiranje i pretraživanje baze dokumenata i publikovanje**, Dusan Korunović i Svetlana Cvetanović.

Poglavlje 8: **Bibliotečki informacioni sistem**, Dušan Surla i Miroslav Bender.

Poglavlje 9: **Rečnik podataka SNTIS**, Violeta Dikanović, Sinša Nešković i Miodrag Ivković.

U ovoj monografiji sažeto je opisan SNTIS i prikazani rezultati rada na projektu Sistema za formiranje i pretraživanje informacija. Autori su smatrali da su sam Projekat i metodološki pristup njegovoju izradi, posebno efektivna kooperacija više institucija, dovoljno interesantni, a dobijeni rezultati dovoljno znacajni da ih treba prikazati ne samo kao softverski proizvod sa odgovarajućom dokumentacijom, već i na ovaj način. Ovo tim presto su knjige koje prikazuju jedan savremeni pristup razvoju softvera i ilustruju takav pristup na vrlo složenom softverskom sistemu, sa određenim specifičnim originalnim rešenjima, veoma retke, ne samo kod nas.

Autori se zahvaljuju Ministarstvu za nauku i tehnologiju koje je finansiralo rad na projektu i izdavanje ove monografije. Zahvalnost dugujemo i mnogim povremenim članovima projektnog tima koji su dali doprinos pojedinim rezultatima projekta, a nisu autori ove monografije: dr Zori Konjovic, docentu Fakulteta tehničkih nauka u Novom Sadu, Tatjani Vulić i Ivani Tričković, Institut za matematiku Novi Sad, zatim, Novki Šokica, Biblioteka maticе srpske, Gordani Popović-Bosković, Univerzitetska biblioteka "S. Marković", Beograd i Ljiljani Kovačević, Narodna biblioteka Srbije, Dragana Pić, Računski centar Univerziteta u Beogradu, Snežani Arambašić-Aleksić i Zoranu Bošković, Institut "M. Pupin".

Izražavamo zahvalnost recenzentima: Prof. dr Pavlu Moginu i Prof. dr Dušanu Starčeviću na primedbama i korisnim sugestijama koje su ugrađene u poslednju varijantu teksta i koje su svakako doprinele njegovom kvalitetu.

Takođe, autori se zahvaljuju i Miloradu Lađareviću i Sibu Mehmedoviću koji su radili na tehničkom uređivanju knjige.

Beograd, april, 1996. godine

Autori

S A D R Ž A J

Poglavlje 1.

UVOD

Poglavlje 2.

KONCEPCIJA, SADAŠNJE STANJE I PRAVCI DALJEG RAZVOJA SISTEMA NAUČNIH I TEHNOLOŠKIH INFORMACIJA

2.1. UVOD	9
2.2. OSNOVNA OPREDELJENJA U DALJEM RAZVOJU	10
2.2.1. SNTI Srbije i Jugoslavije	10
2.2.2. Ciljevi	11
2.2.3. Obuhvat sistema	11
2.3. STRUKTURA I SADRŽAJ SNTIS	12
2.4. KONCEPCIJA, STANJE I PLANOVI RAZVOJA	15
2.4.1. Načelna opredeljenja	15
2.4.2. Međunarodni i međumesni segmenti mreže SNTIS	18
2.4.3. Mesni segment Beograda, mreža "BEONET"	24
2.4.3.1. Magistralni prsten BEONET-a	25
2.4.3.2. Lokalne mreže BEOBET-a	26
2.4.4. Računarska mreža Univerziteta u Nišu	31
2.4.5. Računarska mreža Univerziteta u Kragujevcu	37
2.4.6. Računarska mreža Univerziteta u Novom Sadu	39
2.4.7. Računarska mreža Univerziteta u Prištini	39
2.5. KORIŠĆENJE SNTI MREŽE	41
2.5.1. Raspoložive servisne usluge u mreži	41
2.5.2. Način priključenja korisnika u mrežu SNTIS	43
2.5.3. Važnije mreže i broj povezanih računara u SNTI mreži	44
2.6. LOGIČKE MREŽE SNTIS	46
2.6.1. Upravljačka mreža SNTIS	46
2.6.2. Plan razvoja bibliotečke mreže	47
2.6.3. Mreža specijalizovanih sistema	49
2.6.3.1. Opšti ciljevi izgradnje mreže specijalizovanih sistema SNTIS (SSNTIS)	51
2.6.3.2. Ciljevi izgradnje pojedinačnih specijalizovanih sistema	51
2.6.3.3. Osnovna opredeljenja u realizaciji mreže SSNTIS	52
2.6.4. Mreža poslovno - tehnoloških informacija (MPTI)	53
2.6.4.1. Ciljevi i zadaci mreže poslovnih i tehnoloških informacija	53
2.7. PERSPEKTIVE DALJEG RAZVOJA MREŽE SNTIS	55
LITERATURA	56

Poglavlje 3.

KARAKTERISTIKE OTVORENIH SISTEMA	59
LITERATURA	63

Poglavlje 4.

METODOLOGIJA	65
4.1. OPŠTE KARAKTERISTIKE STANDARDNE METODOLOGIJE RAZVOJA INFORMACIONIH SISTEMA	65
4.2. STRUKTURNΑ SISTEMSКА ANALIZA (SSA) I OBJEKTNΑ SPECIFIKACIJA SOFTVERA 4.2.1. Konvencionalna strukturalna sistemska analiza 4.2.2. Objektno orijentisana specifikacija kao rezultat SSA	68
4.3. SPECIFIKACIJA BAZNOG OBJEKTA 4.3.1. Struktura PMOV 4.3.2. Operacije u PMOV 4.3.3. Ograničenja i pravila integriteta 4.3.4. Potpuna specifikacija baznog objekta	74
4.4. SPECIFIKACIJA APLIKACIONIH OBJEKATA	87
4.5. PROJEKTOVANJE 4.5.1. Generički implementacioni model 4.5.2. Dekompozicija "atomskih logičkih procesa" 4.5.3. Fizičko projektovanje baznih objekata (baza podataka)	95
4.6. CASE ALATI	101
4.7. PRIMENA METODOLOGIJE LITERATURA	102
	102

Poglavlje 5.

ARHITEKTURA SISTEMA ZA PRETRAŽIVANJE INFORMACIJA U SNTIS	105
5.1. SPECIFIKACIJA SISTEMA ZA PRETRAŽIVANJE INFORMACIJA 5.1.1. Formiranje i održavanje Rečnika podataka 5.1.2. Osnovne funkcije Sistema za pretraživanje informacija	107
5.2. FIZIČKA ARHITEKTURA SISTEMA ZA PRETRAŽIVANJE INFORMACIJA	112
	117

Poglavlje 6.

JEDINSTVENI SISTEM ZA INDEKSIRANJE I PRETRAŽIVANJE DOKUMENATA	119
6.1. KARAKTERISTIKE I KOMPONENTE SISTEMA	119
6.2. ARHITEKTURA SISTEMA 6.2.1. Arhitektura servera 6.2.2. Komunikacije između servera 6.2.3. Apstrakcije računarske mreže i baza podataka	121
	122
	124

6.3. SERVER ZA INDEKSIRANJE I PRETRAŽIVANJE DOKUMENATA	125
6.3.1. Organizacija servera za indeksiranje i pretraživanje dokumenata	125
6.3.2. Podsistem za izvršavanje upita	127
6.3.2.1. Tipovi upita i komandni jezik	127
6.3.2.2. Koraci u izvršavanju upita	129
6.3.2.3. Sitaksna analiza upita	130
6.3.2.4. Semantička analiza upita	130
6.3.2.5. Plan izvršavanja upita	131
6.3.2.6. Transformacija plana izvršavanja upita	133
6.3.2.7. Interpretacija plana izvršavanja upita	134
6.3.3. Podsistem za indeksiranje dokumenata	135
6.3.4. Baza indeksa	136
6.3.4.1. Aplikacioni programski interfejs baze indeksa	139
6.3.4.1.1. Funkcije za pristup bazi indeksa	139
6.3.4.1.2. Obrada grešaka	141
6.3.4.2. Implementacija baze indeksa	142
6.3.5. Mrežna podrška	143
6.3.5.1. Aplikacioni programski interfejs za pretrazivanje i indeksiranje baze indeksa	143
6.3.5.2. Protokol za komunikaciju sa serverom za indeksiranje i pretraživanje dokumenata	144
6.4. SERVER ZA PRISTUP BAZI IZVORNIH DOKUMENATA	148
6.4.1. Aplikacioni programski interfejs za pristup bazi izvornih dokumenata	149
6.4.2. Protokol za komunikaciju sa serverom za pristup bazi izvornih dokumenata	150
6.5. GEJTVJEJI SISTEMA	152
6.5.1. Gopher	153
6.5.2. WWW	155
6.6. ZAKLJUČAK	156
LITERATURA	157

Poglavlje 7.

SISTEM ZA FORMIRANJE I PRETRAZIVANJE BAZE DOKUMENATA I PUBLIKOVANJE

7.1. OPIS SISTEMA ZA FORMIRANJE I ODRŽAVANJE BAZE DOKUMENATA	159
7.1.1. Proces Definisanje standarda za opis i razmenu dokumenata	160
7.1.2. Proces Formiranje i održavanje baze dokumenata	168
7.1.3. Proces Priprema za indeksiranje	171
7.1.4. Proces Direktno pretrazivanje	174
7.1.4.1. Direktno pretraživanje preko standardnih upitnih jezika	174
7.1.4.2. Specijalni jezik za direktno pretraživanje	176
7.1.5. Proces Publikovanje	177

7.2. IMPLEMENTACIJA SISTEMA ZA FORMIRANJE I PRETRAŽIVANJE BAZE DOKUMENATA	178
7.2.1. Programi za formiranje standarda za opis i razmenu	179
7.2.2. Programi za obradu dokumenata	181
7.2.3. Program kojim se vrši priprema za indeksiranje	182
LITERATURA	183

Poglavlje 8.

BIBLIOTEČKI INFORMACIONI SISTEM	185
8.1. FUNKCIJE BIBLIOTEČKOG INFORMACIONOG SISTEMA	186
8.2. SPECIFIKACIJA BAZNOG OBJEKTA BIBLIOTEČKOG INFORMACIONOG SISTEMA	188
8.2.1. Model podataka standarda za održavanje (UNIMARC)	188
8.2.2. Model podataka za radno okruženje bibliotekara	191
8.2.3. Model podataka baze bibliografske građe	193
8.2.4. Model podataka za praćenje cirkulacije bibliografskih dokumenata	195
8.3. SPECIFIKACIJA APLIKACIONIH OBJEKATA	198
8.3.1. Funkcija Formiranje bibliografske građe	198
8.3.2. Specifikacija aplikacionog objekta Formiranje bibliografske građe	200
8.3.3. Fizička arhitektura lokalnog bibliotečkog informacionog sistema	213
8.4. IMPLEMENTACIJA LOKALNOG SEGMENTA BIBLIOTEČKOG INFORMACIONOG SISTEMA	217
8.5. ZAKLJUČAK	218
LITERATURA	218

Poglavlje 9.

REČNIK PODATAKA SNTIS	221
9.1. ARHITEKTURA REČNIKA PODATAKA U SKLADU SA STANDARDIMA	222
9.2. FORMIRANJE I PRETRAŽIVANJE REČNIKA PODATAKA	226
9.3. SADRŽAJ REČNIKA	229
9.3.1. Rečnik podataka za opis opstih karakteristika baza podataka u SNTIS	229
9.3.2. Rečnik podataka za opis mrežnih čvorova u SNTIS	232
9.3.3. Rečnik podataka za opis strukture baze podataka za pretraživanje	235
9.3.4. Rečnik podataka kao baza SPI	235
9.3.4.1. SPI dokument za opis opstih karakteristika baza podataka	235
9.3.4.2. SPI dokument za opis mrežnih čvorova	238
9.4. REČNIK PODATAKA KAO AKTIVNI DEO SPI	240
9.4.1. Izbor baze podataka za pretraživanje i rutiranje upita	241
9.5. IMPLEMENTACIJA REČNIKA PODATAKA	245
LITERATURA	252
INDEKS	253

Poglavlje 1.

UVOD

U Republici Srbiji je 1991. godine usvojena nova koncepcija razvoja Sistema naučnih i tehnoloških informacija (SNTIS) koji je trebalo da zameni odgovarajući sistem prethodne Jugoslavije - SNTIJ. SNTIJ je prestao da funkcioniše, a već je tada moglo da se predvidi da će i koncepcija i tehnološka osnova na kojima je SNTIJ izgrađen (homogena računarska mreža sa jedinstvenim centralnim "hostom", centralizovani razvoj) biti veoma brzo prevaziđeni.

SNTIS je koncipiran kao otvoreni sistem u sadržajnom, institucionalnom tehnološkom i strukturnom smislu. Sadržajno, sistem je trebalo da obuhvati ne samo konvencionalne naučno-tehnološke informacije (bibliografske baze podataka i njima pridružene servise), već i druge tipove baza podataka i druge načine informisanja uopšte, posebno neposrednu razmenu informacija između učesnika u naučno-istraživačkom radu preko različitih servisa koje pružaju savremene računarske mreže. Institucionalno, SNTIS treba da omogući da svaka naučno-istraživačka, razvojna i dokumentaciona organizacija može da postane njegov deo i da sama, u skladu sa opštom koncepcijom, standardima i opštim planovima razvoja, definiše i ostvaruje svoju ulogu u njemu. Tehnološki, SNTIS se razvija kao heterogeni, distribuirani informacioni sistem čije se jedinstvo ostvaruje preko minimuma neophodnih standarda koje svaki deo sistema u svom razvoju mora da poštuje, uz stalno praćenje i implementaciju novih tehnologija u ovoj oblasti. Strukturno, SNTIS je skup većeg broja lokalnih meža i podsistema, sa jedinstvenom fizičkom računarsko-komunikacionom mrežom oslonjenom na javni PTT sistem. Koncepcija na kojoj je zasnovan SNTIS, njegove komponente, sadašnje stanje i pravci njegovog daljeg razvoja detaljno su prikazani u sledećem poglavlju monografije.

Mreža SNTIS podržava sve standardne servise jedne računarske mreže. Poseban značaj i zahtevi postavljeni su pred servis za pretraživanje baza

podataka. Ovaj servis je trebalo da zameni raniji sistem zasnovan na softveru ATLOSS i homogenoj računarskoj mreži VAX- Digital racunara, koji je prvenstveno podržavao bibliotečki podsistem i bio osnova za centralizovani servis za pretrživanje informacija na hostu u Mariboru i razvije novi sistem za pretraživanje informacija baziran na koncepciji otvorenih heterogenih informacionih sistema. Servis je realizovan preko projekta Sistema za pretraživanje informacija (SPI), a rezultati ovog projekta prikazuju se u Monografiji.

Sa tačke gledišta SPI zahtev da se SNTIS razvija kao otvoreni sistem znači da je neophodno da se omoguci da se pojedini lokalni SPI (baze podataka i odgovarajuci servisi) razvijaju autonomno, na specifičnim hardversko - softverskim platformama, uz postovanje minimalnog skupa standarda, a da se istovremeno omoguci kako kooperacija u stvaranju pojedinih distribuiranih baza, tako i njihovo jedinstveno pretraživanje u celokupnom sistemu. Drugim rečima, neophodno je razviti jedinstveni SPI, potpuno transparentan i u odnosu na prostornu distribuciju i u odnosu na karakteristike lokalnih implementacionih okolina pojedinih autonomnih podsistema. Jedinstveni SPI treba da se razvija od već postojećih i novoizgrađenih lokalnih podsistema, a otvorenost sistema treba da obezbedi i jednostavnu migraciju sistema na buduće nove tehnologije.

Da bi se realizovali ovakvi zahtevi i SPI izgradio kao distribuirani, heterogeni otvoreni sistem bilo je neophodno usvojiti jednu opstu arhitekturu otvorenih sistema kao osnovu za definisanje komponenti preko kojih ce se SPI razvijati. U Poglavlju 3 prikazuje se jedna ovakva arhitektura i iz nje je izведен skup standarda preko koga se SPI SNTIS razvija. Posebno se ističe potreba za definisanjem standardne metodologije razvoja IS, standarda za rečnik podataka koji treba da bude osnova za upravljanje jednim ovakvim složenim sistemom i alat za ostvarivanje zahtevane transparentnosti, standardi razmene podataka i mnogi drugi koji se u poglavljima koja slede detaljnije opisuju. Prikazana arhitektura otvorenih informacionih sistema nije samo značajna za razvoj SPI, vec i svih drugih složenih distribuiranih sistema, a može da pretstavlja i osnov za definisanje državne politike razvoja specifičnih informatičkih standarda.

Standardna metodologija razvoja IS je veoma znacajna za otvorene sisteme, jer se može smatrati da je jedna od bitnih karakteristika otvorenih sistema njihov distribuirani razvoj u kome se u razvoju jednog funkcionalnog dela sistema koriste komponente razvijene u drugim podsistemima. I SPI je razvijan na takav način. U njegovom razvoju učestvovalo je više institucija, pa je za efikasan razvoj i efektivnu integraciju različitih komponenti bilo neophodno usvojiti jedinstvenu metodologiju. Ponovo naglašavamo da je

ovakva jedinstvena metodologija neophodna ne samo za razvoj SPI i SNTIS uopšte, nego ona treba da bude i osnova za razvoj svih velikih "državnih" infrastrukturnih informacionih sistema. Zbog toga su mnoge vlade za razvoj svojih sistema usvojile standardne metodologije. Metodologija korišćena u ovom Projektu ima sve karakteristike ovakvih "de facto" standarda. U čestim promenama informacionih tehnologija samo će jedna apstraktna specifikacija IS, zasnovana na jedinstvenoj metodologiji, ostati nepromenjena.

Standardna metodologija, opisana detaljnije u Poglavlju 4, predstavlja jedan transformacioni, objektno orijentisan pristup razvoju IS koji se zasniva, s jedne strane, na formalnoj specifikaciji IS i automatizovanoj transformaciji ove specifikacije u različite implementacione okoline pomoću specijalno razvijenog CASE alata, a s druge, na objektnoj dekompoziciji sistema. Osnovne komponente specifikacije sistema predstavljaju specifične verzije Strukturne sistemske analize koja rezultuje u objektno orijentisanu specifikaciju aplikacija i Prošireni model objekti veze kao način specifikacije baze podataka posmatranog sistema. Metodologija definiše i standarde implementacije otvorenog sistema preko jedne opšte "client-server" arhitekture. Metodologija je primenjivana za razvoj svih komponenti softvera za SPI, a njena primena je detaljno prikazana na primeru razvoja Bibliotečkog pod система у Poglavlju 8.

U Poglavlju 5 opisana je arhitektura Sistema za pretraživanje informacija u SNTIS. Definisani su koncepti autonomnih SPI i kooperativnih sistema kao skupa autonomnih SPI koji zajednički formiraju i održavaju pojedine baze podataka. Definisan je i koncept Rečnika podataka SNTIS kao baze podataka koja, pored svoje uloge u upravljanju razvojem i korišćenjem celokupnog sistema, sadrži i sve neophodne modele i pravila njihovog uzajamnog preslikavanja koja su neophodna za ostvarivanje zahtevane transparentnosti sistema.

Sistem za pretraživanje informacija dekomponovan je na sledeće funkcije:

(1) Formiranje i održavanje rečnika podataka, koji pored procesa formiranja i održavanja baze podataka o institucijama, čvorovima i lokalnim mrežama SNTIS, sadrži detaljan opis baze podataka po čvorovima mreže, preko jedinstvenog (standardnog) modela baze podataka za pretraživanje. Standardna struktura baze podataka za pretraživanje je jedna ravna struktura u kojoj je svaki zapis agregacija polja različitih tipova. Nad ovakvom ravnom strukturom definisan je i realizovan jedinstveni jezik za pretraživanje u SNTIS. Bilo koja baza drugačije strukture se mora preslikati u ovaj standardni model, da bi se "otvorila" za jedinstveno pretraživanje u

SNTIS. Ako je struktura lokalne baze opisana preko SGML standarda, može se za ovu transformaciju primeniti opšti algoritam i odgovarajuci program opisan u Poglavlju 7.

(2) Osnovne funkcije sistema za pretraživanje, Formiranje i održavanje baze (primarnih) dokumenata, Obrada za pretraživanje, koja podrazumeva transformaciju baze podataka proizvoljne strukture u standardnu strukturu za pretraživanje i indeksiranje tako dobijene strukture, odnosno formiranje baze indeksa za odgovarajucu bazu dokumenata, Pretraživanje baza dokumenata korišćenjem jedinstvenog jezika i baze indeksa i Izvestavanje i publikovanje direktno na osnovu baze dokumenata ili na osnovu rezultata pretraživanja.

(3) Ostale funkcije, pratežno specifične funkcije administracije pojedinih lokalnih i kooperativnih sistema, kao što je na primer bibliotečka razmena, nabavka primarnih dokumenata i slično.

Celokupna arhitektura SPI SNTIS opisana je u ovom poglavlju preko skupa opstih Dijagrama tokova podataka i Modela objekti-veze, a ovi modeli se, za pojedine funkcije, detaljno razrađuju u poglavljima koja slede.

Poglavlje 6 prikazuje jezgo SPI SNTIS, Jedinstveni sistema za indeksiranje i pretraživanje dokumenata. Arhitektura ovoga sistema opisana je na standardni metodoloski način, preko četvoroslojnog modela "client-server sistema" i čine je:

- Server za indeksiranje i pretraživanje baze dokumenata (indeksa) na kome se vrši prihvatanje i indeksiranje dokumenata standardne "ravne strukture", leksicka, sintaksna i semanticka analiza upita, formiranje plana izvršenja, kao i optimizacija i interpretacija plana izvršenja upita;
- Server za pristup bazi izvornih dokumenata koji prihvata zahtev za izvornim dokumentom, formira polja izvornog dokumenta i na kome se, sa druge strane obavlja formiranje i održavanje baze dokumenta;
- Gopher i WWW gejtvej za vezu ovoga sistema sa ovim široko korišćenim Internetovim servisima preko kojih se ostvaruje distribuirano pretraživanje korišćenjem standardnih korisničkih aplikacija, što ukida potrebu za definisanjem i distribucijom posebne klijentske aplikacije SPI.

U ovom poglavlju se opisuje sintaksa i semantika jezika za pretraživanje, koji je veoma sličan jeziku sistema DIALOG i detaljno se prikazuju sve faze izvršenja upita. Detaljno se prikazuje implementacija Podistema za

indeksiranje dokumenata, daje se struktura baze indeksa, prikazuje način njene realizacije u db_VISTA sistemu za upravljanje bazom podataka i opisuju aplikacije indeksiranja. Posebna pažnja je posvećena softveru za mrežnu podršku, opisuju se aplikacioni programski interfejs za pretraživanje i indeksiranje, protokol za komunikaciju sa serverom za indeksiranje i pretraživanje, zatim aplikacioni interfejs za server za pristup bazi izvornih dokumenata i protokol za komunikaciju sa ovim serverom, kao i gejtevi sa servisima Interneta, prvenstveno Gopher-om i WorldWideWeb-om.

Bez obzira što SPI SNTIS može da omogući pristup bilo kojoj bazi dokumenata, sama funkcija "otvaranja" pojedinih baza podataka za pretraživanje, kooperativni rad autonomnih SPI pri formiraju zajednickih baza dokumenata i druge funkcije SPI može se značajno unaprediti standardizacijom procesa formiranja i održavanja baze izvornih (full text) dokumenata. Poglavlje 7. prvenstveno je posvećeno ovom problemu. Diskutuju se različiti standardi strukturiranja dokumenata, obrazlaže prihvatanje Standarda ISO 8879 - SGML (Standard Generalized Markup Language) i on se detaljno opisuje. SGML je formalizovan jezik kojim se tekst nekog dokumenta može obeležavati tako da se u potpunosti očuva njegov sadržaj i opiše njegova struktura i njegov izgled. Strukturiranje dokumenta pomoću SGML-a omogućava da se obeleženi izvorni dokument pretražuje po bilo kom svom obeležju, bez dodatne manuelne obrade. U tom cilju je razvijen i implementiran algoritam za transformaciju bilo kog standardizovanog i označenog izvornog dokumenta u strukturu koju koristi Jedinstveni sistem za pretraživanje i indeksiranje.

U Poglavlju 7 analizira se proces "direktnog pretraživanja". Pod "direktnim pretraživanjem" podrazumeva se proces pretraživanja baze dokumenata bez prethodnog njihovog podvrgavanja procesu indeksiranja, odnosno pretraživanje koje je zasnovano na SGML opisu dokumenta, a ne na izvedenoj bazi indeksa. Analiziraju se dve mogućnosti ovakvog pretraživanja: (i) realizacijom baze primarnih dokumenata preko relacione baze podataka čija se struktura dobija transformacijom konkretnog SGML opisa ili (ii) preko specijalizovanog jezika za direktno pretraživanje, baziranog neposredno na SGML opisu. Direktno pretraživanje je neposredno vezano sa procesom publikovanja koga SPI takođe treba da podrži. Koristeći direktno pretraživanje, bilo nad celokupnom bazom izvornih dokumenata ili nad "privremenom" bazom formiranom od dokumenata koji su rezultat pretraživanja primenom Jedinstvenog sistema za pretraživanje, mogu se odabrati i pogodno strukturirati dokumenti koji se dalje obrađuju u procesu publikovanja, što se u ovom poglavlju takođe ukratko prikazuje.

Jedan od najznačajnijih razloga razvoja sopstvenog softvera za SPI bio je zamena postojećeg, odnosno razvoj novog Bibliotečkog informacionog sistema SNTIS. Kao što je već ranije rečeno, postojeći Bibliotečki sistem realizovan je preko homogene mreže VAX računara, sa centralizovanim hostom, centralizovanim bazama podataka (katalozima) na njemu i sa centralizovanim razvojem i održavanjem celokupnog podsistema. Prestankom rada SNTIJ, bibliotečki sistem se raspao na skup izolovanih lokalnih sistema, sa sopstvenim bazama podataka, bez mogućnosti kooperacije, sa softverom za koga ne postoji izvorni kod, niti bilo koja druga dokumentacija. Zbog svega toga ovaj sistem nije bilo moguće dalje održavati i razvijati, a postojeće baze, opremu i rešenja sa kojima su korisnici zadovoljni trebalo je sačuvati. Poglavlje 8 prikazuje rezultate rada na razvoju novog Bibliotečkog sistema. Pored toga što prikazuje ovaj podsistem, Poglavlje 8 najdetaljnije prikazuje primenu usvojene standardne metodologije razvoja IS.

Osnovne funkcije Bibliotečkog sistema su:

1. Definisanje standarda za obradu i pretraživanje. Osnovni standard za obradu bibliotečkih dokumenata je UNIMARC standard i način formiranja i održavanja baze podataka ovog standarda, koji je osnova za sve druge obrade, detaljno se prikazuje.
2. Formiranje bibliografske građe, osnovni proces u sistemu u kome se, u skladu sa UNIMARC standardom, na bazi primarnog dokumenta, formira odgovarajući bibliotečki sekundarni dokumenat. Ova funkcija koristi Jedinstveni sistem za indeksiranje i pretraživanje dokumenata i to kako njegove osnovne funkcije, tako i njegove module za mrežnu komunikacionu podršku, preko koje se može u distribuiranom okruženju realizovati proces uzajamne katalogizacije.
3. Bibliotečko izveštavanje i dokumentovanje u kojoj se formiraju specifični bibliotekarski izvestaji i katalozi.
4. Korisničko pretraživanje, obezbeđuje obradu upita korisnika i prikazivanje rezultata pretraživanja u zahtevanom predefinisanom formatu. I ova funkcija se realizuje preko Jedinstvenog sistema za pretraživanje dokumenata.
5. Cirkulacija dokumenata, koja obuhvata evidentiranje korisnika biblioteke, praćenje korišćenja fonda biblioteke, rezervisanje i zaduživanje dokumenata.

Detaljno je opisana baza podataka celokupnog sistema preko koncepta baznog objekta i data objektno orijentisana specifikacija funkcije Formiranje bibliografske građe. Opisana je fizička arhitektura lokalnog bibliotečkog sistema preko opšteg implementacionog modela "client-server" arhitekture.

Softver za lokalni bibliotečki sistem realizovan je u više različitih okruženja (VAX/VMS, SCO UNIX). Njegovu jednostavnu prenosivost iz jednog u drugo okruženje omogućili su njegova modularnost, programski jezik C i sistem za upravljanje bazom podataka db_VISTA, koji se može implementirati na većem broju sistema, kao i iskustvo u modifikovanju izvornog koda ovog SUBP-a, sa kojim se raspolaže.

Softver za Bibliotečki sistem realizovan je tako da preuzme "softversku zaostavštinu" postojećeg sistema i on je koncipiran tako da u najvećoj meri "imitira" postojeći sistem (ATLASS). Međutim, celokupan Bibliotečki sistem u okviru SNTIS ne zahteva da se na svakoj lokaciji implementira ovaj softver. Bibliotečki sistem takođe treba da bude otvoreni heterogeni IS, sa različitim lokalnim "bibliotečkim softverima". Ako svi lokalni sistemi prate jedinstveni UNIMARC standard, SPI SNTIS može da obezbedi kooperaciju (na primer uzajamnu katalogizaciju) lokalnih podsistema i bez centralnog servera, odnosno centralnih kataloga. U tom smeru će se Bibliotečki sistem SNTIS dalje razijati.

Poslednje poglavlje Monografije opisuje Rečnik podataka SNTIS. Rečnik podataka SNTIS koncipiran je tako da ima i standardnu "pasivnu" ulogu, ulogu "vodiča" kroz SNTIS i "aktivnu" ulogu preko koje se ostvaruje prostorna i implementaciona transparentnost celokupnog SPI, odnosno omogućava da se on tretira kao jedinstveni sistem, bez obzira na svoju veoma složenu strukturu.

Arhitektura Rečnika podataka je izvedena iz odgovarajućih ANSI i ISO standarda. Rečnik podataka je "informacioni sistem o informacionom sistemu", u ovom slučaju informacioni sistem o SNTIS. ANSI i ISO standardi predviđaju da se informacioni sistemi u rečniku podataka opisuju preko definisane četvoronivoske strukture korišćenjem Modela objekti-veze. Kako je Model objekti veze metodološka osnova za razvoj SPI, očigledno je da se modeli objekti-veze koji se formiraju u toku razvoja pojedinih lokalnih SPI, odnosno njihovih baza mogu iskoristiti za formiranje Rečnika podataka SNTIS. Ako se, još, za modeliranje podataka koristi jedinstveni CASE alat, kao što je u razvoju SNTIS bio slučaj, onda se Rečnik podataka može automatski formirati korišćenjem baze podataka CASE alata. Prvi deo Poglavlja 9 opisuje standardnu četvoronivosku strukturu Rečnika i ovakav način njegovog formiranja i održavanja.

Rečnik podataka SNTIS sadrži:

- Bazu podataka o bazama podataka u SNTIS, sa opisom njihovih opštih karakteristika (poreklo baze, predmet koji obrađuje, čvor u mreži na kome se nalazi, pravila korišćenja i slično) i sa detaljnim opisom njene strukture za pretraživanje;
- Bazu podataka čvorova mreže SNTIS, sa opisom fizičkih karakteristika čvora, njegove uloge u sistemu, organizacije koja čvorom upravlja i drugo.

Ove baze podataka, pored struktuirane verzije realizovane preko sistema za upravljanje bazom podataka db_VISTA, mogu biti struktuirane u formu sličnu formi "Blusheet-a" za opis baza podataka u DIALOG-u. Ovakva struktura je pogodna za pretraživanje korišćenjem Jedinstvenog sistema za pretraživanje.

Osnovna namena aktivnog dela Rečnika podataka je da obezbedi jedinstvenost pretraživanja na nivou celokupnog SNTIS, primenom opštег modela baze podataka i Jedinstvenog sistema za indeksiranje i pretraživanje. Rečnik podataka je takođe distribuirana baza podataka, sa lokalnim bazama, delimičnim replikacijama lokalnih baza na nekom kooperativnom nivou i centralnom nivou. Na lokalnom nivou čuvaju se opisi lokalnih baza podataka i pravila preslikavanja u SPI format. Na kooperativnom i centralnom nivou Rečnika podataka čuva se opis SPI formata svih lokalnih baza. Aktivna uloga na ovim nivoima odnosi se na procesiranje upita nad ovim opisima koje rezultuje u izbor baza u kojima se na dati upit može da dobije odgovor. Time se ostvaruje zahtevana transparentnost, korisnik može da dobije odgovore na svoj upit iz više različitih ili jedne distribuirane baze bez specificiranja ovih baza i njihove prostorne distribucije. Način procesiranja korisničkog upita nad Rečnikom i rutiranje upita prema odgovarajućim bazama, detaljno je opisana u ovom poglavlju.

Rečnik podataka je, kao što je rečeno, implementiran kao distribuirana baza. Centralni rečnik je struktuirana baza implementirana preko sistema za upravljanje bazom podataka db_VISTA. Uz ovu bazu u Centralnom rečniku nalazi se i baza sekundarnih SPI dokumenta ("Bluesheets") koja se indeksira i pretražuje pomoću Jedinstvenog sistema za pretraživanje i indeksiranje. Distribuirano pretraživanje se realizuje primenom Gopher i WordWideWeb Internet servisa, kako je to u Poglavlju 9. detaljno opisano.

Poglavlje 2.

KONCEPCIJA, SADAŠNJE STANJE I PRAVCI DALJEG RAZVOJA SISTEMA NAUČNIH I TEHNOLOŠKIH INFORMACIJA

2.1. UVOD [1,2]

U toku 1991.godine započete su u Srbiji značajne aktivnosti na razvoju novog koncepta sistema naučnih i tehnoloških informacija (SNTIS) i odgovarajuće mreže SNTIS. Ova mreža ima za cilj da poveže akademske i druge naučne, istraživačke i dokumentacione organizacije u jedinstveni sistem, a takođe i da omogući veze sa sličnim mrežama i sistemima u svetu. Osnovni zadatak razvoja SNTIS je da se za potrebe istraživača i stručnjaka različitih profila, odnosno za potrebe odgovarajućih organizacija, razvije neophodan infrastrukturni sistem za međusobnu komunikaciju, za razmenu poruka i podataka, a takođe i za stvaranje i korišćenje nacionalnih i internacionalnih sistema baza podataka koje su potrebne za podršku naučnog, istraživačkog i razvojnog rada. Smernice za razvoj ovog tipa infrastrukture nauke Srbije date su u "Koncepciji razvoja sistema naučnih i tehnoloških informacija Srbije", koja je završena sredinom 1991. godine. Koncepcijom i kasnije izrađenim planovima i projektima definisano je više fizičkih i logičkih mreža SNTIS, koje će biti detaljno opisane u daljem tekstu.

Realizacija SNTI Srbije i Jugoslavije sprovodi se po fazama. Međutim, SNTIS je praktično od samog početka koordiniranog razvoja bio operativan, mada sa relativno skromnim performansama koje su diktirane faznom izgradnjom infrastrukture i raspoloživim servisima. Sa druge strane, ne gubeći ništa na operativnosti, SNTI će se i dalje intenzivno dopunjavati i razvijati. Ovaj razvoj time samo prati moderne tendencije koje su prisutne i u svetu, u kome upravo razvoj distribuiranih informacionih sistema i računarskih odnosno telekomunikacionih mreža beleži izuzetno brzi rast.

U ovom poglavlju prikazano je trenutno stanje razvoja i postignuti rezultati, odnosno izloženi su koncepti i planovi daljeg razvoja SNTIS.

2.2. OSNOVNA OPREDELJENJA U RAZVOJU SNTI

2.2.1. SNTI Srbije i Jugoslavije

U periodu do 1990.godine postojao je koncept razvoja Sistema naučno-tehnoloških informacija SFR Jugoslavije, baziran na homogenoj mreži VAX-Digital računara i na centralizovanom sistemu baza podataka ATLOSS, koji je delimično realizovan u bibliotečkom okruženju. Jugoslavija je do sankcija bila član EARN-a, evropske asocijacije istraživačkih mreža. Članovi jugoslovenskog dela EARN-a bili su jugoslovenski univerziteti, sa Univerzitetom u Beogradu kao koordinatorom. Sa nastupanjem sankcija zamrznut je status SR Jugoslavije u EARN-u i fizički prekinuta veza tadašnje SNTI mreže sa EARN-om, čime je prekinut i jedini tada postojeći servis elektronske pošte.

U periodu posle 1990. godine došlo je do velikih promena. U Republici Srbiji je 1991. godine usvojena nova koncepcija razvoja SNTIS. Na osnovu ove nove koncepcije, za manje od četiri godine, razvijena je potpuno nova distribuirana TCP/IP mreža SNTIS, koja je na nivou srednje razvijenih mreža u drugim evropskim zemljama. Takođe, razvijeni su novi, u priličnoj meri originalni distribuirani servisi, koji su takođe potpuno kompatibilni sa Internet servisima. Razvoj i instalacija novih sadržaja SNTIS ide nešto sporije, ali se i u toj oblasti beleže značajni uspesi.

Sredinom 1994. dobijeno je zvanično pravo međunarodnog administriranja tzv. YU-domena u okviru globalne mreže Internet, čime se Jugoslavija izjednačila sa drugim razvijenim zemljama. Posredno priključenje u Internet moguće je i koristi se odavno, a neposredno priključenje se očekuje u neposrednoj budućnosti. Krajem 1995.godine dobijeno je i pravo lokalnog dodeljivanja tzv. IP adresa, čime je načinjen još jedan korak prema konačnom uključenju u svetske mreže.

Jugoslavija je i dalje član zajednice evropskih istraživačkih mreža TERENA koja je nasledila raniji EARN, ali trenutno bez mogućnosti direktnog priključenja.

2.2.2. Ciljevi [1,2]

SNTIS je deo informacione infrastrukture Srbije i Jugoslavije. Osnovni ciljevi i zadaci SNTIS definisani su u dokumentu "Koncepcija razvoja SNTI u Republici Srbiji što podrazumeva sledeće aktivnosti:

- ostvarivanje mesta Srbije u međunarodnoj informacionoj i komunikacionoj infrastrukturi;
- formiranje nacionalnog sistema u oblasti naučno-tehnoloških informacija i računarskih telekomunikacija;
- projektovanje, izgradnja i eksploatacija mreža za prenos i mreže korisnika naučno-tehnoloških informacija, i njihova integracija u slične mreže na međunarodnom planu;
- omogućavanje pristupa korisnicima do inostranih informacionih resursa, sa što manje posrednika i na što efikasniji način;
- razvoj domaćih izvora naučno-tehnoloških informacija i znanja;
- obezbeđivanje standarda u izgradnji i funkcionisanju SNTI;
- obrazovanje kadrova za razvoj i korišćenje SNTI.

2.2.3. Obuhvat sistema

Da bi se ostvarili osnovni ciljevi razvoja, Sistem naučno-tehnoloških informacija Srbije je planiran kao otvoren sistem u sadržajnom, institucionalnom, tehnološkom, strukturnom i tržišnom smislu.

Sadržajno, za razliku od SNTI koji je razvijan u periodu do 1990 godine, i koji je bio prvenstveno usmeren na konvencionalne naučno-tehnološke informacije, bibliografske baze podataka koje obuhvataju Bibliotečki informacioni sistem i skup unapred definisanih specijalizovanih sistema, SNTI se danas gradi prema realnim potrebama korisnika. Pored potreba za korišćenjem faktografskih i "full text" baza podataka, uz orientaciju na korišćenje mreža, daje se prednost drugim vidovima komunikacija, korišćenju elektronske pošte, elektronskih časopisa, elektronskih konferencijskih foruma i tribina, softvera iz domena, udaljenih računarskih i informacionih resursa. Isto tako, SNTIS je trebalo u znatno većoj meri usmeriti prema primenjenim i razvojnim istraživanjima.

Institucionalno, SNTIS ne treba ograničavati unapred definisanom strukturu (na primer unapred definisanim skupom institucija, samo jednim informacionim servisom - hostom, unapred definisanim skupom specijalizovanih centara i sistema, i slično). Naprotiv, SNTI treba da bude otvoren sistem tako da svaka institucija može postati njegov deo, odnosno

da može sama, u skladu sa opštom koncepcijom i standardima, definisati svoju ulogu u njemu.

Tehnološki, SNTIS treba razvijati kao heterogeni distribuirani informacioni sistem čije se jedinstvo ostvaruje preko minimuma neophodnih standarda koje svaki deo sistema u svom razvoju mora da poštije, uz stalno pracenje i implementiranje novih tehnologija u ovoj oblasti.

Strukturno, SNTI je otvoreni sistem više institucija, lokalnih mreža i podistema - sa zajedničkom mrežom SNTI, oslonjenom na javni PTT sistem.

Tržišno, složeni sistem kao što je SNTI mora imati određenu komercijalizaciju pojedinih servisa, sa pokrivanjem troškova srazmerno učešću.

Razvoj SNTIS treba, u najvećoj mogućoj meri, da bude vođen stvarnim potrebama korisnika. SNTIS treba da se organski razvija, na osnovu neposrednih potreba korisnika koji samostalno razvijaju svoje informacione sisteme, otvaraju ih, uključuju u pojedine podsisteme i međusobno povezuju, na bazi definisanih standarda. Uloga države treba da bude prvenstveno u tome da obezbedi da se SNTIS razvija kao deo informacione infrastrukture Srbije, Jugoslavije i sveta.

2.3. STRUKTURA I SADRŽAJ SNTIS

SNTIS je infrastrukturni sistem namenjen potrebama naučnoistraživačkih radnika i privrede koji ima zadatak da obezbedi međusobnu računarsku komunikaciju, razmenu poruka i podataka, pristup i korišćenje baza podataka i mrežnih servisa, korišćenje računarskih resursa u mreži. SNTIS je, takođe, otvoreni distribuirani informacioni sistem koji se sastoji od skupa institucija koje se, na osnovu zajedničkog interesa za razvoj i korišćenje informacionih sistema u oblasti nauke, tehnologije i poslovanja, povezuju u podsisteme (logičke mreže). Logičke mreže se implementiraju i međusobno povezuju preko razvijene tehnološke infrastrukture, mreže SNTI sa realizovanim i raznovrsnim informacionim servisima. Vrednost informacionog sadržaja i efektivnost korišćenja celokupnog sistema isključivo zavisi od razvoja pojedinih logičkih mreža, dok mreža SNTI treba da obezbedi jednostavno i efikasno korišćenje tog sadržaja. Jedinstvo sistema se ostvaruje:

- (a) Povezivanjem podistema tako da korisnici, pretežno vezani za jedan podistem, mogu jednostavno da koriste i druge podisteme;

- (b) Razvojem sistema na bazi jedinstvenih standarda; i
- (c) Globalnim upravljanjem, koje treba da eliminiše eventualne negativne efekte autonomnog razvoja pojedinih podsistema.

Imajući u vidu osnovna opredeljenja, SNTIS predstavlja deo naučno-tehnološke infrastrukture zemlje i deo informacione infrastrukture Evrope i sveta. Razvija se kao distribuirani heterogeni informacioni sistem, sa velikim stepenom lokalne autonomije, ali i sa neophodnim stepenom koordinacije razvoja sa državnog nivoa.

Konceptom daljeg razvoja, SNTIS treba da se organizuje kao sistem čiji su osnovni, konstitutivni, elementi:

- (a) Institucionalne jedinice koje su stvaraoci i korisnici informacione infrastrukture - naučnoistraživačke, istraživačko-razvojne i visokoobrazovne organizacije, privredne i uslužne organizacije, ostale organizacije i institucije, odnosno pojedinci;
- (b) Institucionalne jedinice koje su po osnovnoj delatnosti organizatori ili korisnici informacionih fondova (biblioteke, specijalizovani centri, INDOK službe i centri, arhivi i muzeji);
- (c) Analitički informacioni centri, koji imaju posebnu ulogu u razvoju, efikasnom i efektivnom korišćenju SNTIS;
- (d) Upravljačke i monitoring institucije, koje obezbeđuju praćenje razvoja, evaluaciju korišćenja sistema, stabilno finansiranje, odnosno realizuju ulogu države u razvoju SNTIS.

U osnovi mogu se posmatrati tri organizaciona bloka u okviru SNTI:

- a) SNTI mreža u užem smislu, koja sa jedne strane povezuje različite korisnike, a sa druge strane obuhvata i u sebi sadrži raspoložive servisne usluge i programski sistem za pretraživanje i upravljanje;
- b) Logičke mreže različitih namena, pri čemu se i sama SNTI mreža u širem smislu može posmatrati kao jedna od logičkih mreža; i
- c) Dokumentacioni, informacioni i programski sadržaji, domaći i inostrani.

Može se očekivati da će se u budućnosti spontano pojavljivati veći broj različitih mreža od interesa za SNTIS. U skladu sa osnovnim opredeljenjima dosadašnjeg razvoja SNTIS i iskustava u razvoju pojedinih specifičnih informacionih infrastruktura u Evropi i svetu, razvijaju se:

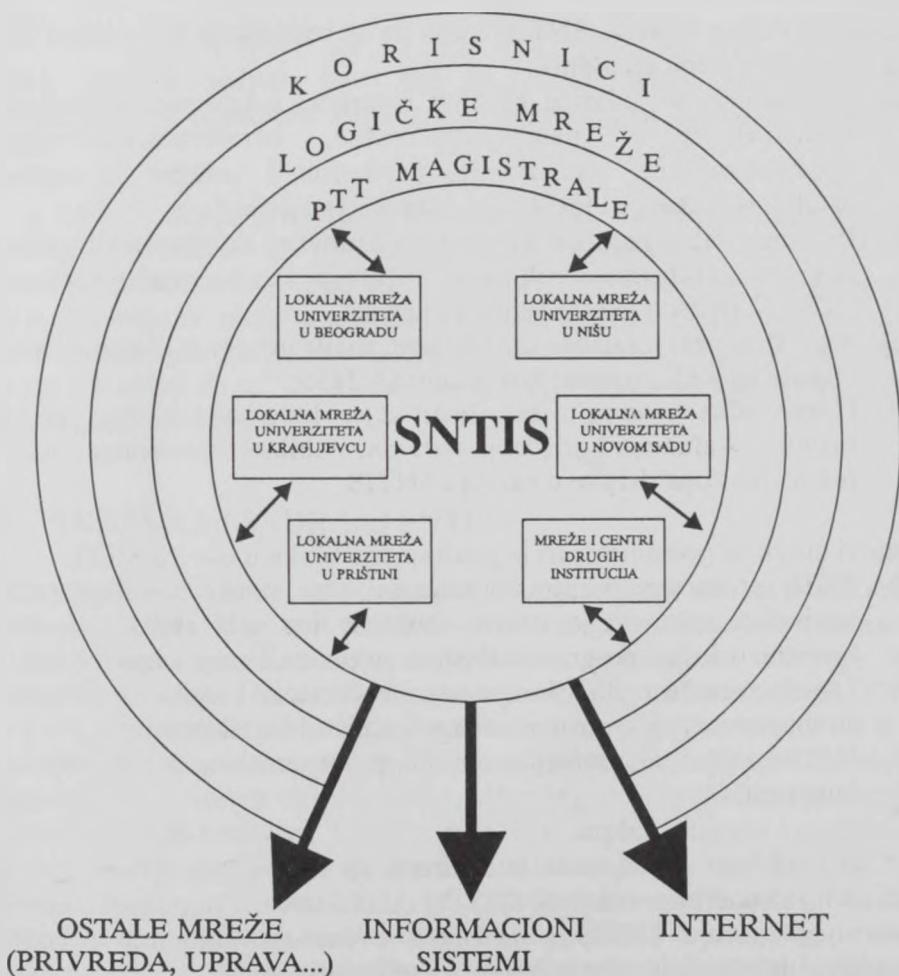
- (a) SNTI mreža (WAN - Wide Area Network) koja obezbeđuje izlaz u svet i povezuje Beograd, Niš, Novi Sad, Kragujevac i Priština, kao i gradske odnosno lokalne mreže u okviru ovih centara, ova mreža u

isto vreme u logičkom smislu predstavlja mrežu za neposrednu komunikaciju i transfer;

- (b) više logičkih mreža.

Osnovu fizičkog funkcionisanja SNTIS, na kojoj se pretpostavlja i dalji razvoj SNTIS, čini univerzitetska i naučnoistraživačka mreža SNTI.

Na slici 2.1 dat je prikaz opšte arhitekture SNTI.



Slika 2.1. Prikaz opšte arhitekture SNTI

2.4. KONCEPCIJA, STANJE I PLANOVI RAZVOJA MREŽE SNTI

2.4.1. Nacelna opredeljenja

Mreža Sistema naučno-tehnoloških informacija Srbije (SNTIS) obezbeđuje povezivanje računskih centara, lokalnih mreža, računara i terminalne opreme pojedinih učesnika sa područja Srbije i SR Jugoslavije, kako međusobno, tako i sa odgovarajućim partnerima u inostranstvu. Učesnici u Srbiji su, prvenstveno, univerzitetski centri i članice univerziteta u Beogradu, Nisu, Novom Sadu, Kragujevcu i Prištini, zatim Srpska akademija nauka i umetnosti (SANU), naučni instituti izvan univerziteta, nacionalne i regionalne biblioteke i drugi potencijalni korisnici. Kao partneri koji su ravnopravni univerzitetima iz Srbije, javljaju se Univerzitet i druge naučne institucije iz Podgorice odnosno iz Crne Gore, i, eventualno, zainteresovane univerzitetske i naučne institucije sa još nekim područja bivše Jugoslavije.

U okvirima pojedinih univerziteta, mrežom su obuhvaćene ili treba da budu obuhvaćene sve visokoškolske, istraživačke i servisne organizacije. Osnovna ideja je da sve te institucije, preko svojih računskih centara, lokalnih mreža ili direktno, imaju pristupe do odgovarajućih računarskih čvorova svojih univerziteta, od kojih bi dalje bili povezani sa ostalim delom SNTI mreže.

Koncepcija otvorenog heterogenog sistema, definisana 1991. godine, neprestano se dalje nadgrađuje i konkretizuje, pri čemu opšti trendovi tehnološkog razvoja mreže SNTIS značajno odstupaju od trendova ranijeg razvoja starog sistema NTI SFRJ. Posebne promene u odnosu na prethodnu koncepciju su:

1. Mogućnost povezivanja različitih tipova mreža uz korišćenje zajedničke infrastrukture;
2. Mogućnost povezivanja u mrežu različite vrste računarske opreme, kako personalnih računara i specijalizovanih radnih stаница које širom sveta proizvode nezavisni manji proizvođači, tako i opreme velikih svetskih proizvođača као што су IBM, DEC, HP, SUN, Silicon Graphics, Bull, itd.;
3. Mogućnost korišćenja više različitih operativnih sistema, на првом mestu UNIX-a, DOS-a, VMS-a, а поред тога свакако и savremenih grafički orijentisanih okruženja типа WINDOWS-a;
4. Primena moderne tehnologije i velikih protoka podataka;
5. Primena više standardnih komunikacionih modela, на првом mestu TCP/IP, ali i OSI, DECnet, itd.;



6. Princip da se mreža ne zasniva na povezivanju pojedinih računara, nego na povezivanju lokalnih, mesnih i regionalnih mreža, uz korišćenje više standarda, na primer IEEE 802.3, IEEE 802.4 i IEEE 802.5.

Polazeći od izloženih načelnih stavova, može se konstatovati da ovako planirana mreža SNTI ima sledeće segmente:

- **Interne mreže pojedinačnih korisnika** (koje se mogu sastojati od samo nekoliko pa do više stotina računara različitih tipova);
- **Lokalne mreže grupa korisnika**, koje obuhvataju interne mreže više institucija lociranih na istoj ili na bliskim lokacijama;
- **Univerzitetske, mesne i regionalne mreže**, koje obezbeđuju povezivanje svih članova jednog univerziteta, svih institucija iz jednog grada ili jednog regiona, na odgovarajuće univerzitske ili druge čvorne centre;
- **Lokalne, univerzitske, mesne i međumesne telekomunikacione magistrale**, kojima se povezuju pojedine institucije, odnosno univerzitski i drugi čvorni centri iz svih gradova koji imaju svoje mesne mreže;
- **Međunarodne telekomunikacione magistrale**, koje služe za veze sa inostranim partnerima i međunarodnim telekomunikacionim mrežama,,

čime je zapravo formirana

- SNTI mreža Srbije, odnosno SNTI mreža Jugoslavije.

Naravno, prilikom praktičnih realizacija, nije uvek moguće oštro razdvojiti sve ove segmente, pošto se oni često prepliću odnosno preklapaju.

Međunarodne i međumesne telekomunikacione magistrale mreže SNTI realizuju se posredstvom transmisionih kapaciteta javne PTT mreže, za sada sa protocima od 19.2, 28.8 ili 64Kb/s. Paralelno sa stalnom modernizacijom i razvojem sistema prenosa PTT, u toku su i aktivnosti za ostvarenje magistralnih veza sa znatno većim protocima, na primer 128, 256, 512Kb/s odnosno 2Mb/s, što je "de facto" standard za telekomunikacione magistrale računarskih mreža u Evropi.

Mesni segmenti mreže SNTIS realizuju se u više varijanti, zavisno od mogućnosti i potreba pojedinih učesnika. Načelno, može se reći da postoje

magistralne veze, koje povezuju veće grupacije institucija, međusobno i sa centrima, i pojedinačne veze, kojima se povezuju prostorno izdvojene institucije. Za magistralne mesne veze koriste se protoci reda 10 ili 100Mb/s, FDDI ili ETHERNET tipa. Tamo gde to nije moguće, mogu se kao magistrale privremeno prihvati i 28.8Kb/s asinhrono odnosno 64 ili 128Kb/s sinhrono veze. Tipični primeri su mesne mreže Beograda i Niša, koji će kasnije biti detaljnije opisani.

Lokalni segmenti mreže realizuju se kao mreže FDDI, ATM, TOKEN RING ili ETHERNET tipa, sa odgovarajućim protocima od 100, 16 ili 10Mb/s, pri čemu su FDDI, ATM, TOKEN RING i ETHERNET oznake za različite standarde povezivanja i prenosa podataka u mrežama.

Veze izdvojenih korisnika sa odgovarajućim centrima ili čvornim tačkama u okviru jednog mesta ili manjeg regiona ostvaruju se u skladu sa mogućnostima i potrebama, do sada najčešće sa manjim protocima, preko javne JUPAK mreže za prenos podataka ili posebnim vezama. Ubuduće će se ove veze praktično uvek ostvarivati iznajmljenim vezama sa protocima od 28.8Kb/s asinhrono na rastojanjima većim od 10 km, odnosno 64 ili 128Kb/s sinhrono na manjim rastojanjima.

Navedeni tokovi prenosa i vrednosti protoka određuju i vrste odgovarajućih sredstava veze. U međunarodnom i međumesnom segmentu mreže koristi se PTT sistem, sa vezama kakve postoje na datim pravcima. U mesnom segmentu takođe se dominantno koriste veze PTT, a samo izuzetno sopstvene veze. Za velike protoke koriste se optički ili koaksijalni kablovski vodovi, za niže vrednosti protoka simetrične parice ili četvorke. U lokalnim segmentima koriste se takođe optički ili koaksijalni vodovi za velike protoke, a simetrični kablovi za manje.

U međunarodnom i međumesnom segmentu korišćenje PTT sredstava zasniva se na zakupu potrebnih kapaciteta, dok se u mesnom segmentu koriste i druga rešenja (na primer participacija pri izgradnji, realizacija privatnih delova mreže, itd.).

Izloženi principi i osnovna opredeljenja u skladu su sa usvojenom Koncepcijom razvoja SNTIS i sa izrađenim planovima razvoja SNTIS i pojedinih univerzitetskih centara.

2.4.2. Međunarodni i međumesni segmenti mreže SNTIS [2]

Za realizaciju međunarodnog i međumesnog segmenta mreže SNTI Srbije neophodna je svakako saradnja sa Javnim preduzećem PTT saobraćaja "SRBIJA", odnosno jedino logično i prihvatljivo rešenje je da se koriste njihovi transmisioni putevi i, po potrebi, usluge odgovarajućih komutacionih čvorista.

Korišćenje usluga PTT "SRBIJA" obavezuje korisnike da se, u određenoj meri, prilagode organizaciji i strukturi PTT mreže. Na prvom mestu, treba istaći da ceo međunarodni saobraćaj mora da se odvija preko međunarodne centrale, koja se nalazi u Beogradu, u sklopu objekta "TK centar". Prema tome, svi učesnici koji žele da komuniciraju sa nekim partnerom iz inostranstva, moraju da proslede svoje signale kroz Beograd. Slično tome, tokovi saobraćaja SNTI u međumesnoj mreži moraju biti prilagođeni organizaciji javne telefonske mreže, odnosno njenoj hijerarhijskoj strukturi.

Za mrežu SNTI pogodno je što su univerzitetski centri locirani u gradovima u kojima su istovremeno i sedišta telefonskih centrala visokog ranga, tranzitnih i glavnih, što dozvoljava relativno lako uključenje u sisteme prenosa PTT. Takođe je značajno što je upravo na pravcima između Beograda i tih gradova u toku ubrzana izgradnja savremenih digitalnih transmisionih puteva, zasnovanih na korišćenju optičkih kablova.

U prethodnom tekstu specificirano je da se u međunarodnom i međumesnom segmentu predviđa prenos sa protokom od najmanje 64Kb/s. U digitalnoj mreži taj se zahtev lako zadovoljava jer se time zauzima samo jedan telefonski kanal. U analognim mrežama se za takav prenos mora angažovati cela primarna grupa od 12 kanala (CCITT, Rec. V 36), što je znatno nepovoljnije. Međutim, kako se i u međunarodnim i u međumesnim vezama intenzivno prelazi na digitalni prenos, to su rešenja sa analognim uređajima samo privremena.

Polazeći od izloženih stavova, u daljem tekstu su dati elementi konkretnih rešenja međunarodnog i međumesnih segmenata.

Za realizaciju mreže usvaja se zvezdasta struktura, sa Beogradom kao centralnom tačkom. Iz magistralnog dela mreže BEONET do međunarodne i međumesne automatske telefonske centrale (ATC) u TK centru treba da se prosledi više kanala za različite binarne protoke od najmanje 64Kb/s do 2Mb/s. Od ovih kanala jedan ili više se vode u međunarodnu centralu, za vezu sa inostranim partnerima na pravcu Budumpešta - Zapadna Evropa i eventualno Solun - Atina, a ostali u međumesnu centralu, za veze sa

centrima u Nišu, Novom Sadu, Kragujevcu, Prištini i Podgorici. Pored ovih obaveznih kanala, po potrebi se mogu realizovati slične veze prema Skoplju, prema centrima u BIH, međunarodne veze prema Rumuniji, Italiji i Bugarskoj, javne i "privatne" satelitske veze, itd. S obzirom na ranije iskustvo lakog prekidanja veze kada je u pitanju samo jedan izlaz prema međunarodnim mrežama, kao što se desilo u slučaju veze sa evropskim delom EARN mreže sa početkom primene sankcija, više istovremenih alternativnih izlaza prema Internetu i drugim mrežama izgleda razuman, mada skup izbor.

Za ozbiljan budući rad u Internet mreži neophodni su značajni protoci i digitalne magistralne veze. Kao što je rečeno, standardni protok za magistralno povezivanje je u Evropi 2Mb/s ili više, ali se za početak mogu koristiti i nešto manji protoci. Jedini jugoslovenski digitalni izlazi u evropske telekomunikacione tokove su u ovom trenutku sistem sa protokom od 144Mb/s prema Budimpešti i sličan, delimično opremljen sistem prema Grčkoj. U sistemu prema Grčkoj je praktično osposobljeno samo 2Mb/s od Beograda do Atine, sa tendencijom daljeg brzog povećanja kapaciteta. Takođe, veoma brzo će biti osposobljeni novi dopunski digitalni sistemi prema Budimpešti, prema kojoj je položen višeparični optički kabal, a radi se i na osposobljavanju digitalnih sistema preko Crne Gore do Italije. Realno je očekivati da će jedan od ovih izlaza, verovatno onaj preko Budimpešte, biti korišćen i za realizaciju Internet veze. Treba ipak navesti da dodatnu teškoću predstavlja i to što JP PTTS "Srbija" može da omogući digitalni ulaz u mrežu ili samo sa 64Kb/s ili 2Mb/s (32 digitalna kanala). Za sve druge protoke između ove dve granične vrednosti potrebno je obezbediti dodatno multipleksiranje i demultipleksiranje.

Moguće je naravno da se za internacionalni priključak u Internet mrežu koriste i satelitske veze, što predstavlja jedno od standardnih svetskih rešenja. Ipak, sa vezama preko satelita treba uzeti u obzir postojanje relativno velikog kašnjenja na trasi, što donekle otežava primenu budućih "on line" multimedijalnih dvostranih servisa, tako da je za glavni izlaz pogodnije koristiti zemaljske digitalne veze kada god je to moguće.

Predložena konfiguracija obavezuje učesnike van Beograda da u svim komunikacijama u okviru SNTIS, između sebe i sa inostranstvom, prolaze kroz centar u Beogradu. Ovo rešenje na prvi pogled može da izgleda kao nepotrebno komplikovano, ali u prilog prethodnog stava može se izneti više značajnih argumenata:

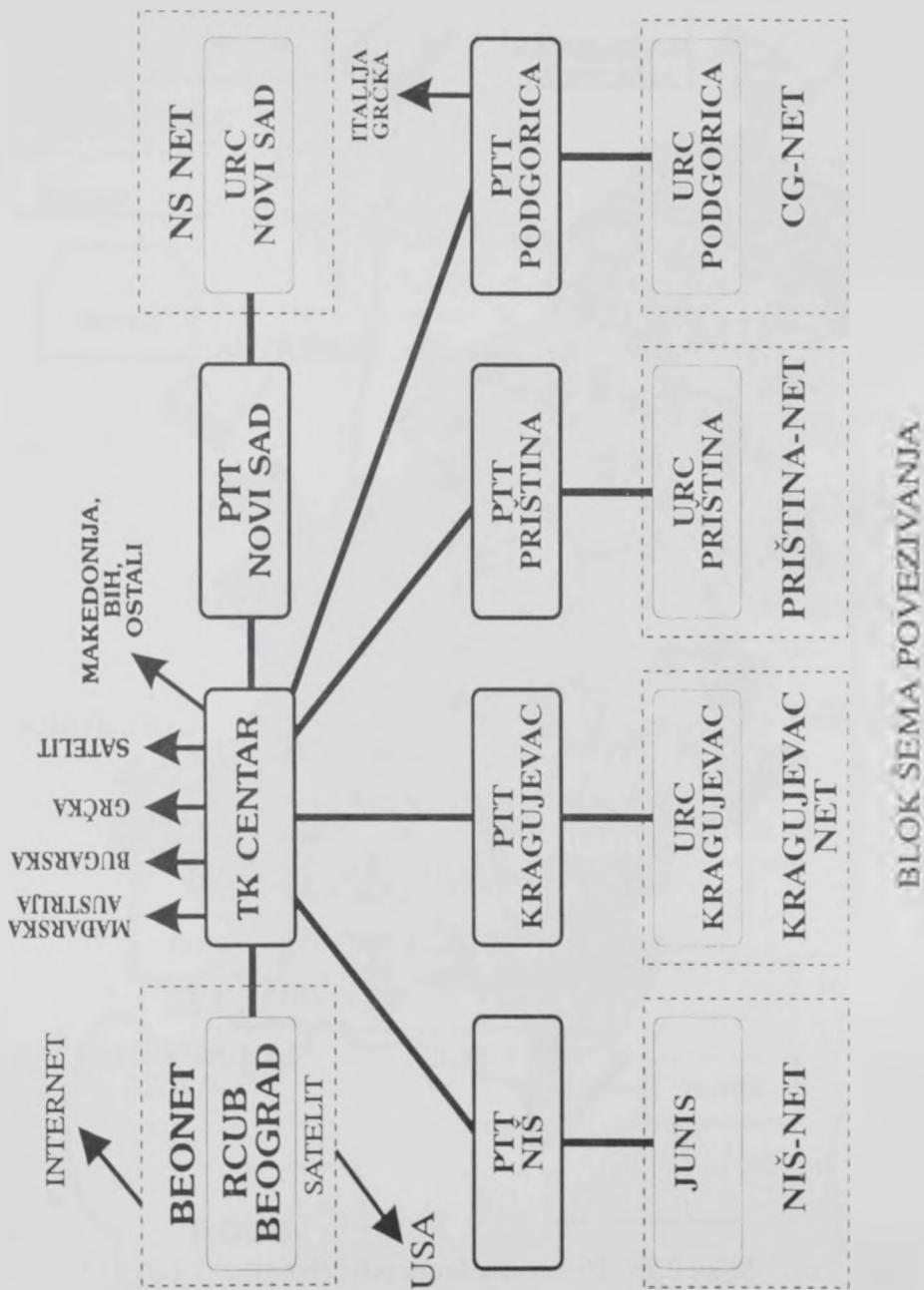
- predložena strukturalna organizacija najbolje je prilagođena PTT sistemu;

- usvajanjem predložene organizacije minimizirani su zahtevi za dodatnim uslugama PTT, odnosno u istoj meri bi se smanjili i troškovi, koji svakako nisu zanemarljivi;
- najveći saobraćaj u nacionalnom delu mreže, bar u početnom periodu, očekuje se upravo na relacijama između centra u Beogradu i centara u drugim mestima;
- obezbeđuju se jedinstveni izlazi prema stranim partnerima, što je pogodno i sa administrativnog i sa materijalno-finansijskog stanovišta;
- opredeljenje na vise izlaza prema međunarodnim mrežama je rezultat ranijih nepovoljnih iskustava da se samo jedan izlaz može veoma lako blokirati, a sa druge strane to eventualno omogucava podešavanje balansa saobraćaja i obezbeđuje neophodne alternativne puteve u slučajevima otkaza;
- telekomunikaciona zvezdasta struktura mreže ni na koji nacin ne sugerise niti podrazumeva niži logički hijerarhijski nivo centara van Beograda, pošto je to pitanje koje se rešava upravljačkom strukturom mreže.

U okviru međunarodnog i međumesnih segmenata mreže potrebno je da se reše veze između univerzitetskih centara u svim gradovima i odgovarajućih PTT centrala. U svim mestima osim Beograda (Niš, Novi Sad, Kragujevac, Priština), u kojima postoji ili se očekuje prostorna koncentracija odnosno usmeravanje svih korisnika prema odgovarajućim univerzitetским centrima, ova rešenja se formiraju u saradnji sa lokalnim nadležnim PTT službama. Po pravilu, polažu se posebni optički kablovi i preko odgovarajućih telekomunikacionih uređaja povezuju se računski centri sa odgovarajućim PTT centrima, kao što je realizovano u Nišu.

U izvesnom smislu izuzetak od izloženog koncepta predstavlja Beograd, kako zbog broja učesnika, tako i zbog izuzetne prostorne razučenosti mreže. Pošto se u okviru mesne mreže planiraju značajni radovi, pokazalo se da je moguće i pitanje povezivanja mesnog odnosno međunarodnog segmenta rešiti u sklopu mesnog segmeta.

Opisana koncepcija međunarodnog i međumesnih segmenata mreže SNTI prikazana je u organizacionom i prostornom smislu na slikama 2.2a i 2.2b. Istovremeno, koncepciona organizacija mreže u tehničkom smislu prikazana je na slici 2.2c. Šema sa slike 2.2c ne gubi na generalnosti bez obzira koje se brzine protoka i koja se tehnološka rešenja koriste za povezivanje, sto naravno zavisi i od stvarnih mogućnosti PTT mreže i od raspoloživih finansijskih sredstava.

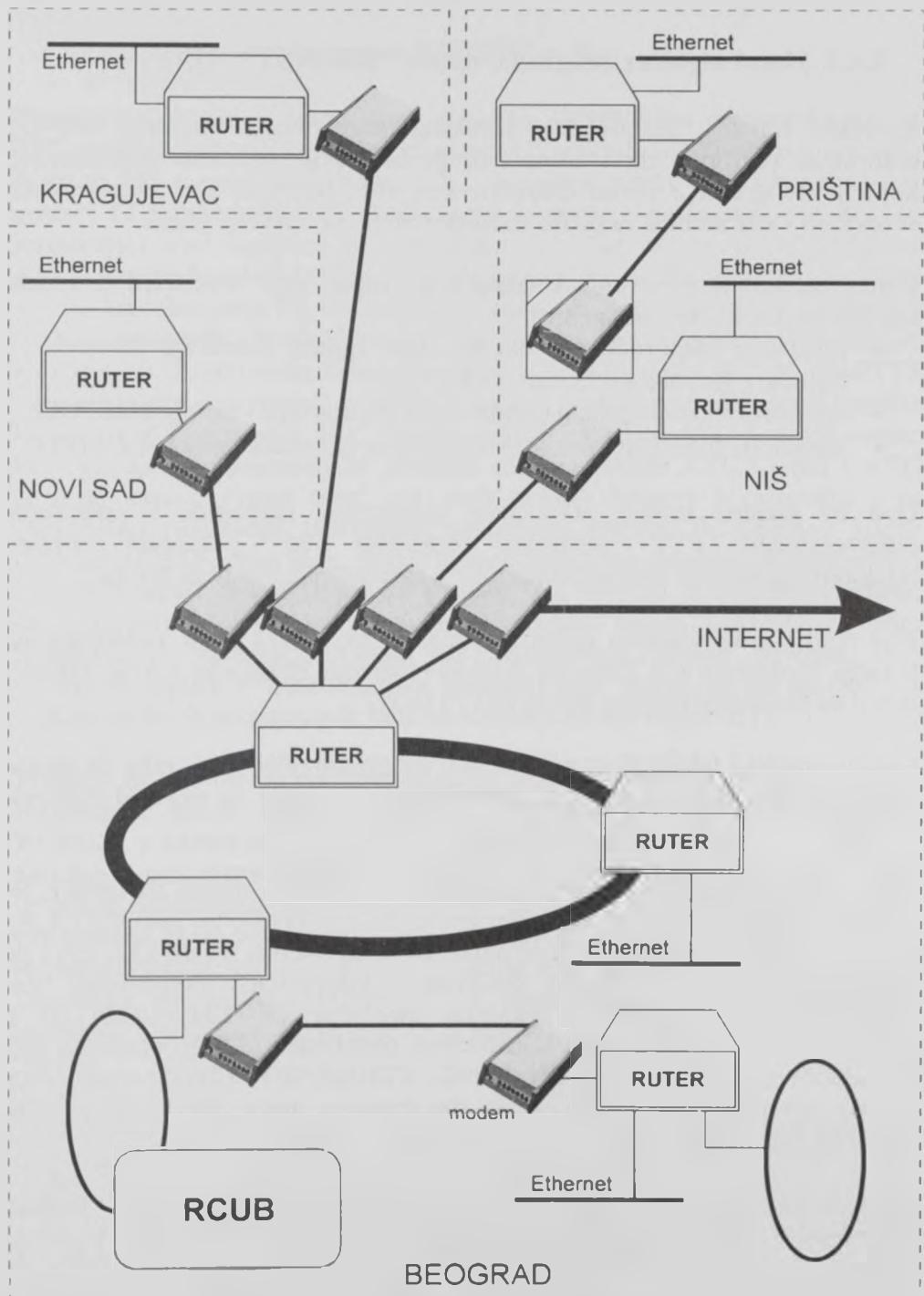


BLOK ŠEMA POVEZIVANJA

Slika 2-2a: Organizaciona šema povezivanja



Slika 2.2b. Prostorna sema povezivanja



Slika 2.2c. Tehnička šema povezivanja

2.4.3. Mesni segment Beograda, mreža "BEONET" [3,4]

Korisnici mreže "BEONET", fakulteti, registrovani instituti, SANU, biblioteke, rektorati univerziteta i druge institucije odnosno organizacije, kojih u Beogradu u ovom trenutku ima više desetina, a u kasnijoj fazi verovatno i više stotina, različito su raspoređeni na teritoriji grada.

Postoji nekoliko prostorno kompaktnih grupa, koje formiraju posebne lokalne međuinstitucionalne mreže:

- Institucije raspoređene u okolini stare zgrade Tehničkih fakulteta u Bulevaru Revolucije 73 - mreža LOTEH;
- Institucije raspoređene u okolini Studentskog trga - mreža LOSTUD;
- Institucije u okolini lokacije Medicinskog fakulteta - mreža LOMED.

Ove tri lokalne mreže, zbog svog prostornog rasporeda i rasporeda odgovarajućih PTT centrala, povezuju se posebnim brzim telekomunikacionim magistralama, što je razmotreno u daljem tekstu.

Kao relativno kompaktne grupacije mogu se smatrati i grupe institucija na lokaciji Voždovac i u Zemunu odnosno Novom Beogradu - koje takođe mogu da formiraju lokalne mreže LOV i LOZ.

Ostali instituti i fakulteti su međusobno prostorno udaljeni i treba da imaju svoje zasebne segmente koji će direktno biti povezani na neki od čvorova mreže.

Jedno od važnih čvorišta mesnog segmenta Beograda - mreže BEONET, je Računski centar Univerziteta u Beogradu, RCUB (Poslovni centar LOLA, Bulevar Revolucije 84). RCUB se nalazi u blizini lokalne mreže LOTEH i sa njom je višestruko povezan optičkim i višeparičnim kablovima. Za komuniciranje RCUB-a sa lokalnim mrežama LOTEH, LOSTUD i LOMED, odnosno sa drugim korisnicima, obezbeđen je ili se predviđa rad sa velikim protocima (TOKEN RING, ETHERNET, FDDI, eventualno ATM), što podrazumeva da se na tim trasama mora obvezediti prenos optičkim kablovima.

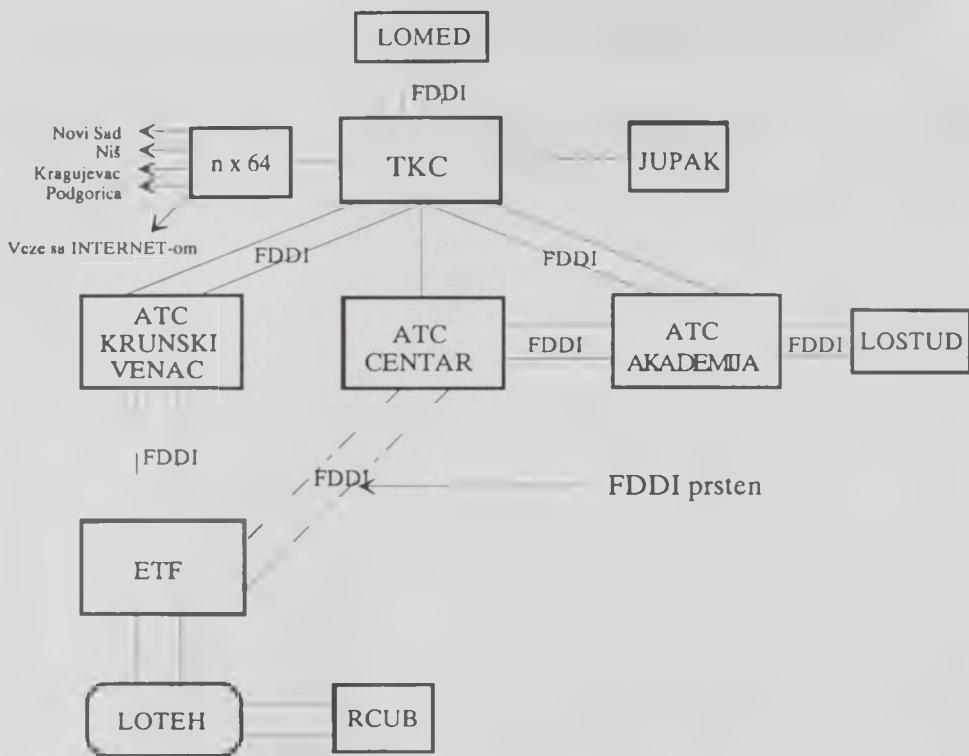
Pobrojanim resenjima nisu isključene i druge alternative, na primer povezivanje pojedinih zainteresovanih institucija tzv. poprečnim vezama, formiranje i drugih lokalnih mreža više institucija, itd., ali ova resenja su prepustena potrebama i mogućnostima samih učesnika, pod uslovom da ne narusavaju osnovnu zajedničku koncepciju.

2.4.3.1. Magistralni prsten BEONET-a [5]

Osnovu magistralnog povezivanja u okviru BEONET-a predstavlja mreža PTT optičkih kablova kojom su gradske AT centrale međusobno povezane, ili će to biti u najskorijoj budućnosti. Izgradnja mreže optičkih kablova u mesnoj TT mreži deo je šire akcije za njenu modernizaciju, u sklopu koje se instaliraju i nove digitalne telefonske centrale. Planirano je da sve digitalne ATC budu međusobno povezane optičkim kablovima. Za sada, takvi kablovi postoje na relacijama TK centar-ATC Centar, ATC Centar-ATC Akademija i TK centar-ATC Krunski venac. Na osnovu svojih potreba i u sklopu saradnje sa Elektrotehničkim fakultetom u Beogradu (ETF), JP PTTS "Srbija" (Sektor za razvoj) odlučilo je da kao svoju investiciju položi optičke kablove koji bi povezali ATC Centar sa ETF i ETF sa ATC Krunski venac. Položeni kablovi primarno bi poslužili za povezivanje ATC Centar i ATC Krunski venac. Pored toga, ovi novi optički kablovi bi posluzili i za eventualnu realizaciju izdvojenog digitalnog stepena ili digitalne kućne centrale na ETF, čime bi se, pored rešenja problema telefonskih priključaka tehničkih fakulteta koji se nalaze u blizini ove lokacije, bolje rešilo i pitanje ulaza iz javne telefonske mreže u računarsku mrežu LOTEH. U svim položenim i budućim optičkim PTT kablovima postoji dovoljan broj slobodnih optičkih monomodnih vlakana koja mogu da se koriste za realizaciju brzih magistralnih veza akademske mreže BEONET.

Polazeći od navedenih činjenica moguće je dati koncepciju i osnovnu šemu magistralne mreže, čiji bi se terminali nalazili u Rektoratu Univerziteta, RCBU-u, u okviru mreže LOTEH, u okviru mreže LOMED i u TK centru, kao što je prikazano na slici 2.3. Pošto će navedeni terminali biti povezani optičkim kablovima, to će biti mogući protoci od 100Mb/s i više. Za sada, predviđa se da se ova magistralna mreža realizuje kao tzv. FDDI mreža, što je preovlađujuće rešenje u Evropi. Kasnije, nisu isključena ni druga resenja. Dinamika realizacije FDDI prstena zavisi uglavnom od raspoloživih sredstava i dinamike realizacije dela prstena koji treba da zavrssi JP PTTS "Srbija".

Kao što je već izloženo, u TK centru PTT nalaze se međumesna i međunarodna centrala. Stoga se predviđa da se preko tog terminala FDDI realizuju sve međumesne i međunarodne magistralne veze, kroz kapacitete PTT sistema, odnosno da to istovremeno bude i priključna tačka buduće mreže LOMED, čija se izgradnja planira u 1996. godini.



Slika 2.3. Šematski prikaz magistralnog prstena BEONET-a

2.4.3.2. Lokalne mreže BEONET-a

A) LOTEH

Lokalna mreža LOTEH je praktično već završena i povezana sa RCUB-om posebnim optičkim kablom, koji istovremeno međusobno povezuje i druge institucije sa ove lokacije. Optički kablovi ove mreže se mogu lako produžiti i do Pravnog fakulteta, što je u toku. LOTEH je svakako jedna od najkompleksnijih lokalnih mreža više institucija u Jugoslaviji i u ovom delu sveta. Njena specifičnost je da sadrži praktično sve moguće konfiguracije računarske opreme, opremu svih vodećih svetskih proizvođača i veliki broj različitih protokola. LOTEH, kao lokalna mreža više institucija, zapravo predstavlja otvoren skup internih mreža Računskog centra Univerziteta u Beogradu, više internih mreža Elektrotehničkog fakulteta i mreža ostalih institucija sa ove lokacije. Instalirano je nekoliko stotina računara različitih konfiguracija i namena. Ovo je najprometniji ogrank SNTIS mreže - na

glavnom segmentu konstatovano je čak 13 različitih mrežnih protokola i u trenucima vršnog saobraćaja postiže se čak 60% od teorijskog maksimalnog saobraćaja. Takav saobracaj posledica je činjenice da se preko te mreže odvija skoro kompletan tranzitni saobraćaj između Beograda i drugih akademskih centara i između SNTI mreže i sveta, kao i da se na toj mreži nalaze centralni serveri i ruteri u mreži. Na RCUB i LOTEH trenutno su na različite načine povezane i druge javne odnosno udaljene i lokalne mreže.

Na osnovu iznetih podataka vidi se da u tehničkom smislu RCUB i mreža LOTEH u ovom trenutku zapravo predstavljaju okosnicu SNTI mreže Srbije i Jugoslavije, mada su u logičkom smislu sve univerzitetske mreže ravноправne.

B) LOSTUD [6]

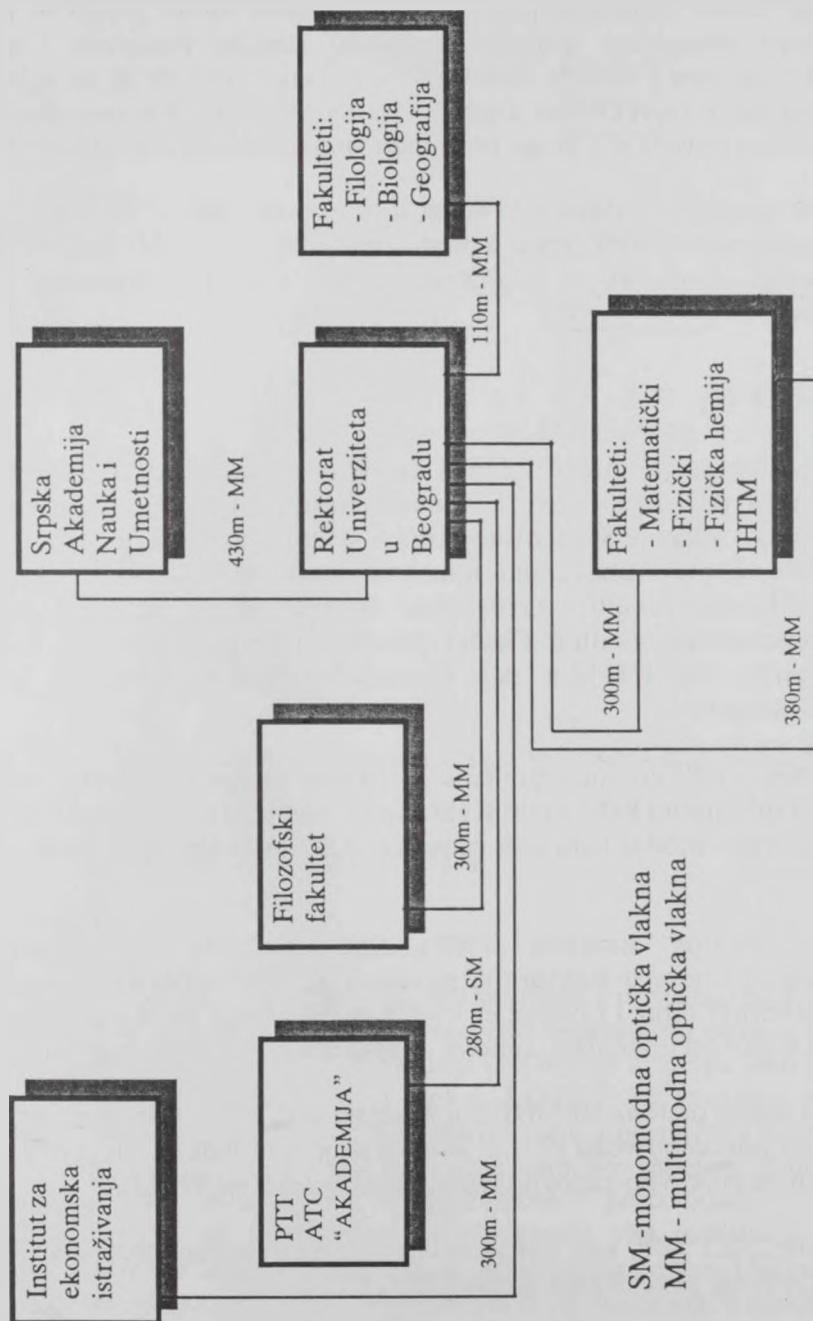
U toku je izgradnja mreže LOSTUD, koja u sadašnjoj fazi treba da kompletno ili delimično poveže trinaest institucija ili delova institucija okupljenih oko lokacije Studentskog trga u Beogradu: Rektorat Univerziteta u Beogradu, Srpska akademija nauka i umetnosti (SANU), Filozofski fakultat, Filološki fakultet, Geografski fakultet (deo), Biološki fakultet (deo), Matematički fakultet, Fizički fakultet, Hemijski fakultet, Fakultet fizičke hemije, deo IHTM-a, deo Rudarsko-geološkog fakulteta, Institut ekonomskih nauka.

U toku 1994. i 1995. godine izgrađena je lokalna mreža Rektorata, položeni su optički multimodni kablovi od Rektorata Univerziteta do zgrada svake od institucija i monomodni kabl koji povezuje ATC Akademija sa Rektoratom UB.

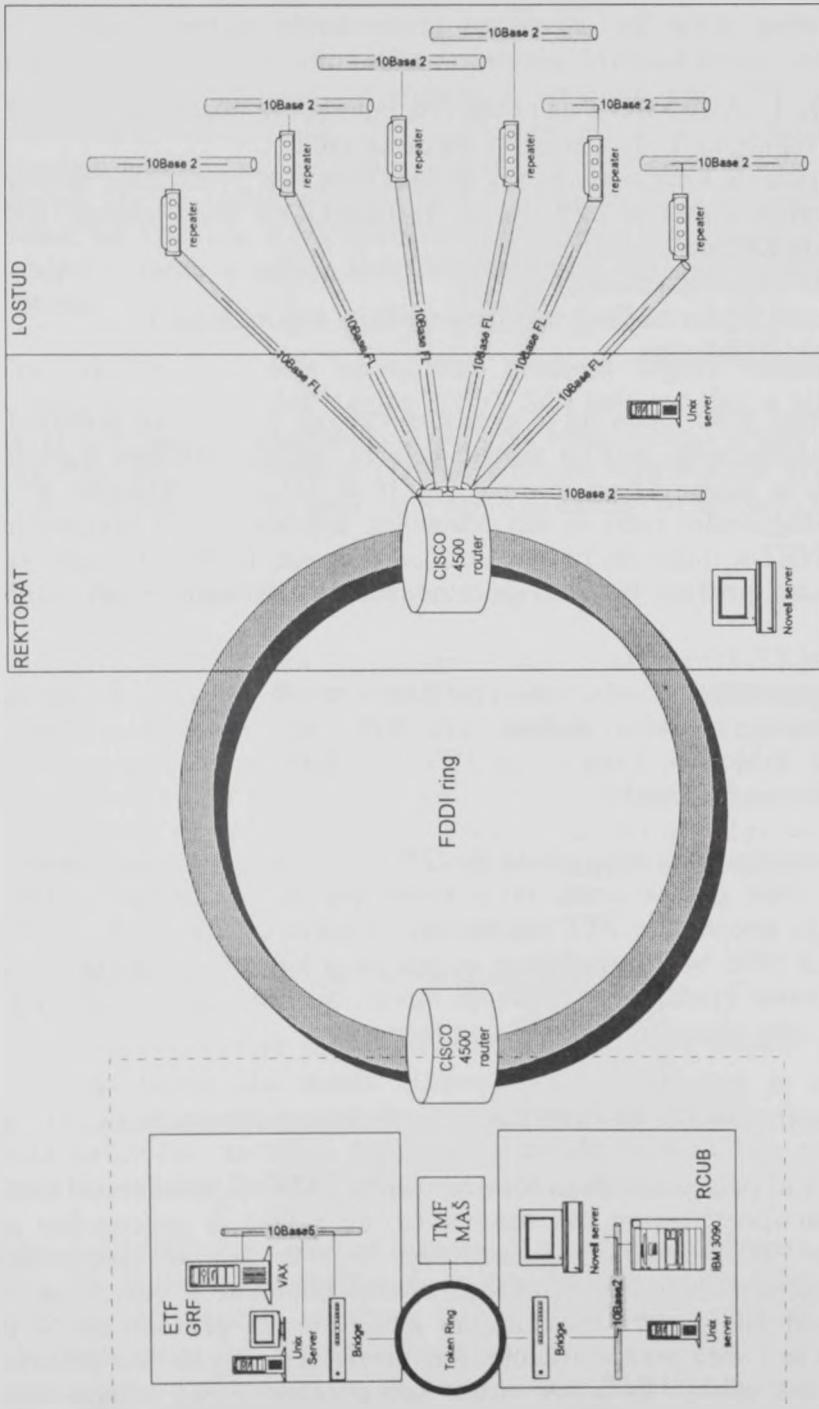
U toku četvrtog kvartala 1995. godine predviđa se postavljanje odgovarajućeg ruteru u Rektoratu, povezivanje svih institucija u zvezdastu optičku Ethernet mrežu i prelaz sa optičkog Etherneta na tanki Ethernet u zgradama svake od institucija, čime bi ova mreža bila lokalno završena.

Do magistralnog prstena BEONET-a, mreža LOSTUD će koristiti postojeću realizovanu poprečnu vezu kojom je lokalna mreža Rektorata povezana sa RCUB-om, sa relativno zadovoljavajućim protokom od 128Kb/s.

Na slikama 2.4a i 2.4b, kao tipični primeri, ilustrovani su struktura i detalji tehničkih rešenja povezivanja mreže LOSTUD.



Slika 2.4a. Struktura mreže LO STUD



2.4b. Tehničko rešenje povezivanja mreže LOSTUD

C) LOMED

Lokalna mreža LOMED treba da okupi i da poveže sve institucije u okolini Medicinskog fakulteta. U ovoj mreži treba da se nađu:

- a) Sve zgrade u krugu Medicinskog fakulteta, uz eventualno kasnije povezivanje i ostalih medicinskih institucija na ovoj lokaciji koje pripadaju Kliničkom centru;
- b) Narodna biblioteka Srbije;
- c) Sve zgrade Stomatološkog fakulteta u blizini ove lokacije; i
- d) Veterinarski fakultet.

Inicijalno je bilo predviđeno da se optičkim kablom povežu sve pomenute institucije u jedinstvenu optičku mrežu, koja bi, takođe optičkim kablom, bila povezana sa magistralnim ruterom koji bi se nalazio u obližnjem PTT TK centru. Magistralni ruter bi bio postavljen neposredno na magistralni prsten BEONET-a, tako da bi sve institucije mreže LOMED imale sve pogodnosti koje pružaju brze magistralne veze sa ostalim delom mreže BEONET.

Ipak, treba napomenuti da bi ovako planirana mreža zahtevala polaganje optičkih kablova značajne dužine, pri čemu se postavlja i pitanje raspoloživosti kablovske kanalizacije PTT, bez koje se realizacija mreže LOMED teško može zamisliti.

Jednim od planova realizacije mreže BEONET bilo je predviđeno da se u toku 1993. i 1994. godine urade svi potrebni projekti, pripreme tehnička rešenja, dobiju neophodne PTT saglasnosti i nabave optički kablovi. Kraj 1995. i početak 1996. bili su predviđeni za polaganje kablova, posle čega bi se pristupilo nabavci uređaja i povezivanju mreže. Nažalost, zbog nedostatka sredstava, sa ovim planovima se značajno kasni.

D) Ostale lokalne mreže BEONET-a, povezivanje ostalih institucija

Četvrta odnosno peta predviđena lokalna mreža, LOV na Voždovcu i LOZ u Zemunu, su specifične po dva osnova: (a) ne nalaze se neposredno na magistralnom FDDI prstenu, (b) institucije koje ih čine su medusobno udaljenije nego u slučaju ostalih lokalnih mreža. Usvojeno je rešenje da se navedene institucije povežu standardnim kablovima za protoke od 64 ili 128Kb/s, i da se i veza prema ostalom delu mreže realizuje za iste protoke. Postavlja se čak pitanje da li ove mreže kao posebne celine uopšte treba realizovati ili je bolje da se svaka pojedinačna institucija poveže neposredno na magistralni prsten.

Povezivanje ostalih pojedinačnih institucija, od kojih neke zbog svoje veličine imaju i relativno velike sopstvene institucionalne lokalne mreže, ostvaruje se uspešnom posredstvom iznajmljenih PTT linija mesne kablovske mreže. U toku 1995. godine počelo se sa uspesnom primenom relativno jestinih serijskih rutera i serijskih modema koji rade u osnovnom opsegu, što na rastojanjima od nekoliko kilometara obezbeđuje značajan protok od 128Kb/s, u najvećem broju slučajeva dovoljan i za duže vreme. Priključne tačke u načelu treba da budu bilo gde na magistralnom FDDI prstenu.

Veza akademske mreže sa mrežom državnih organa realizuje se preko servera NAUKA (radna stanica INDY SG) koji se nalazi u Ministarstvu za nauku i tehnologiju Republike Srbije. Na ovom racunaru planirano je da se formiraju Web prezentacije rezultata naučno istraživačkih projekata kao i različiti podaci i prezentacije relevantne za naučno istraživački rad.

E) Globalni prikaz BEONET-a

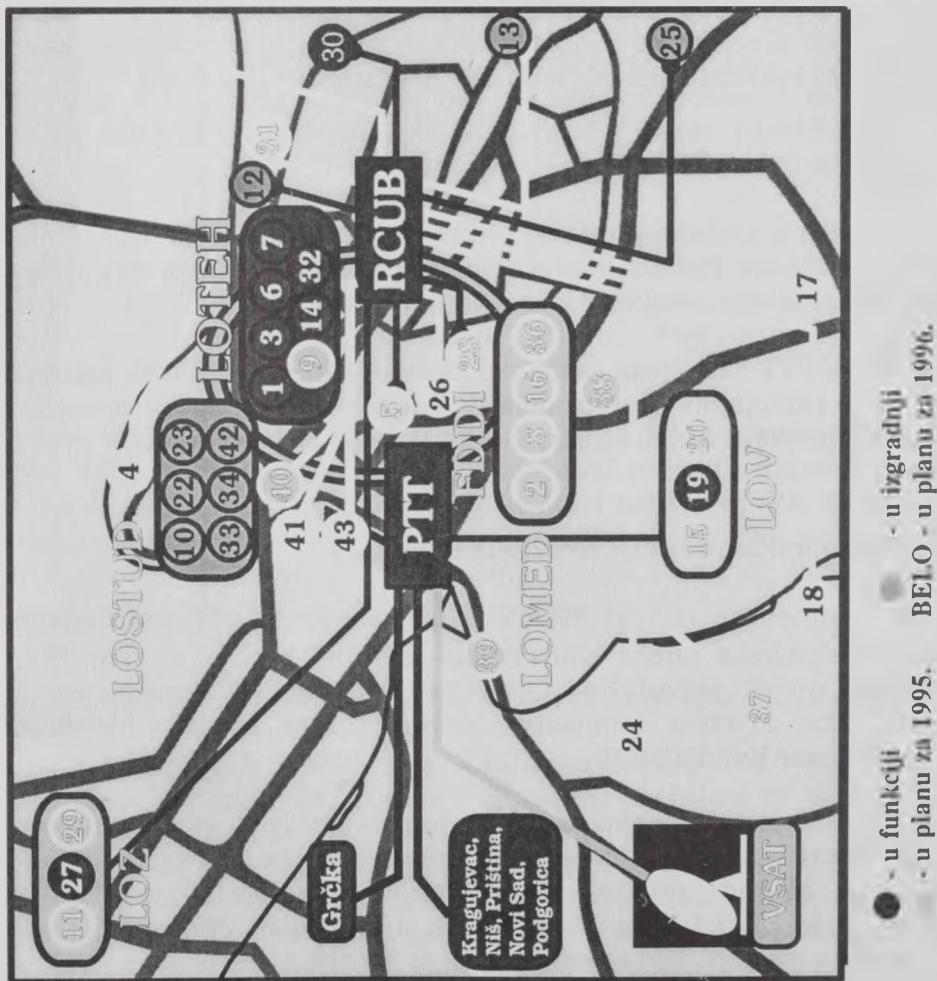
Na slici 2.5 dat je globalni prostorni prikaz mreže BEONET sa naznačenim glavnim korisnicima. Posto se radi o otvorenoj mreži kojoj mogu da pristupe svi zainteresovani korisnici, spisak naznačenih institucija nije isključiv.

Pored fiksnih PTT veza, zemaljskim kablovima i drugim sistemima prenosa, koncepcijom razvoja SNTI predviđeni su, kao što je već pomenuto, mogući izlazi prema međunarodnim mrežama korišćenjem satelitskih veza.

2.4.4. Računarska mreža Univerziteta u Nišu [7]

U okviru Koncepcije razvoja SNTIS [1] urađen je projekt pod nazivom "Lokalna računarska mreža Univerziteta u Nišu" koji je završen 1992. godine. Ovaj projekt je poslužio kao osnova da se odmah pristupi zameni do tada delimično završene terminalske mreže, vezane na tada instalirani centralni računar Bull DPS8/70.

Mesna odnosno institucionalna mreža Univerziteta u Nišu je projektovana tako da stvori i obezbedi sve neophodne uslove za realizaciju JUNIS (Jedinstveni naučno nastavni univerzitetski informacioni sistem), da obezbedi vezu sa istraživačko-razvojnim i stručnim organizacijama i pojedincima u privredi, i da obezbedi vezu sa SNTIS/Akademskom mrežom Srbije, odnosno preko nje sa Internetom i ostalim međunarodnim mrežama i informacionim servisima.



Slika 2.5. Globalni prostorni prikaz mreže BEONET

U toku realizacije projekta mreže, u periodu 1992 i 1993. godine, zamenjen je i centralni računar, odnosno kao host i glavni server mreže instaliran je računar proizvodnje Silicon Graphics, model Challenge L, (dvoprocesorski UNIX/RISC računar).

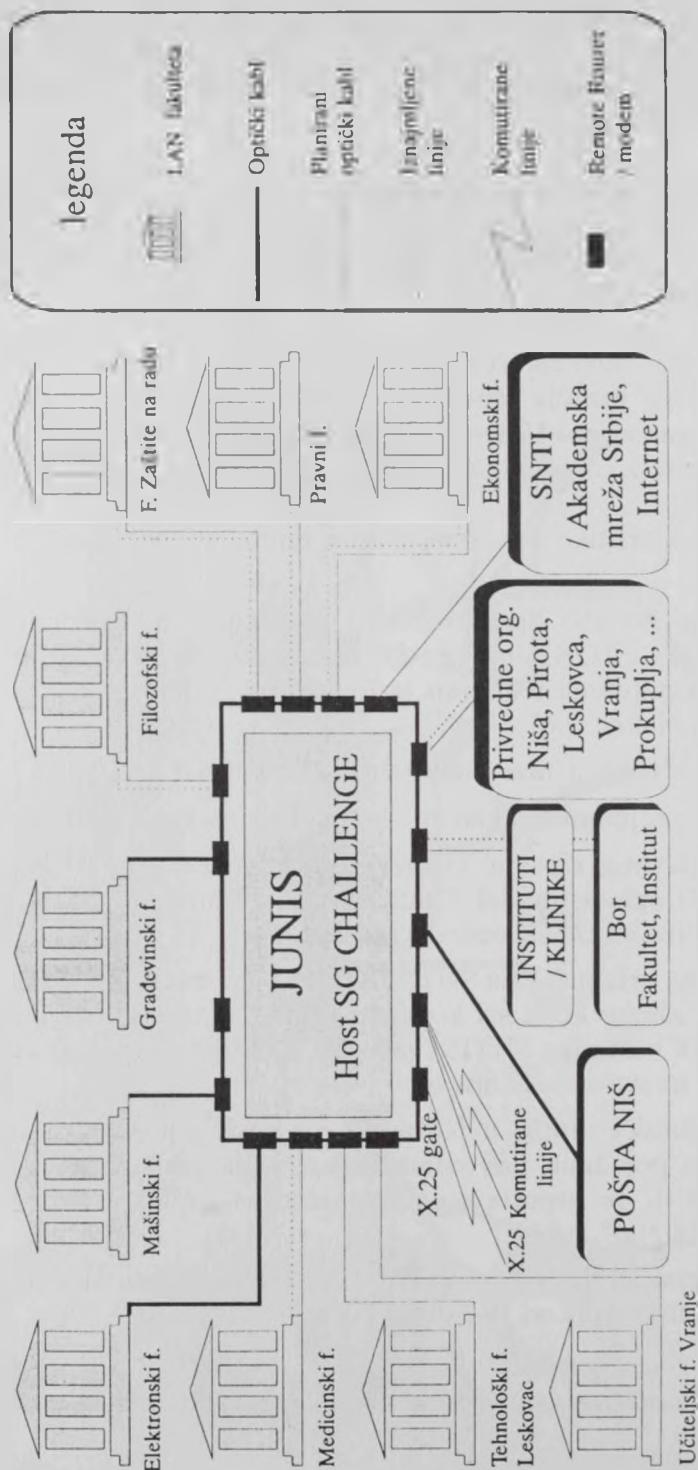
U skladu sa Koncepcijom razvoja SNTIS, mreža Univerziteta u Nišu, koja već značajno prevazilazi okvire samo univerzitetskih institucija, danas predstavlja deo SNTIS/Akademске mreže Srbije, odnosno SRJ, a time i deo globalne svetske mreže Internet.

Na osnovu prethodnog stanja razvoja mreže Univerziteta u Nišu i uzimajući u obzir trendove razvoja modernih mreža, odabran je koncept otvorene "privatne" mesne mreže Univerziteta u Nišu, koja objedinjuje lokalne mreže pojedinih fakulteta i drugih institucija u jedinstvenu gradsku mrežu.

Osnovne karakteristike ove realizovane mreže su pobrojane u nastavku teksta:

- Mreža je WAN tipa zvezdaste strukture sa glavnim serverom smestenim u JUNIS-u (zgrada Rektorata) sa realizovanim vezama prema fakultetima i drugim institucijama u gradskom i regionalnom okruženju (slika 2.6).
- Čvorne računare Univerzitetske mreže čine UNIX računari.
- Mreža je projektovana kao mreža tipa Ethernet sa TCP/IP protokolom.
- Ulogu glavnog servera Univerzitetske mreže ima UNIX računar Silicon Graphics, model Challenge L, dvoprocesorska mašina RISC arhitekture, sa UNIX operativnim sistemom.
- Mreža ima priključak na SNTIS/Akademsku mrežu Srbije iznajmljenim vezama, preko kojih svi korisnici Univerzitetske mreže imaju pristup do ostalih korisnika SNTIS i izlaz (za elektronsku poštu) preko mreže Internet na sve svetske mreže.
- Univerzitetska mreža u Nišu ima i direktnе priključke sa lokalnim mrežama pojedinih privrednih organizacija grada Niša sa kojima je značajno da se uspostavi naučno-tehnička saradnja (MIN, DIN, Ei, Institut "1 maj", itd.).
- Mesne magistrale mreže čine optički vodovi i iznajmljene telefonske linije sa protocima od 19.2 do 64 Kb/sek.

RAČUNARSKA MREŽA UNIVERZITETA U NIŠU



Slika 2.6. Globalno rešenje mreže Univerziteta u Nišu

- Na svakom fakultetu, odnosno institutu, klinici ili drugoj instituciji koja je priključena na mrežu, realizovana je najmanje jedna lokalna mreža tipa Ethernet.
- Na katedrama i laboratorijama fakulteta formiraju se posebne lokalne mreže koje se preko odgovarajuće komunikacione opreme (routeri, mostovi (bridge)) povezuju na magistralu fakulteta i dalje prema mesnoj mreži Univerziteta. Komunikacija između pojedinih lokalnih mreža katedri jednog fakulteta ostvaruje se preko magistrala fakulteta, a sa lokalnim mrežama katedri na drugim fakultetima preko magistrala Univerziteta. Na pojedinim fakultetima postoji više povezanih mreža. Na primer na Elektronskom fakultetu realizovano je devet lokalnih mreža u okviru samog fakulteta (slika 2.7).

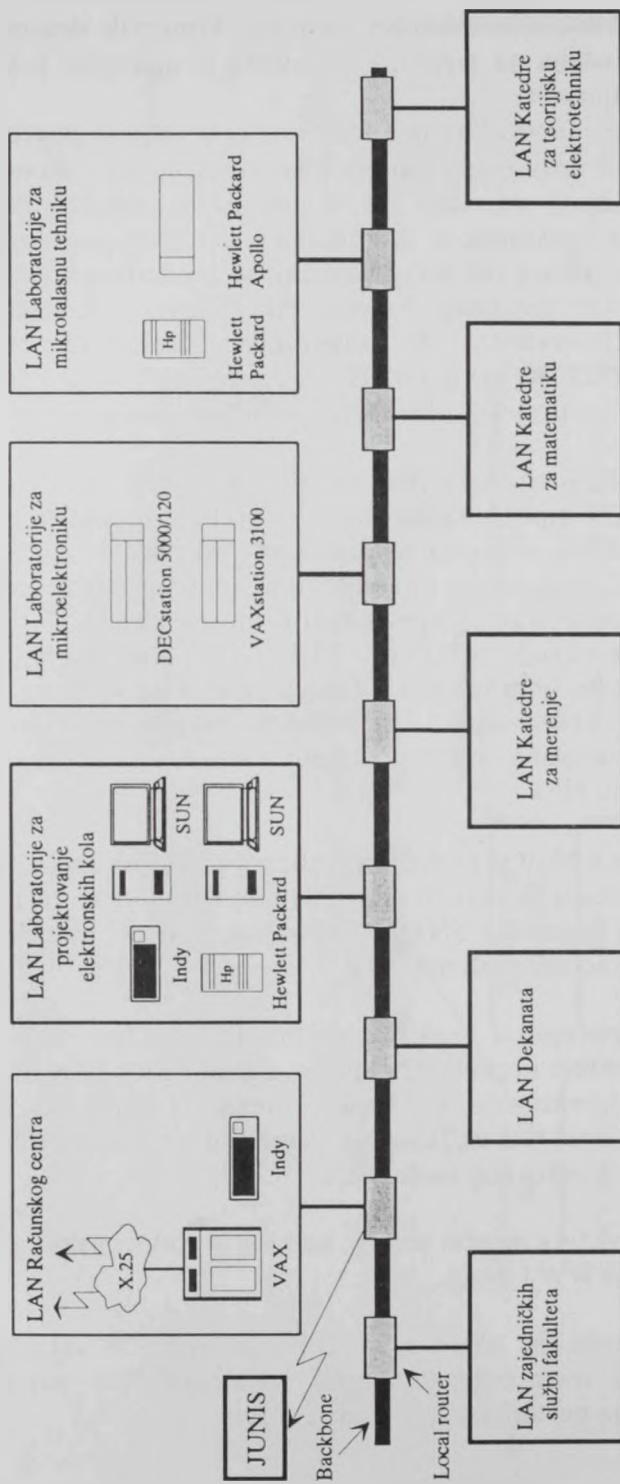
Povezivanje lokalnih mreža koje čine mesnu odnosno institucionalnu mrežu Univerziteta (WAN tipa) realizovano je preko iznajmljenih linija sa protocima od 19.2Kb/s, odnosno sa protocima od 28.8Kb/s za fakultete u Leskovcu i Vranju. Veza preko optičkih kablova, koji će omogućiti mnogo veće protokle pri komunikaciji, u prvoj fazi 10Mb/s a kasnije do 100Mb/s, za sada je obezbeđena na relaciji JUNIS - PTT Niš, a nabavljen je optički kabl za povezivanje sa lokacijom tehničkih fakulteta. Planira se da veza optičkim kablovima bude realizovana i sa ostalim fakultetima, institutima i privrednim organizacijama, osim pojedinih klinika i organizacija koje se nalaze u široj okolini Niša.

Mreža Univerziteta u Nišu je preko iznajmljenih direktnih linija povezana sa mrežama Univerziteta u Beogradu (RCUB) i Prištini (Rektorat) i tako čini deo međugradskog segmenta SNTIS. Takođe se koristi i direktni JUPAK priključak (X.25), kao alternativna veza.

Mreži se može pristupiti i preko komutiranih veza, za sada preko 10 priključaka - telefonskih brojeva. U planu je izgradnja još 10 priključaka, što će omogućiti još intenzivnije korišćenje mrežnih resursa od kuće ili sa lokacija koje nisu direktno uključene u mrežu, u vremenu kada je mreža rasterećena dnevnom redovnom saobraćaju.

Realizovani su svi važniji mrežni servisi, kao što je elektronska pošta, telnet, ftp, gopher, news, WWW i drugi.

Glavni server mreže je aktivan neprekidno svih 24 sata dnevno, a zahvaljujući ranije obezbeđenoj opremi za neprekidno napajanje radi stabilno i u uslovima nestanaka električne energije.



Slika 2.7. Lokalne mreže Elektrotehničkog fakulteta u Nišu

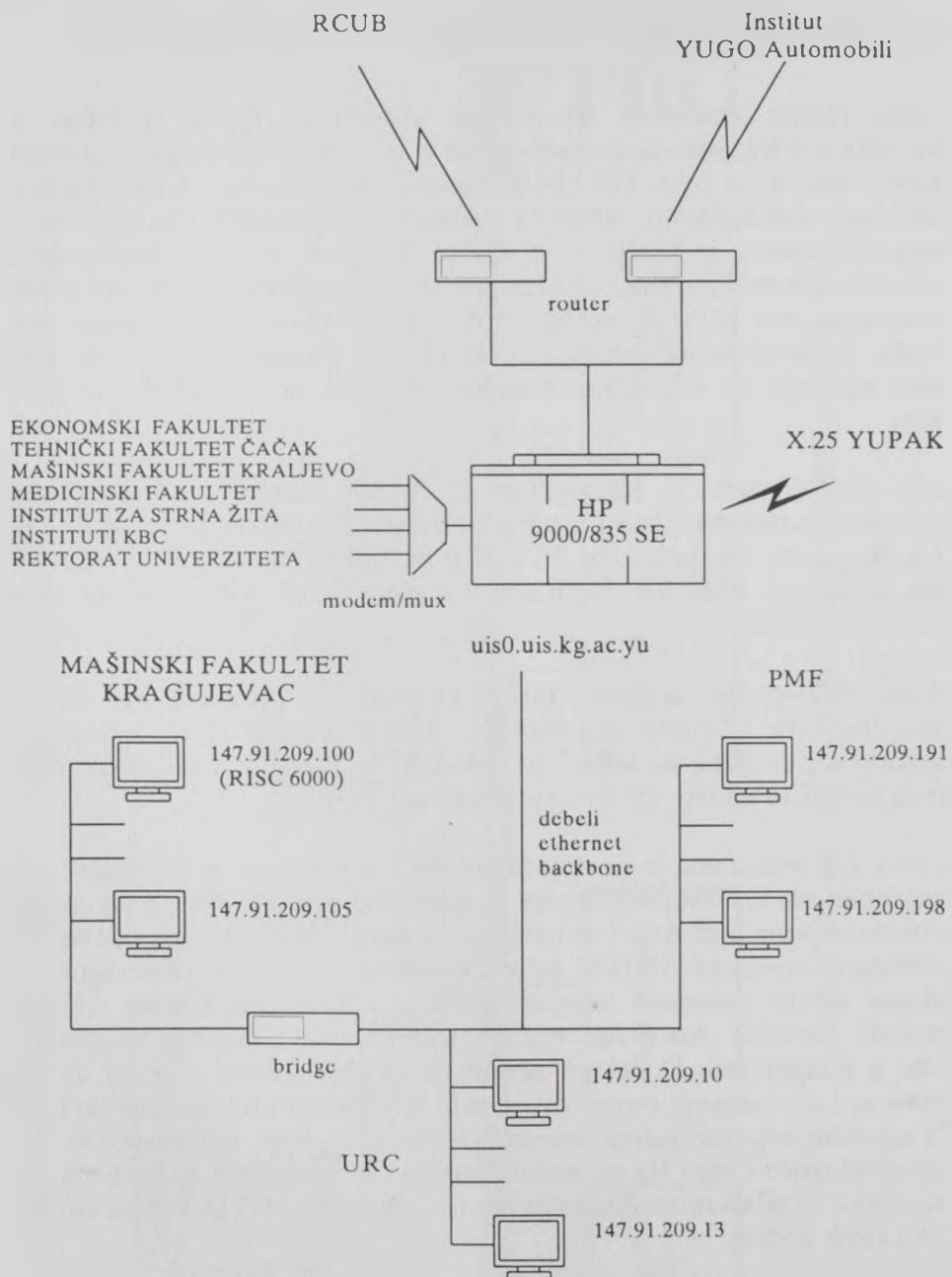
2.4.5. Računarska mreža Univerziteta u Kragujevcu [8]

U toku ranijih aktivnosti na razvoju akademske mreže u Srbiji, na Univerzitetu u Kragujevcu je realizovana tehnološki jedinstvena za to vreme moderna mreža, sa četiri HP UNIX čvorna računara, koji su obezbeđivali povezivanje svih tadašnjih fakulteta u sastavu Univerziteta u Kragujevcu, iz samog Kragujevca, iz Kraljeva i iz Čačka. Nažalost, prilikom realizacije te mreže nisu bile realizovane odgovarajuće stalne magistralne veze sa Čačkom i Kraljevom, već su se koristile, i još uvek se koriste, komutirane veze. Takođe, institucionalna mreža Univerziteta u Kragujevcu sve do 1994. godine nije bila na odgovarajući način povezana sa ostalim delom SNTI mreže.

Mreža Univerziteta u Kragujevcu je WAN mreža zvezdastog tipa. Univerzitet je trenutno, kao i ostali univerziteti u Srbiji, povezan sa RCUB-om u Beogradu iznajmljenom PTT linijom, pri čemu se koristi relativno spora asinhrona veza od 19,2Kb/s. Postojeći X-25 priključak na javnu paketsku mrežu koristi se za neposredne paketske veze i kao rezervna veza.

U toku 1995. godine u mesnu mrežu Univerziteta povezana je i lokalna mreža Instituta "Zastava automobili". Takođe, u sastav Univerziteta u Kragujevcu primljeni su Učiteljski fakulteti u Jagodini i u Užicu, pa se odmah postavilo pitanje njihovog priključenja u mrežu.

Na slici 2.8 prikazana je postojeća mreža Univerziteta u Kragujevcu, sa naznačenim planovima povezivanja u narednom periodu. Pored planova za osavremenjivanje postojeće i za nabavku nove opreme, posebno više servera za instalaciju osnovnih SNTIS i Internet servisa, posebno je važno da se sve udaljene mesne odnosno lokalne mreže (u Kraljevu, Čačku, Užicu i Jagodini) povežu što bržim stalnim iznajmljenim linijama sa centrom mreže u Kragujevcu. U sklopu aktivnosti za dalji razvoj sistema SNTIS, planira se i povezivanje centra ove mreže sa najbližim telekomunikacionim PTT centrom odgovarajućom vezom sa većim protokom, minimalno 64 Kb/s a ako je moguće i više. Na taj način i mreža Univerziteta u Kragujevcu bila bi spremna za modernizaciju međumesnog segmenta SNTIS koji se očekuje u toku 1996. godine.



Slika 2.8. Mreža Univerziteta u Kragujevcu

2.4.6. Racunarska mreža Univerziteta u Novom Sadu [9]

Institucionalna mreža Univerziteta u Novom Sadu je organizovano počela da se formira relativno skoro, mada je nekoliko institucija iz Novog Sada i Vojvodine, posebno biblioteka Matice Srpske, Rektorat Univerziteta i Institut za Matematiku, celo vreme intenzivno sarađivalo i sarađuje u razvoju SNTIS.

Po svojoj prirodi mreža Univerziteta u Novom Sadu je WAN tipa. Postoje u osnovi tri nivoa povezivanja u mreži:

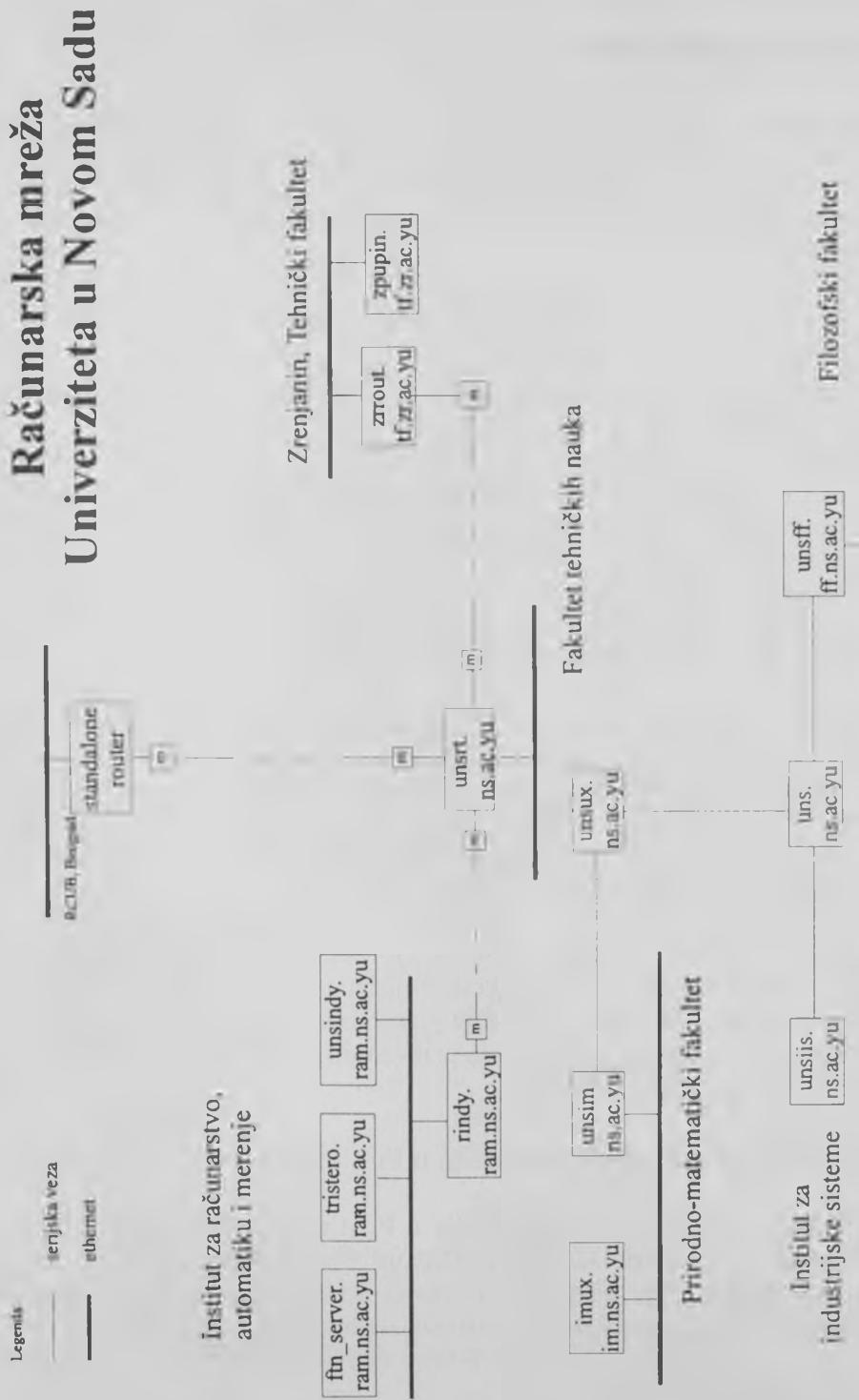
- (a) povezivanje institucija koje se nalaze koncentrisane u neposrednoj blizini Rektorata, u okviru tzv. Kampusa Univerziteta;
- (b) povezivanje ostalih članova Univerziteta i drugih institucija, posebno Matice Srpske, koje se nalaze na teritoriji grada ali izvan Kampusa;
- (c) povezivanje ostalih institucija od interesa koje su članovi Univerziteta ili gravitiraju ka Univerzitetu, na prvom mestu Građevinskog i Ekonomskog fakulteta u Subotici, Tehničkog fakulteta "M. Pupin" u Zrenjaninu i Učiteljskog fakulteta u Somboru, zatim Instituta u Sremskoj Kamenici, itd.

Krajem 1995. godine u toku su radovi na projektovanju i realizaciji lokalne mreže u okviru Kampusa Univerziteta, zatim lokalnih mreža pojedinih institucija, nabavka i kompletiranje opreme, posebno glavnog servera mreže računara DEC ALFA, kao i povezivanje dela mreže. Za 1996. godinu su u planu sva povezivanja sa pomenutim udaljenim institucijama na teritoriji i van teritorije grada Novog Sada.

Univerzitet u Novom Sadu povezan je sa RCUB-om u Beogradu iznajmljenom asinhronom telefonskom linijom sa protokom od 19,2Kb/s, a postoji i rezervna X-25 veza. Sadašnje stanje mreže Univerziteta u Novom Sadu i naznačeni pravci daljeg razvoja prikazani su na slici 2.9.

2.4.7. Racunarska mreža Univerziteta u Prištini [10]

Projekat računarske mreže Univerziteta u Prištini urađen je 1994. godine u okviru saradnje Univerziteta u Nišu i Prištini. Projektovana mreža, u skladu sa Koncepcijom razvoja SNTIS, predstavlja po svim elementima integralni deo mreže SNTIS, a time u potpunosti odgovara standardima Interneta.



Slika 2.9. Mreža Univerziteta u Novom Sadu

Planirane su sledeće osnovne karakteristike mreže:

- Mreža je WAN tipa, sa zvezdastom Ethernet strukturu i TCP/IP protokolom;
- Predviđeno je da svi serveri u mreži budu UNIX tipa, da glavni server bude smešten u zgradu Rektorata i da se ostvare veze (a) lokalno prema svim fakultetima u Prištini, (b) regionalno prema fakultetima u Obiliću i Prizrenu, i (c) magistralno prema ostalom delu SNTIS mreže preko Univerziteta u Nišu (slika 2.10);
- Glavni server mreže je Silicon Graphics-Indigo (Risc/Unix) računar;
- Sistem prenosa je planiran preko iznajmljenih PTT linija sa protocima od 19,2 do 64Kb/s, uz korišćenje X-25 priključka kao alternativne veze;
- Predviđeno je da se na svakom fakultetu realizuje minimalno jedna lokalna mreža.

U odnosu na projekat do sada je realizovano sledeće:

- Nabavljen je i instaliran glavni server mreže, Silicon Graphics Indigo radna stanica, ali nisu obezbeđeni svi uslovi za njegov neprekidni rad;
- Realizovan je deo lokalne mreže u zgradi Rektorata;
- Na pojedinim fakultetima su nezavisno nabavljeni različiti PC računari i delom povezivani u različite lokalne mreže, ali bez obezbeđenih veza prema glavnem serveru mreže, osim realizovane veze do Elektrotehničkog fakulteta;
- Mreža Univerziteta u Prištini je preko glavnog servera mreže povezana u SNTIS mrežu iznajmljenom PTT linijom do Niša sa protokom od 28,8Kb/s.

U planu za 1996. godinu je povezivanje još četiri fakulteta. Istovremeno je u toku izrada projekta Informacionog sistema Univerziteta, kojim bi se definisali svi potrebni uslovi, aktivnosti i resursi za realizaciju i IS i računarske mreže Univerziteta. Ova mreža obuhvatila bi i sve druge istraživačke i stručne institucije koje gravitiraju ka Univerzitetu.

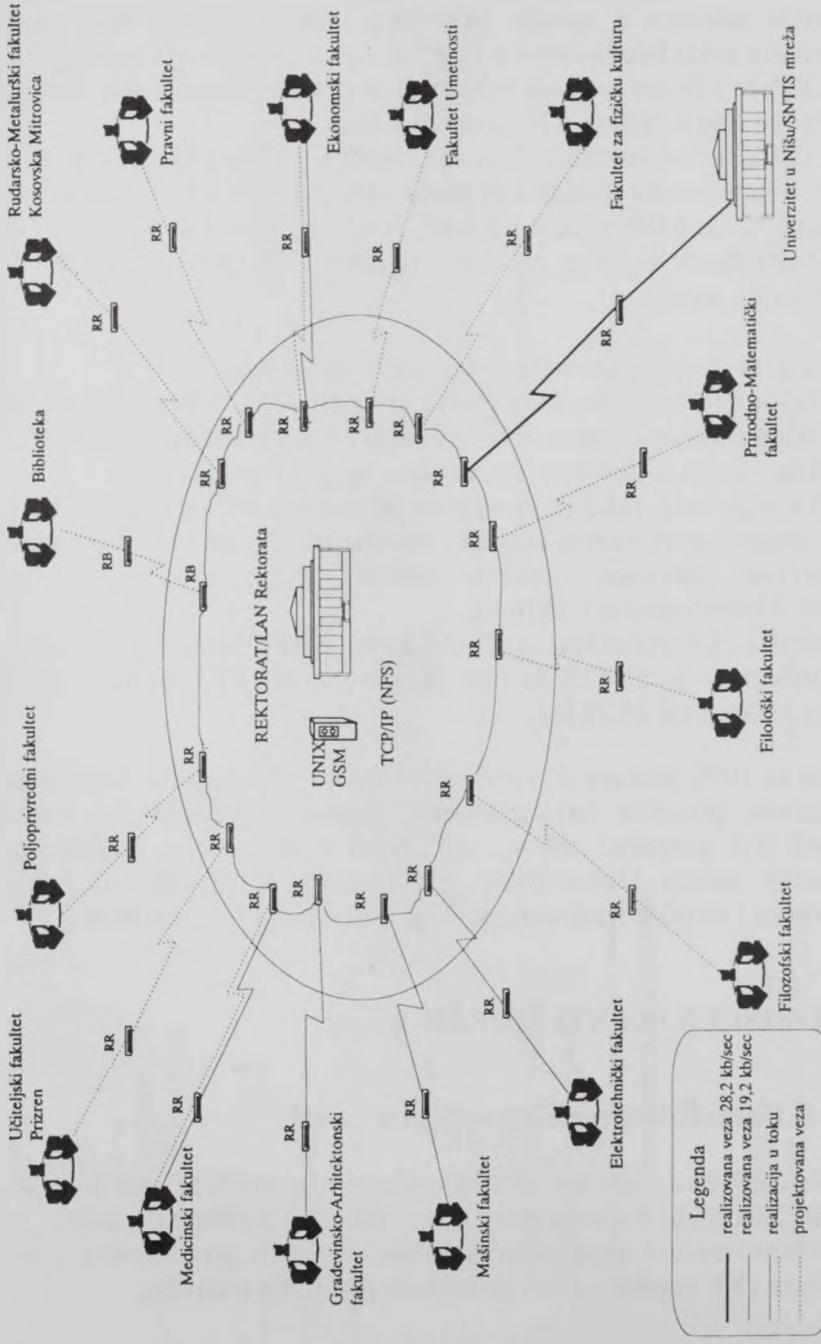
2.5. KORIŠĆENJE SNTI MREŽE [11,12]

2.5.1. Raspoložive servisne usluge u mreži

U okviru mreže i sistema SNTIS trenutno su moguće standardne mrežne funkcije, slično kao i u svim drugim savremenim mrežama u svetu:

- Transparentni nacionalni distribuirani servis pretraživanja i korišćenja baza (YU gopher sa tzv. Rečnikom podataka o mreži);

Računarska mreža Univerziteta u Prištini



Slika 2.10. Mreža Univerziteta u Prištini

- Elektronska pošta (**e-mail**), unutar SNTI i u odnosu na ceo svet;
- Elektronske oglasne table (**gopher**) i servis za njihovo pretraživanje (**veronica**);
- Prenos datoteka i pristup bazama javno dostupnog softvera (**anonymous ftp**), neposredno unutar delova sadašnje SNTI mreže, posredno u odnosu na svetske mreže;
- Interaktivno priključenje na udaljeni računar (**telnet** i slični servisi), neposredno unutar pojedinih delova sadašnje SNTI mreže, posredno u odnosu na svetske mreže;
- Interaktivne poruke (**talk, irc, phone** i slični servisi);
- Elektronske konferencije, elektronska obaveštenja, itd.;
- Organizovana distribucija vesti (**news**);
- Informacije o adresi u elektronskoj pošti (**finger, netfind**);
- Distribuirane nove usluge, poznate iz inostranih mreža (**World Wide Web - WWW, archie**).

U neposrednoj budućnosti se očekuje značajno proširenje i podizanje kvaliteta ovih servisnih usluga:

- Neposredno priključenje na Internet mrežu i obezbeđenje "Full Internet Connectivity", za sve korisnike SNTIS;
- Distribuirane on-line usluge iz inostranih mreža (**gopher, veronica, finger, ftp anonymous, world telnet, WWW, archie,...**);
- Razvoj, instalacija na mrežu, stalno ažuriranje i korišćenje više baza od zajedničkog interesa;
- Uključenje specijalizovanih baza podataka.

Treba naglasiti da je prvi pomenuti servis, transparentni nacionalni distribuirani servis pretraživanja i korišćenja baza podataka u okviru Sistema naučnih i tehnoloških informacija Srbije i Jugoslavije originalno ostvarenje koje je namenski razvijano za potrebe SNTI. Ovaj servis će kasnije u ovoj monografiji biti detaljno objašnjen, odnosno biće dat i pregled informacionih sadržaja koji se mogu postaviti, pretraživati i koristiti primenom ovog originalnog sistema.

2.5.2. Nacin priključenja korisnika u mrežu SNTIS

Pristup u mrežu SNTIS je moguć na više načina:

- sa terminala svakog računara koji je čvor SNTI mreže;
- sa računara svake lokalne mreže koja je povezana na neki od više ravnopravnih čvorova;

- sa svakog računara u zemlji ili svetu koji ima izlaz u javnu X-25 mrežu, sa mogućnošću ravnopravnog pristupa na više različitih čvorova;
- sa svakog računara u zemlji ili svetu koji ima izlaz u javnu telefonsku mrežu, preko odgovarajućih modema, sa mogućnošću ravnopravnog pristupa na više različitih čvorova.

Naravno, konfor rada i tehničke mogućnosti korišćenja različitih postojećih servisa u mreži u sadašnjem trenutku izrazito zavise od mesta priključenja. Prividno je potpuno transparentan jedino e-mail servis, koji se zapravo koristi "of line" tako da korisnik nema utisak da odziv zavisi od mesta priključenja (naravno pod uslovom da je mreža ispravna i aktivna). Svi ostali servisi u manjoj ili većoj meri zavise od mesta i načina priključenja, kao i od konfiguracije puta prenosa između korisničkog računara i odgovarajućih servera koji obezbeđuju tražene servise. U principu, najbolji je odziv mreže (nekoliko Mb/s) ukoliko su i klijent i server u okviru iste lokalne Ethernet mreže. U takvom slučaju i servisi sa prenosom slike, na primer WWW, rade brzo i efikasno. Uslovno rečeno najlošiji uslovi rada se ostvaruju ako je deo puta prenosa komutirana standardna analogna telefonska linija, ili iznajmljena analogna linija u međumesnom saobraćaju, posto se tada ni teorijski ne može ostvariti protok veći od 28,8Kb/s, a često je ova vrednost znatno manja. U istu relativno losu kategoriju spada nažlost i JUPAK (X-25) javna paketska mreža, kod koje se retko ostvaruju protoci veći od 9,6Kb/s.

U toku 1996. godine očekuje se postepeno uvođenje digitalnih puteva prenosa u međumesnom saobraćaju na većem broju trasa, što će omogućiti da se relativno lako i brzo poveća propusna moć SNTIS mreže. U granicama mogućnosti planira se da svi magistralni međumesni pravci imaju protok od najmanje 64Mb/s, a da sve mesne i institucionalne magistrale budu sa najmanje 128Mb/s.

2.5.3. Važnije mreže i broj povezanih računara u SNTI mreži

U tabelama T-2.1 i T-2.2 prikazan je presek stanja mreže SNTI početkom novembra 1995. godine. Kao što se vidi, u mrežu je bilo uključeno približno 1280 računara, pri čemu je oko dve trećine povezano u Beogradu. Podaci u tabelama su samo jedan od mogućih pokazatelja pošto se na taj način ne može videti broj korisnika niti obim korišćenja mreže. U svakom slučaju, na osnovu evidentirane brzine povezivanja pojedinih lokalnih i institucionalnih mreža u toku 1995. godine u mrežu SNTI, kao i na osnovu podataka o

radovima koji su u toku, može se očekivati najmanje dvostruki broj računara u mreži na kraju 1996. godine.

Institucija	Domen	Povezanost	Br. rač.
Elektrotehnički fakultet	<i>ctf.bg.ac.yu</i>	TCP/IP	150
RC Univerziteta (RCUB)	<i>rcub.bg.ac.yu</i>	TCP/IP	30
Građevinski fakultet	<i>grf.bg.ac.yu</i>	TCP/IP	20
Tehnološko-metalurški fak.	<i>tmf.bg.ac.yu</i>	TCP/IP	30
LOTEH (UKUPNO)			230
Rektorat Univerziteta	<i>rect.bg.ac.yu</i>	TCP/IP	20
Matematički fakultet	<i>matf.bg.ac.yu</i>	UUCP	30
Hemijski fakultet	<i>chem.bg.ac.yu</i>	UUCP	50
SANU	<i>sanu.ac.yu</i>	UUCP	200
LOSTUD (UKUPNO)			300
Fakultet organizacionih nauka	<i>fon.bg.ac.yu</i>	TCP/IP	50
LOV (UKUPNO)			50
Institut za fiziku, Zemun	<i>phy.bg.ac.yu</i>	TCP/IP	50
Institut "Iritel"	<i>iritel.bg.ac.yu</i>	UUCP	50
LOV (UKUPNO)			100
Institut "Mihajlo Pupin"	<i>imp.bg.ac.yu</i>	TCP/IP	50
Instit. za nukl. nauke "Vinča"	<i>vin.bg.ac.yu</i>	TCP/IP	70
Tehnički fakultet u Boru	<i>bor.ac.yu</i>	TCP/IP	20
Ostali (UKUPNO)			140
Univ. u Beogradu	<i>bg.ac.yu</i>		820

Tabela 2.1. Pregled broja računara u mreži BEONET

Institucija	Domen	Povezanost	Br. rač.
Univerzitet u Beogradu	<i>bg.ac.yu</i>	TCP/IP	820
Univerzitet u Nišu	<i>ni.ac.yu</i>	TCP/IP	300
Univerzitet u Novom Sadu	<i>ns.ac.yu</i>	TCP/IP	30
Fakultet u Zrenjaninu	<i>zr.ac.yu</i>	TCP/IP	10
Univerzitet u Kragujevcu	<i>kg.ac.yu</i>	TCP/IP	50
Univerzitet u Prištini	<i>pr.ac.yu</i>	TCP/IP	10
Univerzitet u Podgorici	<i>cg.ac.yu</i>	TCP/IP	20
UKUPNO U SRJ	<i>ac.yu</i>		1240

Tabela 2.2. Pregled broja racunara po gradovima

2.6. LOGIČKE MREŽE SNTIS

Koncepcijom razvija SNTIS je utvrđeno da institucionalno SNTI ne treba ograničavati unapred definisanom strukturom. Institucije SNTI autonomno razvijaju svoje informacione sisteme, povezuju ih sa drugim sistemima u logičke mreže na osnovu prepoznatih zajedničkih interesa (na primer zajedničkih informacionih fondova), na osnovu potreba za intenzivnom komunikacijom i transferom znanja, radi ostvarivanja zajedničkih funkcija, uzajamnog proširivanja resursa, komercijalnih interesa, itd. U ovom trenutku razvoja mreže SNTIS razvijaju se sledeće logičke mreže:

- (a) Upravljačka mreža;
- (b) Bibliotečka mreža;
- (c) Mreža specijalizovanih informacionih sistema;
- (d) Mreža tehnološko - poslovnih informacija.

2.6.1. Upravljačka mreža SNTIS [13]

Upravljačka mreža u okviru svoje osnovne funkcije treba da obezbedi efikasno upravljanje jednim od najznačajnijih resursa za razvoj naučno-istraživačkog i istraživačko-razvojnog rada - naučno-tehnološkim (i delom poslovnim) informacijama. Upravljanje ovim resursom je u tesnoj vezi sa upravljanjem ostalim resursima za razvoj NIR-a - opremom, kapitalom i kadrovima, odnosno u vezi je i sa praćenjem primene rezultata NIR-a. Zbog toga Upravljačka mreža predstavlja samo jedan deo informacionog sistema za upravljanje razvojem nauke i tehnologije u Republici Srbiji. Ovakva mreža, do projekta razvoja SNTIS, nije ranije bila definisana i tretirana na način koji je prikazan u nastavku ovog teksta.

Upravljačka mreža SNTIS treba da poseduje svoj informacioni sistem koji će obezbediti neophodnu podršku za upravljanje funkcionisanjem i razvojem SNTI. Ovaj informacioni sistem treba da pruži sve podatke o SNTIS. Pošto je i sam SNTIS takođe informacioni sistem, Upravljačka mreža predstavlja zapravo "informacioni sistem o informacionom sistemu" - takozvani "meta informacioni sistem" odnosno Rečnik podataka. Rečnik podataka (koji se ponekad naziva i katalog, direktorijum, enciklopedija ili metabaza podataka) definiše se kao baza podataka koja opisuje ostale objekte u nekom informacionom sistemu. Struktura i sadržaj rečnika podataka su definisani odgovarajućim ANSI i ISO standardima. Osim toga, rečnik podataka obuhvata i druge baze podataka relevantne za SNTIS kao što su: baza podataka naučno istraživačkih organizacija, istraživačkih projekata u

toku, rezultata naučnih i istraživačkih projekata i slično. Sve ove baze predviđene su za pretraživanje u mreži SNTIS na jedinstven način, kao što je opisano u drugim delovima ove monografije.

Rečnik podataka SNTIS treba da opiše jedan složen, otvoren i heterogen sistem, njegovu logičku strukturu, na primer "logičke mreže", odnosno institucije sistema, njihove uloge i njihove različite međusobne odnose, korisnike, informacione servise, njima pripadajuće baze podataka, sadržaj baza, "ponudu" i "tražnju" informacija i slično. Kada se radi o fizičkoj strukturi sistema, o telekomunikacionoj mreži, treba opisati "čvorove" i fizičke karakteristike njihovog povezivanja, funkcije pojedinih čvorova, njihovu opremu, njihov odnos sa institucijama i slično. Da bi se ovakav rečnik podataka projektovao i implementirao, a zatim i da bi se obezbedilo njegovo stalno i efikasno ažuriranje, neophodno je striktno primeniti osnovne postavke standarda, i njihove slobodne delove dodatno standardizovati sa ciljem da se u najvećoj mogućoj meri standardizuje razvoj i funkcionisanje celokupnog SNTIS.

Rečnik SNTIS ima pasivnu i aktivnu ulogu u sistemu. Pasivna uloga se sastoji u tome da Rečnik podataka predstavlja vodič kroz SNTIS i iz njega se mogu dobiti podaci o bazama podataka u sistemu, opisi čvorova mreže SNTI, podaci vezani za naučno istraživački rad i sl. Aktivna uloga Rečnika podataka je da obezbedi ostvarivanje transparentnosti sistema za pretraživanje baza podataka u odnosu na distribuciju podataka. Drugim rečima, Rečnik omogućuje da korisnik pretražuje informacije u mreži SNTIS ne poznavajući baze u kojima se one nalaze niti njihovu prostornu distribuciju.

Koncept i relizacija Rečnika podataka detaljno su opisani u Poglavlju 9 ove monografije.

2.6.2. Plan razvoja bibliotečke mreže [14]

Bibliotečki sistem Srbije obuhvata oko 500 biblioteka i oko 250.000 korisnika. Nacionalne, univerzitetske biblioteke, biblioteke akademija nauka i najznačajnije specijalne biblioteke treba da postanu aktivni učesnici u svim procesima koji su opisani modelom Bibliotečke mreže SNTIS. U već postojeću, delimično kompjuterizovanu bibliotečku mrežu Srbije, koja je formirana u okviru ranije započetog i nezavršenog sistema SNTI SFR Jugoslavije, uključeno je oko 30 biblioteka, među kojima su: Narodna biblioteka Srbije, Biblioteke Univerziteta u Beogradu, Nišu, Novom Sadu i Prištini, Jugoslovenski bibliografski institut, Biblioteka Matice srpske i jedan

broj specijalnih biblioteka. Međutim, nisu uključene važne biblioteke kao što su Univerzitetska biblioteka u Kragujevcu i Biblioteka SANU, kao i znatan broj važnih specijalnih biblioteka, biblioteke naših najvećih instituta "M.Pupin", fakulteta itd. Najveći broj bibliotečkih transakcija , kao što su uzajamna katalogizacija, međubibliotečka pozajmica, dostup do inostranih izvora informacija i domaćih baza podataka, odvija se u 30-tak najvećih biblioteka. Iz analize snimljenog stanja može se takođe videti da jedan broj institucija, koje nisu uključene u bibliotečku mrežu iz 90-tih godina , koristi različitu računarsku i programsku podršku za bibliotečko poslovanje. Među njima je od nacionalnog značaja bibliotečka mreža SANU, koja razvija programski paket za sopstvene potrebe, i veći broj biblioteka koje koriste različite tipove personalnih računara i različite vrste softvera za bibliotečko poslovanje.

Logička struktura bibliotečke mreže je, verovatno, među svim navedenim u ovom radu, jedna od najlakše prepoznatljivih. Njeni elementi su biblioteke, sa hijerarhijom i međusobno uspostavljenim odnosima od ranije. Jedan deo ove mreže je javno dostupan, a u drugom se obavljaju lokalne poslovno-tehničke bibliotečke aplikacije. U javnom delu mreže postoje sledeći servisi:

- katalogizacija primarnih dokumenata ("shared cataloguing");
- "on-line" pretraživanja po više različitim kriterijuma;
- diseminacija informacija iz bibliotečkih i drugih informacionih resursa;
- vođenje centralnih kataloga;
- međubibliotečka pozajmica;
- koordinacija nabavke primarnih dokumenata.

U kategoriju lokalnih bibliotečkih aplikacija spadaju:

- kompjuterizacija nabavke;
- kontrola cirkulacije primarnih dokumenata;
- OPAC ("on-line" javni katalozi).

Kao posebna celina, CD-ROM tehnologija rada se poslednjih godina značajnije koristi u procesu informisanja korisnika, mada je broj CD-ROM baza podataka u Srbiji srazmerno mali u odnosu na njihovu sadašnju svetsku produkciju.

U okviru bibliotečko-informacionog sistema Srbije izgrađuje se nekoliko značajnih, uglavnom bibliografskih baza podataka. U daljem tekstu su pomenute najznačajnije ovakve baze podataka, sa stanovišta njihovog sadržaja i mogućnosti korišćenja:

- Centralni katalog serijskih publikacija sadrži podatke o stranim i domaćim stručnim i naučnim periodičnim publikacijama u fondovima biblioteka u Srbiji. Podaci CKP (strane publikacije) Srbije nalaze se

na lokalnom sistemu u Narodnoj biblioteci Srbije. Za period od 1984-1990. godine baza sadrži ukupno 480.000 kratkih zapisa o 19.000 naslova serijskih publikacija.

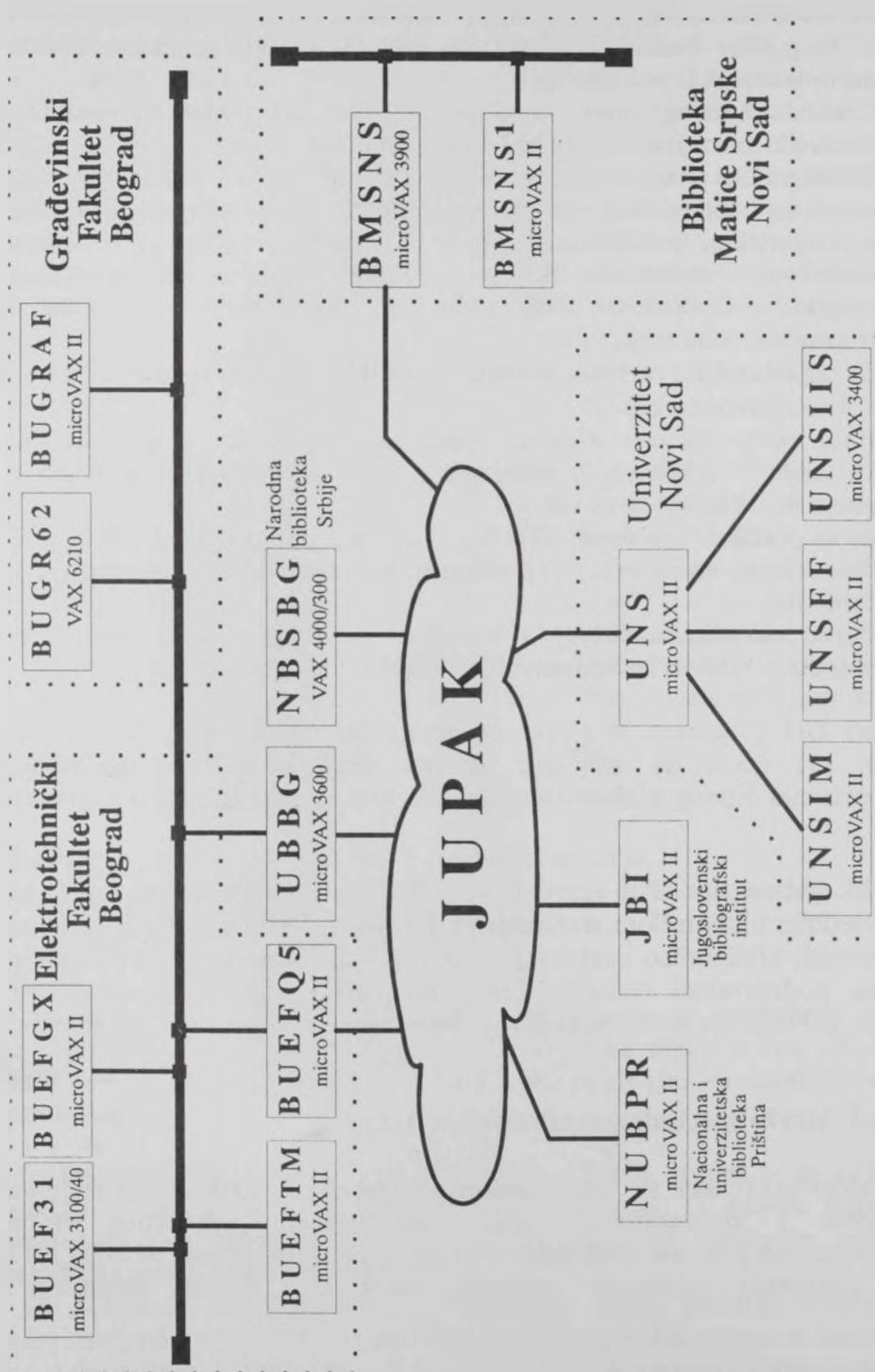
- Centralni katalog monografskih publikacija sadrži podatke o stranim i domaćim monografskim publikacijama u bibliotekama.
- Bibliografska baza nacionalne bibliografije sadrži podatke o nacionalnoj izdavačkoj produkciji. Pored podataka o tekućim monografskim publikacijama, ova baza sadrži i podatke o novim naslovima serijskih publikacija. U Jugoslovenskom bibliografskom institutu se, počev od 1988. godine do danas, nalaze svi podaci o izdavačkoj produkciji Srbije.
- Baza podataka o raritetima sadrži podatke o starim retkim knjigama u fondu Matice srpske.
- Baza podataka o neknjižnoj građi sadrži podatke o geografskim kartama, mapama, atlasima, mikrofilmovima, disketama, gramofonskim pločama i sl.
- Bibliografska baza doktorskih i magistarskih radova sadrži podatke o doktorskim, magistarskim i specijalističkim radovima sa univerziteta u Srbiji itd.
- Pored navedenih, postoji i veći broj lokalnih baza o fondovima pojedinih biblioteka i o standardima bibliotečkog poslovanja.

Na Slici 2.11 prikazana je struktura bibliotečke mreže u Srbiji iz 1992. godine, pri čemu se značajne izmene očekuju tek sa početkom funkcionisanja Novog bibliotečkog sistema koji će biti opisan u nastavku teksta.

Od 1993. godine započet je razvoj nove softverske podrške, koja u skladu sa međunarodnim bibliotečkim standardima, Internet i SNTI standardima treba da obezbedi bibliotečko poslovanje u okruženju otvorenog informacionog sistema, podržavajući različite tipove računarske opreme i operativne sisteme. U Poglavlju 8 ove monografije biće više reči o ovom novom sistemu.

2.6.3. Mreža specijalizovanih sistema [15]

Specijalizovani sistemi su "informacioni sistemi čija se funkcija realizuje na jedinstven i kooperativan način, uključivanjem određenog broja institucionalnih jedinica okupljenih oko zajedničkog programa protoka NTI koje pokrivaju određeno područje nauke ili granu delatnosti".



Slika 2.11. Struktura bibliotečke mreže SNTIS iz 1997. godine

Najznačajnije jedinice specijalizovanih sistema su specijalizovani centri koji obavljaju zadatke sakupljanja, obrade, evaluacije numeričkih i faktografskih podataka, i njihovu diseminaciju. U okviru specijalizovanih sistema ostvaruju se: (a) izgradnja specijalizovanih baza podataka, (b) izrada bibliografija, (c) izrada sekundarnih publikacija, (d) izgradnja baza znanja, (e) retrospektivna pretraživanja, (f) selektivna diseminacija informacija, i (g) obrazovanje korisnika. Načelno, razvijaju ih sami korisnici, u skladu sa svojim potrebama i potrebama ostalih korisnika. Povezuju se po potrebi u mrežu za opštu namenu. U načelu, može se izvršiti klasifikacija na (a) sisteme od zajedničkog interesa, i (b) specijalizovane sisteme, centre i baze podataka po oblastima nauke i po granama delatnosti. Specijalizovani sistemi su nedovoljno razvijeni, i u daljem tekstu nabrojani su samo neki od postojećih sistema u Srbiji, ili sistema koji su u razvoju.

2.6.3.1. Opšti ciljevi izgradnje mreže specijalizovanih sistema SNTIS (SSNTIS)

Opšti ciljevi izgradnje mreže specijalizovanih sistema NTI su:

- Razvoj mreže SSNTIS kao dela SNTIS koja se razvija u skladu sa opštom koncepcijom razvoja SNTIS i koja koristi bazičnu infrastrukturu SNTIS, pre svega računarsko-komunikacionu mrežu;
- Stvaranje preduslova i iniciranje intenzivnog razvoja informacionih proizvoda i usluga za podršku NIR-u i tehnološkom razvoju u Republici (pre svega, baza podataka i softverskih aplikacija za njihovu efikasnu eksploraciju);
- Povezivanje pojedinih specijalizovanih sistema sa odgovarajućim međunarodnim informacionim sistemima;
- Transparentnost SNTIS za različite korisničke potrebe, a posebno za zadovoljavanje informacionih potreba krajnjih korisnika.

2.6.3.2. Ciljevi izgradnje pojedinačnih specijalizovanih sistema

Ciljevi izgradnje pojedinačnih specijalizovanih sistema su sledeći:

- Izgradnja specijalizovanih baza podataka, sa dodatnim vrednostima i distribuiranje specijalizovanih informacija;
- Izgradnja/implementacija programskih paketa za podršku rešavanju konkretnih problema u oblasti (viši oblici obrade informacija i informisanja);
- Uključenje u međunarodne sisteme NTI i međunarodnu razmenu informacija;

- Obezbeđenje dostupnosti NTI velikom broju korisnika na optimalan način (cost/benefit kriterijum), npr. CD-ROM baze, online dostupne baze podataka na hostovima;
- Obuka za korišćenje sistema i širenje kulture korišćenja NTI;
- Podsticaj razvoju i primeni informacionih tehnologija u nauci, tehnologiji i privredi pojedinačnim rešavanjem razvoja konkretnih SSNTI;
- Podsticaj razvoju i primeni informacione i dokumentacione delatnosti;
- Poboljšanje komercijalnog okruženja informacionih i drugih proizvoda i usluga;
- Jačanje transfera informacija između nauke i privrede.

2.6.3.3. Osnovna opredeljenja u realizaciji mreže SSNTIS

- Specijalizovani sistemi i centri NTI se stvaraju slobodno u skladu sa identifikovanim korisničkim i tržišnim potrebama (KORISNIČKA I TRŽIŠNA ORIJENTACIJA);
- Specijalizovani sistemi i centri NTI se podstiču od strane države u onim oblastima koje su od strateškog značaja za razvoj Republike (PRIORITETI);
- Dalji razvoj specijalizovanih sistema i centara NTI mora biti prisutan u onim oblastima gde su već prepoznati i podržani korisnički interesi (KONTINUITET DOSADAŠNJEG RAZVOJA);
- Razvoj SSNTI mora biti u skladu sa standardima usvojenim na nivou SSNTIS (STANDARDIZACIJA);
- Mreža SSNTIS je otvoren sistem u sadržajnom, institucionalnom i tehnološkom smislu (OTVORENOST SISTEMA), to podrazumeva zajednički standard za umrežavanje različitih računara, distribuiranu obradu podataka i razvoj/korišćenje prenosivog softvera;
- Mora biti obezbeđeno zalaganje za razvoj onih konstitutivnih elemenata u mreži koji će obezbeđivati informacione proizvode i usluge za podršku efikasnom rešavanju konkretnih istraživačkih i razvojnih zadataka u određenoj oblasti (REŠAVANJE PROBLEMA);
- Mreža SSNTIS se razvija postupno (POSTUPAN RAZVOJ);
- Za pristup i distribuciju informacija koriste se svi medijumi i alati, na hostu ili korisnikovoj radnoj stanici, tako da specijalizovanje NTI u mreži SSNTI budu lako dostupni i spremni za efikasno korišćenje (DOSTUPNOST I EFIKASNOST KORIŠĆENJA).

2.6.4. Mreža poslovno - tehnoloških informacija (MPTI) [16]

Informacione potrebe privrede, u uslovima sve veće globalizacije tržišta i poslovnih procesa i izrazito jake konkurenkcije, su sve veće ali i selektivnije. Istovremeno, globalizacija poslovnih procesa i tržišta je posledica intenzivnog razvoja računarsko-komunikacionih mreža i servisa za elektronsku razmenu podataka i prisutna je u sve većem broju privrednih delatnosti (bankarstvo, trgovina, itd.). Za uspešno poslovanje u ovakvom okruženju zahtevaju se pouzdane, aktuelne i kompletne informacije o tržišnim subjektima, raspoložive u svakom trenutku, bez obzira na prostornu i vremensku distribuciju poslovnih procesa. Osnovu za zadovoljenje ovih zahteva čini lokalna i globalna mrežna infrastruktura za razmenu poslovno-tehnoloških informacija.

2.6.4.1. Ciljevi i zadaci mreže poslovnih i tehnoloških informacija

Mreža poslovno-tehnoloških informacija ima za cilj da bude podrška obezbeđenju boljih uslova poslovanja privrede Srbije i stvaranju preduslova za njenu veću konkurentnost na globalnom tržstu.

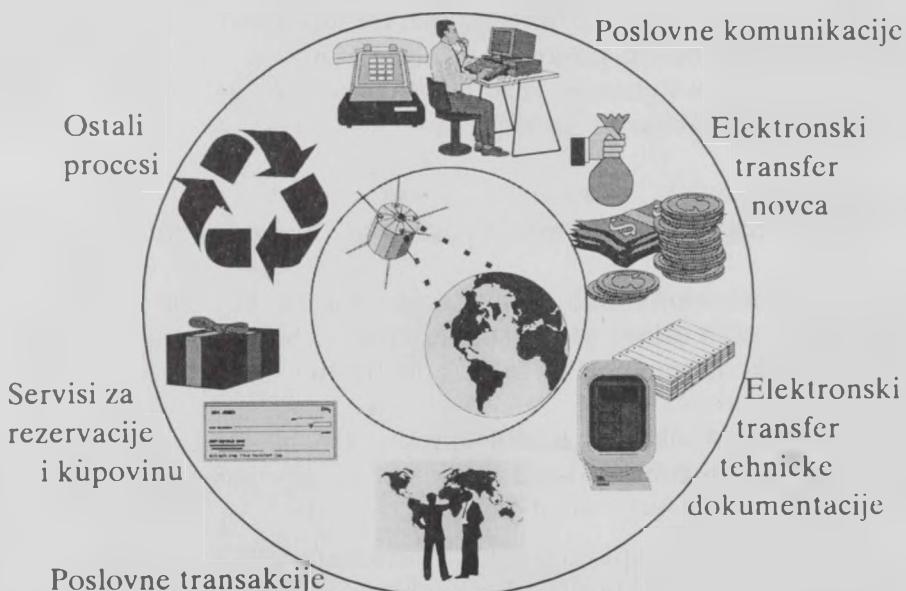
MPTI se definiše kao infrastrukturni sistem namenjen potrebama privrednih i drugih komercijalnih organizacija i asocijacija koja ima zadatak da obezbedi:

- neposrednu komunikaciju između subjekata MPTI,
- razmenu svih vrsta poslovnih podataka i dokumenata,
- primenu standarda usklađenih sa svetskim standardima u ovoj oblasti,
- uslove za integraciju naše mreže u međunarodne mreže sličnog tipa,
- preduslove za brži transfer rezultata IR aktivnosti ka industriji.

MPTI treba da obuhvati subjekte iz raznih sektora (delatnosti). To su: privredne organizacije i asocijacije, naučno-istraživačke i obrazovne organizacije, državni organi i organizacije, bankarske i finansijske organizacije, kao i druge subjekte koji imaju interes da stvaraju i ili koriste mrežne servise i informacione sadržaje dostupne putem MPTI.

Da bi uspešno poslovale na svetskom tržstu, privredne organizacije iz Srbije moraju da nađu mesto u svetskoj podeli rada i tehnološki postanu deo globalnih poslovnih procesa. To zahteva da podaci budu prenošeni između svih sektora (komercijalni, administrativni, razvojni, proizvodni, itd.) i

korisnika, bez obzira na tehničke različitosti. Drugim rečima, u Srbiji se mora razvijati infrastruktura koja će omogućiti poslovanje sa svetom u informatizovanom poslovnom okruženju. Jačanje računarsko-komunikacione infrastrukture na nacionalnom nivou je preduslov za elektronsku razmenu podataka (EDI) i efikasno povezivanje poslovnih partnera (proizvođača, kupaca, dobavljača i finansijskih institucija; špeditera i proizvođača itd. (slika 2.12)



Slika 2.12. Globalizacija poslovnih procesa

Osnovnu infrastrukturu MPTI čini računarsko-komunikaciona mreža sa odgovarajućim skupom mrežnih servisa, po pravilu Internet tipa, nad kojom se razvijaju i nude i druge korisnički orijentisane specijalizovane usluge. Koristeći bazičnu mrežnu infrastrukturu moguće je razviti i niz drugih usluga koje su orijentisane ka specijalizovanim grupama korisnika. Svetska praksa pokazuje da se preko MPTI posebno koriste sledeći servisi:

- Pristup i pretraživanje baza podataka tehnoloških, tržišnih i drugih poslovnih informacija;
- Razmena komercijalnih poslovnih dokumenata u okviru pojedinih korisničkih grupa;

- Podrška ponudi i potražnji u oblasti roba, usluga, hartija od vrednosti itd.;
- EDI (elektronska razmena dokumenata) servisi sa dodatim vrednostima;
- Podrška telematskim servisima u pojedinim delatnostima (prevoz putnika i robe, hotelske usluge, itd.).

Korišćenje navedenih servisa (opšti mrežni i specijalizovani) otvara mogućnosti za značajne poslovne promene koje su često nametnute preduzeću da bi uopšte opstalo na svetskom tržištu.

Polazeći od ciljeva i zadataka MPTI definišu se osnovna opredeljenja pri razvoju i organizovanju MPTI. To su:

- razvoj mreže koji je marketinški orijentisan,
- razvoj mreže vođen potrebama korisnika,
- razvoj mreže koji će omogućiti nove poslovne procese (u samim preduzećima, i na širem poslovnom planu),
- razvoj mreže koji će biti usklađen sa međunarodnim okruženjem.

Odrednice razvoja infrastrukture mreže poslovno-tehnoloških informacija, s obzirom na korisnike kojima je namenjena, su i sledeće:

- pouzdan rad i dostupnost servisa u svakom trenutku,
- razvoj mreže koji će biti tržišno vođen,
- uključenje u mrežu u skladu sa međunarodnim standardima.

2.7. PERSPEKTIVE DALJEG RAZVOJA MREZE SNTIS

Mreža Sistema naučnih i tehnoloških informacija predstavlja složen otvoreni informacioni i računarsko komunikacioni sistem usklađen sa međunarodim standardima. Polazeći od toga i dalji razvoj ovoga sistema treba da bude usklađen sa svetskim kretanjima u ovoj oblasti.

U funkcionalnom smislu razvoj mreže SNTIS treba da u bliskoj budućnosti preraste u jedan kooperativni sistem u oblasti naučno-istraživačkog i razvojnog rada. Kooperativni sistemi su informacioni sistemi i tehnologije za podršku interaktivnog rada grupa, pojedinaca i organizacija bez obzira na njihovu prostornu udaljenost. Najčešće korišćenje ovih sistema je u oblasti upravljanja i odlučivanja, naučno istraživačkog rada i obrazovanja. Kao tehnološka osnova sistema koristi se neki od Distribuiranih sistema za podršku rada grupa (Distributed Group Support System). Podrška može

biti za direktnu komunikaciju učesnika u radu ili indirektnu podršku komuniciranju i procesiranju dokumenata. Komunikacione tehnologije koje pružaju podršku ovakvim sistemima mogu biti različitog tipa: elektronske ili video konferencije sa servisima kao što su E-mail, voice-mail, kao i takozvani "work flow" upravljački sistemi. Osnovni softverski zadaci kod ovakvih sistema su da pored podrške komunikacije obezbedi podršku razrešenja konflikta koji nastaje u grupnom radu, ostvari povratnu spregu učesnika u grupnom radu i obezbedi veze sa konvencionalnim bazama podataka i programima.

U tehnološkom smislu razvoj SNTIS kretaće se ka u svetu aktuelnom informacionom highway-u. U skorije vreme su se mnoge diskusije u javnosti usredsredile na kreiranje napredne komunikacijske mreže sposobne da drži velike količine informacija u različitim oblicima i koja će biti pristupačna širokom krugu korisnika. Neka postojana definicija za ovu mrežu još nije ustanovljena, ali u zamenu za to popularnost je stekla metafora koja je opisuje: "informacioni (super) autoput".

U organizacionom smislu SNTIS treba da se razvija kao otvoreni sistem sa maksimalno mogućom autonomijom u razvoju svojih pojedinih segmenata i koordiniranom razvoju preko izrade "dejure" i "defacto" standarda, kao i praćenjem funkcionisanja sistema. Za realizaciju planiranih funkcija, neophodno je stabilno rešiti izvršnu organizaciju formiranjem Agencije ili tela koje će takve poslove opertivno realizovati. Podsticaj države u delu korišćenja mreže za potrebe naučnoistraživačkih i obrazovnih organizacija i usmeravajuća uloga države u istraživanju i razvoju predstavljaju ključno opredeljenje za povećanjem efekata korišćenja mreže. Povezivanje naučno istraživačkog rada u oblasti informacionih tehnologija sa implementacijom rezultata na mreži SNTI stvaraju se uslovi za uključenje ne samo nauke već i privrede u globalne svetske informacione tokove.

LITERATURA

- [1] Lazarević, B., Kutlača, Đ., Ivković M., Radenković, B., Vitas, D., Konstantinović, Z., Koncepcija razvoja Sistema naučnih i tehnoloških informacija u Republici Srbiji, Ministarstvo za nauku i tehnologiju, Beograd, 1991.

- [2] Paunović, Đ., Lazarević, B., Ivković, M., Simić N., *Stanje, pravci i perspektiva razvoja mreže i sistema naučnih i tehnoloških informacija*, časopis TELEKOMUNIKACIJE, XLII, br.1, pp 3-20, 1994.
- [3] Paunović, Đ., Simić N., Oršolić, Z., Nerandžić, D., Tasić, M., Jovančević, R., Radojkov, S., Plan razvoja telekomunikaciono-računarske mreže "Beonet" Univerziteta u Beogradu, uz saradnju stručnjaka JP PTT "Srbija", Univerzitet u Beogradu, Elektrotehnički fakultet, Beograd 1992.
- [4] Paunović, Đ., Simić, N., Krajnović, N., Todorović, B., Predlog idejnog rešenja FDDI mreže u Beogradu, Elektrotehnički fakultet, Beograd 1995.
- [5] Simić, N., Paunović, Đ., Korunović, D., Oršolić, Z., Nerandžić, D., Tasić, M., Plan razvoja magistralnog povezivanja telekomunikaciono-računarske mreže SNTIS, uz saradnju stručnjaka JP PTT "Srbija", Univerzitet u Beogradu, Elektrotehnički fakultet, Beograd 1992.
- [6] Paunović, Đ., Simić, N., Boljević, I., Krajnović, N., Mihaljević, V., Tehničko rešenje lokalne računarske mreže Studentskog trga - LOSTUD i njena integracija u Akademsku mrežu Beograda - BEONET, Elektrotehnički fakultet u Beogradu, Beograd 1995.
- [7] Korunović, D., Veljković, D., Jovev, Lj. Stojanović, M., Popović, I., Petković, I., Dragić, S., Stojković, B., Dašić, R., Lokalne računarske mreže Univerziteta u Nišu, Univerzitet u Nišu, Niš 1992.
- [8] Lokalna mreža Univerziteta u Kragujevcu, Univerzitet u Kragujevcu, radni materijal, Kragujevac 1991.
- [9] Lokalna mreža Univerziteta u Novom Sadu, Univerzitet u Novom Sadu, 1995.
- [10] Lokalna mreža Univerziteta u Prištini, Univerzitet u Pristini, 1995.
- [11] Paunović, Đ., Ivković, M., *Sistem naučnih i tehnoloških informacija 1994.g.*, rad po pozivu, časopis INFO, br.6, pp.20-24, 1994.
- [12] Paunović, Đ., *Telekomunikacioni aspekti razvoja Sistema naučnih i tehnoloških informacija Srbije i Jugoslavije*, uvodno predavanje i pregledni rad po pozivu, Telfor'94, Beograd, 1994.
- [13] Lazarević, B., Kutlača Đ., Korunović, D., Ivković, M., Plan razvoja upravljačke mreže SNTIS, Fakultet organizacionih nauka, Univerziteta u Beogradu, Beograd 1992.

- [14] Surla, D., i grupa autora, Plan razvoja bibliotečke mreže, PMF, Novi Sad, 1992.
- [15] Pantelić, S., Korunović, D., i grupa autora, Mreža specijalizovanih sistema, Ministarstvo za nauku i tehnologiju, Beograd, 1993.
- [16] Radenković, B., Pantelić, S., Ivković, M., *Mreža poslovno-tehnoloških informacija*, YU INFO, Brezovica, 1995.

Poglavlje 3.

KARAKTERISTIKE OTVORENIH SISTEMA

Sistem naučno-tehnoloških informacija je koncipiran kao otvoreni heterogeni sistem u kome se pojedini njegovi delovi autonomno razvijaju u skladu sa sopstvenim potrebama, opštom koncepcijom i sopstvenim i zajedničkim planovima. Jedinstvo ovoga sistema se ne ostvaruje strogim centralizovanim upravljanjem, već razvojem sistema zasnovanim na standardima, uz koordinaciju specifičnih zajedničkih aktivnosti. U ovom poglavlju se prikazuje opšta arhitektura otvorenih informacionih sistema, na osnovu koje se definiše skup standarda na kojima je SNTIS razvijan.

Pri definisanju arhitekture otvorenih sistema neophodno je izvršiti njihovu dekompoziciju na skup komponenti koji je lakše specificirati i razvijati. Postoje različiti načini na koje se može izvršiti dekompozicija ovakvih sistema. Ovde je data dekompozicija koja se uglavnom bazira na modelu CCTA (Central Computer and Telecommunication Agency) [1]. Referentni model otvorenog sistema SNTIS predstavljen je na slici 3.1. Uopšteno, komponente na vrhu slike predstavljaju komponente preko kojih korisnik razvija i koristi sistem, dok one na nižim nivoima pružaju neophodne servise za razvoj i funkcionisanje sistema, sve do hardvera i fizičkih komunikacija, preko kojih se sistem realizuje.

Za kasnije kooperativno funkcionisanje otvorenog sistema jedan od značajnih preduslova je standardna metodologija razvoja pojedinih softverskih (aplikativnih) sistema. Pojedine vladine organizacije mnogih zemalja su usvojile detaljne standardne metodologije razvoja IS koje moraju da primenjuju i sve druge institucije koje za te vlade razvijaju pojedine softverske sisteme [2,3,4,5]. Ovi standardi obično obuhvataju sistemsку

analizu, modeliranje procesa, modeliranje podataka i specifikaciju aplikacija i dinamike sistema.



Slika 3.1. Komponente otvorenih IS

Kao standardna metodologija razvoja SPI prihvaćena je Objektno orijentisana transformaciona metoda razvoja koja se ukratko opisuje u sledećem poglavlju, a detaljno u [5]. Ova metodologija se koristi, u ostalim poglavljima, za specifikaciju pojedinih podsistema. Standard za specifikaciju nekog aplikativnog programskog sistema zasniva se na Strukturnoj sistemskoj analizi i Proširenom modelu objekti-veze [5]. U ovom modelu baza podataka se specificira preko strukture modela objekti-veze i strukturnih i vrednosnih dinamičkih pravila integriteta. Zbog toga, baza podataka ovde ne predstavlja samo statičku strukturu, već uključuje i specifikaciju osnovnog dinamičkog ponašanja sistema, odnosno specifikaciju odziva sistema na osnovne operacije ažuriranja. Ovi odzivi su osnova za apstraktan (transparentan u odnosu na konkretan sistem za upravljanje bazom podataka) razvoj programa za održavanje baze podataka. Prošireni model objekti-veze definiše i potpun skup operacija preko kojih je moguće specificirati i aplikacije aplikativnog softverskog sistema. Aplikacije se

specifikuju preko koncepta složenog ulazno-izlaznog objekta koji opisuje način komunikacije korisnika sa sistemom. Koncept složenog ulazno-izlaznog objekta definisan je tako da omogući savremen objektno-orientisan razvoj aplikacija, korišćenje savremenih objektno orientisanih jezika i baza podataka, kao i razvoj standarda za korisnički interfejs.

Pored standardne metodologije za specifikaciju sistema neophodno je razviti i standarde za implementaciju sistema u različitim tehnološkim okruženjima, podrazumevajući prikazani referentni model otvorenog sistema. Detaljne standarde implementacije treba da razviju organizacije koje izrađuju pojedine podsisteme u različitim okruženjima, a preporuke za izgradnju ovih standarda treba posebno razraditi.

Bitna komponenta prikazanog referentnog modela je organizacija i upravljanje realizacijom i eksploatacijom sistema. Ova komponenta obuhvata pravila upravljanja autonomnim SPI, fizičkom mrežom, definisanje standarda, planova razvoja i slično. O organizaciji SNTI i strategiji njegovog razvoja diskutovano je u prethodnom poglavlju.

Komponente SPI se mogu izgraditi korišćenjem gotovih paketa programa ili razvojem sopstvenih softverskih sistema. Za razvoj sopstvenih softverskih sistema preporučuje se korišćenje CASE alata. Kao standardni CASE za razvoj SNTIS usvojen je ARTIST [6] koji u potpunosti podržava standardnu metodologiju specifikacije sistema. Pored specifikacije sistema ARTIST delimično podržava i transformaciju specifikacije u pojedina implementaciona okruženja.

U daljem razvoju je neophodno u što je moguće većoj meri koristiti standardnu kolekciju alata za razvoj softvera kakvu preporučuje ESPRIT (European Strategic Programme for Research and Development in Information Technology) projekat PCTE (Portable Common Tool Environment). Korišćenje ovakvog standarda, s jedne strane omogućuje konzistentan razvoj različitih aplikacija, jer se sve one zasnivaju na istim mehanizmima komunikacije sa bazom podataka i korisničkim interfejsima, a sa druge, jednostavnije prenošenje aplikacije iz jedne u drugu okolinu koja podržava PCTE.

Rečnik podataka SNTIS je osnova za ostvarivanje integrisanosti, odnosno kooperativnosti ovog otvorenog distribuiranog IS. Struktura ovog rečnika podataka je izvedena iz ANSI standarda Information Resource Dictionary System (IRDS) [7] i detaljno je specificirana u devetom poglavlju. Ovde treba napomenuti da rečnik podataka u SNTIS treba da ima dvostruku ulogu, "pasivnu" kao baza podataka koja opisuje ostale objekte u SNTIS i

služi kao vodič kroz SNTIS za razne vrste korisnika i projektanata, i "aktivnu" (interpretativnu), koja je osnova funkcionisanja pojedinih komponenti sistema, a posebno osnova za ostvarivanje njihove potpune transparentnosti u odnosu na distribuciju, u poslednjoj fazi razvoja celokupnog SPI. Aktivna uloga rečnika se sastoji u tome da aplikacije treba da budu "vođene rečnikom" koji će aplikaciji, pored ostalog, prvenstveno pružiti informacije o tome na koji je način apstraktna specifikacija baze podataka i programa implementirana u konkretnim tehnološkim okruženjima.

Sledeća dva nivoa prikazanog modela čine apstrakcije i standardi koji treba da obezbede prenosivost i transparentnost sistema u odnosu na konkretnu hardversku i softversku platformu. Ovde su posebno interesantne apstrakcije operativnih sistema i sistema za upravljanje podacima, kao i standardi za razmenu podataka. Primer apstrakcije operativnih sistema je POSIX koji predstavlja standardni interfejs prema različitim operativnim sistemima. Iako je ovaj standard orientisan prvenstveno prema operativnom sistemu Unix, implementiran je i za neke druge operativne sisteme kao što su DEC-ov OpenVMS i Microsoft-ov Windows NT. Kao primer apstrakcije sistema za upravljanje bazom podataka može se navesti ODBC (Open Data Base Connectivity) firme Microsoft. Ovaj proizvod kao i bilo koja druga apstrakcija upravljanja podacima treba da omogući jedinstvenu komunikaciju programa sa podacima iz baze podataka bez obzira na sistem za upravljanje bazom podataka. Samim tim se ostvaruje i, za transparentnost SPI bitna mogućnost, da jedna aplikacija radi sa većim brojem različitih servera baza podataka. Pored apstrakcije sistema za upravljanje bazom podataka trenutno se dosta radi i na standardizaciji samih sistema za upravljanje bazom podataka, što se posebno odnosi na standardizaciju SQL jezika za rukovanje podacima.

Sa tačke gledišta razmene podataka između računarskih sistema, na najvišem nivou apstrakcije definišu se standardi razmene u različitim oblastima. Primeri standarda razmene podataka su: ODA (Office Document Architecture) koji predstavlja standardni format za procesiranje teksta i "desktop publishing" fajlova, dozvoljava da se dokument iz jednog "kancelarijskog sistema" prebaci u drugi zajedno sa kodovima za formatiranje, opisom fontova, grafikom i slično; SGML (Standard Generalized Markup Language) na kome se bazira struktuiranje dokumenata i njihova razmena u SNTIS (Poglavlje 7); EDIFACT koji je od posebnog značaja za poslovne komunikacije; UNIMARC koji se koristi u bibliotečkom podsistemu SNTIS (Poglavlje 8).

Jedna od karakteristika razvoja savremenih otvorenih IS je tendencija da se, kao komponente sistema, sve više koriste gotovi programski paketi i aplikacije. Ovom trendu posebno je doprineo razvoj "posrednika" (Request Broker) koji omogućavaju interoperabilnost komponenti (aplikacija) različitih proizvođača. Trenutno postoje dve specifikacije ovakvih posrednika: CORABA (Common Request Broker Architecture) konzorcijuma Object Management Group i COM (Component Object Model) firme Microsoft.

Dva najniža sloja čine operativni sistemi, računarska mreža, hardver i telekomunikaciona mreža, sa dobro poznatim standardima na kojima se zasniva savremeni razvoj informacionih tehnologija.

LITERATURA

- [1] Informatics in Europe, *Standards and Open Systems*, pp.114-117.
- [2] Federal Information Processing Standard Publications, Information Definition for Function Modelling (IDEF0), U.S. Department of Commerce, FIS PUB 183, Decembre 1993.
- [3] Federal Information Processing Standard Publications, Integrated Information Support System (IISS), Volume 5 - Common Data Model Subsystem, Part4 - Information Modelling Manual, IDEF! Extended, Decembre 1985.
- [4] Bruce, T.A., *Designig Quality Database with IDEF1X Information Models*. New York: Dorset House Publishing, 1992.
- [5] Lazarević, B., Nešković, S., Bataveljić, P., *Objektno-orjentisana transformaciona metoda razvoja informacionih sistema*. Publikacija Laboratorije za informacione sisteme FON-a, ILIS1 1-1993, Beograd, 1992.
- [6] Nešković, S., Priručnik za CASE alat ARTIST, FON, 1995, interna publikacija.
- [7] American National Standard for Information Systems, Inormation Resource Dictionary Systems, ANSI X3.138-1988.

Poglavlje 4.

METODOLOGIJA

Vec je u prethodnim poglavljima naglašena potreba da se u razvoju otvorenih sistema, kakav je i SPI, primenjuje standardna metodologija razvoja, jer se time "otvara" razvoj sistema i omogućava se povezivanje posebno razvijenih podsistema i na najapstraktnijem nivou, nivou specifikacije sistema i u najranijim fazama njegovog razvoja. Ovo je posebno značajno sa tačke gledišta budućeg "adaptivnog održavanja" celokupnog sistema, izmena u sistemu koje je neophodno izvršiti zbog promene u tehnološkom okruženju u kome se sistem realizuje. U čestim promenama informacione tehnologije, samo će najapstraktnija specifikacija jednog otvorenog informacionog sistema ostati nepromenjena.

U ovom poglavlju prvo se diskutuju karakteristike koje jedna ovakva metodologija treba da poseduje, a zatim se ukratko opisuju njene osnovne komponente: objektno-orientisana specifikacija zasnovana na modelu procesa i modelu podataka, postupci projektovanja baze podataka i programa i principi implementacije nezavisni od konkretnog implementacionog okruženja. Na kraju se eksplicitno navode koraci standardne metodologije koji su opisani u prethodnim delovima

4.1. OPŠTE KARAKTERISTIKE STANDARDNE METODOLOGIJE RAZVOJA INFORMACIONIH SISTEMA

Prilikom definisanja, odnosno izbora jedne standardne metodologije razvoja informacionih sistema (IS) neophodno je analizirati njene karakteristike sa teorijske (koliko ona prati savremeni razvoj modela, metoda i tehnika u oblastima računarstva koje obuhvata) i pragmatske (koliko će ona biti prihvatljiva širokom krugu projektanata i korisnika) tačke gledišta. Pri tome je posebno značajno istaći šta se pod pojmom "standardna" podrazumeva,

odnosno koje sve vrste projektanata i korisnika standard treba da obuhvati. U prethodnim poglavljima je istaknuta potreba da se definiše "državna standardna metodologija", metodologija koja će biti primenjena za razvoj svih sistema u državnoj upravi i prikazana su iskustva kako se ovako definisan državni standard širi na druge privredne i komunalne sisteme. Ako se polazi od pretpostavke da metodologija koja je primenjena u ovom projektu treba da postane "državna", zbog potrebe da se ona široko primenjuje, pri definisanju njenih karakteristika prednost treba dati pragmatskim aspektima, ne zanemarujući, zbog budućeg razvoja, njenu savremenu teorijsku zasnovanost.

Teorijski aspekti. Problem savladavanja složenosti, osnovni problem u razvoju informacionih sistema (IS), u različitim pristupima i metodama razvoja IS napada se korišćenjem dva fundamentalna metodološka principa: (1) podela celokupnog razvoja na faze, odnosno model "životnog ciklusa" IS koji se koristi i (2) dekompozicija složenog sistema na komponente (podsisteme) savladive složenosti. Različite metode razvoja IS mogu se uporedno analizirati sa tačke gledišta načina na koji koriste ova dva metodološka principa. Na sadašnjem stepenu razvoja ovih metoda značajnu prednost treba dati transformacionom (operacionalnom) modelu "životnog ciklusa" i objektno orijentisanoj dekomplicaciji sistema [1].

Transformacioni pristup razvoju IS sastoji se od sledećih koraka [1]:

- (1) Definisanje formalne specifikacije P datog IS;
- (2) Primena neke transformacije T na formalnu specifikaciju P da bi se dobila nova specifikacija P;
- (3) Ponavljanje koraka (2) dok se ne dobije verzija P koja zadovoljava sve zahteve i zadate uslove (ograničenja) implementacije.

Svaka izvršljiva (na računaru) specifikacija P pretstavlja jedan prototip IS koji se može iskoristiti za validaciju specifikacije i verifikaciju posmatrane implementacije. Transformacioni pristup dovodi u toku razvoja IS do formiranja jedne serije prototipova, pa samim tim uključuje u sebe i sve prednosti prototipskog razvoja IS.

Za realizaciju transformacionog pristupa očigledno je neophodan skup alata koji čine: CASE (Computer Aided Software Engineering) sistem sa bogatim "transformacionim mogućnostima", odnosno mogućnošću generisanja baze podataka i pojedinih programa ili njihovih delova, različiti generatori aplikacija i ostali alati za razvoj prototipova. Savremeni CASE sistemi podržavaju standardne metodologije razvoja pojedinih država. O CASE

alatu koji je korišćen u Projektu, ukratko se diskutuje na kraju ovog poglavlja.

Osnovne prednosti objektne dekompozicije sistema (u odnosu na tradicionalni funkcionalni pristup) su [2,3]:

- Objekti sistema pretstavljaju najstabilniju osnovu dekompozicije sistema;
- U modeliranju sistema se ne koriste samo algoritamske (dinamičke) apstrakcije, već i apstrakcije podataka (statičke apstrakcije) što rezultuje u prirodniji i prihvatljiviji model;
- "Učaurenje" modula dobijenih objektnom dekompozicijom je gotovo idealno, svaki modul se zaista može koristiti kao "crna kutija", bez ikakvih informacija o njegovoj implementaciji;
- Apstrakcija agregacije objekata, omogućuje da se, s jedne tačke gledišta, jedan složeni objekat posmatra kao celina i sa njim kao celinom jednostavno manipuliše, a sa druge, da se on posmatra kao struktura međusobno povezanih komponenti, objekata koji ga čine. Ponovno posmatranje neke komponente kao agregacije jednostavnijih objekata, daje jedan sistematizovan pristup dekompoziciji sistema, jednu vrstu hijerarhijskog opisa sistema.
- Apstrakcija generalizacije, odnosno generalizaciona hijerarhija tipova objekata, daje drugi način hijerarhijskog opisa sistema koji uvodi koncepte "nasleđivanja" osobina i operacija od tipova objekata na višem nivou u posmatranoj generalizacionoj hijerarhijskoj grani, što u fazi implementacije omogućava formalizovan pristup korišćenju već razvijenih modula i time revolucionarno povećava produktivnost u razvoju softvera.

Metodologija koja je u Projektu korišćena i koja se predlaže za standard, pokušava da kombinuje osnovne prednosti transformacionog "životnog ciklusa" i objektne dekompozicije sistema. Pri tome, takođe, pokušava da pomiri i dva na izgled oprečna zahteva, da u fazi analize sistema dovede do formalne specifikacije sistema (kako to transformacioni pristup zahteva), a da taj visok stepen formalizacije ne onemogući produktivnu saradnju projektanata i korisnika sistema.

Pragmatski aspekti. Sa tačke gledišta korisnika, po objektnoj paradigm, celokupan IS pretstavlja jedan objekat sa kojim korisnik treba da komunicira preko definisanih operacija, odnosno poruka koje okidaju dati skup operacija nad posmatranim objektom. Jedan informacioni sistem ima

mnoštvo korisnika, a svaki od njih mnoštvo operacija. Da bi i na ovom nivou sistem ostao učauren, ove operacije su međusobno nezavisne u smislu da jedna drugu ne indukuju, već svaka nezavisno menja stanje sistema. Međusobni uticaj rezultata operacija, ostvaruje se isključivo preko stanja sistema. Postavlja se pitanje kako identifikovati i opisati mnoštvo operacija i semantiku svake operacije posebno?

Istovremeno jasan i detaljan opis ovoga skupa operacija moguće je dati samo preko neke hijerarhije operacija. Do ove hijerarhije operacija je, u komunikaciji sa korisnikom, najlakše doći preko koncepta funkcije sistema, odnosno procesa u sistemu.

Koncept učaurenih kooperativnih objekata nije koncept na kome bi mogla da se zasniva komunikacija sa korisnikom, a ne postoji gotovo nikad nijedan drugi izvor znanja struktuiran na taj način. Mogućnosti da se objekti identifikuju i definišu iz tekstova koji opisuju pojedine sisteme, što kao metodu preporučuje više autora, gotovo da izgleda smešno. Zbog toga struktturna sistemska analiza koja se bazira na konceptu procesa (funkcije) ostaje kao najpogodniji metod za definisanje skupa korisnickih operacija.

Konvencionalna struktturna sistemska analiza [4,5] zasniva se na Dijagramima toka podataka (DTP), jednostavnom grafičkom prikazu procesa, tokova i skladišta podataka koje oni koriste, odnosno generišu i objekata van sistema sa kojima, preko tokova podataka, ostvaruju komunikaciju. DTP i pravila njihovog formiranja daju jednostavan način dekompozicije i hijerarhijskog opisa sistema. Međutim, struktturnu sistemsku analizu treba primenjivati po metodološkim preporukama i formalizmu detaljno opisanom u [6] koji prilagođava klasične postavke ovakvom objektno-orientisanom pristupu. Struktturna sistemska analiza treba da rezultuje u objektno-orientisanu specifikaciju informacionog sistema (softvera), da bi se u daljim fazama razvoja primenile sve prednosti ovog savremenog pristupa.

4.2. STRUKTURNΑ SISTEMSКА ANALIZA (SSA) I OBJEKTNA SPECIFIKACIJA SOFTVERA

U ovom delu će se prvo prikazati osnovni postulati konvencionalne SSA, a zatim će se definisati modifikacije i proširenja koji dovode do objektno - orientisane specifikacije softvera.

4.2.1. Konvencionalna strukturalna sistemska analiza

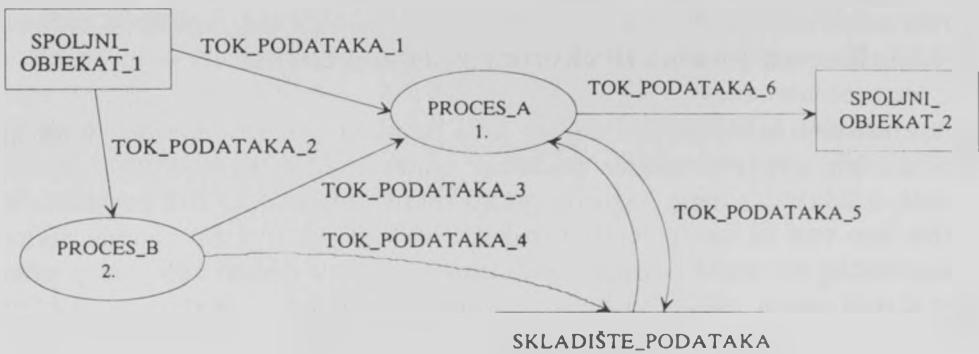
SSA posmatra informacioni sistem kao funkciju (proces obrade) koja na bazi ulaznih, generiše izlazne podatke. Ulazni podaci se dovode u proces obrade, a izlazni iz njega odvode preko tokova podataka. Tok podataka se tretira kao vod ili kao pokretna traka kroz koji stalno teku ili koja stalno nosi podatke na najrazličitijim nosiocima - papirni dokumenti, niz poruka koje čovek unosi preko tastature terminala, "paket" informacija dobijen preko neke telekomunikacione linije ili slično. Imajući u vidu zahtev da specifikacija treba da se osloboди svih implementacionih detalja od interesa su samo sadržaj i struktura ulaznog toka, a ne i medijum nosilac toka.

Izvori ulaznih, odnosno ponori izlaznih tokova podataka mogu biti objekti van IS koji sa IS komuniciraju i koji se u SSA nazivaju interfejsi, drugi procesi u sistemu, ili tzv. skladišta. Skladišta podataka se posmatraju kao "tokovi u mirovanju", odnosno odloženi, akumulirani tokovi, različite vrste evidencija, arhiva, kartoteka i datoteka. I za skladišta kao i za tokove od interesa su isključivo njihov sadržaj i struktura.

Osnovni koncepti za specifikaciju IS u SSA su, znači, funkcije, odnosno procesi obrade podataka, tokovi podataka, skladišta podataka i interfejsi. Njihov međusobni odnos se prikazuje preko dijagrama toka podataka koji prikazuje vezu interfejsa, odnosno skladišta kao izvora ili ponora podataka, sa odgovarajućim procesima, kao i međusobnu vezu procesa. Na slici 4.1 prikazan je jedan opšti primer dijagrama toka podataka koji ima za cilj da uvede sledeće grafičke simbole:

- (i) krug ili elipsa pretstavlja funkciju ili proces obrade podataka,
- (ii) pravougaonik pretstavlja interfejs,
- (iii) usmerena linija pretstavlja tok podataka,
- (iv) (dve paralelne linije ("otvoreni" pravougaonik) pretstavlja skladište podataka.

Očigledno je da se jedan IS sastoji iz mnoštva procesa, interfejsa, tokova i skladišta podataka. Specifikacija IS treba da bude potpuna (detaljna) i jasna. Kada bi se jedan sistem detaljno opisao i prikazao jednim dijagrame toku podataka, dobio bi se veoma nejasan opis sistema, paukova mreža procesa, tokova, skladišta i interfejsa. Istovremeno detaljan i jasan opis sistema zahteva opis na "različitim nivoima apstrakcije", odnosno hijerarhijski opis u kome se na višim nivoima sistem opisuje opštije, a na nižim, postepenim i organizovanim uvođenjem detalja, potpuno i detaljno.



Slika 4.1. Osnovni koncepti DTP

Hijerarhijski opis sistema u tehnici dijagrama tokova podataka se svodi na to da se na višim nivoima definišu globalniji procesi, a da se zatim svaki takav globalni proces, na sledećem nižem nivou, pretstavi novim dijagramom toka podataka. Ovaj princip hijerarhijske dekompozicije sistema prikazan je na Slici 4.2.

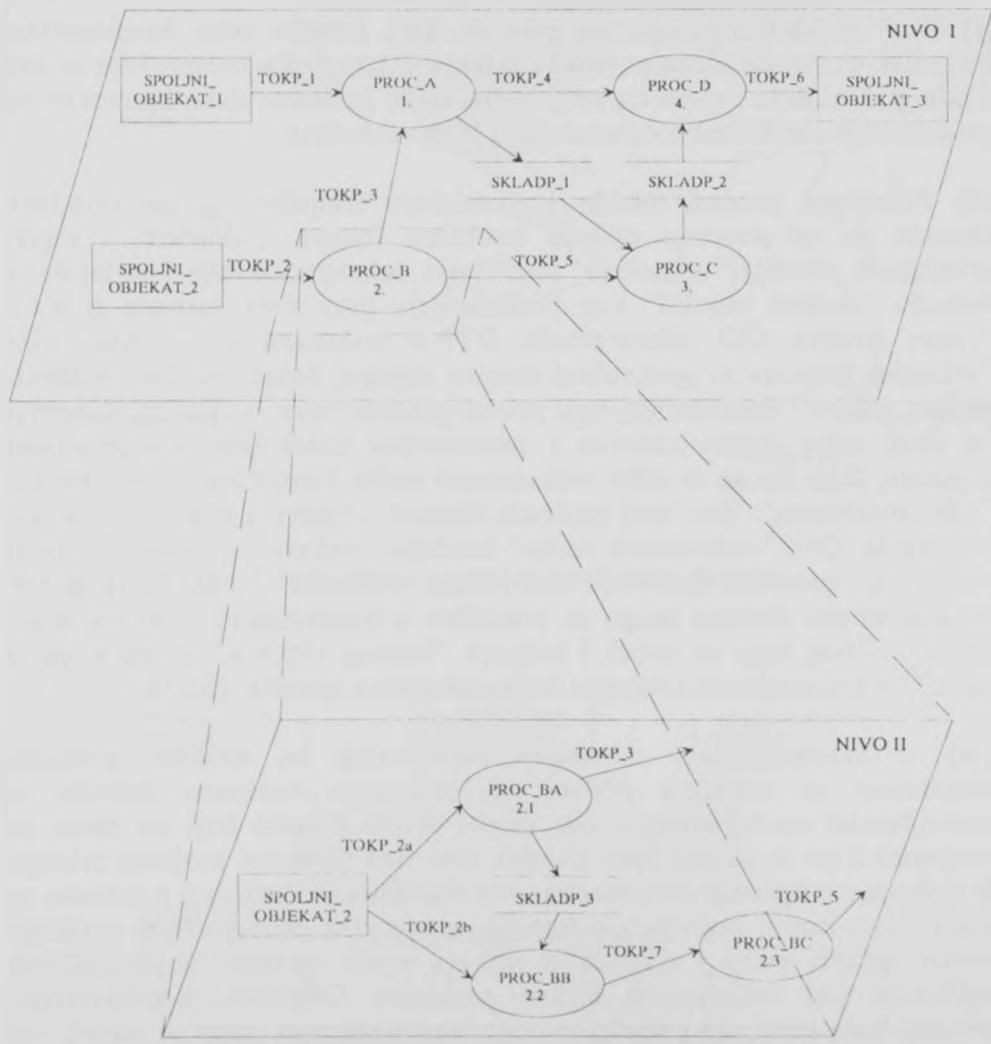
Pored nekih opštih uputstava o načinu dekompozicije i formiranja DTP (na primer "pravilo 7 ± 2 ", koje kaže da se jedan proces treba da dekomponuje, odnosno na jednom DTP treba da pretstavi sa 7 ± 2 procesa) osnovno pravilo koje obezbeđuje konzistentnost dekompozicije je pravilo balansa tokova ("zakon o održanju tokova"): Neto ulazni i neto izlazni tokovi u jedan proces na DTP-u viseg nivoa moraju biti jednak neto ulazima i neto izlazima za celokupan odgovarajući DTP na nižem nivou. Drugim rečima, svi ulazni i izlazni tokovi procesa na jednom nivou moraju biti ulazni i izlazni tokovi u DTP u koji je posmatrani sistem dekomponovan i nijedan novi ulazno-izlazni tok se ne može pojaviti na ovom DTP-u. Napomenimo da se pravilo realizuje i preko rečnika podataka u kojem je prikazana eventualna dekompozicija tokova.

Dijagram toka podataka na vrhu ovakve hijerarhije naziva se dijagram konteksta, a procesi na najnižem nivou (procesi koji se dalje ne dekomponuju) nazivaju se primitivni procesi.

Konvencionalna SSA rezultuje u funkcionalnu specifikaciju IS koju čine:

- (1) Hijerarhijski organizovan skup dijagrama toka podataka;
- (2) Rečnik podataka koji opisuje sadržaj i strukturu svih tokova i skladišta podataka. Za opis strukture tokova i skladišta koristi se specifična sintaksa [4,5,6];

(3) Specifikacija logike primitivnih procesa. Za specifikaciju logike primitivnih procesa moguce je koristiti razlicite alate, od Dijagrama toka programa, preko tabela odlučivanja do Nassi-Schneiderman dijagrama, a najčešće se koristi neka vrsta pseudokoda [4,5,6];



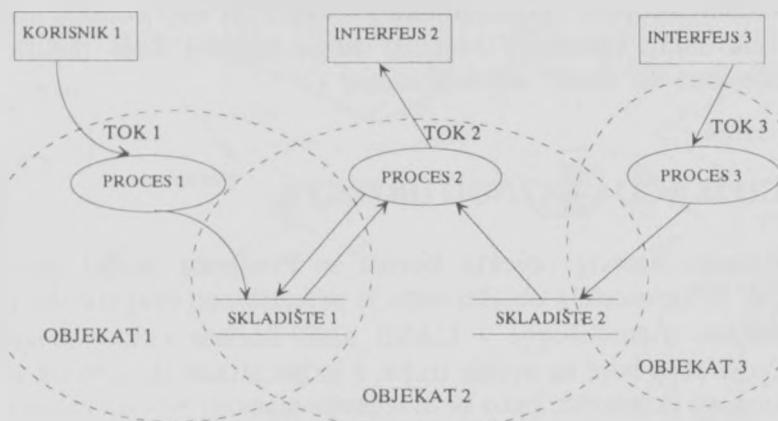
Slika 4.2. Dekompozicija DTP

4.2.2. Objektno orijentisana specifikacija kao rezultat SSA

Da bi funkcionalna analiza sistema rezultovala u objektno-orijentisanu specifikaciju (OOS) neophodno je uvesti sledeća ograničenja i proširenja konvencionalne SSA [6,7]:

- (i) DTP se dekomponuju na procese koji između sebe komuniciraju isključivo preko skladišta podataka (stanja sistema). Dekompozicija se vrši dokle god se neki proces na ovaj način može prirodno dekomponovati na procese koji između sebe komuniciraju preko skladišta.
- (ii) Primitivni procesi takođe komuniciraju isključivo preko skladišta. Umesto da se posebno opisuje struktura tokova i skladišta i logika primitivnih procesa, "pomešaju" se ova dva dela specifikacije i na taj način definišu "složeni objekti" koji predstavljaju primitivne procese sistema. Primer ovakve OO interpretacije DTP-a prikazana je na Slici 4.3a. Crtkastim linijama su zaokruženi mogući objekti. Jedan objekat obuhvata proces, tokove i skladišta koji dati proces generiše odnosno koristi. Sistem je na ovaj način dekomponovan i predstavljen kolekcijom kooperativnih objekata. Kao što se sa slike vidi, ovakav način komunikacije objekata je "loše strukturiran". Veći broj različitih objekata obuhvata ista skladišta, ima ista stanja. Ova "redundansa stanja" značajno otežava problem ažuriranja podataka, operacije ažuriranja nad jednim složenim objektom, zbog loše struktuiranosti sistema mogu se praktično nekontrolisano širiti na druge objekte. Zbog toga se uvodi i koncept "baznog objekta" preko koga se uključuju sve prednosti koncepta baza podataka u specifikaciju IS.
- (iii) U informacionim sistemima zasnovanim na bazama podataka eksplicitno se odvajaju podaci od programa, odnosno definiše se konceptualni model sistema kao model stanja sistema koji ne zavisi od programa koje će se nad njim izvoditi. Osnovna prednost ovakvog pristupa je jedinstveno upravljanje podacima kao zajedničkim resursom u sistemu, uz autonoman razvoj pojedinih aplikacija, mogućnost pristupa bazi podataka preko upitnih jezika i mogućnost razvoja novih, unapred nepredviđenih, aplikacija nad definisanom bazom podataka. Očigledno kontradiktorni principi baza podataka i objektno-orijentisanih pristupa mogu se razrešiti na taj način što će se baza podataka tretirati kao jedinstveni objekat (Slika 4.3b), koji sadrži sva stanja sistema. Imajući u vidu ciljeve baze podataka i probleme u očuvanju integriteta, objekat "baza podataka" treba da bude potpuno učairena za ažuriranje preko definisanja skupa operacija za ažuriranje koje čuvaju integritet baze, a učairena za pretraživanje preko

koncepta pogleda. Ako je baza podataka poseban objekat, nju treba nezavisno specifikovati i projektovati korišćenjem poznatih metoda modeliranja podataka i transformacija specifikacionog u implementacioni model [6]. Usvajajući ovakvu definiciju baze podataka model IS se dekomponuje na jednostavniji model: kolekciju nezavisnih aplikacionih objekata koji komuniciraju sa "baznim objektom". Jedan aplikativni objekat čine sada jedan primitivni proces i njegovi ulazno-izlazni tokovi.



(a) Međusobno povezani aplikativni objekti



(b) Baza podataka kao jedinstveni objekat

Slika 4.3. Objektana specifikacija IS preko Strukturne sistemske analize i modela podataka

Imajući u vidu sve što je prethodno rečeno, objektna specifikacija softvera se sastoji od:

- Objektne specifikacije "baznog objekta" - baza podataka se tretira kao jedinstveni objekat i na taj način specifikuje.
- Specifikacije skupa aplikacionih objekata koje čine primitivni procesi dobijeni SSA, zajedno sa svojim ulazno-izlaznim tokovima. Dekompozicija koja je dovela do primitivnih procesa može se predstaviti skupom "meni objekata", objekata koji nemaju stanja, već ih čine samo operacije koje se mogu birati i koje iniciraju nove aplikacione ili "meni" objekte.

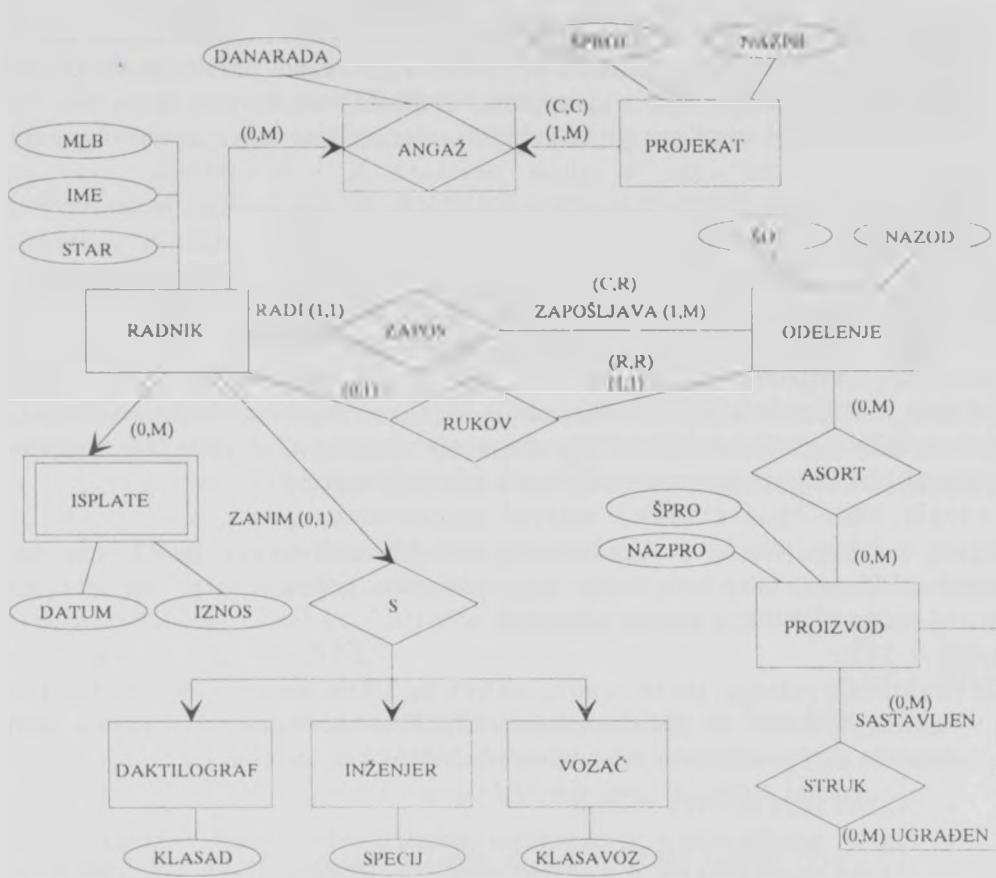
4.3. SPECIFIKACIJA BAZNOG OBJEKTA

Za specifikaciju baznog objekta koristi se Prošireni model objekti-veze (PMOV)[6]. Izbor modela objekti-veze je prvenstveno pragmatske prirode - sve standardne metodologije i CASE alati koriste neku verziju ovoga modela. Prosirenja koja se uvode treba, s jedne strane da strožije formalno definišu njegove koncepte, kako bi se transformacioni pristup mogao što šire primeniti, a sa druge da ga, definišući i operacije modela, učine objektno-orientisanim.

U odnosu na orginalni Chen-ov model [7], pored već standardnih prosirenja sa konceptima molekularne semantike (agregacijom i generalizacijom) [8,9] uvodi se i precizna formalna specifikacija strukturalnih i vrednosnih ograničenja i definisu operacije modela [6,10], tako da kao i svaki drugi potpuni model podataka i PMOV poseduje tri osnovne grupe koncepata: (1) strukturu, (2) ogranicenja i (3) operacije.

Uobičajeno je, zbog jednostavnosti primene, preglednosti, komunikacije sa korisnikom i primene CASE alata, koristiti "grafički jezik", skup grafičkih simbola koji, preko tzv "Dijagrama objekti veze (DOV)", opisuje strukturu sistema.

Sledi definicija koncepata i njihova ilustracija na primeru DOV-a prikazanog na Slici 4.4.



Slika 4.4. Primer specifikacije baznog objekta

4.3.1. Struktura PMOV

Osnovni koncept modela objekti veze je klasa objekata. Koristeći apstrakciju klasifikacije, pojedinačni objekti nekog sistema se klasificuju u tipove objekata. Tip objekta je opšti pretstavnik neke klase, a svaki pojedinačni objekat predstavlja jedno pojavljivanje datog tipa. Klasa objekata je skup pojavljivanja objekata datog tipa. Ako posmatramo neki skup sličnih objekata, tip objekta predstavlja intenziju tog skupa, a klasa objekata istovremeno i intenziju i ekstenziju tog skupa. Na primer, tip objekta je Radnik, pojavljivanja toga tipa su Jovan, Petar, Ana i drugi, a klasa objekata RADNIK predstavlja skup pojavljivanja objekata tipa Radnik. Na DOV klasa objekata se predstavlja pravougaonikom.

Veze u modelu opisuju način povezivanja (uzajamna dejstva) objekata. Apstrakcija klasifikacije se može primeniti i na veze i definisati, analogno pojmovi pojavljivanja, tipa i klase veza. U PMOV se koriste samo binarne veze. Višestruke veze se pretstavljaju apstrakcijom agregacije objekata. Svaki tip binarne veze V klasa objekata X i Y definise dva tipa preslikavanja: (1) X^VY: X --> Y i (2) Y^VX: Y --> X. Kardinalnost ovih preslikavanja se pretstavlja parom (DG,GG), donjom granicom kardinalnosti preslikavanja (DG) koja pretstavlja najmanji broj i gornjom granicom kardinalnosti preslikavanja GG koja pretstavlja najveći broj objekata klase kodomena koja se u datom preslikavanju može pridružiti jednom pojavljivanju objekta i klase domena preslikavanja, respektivno. Na primer, jedan radnik RADI u najmanje jednom i najviše jednom odelenju, a jedno odelenje ZAPOŠLJAVA najmanje jednog i najviše M radnika. (Oznaka M se koristi za neki nepoznat ceo broj veći od 1).

Može se uspostviti više veza između istih klasa objekata, kao i veza nad jednom klasom objekata, veza koja povezuje pojavljivanja objekata iste klase (veza STRUK u datom primeru).

Iz praktičnih razloga, da bi se broj naziva na DOV smanjio, a s obzirom da su koncepti veze i preslikavanja delimično redundantni (jedna veza pretstavlja dva preslikavanja), usvaja se sledeća konvencija:

- Nazivi veza se uvek zadaju;
- Nazivi preslikavanja se obavezno zadaju u rekurzivnim vezama;
- Nazivi preslikavanja u ostalim vezama su opcioni i ako nisu zadati, po "difolto" uzimaju bilo ime veze, bilo ime kodomena preslikavanja.

Objekti u sistemu se opisuju preko svojih svojstava, odnosno atributa. Na primer atributi klase objekata RADNIK su MLB (Matični lični broj), IME i STAR(ost). Svaki atribut, u jednom trenutku vremena ima neku vrednost. Skup mogućih vrednosti nekog atributa se naziva domen atributa. Formalno se atribut definiše kao preslikavanje iz klase objekata u domen. U PMOV se koriste samo jednoznačni atributi. To znači da preslikavanje ATRIBUT --> DOMEN ima uvek DG = 1 i GG = 1. Donja granica ovog preslikavanja ne može imati vrednost nula, jer bi to značilo da je posmatrani atribut "neprimenljivo svojstvo" za sve objekte posmatrane klase, pa objekti na koje je ovo svojstvo primenljivo treba da budu izdvojeni u poseban podtip (podklasu). Inverzno preslikavanje DOMEN --> ATRIBUT ima DG = 0 ili DG = 1, a GG = M. Ako je za ovo preslikavanje GG = 1, posmatrani atribut je identifikator klase objekata.

Zbog svega ovoga i jednostavnosti prikazivanja domeni i kardinalnosti preslikavanja se ne navode na DOV, već se na njemu samo označavaju identifikatori (zvezdicom ili podvlačenjem), a popis atributa i njihovih domena se daje u posebnoj tabeli i tretira kao ograničenje u modelu.

U PMOV se ne koriste višeznačni atributi, već se odgovarajuća svojstva objekata pretstavljaju bilo kao preslikavanja datog objekta prema novodefinitivnom objektu koji pretstavlja domen višeznačnog atributa ili kao "slabi objekti". Prvi način se koristi ako domen višeznačnog svojstva ima unapred zadat, semantički značajan skup vrednosti. Ako to nije slučaj, ako pojedinačne vrednosti takvog atributa nema smisla posebno i unapred identifikovati, već se nova vrednost domena identificuje samo kada se zada višeznačni atribut, takav domen se pretstavlja kao "slab objekat". Očigledno je da pojavljivanja "slabog objekta" i identifikaciono i egizistencijalno zavise od datog pojavljivanja nadređenog objekta (objekta koga slab objekat opisuje). Zbog toga se slab objekat ponekad naziva i opisni objekat, jer je njegova jedina uloga da opiše (definiše višeznačna svojstva) neki drugi objekat. Grafička oznaka i ilustracija definicije slabog objekta je očigledna iz primera objekata ISPLATE.

Pored ovih "atomskih" semantičkih koncepcata koji su izvedeni primenom apstrakcije klasifikacije, u PMOV definišu se i "molekularni" semantički koncepti koji se izvode na osnovu apstrakcija generalizacije i agregacije.

Generalizacija je apstrakcija u kojoj se skup sličnih tipova objekata tretira kao generički tip objekta (nadtip). Specijalizacija je inverzni postupak u kome se za neki tip objekta definišu njegovi podtipovi koji imaju neke specifične atribute, veze i/ili operacije. Da bi se strukturno iskazale vrste specijalizacije (generalizacije) uvodi se posebna klasa objekata, klasa kategorizovanih podtipova (romb sa oznakom "S") koja pretstavlja nepovezanu uniju klasa podtipova, odnosno skup u kome se javlja svaki nadtip onoliko puta koliko je puta specijalizovan u različite podtipove. Specijalizacija i generalizacija tada pretstavljaju kompozicije sledećih preslikavanja:

DG,GG 1,1
Specijalizacija: NADIP -----> S -----> PODTIP

1,1 1,1
Generalizacija: PODTIP -----> S -----> NADTIP

Kardinalnost preslikavanja NADTIP --> S definiše vrstu specijalizacije na sledeći način:

- (1) DG = 0, neobavezna specijalizacija;
- (2) DG = 1, obavezna specijalizacija;
- (3) GG = 1, ekskluzivna specijalizacija;
- (4) GG > 1, neekskluzivna specijalizacija.

Osim preslikavanja NADTIP --> S, sva ostala su trivijalna i ne prikazuju se na modelu. Naziv preslikavanja NADTIP --> S definiše kategoriju (kriterijum) specijalizacije. Na primer ZANIM(anje) je kriterijum po kome se specijalizuju RADNICI, a ova specijalizacija je neobavezna i ekskluzivna.

U apstrakciji generalizacije, važi pravilo nasleđivanja osobina i operacija. Podtip nasleđuje sve atribute, veze i operacije nadtipa.

Agregacija je apstrakcija u kojoj se veza između dva ili više tipova objekata tretira kao tip objekta na višem nivou apstrakcije, agregirani objekat. Objekti koji čine agregaciju nazivaju se komponentama agregacije. Kardinalnost preslikavanja KOMPONENTA --> AGREGACIJA se zadaje, dok je inverzno preslikavanje trivijalno, uvek je (1,1). Agregirani objekat se razlikuje od ostalih i po tome što nema svoj sopstveni identifikator, već ga identificuju objekti koje on agregira. Grafička pretstava agregacije je uokvireni romb, čime se asocira na njegov drugi naziv "mešoviti tip objekat-veza", jer je on istovremeno i objekat kao celina i veza njegovih komponeneti.

4.3.2. Operacije u PMOV

Pogodno je operacije u PMOV podeliti na operacije održavanja (ažuriranja) baze podataka i operacije pretraživanja (izveštavanja).

Operacije održavanja. Iz same strukture PMOV očigledno je da se operacije održavanja baze podataka svode na operacije dodavanja novog objekta u klasu, izbacivanja objekta iz klase, izmenu vrednosti nekog atributa, zatim povezivanja i razvezivanja dva objekta preko zadatog preslikavanja, odnosno veze. Sintaksa i detaljnije objašnjenje ovih operacija slede.

(1) UBACIVANJE

insert naziv_objekta(lista parova atribut-vrednost, lista parova id_susednog_objekta-vrednost)

(I ovde, kao i u drugim iskazima, podrazumeva se da su elementi liste razdvojeni sa „, a elementi para sa „“.)

Na primer

insert RADNIK (MLB:307,IME:Ana, STAR:25, ŠO:03)

Lista parova id_susednog_objekta-vrednost daje identifikatore svih susednih objekata prema kojima dati objekat ima "obavezno" preslikavanje (preslikavanje sa donjom granicom većom od nule).

Podrazumeva se da operacija insert prouzrokuje operacije connect prema svim okolnim objektima prema kojima posmatrani objekat ima obavezno preslikavanje.

(2) IZBACIVANJE

delete naziv_objekta (naziv_id_objekta:vrednost)

Na primer

delete RADNIK (MLB:307)

Podrazumeva se da operacija delete prouzrokuje operacije disconnect za sva preslikavanja u kojima učestvuje posmatrani objekat.

(3) AŽURIRANJE

update naziv_objekta (naziv_id_objekta:vrednost, liste parova atribut-vrednost)

Na primer

update RADNIK(MLB:307,IME:Mira)

(4) POVEZIVANJE

connect naziv_preslikavanja (id_domena:vrednost, id_kodomena:vrednost)

Operacija connect se ne može direktno primeniti na preslikavanja čije su kardinalnosti DG = 1 i GG = 1, jer se to povezivanje ostvaruje automatski sa ubacivanjem objekta domena preslikavanja.

Na primer

connect RUKOV (MLB:307,ŠO:03)

(5) RAZVEZIVANJE

disconnect naziv_preslikavanja (id_domena:vrednost, id_kodomena:vrednost)

Operacija disconnect se ne može direktno primeniti na preslikavanja čije su kardinalnosti DG = 1 i GG = 1 inače bi uvek narušila uslov integriteta. Na primer,

disconnect ASORT(ŠO:03,ŠPRO:A325)

(6) PREVEZIVANJE

reconnect naziv_preslikavanja (id_domena:vrednost,
id_kodomena_sa:vrednost,
id_kodomena_na:vrednost)

Ova operacija se primenjuje na ona preslikavanje na koja se ne mogu primeniti operacije disconnect i connect, a pretstavlja uzastopnu primenu ovih operacija u jednoj transakciji.

Operacije pretraživanja. U PMOV je moguće definisati dve vrste operacija za pretraživanje. Prvi skup operacija je "navigacione" prirode, slično operacijama u mrežnom i hijerarhijskom modelu, i one omogućuju kretanje kroz model i pristup pojedinačnim pojavljivanjima objekata "tekuće" klase, klase kojoj se u tom trenutku pristupilo. Drugu grupu operacija čine tzv "specifikacione" operacije, odnosno odgovarajući upitni jezik, kojima se iskazuje struktura i uslovi koje rezultat operacije treba da zadovolji.

Da bi se definisale operacije nad klasama objekata i omogucilo jednostavno povezivanje klasa objekata preko preslikavanja definisu se promenljive u jeziku na sledeći način:

naziv_promenljive naziv_klase_objekata, što znači da promenljiva uzima vrednosti iz klase (skupa) objekata. Na primer x:RADNIK, promenljiva x uzima za svoje vrednosti pojavljivanja klase objekata RADNIK

ili

naziv_promenljive_1 ":" naziv_promenljive_2 naziv_preslikavanja (ili naziv veze),

što znači da prva promenljiva uzima vrednosti iz kodomena navedenog preslikavanja (veze) za vrednost domena preslikavanja datog vrednošću

druge promenljive. Na primer, ako se definiše:

x: ODELENJE;

y: x.ZAPOŠLJAVA,

y uzima vrednosti iz podskupa klase RADNIK koji čine pojavljivanja koja su preko preslikavanja ZAPOŠLJAVA vezani za dato odelenje preko tekuće vrednosti promenljive x.

Kako je objekat agregacija njegovih atributa, promenljivoj x se takođe mogu pridodati atributi preko iskaza:

naziv_promenljive naziv atributa

Na primer x.NAZOD ili y.IME ili y.MLB i slično.

Može se, što ćemo ovde i učiniti, umesto eksplisitne definicije promenljive usvojiti konvenciju da su nazivi promenljivih jednaki nazivima klasa iz koje uzimaju vrednosti.

Ako se i u navigacionom jeziku usvoji ovakva definicija promenljivih, kretanje ("navigacija") kroz strukturu se ostvaruje samom definicijom promenljivih, tako da je osnovna operacija pretraživanja izvlačenje jednog pojavljivanja objekta iz zadate klase. Ako je klasa prazna, odgovarajuća operacija vraća kod "EOK".

Definišu se sledeće navigacione operacije pretraživanja u modelu objekti veze:

(1) GET ANY naziv_promenljive {daje bilo koje pojavljivanje klase nad kojom je promenljiva definisana ili EOK ako je klasa prazna }. Na primer:

x:ODELENJE, y:x.ZAPOŠLJAVA

GET ANY x - daje bilo koje pojavljivanje objekta odelenje,

GET ANY y - daje bilo koje pojavljivanje objekta radnik iz podskupa radnika vezanih za tekuće pojavljivanje objekta odelenje.

(2) GET ANY naziv_promenljive WHERE uslov { Uslov se može iskazati preko relacionih operatora (=, <, > i slično) koji se mogu definisati nad domenima atributa klase nad kojom je definisana promenljiva i logičkih

operatora AND OR i NOT. Rezultat operacije je jedno, bilo koje, pojavljivanje objekta koje zadovoljava uslov}. Na primer, sekvenca naredbi

GET ANY x WHERE x.ŠIFOD = 011;

GET ANY y WHERE y.ZANIMANJE = 'lekar' AND y.STAROST > 40;

daje odelenje sa šifrom 011 i prvog lekara starijeg od 40 godina iz tog odelenja ili EOK ako takvih radnika nema.

Operacija GET ANY daje samo jedno pojavljivanje date klase, odnosno jedno pojavljivanje koje zadovoljava dati uslov. Da bi se mogli izvuci i druga (ili sva) pojavljivanja uvode se operacije

(3) GET DUPLICATE naziv_promenljive; i

(4) GET DUPLICATE naziv_promenljive WHERE uslov;

koje daju jedno od preostalih pojavljivanja date klase, posle primene operacija GET ANY ili GET DUPLICATE.

Ovakav navigacioni jezik neznatno se razlikuje od specifikacionog (upitnog) jer oba na isti način definišu promenljive. U specifikacionom jeziku, umesto operacije koja "vadi" jedno pojavljivanje objekta iz klase, moguce je zadati strukturu rezultata i selektovati sve komponente koje zadovoljavaju zadate uslove jednim upitom.

Upitni jezik nad modelom objekti-veze veoma je sličan upitnim jezicima za nenormalizovani relacioni model (NFNF model) [11] ili upitnim objektnim jezicima [12]. Sintaksa upitnog jezika prati sintaksu standardnog SQL-a [13]. Pored već pokazanog načina definisanja promenljivih, upitni jezik za PMOV se razlikuje od SQL-a i po tome što se uz ključnu reč SELECT ne navode samo atributi, već i strukture koje se mogu dobiti krećući se od posmatranog objekta preko njegovih preslikavanja. Deo strukture može da bude i novi upit. Ovde se daju samo osnove sintakse ovog jezika, a detaljni opis se može naći u [6].

upitni blok ::

 SELECT struktura [INTO naziv komponente aplikacionog objekta]
 FROM promenljiva
 WHERE uslov;

struktura ::= komp_upit | komp_upit, struktura

komp_upit ::= atribut objekta | upitni blok

uslov ::= predikat nad promenljivom nad kojom je definisan upitni blok.

Svaki upitni blok se definiše nad jednom promenljivom, a promenljive se definišu na napred opisan način, uz uvedenu konvenciju.

Struktura komponente aplikacionog objekta kojoj se iskazom INTO dodeljuje rezultat upita mora biti u saglasnost sa strukturom upita.

U svemu ostalom podrazumevaju se delovi sistakse analogne konvencionalom SQL-u.

Daje se samo jedan primer upita, čiji je verbalni iskaz: Izlistati šifre i nazive odelenja i za svako odelenje matične brojeve imena i starosti radnika starijih od 40 godina, kao i srednju vrednost njihovih isplata.

```
SELECT ŠO,NAZOD (SELECT MLB, IME STAR,  
                  SELECT AVG(IZNOS)  
                  FROM ODELENJE.ZAPOŠLJAVA.ISPPLATE  
                  FROM ODELENJE.ZAPOŠLJAVA  
                  WHERE ODELENJE.ZAPOŠLJAVA.STAROST >40  
FROM ODELENJE;
```

4.3.3. Ogranicenja i pravila integriteta

Pri uvođenju i definisanju pojedinih koncepata strukture PMOV, definisana su i neka ograničenja, od kojih je najznačajnije kardinalnost preslikavanja u vezi dva objekta. Ova ograničenja koja se definišu na Dijagramu objekti veze nazivamo **strukturnim ograničenjima**. Međutim postoje i mnogo složenija semantička ograničenja koje je nemoguće ili nepraktično prikazati strukturom PMOV, pa je neophodno definisati jezik za njihovu specifikaciju. Ova ograničenja najčešće se definišu nad vrednostima atributa, pa se zato nazivanju vrednosna ogranicenja u PMOV.

Za iskazivanje vrednosnih ogranicenja koristi se "objektni račun", analogan relacionom računu [13]. Promenljive u ovom računu uzimaju vrednosti iz klase objekata, a definišu se na ranije pokazani način. Atomske formule, odnosno "atomi" u objektnom računu su:

$x.A \Theta y.B$ gde su x i y promenljive, A i B su atributi klase iz kojih x i y uzimaju vrednosti, a Θ je operator poređenja (na primer $<$, $>$, $=$, $IZMEĐU...$) definisan nad domenom atributa A i B (A i B moraju biti definisani nad istim domenom).

$x.A \Theta c$ gde su x , A i Θ kao i u prethodnom stavu, a c je konstanta koja ima isti domen kao i A .

Dobro definisane formule (ddf) u objektnom računu se formiraju preko sledeće sintakse (produkcionih pravila):

```
ddf:: atom|
(ddf)|
NOT ddf|
ddf AND ddf|
ddf OR ddf|
EXISTS naziv_promenljive (ddf)|
FOREACH naziv-promenljive (ddf)|
IF ddf THEN ddf|
```

EXISTS (\exists) je egzistencijalni kvantifikator, a **FOREACH** (\forall) je univerzalni kvantifikator.

Navedimo nekoliko primera ograničenja:

(i) Prosta ograničenja na atribute pojedinih objekata:

```
OGR1 := RADNIK.STAR < 66;
OGR2 := PROIZVOD.NAZPRO NOTNULL;
```

(ii) Složena vrednosna ograničenja, ograničenja koja vezuju atribute više različitih objekata:

```
OGR3 := FOREACH RADNIK (IF RADNIK.ZAPOS.NAZOD
= Računovod' THEN SUM(RADNIK.ANGAŽ.DANARAD) <30);
```

(Radnik koji radi u računovodstvu, ne može biti angažovan na projektima više od ukupno 30 dana).

Za potpunu specifikaciju baznog objekta pored ograničenja, od interesa je i specifikacija (dinamičkih) pravila integriteta. Jedno dinamičko pravilo integriteta čini trojka

<OPERACIJA, OGRANIČENJE, AKCIJA>

preko koje se, za svaku OPERACIJU održavanja BP i svako OGRANICENJE koje ona može da naruši, definiše AKCIJA koju treba preduzeti ako je ograničenje narušeno. Akcija može biti neka operacija nad baznim objektom ili poziv operacije (poruka) nad nekim aplikacionim objektom. Najpogodnije je gornje trojke prikazati preko odgovarajuće (nenormalizovane) tabele. Za prethodna ograničenja imamo:

OGRANIČENJE	OPERACIJA	AKCIJA
OGR1	INSERT RADNIK UPDATE RADNIK.STAROST	Odbij operaciju uz poruku aplikacionom objektu "Nekorektna starost"
OGR2	INSERT PROIZVOD UPDATE PROIZVOD.NAZPRO	Odbij operaciju uz poruku aplikacionom objektu "Nedostaje naziv"
OGR3	INSERT ANGAŽ RECONNECT RADNIK UPDATE ODELENJE.NAZOD	Odbij operaciju uz poruku aplikacionim objektima "Preveliko angažovanje na projektima"

Strukturalna dinamička pravila integriteta. Strukturalna ograničenja u PMOV definisana su samom strukturom modela, odnosno načinom međusobnog povezivanja različitih klasa objekata. Operacije održavanja baze podataka mogu da naruše strukturalna ograničenja narušavajući kardinalnost preslikavanja preko kojih su ostvarene veze između klasa objekata. Pri definiciji semantike pojedinih operacija održavanja baze podataka predpostavljeno je sledeće:

- Operacija insert poziva automatsko izvršavanje operacije connect za sva obavezna preslikavanja (preslikavanja sa $DG > 0$) koja poseduje posmatrana klasa;
- Operacija delete poziva automatsko izvršavanje operacije disconnect za sva preslikavanja date klase;
- Operacija reconnect se svodi na uzastopnu primenu operacija disconnect i connect, u jednoj atomskoj transakciji;
- Operacije connect i disconnect na jednom preslikavanju podrazumevaju da se odgovarajuća operacija obavi i na njemu inverznom preslikavanju;
- Pretpostavlja se da se operacija update neće primenjivati na identifikator objekta.

Imajući ovo u vidu, očigledno je da struktura ograničenja mogu narušiti samo operacije connect i disconnect, bilo da se neposredno primenjuju, bilo da su automatski indukovane od drugih operacija, pa samo za ove operacije treba definisati dinamička struktura pravila integriteta.

Analizirajmo način na koji ove operacije mogu da naruse strukturno pravilo integriteta i akcije koje se u tim sličajevima mogu preuzeti.

Operacija

connect naziv_preslikavanja(id_domena:vrednost,id_kodomena:
vrednost)

na primer

connect RUKOVODI (MLB:307, ŠIFOD:03)

narušava integritet ako u klasi objekata koja pretstavlja kodomen preslikavanja ne postoji navedeno pojavljivanje. Pretpostavlja se da pojavljivanje klase domena mora uvek da postoji da bi se ova operacija uopšte izvršila.

Akcije koje se u ovom slučaju mogu da preduzmu su:

- (i) Operacija se odbija jer ne postoji objekat kodomen. Ovu opciju ćemo nazvati RESTRICTED (R);
- (ii) Umesto sa navedenim objektom kodomena, ako to pojavljivanje ne postoji, povezivanje se izvršava sa specijalnim pojavljivanjem kodomena, tzv. "nula objektom", čije je značenje "nepoznati objekat". Pretpostavlja se da je u ovom slučaju definisano pojavljivanje "nula objekta" u klasi kodomena. Ovu opciju nazvaćemo NULLIFIES (N).
- (iii) Umesto sa navedenim objektom kodomena, ako to pojavljivanje ne postoji, povezivanje se izvršava sa "difolt" objektom kodomena. Pretpostavlja se da je difolt objekat kodomena definisan. Ovu opciju zadovoljavanja integriteta nazvaćemo DEFAULT (D).
- (iv) Ako objekat kodomene sa kojim dati objekat domena treba da se poveže ne postoji moguće ga je prvo kreirati operacijom insert, pa zatim izvršiti odgovarajuće povezivanje. Ovu opciju zadovoljavanja integriteta nazvaćemo CASCADES (C). Očigledno je da ova opcija može da okine (trigger) čitavu kaskadu operacija u bazi podataka. Ubacivanje nepostojećeg pojavljivanja u klasu kodomena povlači automatski operacije connect za sva

njegova obavezna preslikavanja, a ove preko svojih mogucih opcija zadovoljenja integiteta CASCADES ubacivanje objekata u odgovarajuce klase kodomena i tako dalje.

Iz prethodne diskusije je očigledno da se dinamička struktura pravila integriteta zadaju za svako preslikavanje na modelu i to posebno za operaciju connect, a posebno za operaciju disconnect. Ako se iskoriste skraćenice navedene u zagradama i ako se pretpostavi da prvo slovo u paru znači opciju za operaciju connect, a drugo za disconnect, tada se opcije zadovoljavanja dinamičkih pravila strukturnog integriteta mogu zadati na Dijagramima objekti veze, kako je to na Slici 4.4 i prikazano. Pri tome se "po difoltu" uvodi da je opcija R uvedena za sve operacije connect, a opcija C za sve operacije disconnect, pa se na dijagramu navode samo druge izabrane opcije.

4.3.4. Potpuna specifikacija baznog objekta

Potpuna specifikacija baznog objekta obuhvata:

- (i) Dijagram modela objekti-veze koji opisuje strukturu baze (baznog objekta);
- (ii) Skup operacija ažuriranja, zajedno sa vrednosnim i stukturnim pravilima integriteta koje definisu semantiku svake operacije.

4.4. SPECIFIKACIJA APLIKACIONIH OBJEKATA

Kao što je ranije rečeno, jedan aplikacioni objekat čine jedan proces dobijen Strukturnom sistemskom analizom, zajedno sa svim ulazno izlaznim tokovima (Slika 4.3b). Proces se realizuje preko skupa operacija za formiranje komponenti objekta, za ažuriranje baze podataka i operacija za komunikaciju sa interfejsom (objektom van sistema). Imajući sve ovo u vidu, aplikacioni objekat se opisuje preko svoje strukture, skupa operacija koje se nad (sa) njim izvode i svoje dimanike koja definiše uslove za "okidanje" pojedinih operacija.

Struktura aplikacionog objekta integrise strukturu svih tokova posmatranog procesa. Drugim rečima, iz strukture aplikacionog objekta treba da bude moguće izvući strukturu bilo kog toka.

Komponenta strukture aplikacionog objekta može da bude neki drugi objekat, tzv "ugrađeni" objekat. Ugrađeni objekat nije aplikacioni u posmatranom sistemu, jer aplikacioni objekti u jednom sistemu, po definiciji, komuniciraju između sebe samo preko baznog objekta. Ugrađeni objekti su prvenstveno posledica projektovanja (jednostavnijeg opisivanja) strukture aplikacionog objekta ili način realizacije nekih operacija, a preko njih se može ostvariti i efikasno korišćenje ranije razvijenog softvera. Način opisivanja ugrađenih (Embedded) objekata i aplikacionih objekata je isti, jedino se uz njihov naziv preko odgovarajućih ključnih reči, iskazuje njihova vrsta, što ima samo metodološki značaj.

Sintaksa za opis objekata se značajno pojednostavljuje ako se odvoji opis elementarnih struktura (polja - odeljak FIELDS) od opisa složenih (odeljak STRUCTURE) i ako se navođenje skupa operacija (odeljak OPERATIONS) odvoji od njihovih "tela" (odeljak FUNCTION LIBRARY) Sledi sintaksa za opis objekta. Ključne reči su ispisane velikim slovima ili su navedene između znakova navoda ("ključna reč"). Komentari sintakse se navode između zvezdica.

objekat :: APOB|EMOB naziv objekta opis objekta

opis objekta :: STRUCTURE opis_strukture

OPERATIONS lista_naziva_operacija

DYNAMICS opis_dinamike

FIELDS opis_polja

FUNCTION LIBRARY opis_operacija

opis_strukture :: "<"lista-komponenti ">" * agregacija komponenti*

| "[" lista-komponenti "]" * unija komponenti*

| "{" komponenta "}" * skup komponenti*

lista-komponenti :: komponenta | komponenta lista-komponenti

komponenta :: naziv_polja * naziv polja koje je posebno opisano *

| EMOB naziv objekta * Naziv ugrađenog objekta koji je
posebno opisan*

| opis_strukture * komponenta jedne strukture može da
bude druga struktura*

lista_naziva_operacija :: naziv_operacije | naziv_operacije lista_naziva_operacija

opis_dinamike :: naziv_operacije tip_operacije uslov napomena

tip_operacije ::FUN | MESS

* operacija može da bude sopstvena, odnosno funkcija (procedura) koja na bazi ulaznih parametara daje rezultat i menja stanje objekta ili poruka (MESS) nekom drugom objektu, moguće i ugrađenom u dati objekat da uradi svoju operaciju*

uslov :: AU naziv_događaja | REQ

* navodi se događaj koji operaciju automatski (AU) okida, događaj može da bude i druga operacija, ili se operacija izvršava na eksplicitan zahtev korisnika(REQ)*

napomena :: proizvoljan tekst, koji može da bude i telo funkcije, za jednostavnije funkcije*

* Odeljak DYNAMICS je pogodno prikazati i kao tabelu sa kolonama Naziv_operacije, Tip_operacije, Uslov, Napomena *

opis_polja ::naziv_polja tip_polja [ograničenje]

tip_polja :: predefinisani | semantički

* Pod predefinisanim tipom se podrazumevaju tipovi podataka koji su ugrađeni u većinu programskih jezika. Pod semantičkim se podrazumevaju semantički domeni, odnosno apstraktni tipovi podataka koji su posebni objekti*

predefinisani :: INTEGER "(" dužina ")"
| CHARACTER "(" dužina ")"
| REAL "("ukupna_dužina "," dužina_posle_zapete ")"
| LOGICAL
| DATE

semantički :: EMOB naziv_objekta

ograničenje :: * bilo koji predikat koji se može definisati nad datim tipom polja *

* I odeljak FIELDS je pogodno prikazati kao tabelu sa kolonama Naziv_polja, Tip_polja, Ograničenje*

opis_operacija * biblioteka tela operacija koje su tipa FUN, data bilo kao kod ili pseudokod*

Svaki aplikacioni i ugrađeni objekat je podtip "opšteg objekta", pa samim tim nasleđuje i sve "opštne" operacije, kao što su:

KREIRAJ_OBJEKAT, DAJ_PRVO_POJAVLJIVANJE,
DAJ_SLEDEĆE_POJAVLJIVANJE, EDITOVANJE komponenti i slične.

"Objektni analizator". Kao opštne operacije mogu se smatrati i operacije održavanja baze podataka preko aplikacionog objekta:

UBACI objekat

* Celokupan sadržaj aplikacionog objekta ili nekog njegovog dela (tok od procesa prema bazi) se ubacuje u bazu. Pri tome se pozivaju operacije insert nad baznim objektom i obavljaju po definisanim pravilima integriteta.*

IZBACI objekat

* Izbacuje se sadržaj aplikacionog objekta ili nekog njegovog dela iz baze objekata preko odgovarajućih baznih delete operacija uz postovanje navedenih pravila integriteta.*

PROMENI objekat

* Sadržaj aplikacionog objekta menja odgovarajući sadržaj baze podataka. U aplikacionom objektu postoje "više značni atributi", odnosno skupovne komponente, označene sa konstrukturom {}. Ažurirati skupovnu komponentu znači bilo dodati novi element u skup, izbaciti postojeći ili promeniti neki atribut postojećeg elementa. Neophodno je naznačiti, uz svaki elemenat skupovne komponente, koja je od ovih operacija, u postupku "editovanja" aplikacionog objekta, izvršena. Podrazumeva se postojanje "sakrivenog" polja, uz svaki element svake skupovne komponente, u kome se čuva kod izvršene operacije.*

Implementacioni modul koji transformiše ove operacije aplikacionog objekta u operacije nad baznim objektom nazivamo "objektni analizator". Algoritam objektnog analizatora dobija se analizom strukture aplikacionog objekta po pravilima projektovanja programa na osnovu strukture podataka [14,15]:

- Agregacija komponenti se transformiše u sekvencu odgovarajućih operacija nad baznim objektom;
- Unija komponenti se transformiše u "CASE" razgranatu strukturu odgovarajućih operacija nad baznim objektom;
- Skup komponenti se transformiše u iteraciju odgovarajućih komponeneti nad baznim objektom.

Imajući u vidu ovakav način izvođenja operacija održavanja baze podataka, preko "objektnog analizatora", kodove za operacije UBACI, IZBACI i PROMENI nije neophodno navoditi u odeljku FUNCTION LIBRARY. Dovoljno je samo uz definiciju aplikacionog objekta navesti koja se operacije održavanja baze podataka sa njim izvode.

Primer. Pretpostavimo da je jedan od procesa u sistemu čiji je bazni objekat pretstavljen modelom na Slici 4.4 i proces Održavanje podataka o radnicima odelenja, kako je to na Slici 4.5 i prikazano. Aplikacioni objekat za ovu aplikaciju se definiše kao:



Slika 4.5. Primer za specifikaciju aplikacionog objekta

APOB Radnici_odelenja

STRUCTURE

```
<Šif_odel, Naziv_odel,  
 {<Matbr, Ime, Starost, [Klasadakt, Specijalnost, Vozkласа],  
 EMOB Angažovanje}>>
```

OPERATIONS

Nađi_odelenje
Prikaži_radnike
Prikaži_angajovanje_radnika
PROMENI Radnici_odelenje

Ne dozvoljava se promena Šif_odel i Naziv_odel. Sva ostala korisnička editovanja su dozvoljena.

DYNAMICS

Naziv	Tip	Uslov	Napomena
Nađi_odelenje	MESS	AU-po unosu Šif_odel	Poziva operaciju Izabrano_odeljenje EMOB Odelenje
Prikaži_radnike	FUN	REQ	-
Prikaži_angajovanje _radnika	MESS	REQ	EMOB Angajovanje

FIELDS

Naziv	Tip	Ograničenje
Šif_odel	CHAR(7)	-
Naziv_odel	CHAR(15)	-
Matbr	CHAR(13)	-
Ime	CHAR(15)	-
Starost	INT (2)	< 65
Klasadakt	CHAR(1)	IN (A,B,C)
Specijalnost	CHAR(3)	-
Vozkласа	CHAR(1)	IN (A,B,C,D,E,F)

FUNCTION LIBRARY

Prikaži_radnike
SELECT MLB, IME, STAR INTO <Matbr, Ime, Starost>
 (SELECT KLASAD INTO Klasadakt
 FROM RADNIK.ZANIM)
 UNION (SELECT SPECIJ INTO Specijalnost
 FROM RADNIK.ZANIM)
 UNION (SELECT KLASAVOZ INTO Vozklasa
 FROM RADNIK.ZANIM)

FROM RADNIK

WHERE RADNIK.RADI.ŠO = Radnici_odelenja.Šif_odel;

EMOB Odelenje

STRUCTURE

{<Šifra_odelenja, Naziv_odelenja>}

OPERATIONS

Prikaži_odelenja

Izabrano_odelenje

Nisu dozvoljene korisničke operacije editovanja.

DYNAMICS

Naziv	Tip	Uslov	Napomena
Prikaži_odelenje	FUN	AU- Po pozivu	-
Izabrano_odelenje	FUN	REQ	-

FIELDS

Naziv	Tip	Ograničenje
Šifra_odelenja	CHAR(7)	-
Naziv_odelenja	CHAR(15)	-

FUNCTION LIBRARY

Prikaži_odelenje

SELECT ŠO, NAZOD INTO <šifra_odelenja, Naziv_odelenja>

FROM ODELENJE

WHERE ŠO = Radnici_odelenja.Šif_odel;

Izabrano_odelenje;

Izaberu jedno od prikazanih odelenja;

prenesi ŠO u Radnici_odelenja.Šif_odel i

NAZOD U Radnici_odelenja.Naziv.odel.

EMOB Angažovanje

STRUCTURE

<Šifra_radnika,
{<Šifra_projekta, Naziv_projekta, Dana>}

OPERATIONS

Prikaži_projekte

Dozvoljene su sve operacije editovanja za navedenu komponentu tipa skupa

DYNAMICS

Naziv	Tip	Uslov	Napomena
Prikaži_Projekte	FUN	AU - pri otvaranju ovog objekta	-

FIELDS

Naziv	Tip	Ograničenje
Šifra_radnika	CHAR(13)	-
Šifra_projekta	CHAR(3)	-
Naziv_projekta	CHAR(25)	-
Dana	INT(3)	-

FUNCTION LIBRARY

Prikaži_Projekte

```
SELECT RADNIK.MLB INTO Šifra_radnika,
      (SELECT DANARADA INTO Dana,
       (SELECT ŠPROJ, NAZPR INTO
            <Šifra_projekta, Naziv_projekta>
            FROM RADNIK.ANGAŽ.PROJEKAT)
       FROM RADNIK.ANGAŽ)
      FROM RADNIK
      WHERE RADNIK.MLB = Izabrani radnik;
```

Izabrani radnik je onaj radnik na koga je korisnik u objektu Radnici_odeljenja, na primer, postavio kurzor.

Korisnik u objektu Angažovanje edituje angažovanje radnika na projektima, a promene se prenose u bazu preko operacije PROMENI objekta Radnici_odeljenja čiji je objekat Angažovanje komponenta.

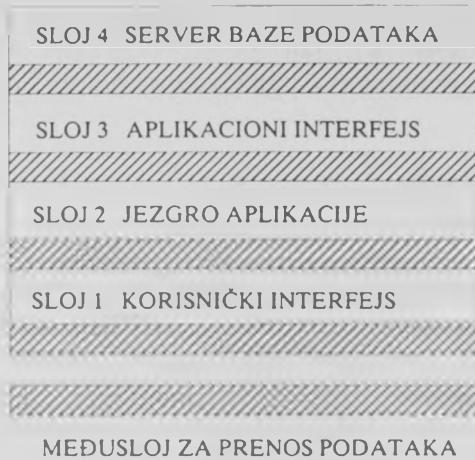
4.5. PROJEKTOVANJE

U prethodnim delovima detaljno je prikazana specifikacija IS koju čine specifikacija baznog objekta i specifikacije aplikacionih objekata. Ova specifikacija je "potpuno logička", nezavisna od implementacione tehnologije. U sledećem koraku metodologije neophodno je detaljnije projektovati sistem, dekomponovati ga u manje celine, koje će biti pogodne sa implementacione tačke gledišta. Pri tome se posebno mora voditi računa o osnovnom zahtevu koji je pred Sistem za pretraživanje informacija postavljen - otvorenost, heterogenost i distribuiranost. Imajući ovo u vidu, ovde se daje jedan generički implementacioni model, izведен na osnovu predloga datih u [16,17,18] i opšte strukture otvorenih sistema prikazanih u Poglavlju 2. Ovaj model je osnova za dalje projektovanje sistema. Ovde se definišu koraci i opšti postupci projektovanja a od sledećeg poglavlja oni se primenjuju na Sistem za pretraživanje informacija.

4.5.1. Generički implementacioni model

Generički implementacioni model pretstavlja dalju funkcionalnu dekompoziciju logičke specifikacije i objektnu implementaciju pojedinih tako dobijenih delova. Dekompozicija se vrši sa tačke gledišta implementacione tehnologije, a ne karakteristika samog sistema koji se razvija: objekti iz specifikacije se nadalje dekomponuju na implementacione objekte određenih vrsta. Ova dekompozicija treba da omogući prostornu i tehnološku transparentnost (transparentnost u odnosu na lokalni hardver i softver i komunikacione protokole), pogodnu komunikaciju komponenti, poštovanje standarda, poštovanje ograničenja koju nameće postojeća tehnologija, uz ostvarivanje zahtevanih performansi.

Generički implementacioni model prikazan je na Slici 4.6. Definišu se četiri osnovna sloja, sa međuslojem za prenos podataka između svaka dva osnovna sloja. Posebno treba naglasiti da je cilj definisanje sva četiri osnovna sloja i međuslojeva ostvarivanje što je moguće veće autonomije osnovnih slojeva. Funkcije jednog sloja se mogu realizovati bez poznavanja načina realizacije funkcija susednih slojeva, što znači da se realizacija susednih slojeva može promeniti, a da to nema nikakav ili veoma mali uticaj na posmatrani sloj.



Slika 4.6. Generički implementacioni model

Sloj 1: Korisnički interfejs. Korisnički interfejs je "front-end" sloj modela preko koga se komunicira sa korisnikom, bio on čovek ili proces (na primer scener bar kod ili "point-of sale" terminal). Primarna funkcija ovog sloja je da vrši prihvatanje podataka i komandi i prikazivanje rezultata. U objektno orijentisanim interaktivnim sistemima ovaj se sloj realizuje preko "interfejs objekata" objekata preko kojih se realizuje savremeni "Graphical User Interface (GUI)". U ovom sloju se vrši i sintaktička validacija ulaznih podataka, formatiranje poruka i realizuje konteksno osetljivi "Help".

Sloj 2: Jezgro aplikacije. U ovom sloju se realizuje suština aplikacije, odnosno obuvata najveći deo aplikacionog objekta sistema. U njemu se definiše struktura aplikacionih i potrebnih "ugrađenih" objekata, implementiraju sve operacije aplikacionih objekata, uz prepostavku da je komunikacija sa baznim objektom realizovana posredstvom Sloja 3.

Sloj 3: Aplikacioni interfejs. Sloj 3 pretstavlja interfejs Jezgra aplikacije sa Serverom baze podataka sa osnovnim ciljem da sakrije specifične karakteristike servera baze podataka of Jezgra aplikacije. U njemu se, po pravilu, mogu da realizuju sve operacije nad bazom podataka, a prvenstveno "objektni analizator", način na koji se operacije održavanja baze podataka, definisane u Jezgru aplikacije transformišu u operacije nad bazom podataka koje obavlja Server baze podataka. Realizacija ovog sloja zahteva detaljno poznavanje Sloja 4, a njegova složenost očigledno zavisi od karakteristika Servera baze podataka (vrste sistema za upravljanje bazom podataka, distribucije baze, načina zadovoljavanja dinamičkih pravila integriteta i slično).

Sloj 4: Server baze podataka. Ovaj sloj pretstavlja neki konkretan server baze podataka, odnosno softver za upravljanje bazom podataka. Ovo je jedini sloj za koga su definisani javni standardi, pa je način komunikacije sa njim jednostavnije standardizovati.

Međuslojevi za prenos podataka povezuju svaka dva osnovna sloja. Međusloj može da sadrži značajne implementacione funkcije (procese) u sistemu. On treba da omogući nezavisnost aplikacije od prostorne distribucije i komunikacionih protokola. Drugi primer funkcije sloja za prenos podataka je konverzija podataka, ako su dva susedna sloja implementirana na različitim tehnologijama, konverzija teksta sa ASCII na EBCDIC kod, sa jednog formata brojeva u tekućem zarezu na drugi i slično.

4.5.2. Dekompozicija "atomske logičke procese"

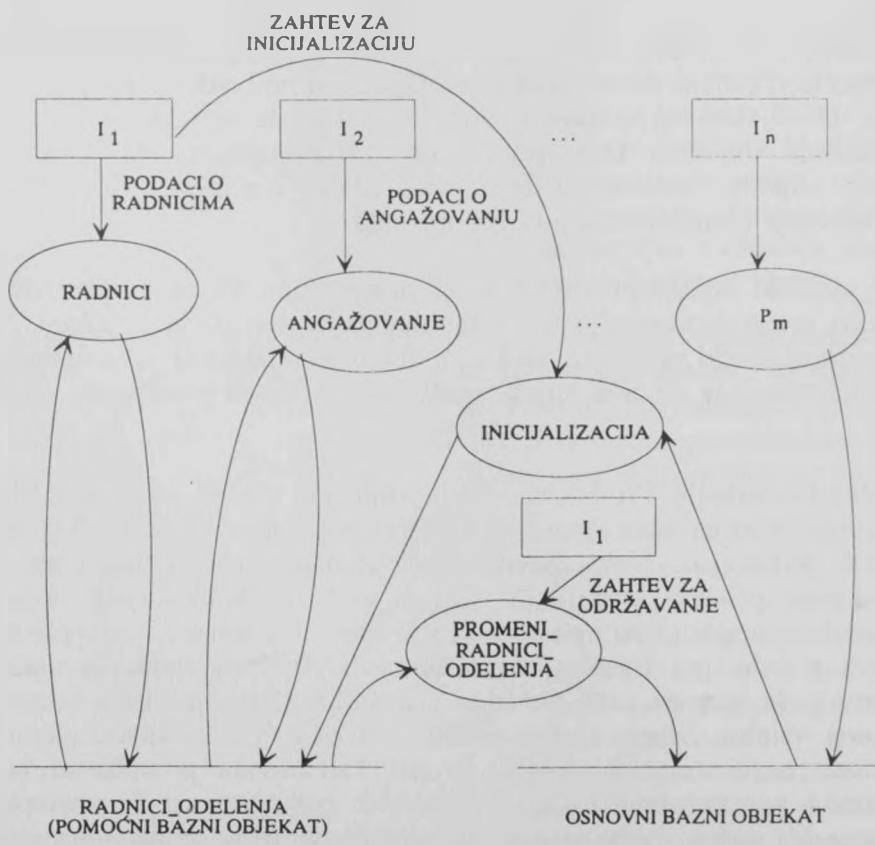
Specifikacija IS je rezultovala u skup "atomske logičke procese (transakcije)" koji komuniciraju između sebe samo preko baze podataka, odnosno koji svojim završetkom ostavljaju bazu podataka u konsistentnom stanju. U objektno-orientisanoj terminologiji može se reći da je IS skup aplikacionih objekata koji između sebe komuniciraju isključivo preko baznog objekta. Razvoj sistema se nadalje realizuje kroz "fizičko" projektovanje i implementaciju ovih objekata.

Svaki atomska logički proces se može dekomponovati na fizičke procese, pogodne za implementaciju u konkretnoj implementacionoj okolini. Svaki elementarni fizički proces se, na kraju, dekomponuje na slojeve definisanog implementacionog modela, što ne znači da svaki mora posedovati sva četiri sloja.

Dugacke transakcije. Pri dekompoziciji atomske logičke procesa na fizičke prvi korak treba da bude identifikacija i dekompozicija "dugačkih" atomske logičke transakcije. Neki specifikovani atomske procesi mogu da budu interaktivni procesi formiranja veoma složenih aplikacionih objekata, procesi koji mogu trajati nekoliko sati ili dana. Pre nego što se aplikacioni objekat u celini ne formira, ne može se izvesti odgovarajuća operacija održavanja baznog objekta. Da bi se sačuvali međurezultati pri formiranju složenog objekta neophodno je uvesti "pomoćni bazni objekat", odnosno pomoćnu bazu. Atomska logički proces zajedno sa pomoćnom bazom pretstavlja potpuno novi sistem, i njegovom razvoju se može pristupiti na potpuno isti način: može se dekomponovati na svoje aplikacione objekte (logičke atomske transakcije u odnosu na pomoćnu bazu) i svoj bazni objekat. Sa polaznim sistemom ovaj novi sistem komunicira preko nove

operacije osnovnog baznog objekta "inicijalizacija", kojom se inicijalizuje početno stanje pomoćnog baznog objekta na osnovu stanja osnovnog baznog. Ova operacija, kao i operacija održavanja osnovnog baznog objekta poziva se iz pomoćnog sistema.

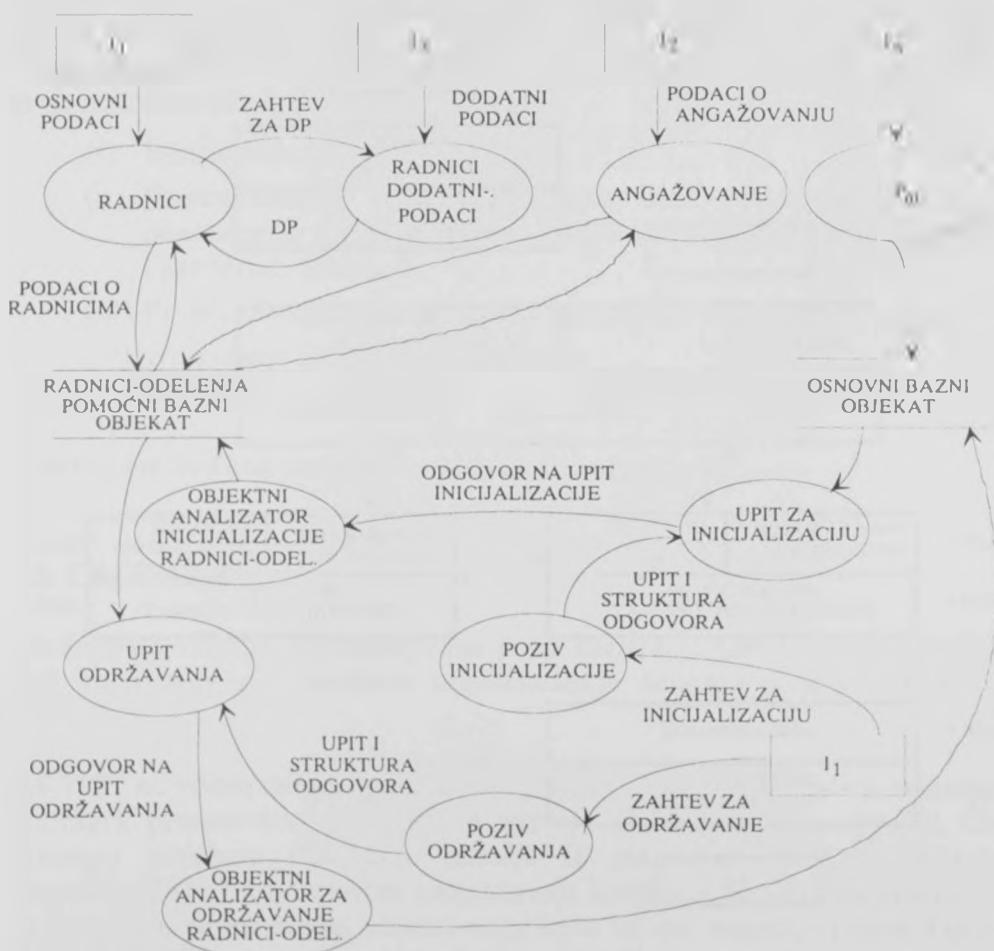
Na Slici 4.7 dat je primer dekompozicije složenog objekta Radnici_odelejna na pomoćne aplikacione objekte i pomoćni bazni objekat. Pomoći aplikacioni objekti Radnici i Angažovanje su "nezavisni", komuniciraju samo preko pomoćne memorije. Radnici i njihovo angažovanja se unose posebno, a provera integriteta se vrši pre nego što se pozove operacija održavanja osnovnog baznog objekta. Struktura pomoćnog baznog objekta je odgovarajući podmodel PMOV osnovnog baznog objekta, sa relaksiranim dinamičkim pravilima integriteta, da bi se dekompozicija na nezavisne pomoćne aplikacione objekte mogla izvršiti. Specifikacija ovog i pomoćnih baznih objekata je očigledna, pa se dalje ne daje.



Slika 4.7. Primer dekompozicije sa pomoćnim baznim objektom

Međuprocesna komunikacija. U daljem fizičkom projektovanju, može se svaki proces dekomponovati na podprocese koji između sebe komuniciraju direktno. Tradicionalno, dva lokalna procesa na istom računaru mogu da komuniciraju preko deljivih lokalnih radnih memorija. Međutim, savremeni operativni sistemi i mrežni protokoli dozvoljavaju i međuprocesorsku komunikaciju između dva čvora u mreži. Otvorenost aplikacije i efikasno adaptivno održavanje sistema zahteva da se ova dekompozicija izvrši nezavisno od toga gde će se međusobno zavisni procesi nalaziti. Sloj za prenos podataka treba da omogući ovakvu vrstu transparentnosti.

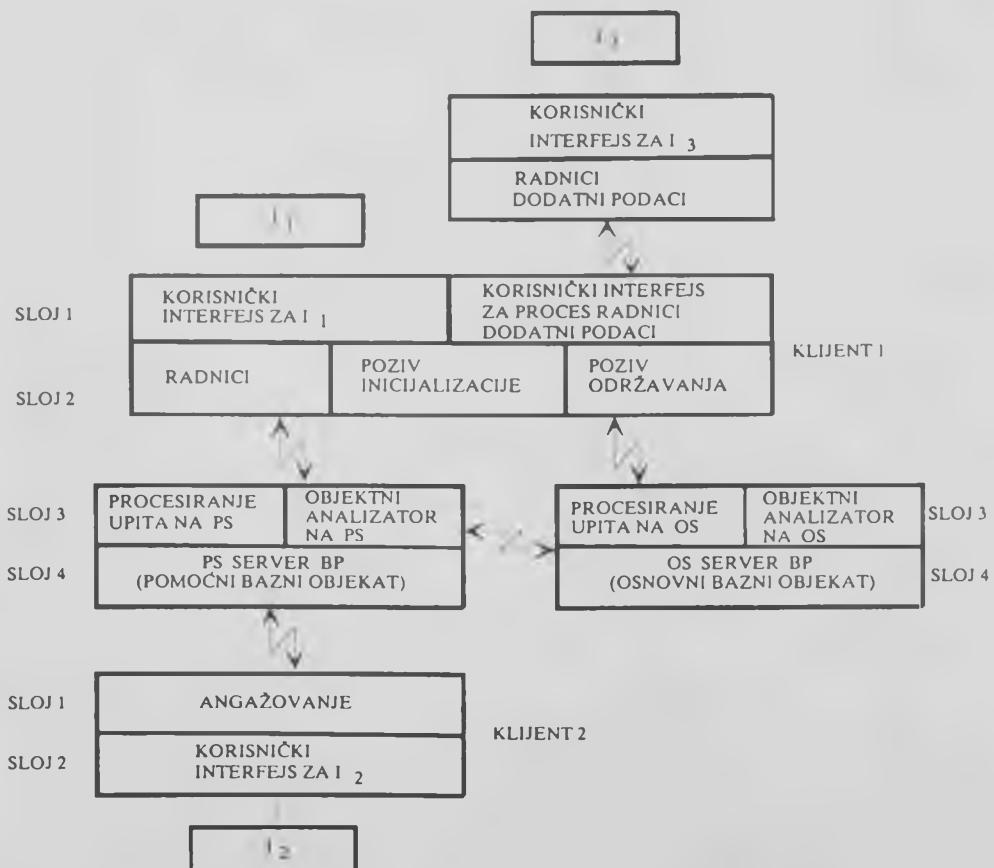
Na Slici 4.8 data je konačna dekompozicija procesa sa slike 4.7.



Slika 4.8. Primer dekompozicije na međuzavisne procese

Proces Radnici se sada obavlja na osnovu dva toka podataka: Osnovni podaci koji interaktivno daje interfejs I1 i DP koji se, na zahtev, dobijaju iz procesa Radnici_dodatni_pod, koji te podatke dobija od interfejs I3. Procesi Inicijalizacija i Promeni_radnici_odelenja takođe se dekomponuju na po tri podprocesa, odgovarajuci poziv operacije, postavljanje upita nad jednom bazom i ažuriranje druge.

U daljem procesu fizičkog projektovanja vrši se distribucija aplikacija, pojedini procesi se dodeljuju slojevima i različitim čvorovima mreže i na taj način ostvaruju različite "client-server" arhitekture. Jedna moguća "client-server" arhitektura primenjena za dati primer pretstavljena je na Slici 4.9.



Slika 4.9. Moguća "client-server" arhitektura

Pretpostavlja se da su pomocna baza i osnovna baza implementirane na dva posebna servera sa različitim sistemima za upravljanje bazom podataka. Aplikacioni interfejsi su pridruženi ovim serverima i oba poseduju softver za interpretaciju i izvršavanje upita i "objektnog analizatora" nad odgovarajućim sistemima za upravljanje bazom podataka. Dva prikazana klijenta, na svojim slojevima Jezgra aplikacije, obuhvataju pojedine procese sa slike 4.8, a preko Sloja 4 i odgovarajuće korisničke interfejsne.

4.5.3. Fizičko projektovanje baznih objekata (baza podataka)

Fizičko projektovanje baznih objekata, odnosno baza podataka, pored prevođenja specifikacije date preko PMOV u neki implementacioni model (relacioni, mrežni, konkretni sistem za upravljanje bazom podataka), svodi se na nekoliko dobro poznatih koraka [18]:

- (i) Distribucija baze;
- (ii) Restruktuiranje logičke strukture baze podataka da bi se minimizirao volumen prenosa podataka između centralne jedinice i perifernih memorija;
- (iii) Projektovanje formata zapisa i kompresija podataka;
- (iv) Grupisanje (klasterovanje) zapisa;
- (v) Izbor metoda pristupa;

O detaljima fizičkog projektovanja ovde se neće govoriti.

4.6. CASE ALATI

Izložena metodologija primenjivana je uz korišćenje CASE alata ARTIST koji je razvijen na Fakultetu organizacionih nauka i detaljno opisan u [19,20,21].

ARTIST najvećim delom spada u kategoriju "upper CASE" alata, odnosno podržava prvenstveno prve faze u razvoju IS, specifikaciju sistema. On potpuno podržava Planiranje razvoja IS metodom "Business System Planning (BSP)", koja u ovom projektu nije korišćena, Strukturu sistemsku analizu i Prošireni model objekti-veze kako su oni napred opisani. Pored toga ARTIST generiše opise baza podataka za različite sisteme za upravljanje bazom podataka korišćene u ovom i u drugim projektima, generiše opise strukture aplikacionih objekata i, za neke sisteme za

upravljanje bazom podataka generiše i procedure za operacije održavanja baze koje čuvaju integritet baze, na osnovu definisanih dinamičkih pravila integriteta.

U poglavljima koja slede biće prikazani modeli koji su razvijeni korišćenjem ARTIST-a, a posebno će se diskutovati njegova uloga u formiranju i održavanju Rečnika podataka.

4.7. PRIMENA METODOLOGIJE

Navedimo na kraju eksplizitno, zaključujući ovo poglavlje, redosled aktivnosti (koraka) koje se obavljaju u primeni Standardne metodologije:

1. Analiza sistema primenom metode Strukturne sistemske analize;
2. Objektno orjentisana specifikacija informacionog sistema (softvera): Specifikacija aplikacionih i baznog objekta;
3. Analiza i dalja dekompozicija pojedinih aplikacionih objekata koji pretstavljaju "dugačke transakcije" na pomoćne aplikacione objekte i pomoćni bazni objekat;
4. Fizičko projektovanje svih baznih objekata, odnosno odgovarajućih baza podataka;
5. Dekompozicija osnovnih i pomoćnih aplikacionih objekata na manje celine koji su međusobno povezani direktnom komunikacijom procesa;
6. Distribucija objekata (procesa) po slojevima opsteg implementacionog modela, prostorna distribucija i definisanje "client-server" arhitekture celokupnog sistema.

LITERATURA

- [1] Lazarević, B., Nešković, S., *Objektno-orjentisana transformaciona metoda razvoja informacionih sistema*, Zbornik radova SYMOPIS '90, pp 3 - 7, Dubrovnik, 1990.
- [2] Conger S., The New Software Engineering, Wasfworth Publishing Co. 1994.
- [3] Hutt, E.T.F. editor, Object Analysis and Design: Comparison of Methods, John Wiley, 1994.

- [4] DeMarco, T., Structured Analysis and System Specification, New York, Yourdon Press, 1978.
- [5] Page-Jons, M., The Practical Guide to Structured System Design, New York, Yourdon Press, 1980.
- [6] Lazarević, B., Nešković, S., Marjanović, Z., Bataveljić, P., Prošireni model objekti-vezce, Beograd, FON, ILIS 01/01, 1991.
- [7] Chen, P., *The Entity-Relationship Model: Toward Unified View of Data*, ACM TODS, Vol 1, No 1, 1976.
- [8] Iivari, J., Koskela, E., *An Extended EAR Approach for Information System Specification*, Proceedings of the Entity Relationship Approach to Software Engineering, Norh-Holland, 1983.
- [9] Lazarević, B., Ilić, S., Marjanović, Z., *An approach to computer-aided transformational method of information system development*, Proceedings of the International Academy of Management and Marketing (II), Dalas, April 1990.
- [10] Lazarević, B., Misić, V., *Extending the entity-relationship model to capture dynamic behaviour*, Eur. J. Inf. Systems, Vol. 1, No. 2, 1991.
- [11] Pistor, P., Dadam, P., An Advanced Information Management Prototype in Nested Relations and Complex Objects in Databases, Lecture and Notes in Computer Science, Springer Verlag, 1987.
- [12] Loomis, M. E. S., Object Databases, The Essentials, Addison Wesley, 1995.
- [13] Ullman, J.D., Database and Knowledge-base Systems, Vol I, Maryland, Computer Science Press, 1988.
- [14] Warnier, J.D., Logical Construction of Programs, Van Nostrand, 1976.
- [15] Jackson, M., System Development, Prentice Hall, 1983.
- [16] Judah, N., *Developing Applications for the Client-Server Model*, Microsoft Development Library, April 1995.
- [17] McGregor, J.D., Sykes, D.A., Object-Oriented Software Development: Engineering Software for Reuse, Van Nostrand Reinhold, New York, 1992.
- [18] Teorey, D.A., Fry, J.P., Design of Database Structures, Prentice Hall, 1982.
- [19] Lazarević, B., Nešković, S., *Objektno orijentisani transformacioni razvoj jednog CASE sistema*, Zbornik radova, Nove tendencije u informatici, Beograd, 1992.

- [20] Lazarević, B., Nešković, S., *Objektno-orientisana metodologija razvoja informacionih sistema i CASE alat koji je podržava*, IS '93, Zbornik radova, SANU, Institut za industrijske sisteme, Novi Sad, pp 051-054, 1993.
- [21] Nešković, S., Priručnik za CASE alat ARTIST, FON, 1995, interna publikacija.

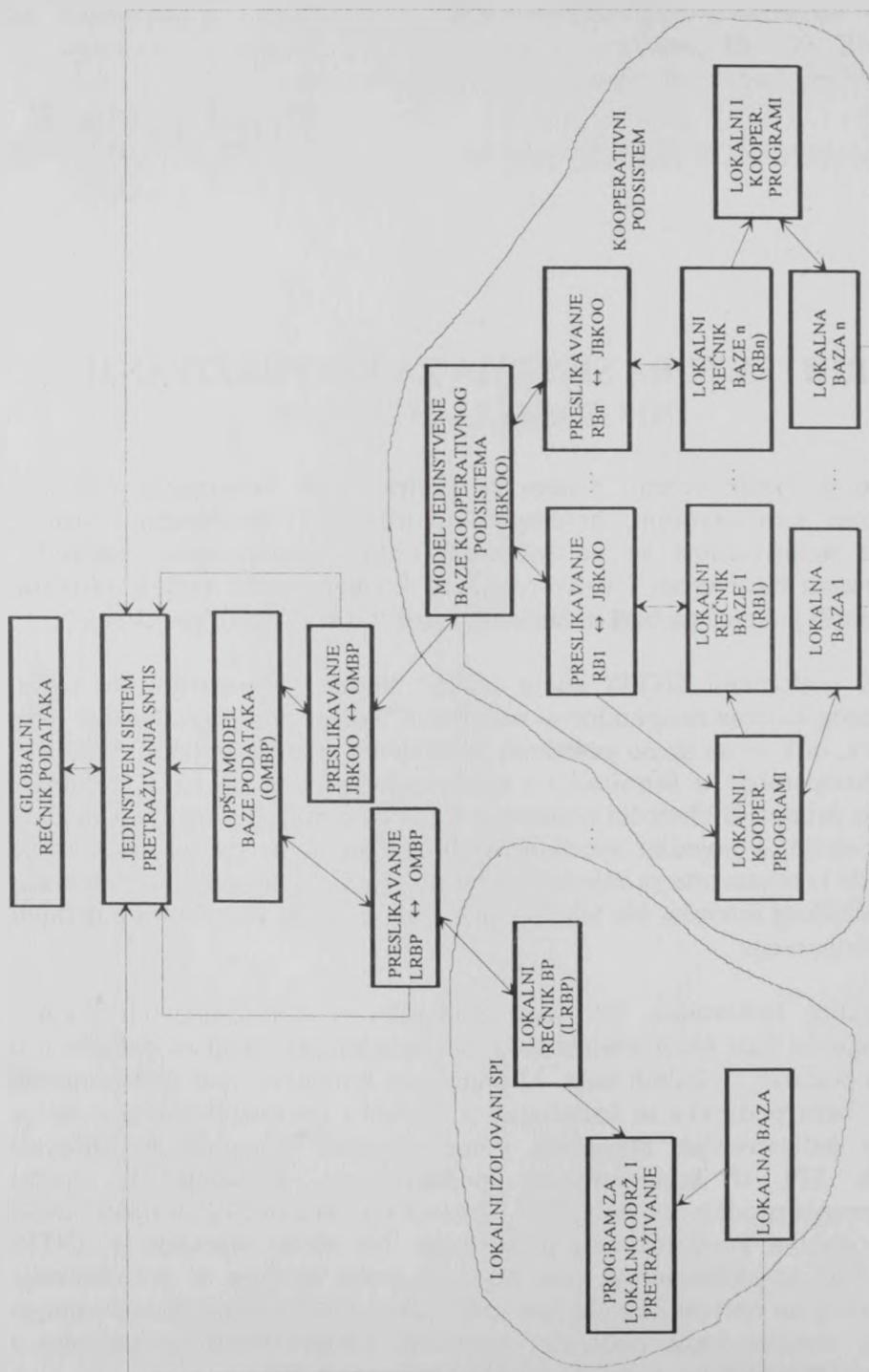
Poglavlje 5.

ARHITEKTURA SISTEMA ZA PRETRAŽIVANJE INFORMACIJA U SNTIS

Kao što je ranije rečeno Sistem za pretraživanje informacija (SPI) je koncipiran kao otvoreni, heterogeni distribuirani informacioni sistem. Njegova jedinstvenost se ne ostvaruje centralizacijom baza podataka, jedinstvenim hardverom i softverom, već kooperativnim radom različitih autonomnih jedinica na bazi definisanih i striktno korišćenih standarda.

Različiti podsistemi SNTIS imaju različit stepen kooperacije. Na nivou celokupnog sistema neophodno je obezbediti jedinstveno pretraživanje baza podataka, dok se na nivou pojedinih podsistema mogu obezbediti i različiti vidovi kooperacije u formiranju i održavanju zajedničkih baza podataka. Tako, na primer, Bibliotečki podsistem treba da omogući funkciju uzajamne katalogizacije, a pojedini specijalizovani sistemi mogu da ustanove svoje standarde i mehanizme za zajedničko formiranje i održavanje baza podataka od zajedničkog interesa. Na Slici 5.1 prikazan je odnos i struktura pojedinih nivoa kooperacije.

U lokalnim izolovanim SPI baze podataka se formiraju, održavaju i pretražuju na bazi specifičnih standarda i mehanizama koji su definisani u rečniku podataka lokalnih baza. U pojedinim kooperativnim podsistemima lokalne baze podataka se formiraju, održavaju i pretražuju na bazi što je moguće jedinstvenijih standarda i mehanizama. Pojedine specifičnosti lokalnih SPI u kooperativnim podsistemima savlađuju se preko preslikavanja modela lokalnog SPI i (rečnika podataka SPI) u opšti model baze podataka kooperativnog podsistema. Na nivou celokupnog SNTIS obezbeđuje se jedinstvenost pretraživanja preko sistema za pretraživanje zasnovanog na opštem modelu baze podataka i preslikavanja između ovoga modela, modela baza podataka pojedinih kooperativnih podsistema i modela baza podataka pojedinih lokalnih izolovanih SPI.



Slika 5.1 Struktura nivoa kooperacije u distribuiranom SNTIS

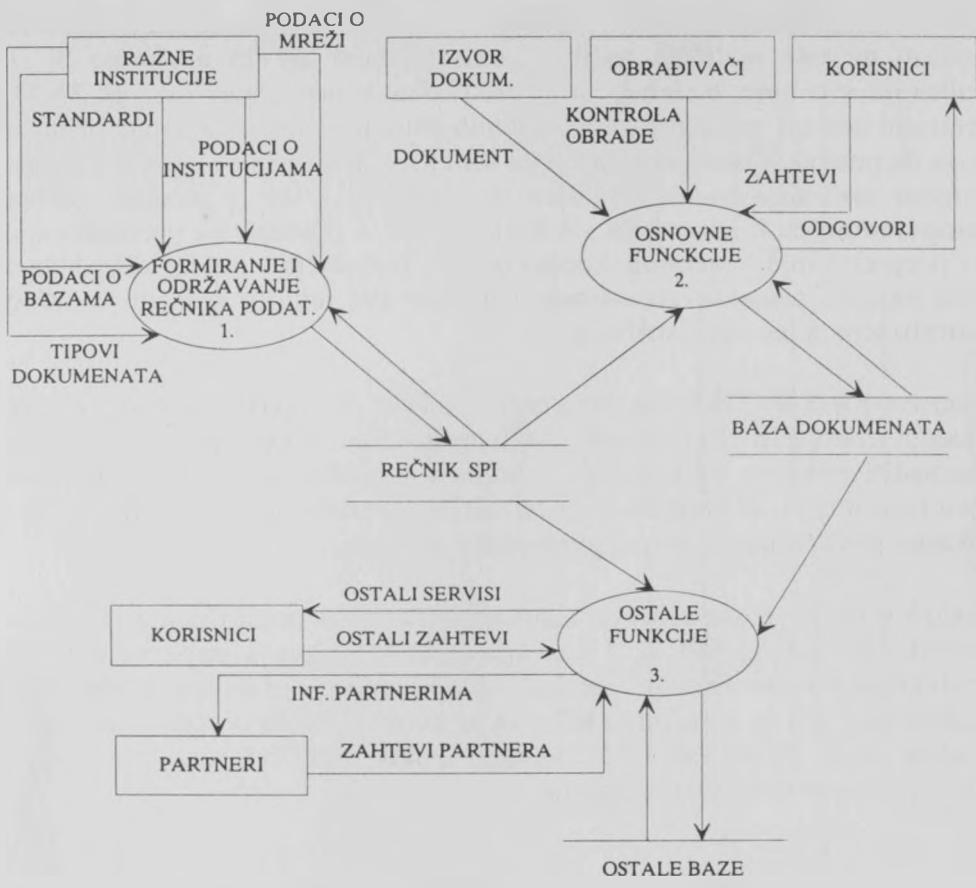
Distribucija celokupnog SPI i pojedinih kooperativnih podsistema može se ostvariti na više različitih načina. Centralni host SNTIS na kome bi se replicirale sve baze podataka nije predviđen koncepcijom razvoja SNTI. Centralni hostovi pojedinih kooperativnih sistema (Bibliotečkog, na primer) mogu da postoje. Osnovna koncepcija SNTIS je distribucija sa što je moguće manjom centralizovanom replikacijom podataka i što je moguće većom transparentnošću u odnosu na sve funkcije SPI, a posebno na pretraživanje. Pri projektovanju pojedinih kooperativnih podsistema model distribucije treba pažljivo projektovati uzimajući u obzir sve faktore koji idu u prilog centralizaciji, odnosno distribuciji.

Jedinstveni SPI SNTIS treba da se razvija od već postojećih i novoizgrađenih lokalnih izolovanih SPI i kooperativnih podsistema. Zbog toga je neophodno obezbediti efikasno preuzimanje i migraciju "softverske zaostavštine" na novu tehnologiju, uz najveću moguću zaštitu dosadašnjeg investiranja i njenog efikasno povezivanje sa novoizgrađenim sistemima.

Imajući u vidu sve ove zahteve, u prvom delu ovoga poglavlja daje se opšta specifikacija jednog SPI, a u drugom se definiše opšta implementaciona arhitektura sistema. Detaljni opisi pojedinih komponenti daju se u posebnim poglavljima. SPI je koncipiran tako da se kombinacijom pojedinih njegovih modula mogu realizovati kako pojedini izolovani ili kooperativni sistemi, tako i jedinstvo celokupnog sistema.

5.1. SPECIFIKACIJA SISTEMA ZA PRETRAŽIVANJE INFORMACIJA

Na Slici 5.2 pretstavljen je dijagram toka podataka (DTP) prvog nivoa za opšti SPI. Pored procesa OSNOVNE FUNKCIJE SPI (održavanje BP, indeksiranje, pretraživanje, izveštavanje i publikovanje, razmena) i ADMINISTRATIVNIH I OSTALIH FUNKCIJA koje su specifične za pojedine lokalne SPI i kooperativne sisteme, SPI mora da obuhvati i funkciju FORMIRANJE I ODRŽAVANJE REČNIKA PODATAKA. Nadalje se detaljnije opisuju samo Osnovne funkcije SPI i Formiranje i održavanje rečnika podataka, dok je skup Administrativnih i ostalih funkcija interesantan samo za lokalne sisteme i detaljnije se opisuje u poglavljju o Bibliotečkom sistemu.



Slika 5.2. Funkcije softvera za SPI

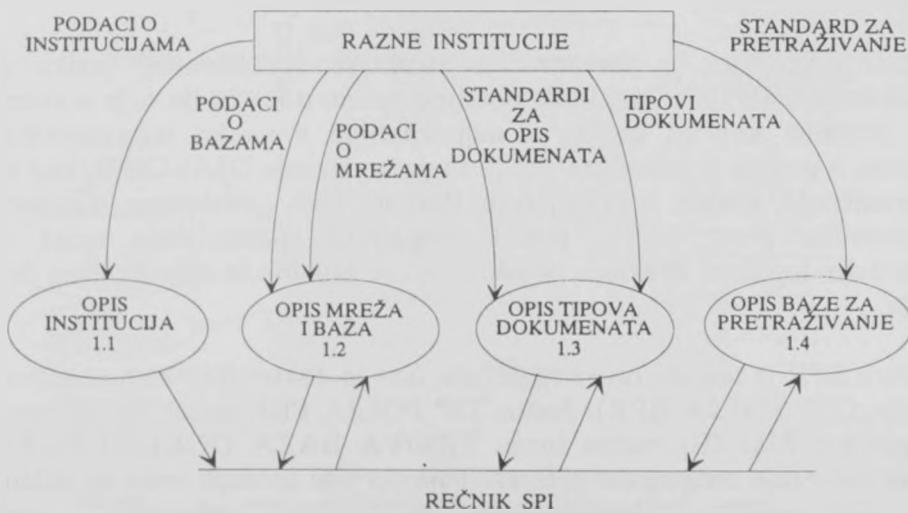
5.1.1. Formiranje i održavanje Rečnika podataka

Rečnik podataka u heterogenom, distribuiranom i otvorenom SPI ima dvostruku ulogu:

- Pasivnu, da predstavlja bazu podataka sa opisom celokupnog sistema i time bude, s jedne strane "vodič" korisnicima sistema, a sa druge osnova za dalji razvoj sistema;
- Aktivnu, transformacionu, odnosno interpretativnu, da transformiše iste logičke specifikacije u različite implementacije, odnosno da na bazi definisanih opštih struktura i pravila, interpretira konkretne akcije u pojedinim aplikacijama. Drugim rečima, Rečnik treba da bude i osnova za pristup razvoju aplikacija,

odabran u ovom Projektu, koji se naziva "Rečnikom vođene aplikacije". Naime, da bi se ostvarilo funkcionisanje softvera za SPI u heterogenoj sredini, neophodno je da sam softver bude specifikovan i implementiran apstraktno, a da se preslikavanja između pojedinih komponenti i izvršenje u specifičnim okolinama dobijaju interpretacijom apstrakcija preko odgovarajućih rečnika podataka i pravila preslikavanja.

Na Slici 5.3 prikazan je DTP za funkciju Formiranje i održavanje Rečnika podataka. Ova funkcija se sastoji od dve grupe nezavisnih procesa: (i) grupe procesa za formiranje podataka o institucijama, računarskim mrežama, čvorovima u njima i bazama podataka u čvorovima i slično i (ii) grupe procesa za formiranje i održavanje baza podataka o različitim standardima i opisivanje pojedinačnih komponenti sistema (baza podataka, tipova dokumenata) preko tih standarda. Posebno se definišu standardi za razmenu podataka, standardi za opis strukture baze za formiranje i održavanje dokumenata, standardi za struktuiranje baze podataka za pretraživanje i standardi za izveštavanje i publikovanje.



Slika 5.3. Formiranje i održavanje Rečnika podataka

Rečnik podataka je detaljno opisan u poslednjem poglavlju, pojedini modeli standarda u poglavljima koji se na njima baziraju, a ovde se, za potrebe objašnjenja pojedinih komponenti sistema, daju samo neki njegovi podmodeli.

Baza dokumenata može da bude veoma specifične strukture, počev od neke strogo strukturirane baze (na primer relacione, predstavljene skupom tabela), do tzv. "full text" baza formiranih po nekom standardu za opis ovoga tipa baza. Međutim, standardni jezik za pretraživanje dokumenata, pa samim tim i jedinstveni sistem za indeksiranje, prepostavljuje veoma jednostavnu strukturu baze dokumenata - baza dokumenata predstavlja "ravnu" strukturu, agregaciju (listu) polja. Ako stvarna baza poseduje složeniju strukturu, neophodno je definisati odgovarajuću ravnu strukturu i pravila (algoritam) transformacije složene u ravnu strukturu. U poglavlju 7 daje se ovaj algoritam za sve baze dokumenata strukturirane na osnovu SGML (Standard Generalized Markup Language) standarda.

Struktura Rečnika podataka za opis struktura "ravnih" baza dokumenata prikazan je preko modela objekti-veze na Slici 5.4. Levi deo slike definise TIP BAZE DOKUMENATA, TIPOVE POLJA BD i njihov REDOSLED u dokumentu. Takođe se u Rečniku čuva i informacija o ČVOROVIMA MREŽE na kojima postoji odgovarajući tip baze dokumenata, za potrebe distribuiranog pretraživanja. Desni deo Slike 5.4 predstavlja strukturu baze za pretraživanje (BPR).

Struktura ove baze je izvedena iz strukture Jedinstvenog jezika za pretraživanje u SNTIS. Ovaj jezik, detaljno opisan u Poglavlju 6, je u svemu sličan jezicima koji se koriste u najpoznatijim svetskim informacionim servisima, a gotovo je identičan jeziku za pretraživanje DIALOG-a, koji se, uz ograničenja, koristi u postojecem Bibliotečkom podsistemu. Osnovne karakteristike ovog jezika, pored mogućnosti postavljanja upita sa standardnim logičkim izrazima, objasnjavaju se zajedno sa opisom ovog dela modela.

Struktura BPR je takođe ravna struktura, čine je dokumenta koja se sastoje od polja (TIP POLJA BPR). Jedan TIP POLJA BPR može da odgovara (agregacija OPISUJE) većem broju TIPOVA BAZA DOKUMENATA, čime se ostvaruje mogućnost pretraživanja po više srodnih baza sa jednim upitom. Odnos TIPI POLJA BPR i TIPI POLJA BD prikazan je vezom OBHVAT koja pokazuje koja su polja baze dokumenata obuhvaćena jednim poljem strukture baze za pretraživanje. Veza OBHVAT može da poveže TIPI POLJA BPR i TIPI POLJA BD samo za isti TIP BAZE DOKUMENATA.

TIP POLJA BPR u jednom TIPI BAZE DOKUMENATA može biti pretraživ ili nepretraživ (specijalizacija PRETRAŽIVOST agregacije

OPISUJE). Nepretraživo polje se može samo prikazati u rezultatu. Kao atributi pretraživog polja javljaju se:

- **NAČIN-INDEKS**, koji govori o tome da li se sadržaj polja razbija na reči i indeksira preko njih i/ili se tretira kao jedinstvena "fraza";
 - **VRSTA-OPERATORA**, koji govori o tome da li se u pretraživanju po posmatranom polju mogu koristiti "operatori položaja" (ista rečenica, udaljenost i slično);
 - **SORTIRANOST**, u kome se naznačava da li se odgovor može sortirati po vrednosti polja;
 - **PRIMER-UPITA**, u kome se daje primer upita po tom polju.



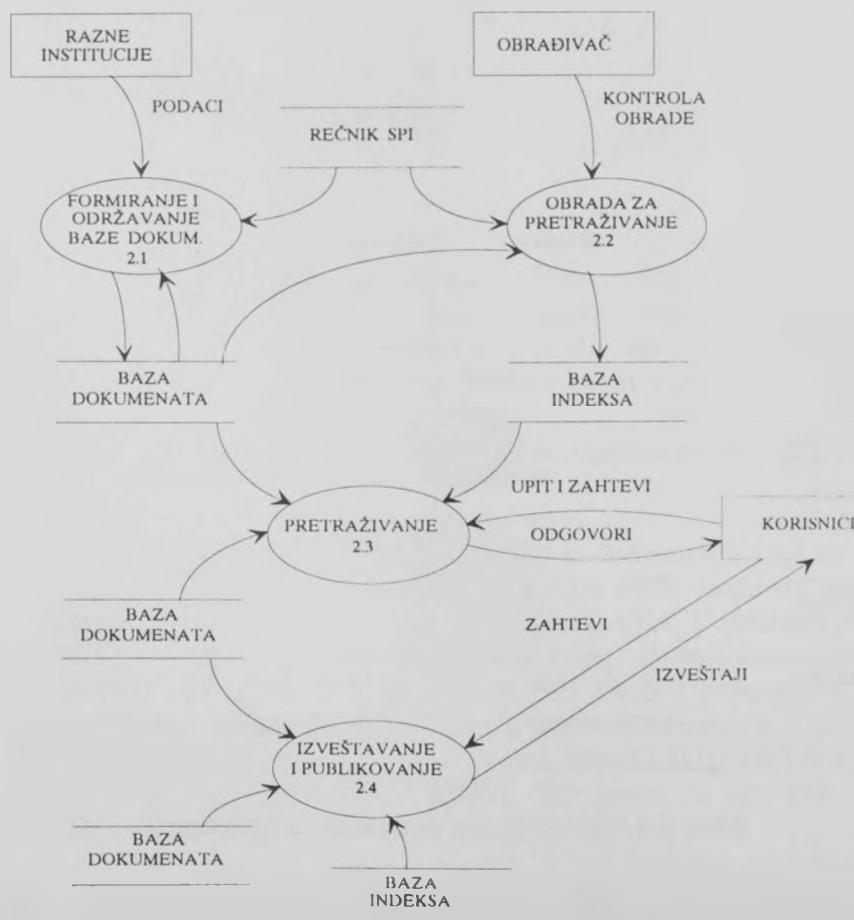
Slika 5.4. Deo Rečnika podataka za opis baze

Pretraživa polja mogu ulaziti u "bazni indeks" (podtip BAZNI INDEKS neeksluzivne specijalizacije VRSTA_IN) ili pretstavljaju dodatne indekse (podtip DODATNI INDEKS iste specijalizacije). Atribut REDBR definiše

redosled polja u baznom indeksu. Bazni indeks predstavlja grupu polja nad kojim se u celini, ako drugačije nije naglašeno, pretraživanje po zadatom upitu uvek izvršava. Pretraživanje se može suziti u polaznom upitu ili naknadnim "kvalifikovanjem" rezultata polaznog upita, na delove baznog indeksa sa navedenim sufiksima(atribut SUFIKS). Dodatni indeksi predstavljaju pojedina polja nad kojima se može izvršiti upit zadavanjem njihove vrednosti u polaznom upitu, bez naknadne kvalifikacije rezultata upita. Atribut PREFIKS predstavlja naziv polja koji se koristi u upitu.

5.1.2. Osnovne funkcije Sistema za pretraživanje informacija

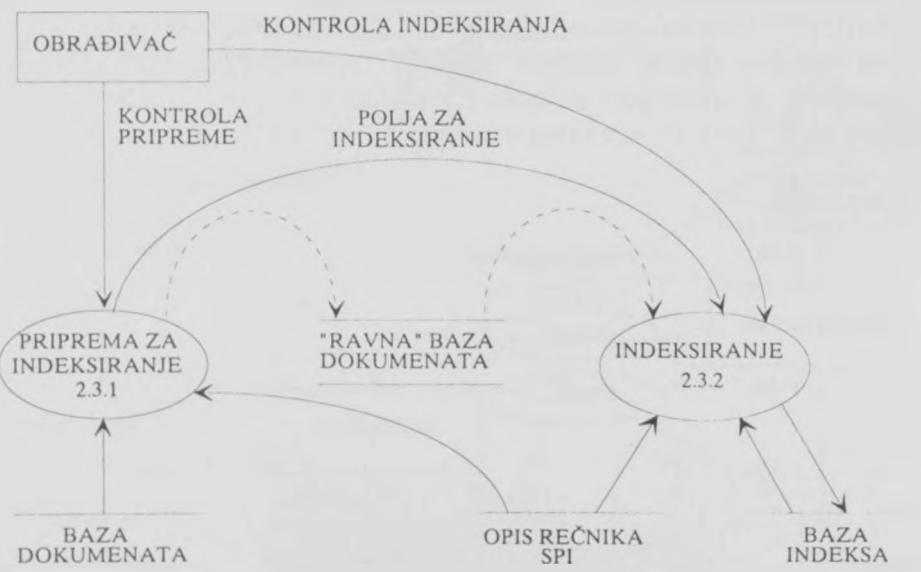
Na Slici 5.5. dat je DTP koji opisuje osnovne funkcije SPI.



Slika 5.5. Osnovne funkcije SPI

Proces Formiranje i održavanje baze podataka obavlja se lokalno, korišćenjem različitih specifičnih programskih sistema i zbog toga se ovde dalje ne opisuje. Projekat SNTIS, međutim, je obuhvatio jednu opštu široku klasu baza podataka, tzv. "full text" baze i standard SGML za njihovo opisivanje i za tu klasu razvio i ovu funkciju sistema, kako je to opisano u Poglavlju 7.

Proces Obrada za pretraživanje detaljnije je prikazan na Slici 5.6 i čine ga dva osnovna procesa: Priprema za indeksiranje i Indeksiranje. Priprema za indeksiranje je u osnovi postupak "ravnjanja" složene strukture Baze dokumenata, kako je to ranije objašnjeno. Za svaku specifičnu bazu podataka potrebno je razviti poseban program za implementaciju ovog procesa, a opšti program ovakve vrste, za Baze dokumenata opisane SGML standardom, dat je u Poglavlju 7.



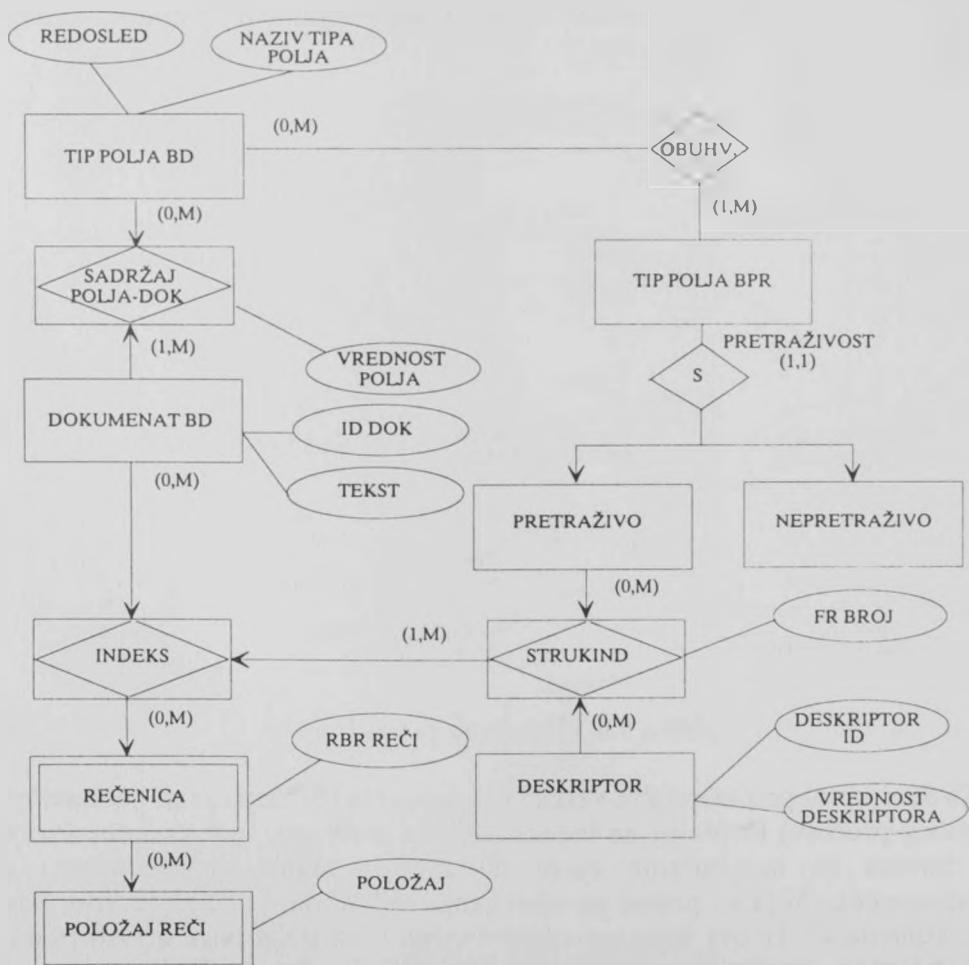
Slika 5.6. Obrada za pretraživanje

Na Slici 5.6 su prikazane dve verzije veze procesa (pa samim tim i dve verzije samog procesa) Priprema za indeksiranje sa procesom Indeksiranje. Proses priprema za indeksiranje može da formira stalnu (redundantnu) ili privremenu (koja se prazni po obavljanju indeksiranja) bazu "Ravna baza dokumenata". Iz ove baze, na osnovu opisa veza polja baze dokumenata i standardne strukture za pretraživanje koji je dat u Rečniku, proces Indeksiranje uzima polje po polje i indeksira ga. Ova mogućnost prikazana je crtkastim tokovima prema skladištu "Ravna" baza dokumenata. Druga

mogućnost je da proces Priprema za indeksiranje, ponovo na osnovu opisa iz Rečnika, u direktnoj međuprocesnoj komunikaciji, dostavlja na indeksiranje polje po polje jednog dokumenta.

Proces Indeksiranje, na osnovu podataka iz Rečnika, razbija sadržaj pojedinih polja na reči, odnosno fraze, pronalazi ih ili ubacuje u Rečnik pojmove (skup ključnih reči - deskriptora) i označava njihov položaj u polju. Proces indeksiranje se obavlja lokalno, za svaku bazu ili njen fragment u jednom čvoru mreže.

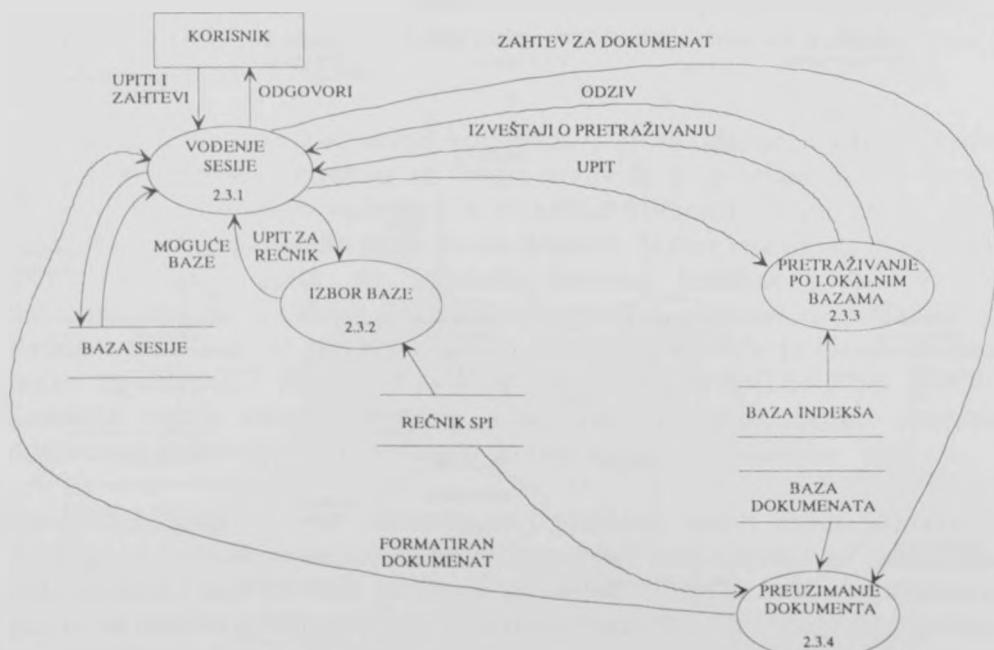
Model Baze indeksa prikazan je na Slici 5.7.



Slika 5.7. Baza indeksa

Pošto se indeksiranje radi lokalno, za jednu bazu TIP POLJA BD nije ovde "slab objekat" kao u Rečniku SPI, koji opisuje više baza. Iz istog razloga Specijalizacija PRETRAŽIVOST se daje za objekat TIP POLJA BPR, a ne za agregaciju OPISAN, kao što je bilo u Rečniku. Pretraživa polja učestvuju, zajedno sa DESKRIPTORIMA (sadržinskim indeksima koji pretstavljaju reči ili fraze), u formiraju INDEKSA nad DOKUMENTIMA. Agregacija STRUKIND sa atributom FRBROJ pokazuje koliko se puta u bazi podataka pojavio dati sadržinski deskriptor u datom polju. Agregacija INDEKS pokazuje u kojim se dokumentima nalazi dati sadržinski deskriptor u datom polju. Ako je na neko polje moguće primeniti i operatore položaja, tada se u objektu REČENICA daje redni broj rečenice u polju datog dokumenta, u kojoj se nalazi posmatrani deskriptor, a u objektu POLOŽAJ REČI, redni broj reči (deskriptora) u rečenici.

Proces Pretraživanje detaljnije je opisan preko DTP na Slici 5.8. Pretpostavlja se pretraživanje u distribuiranoj okolini. Pretraživanje pretstavlja "korisničku sesiju" (Proces Vođenje sesije) u kojoj korisnik postavlja upite i zahteve i dobija i procesira odgovore iz distribuiranog sistema. Korisnička sesija je "dugačka transakcija" i zbog toga poseduje privremenu bazu (Baza sesije).

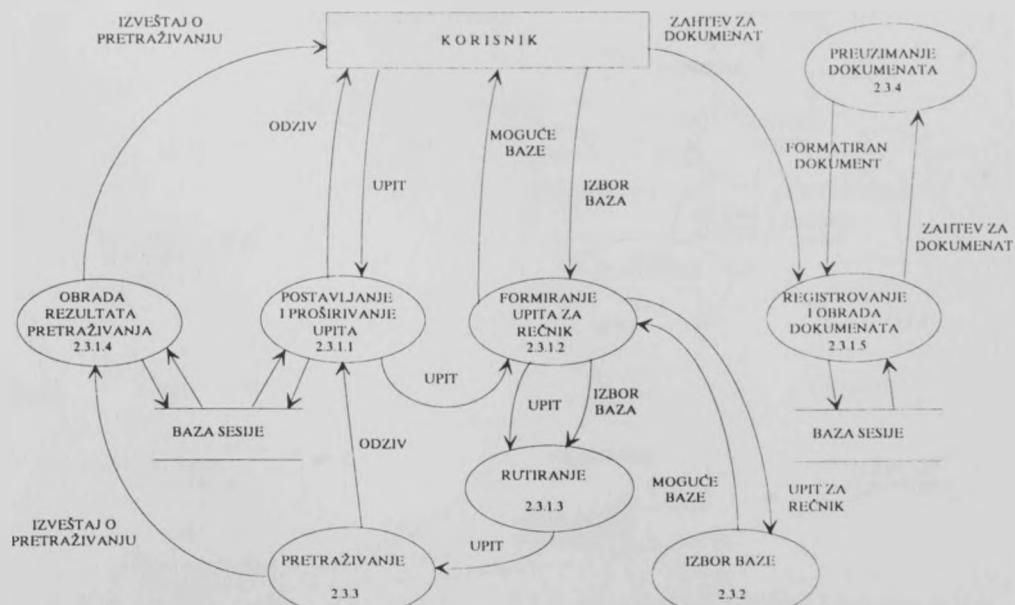


Slika 5.8. Pretraživanje

Samo pretraživanje i preuzimanje dokumenata (po izabranom formatu, čiji se opis čuva u lokalnom rečniku podataka) se obavlja po lokalnim bazama indeksa, odnosno iz lokalnih baza dokumenata. Pretraživanje se obavlja iz dva koraka: (i) "predviđanje odziva" u kome se korisniku saopštava broj odgovora na njegov upit (tok Odziv) i (ii) prikazivanje liste identifikatora dokumenata (i njihovih osnovnih podataka) koji zadovoljavaju upit (Tok Izveštaj o pretraživanju).

Proces Vođenje korisničke sesije detaljnije je prikazan na DTP na Slici 5.9. Čine ga sledeći procesi:

- Postavljanje i "prosirivanje upita", u kome se postavlja upit i menja na osnovu Odziva sistema;
- Formiranje upita za Rečnik, koji iz postavljenog upita selektuje nazine polja za pretraživanje i njihove odnose, da bi se na osnovu tako transformisanog upita izabrale baze za pretraživanje;
- Rutiranje upita, koji na bazi toka Moguće baze i/ili izbora korisnika rutira upit u odgovarajuću lokalnu bazu indeksa;
- Obrada rezultata pretraživanja-identifikacija i osnovni podaci o dokumentima koji zadovoljavaju upit i njihovo registrovanje u bazu sesije;
- Registrovanje i obrada dokumenata.



Slika 5.9. Vođenje sesije

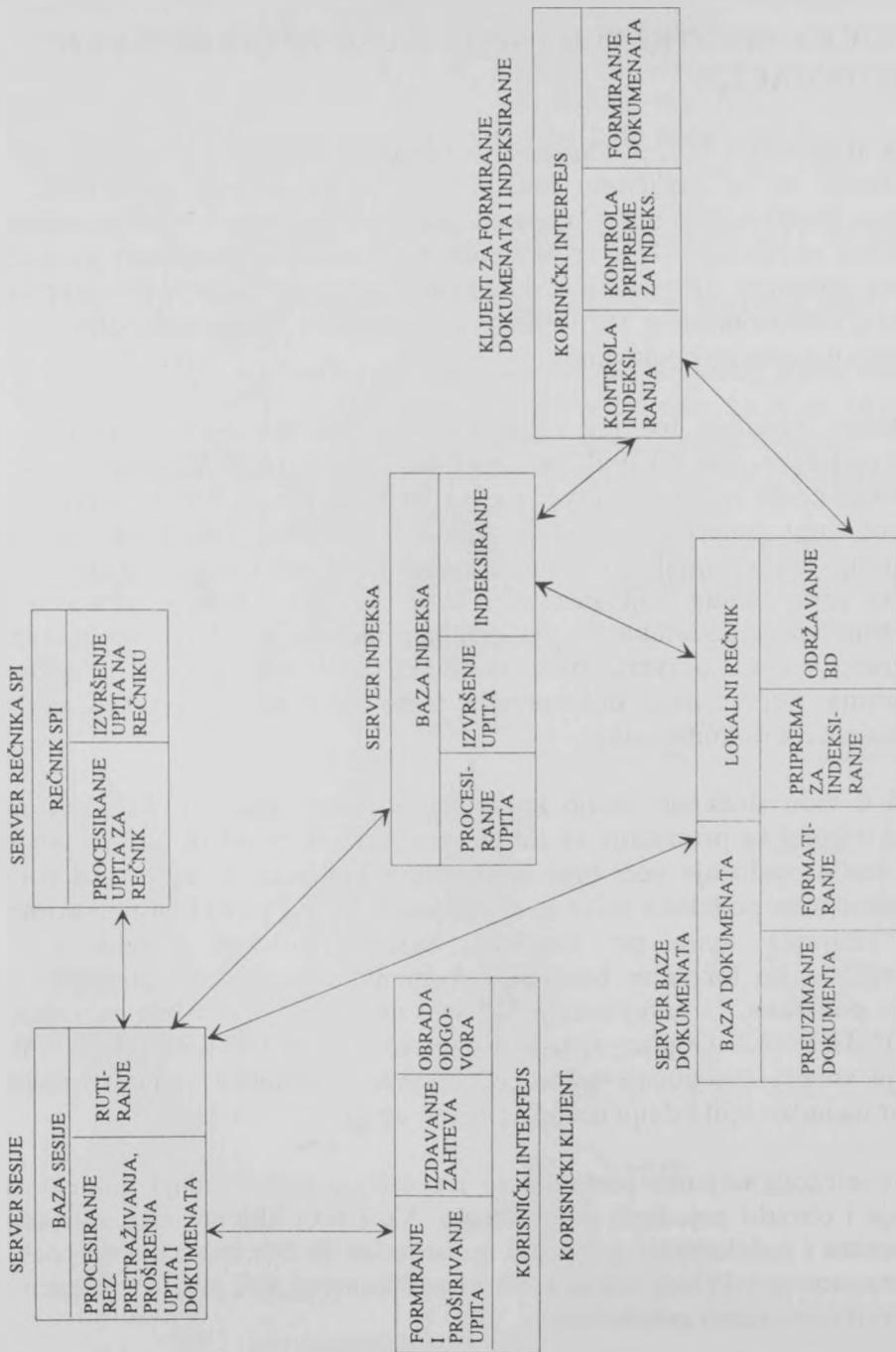
5.2. FIZIČKA ARHITEKTURA SISTEMA ZA PRETRAZIVANJE INFORMACIJA

Logička arhitektura SPI, prikazana u prethodnim delovima ovog poglavlja, transformiše se u različitim lokalnim i kooperativnim sistemima, u konkretan fizički model, u zavisnosti od raspoloživih resursa i drugih zahteva konkrenog okruženja. Ovde se, u skladu sa generičkim implementacionim modelom opisanim u Poglavlju 4, daje opšta fizička arhitektura integralnog, otvorenog i distribuiranog SPI (Slika 5.10). Pojedine komponente detaljnije su opisane u sledećim poglavljima.

Arhitektura poseduje tri tipa servera baza podataka: Server centralnog rečnika podataka, više lokalnih Servera baze indeksa i vise lokalnih Servera baza dokumenata sa lokalnim rečnicima podataka. Na svim ovim serverima u trećem sloju generičkog modela (sloj Aplikacionog interfejsa) vrši se procesiranje upita (sintaksa i semantička analiza i optimizacija) i izvršenje upita, sa programima koji sakrivaju karakteristike lokalnog sistema za upravljanje bazom podataka. Na Serveru baze indeksa se odvija i program za indeksiranje, a na Serveru baze dokumenata i program za pripremu indeksiranja. Server baze dokumenata treba da omogući i formiranje i održavanje baze dokumenata.

Imajući u vidu složenost sesije korisnika, s jedne strane, i distribuiranu okolinu u kojoj se pretražuje sa druge, predviđa se poseban Server sesije. Server sesije opslužuje veći broj Korisničkih klijenata. U njemu se vode privremene baze podataka sesije za sve klijente. Server ima i komunikacionu ulogu, rutiranja upita po lokalnim bazama indeksa i zahteva za dokumentima po lokalnim bazama dokumenata, na osnovu podataka iz Rečnika podataka. U prvoj verziji SPI ova uloga servera je bila ostvarena preko INTERNET Gopher-a, a u drugoj se koristi INTERNET WWW. Sadašnja verzija omogućuje samo jednostavne korisnicke sesije, direktni odgovor na jedan upit i dalju obradu po tom upitu.

Na Korisničkom klijentu postavlja se i proširuje upit i izdaju zahtevi za dobijanje i obradu pojedinih dokumenata. Veći broj klijenta za formiranje dokumenata i indeksiranje povezuje se na jedan Server baze dokumenata, gde se na osnovu lokalnog rečnika, formira dokument, vrši njegova priprema za indeksiranje i samo indeksiranje.



Slik 5.10. Opšta fizička arhitektura SPI

Poglavlje 6.

JEDINSTVENI SISTEM ZA INDEKSIRANJE I PRETRAŽIVANJE DOKUMENATA

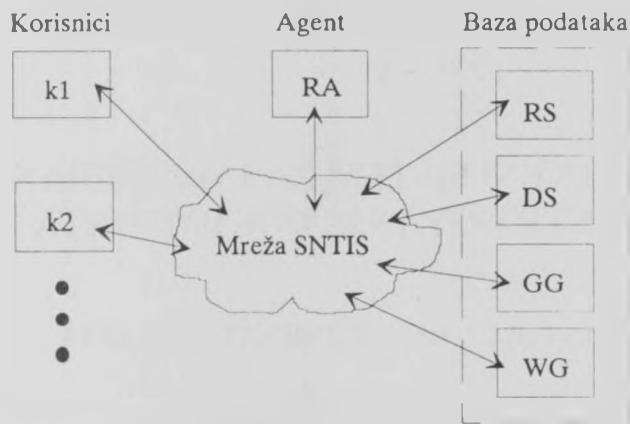
6.1. KARAKTERISTIKE I KOMPONENTE SISTEMA

Jedinstveni sistem za indeksiranje i pretraživanje dokumenata je, za potrebe SNTIS-a, razvijen u Laboratoriji za računarske sisteme Instituta Mihajlo Pupin. Pored ovde prikazane, razvijena je i komercijalna verzija istog sistema, zbog čega se u ovom poglavlju koriste engleska imena modula, procedura, struktura podataka i sl. Sistem ima sledeće karakteristike:

1. Bazama podataka, koje se zasnivaju na ovom sistemu, pristupa se putem računarske mreže SNTIS-a. Ovo se radi preko za ovaj sistem posebno razvijenih protokola, ili putem standardnih Internetovih protokola (Gopher [1], HTTP [2], ...).
2. Sistem se bazira na klijent/server arhitekturi. Zadovoljavajuća efikasnost realizovanih servera omogućava da se sistem koristi, kako za relativno male baze podataka (na primer, rečnik podataka SNTIS-a), tako i za veoma velike, kakve su, recimo, baze hostova bibliotečke mreže.
3. Pretpostavlja se da su dokumenti koji se pretražuju organizovani na isti način kao i u sistemu DIALOG [3], to jest da se sastoje od polja koja imaju svoja imena i koja mogu biti ponovljiva. Ukoliko dokumenti nisu organizovani na ovaj način, mora postojati programska podrška koja konvertuje dokumente u ovaj oblik. Jezik za postavljanje upita je u velikoj meri takođe kompatibilan sa DIALOG-om.
4. Sistem je implementiran na različitim operativnim sistemima (SCO Unix, AIX, OpenVMS).
5. Sistem je projektovan u skladu sa napred definisanim arhitekturom otvorenih sistema. Takođe, specifikacije svih protokola i interfejsa prema

podsistemima, koje su značajne za korišćenje ili eventualnu nadgradnju sistema su javno dostupne.

Opšta organizacija sistema za indeksiranje i pretraživanje dokumenata je prikazana na slici 6.1.



Slika 6.1. Organizacija sistema

Na slici je prikazana samo jedna baza tekstualnih podataka. Ona se sastoji od 4 mrežna servera:

- RS - Server za indeksiranje i pretraživanje dokumenata (*Retrieval Server*).
- DS - Server za pristup bazi izvornih dokumenata (*Document Server*). U bazi izvornih dokumenata se čuvaju dokumenti u svom izvornom obliku. Sistem za pretraživanje ne definiše format ove baze, već isključivo protokol kojim se njoj pristupa.
- GG - Gopher gejtvej (*Gopher Gateway*). Omogućava pristup bazi podataka preko standardnog Gopher protokola.
- WG - World Wide Web gejtvej (*WWW Gateway*). Omogućava pristup bazi podataka preko standardnog HTTP protokola.

U mreži može postojati praktično neograničen broj baza podataka. Adresa svakog servera se sastoji iz adrese računara (standardna Internet adresa) i broja porta (socket-a) preko koga server komunicira. Serveri koji upravljaju jednom bazom podataka mogu, ali ne moraju, biti na istom računaru. Komunikacija između servera se obavlja isključivo preko mreže.

Postoje dve grupe korisnika sistema za indeksiranje i pretraživanje dokumenata. Prva grupa korisnika sistem koristi preko aplikacija koje se

posebno razvijaju za potrebe NTIS, kao što je na primer bibliotečki informacioni sistem. Drugu grupu čine korisnici koji za pristup sistemu koriste standardne Internet klijentske aplikacije WWW i Gopher. Ukoliko nije potrebno da se nekoj bazi dokumenata omogući pristup preko Web-a i Gopher-a, nije potrebno ni da se ovi gejtvjeji uključuju u sistem.

Agenti su komponente informacionih sistema koje u novije vreme doživljavaju sve veću popularnost. U sistemu za indeksiranje i pretraživanje dokumenata predviđena je realizacija jednog servisa koji se bazira na ovoj tehnologiji - RA (*Retrieval Agent*). Ideja je da korisnik agentu zada upite koji treba da izdvoje dokumente koji su mu interesantni. Agent nadalje periodično proverava da li su se u bazi pojavili dokumenti koji zadovoljavaju postavljene uslove, a o kojima korisnik nije obavešten. Ukoliko ovakvi dokumenti postoje, korisnik se o njima obaveštava preko elektronske pošte.

6.2. ARHITEKTURA SISTEMA

6.2.1. Arhitektura servera

Kao što je već rečeno, arhitektura sistema za pretraživanje informacija se bazira na generičkom implementacionom modelu opisanom u Poglavlju 4. U jedinstvenom sistemu za indeksiranje i pretraživanje informacija, serverima odgovaraju treći i četvrti sloj modela:

- a) sloj aplikacionog interfejsa,
- b) sloj servera baze podataka.

Funkcije ovih slojeva u pojedinim serverima date su u sledećoj tabeli:

	Aplikacioni interfejs	Server baze podataka
Server za indeksiranje i pretraživanje dokumenata	<ul style="list-style-type: none">• prihvatanje zahteva i određivanje njihovog tipa• upravljanje procesom indeksiranja i obezbeđivanje dokumenta za indeksiranje• leksička, sintaksna i semantička analiza upita, formiranje plana izvršavanja upita• optimizacija i interpretacija plana izvršavanja upita	<ul style="list-style-type: none">• upravljanje bazom indeksa• indeksiranje dokumenata (upis podataka o poziciji deskriptora u bazu indeksa)• obezbeđivanje podataka o deskriptorima i položaju deskriptora u dokumentu

Server za pristup bazi izvornih dokumenata	<ul style="list-style-type: none">• prihvat zahteva za izvornim dokumentom• formiranje polja izvornog dokumenta	<ul style="list-style-type: none">• održavanje baze izvornih dokumenata
Gopher i WWW gejtvej	<ul style="list-style-type: none">• prihvat zahteva i njihova transformacija iz oblika propisanog Gopher i HTTP protokolom u oblik koji zahtevaju RS i DS serveri• prosleđivanje zahteva RS i DS serveru i prihvat odgovora• formatiranje odgovora i vracanje klijentu koji je postavio zahtev	<ul style="list-style-type: none">• održavanje baze predefinisanih formi dokumenata

O nabrojanim funkcijama ce biti više govora kasnije, kada budu opisivani pojedini serveri.

6.2.2. Komunikacije između servera

Sa stanovišta korisnika sistema, odnosno klijentskih aplikacija, serveri (RS i DS) i gejtveji (GG i WG) se ponašaju kao tipični serveri koji ovim aplikacijama obezbeđuju određene servise. Klijentska aplikacija se za određeni servis obraća samo jednom serveru, i to tako što mu prosleđuje odgovarajuću komandu ili upit. Sledeće komande i upiti zahtevaju za svoje izvršenje učešće više servera:

1. Komanda za indeksiranje koja se upućuje serveru za indeksiranje i pretraživanje zahteva da se pristupa bazi izvornih dokumenata.
2. Upiti za pretraživanje i komande za preuzimanje dokumenata, koji se zadaju iz Gopher i WWW klijenata, zahtevaju da odgovarajući gejtvej pristupa RS i DS serverima.

U principu, postoje dva načina da se ovakvi zahtevi izvršavaju. Prvi je da se izvršavanje zahteva bazira na kooperativnom radu više servera, to jest da komunikacija između servera bude zasnovana na principu ravnopravnih učesnika (*peer to peer*). Drugi način je da se komunikacija između servera obavlja saglasno klijent/server modelu: server koji zahteva uslugu od drugog servera, u ovoj komunikaciji igra ulogu klijenta. U Jedinstvenom sistemu za indeksiranje i pretraživanje dokumenata izabran je drugi način komunikacije između servera. Ovaj način komuniciranja između servera ilustrovan je na primeru zahteva za indeksiranjem dokumenta - slika 6.2.



Slika 6.2. Obrada komande za indeksiranje

Proces indeksiranja se obavlja u četiri koraka:

- Klijentska aplikacija upućuje serveru za indeksiranje i pretraživanje (RS) komandu da se izvrši indeksiranje odgovarajućeg dokumenta.
- Server RS upućuje serveru za pristup bazi izvornih dokumenata (DS) zahtev za dokumentom koji se indeksira.
- DS vraća traženi dokument.
- Po obavljenom indeksiranju RS vraća klijentskoj aplikaciji izveštaj o obavljenom indeksiranju.

Server RS učestvuje u sva četiri koraka procesa indeksiranja i to u komunikaciji sa klijentskom aplikacijom, koraci (a) i (d), ima ulogu servera, a u komunikaciji sa serverom DS, koraci (b) i (c), igra ulogu klijenta. Klijentska aplikacija u procesu indeksiranja komunicira isključivo sa RS serverom i to uvek kao klijent.

Komunikacija standardnih Internet klijentskih aplikacija, Gopher-a i WWW-a, sa odgovarajućim gejtvejima se obavlja na sličan način. U toku obrade zahteva klijentske aplikacije gejtvej se u ulozi klijenta, u zavisnosti od tipa zahteva obraća, bilo RS, bilo DS serveru. U komunikaciji između klijentske aplikacije i gejtveja, gejtvej isključivo ima ulogu servera.

6.2.3. Apstrakcije računarske mreže i baza podataka

Da bi se obezbedila nezavisnost servera od pojedinih tehnoloških rešenja, u Jedinstvenom sistemu za indeksiranje i pretraživanje dokumenata definisu se apstrakcije pojedinih komponenata servera. Ovo se pre svega odnosi na apstrakcije mreže i apstrakcije baza podataka. Apstrakcija komponenti se vrši definisanjem aplikacionih programskih interfejsa (*API-a*) za datu komponentu. U daljem tekstu termini apstrakcija neke programske komponente i API date komponente će se koristiti kao sinonimi.

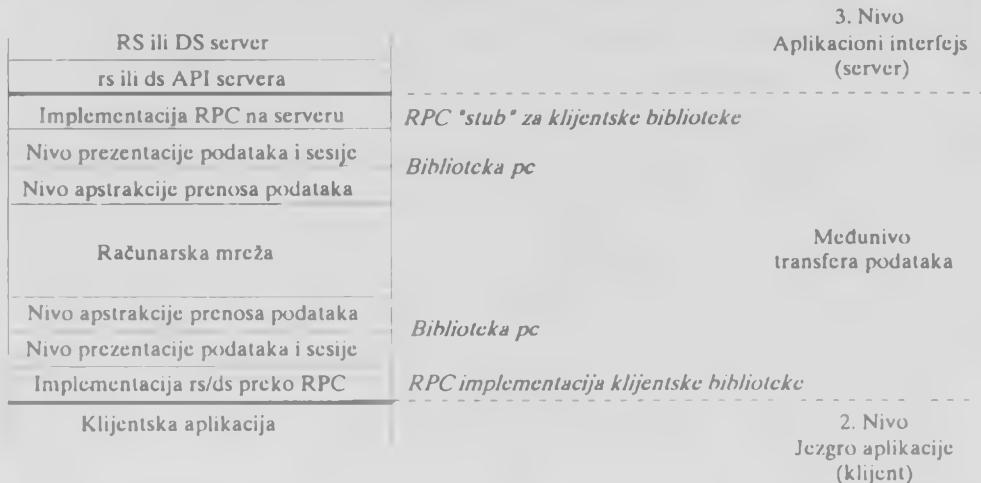
Sa stanovišta arhitekture celokupnog sistema interesantan je, pre svega, način na koji se vrši apstrakcija mreže. Apstrakcija mreže se vrši na tri nivoa:

1. Najviši nivo apstrakcije obezbeđuje *RPC* (*Remote Procedure Call*) mehanizam. *RPC* obezbeđuje da aplikacija, koja u posmatranom procesu komunikacije ima ulogu klijenta, koristi API definisan za pristup aplikacionom interfejsu servera.
2. Nivo prezentacije podataka i sesije definiše format podataka pri prenosu kroz mrežu i pravila uspostavljanja i raskidanja sesije.
3. Nivo prenosa podataka obezbeđuje nezavisnost prethodna dva nivoa od aplikacionog programskog interfejsa transportnog nivoa računarske mreže.

Za sada se u Jedinstvenom sistemu za indeksiranje i pretraživanje dokumenata ne koristi ni jedan od generalnih mehanizama za implementaciju *RPC-a*, kao što su Xerox Courier [4] i Sun RPC [5], i to pre svega iz razloga što nije bilo moguće obezrediti sistemsku podršku za sve platforme na kojima je sistem implementiran. Umesto toga su, za sve interesantnije klijentske platforme, implementirane programske biblioteke koje omogućavaju pristup RS i DS serverima i to prema specifikacijama *API-a* ovih servera (rs [6] i ds [7]). Nivoi prezentacije podataka, sesije i prenosa podataka su definisani u specifikaciji *API-a* komunikacione biblioteke pc [8].

Opisani nivoi apstrakcije, kao što je to prikazano na slici 6.3 zauzimaju prostor između drugog i trećeg nivoa generičkog implementacionog modela.

Kasnije u ovom poglavlju biće dat opis mrežne podrške RS i DS servera. Ovi opisi se sastoje od opisa odgovarajućih *API-a* i protokola (način ostvarivanja sesije i format poruka) preko kojih se realizuje *RPC* mehanizam.



Slika 6.3. Nivoi apstrakcije mreže u odnosu na četvoronivoski model aplikacije

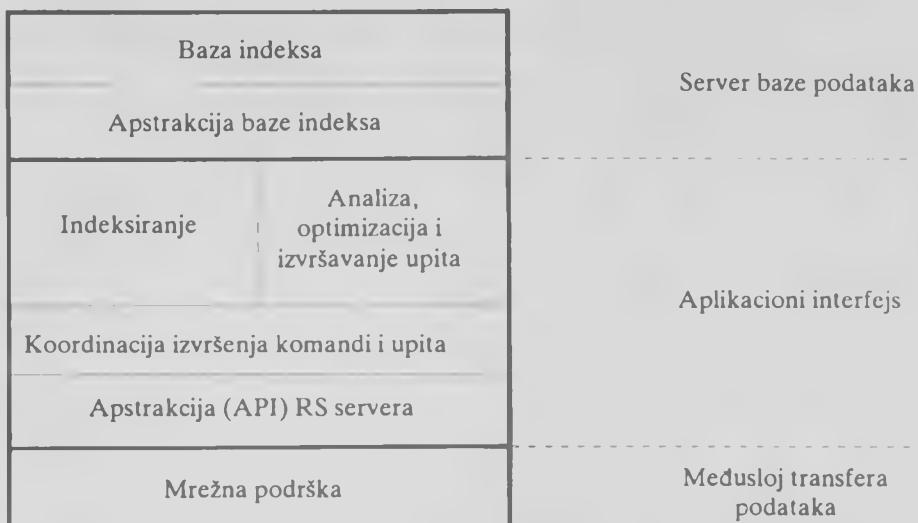
Što se tiče apstrakcije baza podataka najinteresantnija je apstrakcija baze indeksa, koja omogućava da server za indeksiranje i pretraživanje dokumenata bude nezavisan od sistema za upravljanje bazom podataka.

6.3. SERVER ZA INDEKSIRANJE I PRETRAŽIVANJE DOKUMENATA

6.3.1. Organizacija servera za indeksiranje i pretraživanje dokumenata

Organizacija servera za indeksiranje i pretraživanje dokumenata [9,10] prikazana je na slici 6.4.

Server za indeksiranje i pretraživanje dokumenata prihvata komande i upite preko podsistema mrežne podrške. Putem ovog podsistema se klijentima prosleđuju i statusi izvršenja komandi, kao i rezultati upita. Pristup serveru se obavlja preko aplikacionog programskog interfejsa rs. Ovaj API specificira funkcije kojima se inicira proces indeksiranja dokumenata i kojima se postavljaju upiti serveru. Podsistemi koji se odnose na indeksiranje su nezavisni od podsistema preko kojih se izvršavaju upiti. Ove komponente su povezane s jedne strane podistemom za koordinaciju izvršavanja komandi i upita, a sa druge strane bazom indeksa.



Slika 6.4. Organizacija servera za indeksiranje i pretraživanje dokumenata

Implementirana su dva tipa servera za pretraživanje i indeksiranje dokumenata, iterativni i konkurentni.

Struktura iterativnog servera je jednostavna, tako da je realizacija ista na svim operativnim sistemima na kojima je sistem implementiran. Posle inicijalizacije, proces čeka konekciju na zadatom portu, a primljeni zahtev prosledjuje modulu za parsiranje zahteva. Modul za parsiranje zahteva aktivira modul za izvršavanje upita koji posredstvom aplikacionog programskog interfejsa pristupa bazi indeksa i preko zadate funkcije vraca rezultate pretraživanja klijentu. Po izvršenom zahtevu server raskida konekciju. -

Nedostatak ovog servera je što ne omogućava istovremeni pristup bazi indeksa od strane više klijentata, tako da je praktično neupotrebljiv u periodima povećane eksploatacije sistema. Pored toga, iterativni server je osjetljiv na neregularnosti u komunikaciji sa klijentom, koje su rezultat sintaksnih ili drugih grešaka. Zahtev za sistemom koji bi bio u stanju da opsluži veći broj korisnika istovremeno, uz maksimalnu otpornost na greške, je uslovio realizaciju konkurentnih servera.

Kod konkurentnog servera proces posle inicijalizacije čeka na zadatom portu, i po priјemu zahteva kreira novi proces koji preuzima dalju obradu zahteva, dok glavni proces očekuje novu konekciju. Za razliku od iterativnog

servera, realizacija konkurentnog servera se razlikuje u zavisnosti od operativnog sistema.

Konkurentni server se može realizovati na dva načina. Server po modelu konkurentnog, konekciono orijentisanog *standalone* servera, koji sam prihvata zahteve na zadatom portu, realizovan je na Unix operativnom sistemu. Drugi način implementacije servera se oslanja na korišćenje tzv. superservera. Superserver očekuje konekciju na različitim portovima. Po prijemu zahteva superserver aktivira potproces, utvrđuje o kom servisu se radi i pokreće aplikaciju vezanu za taj servis. Upotreba superservera omogućava da se za svaki servis proces pokreće tek onda kada je potreban i da se po obavljenom zadatku okonča. Ovaj način realizacije servera je primjenjen na VMS operativnom sistemu, koji koristi Multinet, kao i na Unix-u.

6.3.2. Podsistem za izvršavanje upita

6.3.2.1. Tipovi upita i komandni jezik

Podsistem za izvršavanje upita zahtevc dobija u obliku stringova, praktično u istom obliku u kome ih zadaju korisnici. Upit počinje ključnom reči koja zadaje njegov tip. Upiti se mogu podeliti u dve grupe. U prvu spadaju tipovi upita *select*, *selects* i *selectc*, a u drugu *expand* i *expandd*.

Select upit se sastoji od ključne reči *select* i od izraza koji se izvršava. Rezultat upita je skup dokumenata koji zadovoljavaju zadati izraz. Izraz se sastoji od termova¹, (reči i fraze), povezanih skupovnim ili blizinskim operatorima. Fraze se pišu između navodnika. Između navodnika se, takođe, pišu i ključne reči, kao što su *and*, *or* i *not*, kada se koriste kao termovi.

Najjednostavniji izraz u *select* upitu se sastoji samo od jednog terma. U tom slučaju upit ima sledeći oblik:

select term

¹ Termini *deskriptor* i *term* ovde imaju slična značenja. Termin deskriptor se koristi kada se govori o elementima izvornog dokumenta (rečima i frazama) po kojima se posmatrani dokument indeksira, te su prema tome uvek vezani za neko konkretno pojavljivanje ovog elementa u dokumentu. Termin term se koristi za elementarne delove upita (takođe reči i fraze) po kojima se vrši pretraživanje. Pri ovome termini mogu da sadrže džoker znake "?" i "?", te im, u principu, odgovara neki skup deskriptora.

Semantika ovog upita je jednostavna. Rezultat upita je skup dokumenata u kojima se pojavljuje dati term. Ukoliko se termovi povežu logičkim operatorima (and, or i not), rezultat upita je skup dokumenata koji se dobija kada se odgovarajuće skupovne operacije (presek, unija i skupovna razlika) primene na rezultate *select* upita čiji su izrazi dati termovi. Pošto se ovde radi o operacijama nad skupovima dokumenata, ravnopravno sa izrazom logički operatori, može se koristiti i izraz skupovni operatori.

Značenje blizinskih operatora je nešto drugačije. Izraz u kome se pojavljuju dva terma povezana blizinskim operatorom (F) znači da se zahteva da se u dokumentu ta dva terma pojavljuju u istom polju. Slično, operatorom (S) se zahteva da se termovi nalaze u istoj rečenici, a operatorom (*n* W) da termovi ne budu udaljeni više od *n* reči jedan od drugog.

Standardni redosled izvršavanja operatora koji je definisan njihovim prioritetom ((*n* W), (S), (F), and, or i not) može se promeniti upotrebom zagrada na uobičajen način.

Termovi koji se javljaju u izrazima mogu biti kvalifikovani i nekvalifikovani. Za term se kaže da je kvalifikovan kada se uz njega navodi i oznaka polja u kome se zahteva da se ovaj term pojavljuje. Upit:

`select "informacioni sistemi"/T1`

daje skup svih dokumenata u kojima se fraza "*informacioni sistemi*" javlja u polju *T1* (uobičajena oznaka za polje naslova u bibliotečkim bazama podataka). Nekvalifikovani term u izrazu znači da se zahteva njegovo pojavljivanje u nekom od polja koja pripadaju osnovnom indeksu. Koja polja pripadaju osnovnom, a koja dodatnom indeksu, definiše se prilikom inicijalizacije baze indeksa. Ovde je bitno primetiti da se značenje upita neće promeniti kada se neki nekvalifikovani term u izrazu zameni nizom kvalifikovanih termova povezanih operatorom or pri čemu se kvalifikacija vrši po svim poljima iz baznog indeksa. Razlika u sintaksi pri kvalifikaciji termova poljima iz osnovnog, odnosno dodatnog indeksa, koja je nasleđena iz sistema DIALOG, pri izvršavanju upita ne igra nikakvu bitnu ulogu.

Upiti *selects* i *selectc* se sastoje od ključne reči *selects*, odnosno *selectc* i izraza koji se formira na isti način kao kod naredbe *select*. Razlika nastaje u izvršavanju.

Drugi tip upita je *expand*. On je znatno jednostavniji od *select*-a i ima oblik:
expand term

pri čemu se u *term*-u mogu pojavljivati džoker zanci "?" i *. Rezultat upita je skup svih termova koji se pojavljuju u dokumentima, a koji odgovaraju zadatom termu. Upotreba termova sa džoker znacima nije samo ograničena na *expand* upite. Džoker znaci mogu da se koriste i u termovima u *select* upitu. Slično kao i kod upotrebe nekvalifikovanih termova, neekspandovan term je moguće zameniti nizom termova dobijenih njegovom ekspanzijom i povezanih operatorom or.

Upit *expandd* je sličan prethodnom, s tim što je *term* potrebno kvalifikovati po željenom polju. Kao rezultat se dobija spisak termova koji odgovaraju zadatom obrascu, a javljaju se u specificiranom polju.

U nastavku poglavљa će biti opisan način na koji je implementirano izvršavanje upita tipa *select*. Implementacija upita tipova *expand* i *expandd* se bazira na jednostavnom pozivu odgovarajućih funkcija baze indeksa, a upiti tipova *selectc* i *selects* se implementiraju slično upitu tipa *select*.

6.3.2.2. Koraci u izvršavanju upita

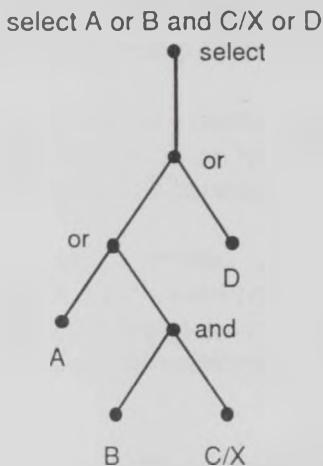
U podsistemu za izvršavanje upita koriste se metode koje su slične tehnikama interpretacije upita u savremenim relacionim bazama podataka [11]. Koraci u izvršavanju upita su uobičajeni:

1. Leksička analiza upita. U ovom koraku se određuje tip upita i izraz se razbija na sastavne termove, kvalifikatore i operatore.
2. Sintaksna analiza upita. Rezultat ove faze je formirano sintaksno stablo izraza koji se evaluira. S obzirom na to da se struktura sintaksnog stabla poklapa sa strukturom plana izvršenja upita pre izvedenih transformacija, u ovoj fazi se praktično formira ovaj plan.
3. Semantička analiza upita. Semantička analiza se sastoji od provere ispravnosti upotrebe: a) oznaka polja u kvalifikatorima termova; b) redosleda izvršavanja blizinskih operatora.
4. Transformacije plana izvršavanja upita. Plan se, iz početnog oblika, transformiše u formu u kojoj je moguća njegova interpretacija.
5. Interpretacija plana izvršavanja upita.

Alternativa interpretaciji plana (korak 5) je njegova kompilacija u jedinstven iterativni program koji se bazira na upotrebi ugnježdenih petlji i drugih kontrolnih struktura.

6.3.2.3. Sintaksna analiza upita

Rezultat sintaksne analize je formirano sintaksno stablo. Na slici 6.5 je dat primer jednog upita i sintaksnog stabla koje mu odgovara.



Slika 6.5. Primer sintaksnog stabla za dati upit

Iz navedenog primera se vidi da termovima u izrazu odgovaraju listovi stabla, a operatorima ostali čvorovi. U listovima se čuva informacija o tome da li je i kako posmatrani čvor kvalifikovan. Koren stabla reprezentuje tip upita.

6.3.2.4. Semantička analiza upita

Provera ispravnosti upotrebe oznaka polja u kvalifikatorima se obavlja paralelno sa sintaksnom analizom, znači pre kreiranja celog sintaksnog stabla. Proverava se da li polje datog imena postoji u tekstuualnoj bazi podataka i da li je upotrebljena ispravna sintaksna forma kvalifikatora.

Provera redosleda upotrebe blizinskih operatora se obavlja na formiranom sintaksnom stablu. Sistem zabranjuje primenu strožijih uslova na rezultatima dobijenim upotrebom opštijih uslova. Na primer, sledeći upit se proglašava semantički neispravnim:

select (a (F) b) (S) c

S druge strane nije moguća automatska transformacija ovog upita u:

select (a (S) c) (F) (b (S) c)

jer se semantika ova dva upita očigledno razlikuje.

Provera redosleda upotrebe blizinskih operatora se obavlja tako što se svakom čvoru koji predstavlja operator dodeljuje jedan atribut, kategorija čvora, koji ima sledeće vrednosti:

Operator	Kategorija čvora
and, or, not	1
(F)	2
(S)	3
(n W)	4

Listovima stabla se dodeljuje ista kategorija koju ima i roditeljski čvor. Ukoliko se stablo sastoji samo od jednog čvora, njegova kategorija je 1. U stablu se ne dozvoljava da potomci imaju kategoriju koja je manja od kategorije posmatranog čvora.

6.3.2.5. Plan izvršavanja upita

Plan izvršavanja upita, pre transformacija, ima istu strukturu kao i sintaksno stablo upita. Teoretski bi bila moguća interpretacija i ovakvog plana. Ako, za sada, prepostavimo da se u upitu koriste samo logički operatori, algoritam interpretacije netransformisanog plana izvršavanja upita bio bi sledeći:

- Stablo se obilazi od dna ka vrhu.
- Svakom od listova stabla pridruži se skup dokumenata u kojima se odgovarajući term nalazi. Pri ovome se vodi računa o tome da li je term kvalifikovan ili nije. Za kvalifikovane termove se postavlja dodatni uslov da se term pojavljuje u polju koje ima u upitu navedeni naziv. U slučaju nekvalifikovanih termova, uslov je da se term pojavljuje u polju koje pripada osnovnom indeksu.
- Čvorovima koji nisu listovi pridruži se skup dobijen primenom odgovarajuće skupovne operacije nad skupovima koji su pridruženi njegovoj deci.
- Rezultat upita, po obilasku svih čvorova stabla, je skup pridružen korenu.

Ukoliko se dopusti upotreba blizinskih operatora, algoritam se modifikuje utoliko što se više ne radi o skupovima dokumenata, već o skupovima pozicija termova. Pozicija terma je uređena četvorka:

$$(p_doc, p_field, p_sent, p_word)$$

Elementi pozicije su redom: identifikator dokumenta, redni broj polja u dokumentu, redni broj rečenice u polju, i redni broj reci u polju. U ovom slučaju, logički operatori se interpretiraju tako što ignorišu sve elemente pozicije osim prvog (p_doc). Cvorovima koji predstavljaju blizinske operatore pridružuje se skup pozicija koje zadovoljavaju operatorom postavljeni uslov. S obzirom na to da su skupovi pozicija, u stvari relacije, vidi se da se izvršavanje blizinskih operatora svodi na određivanje spoja relacija sa sledećim uslovima:

Operator	Uslov
(F)	$p_doc_1 = p_doc_2 \wedge p_field_1 = p_field_2$
(S)	$p_doc_1 = p_doc_2 \wedge p_field_1 = p_field_2 \wedge p_sent_1 = p_sent_2$
(n W)	$p_doc_1 = p_doc_2 \wedge p_field_1 = p_field_2 \wedge p_word_1 - p_word_2 \leq n$

Očigledno je da bi izloženi algoritam bio veoma neefikasan. Dva su osnovna razloga za ovu neefikasnost:

1. Svi skupovi (relacije) bi trebalo da se čuvaju u privremenim fajlovima.
2. Da bi se efikasno realizovale, i skupovne operacije i relacioni spoj zahtevaju da se elementima relacija nad kojima se obavljaju, može asocijativno pristupiti (*hash join* algoritam), ili da elementi budu sortirani u pogodnom redosledu (*merge join* algoritam).

Merge join algoritam se može realizovati tako da se izbegne potreba za privremenim smještanjem međurezultata na disk. Osnovna ideja ovog metoda je da se podaci koji su rezultat neke operacije direktno prosleđuju sledećoj operaciji, i to u takvoj formi da se i ta sledeća operacija može efikasno realizovati. Lako se može videti da je za sve operacije (i skupovne i blizinske) pogodno da podaci budu sortirani po sledećim ključevima: p_doc , p_field , p_sent i p_word . Veoma je značajno to što operatori ne menjaju redosled podataka.

Da bi prikazani model izvršavanja upita funkcionisao, potrebno je da listovi plana izvršavanja takođe daju sortirane podatke.

6.3.2.6. Transformacija plana izvršavanja upita

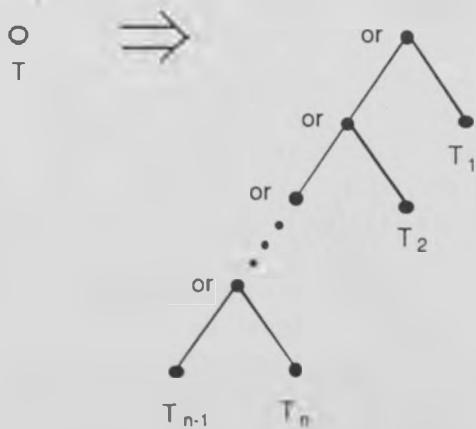
Zahtev da listovi plana izvršavanja, koji zapravo odgovaraju termovima upita, daju sortirane podatke, se svodi na to da to radi baza podataka indeksa. Ovde se javljaju dva problema koji bi mogli znatno da uslože realizaciju sistema za upravljanje ovom bazom:

1. Sortiranje termova u kojima se pojavljuju džoker znaci.
2. Sortiranje nekvalifikovanih termova, jer je prirodan način da se ulančavanje termova obavi po poljima.

S obzirom da se u oba slučaja radi o sortiranju u kome je prirodno da se primeni *merge-sort* algoritam, logično je da se ove obrade obavljaju u modulu za izvršavanje upita. Zbog toga je potrebno da se na planu izvršavanja upita primene sledeće transformacije:

- a) ekspanzija neekspandovanih termova i
- b) ekspanzija nekvalifikovanih termova.

Obe ove transformacije se sastoje od zamene nekih listova plana izvršavanja upita stablima kojima se realizuje or operacija između termova dobijenih ekspanzijom, odnosno kvalifikacijom datog terma, kao što je prikazano na slici 6.6.



Slika 6.6. Ekspanzija termova u planu izvršavanja upita

Redosled izvršavanja transformacija nije od prevelikog značaja za izvršavanje upita, ali može veoma mnogo uticati na ukupno vreme izvršavanja transformacija. S obzirom da se informacije o poljima u

inicijalizaciji sistema učitavaju iz baze podataka i stalno čuvaju u memoriji, a da je ekspanzija terma relativno dugotrajan proces, mnogo je bolje da se prvo obavi ekspanzija, a zatim kvalifikacija termova.

6.3.2.7. Interpretacija plana izvršavanja upita

Sinhronizacija i transfer podataka su osnovna pitanja koja se tiču sistema za interpretaciju upita. Najjednostavniji metod je čuvanje međurezultata u privremenim fajlovima, koji bi u ovom slučaju bili sortirani. Drugi način je da se kreira jedan proces za svaki operator i da se koristi neki od metoda interprocesne komunikacije koji obezbeđuje operativni sistem. Oba ova metoda ne mogu da daju prihvatljive performanse interpretacije upita. Jedino prihvatljivo rešenje je da se problem sinhronizacije i transfera podataka reši u okviru jednog procesa i to bez smeštanja međurezultata na disk. Ovo se ostvaruje upotrebom posebnih programskih struktura - iteratora.

Iterator se sastoji od pointera na tri procedure *open*, *next* i *close*, i eventualno od nekih lokalnih podataka koji su potreбni za izvršavanje pomenutih procedura.

Princip interpretacije plana izvršenja upita je sledeći:

- Svakom čvoru u planu pridružuje se jedan iterator.
- Procedure iteratora se pozivaju isključivo iz iteratorskih procedura roditeljskog čvora. Ovo ne važi jedino za koren plana izvršavanja upita, čije se procedure pozivaju iz drugih delova sistema.
- Iteratorska procedura *open* inicijalizuje lokalne podatke iteratora i poziva proceduru *open* iteratora koji su pridruženi deci posmatranog čvora. Slično, procedura *close* oslobađa resurse koje je iterator eventualno alocirao i zatim poziva proceduru *close* dece datog čvora.
- Obradu obavlja procedura *next*, koja u svakom svom pozivu generiše jedan podatak - uređenu četvorku (*p_doc*, *p_field*, *p_sent*, *p_word*), ili ako to nije moguće, signal da nema više podataka. Kada su potrebni podaci, pozivaju se procedure *next* dece posmatranog čvora.
- Iteratori koji su pridruženi listovima plana (termovima) imaju kao iteratorske procedure funkcije modula baze podataka indeksa: *rsbOpenTerm*, *rsbNextTerm* i *rsbCloseTerm*.

Algoritam interpretacije upita je sada trivijalan: a) pozove se procedura *open* iteratora koji je pridružen korenu stabla; b) u petlji se poziva procedura *next* istog iteratora, sve dok se ne dobije signal da nema više podataka; c) pozove se procedura *close*.

Posmatrani mehanizam interpretacije upita omogućava da se jednostavno implementira i jedan tip optimizacije upita koji se bazira na kategoriji čvorova određenoj u procesu semantičke analize upita. Kategorija čvora se predaje kao jedan od parametara iteratorskoj proceduri *open* tako da je u iteratoru moguće filtrirati podatke, i poziciju terma davati sa preciznošću koja je potrebna za pravilan rad roditeljskog iteratora. Na ovaj način se bitno smanjuje količina podataka koja se obrađuje u iteratorima, pogotovo na čvorovima koji su bliži korenu stabla. U modulu za upravljanje bazom podataka indeksa koristi se naziv *dubina pretraživanja* sa istim značenjem koje ovde ima kategorija čvora.

6.3.3. Podsistem za indeksiranje dokumenata

Podsistem za indeksiranje dokumenata zahteve dobija u obliku stringova, u obliku u kome ih zadaju korisnici. Zahtevi počinju ključnom reči koja određuje njihov tip. Zahtevi se mogu podeliti u dve grupe: zahteve za indeksiranje dokumenata i zahteve za brisanje podataka iz baze indeksa.

Zahtev se sastoji od ključne reči (*index*, *reindex*, *delete* i *clear*) i stringa koji predstavlja ime dokumenta.

Standalone konkurentni server za indeksiranje i pretraživanje dokumenata prihvata zahtev, identificiše ga i smešta u red za čekanje. Server detektuje trenutak kada ne pristižu zahtevi za pretraživanje i startuje indeksiranje kroz potproces za indeksiranje koji je stalno aktivran. U periodu dok traje indeksiranje, odnosno brisanje, ne izvršavaju se novi zahtevi za pretraživanje ili indeksiranje. Indeksiranje i brisanje dokumenata se obavlja iterativno. Konkurentno izvršavanje ovih zahteva bi podrazumevalo korišćenje mehanizama zabrane pristupa podacima na nivou sistema za upravljanje bazama podataka, što bi uslovilo znatan gubitak na performansama sistema. Indeksiranje i brisanje podataka je preporučljivo vršiti u periodima smanjenog opterećenja sistema.

Indeksiranje kod konkurentnog servera realizovanog preko superservera se izvršava na nešto drugačiji način. Po prijemu zahteva za indeksiranje, odnosno brisanje, server ga smešta u fajl za indeksiranje. U periodima rada sistema kada se ne vrši pretraživanje, administrator sistema pokreće poseban program za indeksiranje koji šalje signal superserveru za obustavljanje prijema zahteva i učitavajući redom zahteve iz fajla vrši indeksiranje. Po završenom indeksiranju server nastavlja sa normalnim prijemom zahteva.

6.3.4. Baza indeksa

Osnovna funkcija podsistema baze indeksa je da podsistemu za izvršavanje upita obezbedi informacije o poziciji termova. Podsetimo se da je pozicija terma, u opštem slučaju uređena četvorka:

$$(p_doc, p_field, p_sent, p_word)$$

čiji su elementi redom: identifikator dokumenta, redni broj polja u dokumentu, redni broj rečenice u polju, i redni broj reči u rečenici. Podsistem za izvršavanje upita zahteva pri tome da podsistem baze indeksa ispuni i sledeće uslove koji se tiču pozicije termova:

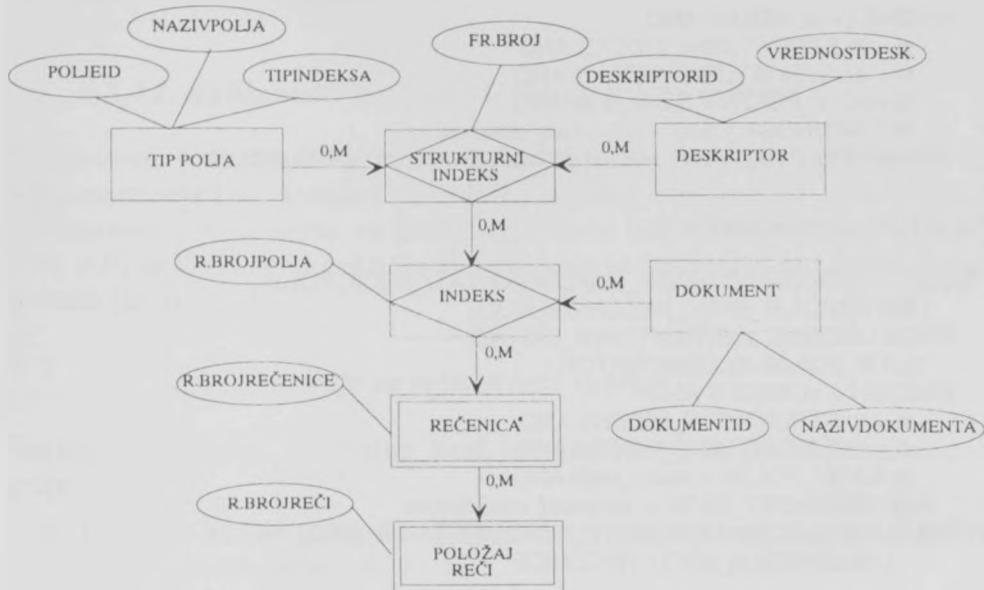
- Pozicije se čitaju iz iteratorskog procesa koji odgovara listu plana izvršenja. U trenutku izvršavanja upita svi termovi su ekspandovani i kvalifikovani, te je pogodno da interfejs prema bazi indeksa obezbedi iteratorski proces koji bi za zadati par term - naziv polja davao jednu po jednu poziciju terma, pri čemu se zahteva da se term pojavljuje u specificiranom polju.
- S obzirom da se interpretacija plana izvršavanja upita zasniva na *merge-join* algoritmu zahteva se da pozicije termova budu sortirane².
- Podsistem za izvršavanje upita može zahtevati da mu se daju nepotpune pozicije termova. Na primer, ako je kategorija čvora koji odgovara posmatranom termu 2, zahteva se da se da samo po jedna pozicija terma po polju dokumenta, pri čemu će se podaci o poziciji rečenice i reči ignorisati, to jest podsistem baze indeksa može da u poslednja dva člana uređene četvorke koja čini poziciju terma, stavi proizvoljne vrednosti.

Pored pozicije termova podsistem baze indeksa treba da podsistemu za izvršavanje upita pruži informacije o tipovima polja (koriste se u semantičkoj analizi i pri ekspanziji nekvalifikovanih termova) i deskriptorima koji odgovaraju zadatom obrascu (ekspanzija neekspandovanih termova).

Opšti model podataka baze indeksa je dat u Poglavlju 5. Na slici 6.7 je prikazan deo ovog modela koji je neophodan za implementaciju jedinstvenog sistema za indeksiranje i pretraživanje dokumenata.

² Podrazumeva se leksikografsko uređenje pozicija termova.

EER DIAGRAM: Baza indeksa



Slika 6.7. Model podataka baze indeksa

Može se pokazati da prikazani model omogućava efikasnu realizaciju zahteva podistema za izvršavanje upita. Najsloženiji upiti se tiču pozicije termova. Koristeći upitni jezik definisan metodologijom (poglavlje 4.), nizovi pozicija koje treba da vraca odgovarajući iterativni proces mogu se, za različite *dubine pretraživanja* (kategorije čvora koji koristi dati iteratorski proces) i za dati naziv polja (*naziv_polja*) i deskriptor (*vrednost deskriptora*), definisati kao:

Za dubinu pretraživanja 1:

```

SELECT DISTINCT si.Indeks.Dokument.Naziv_Dokumenta, NULL, NULL, NULL
FROM si:STRUKTURNI_INDEKS, tp:TIP_POLJA, ds:DESKRIPTOR
WHERE si = tp:STRUKTURNI_INDEKS AND
      si = ds:STRUKTURNI_INDEKS AND
      tp.Naziv_Polja = naziv_polja AND
      ds.Vrednost_Desk = vrednost_deskriptora
ORDER BY si.Indeks.Dokument.Naziv_Dokumenta
  
```

Za dubinu pretraživanja 2:

```
SELECT i.DOKUMENT.NAZIV_DOKUMENTA, i.R_BROJ_POLJA, NULL, NULL
  FROM i:INDEKS, si:STRUKTURNI_INDEKS,
       tp:TIP_POLJA, ds:DESKRIPTOR
 WHERE i = si.INDEKS AND
       si = tp.STRUKTURNI_INDEKS AND
       si = ds.STRUKTURNI_INDEKS AND
       tp.NAZIV_POLJA = naziv_polja AND
       ds.VREDNOST_DESK = vrednost_deskriptora
 ORDER BY i.DOKUMENT.NAZIV_DOKUMENTA, i.R_BROJ_POLJA
```

Za dubinu pretraživanja 3:

```
SELECT i.DOKUMENT.NAZIV_DOKUMENTA, i.R_BROJ_POLJA,
       i.RECENICA.R_BROJ_RECENICE, NULL
  FROM i:INDEKS, si:STRUKTURNI_INDEKS,
       tp:TIP_POLJA, ds:DESKRIPTOR
 WHERE i = si.INDEKS AND
       si = tp.STRUKTURNI_INDEKS AND
       si = ds.STRUKTURNI_INDEKS AND
       tp.NAZIV_POLJA = naziv_polja AND
       ds.VREDNOST_DESK = vrednost_deskriptora
 ORDER BY i.DOKUMENT.NAZIV_DOKUMENTA, i.R_BROJ_POLJA,
          i.RECENICA.R_BROJ_RECENICE
```

Za dubinu pretraživanja 4:

```
SELECT i.DOKUMENT.NAZIV_DOKUMENTA, i.R_BROJ_POLJA,
       r.R_BROJ_RECENICE, r.POLOZAJ_REC.I.R_BROJ_REC
  FROM r:i.RECENICA, i:INDEKS, si:STRUKTURNI_INDEKS,
       tp:TIP_POLJA, ds:DESKRIPTOR
 WHERE r = i.RECENICA AND
       i = si.INDEKS AND
       si = tp.STRUKTURNI_INDEKS AND
       si = ds.STRUKTURNI_INDEKS AND
       tp.NAZIV_POLJA = naziv_polja AND
       ds.VREDNOST_DESK = vrednost_deskriptora
 ORDER BY i.DOKUMENT.NAZIV_DOKUMENTA, i.R_BROJ_POLJA,
          r.R_BROJ_RECENICE, r.POLOZAJ_REC.I.R_BROJ_REC
```

Upiti koji vraćaju informacije o poljima i deskriptorima koji odgovaraju zadatim obrascima (term sa džoker znacima) su trivijalni:

```
SELECT TIP_POLJA.NAZIV_POLJA, TIP_POLJA.TIP_INDEKSA
```

```
i
SELECT DESKRIPTOR.VREDNOST_DESK
 WHERE DESKRIPTOR.VREDNOST_DESK LIKE obrazac
```

Na sličan način, preko operacija na PMOV, mogu se definisati i operacije održavanja baze podataka. Ovde to nećemo uraditi, već ćemo kasnije, u opisu aplikacionog programskog interfejsa baze indeksa dati neformalan opis ovih operacija.

6.3.4.1. Aplikacioni programski interfejs baze indeksa

Aplikacioni programski interfejs baze indeksa [12], [13] skriva detalje implementacije baze indeksa od ostatka sistema, odnosno od podsistema za indeksiranje i podsistema za izvršavanje upita koji pristupaju bazi indeksa. Ovaj API definiše biblioteku funkcija za pristup bazi indeksa i način obrade grešaka (izuzetaka).

6.3.4.1.1. Funkcije za pristup bazi indeksa

Funkcije biblioteke za pristup bazi indeksa mogu se podeliti na sledeće grupe:

1. Funkcije za inicijalizaciju biblioteke za pristup bazi indeksa i prekid rada sa ovom bibliotekom.
2. Funkcije za održavanje baze indeksa.
3. Funkcije za pretraživanje baze indeksa.

Pre prvog poziva neke druge funkcije za pristup bazi, zahteva se da se pozove funkcija za inicijalizaciju biblioteke *rsbOpen*. Takođe se zahteva da se na kraju rada sa bibliotekom pozove *rsbClose*. API ne definiše semantiku ovih funkcija. One služe da se olakša implementacija biblioteke i najčešće se koriste za dinamičku alokaciju i oslobođanje nekih resursa potrebnih za rad podsistema baze indeksa. Funkcija *rsbOpen* ima argument koji definise da li će se u dатој sesiji baza indeksa koristiti samo za čitanje, ili će se koristiti i za pisanje.

Funkcije za održavanje baze indeksa se mogu podeliti u dve podgrupe:

- a) Funkcije koje se koriste u procesu definisanja baze dokumenata.
- b) Funkcije koje se koriste u procesu indeksiranja.

U procesu definisanja baze dokumenata u bazi indeksa se definišu tipovi polja, i to tako što se za svaki tip polja definiše njegov naziv, tip indeksa (bazni, dodatni ili oba) i tip pretraživosti polja (nepretraživo polje, polje se indeksira po rečima, polje se indeksira po frazama, polje se indeksira i po

rečima i po frazama)³. Za dodavanje tipova polja u bazu indeksa koristi se funkcija *rsbAddField*. Ova funkcija proverava da li u bazi indeksa postoji objekat klase *TIP_POLJA* specificiranog imena i ako ovakav objekat ne postoji, kreira ga i postavlja vrednosti atributa ovog objekta. Normalno se svi tipovi polja definišu pre nego što se počne sa indeksiranjem dokumenata. Međutim, ne postoji nikakva prepreka da se neki tipovi polja dodaju kasnije, kad su neki dokumenti već indeksirani.

Za indeksiranje dokumenta koriste se funkcije *rsbAddDoc* i *rsbAddTermPos*. Funkcijom *rsbAddDoc* se u bazu indeksa upisuju podaci o dokumentu koji se indeksira. Ova funkcija kreira novi objekat klase *DOKUMENT* pod uslovom da u bazi indeksa već ne postoji dokument istog imena. Funkcija *rsbAddTermPos* upisuje u bazu indeksa poziciju terma. Pri ovome se kreiraju objekti klase *DESKRIPTOR*, *STRUKTURNI_INDEKS*, *INDEKS*, *RECENICA* i *POLOZAJ_REC* i odgovarajuće veze između ovih entiteta. Svi navedeni objekti se kreiraju samo u slučaju kada se dati term nije pojavljivao u dokumentima koji su do posmatranog poziva funkcije indeksirani, a u protivnom se kreiraju samo neophodni objekti. Jednom indeksiran dokument može se izbrisati iz baze indeksa pomoću funkcija *rsbDelDoc* i *rsbClrDoc*. Funkcija *rsbDelDoc* briše dati dokument u potpunosti iz baze sa svim pripadajućim deskriptorima i njihovim pozicijama, dok *rsbClrDoc* briše samo deskriptore sa pozicija koji su povezani sa datim dokumentom. Pri ovome se naravno brišu samo oni objekti koji nisu povezani sa drugim dokumentima u bazi.

Funkcije za pretrživanje baze indeksa su definisane tako da obrazuju tri iteratora:

1. Iterator za čitanje podataka o poziciji termova u zadatom tipu polja, funkcije *rsbOpenTermPos*, *rsbNextTermPos* i *rsbCloseTermPos*.
2. Iterator za čitanje informacija o poljima, funkcije *rsbOpenFieldInfo*, *rsbNextFieldInfo* i *rsbCloseFieldInfo*.
3. Iterator za ekspanziju neekspandovanih termova, funkcije *rsbOpenWild*, *rsbNextWild* i *rsbCloseWild*.

Semantika upita koji se izvršavaju ovim iteratorima je data ranije. Način korišćenja iteratora je jednostavan. Funkcijom *rsbOpen...* se inicijalizuje iterator i zadaju parametri upita. Sukcesivnim pozivom funkcija *rsbNext...* dobijaju se jedna po jedna pozicija terma, skup informacija o polju ili ekspandovani term, u zavisnosti od toga o kom se iteratoru radi. Iteratorski

³ Tip pretraživosti polja nije od interesa za funkcionisanje jedinstvenog sistema za indeksiranje i pretraživanje dokumenta, te zbog toga nije ni prikazan na modelu podataka baze indeksa (slika 6.7), niti je razmatran u operacijama pretraživanja baze.

proces se završava pozivom funkcije *rsbClose...* čija je prvenstvena uloga u tome da osloboди resurse koji su alocirani za izvršavanje upita.

6.3.4.1.2. Obrada grešaka

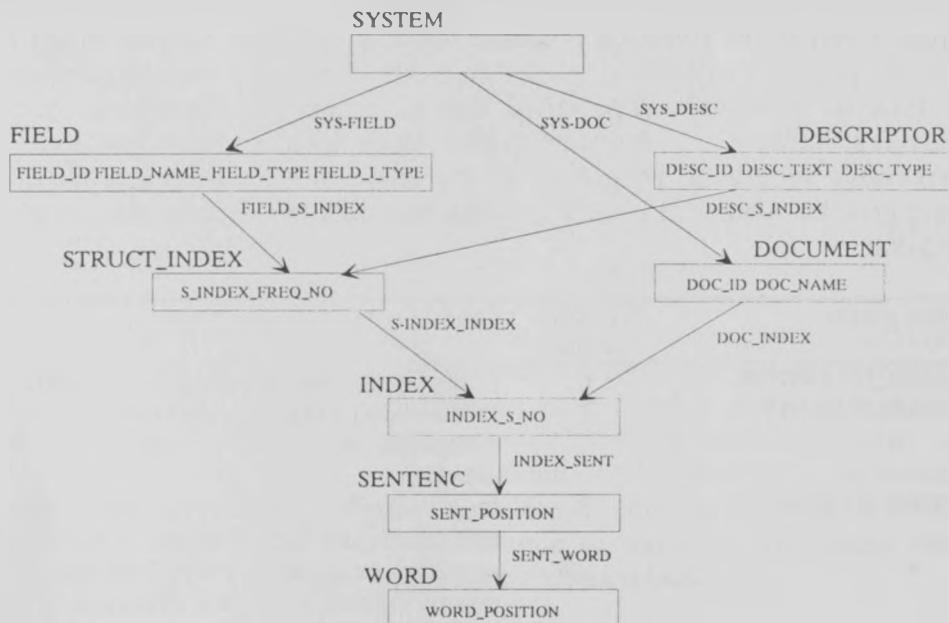
Izvršenje funkcija za pristup bazi indeksa može da bude uspešno, ili se pri izvršenju ovih funkcija može javiti greška. Moduli koji koriste ove funkcije mogu dobiti status izvršenja pozivom funkcije *rseError*. Greške mogu biti različite prirode i najčešće su posledica nekih događaja u okruženju u kome se funkcije izvršavaju (operativni sistem, sistem za upravljanje bazom podataka, hardver, ...). Pojedine greške mogu imati različite posledice na izvršavanje programa, zbog čega su statusi izvršenja funkcija grupisani u klase grešaka. Sistem za obradu grešaka baze indeksa definiše sledeće klase grešaka:

Klasa greške	Brojčana vrednost	Opis
RSERR_NOERROR	0	Pri izvršenju funkcije se nije javila greška.
RSERR_WARNING	1	Pri izvršenju funkcije, detektovana je neka neregularnost, ali je funkcija izvršena uspešno.
RSERR_ERROR	2	Iz određenih razloga izvršavanje funkcije nije uspelo, ali funkcionisanje sistema nije ugroženo, te ponovni poziv funkcije može uspeti. Tipično ovoj klasi pripadaju greške koje se javljaju usled nedostatka nekih računarskih resursa.
RSERR_FATAL	3	Pri izvršavanju funkcije došlo je do greške posle koje nije moguće dalje izvršavanje programa.
RSERR_PROGRAMMING	4	Detektovana je programska greška u modulu koji koristi procedure za pristup bazi indeksa.
RSERR_INTERNAL	5	Detektovana je programska greška u funkcijama za pristup bazi indeksa.

Klasa grešaka statusa izvršenja funkcije se dobija pozivom funkcije *rseErrorClass*. Sistem za pristup bazi podataka, kada se detektuje greška čija je klasa greške RSERR_FATAL ili veća, prekida izvršenje procesa. Ovakvo ponašanje sistema se može izmeniti postavljanjem neke druge klase grešaka za koju će se prekidati proces. Ovo se radi pozivom funkcije *rseExceptionSeverity*. Takođe je umesto jednostavnog prekida izvršenja procesa moguće, pozivom funkcije *rseExceptionProcedure*, naložiti sistemu za obradu grešaka da izvrši neku drugu akciju.

6.3.4.2. Implementacija baze indeksa

U jedinstvenom sistemu za indeksiranje i pretraživanje dokumenata, baza indeksa je realizovana pomoću mrežnog sistema za upravljanje bazom podataka db_VISTA. Mrežni model podataka je dobijen direktnim prevođenjem modela objekti veze (slika 6.7) i prikazan je na slici 6.8.



Slika 6.8. Mrežni model podataka baze indeksa

Ovde neće biti razmatrani detalji implementacije baze indeksa već će biti date samo neke opšte primedbe u vezi ove implementacije.

Zahtev da pozicije termova, koji se dobijaju kao rezultat odgovarajućeg upita, budu sortirane može se realizovati sekvencijalnim pristupom članovima odgovarajućih skupova pod uslovom da su ovi skupovi sortirani. Analizom upita i mrežnog modela može se zaključiti da je potrebno da budu sortirani sledeći skupovi: SYS_DESC, DESC_S_INDEX, S_INDEX_INDEX, DOC_INDEX, INDEX_SENT, SENT_WORD.

Ukoliko se postavi dodatni zahtev podsistemu za indeksiranje da deskriptore u dokumentu indeksira u redosledu u kome se pojavljuju u pojedinim poljima dokumenta, a polja jednog tipa u redosledu kako se javljaju u dokumentu, članovi skupova S_INDEX_INDEX, INDEX_SENT i

SENT_WORD će biti sortirani ako se povezivanje zapisa INDEX, SENTENCE i WORD vrši dodavanjem na kraj lista koje odgovaraju ovim skupovima.

Poljima DESC_TEXT (odgovara atributu VREDNOST_DESKRIPTORA u modelu podataka) i FIELD_NAME (odgovara atributu NAZIV_POLJA) se pristupa po sadržaju, tako da su ova polja indeksirana.

6.3.5. Mrežna podrška

6.3.5.1. Aplikacioni programski interfejs za pretraživanje i indeksiranje baze indeksa

Aplikacioni programski interfejs za pretraživanje i indeksiranje baze indeksa se sastoji od biblioteke funkcija koje treba da omoguće integraciju ovog sistema u druge programske sisteme. Pored svoje osnovne funkcije, ova biblioteka treba da sakrije detalje implementacije komunikacionih mehanizama između programskog sistema koji koristi usluge sistema za pretraživanje i servera koji realizuje ove usluge.

Biblioteka omogućava izvršavanje elementarnih funkcija sistema za pretraživanje dokumenata:

1. Izvršavanje naredbe *select*, jezika za pretraživanje pri čemu se kao rezultat dobija spisak dokumenata koji zadovoljavaju kriterijum pretraživanja.
2. Izvršavanje naredbe *selectc* jezika za pretraživanje pri čemu se kao rezultat dobija spisak dokumenata, u zadatom opsegu, koji zadovoljavaju kriterijum pretraživanja.
3. Izvršavanje naredbe *selects* jezika za pretraživanje pri čemu se kao rezultat dobija izvestaj o broju pogodaka po podizrazima.
4. Izvršavanje naredbe *expand*, jezika za pretraživanje, pri čemu se zadaje term (reč ili fraza) sa džoker znacima, a kao rezultat dobija spisak deskriptora koji se slažu sa zadatim termom, a pri tome se pojavljuju u indeksiranim dokumentima.
5. Izvršavanje naredbe *expandd*, jezika za pretraživanje, pri čemu se zadaje term (reč ili fraza) i polje po kom treba pretraživati, a kao rezultat se dobijaju deskriptori koji se nalaze u zadatom polju, uz broj različitih dokumenata u kojima se mogu naći.
6. Indeksiranje dokumenta.
7. Reindeksiranje dokumenta.
8. Brisanje informacija o dokumentima.

Aplikacioni programski interfejs ne propisuje način na koji će server izvršavati zahtevane usluge, i to, kako u pogledu interpretacije postavljenih upita, tako i u pogledu strategije izvršavanja zahteva.

Korišćenje ove biblioteke je jednostavno. U programu se definiše jedna ili više funkcija koje služe za obradu dobijenih rezultata. Ove funkcije se navode kao argumenti u pozivima procedura kojima se izvršavaju upiti: *rsSelect*, *rsSelects*, *rsSelectc*, *rsExpand* i *rsExpandid*. Za svaki element rezultata (dokument, odnosno deskriptor) poziva se navedena funkcija.

Kada se u bazu podataka izvornih dokumenata unese novi dokument, pozivom funkcije *rsIndex*, potrebno je ovaj dokument indeksirati. Ako je dokument koji se indeksira, ranije već bio indeksiran, koristi se funkcija *rsReindex*.

Ukoliko se želi izbrisati neki dokument iz baze u potpunosti, koristi se funkcija *rsDelete*, a ukoliko se žele obrisati samo podaci o dokumentu, koristi se funkcija *rsClear*.

Pre svih ostalih poziva funkcija biblioteke, pozivom *rsInit*, biblioteka se inicijalizuje, a sesija završava pozivom funkcije *rsTerm*.

6.3.5.2. Protokol za komunikaciju sa serverom za indeksiranje i pretraživanje dokumenata

Koristeći OSI model za međukomputersku komunikaciju, za potrebe sistema za pretraživanje i indeksiranje tekstualnih baza podataka, definisana je komunikaciona biblioteka, preko koje se vrši komunikacija sa serverima. Ova biblioteka obezbeđuje međuprocesnu komunikaciju putem različitih komunikacionih mehanizama, kao što su: socket-i, mail box-ovi i sl. Za kodiranje i dekodiranje podataka upotrebljena je XDR specifikacija[14].

Zahtevi koji se mogu uputiti serveru za pretraživanje i indeksiranje tekstualnih baza podataka su: zahtev za indeksiranje dokumenta, zahtev za pronalaženje reči i fraza koje odgovaraju datom obrascu, zahtev za pružanje informacija o poljima, zahtev za pronalaženje imena dokumenta koji zadovoljava uslov pretraživanja i sl. Radi ilustracije komunikacije sa serverom za pretraživanje baze indeksa, neki od zahteva će biti detaljnije opisani. Pri tome, treba naglasiti da se za svaki zahtev serveru za pretraživanje baze indeksa otvara nova konekcija sa serverom.

(1) *ExpandRequest* je zahtev serveru za pronalaženjem u bazi indeksa svih deskriptora koji odgovaraju zadatom obrascu. Opis komunikacije pri posleđivanju zahteva je sledeći:

- klijent: otvara konekciju sa serverom
server: prihvata konekciju, ali ne vraća informaciju o statusu
klijent: šalje zahtev - u obliku *Request1*
server: vraća rezultat operacije kao niz poruka oblika *Result1* i zatvara konekciju

Slede opisi zahteva i odgovora:

```
const MAXLEN = 64
/* Request1 */
struct Request1{
    int MessageCode;
    string InString<MAXLEN>;
}

/* Result1 */
struct Result1 {
    int Status;
    int Value;
    string OutString<MAXLEN>;
}
```

gde je *MessageCode* - kod poruke za realizaciju udaljenog pristupa, *InString* - obrazac po kom se traže deskriptori, *Status* - status pretraživanja baze indeksa, *Value* - vrednost koja pokazuje da li ima još deskriptora koji se poklapaju s obrascem i *OutString* - reč ili fraza koja se poklapa sa obrascem.

(2) *ExpandDRequest* je zahtev serveru za pronalaženje u bazi indeksa svih deskriptora u zadatom polju koji odgovaraju zadatom obrascu. Opis komunikacije pri posleđivanju zahteva je sledeći:

- klijent: otvara konekciju sa serverom
server: prihvata konekciju, ali ne vraća informaciju o statusu
klijent: šalje zahtev - u obliku *Request1*
server: vraća rezultat operacije kao niz poruka oblika *Result2* i zatvara konekciju

```
/* Result2 */
struct Result2{
    int Status;
    int Value;
    unsigned long Count;
    string OutString<MAXLEN>;
}
```

Request1 je opisan kao u (1), dok su *Status* - status pretraživanja baze indeksa, *Value* - vrednost koja pokazuje da li ima još deskriptora koji se poklapaju s obrascem, *Count* - broj pojavljivanja deskriptora u bazi indeksa i *OutString* - reč ili fraza koja se poklapa sa obrascem.

(3) *SelectRequest* je zahtev serveru za pronalaženjem u bazi indeksa imena onih dokumenata koji sadrže zadati deskriptor ili više deskriptora na koje su primenjeni skupovni operatori ili operatori blizine. Opis komunikacije pri prosleđivanju zahteva je sledeći:

- | | |
|----------|---|
| klijent: | otvara konekciju sa serverom |
| server: | prihvata konekciju, ali ne vraća informaciju o statusu |
| klijent: | šalje zahtev - u obliku <i>Request1</i> |
| server: | vraća rezultat operacije kao niz poruka oblika <i>Result1</i> i zatvara konekciju |

Request1 i *Result1* su opisani kao u (1), pri čemu su *InString* - reči ili fraze na koje mogu biti primenjeni skupovni ili operatori blizine, *Value* - vrednost koja pokazuje da li ima još dokumenata koji zadovoljavaju uslov pretraživanja i *OutString* - ime dokumenta koji zadovoljava uslov pretraživanja.

(4) *SelectSRequest* je zahtev serveru za izveštajem o broju pronađenih dokumenata po podizrazima. Opis komunikacije pri prosleđivanju zahteva je sledeći:

- | | |
|----------|---|
| klijent: | otvara konekciju sa serverom |
| server: | prihvata konekciju, ali ne vraća informaciju o statusu |
| klijent: | šalje zahtev - u obliku <i>Request2</i> |
| server: | vraća rezultat operacije kao niz poruka oblika <i>Result2</i> i zatvara konekciju |

Result2 je kao u (2), pri čemu su *Value* - vrednost koja pokazuje da li ima još podizraza, *Count* - broj dokumenata u kojima se nalazi podizraz i *OutString* - podizraz.

```
/* Request2 */
struct Request2{
    int MessageCode;
    int Flag;
    string InString<MAXLEN>;
```

gde je *MessageCode* - kod poruke za realizaciju udaljenog pristupa, *Flag* - pokazuje nivo raščlanjivanja ulaznog stringa na podizrave i *InString* - reči ili fraze na koje mogu biti primenjeni skupovni ili operatori blizine.

(5) *SelectCRequest* je zahtev serveru za pronalaženjem u bazi indeksa imena onih dokumenata koji sadrže zadati deskriptor ili više deskriptora na koje su primenjeni skupovni operatori ili operatori blizine. Vracaju se samo imena dokumenata iz zadatog opsega. Opis komunikacije pri prosleđivanju zahteva je sledeći:

klijent:	otvara konekciju sa serverom
server:	prihvata konckciju, ali ne vraća informaciju o statusu
klijent:	šalje zahtev - u obliku <i>Request</i>
server:	vraća rezultat operacije kao niz poruka oblika <i>Result</i> i zatvara konekciju

Opis zahteva je sledeći:

```
/* Request3 */
struct Request3{
    int MessageCode;
    long Start;
    long Range;
    string InString<MAXLEN>;
```

gde je *MessageCode* - kod poruke za realizaciju udaljenog pristupa, *Start* - pokazuje dokument od kog počinje izveštavanje, *Range* - broj dokumenata u izveštaju i *InString* - reči ili fraze na koje mogu biti primenjeni skupovni ili operatori blizine.

Result1 je kao u (3).

(6) *IndexRequest*, *ReindexRequest*, *DeleteRequest* i *ClearRequest* su zahtevi serveru za indeksiranje, reindeksiranje i brisanje nekog dokumenta. Za razliku od do sada opisanih zahteva, server ne izvršava ove zahteve odmah, već se njihovo izvršavanje odlaže za period manje opterecenosti sistema. Opis komunikacije pri prosleđivanju zahteva je sledeći:

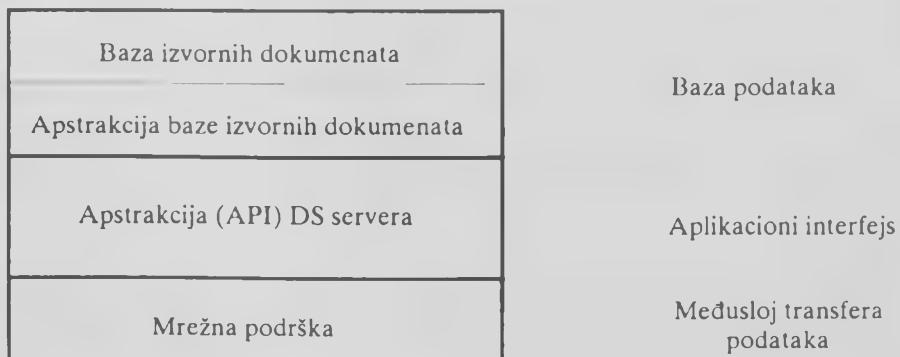
- klijent: otvara konekciju sa serverom
server: prihvata konekciju, ali ne vraća informaciju o statusu
klijent: šalje zahtev - u obliku *Request1*
server: zatvara konekciju

Request1 je kao u (1) pri cemu je *InString* - ime dokumenta koji treba indeksirati/izbrisati.

Napomena: za povratnu informaciju o izvršenom indeksiranju predviđeno je korišćenje drugih komunikacionih mehanizama (na pr. e-mail).

6.4. SERVER ZA PRISTUP BAZI IZVORNIH DOKUMENATA

Organizacija servera za pristup bazi izvornih dokumenata je prikazana na slici 6.9.



Slika 6.9. Organizacija servera za pristup bazi izvornih dokumenata

Jedinstveni sistem za indeksiranje i pretraživanje dokumenata ne definise organizaciju baze izvornih dokumenata. Ovaj sistem samo prepostavlja da se bazi izvornih dokumenta može pristupiti preko definisanog aplikacionog programskog interfejsa. Jedina stvarna uloga DS servera je da pomocu RPC mehanizma obezbedi da klijenti mogu da koriste ovaj API. Zbog ovoga postoje samo minimalne razlike između apstrakcije DS servera i apstrakcije

baze izvornih dokumenata. Ove razlike se isključivo tiču inicijalizacije mrežne podrške i opisane su u [7].

6.4.1. Aplikacioni programski interfejs za pristup bazi izvornih dokumenata

Aplikacioni programski interfejs za pristup bazi izvornih dokumenata se sastoji od biblioteke funkcija koje omogućavaju pristup dokumentima iz baze podataka izvornih dokumenata. Ova biblioteka ima sledeće funkcije:

1. Strukturu dokumenata, bez obzira koliko je ona u stvari složena, biblioteka funkcija za pristup bazi izvornih dokumenata svodi na jednostavnu, linearnu strukturu, u kojoj se dokumenti posmatraju kao niz tekstualnih polja. Svako polje ima svoje ime, koje odgovara imenu polja koje se koristi pri postavljanju upita u sistemu za pretraživanje dokumenata. U jednom dokumentu isto polje se može pojavljivati više puta.
2. Ova biblioteka omogućava da se na isti način pristupa bazi izvornih dokumenata, bez obzira da li se baza nalazi na istom računaru na kome se izvršava program koji je koristi, ili se nalazi na nekom drugom računaru. Biblioteka, takođe, skriva detalje implementacije mrežnih protokola, preko kojih se pristupa udaljenim bazama.

Predložena verzija biblioteke omogućava da se redom pristupa poljima jednog tipa (t.j. poljima istog imena) u okviru izabranog dokumenta. Korišćenjem funkcija ove biblioteke, međutim, nije moguće dobiti informaciju o stvarnom redosledu polja u izvornom dokumentu, što znači da je biblioteka namenjena pre svega podršci indeksiranju dokumenata.

Postoje dva tipa biblioteke za pristup bazi izvornih dokumenata. To su:

1. Biblioteka za neposredan pristup bazi izvornih dokumenata. Ove biblioteke zavise od strukture dokumenata, tako da se pišu posebno za svaki tip baze izvornih dokumenata.
2. Biblioteka za pristup udaljenoj bazi izvornih dokumenata. Ovaj tip biblioteke koristi usluge odgovarajućeg servera, koji se, tipično, izvršava na računaru na kome se nalazi baza izvornih dokumenata. Serveri dokumentima pristupaju koristeći bibliotekе za neposredan pristup bazi izvornih dokumenata.

Biblioteke za pristup udaljenim bazama izvornih dokumenata, zajedno sa odgovarajućim serverom, predstavljaju u stvari sistem koji realizuje RPC (*Remote Procedure Call*) interfejs prema bibliotekama za neposredan pristup.

Korišćenje ove biblioteke je jednostavno. Prvo se inicijalizuje biblioteka funkcijom *dsInit*. Pri ovome se i specificira gde se nalazi baza izvornih dokumenata. Dokument, kome se pristupa, se bira funkcijom *dsOpenDocument*. Funkcijom *dsOpenField* se inicijalizuje iteracija po poljima određenog tipa (imena) u izabranom dokumentu. Tekst polja se dobija pozivom funkcije *dsNextField*, pri čemu suksesivni pozivi ove funkcije vraćaju redom sva polja datog tipa. Kada funkcija *dsNextField* vrati kao svoju vrednost 0, to je znak da u posmatranom dokumentu, više nema polja datog tipa. Funkcija *dsCloseField* prekida proces iteracije po poljima. Funkcijom *dsCloseDocument* prekida se pristup izabranom dokumentu. Veza sa bazom izvornih dokumenata se raskida funkcijom *dsTerm*.

6.4.2. Protokol za komunikaciju sa serverom za pristup bazi izvornih dokumenata

Zahtevi kojima se komunicira sa serverom za pristup bazi izvornih dokumenata imaju sličnu strukturu. Za razliku od rada sa serverom za pretraživanje baze indeksa, gde je za svaki zahtev otvarana nova konekcija, kod servera za pretraživanje baze izvornih dokumenata, jedna konekcija traje do završetka sesije. Tipovi zahteva su:

1. *OpenDocumentRequest* - zahtev serveru za pronalaženje dokumenta sa datim imenom u bazi izvornih dokumenata.
2. *OpenFieldRequest* - zahtev za pronalaženje prvog pojavljivanja datog polja u odabranom dokumentu. Dokumenti se iz baze ne uzimaju u celini već polje po polje.
3. *NextFieldRequest* - zahtev za vraćanje prvog/sledećeg pojavljivanja polja datog imena u odabranom dokumentu.
4. *CloseFieldRequest* - zahtev za zatvaranje pretraživanja po polju.
5. *CloseDocumentRequest* - zahtev za zatvaranje dokumenta.

Napomena: zahtevi 2-4 se obično ponavljaju više puta.

(1) *OpenDocumentRequest*. Opis komunikacije pri prosleđivanju zahteva je sledeći:

- | | |
|----------|---|
| klijent: | otvara konekciju sa serverom |
| server: | prihvata konekciju, ali ne vraća informaciju o statusu |
| klijent: | šalje zahtev - u obliku <i>Request1</i> |
| server: | vraća rezultat operacije kao poruku oblika <i>Result1</i> |

Slede opisi zahteva i odgovora:

```
const MAXLEN = 64
```

```
/* Request1 */
struct Request1{
    int MessageCode;
    string InString<MAXLEN>;
}

/* Result1 */
struct Result1{
    int Status;
}
```

gde je *MessageCode* - kod poruke za realizaciju udaljenog pristupa, *InString* - ime dokumenta, *Status* - status pretraživanja baze izvornih dokumenata.

(2) *OpenFieldRequest*. Opis komunikacije pri posleđivanju zahteva je sledeći:

klijent:	otvara konekciju sa serverom
server:	prihvata konekciju, ali ne vraća informaciju o statusu
klijent:	šalje zahtev - u obliku <i>Request1</i>
server:	vraća rezultat operacije kao poruku oblika <i>Result1</i>

Request1 i *Result1* su kao u (1) pri čemu je *InString* ime polja.

(3) *CloseFieldRequest*. Opis komunikacije pri posleđivanju zahteva je sledeći:

klijent:	otvara konekciju sa serverom
server:	prihvata konekciju, ali ne vraća informaciju o statusu
klijent:	šalje zahtev - u obliku <i>Request2</i>
server:	vraća rezultat operacije kao poruku oblika <i>Result1</i>

```
/* Request2 */
struct Request2{
    int MessageCode;
}
```

Result1 je kao u (1).

(4) *CloseDocumentRequest*. Opis komunikacije pri prosleđivanju zahteva je sledeći:

- | | |
|----------|---|
| klijent: | otvara konekciju sa serverom |
| server: | prihvata konekciju, ali ne vraca informaciju o statusu |
| klijent: | šalje zahtev - u obliku <i>Request2</i> |
| server: | vraća rezultat operacije kao poruku oblika <i>Result1</i> i raskida konekciju |

Request2 i *Result1* su kao i do sada.

(5) *NextFieldRequest*. Opis komunikacije pri prosleđivanju zahteva je sledeći:

- | | |
|----------|---|
| klijent: | otvara konekciju sa serverom |
| server: | prihvata konekciju, ali ne vraća informaciju o statusu |
| klijent: | šalje zahtev - u obliku <i>Request2</i> |
| server: | vraća rezultat operacije kao poruku oblika <i>Result2</i> |

```
/* Result2 */
struct Result2{
    int Status;
    int Value;
    string OutString<MAXLEN>;
```

gde je *Status* - status pretraživanja baze izvornih dokumenata, *Value* - vrednost koja pokazuje da li ima još pojavljivanja tog polja i *OutString* - sadržaj polja.

6.5. GEJTVÉJI SISTEMA

Raširena upotreba različitih Internetovih servisa (Gopher, WorldWideWeb, ...), stvara potrebu integrisanja sistema za pretraživanje i indeksiranje dokumenata u Internet što je postignuto realizacijom gejtvéja za Gopher i HTTP protokole. Realizacijom protokola je omogućeno korišćenje standardnih korisničkih aplikacija, što ukida potrebu za definisanjem i

distribucijom posebne klijentske aplikacije sistema za pretraživanje dokumenata.

6.5.1. Gopher

Gopher gejtvej, je realizovan kao server koji će primati zahteve od Gopher klijenta [1], prevoditi ih na protokol definisan za pristup serverima za pretraživanje, a zatim rezultate pretraživanja interpretirati tako da budu razumljivi Gopheru. Drugim rečima gejtvej prema serverima zauzima poziciju klijenta, a prema Gopher klijentu poziciju servera.

Za pristup Gopher gejtveja, serveru za pretraživanje baze indeksa i serveru za pristup bazi izvornih dokumenata, korišćeni su aplikacioni programski interfejsi za pretraživanje i indeksiranje dokumenata i za pristup bazi izvornih dokumenata koji omogućavaju integraciju sistema za pretraživanje u druge programske sisteme.

Sa strane Gophera, Gopher gejtvej se uključuje u Gopher prostor korišćenjem opcije za definisanje objekata tipa indeks server. Gejtvej je realizovan po standardnom modelu UNIX-ovog konkurentnog, konekciono orijentisanog servera.

Komande sistema za pretraživanje tekstualnih baza podataka, koje je prirodno realizovati u Gopher-u su: selektovanje dokumenata koji ispunjavaju zadati uslov pretraživanja (*select*, *selects*), pronalaženje svih reči u bazi indeksa koje odgovaraju zadatom obrascu (*expand*, *expandd*) i pronalaženje odabranog dokumenta u bazi izvornih dokumenata i njegova prezentacija klijentu.

Komande *select* i *selects* su u Gopher prostoru predstavljene kao objekti tipa index server, selektor string je "select", tj. "selects", kao adresa hosta i port dati su adresa gejtveja i port na kome gejtvej očekuje konekciju.

Zahtev prolazi kroz proceduru za parsiranje i Gopher gejtvej zaključuje o kojoj se komandi radi i poziva odgovarajuću funkciju aplikacionog programskega interfejsa za pretraživanje i indeksiranje dokumenata.

Klijentska aplikacija će rezultate dobijene od servera za pretraživanje prikazati kao listu imena svih dokumenata koji su zadovoljili zadati kriterijum, pružajući mogućnost korisniku da odabere neki od dokumenata i pošalje zahtev za njegovo pronalaženje u bazi izvornih dokumenata.

Koncepcija pri realizaciji komandi *expand* i *expandd* je slična. I ove komande su predstavljene korisniku kao (drugi) objekti tipa index server, selektor string je po analogiji "expand", odnosno "expandd", a adresa hosta i port su isti.

Razlika u odnosu na prethodni slučaj je u tretmanu rezultata. Sada će se kao rezultat pretraživanja dobiti lista reči i fraza (deskriptora) koje odgovaraju zadatom obrascu, a ne lista imena dokumenata. Postavlja se pitanje koji tip objekta dodeliti ovakvom rezultatu. Očigledno, ovakav završetak komande ne bi bio sasvim u duhu Gophera, a prirodno je i pružiti korisniku mogućnost da odmah zada kao kriterijum za pretraživanje neku od reči dobijenih izvršavanjem komandi *expand* i *expandd*. Zbog toga, za obradu rezultata ovog pretraživanja potrebno je definisati drugu funkciju koja će rezultatu dodavati prefiks "select", omogućavajući da se u daljem radu ovaj rezultat tretira kao *select* komanda.

Procedura zadužena za parsirnje ovakvog zahteva će interpretirati na isti način na koji bi interpretirala i "regularni" *select* zahtev, pa će se kao rezultat, na način prethodno opisan, ponovo dobiti lista imena dokumenata. Ovde, međutim, postoji i jedno ograničenje: korisnik nema mogućnost da specifikuje po kom polju pretraživanje treba da se vrši, pa se pretraživanje vrši samo po baznim poljima.

Izveštaj o dokumentu kao rezultat prethodnih pretraživanja može se dobiti na dva načina, u zavisnosti od načina organizacije dokumenata u bazi izvornih dokumenata.

Prvi način podrazumeva korišćenje funkcija aplikacionog programskog interfejsa za pristup bazi izvornih dokumenata koje posredstvom servera za pristup bazi izvornih dokumenata čitaju polje po polje dokumenta i kreiraju izveštaj.

Uočava se da postoje dva moguća pristupa ovom problemu: moguće je pripremiti unapred izvestan broj fiksnih formata za prezentaciju dokumenta, ostavljajući korisniku da izabere najpogodniji, ili kreirati mehanizam kojim će korisnik sam definisati format dokumenta.

Za čuvanje formata dokumenata koji se mogu ponuditi korisniku, za svaku bazu definiše se posebna datoteka pod imenom:

ime_baze.frmt

Struktura ove datoteke je sledeća:

```
[Admin: {komentar} <e-mail adresa>]  
[Redni_broj]  
ime_polja="opis polja"  
{ ime_polja="opis polja" ...}  
{[Redni_broj]  
{ ime_polja="opis polja" ...}}
```

Dakle, prvi red u datoteci je rezervisan za podatke o administratoru baze, u uglastim zagradama, posle čega slede redni broj formata, takođe u uglastim zagradama, i niz linija sa imenima polja i njihovim opisom, pri čemu opis polja može biti i prazan string.

Drugi način se primenjuje u slučaju kada postoje posebni fajlovi za izveštavanje. Gopher gejtvej tada pristupa direktno fajlovima i prezentuje ih korisniku onakve kakvi jesu.

6.5.2. WWW

Pri projektovanju mehanizma koji omogućava korisniku WWW-a korišćenje podataka u SNTIS-u moguća su dva pristupa. Klasičan način realizovanja gejtveja između WWW-a i sistema za indeksiranje i pretraživanje koji je razvijen za potrebe SNTIS-a, podrazumeva kreiranje procesa koji prema korisniku zauzima poziciju WWW servera i na poznatom portu prima zahteve korisnika. Proces transformiše zahteve na jezik razumljiv sistemu za pretraživanje i sa pozicija klijenta prosleđuje ih serveru sistema za pretraživanje. Povratne informacije usklađuju se sa pravilima HTTP protokola [2] i vraćaju se klijentu. Drugi pristup je korišćenje CGI specifikacije (*Common Gateway Interface*) koja je namenjena uključivanju drugih aplikacija u Web servis. Realizacija ovog načina je znatno jednostavnija i preporučljiva i pored izvesnog gubitka na performansama sistema. Osnovni mehanizam u implementaciji gejtveja korišćenjem CGI-a se sastoji u dodavanju zahtevu korisnika, pri njegovom slanju WWW serveru, imena programa koji treba da obradi zahtev. Server po prijemu zahteva aktivira program i prosleđuje mu zahtev na dalju obradu. Zadatak programa je da ga prevede u oblik razumljiv serveru za pretraživanje i indeksiranje. Rezultat pretraživanja se zatim u obliku HTML [15] dokumenta vraća istim putem korisniku.

Komande sistema za pretraživanje koje su realizovane na ovaj način su tipa: pretraživanje po zadatom kriterijumu (*select*, *selects*) ili pronalaženje deskriptora koji odgovaraju zadatom obrascu (*expand* i *expandd*).

Pri realizaciji komandi *select* i *selects* za unošenje argumenta je korišćena forma tipa "tekst" i POST metod za komunikaciju sa serverom. Formi se, korišćenjem opcije ACTION, dodeljuje aplikacija koja obrađuje argumente unete kroz formu. POST metod znači da će aplikacija dodeljena formi primiti argument na standardnom ulazu. Drugi metod za prosleđivanje argumenata serveru je GET metod, ali njegovim korišćenjem string stiže do servera posredstvom promenljive QUERY_STRING, te može doći do odsecanja argumenta. Po priјemu zahteva server pokreće program dodeljen formi koji vrši parsiranje upita i rezultate vraća u obliku liste dokumenata odnosno termova.

Koncepcija pri realizaciji komandi *expand* i *expandd* je slična. Razlika u odnosu na prethodni slučaj je u tretmanu rezultata. Sada će se kao rezultat pretraživanja dobiti lista reči i fraza (deskriptora) koje odgovaraju zadatom obrascu, a ne lista imena dokumenata. Postavlja se pitanje koji tip objekta dodeliti ovakvom rezultatu. Prirodno je pružiti korisniku mogućnost da odmah zada kao kriterijum za pretraživanje neki podskup skupa reči dobijenih izvršavanjem komandi *expand* i *expandd*. Zbog toga, za obradu rezultata ovog pretraživanja potrebno je definisati drugu funkciju koja će formirati "select" upit čiji izraz čini disjunkcija izabranih termova.

Procedura zadužena za parsiranje ovakav zahtev će interpretirati na isti način na koji bi interpretirala i "regularni" *select* zahtev, pa će se kao rezultat, na način prethodno opisan, ponovo dobiti lista imena dokumenata. Ovde, međutim, postoji i jedno ograničenje: korisnik nema mogućnost da specifikuje po kom polju pretraživanje treba da se vrši, pa se pretraživanje vrši samo po baznim poljima.

WWW omogućava dobijanje formatiranih izveštaja korišćenjem unapred definisanih fajlova za izveštavanje u HTML formatu. Drugi pristup je implementacija modula za izvestavanje u okviru gejtveja, pri čemu se polja dokumenta dobijaju preko servera za pristup bazi izvornih dokumenata.

6.6. ZAKLJUČAK

Jedinstveni sistem za indeksiranje i pretraživanje dokumenata je svoje prve primene nasao u Rečniku podataka SNTIS, Bibliotečkom informacionom sistemu SNTIS i bazi podataka projekata programa tehnološkog razvoja

Ministarstva za nauku i tehnologiju Republike Srbije. Iskustva su veoma povoljna, a pogotovo ona koja se tiču otvorenosti i prenosivosti sistema.

Modularnost sistema uz slabu spregnutost modula, sa precizno definisanim aplikacionim programskim interfejsima, kao i dosledna primena prihvaćene metodologije i opšte arhitekture sistema su kao rezultat dali otvoren sistem koji je omogućio da se na njemu bazira implementacija većeg broja eksperimentalnih servera koji bi trebalo da pomognu u izboru optimalnog načina distribucije pretraživanja dokumenata u SNTIS. Kao primer može se navesti realizacija servera za indeksiranje i pretraživanje dokumenata koji podržava rad sa više baza izvornih dokumenta, te na taj način realizuje funkciju centralizovanog indeksa. U planu je da se implementira i server koji će, u spremi sa rečnikom podataka, omogućiti pretraživanje više baza podataka na osnovu jednog upita, pri čemu se od korisnika ne bi tražilo da precizno odredi o kojim se bazama podataka radi.

Od interesa je da se ispita i koje su mogućnosti povezivanja Jedinstvenog sistema za pretraživanje informacija sa drugim srodnim sistemima, kao što je, na primer, popularni CDS/ISIS [16].

U toku je i portiranje sistema i na Windows platformama (Windows 95 i Windows NT), kao i završetak implementacije agenta za pretraživanje informacija.

U narednom periodu bi trebalo raditi i na poboljšanju brzine indeksiranja i smanjenju veličine baze indeksa. Problem efikasnosti procesa indeksiranja je jedini problem koji je uočen u dosadašnjoj eksploataciji sistema.

LITERATURA

- [1] Anklesaria, F., McCahill, M., Lindner, P., Johnson, D., Torrey, D., Alberti, B., *The Internet Gopher Protocol (a distributed document search and retrieval protocol)*, Network Working Group, RFC 1436 March 1993.
- [2] Berners-Lee, T., Fielding, R. T., Frystyk Nielson, H., *Hypertext Transfer Protocol -- HTTP/1.0*, HTTP Working Group, March 1995.
- [3] Searching DIALOG The Complete Guide, Dialog Information Services, Inc. January 1989.
- [4] Stevens, R., UNIX network programming, Prentice Hall Software Series, 1990.

- [5] Sun Microsystems, Inc. *RPC: Remote Procedure Call Protocol Specification Version 2*, Network Working Group, RFC 1057 June 1988.
- [6] rs: Aplikacioni programski interfejs RS servera Jedinstvenog sistema za indeksiranje i pretraživanje dokumenata u SNTIS, Institut Mihajlo Pupin, Beograd 1994. (Specifikacija je dostupna na FTP serveru bisd.imp.bg.ac.yu).
- [7] ds: Aplikacioni programski interfejs DS servera Jedinstvenog sistema za indeksiranje i pretraživanje dokumenata u SNTIS, Institut Mihajlo Pupin, Beograd 1994. (Specifikacija je dostupna na FTP serveru bisd.imp.bg.ac.yu).
- [8] pc: Aplikacioni programski interfejs komunikacione biblioteke Jedinstvenog sistema za indeksiranje i pretraživanje dokumenata u SNTIS, Institut Mihajlo Pupin, Beograd 1994. (Specifikacija je dostupna na FTP serveru bisd.imp.bg.ac.yu).
- [9] Milivojević, Lj., *Modul za izvršavanje upita u tekstualnim bazama podataka SNTIS*, YU-Info, Brezovica, 1995.
- [10] Konečni, A., Milivojević, Lj., *Gopher gateway za pristup tekstualnim bazama podataka u sistemu SNTIS*, YU-Info, Brezovica, 1995.
- [11] Graefe, G., *Query Evaluation Techniques for Large Databases*, acm Computing Surveys, Vol. 25, No. 2, pp 73-170, 1993.
- [12] Đuričić, D., Konečni, A., Milivojević, Lj., *Podsistem za upravljanje bazom podataka indeksa kao deo sistema za pretraživanje dokumenata*, YU-Info, Brezovica, 1995.
- [13] Interfejs baze indeksa sistema za pretraživanje tekstualnih baza podataka SNTIS-a, Institut Mihajlo Pupin, Beograd 1994. (Specifikacija je dostupna na FTP serveru bisd.imp.bg.ac.yu).
- [14] Sun Microsystems, Inc. *XDR:External Data Representation Standard*, Network Working Group, RFC 1057 June 1987.
- [15] Raggett, D., *Hypertext Markup Language Specification Version 3.0*, Internet draft, March 1995.
- [16] Mini-micro CDS/ISIS REFERENCE MANUAL (Version 2.3), Division of Software Development and Applications Office of Information Programmes and Services, Unesco, March 1989.

Poglavlje 7.

SISTEM ZA FORMIRANJE I PRETRAŽIVANJE BAZE DOKUMENATA I PUBLIKOVANJE

Sistem za formiranje i pretraživanje baze dokumenata treba da omogući formiranje i pretraživanje baza podataka u kojima se pamti pun sadržaj dokumenata (full text) tako da se oni mogu pretraživati po bilo kom svom elementu (delu sadržaja dokumenta) odnosno na odgovarajući nacin dalje publikovati. Razvojem ovakvog sistema se stvaraju uslovi za efikasan i standardizovan obuhvat naučnih i stručnih radova na univerzitetima i drugim stručnim i naučnim institucijama u Srbiji i Jugoslaviji uz mogućnost potpunog uvida u njihov sadržaj, a kasnije on postaje standardni deo mreže i softverske infrastrukture SNTIS sa svim elementima koji su karakteristični za kompletne multimedijalne informacione servise.

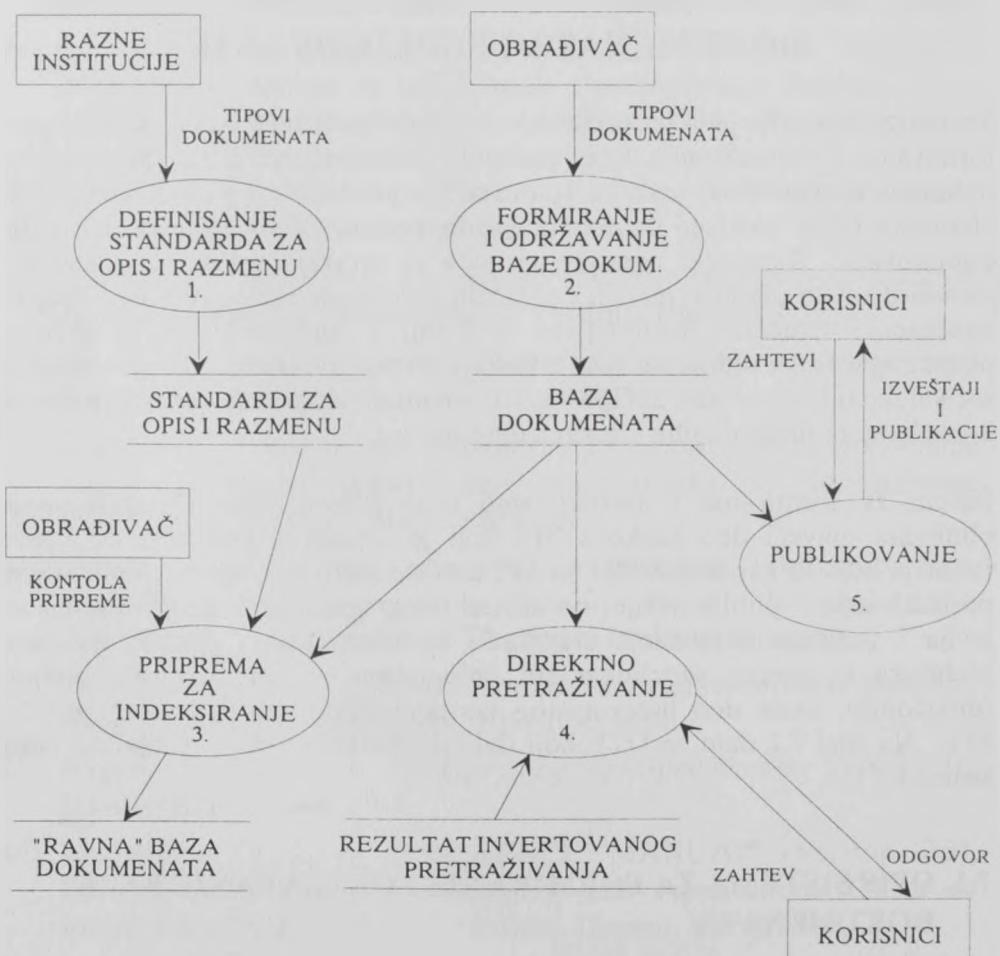
Sistem za formiranje i pretraživanje baze dokumenata i publikovanje obuhvata najveći deo funkcija SPI koji je opisan u Poglavlju 5. Pored funkcija koje su karakteristične za SPI kao što su formiranje baza podataka, pretraživanje i publikovanje, na nivou ovog specijalizovanog sistema se javlja i funkcija definisanja standarda za formiranje i održavanje baza podataka sa punim sadržajem što omogućava da on, iako autonomno funkcioniše, bude deo heterogenog kooperativnog distribuiranog sistema SPI. Na slici 7.1 daje se DTP koji definiše funkcije ovog specijalizovanog sistema.

7.1. OPIS SISTEMA ZA FORMIRANJE I ODRŽAVANJE BAZE DOKUMENATA

U ovom poglavlju detaljno se opisuju pojedini procesi sa dijagrama na slici 7.1.

7.1.1. Proces Definisanje standarda za opis i razmenu dokumenata

U cilju definisanja procedura i formata za obradu, smeštanje i pretraživanje dokumenata koji ulaze u bazu dokumenata, razmatrane su dve grupe standarda koji tretiraju problematiku obrade tekstualnih dokumenata: SGML i ODA standardi. Na bazi iskustava i preporuka u korišćenju ovih standarda, prirodu dokumenata koji cirkulišu u sistemu STNIS i zahteva da oni budu pretraživi po različitim elementima, kao standard za opis i razmenu dokumenata izabran je SGML.



Slika 7.1. Sisitem za formiranje i pretraživanje baze dokumenata

Standardom ISO 8879 (SGML) definisane su karakteristike kao i osnovni koncept SGML jezika koji se može koristiti za obradu punog teksta dokumenata.

SGML je formalizovani jezik kojim se tekst dokumenta može obeležavati tako da se u potpunosti očuva njegov sadržaj, struktura a ako je potrebno i njegov izgled. SGML jezik se zasniva na dva osnovna principa:

a) Elementi za označavanje koji se unose u tekst dokumenta su deskriptivni, svaki od njih ima svoj tzv. generički identifikator (GI) i svoje attribute. Generički identifikatori su mnemonička imena koje korisnik bira sam, tako da na najbolji način okarakteriše tip elementa. Preko ovih elemenata za označavanje se mogu pozivati različite procedure za obradu elemenata, i one nisu sadržane u samom dokumentu. Za svaki element za označavanje se može vezati više različitih procedura, što u principu dozvoljava različitu obradu tog dokumenta.

b) Unošenje elemenata za označavanje u sam tekst dokumenta je formalno definisano za svaki tip dokumenta preko DEFINICIJE TIPA DOKUMENTA (DTD). DTD predstavlja specifikaciju (formalnu gramatiku) elemenata i njihovih atributa koji se mogu pojaviti u datom tipu dokumenta i redosled njihovog pojavljivanja. Na bazi ovako koherentne i nedvosmislene sintakse moguće je utvrditi da li je označavanje svakog pojedinačnog dokumenta korektno ili ne, a u slučaju da to nije mogu se izvršiti odgovarajuće korekcije. SGML-om nisu definisani DTD-ovi za različite vrste dokumenata, već je potrebno da to definiše sam korisnik na bazi elemenata i pravila definisanih SGML jezikom.

Koncept na kojem se SGML jezik zasniva je uspostavljanje veze između samog dokumenta, DTD-a i procedura za obradu.

Dokument se posmatra kao logička struktura sastavljena od elemenata prepoznatljivog tipa koja se može predstaviti strukturom grafa. Svaki element dokumenta je izdvojen posebnim elementom za označavanje kojim se identificuje njihov početak i kraj. Dokument tako sadrži dve vrste podataka: sam tekst - sadžaj dokumenta i elemente za označavanje definisane SGML jezikom. U okviru SGML jezika elementi za označavanje se mogu svrstati u četiri grupe:

a) TAGOVI (Tags) - Njima se označava početak i kraj svakog identifikovanog elementa dokumenta (start-tag i end-tag). Oni imaju deskriptivna svojstva obzirom da se njima opisuju kvalitativne karakteristike elemenata na koje se odnose. Jedna od tih karakteristika je generički identifikator (GI) - mnemoničko ime koje korisnik bira tako da na najbolji način okarakteriše tip elementa kao i njegove attribute. Tagovima se takođe

identificuju strukturalne veze između elemenata dokumenta. Tagovi su najčešće korišćeni elementi za označavanje.

b) **REFERENCE ENTITETA** (Entity Reference) - služe da se povežu različiti delovi dokumenata u slučaju da se dokument fizički pamti u više memoriskih jedinica (datoteka, biblioteka itd.) Entiteti mogu biti definisani u okviru samog dokumenta ili eksterno.

c) **DEKLARACIJA ELEMENATA** (Markup Declaration) - rečenice kojima se kontroliše interpretacija elemenata za označavanje (koriste se npr. za definisanje entiteta, kreiranje DTD-a itd.)

d) **INSTRUKCIJE ZA OBRADU** (Processing Instruction) - instrukcije kojima se u određenom jeziku definišu akcije koje treba preduzeti nad određenim elementima dokumenta. Za razliku od drugih elemenata za označavanje ove instrukcije zavise od konkretnog računara i aplikacije. One se menjaju u slučaju kada se dokument želi obraditi na različite načine ili kada se prenosi na drugi sistem.

Elementi za označavanje kao što su reference entiteta, deklaracije elemenata za označavanje i instrukcije za obradu, kada se unose u tekst dokumenta, označavaju se odgovarajucim delimiterima.

DEFINICIJA TIPA DOKUMENTA je jednoznačno definisan skup deklaracija elemenata za označavanje koji se može primeniti nad dokumentima određenog tipa. U okviru DTD-a se javljaju tri osnovne vrste deklaracija elemenata za označavanje.

a) **Deklaracija elemenata dokumenta** kojom se definišu sledeći parametri:

- Generički identifikator (GI) datog elementa dokumenta;
- Minimizacija- odnosno mogućnost izostavljanja tagova za označavanje kraja u slučaju kada se u okviru sadržaja nekog elementa dokumenta nađe početni tag drugog elementa koji ne može da se pojavi unutar njega;
- Sadržaj elementa, odnosno grupe elemenata koja se može pojaviti u okviru datog elementa na njegovom nižem nivou, označenih takođe svojim generičkim identifikatorima.

Pojedini GI u okviru sadržaja se razdvajaju konektorima kojima se definise redosled njihovog pojavljivanja. Mogući konektori i njihova značenja dati su u tabeli na slici 7.2. Uz svaki GI u okviru sadržaja može da stoji identifikator pojavljivanja kojim se određuje koliko puta se neki GI pojavljuje u tom sadržaju. Mogući identifikatori pojavljivanja dati su u tabeli na slici 7.3.

Konektori	
Značenje konektora	Znak
Sekvenca u kojoj je važan redosled pojavljivanja	
Unija	I
Sekvenca u kojoj nije važan redosled pojavljivanja	\$

Slika 7.2. Tabela konektora

Identifikator pojavljivanja	
Značenje indikatora	Znak
Opciono	?
Obavezno i ponovljivo	+
Opciono i ponovljivo	*

Slika 7.3. Tabela identifikatora pojavljivanja

Sadržaj elemenata može da bude: grupa drugih elemenata, niz od nula ili više karaktera što se označava odgovarajućim rezervisanim imenom (PCDATA), prazan, što se koristi u slučaju kada korisnik ne unosi sadržaj elementa (npr. slike ili drugi grafički sadržaji), već se njihov sadržaj generiše pozivom određenih instrukcija za obradu (pri čemu se koristi rezervisano ime EMPTY).

b) Deklaracija atributa kojim se opisuju svojstva nekog elementa ili DTD-a. Deklaracija atributa se sastoji od parametara kojim se određuju: imena atributa, deklarisane vrednosti arributa (mogu biti različitih tipova - karakter (CDATA), numerik (NUMBERS), validno SGML ime, ID vrednost itd.) i default vrednosti atributa (mogu biti definisane kao niz karaktera ili rezervisane reči #REQUIRED - atribut je obavezan, #IMPLIED - atribut je opcion ili #FIXED - atribut je fiksan).

c) Deklaracija entiteta - sadrži dva osnovna parametra: ime entiteta i tekst entiteta. Pozivanje entiteta koristi se u slučajevima kada se: dugačak niz karaktera, koji se više puta pojavljuje u tekstu može zameniti kratkim imenom entiteta u okviru kojeg je taj tekst definisan, kada treba povezati delove dokumenta koji se nalaze u različitim sistemskim datotekama ili kada treba povezati instrukcije za obradu dokumenta (processing instruction) sa samim dokumentom, tako da postanu njegov integralni deo itd. Pozivanje predhodno definisanog entiteta je moguće bilo gde u okviru sadržaja nekog elementa ili vrednosti atributa.

Instrukcije za obradu - Obzirom da instrukcije za obradu zavise od konkretnog računara i aplikacije, one u principu umanjuju prenosivost dokumenta sa sistema na sistem, te se zbog toga uvek ne definišu. Ako se instrukcije za obradu moraju koristiti, one se definišu kao entiteti tako da se na sistemu na koji se dokument prenosi mogu lako locirati i modifikovati.

Elementi dokumenta sadržani u DTD-u formiraju hijerahiju ili aciklični graf koji je upravljan korenom, tako da se za elemenat dokumenta C_i vezuju

elementi nižeg nivoa S_i tako da se može reći da su elementi S_i podelementi elementa C_i a da je element C_i nadređeni element elemenata S_i . U ovom modelu, jedan element može da ima bilo koji broj podelemenata. Neki sistemi dozvoljavaju da element ima samo jedan nadređeni element dok drugi dozvoljavaju da element ima bilo koji broj nadređenih elemenata. U prvom slučaju elementi formiraju hijerarhiju, dok u drugom elementi formiraju aciklični graf koji ima jedan koren i u kome nema izolovanih čvorova (svi su povezani i do svakog čvora se može stići polazeći od korena). Hijerarhija elemenata može da sadrži i ciklične grane (grana je ciklična ako sadrži elemente C_i i C_j tako da je C_j podelement elementa C_i i C_i podelement elementa C_j).

DTD koji odgovara strukturi i sastavu dokumenata koji se mogu sruštati u grupu monografija a koji je definisan na osnovu predhodno opisanih pravila i principa SGML jezika, dat je na slici 7.4.

```

<!-- Entity -->
<! ENTITY %model_recenice &#40;PCDATA) | (%recenica;)*&#41; model rečenice >
<! ENTITY %recenica &#40;q | (%naglasena_recenica;) | (%referentna_recenica;) | (%prazna_recenica;)*&#41; rečenica >
<! ENTITY %naglasena_recenica &#40;hp1 | hp2| hp3| hp0| q*&#41; >
<! ENTITY %referentna_recenica &#40;href | figref*&#41; >
<! ENTITY %prazna_recenica &#40;fnref | lref*&#41; >
<! ENTITY %sadrzaj_poglavlja &#40;(%prost_paragraf;) | (%podelement_poglavlja;) | (%tema;)*&#41; sadržaj poglavlja >
<! ENTITY %prost_paragraf &#40;p | note*&#41; >
<! ENTITY %podelement_poglavlja &#40;(%drugi_elementi;) | (%liste;)*&#41; >
<! ENTITY %drugi_elementi &#40;top1 | top2| top3| top4* &#41; >
<! ENTITY %liste &#40;xmp | lines|tbl | artwork*&#41; >
<! ENTITY %tema &#40;ol | ul | gl*&#41; >
<! ENTITY %model_paragrafa &#40;(#PCDATA) | (%recenica;) | (%podelement_poglavlja;)*&#41; model paragrafa >
<! ENTITY %sekvenca_paragrafa &#40;p,((%prost_paragraf;) | (%podelement_poglavlja;)*&#41; sekvenca paragrafa >
<! ENTITY %model_teme &#40;(th?, p, (%sadrzaj_poglavlja;)*&#41; model teme >
<! ENTITY %tekuci_elementi &#40;artwork | fn*&#41; tekuci elementi >

<!-- Struktura dokumenta -->
<! ELEMENT biblio_m &#40;title+, stitle*, autor1+, abstrakt*&#41; bibliografski opis za monografije >
<! ELEMENT title &#40;#PCDATA)&#41; naslov monografije >

```

<! ELEMENT stitle		(#PCDATA)		>
<! ELEMENT autorl		((autor inst?)+)	lista autora i institucija u kojima rade	>
<! ELEMENT autor	- 0	(initials, surname)		>
<! ELEMENT initials	0 0	(#PCDATA)		>
<! ELEMENT surname	- 0	(#PCDATA)		>
<! ELEMENT inst	- 0	(#PCDATA)		>
<! ELEMENT abstrakt	--	(p+)	abstrakt	>
<! ELEMENT p	- 0	(#PCDATA)		>
<! ELEMENT body	- 0	<!-- Sadržaj dokumenta --> (h0+ h1+)		>
<! ELEMENT h0	- 0	(h0l, (%sadrzaj_poglavlja)*, h1+)		>
<! ELEMENT h1 float	- 0	(h1l, (%sadrzaj_poglavlja)*, h2*)		>
<! ELEMENT h2	- 0	(h2l, (%sadrzaj_poglavlja)*, h3*)		>
<! ELEMENT h3	- 0	(h3l, (%sadrzaj_poglavlja)*, h4*)		>
<! ELEMENT h4	- 0	(h4l, (%sadrzaj_poglavlja)*)		>
<! ELEMENT h0l h1l h2l h3l h4l	0 0	%model_recenice		>
<! ELEMENT (%naglasena_recenica)	--	<!-- Rečenice --> %model_recenice	naglašena rečenica	>
<! ELEMENT (%referntna_recenica)	- 0	%model_recenice	referentna rečenica	>
<! ELEMENT (%prazna_recenica)	- 0	EMPTY	prazna rečenica	>
<! ELEMENT q		%model_recenice	element pod navodnikom	>
<! ELEMENT (%prost_paragraf)	--	<!-- Elementi iz sadržaja poglavlja --> %model_paragrafa	prost paragraf	>
<! ELEMENT top1	- 0	%model_tcmc	tema1	>
<! ELEMENT top2	- 0	%model_tcmc	tema2	>
<! ELEMENT top3	- 0	%model_teme	tema3	>
<! ELEMENT top4	- 0	%model_teme	tema4	>
<! ELEMENT th	- 0	%model_recenice	naslov teme	>
<! ELEMENT xmp	0 0	<!-- Elementi iz podelementa poglavlja --> %sekvenca_paragrafa	primjeri	>
<! ELEMENT lines	0 0	%sekvenca_paragrafa	linije	>
<! ELEMENT tbl		(r+)	tabele	>
<! ELEMENT r	0 0	(c+)	vrsna u tabeli	>
<! ELEMENT c	0 0	%sekvenca_paragrafa	kolona za određenu vrsnu	>
<! ELEMENT fig	- 0	EMPTY	slika	>
<! ELEMENT (%listc)	-	ol ul gl	listo	>
<! ELEMENT ol ul	-	(li*)	uređena lista	>
<! ELEMENT gl	-	((gl, gd)*)	rečnik termina	>
<! ELEMENT gl	- 0	(#PCDATA)		>
<! ELEMENT gd	- 0	%sekvenca_paragrafa;		>
<! ELEMENT li	- 0	%sekvenca_paragrafa		>
<! ELEMENT fig		<!-- Tekuci elementi --> (figbody, (figcap, figdesc)?)		>
<! ELEMENT figbody	0 0	%sekvenca_paragrafa	telo crteža	>
<! ELEMENT figcap	- 0	%model_recenica	nalov crteža	>
<! ELEMENT figdesc	- 0	%sekvenca_paragrafa	opis crteža	>
<! ELEMENT fn		%sekvenca_paragrafa	fusnote	>

		<!-- Atributi -->			
		IME	VREDNOST	DEFAULT	
<!--	ELEMENT	status	CDATA	#IMPLIED	
<! ATTLIST	general	verzija	CDATA	srpski	
<! ATTLIST	abstract	jezik	(srpski engleski)	#IMPLIED	
<! ATTLIST	(h0 h1 h2)	id	ID	#IMPLIED	
		naslov	CDATA	#IMPLIED	
<! ATTLIST	(h3 h4)	id	ID	#IMPLIED	
<! ATTLIST	figer	size_x	NMTOKEN	#REQUIRED	
		size_y	NUTOKEN	#REQUIRED	
<! ATTLIST	tbl	kolone	NUMBERS	#REQUIRED	
<! ATTLIST	(%referentne_recenice)	refid	IDREF	#CONREF	
<! ATTLIST	fnref	strana	(da ne)	da	
<! ATTLIST	liref	refid	IDREF	#REQUIRED	
		refid	IDREF	#REQUIRED	
<! ATTLIST	artwork	strana	(da ne)	da	
		id	ID	#IMPLIED	
		okvir	(uokvirenje)	ne	
		mesto	(vrh fiksna dno)	vrh	
		sirina	(kolona strana)	strana	
		ravnanje	(levo centrirano desno)	centrirano	
<! ATTLIST	fn	id	ID	#IMPLIED	
<! ATTLIST	hdref	refid	IDREF	#REQUIRED	
<! ATTLIST	figref	refid	IDREF	#REQUIRED	

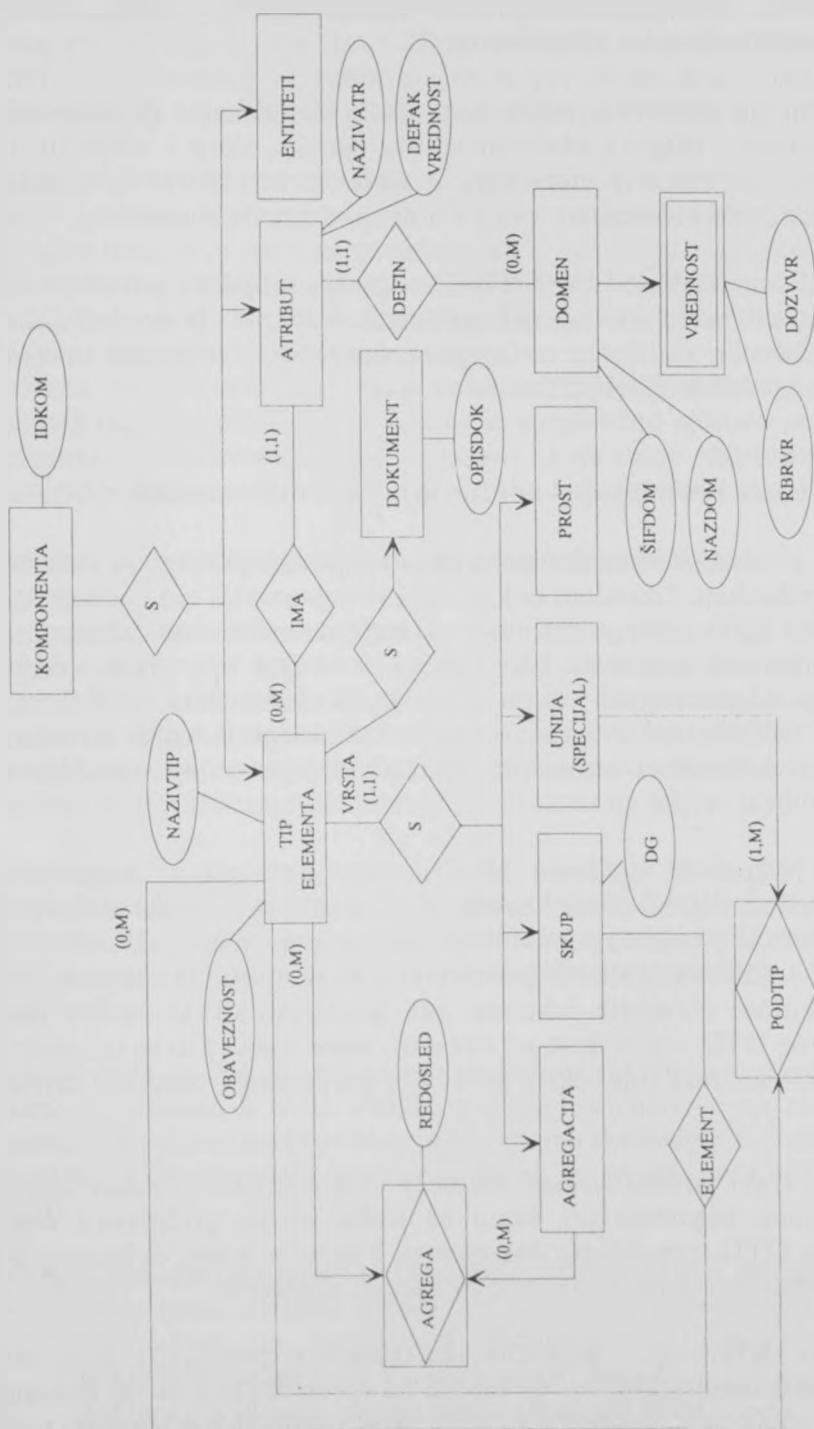
Slika 7.4. DTD za monografije

Za svaku vrstu dokumenta, odgovarajući DTD treba sačuvati u REČNIK-u SPI koji je distribuiran tako da uz svaku lokalnu bazu dokumenata mora da se čuva njen opis. Struktura REČNIKA SPI u kome se pamte DTD-ovi za različite vrste dokumenata može se opisati DOV dijagramom datim na slici 7.5. Može se reći da su osnovni koncepti koji se pojavljuju na ovom dijagramu: tip elementa za označavanje, atribut i entitet.

TIP ELEMENTA za označavanje se na osnovu vrste konektora kojima se definiše redosled pojavljivanja elemenata u okviru sadržaja nekog elementa i identifikatora pojavljivanja po vrsti specijalizuje na:

AGREGACIJU - kada se u okviru sadržaja elementa nalazi grupa elemenata koji su odvojeni konektorom " " što znači da je važan redosled njihovog pojavljivanja (atribut REDOSLED ima vrednost 1) ili konektorom "\$" kada je redosled pojavljivanja nebitan (atribut REDOSLED ima vrednost 0).

SKUP - kada se u okviru sadržaja elementa nalaze elementi uz koje se pojavljuje identifikator pojavljivanja " " ili '+' što znači da su ti elementi ponovljivi. Treba praviti razliku između skupa i elemenata skupa. Treba dati ime i skupu i elementu skupa i prikazati njihovu vezu mada se ova veza može tretirati i kao trivijalna, pa navoditi samo naziv skupa, podrazumevajući naziv njegovih komponenti. Atribut DG daje donju granicu kardinalnosti skupa (DG daje donju granicu kardinalnosti skupa, DG=1 odgovara identifikatoru pojavljivanja " " a DG=0 identifikatoru pojavljivanja " ").



Slika 7.5 DTD Opis dokumenta

UNIJA - SPECIJALIZACIJA - kada se u okviru sadržaja elementa nalaze elementi odvojeni konektorom 'T'.

Uz bilo koji tip elementa može se pojaviti identifikator pojavljivanja kojim se opisuje njegova obaveznost. Agregacija, skup i unija su načini međusobnog povezivanja elemenata u složenije strukture. Za razliku od SGML opisa svaki elemenat, na svakom nivou mora da ima naziv.

ATRIBUT-om TIPA ELEMENTA se opisuju svojstava pojedinih tipova elemenata i on može da ima default vrednost, ili da bude definisan nad nekim domenom, pri čemu su njegove dozvoljene vrednosti prikazane slabim objektom VREDNOST.

7.1.2. Proces Formiranje i održavanje baze dokumenata

Pre unosa u bazu, dokumente treba obraditi prema principima definisanim SGML standardom. Obradom dokumenta se uspostavlja veza između samog dokumenta i DTD-a koji je definisan za identifikovanu vrstu dokumenta. Pri tom se dokument posmatra kao logička struktura tipa grafa u kojoj se unošenjem odgovarajućih tagova i ostalih elemenata SGML jezika (reference entiteta, deklaracije elemenata itd.) mogu izdvojiti svi elementi dokumenta deklarisani na nivou DTD-a. Dokument se obrađuje u tri koraka:

a) Najpre se analizira informaciona struktura i druga svojstva dokumenta kako bi se identifikovao svaki značajan element dokumenta i okarakterisao kao neki od poznatih tipova (zagлавље, paragraf, uređena lista, fusnota itd.). Cilj ove analize je prvenstveno da se utvrdi da li dati dokument sadrži specifične elemente odnosno ima specifičnu strukturu (pa zahteva izradu novog DTD-a u odnosu na već definisane tipove) ili se od utvrđenog tipa dokumenta razlikuje samo po svojoj prezentaciji odnosno layout-u i stilu.

b) Svaki identifikovani element dokumenta se zatim označava odgovarajućim tagovima pri čemu se treba strogo pridržavati unapred definisanog DTD-a za dati tip dokumenta. Tagovi se mogu definisati na više načina: eksplicitno definisani tagovi, pozicioni ili grafički.

Eksplicitno definisanim tagovima se označava početak i kraj svakog elementa dokumenta koji su definisani na nivou DTD-a za tip dokumenta koji se obrađuje (npr. <paragraf> i </paragraf> služe za označavanje početka

i kraja paragrafa). Ovi tagovi imaju formu prirodnog jezika (u ovom slučaju srpskog jezika) pa samim tim i jasno semantičko značenje. Kao takvi su razumljivi za korisnika ali zahtevaju da se pri obradi dokumenta poznaje lista dozvoljenih tagova definisanih odgovarajućim DTD-om.

U grupu tagova čija prezentacija u tekstu nema jasno vidljivo značenje spadaju pozicioni tagovi (npr. početak i kraj paragrafa može da se definise određenim brojem praznih redova) i grafički tagovi (npr. paragraf se definiše korišćenjem različitih ikona, fontova i drugih grafičkih prezentacija). Da bi se olakšalo unošenje grafičkih ili pozicionih tagova u tekst dokumenta mogu se unapred definisati šabloni (template-ovi) za određene vrste dokumenata u koje treba uklopliti sadržaj svakog dokumenta. Pri korišćenju grafičkih i pozicionih tagova treba imati u vidu da je neophodno njihovo prevođenje u odgovarajuće eksplizitno definisane tagove koje može raspozнати program za parsiranje dokumenta.

U okviru ovog projekta koriste se eksplizitno definisani tagovi.

c) Na bazi stila u kojem je dokument pisan mogu se definisati procedure za obradu, kojima se može proizvesti željeni format za dati tip elementa ili predvideti neka druga vrsta obrade dokumenta. Ovaj korak nije obavezan i njega treba definisati samo kada je neophodno dokument pripremiti za publikovanje, pri čemu treba poštovati neke unapred utvrđene procedure za formatiranje dokumenta (dovođene na željeni layout).

Podrazumeva se da dokument koji se obrađuje bude na nekom od magnetnih medija, obzirom da se unošenje tagova (eksplizitno definisanih, pozicionih ili grafičkih) u sam tekst dokumenta vrši preko nekog standardnog programa za obradu teksta ili editora sa ugrađenim elementima SGML jezika.

Za slučaj kada se dokumenti nalaze na papiru potrebno je papirnati oblik dokumenta prevesti u oblik identičan obliku dokumenta na elektronskim magnetnim medijumima (korišćenjem postupka skaniranja dokumenta i S/W za optičko prepoznavanje karaktera-OCR), pa ga na identičan način obraditi.

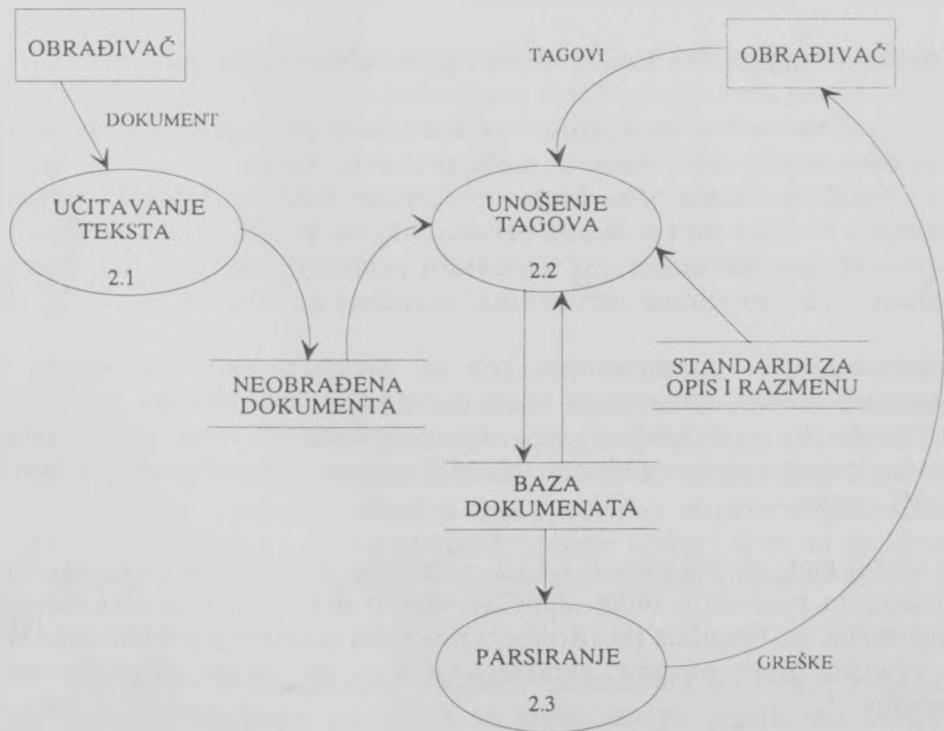
Pri obradi dokumenata mogu se generalno razlikovati dva slučaja:

- Kada treba obraditi dokumente koji su već publikovani (bilo da se oni nalaze na magnetnom mediju ili papiru) za obradu dokumenta se moraju koristiti editori kroz koje je moguće u tekst dokumenta ubaciti odgovarajuće elemente SGML jezika.

- Kada je dokument generisan korišćenjem standardnih tekst procesora koji već imaju ugrađene mogućnosti za unošenje elemenata SGML jezika i proveru njihove sintaksne ispravnosti (WP 6.0, Word for Windows 6.0 ili viši, neki samostalno razvijen word processor sa navedenim mogućnostima) pa nije potrebna nikakva dodatna obrada dokumenta.

DTP dijagram kojim se može opisati proces za formiranje i održavanje baze dokumenata koje treba označiti prema usvojenom standardu za obradu i razmenu dokumenata dat je na slici 7.6.

Proces Učitavanje teksta podrazumeva prihvatanje dokumenta u ASCII obliku nastalog na bilo kojoj lokaciji SNTIS-a koji u principu može biti generisan u bilo kom softveru za obradu teksta.



Slika 7.6. Formiranje i održavanje baze dokumenata

Proces Unošenje tagova predstavlja obradu dokumenta prema napred usvojenom DTD-u za određenu vrstu dokumenta koji je zapamćen u bazi standarda.

Proces Parsiranje se vrši nad označenim dokumentom da bi se proverilo da li je označavanje dokumenta izvršeno u skladu sa odgovarajućim DTD-om. Sintaksna ispravnost SGML dokumenta se ne vrši u trenutku unošenja elemenata u tekst dokumenta, već se ona kontroliše tek pošto se označavanje elemenata u potpunosti završi. Parser treba da iz sadržaja SGML dokumenta prepozna četiri identifikovane vrste elemenata za označavanje i da ih interpretira na odgovarajući način. Korišćenjem pravila SGML jezika parser treba da:

- Skenira tekst kako bi u sadržaju dokumenta prepoznao elemenate za označavanje, izdvojio ih od samog teksta dokumenta i identifikovao kojoj od četiri navedene grupe pripada.
- Zamenjuje referencu nekog entiteta samim entitetom.
- Interpretira deklaracije elemenata.
- Prenosi kontrolu na odgovarajući sistem za obradu koji izvršava procedure pridružene generičkim identifikatorima odnosno instrukcije za obradu.
- Interpretira tagove i prepoznaje odgovarajuće generičke identifikatore i njihove atribute i poštujući pravila definisana DTD-om proverava da li je svaki od GI-a i atributa validan i da li je korektna struktura dokumenta.

7.1.3. Proces Priprema za indeksiranje

U sistemu za formiranje i održavanje baze dokumenata moguće su dve vrste pretraživanja:

- Invertovano - preko definisanih indeksa, koje kao odgovor vraća isključivo identifikatore dokumenta koji zadovoljavaju upit na bazi korišćenja jedinstvenog jezika za pretraživanje;
- Direktno - koje se vrši nad detaljno struktuiranom bazom SGML dokumenata.

Pre nego što se SGML dokument iz baze dokumenata indeksira, on prolazi kroz proces Priprema za indeksiranje čija je prvenstvena funkcija da "poravna" strukturu dokumenata. "Ravnanje" podrazumeva transformaciju složene, hijerarhičke strukture SGML dokumenta u skup jednodimenzionalnih polja (koncepta), prepoznatljivih za proces indeksiranja. Pri tom se koristi činjenica da se svaki SGML dokument može predstaviti struktrom stabla (acikličnog grafa), tako da samo listovi stabla imaju svoj sadržaj (tekst). Da bi se složena hijerarhička struktura SGML dokumenta svela na skup jednodimenzionalnih struktura neophodno je uvesti sledeće pojmove:

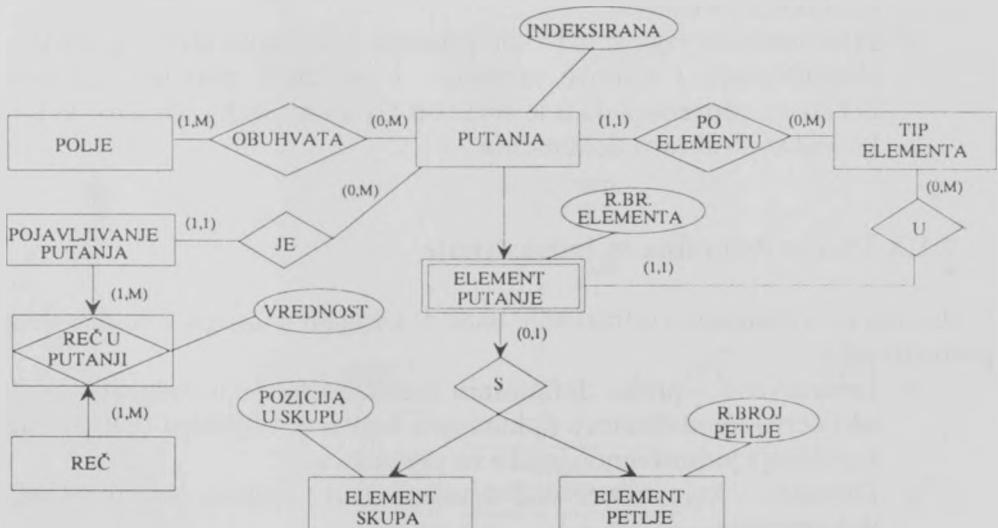
PUTANJA je definisani način "kretanja" kroz SGML strukturu između dva elementa. Putanje kojima se od vrha SGML dokumenta može stići do nekog prostog elmenta (lista stabla) zvace se VREDNOSNA PUTANJA.

POLJE je bilo koja putanja po kojoj se od vrha SGML opisa može da dođe do neke komponente dokumenta. Pod komponentom se podrazumeva elemenat SGML opisa ili atribut nekog elementa.

TIP ELEMENTA je jedna grana stabla ili grafa koji čine elementi SGML dokumenta i može biti različitog tipa: agregacija, skup ili unija.

REČ je deo sadržaja elementa dokumenta za koji je definisana vrednosna putanja.

DOV koji prikazuje vezu između SGML opisa i opisa strukture za pretraživanje dat je na slici 7.7.



Slika 7.7. DOV Putanje

Za invertovano pretraživanje je neophodno definisati polja po kojima će se izvršiti indeksiranje i pretraživanje u skladu sa usvojenim jezikom, baziranim na DIALOG jeziku. Kako se u DIALOG jeziku definišu bazni i dodatni indeksi, ovde je važno istaći da je neophodno indeksirati sva dodatna polja, a pretraživanje po baznom indeksu se vrši preprocesiranjem ovoga upita na upit po dodatnim indeksima.

Dodatni indeks je polje koje ima vrednost i koje se indeksira. Vrednost (sadržaj) polja je unija vrednosti (sadržaja) prostih elemenata i atributa do

kojih se od posmatranog elementa može doći. Unija elemenata se može izvršiti samo nad kompatibilnim elementima, elementima koji su definisani nad istim domenom (na primer tekst).

Bazni indeks je polje od koga se može definisati putanja do dodatnih indeksa, odnosno koje se može izvesti iz dodatnih indeksa. (Može, za razliku od DIALOGA postojati više baznih indeksa).

Na osnovu predhodnog opisa procesa Priprema za indeksiranje, može se reći da je njegova uloga pored "ravnjanja" strukture SGML dokumenta i dobijanje sadržaja (vrednosti) dodatnih indeksa. Pri izboru elemenata koji se mogu tretirati kao dodatni indeksi, prihvaćen je kriterijum da to budu elementi koji se najčešće javljaju u okviru teksta dokumenta bez obzira na njihovu vrstu. Primer mogućih dodatnih indeksa za pretraživanje SGML dokumenta dat je u tabeli na slici 7.8. Kao bazni indeksi mogu se tretirati elementi koji u svojoj strukturi sadrže elemente definisane kao dodatni indeksi kao što su npr: poglavlje drugog nivoa (h1), poglavlje prvog nivoa (h0) itd.

tag	opis elementa
keyw	ključna reč
title	naslov
author	autor
inst	institucija
abstract	abstract
p	paragraf
note	napomena
h0t	naslov poglavlja prvog nivoa
h1t	naslov poglavlja drugog nivoa
h2t	naslov poglavlja trećeg nivoa
q	citat
hp1	naglasena rečenica(italic)
hp2	naglasena rečenica(bold)
hp3	naglasena rečenica(bold_ital.)
hp0	naglasena rečenica(normal)
tbl	tabele
ol	uređena lista
ul	neuređena lista
gl	rečnik termina
figcap	naslov crteža
fn	fusnote

Slika 7.8. Tabela mogućih dodatnih indeksa za pretraživanje

7.1.4. Proces Direktno pretraživanje

Direktno pretraživanje se javlja u slučajevima kada bazu dokumenata treba direktno pretraživati a da se ona predhodno ne podvrgava postupku indeksiranja ili kada se na osnovu dobijenih odgovora iz invertovanog pretraživanja može formirati baza dokumenata, pa zatim ovaj skup direktno pretraživati. Za razliku od konvencionalnih sistema za pretraživanje informacija koji dokumente tretiraju kao skup linearnih struktura organizovanih u nezavisne kolekcije, direktno pretraživanje treba da omogući pretraživanje dokumenata po njihovoј hijerarhiskoj strukturi koja je opisana odgovarajućim DTD-om. Hijerarkijska struktura dokumenta se može grafički prikazati stablom, ili cikličnim grafom gde je svaki čvor stabla jedan element dokumenta, koji može da sadrži sam tekst dokumenta, ukoliko se radi o listu stabla, ili može da sadrži druge elemente.

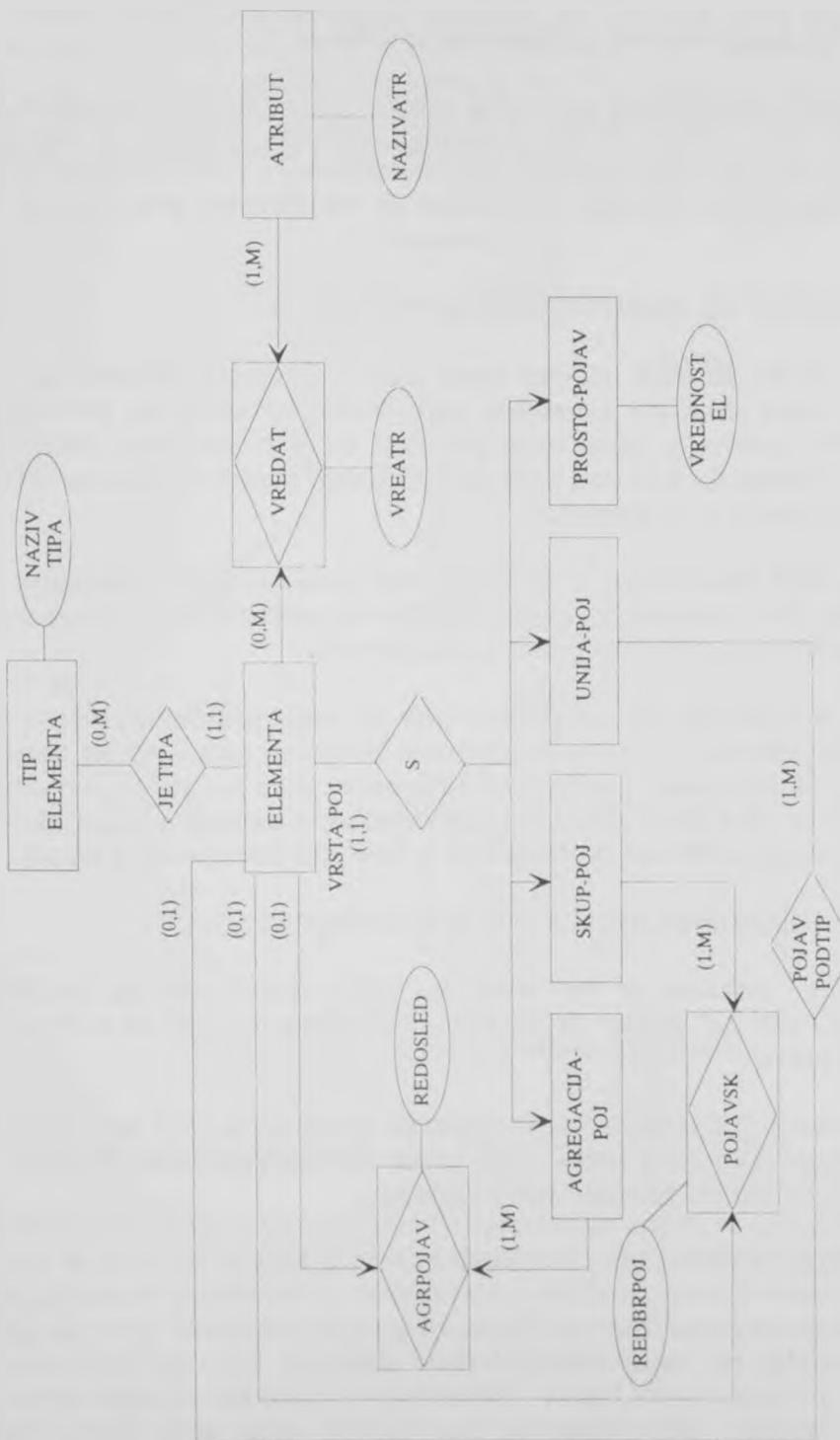
Direktno pretraživanje se u osnovi može realizovati na dva načina:

- tako što bi se baza strukturiranih dokumenata realizovala kao relaciona baza podataka i pretraživanje nad njom realizovalo preko standardnih upitnih jezika,
- preko specijalnog jezika za direktno pretraživanje.

7.1.4.1. Direktno pretraživanje preko standardnih upitnih jezika

U slučaju da se SGML dokumenti pamte u relacionoj bazi podataka, struktura te baze bi mogla da se opiše DOV dijagramom na slici 7.9. U tom slučaju proceduru za strukturiranje dokumenata prema dijagramu 7.9 treba ugraditi u odgovarajuće procese za formiranje baze dokumenata. Ako se SGML dokumenti samo direktno pretražuju, što je pogodna mogućnost pretraživanja dokumenata na lokalnom nivou, ovu proceduru treba ugraditi u proces Formiranje baze dokumenata. U slučaju da se direktno pretraživanje primenjuje nad dokumentima koji su dobijeni kao izlaz iz invertovanog pretraživanja, taj algoritam treba ugraditi u proces za pamćenje dokumenata koji su dobijeni kao rezultat invertovanog pretraživanja.

Rezultat pretraživanja u invertovanom pretraživanju je uvek ceo dokument dok direktno pretraživanje treba da omogući da se izabrani dokumenti strukturiraju na zadat način, pa samim tim direktno pretraživanje treba da omogući dobijanje širokog skupa različitih izveštaja iz baze dokumenata.



Slika 7.9 Baza podataka dokumenta sa definisanim D(D)

7.1.4.2. Specijalni jezik za direktno pretraživanje

Jezik kojim bi strukturirani dokument mogao da se direktno pretražuje bi morao da bude sličan nenormalizovanim upitnim jezicima ili upitnim jezicima za objektne baze podataka koji bi istovremeno imao karakteristike standardnog SQL-a. Sintaksa upita kojim se vrši direktno pretraživanje bi bila:

```
SELECT cilj_upita FROM opseg WHERE uslov
```

Cilj_upita može da bude sadržaj samo jednog elementa dokumenta (list stabla) ili neka struktura elemenata koju želimo da dobijemo po SGML sintaksi. Pre izvršenja upita treba proveriti da li je navedena struktura moguća, odnosno da li se ona slaže sa strukturom koja je definisana DTD-om za određenu vrstu dokumenata.

Opseg definiše promenljive po kojima se vrši pretraživanje. Promenljiva se definiše kao čvor strukture a podrazumeva se da sadrži sve ostale čvorove do kojih se iz nje može stići (sve čvorove nižeg nivoa).

U uslovu se mogu navesti sve putanje koje se mogu generisati polazeći od čvorova definisanih u opsegu, kako prema listovima tako i prema korenu stabla koje je definisano za datu vrstu dokumenta. Ako je u uslovu navedena neka putanja višeg nivoa (prema korenu stabla), u parsiranju upita ona se izvodi kao unija vrednosnih putanja koje je čine (sve putanje nižeg nivoa).

Umesto detaljnog opisa jezika, navodi se sledeći primer:

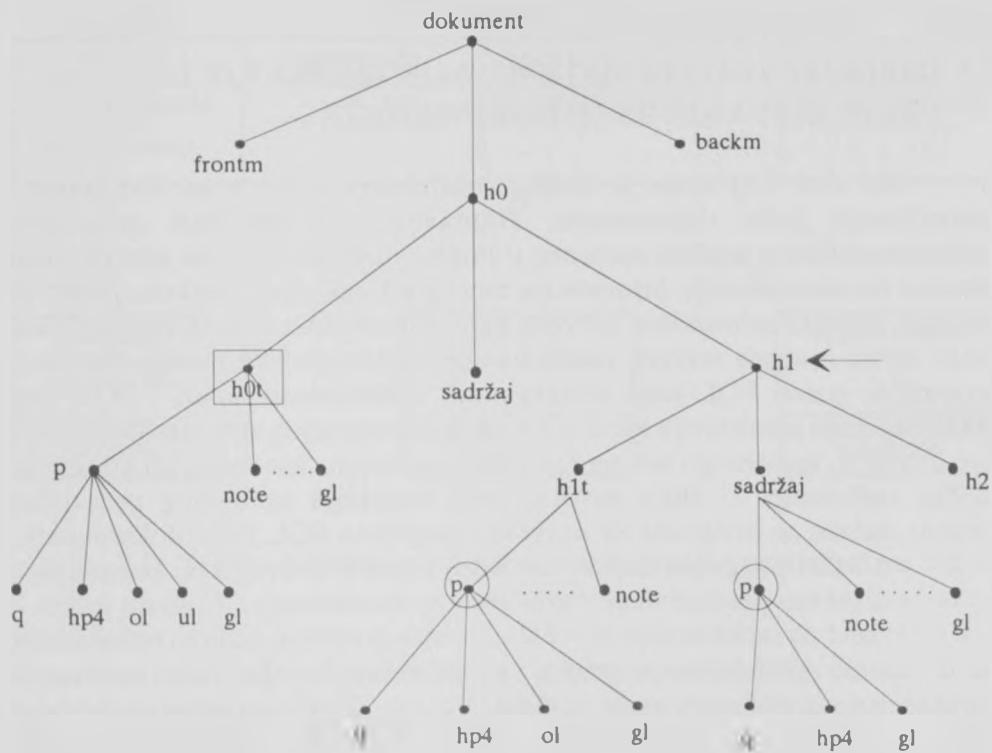
Na slici 7.10 prikazan je deo strukture stabla dokumenta čiji je DTD prikazan na slici 7.4. Zahtev za direktno pretraživanje može da se definiše na sledeći način:

Nađi sve sadržaje paragrafa (p) iz poglavlja prvog nivoa (h1) tako da se u naslovu poglavlja nultog nivoa (h0t) nalazi reč 'fulltext baza'. Na osnovu predhodno definisane sintakse upit bi izgledao:

```
SELECT p FROM h1 WHERE 'fulltext' IN h0t
```

Cilj upita je definisan kao struktura čiji je koren element 'p' - paragraf odnosno treba prikazati sadržaje svih elmenata koji su podelementi elementa 'p' (elementi 'q', 'hp4', 'ol', 'ul' itd). Kako je kao opseg ovog upita definisan element 'h1' - znači da kao rezultat upita treba dobiti samo

elemente 'p' koji su podelementi elementa 'h1'. Pri tom treba uzet u obzir sva pojavljivanja elementa 'p', u okviru elementa 'h1t' i elementa 'sadržaj'. Uslov pretraživanja se odnosi na element 'h0t' što znači da prisustvo zadate reči 'fulltext' treba proveri u sadržajima svih njegovih podelementa koji se nalaze na listovima stabla ('q', 'hp4', 'ol', 'ul' itd.)



Slika 7.10. Deo strukture stabla dokumenta

7.1.5. Proces Publikovanje

Publikovanje (izdavanje) dokumenata je funkcija kojom se ostvaruje reprint celog ili dela dokumenta koji se nalazi u bazi dokumenata. Ova funkcija je integrisana sa sistemom za invertovano i direktno pretraživanje kojima se definiše skup (podskup) dokumenata koji se publikuje i struktura samog izveštaja. Osim toga, rezultat pretraživanja koji se dobija kao SGML opisana struktura se može transformisati u neki drugi jezik za opis punog teksta dokumenta (npr. HTML), ili dopuniti nekim elementima (npr.

instrukcijama za obradu) tako da se dokument može publikovati na razne načine.

Izdavanje dokumenata podrazumeva štampanje, prikaz na ekranu, pamćenje na određenim medijumima, elektronski prenos na druge lokacije i sl.

7.2. IMPLEMENTACIJA SISTEMA ZA FORMIRANJE I PRETRAŽIVANJE BAZE DOKUMENATA

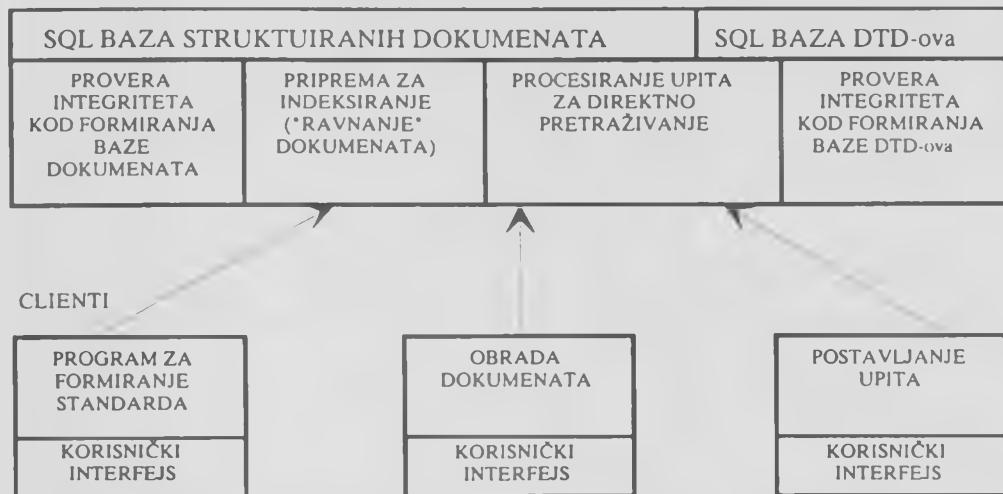
Na slici 7.11 data je fizička arhitektura sistema za formiranje i pretraživanje baze dokumenata. Napravljena je na bazi generičkog implementacionog modela opisanog u Poglavlju 4 i u skladu sa arhitekturom Sistema za pretraživanje informacija datom u Poglavlju 5. Slikom je ustvari detaljno opisana arhitektura Servera baze dokumenata koji se spominje kao jedan od tri moguca servera Sistema za pretraživanje informacija. Na ovom serveru se pored SQL baze strukturiranih dokumenata nalazi i SQL baza DTD-ova koja predstavlja lokalni rečnik podataka ovog sistema. U principu, ove dve SQL baze mogu biti na razlicitim serverima, što zavisi od konkretnе fizičke realizacije. U sloju Aplikacionog interfejsa usvojenog generičkog modela nalaze se programi za proveru integriteta SQL baze dokumenata i DTD-ova kao i programi koji se izvršavaju nad ove dve baze: procesiranje upita za direktno pretraživanje i priprema za indeksiranje. Program kojim se vrši priprema za indeksiranje može da formira posebnu, stalnu (redundantnu) bazu 'ravnih' dokumenata odakle program za indeksiranje sa Servera indeksa uzima polje po polje i indeksira ga. Druga mogucnost je da se program kojim se vrši priprema za indeksiranje izvrsava na zahtev programa za indeksiranje i u direktnoj međusobnoj komunikaciji dostavlja na indeksiranje polje po polje jednog dokumenta.

Kao klijenti Servera baze dokumenata (sloj 2 - Jezgro aplikacije u usvojenom generičkom modelu opisanom u Poglavlju 4) javljaju se: program za formiranje standarda (SQL baza DTD-ova), program za obradu dokumenta i postavljanje upita za direktno pretraživanje nad SQL bazom strukturiranih dokumenata.

SQL baza strukturiranih dokumenata i SQL baza DTD-ova je realizovana kao ORACLE baza. Vecina realizovanih programa iz sloja Aplikacionog interfejsa je realizovana u PRO-C programskom jeziku, koji s jedne strane pruža komotnost u radu svojstven C-u a s druge strane lak pristup ORACLE bazi i korišćenje PL/I blokova. Programi iz sloja 2 - Jezgra aplikacije su

realizovani pod Windows operativnim sistemom pa kako je neophodno da ovi programi komuniciraju sa SQL bazom, za njihovo povezivanje sa ORACLE-om su korišćeni ODBC alati.

SERVER BAZE DOKUMENATA



Slika 7.11. Fizička arhitektura sistema za formiranje i pretraživanje baze dokumenata

Za većinu navedenih programa, koji se pojavljuju u slojevima Aplikacionog interfejsa ili Jezgru aplikacije su urađeni prototipovi koji su opisani u sledećim poglavljima.

7.2.1. Programi za formiranje standarda za opis i razmenu

Osnovna funkcija ovog softvera je da se u SQL bazu DTD-ova unesu odgovarajući DTD-ovi. Struktura ove baze se dobija transformacijom MOV dijagrama sa slike 7.5 u relacioni model a zatim je dobijeni relacioni model (kanonički model) modifikovan tako da bude pogodan za skup aplikacija koje se nad njim rade. Transformacija je izvršena tako da kanonički relacioni model bude pogodan za izvršenje programa koji vrši pripremu za indeksiranje koji će biti objašnjen kasnije. Struktura baze DTD-ova se može opisati sledećim tabelama:

Tabela ELEMENTI koja sadrži sve elemente koji se mogu pojaviti na nivou jedne vrste dokumenta. Opisuje se sledećim atributima:

VR_DOK	sadrži oznaku vrste dokumenta
TAG	sadrži oznaku(tag) elemenata koji su mogući u okviru date vrste dokumenata
OPIS	sadrži tekstualni opis elementa
PTAG	sadrži vrednosti 'O' ili - što određuje da li se obavezno unosi ili ne unosi početni tag elementa
KTAG	sadrži vrednosti 'O' ili - što određuje da li se obavezno unosi ili ne unosi krajnji tag elementa
KONEKTOR	sadrži konektor kojim je dati element odvojen od sebi susednog elementa

Tabela ATRIBUTI sadrži opise atributa koji se mogu pojaviti na nivou elemenata definisanih za određenu vrstu dokumenta. Opisuje se atributima:

VR_DOK	sadrži oznaku vrste dokumenta
TAG	sadrži oznaku(tag) elemenata na nivou kojih se mogu pojaviti neki atributi
DIFAUULT_V	sadrži defaulte vrednosti atributa (IMPLIED, REQUIRED itd.)
VREDNOST	sadrži deklarisanu vrednost atributa

Tabela ENTITET sadrži sve entitete koji se mogu pojaviti na nivou jedne vrste dokumenta. Opisuje se sledećim atributima:

VR_DOK	sadrži oznaku vrste dokumenta
TAG	sadrži oznaku entiteta koji može da se pojavi u strukturi SGML dokumenta
OZ_DOK	sadrži ime entiteta
KONEKTOR	sadrži konektor kojim je dati entitet odvojen od sebi susednog elementa ili entiteta

Tabela ELEMENTI_STR sadrži opis hijerarhische strukture dokumenta određene vrste. Opisana je atributima:

VR_DOK	sadrži oznaku dokumenta
ELEMENT	sadrži oznaku(tag) elementa koji je definisan na nivou date vrste dokumenta. Element može da ima svoju strukturu (elemente ili entitete nižeg nivoa - podelemente) ili da bude list stabla kada je sadržaj elementa npr. tekst

POD_ELEMENT	sadrži oznake(tagove) elementa ili entiteta nižeg nivoa (podelemenata) ili vrednost PCDATA ako je dati element list stabla
ENTITY	sadrži oznaku '%' ako je podelement entitet
INDIKATOR	sadrži indikator pojavljivanja za dati element ('?', '+' ili '*')
RED	sadrži broj koji određuje redosled pojavljivanja podelemenata u okviru datog elementa

Tabela ENTITET_STR sadrži opis strukture entiteta. Opisana je atributima:

ENTITET	sadrži oznaku entiteta
POD_ELEMENT	sadrži oznake(tagove) elementa ili entiteta nižeg nivoa (podelemenata)
ENTITY	sadrži oznaku '%' ako je podelement entitet
RED	sadrži broj koji određuje redosled pojavljivanja podelemenata u okviru datog elementa

Program za unošenje baze standarda je realizovana pod Windows-om (Visul C). Kroz prvu formu koja je obuhvaćena ovom aplikacijom unose se svi elementi koji se mogu pojaviti na nivou jedne vrste dokumenta sa odgovarajućim atributima. Kroz drugu formu se unosi struktura dokumenta tako što se za svaki element unose oznake elemenata (tagovi) koji ulaze u strukturu tog elementa. Pri tom se kontroliše da li je uneta oznaka elementa (podelementa) moguća u okviru vrste dokumenta čija se struktura definiše.

7.2.2. Programi za obradu dokumenata

Program za obradu dokumenata kojim se učitava tekst dokumenta i vrši unošenje tagova je realizovana pomoću Word 6.0 za Windows i po svojoj funkciji se može tretirati kao editor preko kojeg se u tekst dokumenta mogu unositi elementi SGML jezika. Pri ovakvoj obradi dokumenta je potrebno znati oznake (tagove) elemenata definisane na nivou DTD-a, imati informaciju o mogućnosti izostavljanja krajnjih tagova kao i o strukturi u okviru koje se određeni element može naći, što se nalazi u SQL bazi DTD-ova. Iz tih razloga je bilo neophodno ovu WINDOWS aplikaciju preko odgovarajućeg ODBC alata povezati sa ORACLE relacionom bazom po klient-server konceptu.

Deo sadržaja jednog obradenog dokumenta po SGML standardu dat je na slici 7.12.

```
<h0>
    <h0t>5.3. Komunikacija između entiteta
<h1>
    <h1t>5.3.1. Definicije
    <gl>
        <gt>5.3.1.1. (N) veza: <gd> Između (N) slojeva dva ili više (N+1) entiteta uspostavlja se veza radi transfera podataka
        <gt>5.3.1.2. (N) veza i tačka: <gd> terminator na jednom kraju (N) veze u okviru (N) servisa-preko-tačke
        <gt>5.3.1.5. (N) izvor podataka:<gd> (N) entitet koji šalje (N-1) uslužnih jedinica podataka<hhref refid=5.6.1.7> na N-1 vezi
            <fnref refid=2>
                <gt>5.3.1.6. (N) prenos podataka:<gd> (N) sredstvo koje prenosi (N) uslužnih jedinica podataka sa jednog ili više (N+1) entiteta
            </gl>
<h1>
    <h1t>5.3.2. Opis
<p>Za informacije koje se prenose između dva ili više (N+1) entiteta. Veza Veza će biti uspostavljena između njih na (N) nivou korišćenjem (N) protokola.
<note>NOTE - Protokoli mogu biti definisani u okviru.....
<p>Ova asocijacija se zove (N) veza. (N) veze obezbeđuje (N) nivo između dve ili više (N) pristupnih tačaka. Terminator kraja (N) veza
<p>(N+1) entiteti mogu komunicirati samo korišćenjem servisa (N) nivoa. Ima Primana gde servisi koje obezbeđuje (N) nivo ne dozvoljavaju direktni pristup između svih (N+1) entiteta....
```

Slika 7.12. Deo obrađenog dokumenta

7.2.3. Program kojim se vrši priprema za indeksiranje

Ovaj program prolazom kroz strukturu dokumenta definisanu DTD-om (struktura grafa) registruje završne elemente (listove stabla) i za vrenosne putanje formirane na taj način analizira da li za njih postoje odgovarajući tekstualni sadržaji u konkretnom dokumentu obrađenom SGML jezikom koji se nalazi u SQL bazi struktuiranih dokumenata. Analiza baze koja sadrži SGML opise dokumenata se sastoji u tome da se na osnovu vrednosne putanje dobijene analizom DTD-a pronađe sadržaj odnosno korisna informacija koja predstavlja tekst dokumenta. Ovako izdvojene vrednosne putanje sa odgovarajućim sadržajem se mogu prikazati, smestiti u datoteku ili u odgovarajuće ORACLE tabele.

Program za realizaciju ovog procesa je rađen u više faza.

Prva faza je analiza DTD strukture i registrovanje svih vrednosnih putanja. Struktura tabela koja omogućava čuvanje DTD-ova nameće rekurziju kao optimalan metod za analizu i to kroz dva potprograma od kojih jedan prati elemente a drugi entitete, pri čemu ovi potprogrami međusobno komuniciraju pošto se strukture elemenata i entiteta međusobno pozivaju i preliču.

U drugoj fazi za ovako izdvojene vrednosne putanje je bilo potrebno izvršiti analizu konkretnog dokumenta tako što se tagovi registrovanih krajnjih elemenata traže u tekstu i ukoliko ih ima upisuju se u pomoćnu datoteku u kojoj se čuva cela vrednosna putanja i sadržaj koji je izdvojen na ovaj način. Problemi koji su se javili u ovoj fazi su se odnosili na:

- ponovljivost određenih struktura (jedna struktura u tekstu se može javiti više puta npr. paragraf u okviru poglavlja itd.),
- mogućnost da se iste strukture mogu javiti u različitim višim strukturama - kontekstima (na primer paragraf u okviru abstrakta ili paragraf u okviru poglavlja prvog nivoa, drugog nivoa itd.).

Ponovljivost podrazumeva da se ista struktura vrti više puta sve dok u dokumentu postoje elementi koji pripadaju toj strukturi pod uslovom da je ova struktura prethodno registrovana kao ponovljiva. S obzirom da je celokupna analiza strukture realizovana kroz rekurziju i da bi ponovno puštanje neke strukture poremetilo rekurziju neophodno je bilo da se mehanizmom magacina (stack) pamte ponovljivi elementi i njihove dubine pojavljivanja a zatim kada dubine u rekurziji padnu ispod zabeleženih ponovljivih elemenata pozivaju se potprogrami koji imaju za cilj da obrade samo ponovljivost tog elementa odnosno da od njega ponovo puste strukturu. Magacin je iskorišćen zbog toga što ponovljivi elementi na najvišem nivou treba da idu u ponovljivost tek pošto se ponovljivost elemenata na nižim nivoima potpuno završi.

Da bi se razrešio problem pojavljivanja elementa u različitim kontekstima, izdvajan je deo teksta koji sadrži zadati element u određenom kontekstu i on je posebno analiziran. Na taj način je sprečeno da se određeni elementi traže u okviru struktura kojima ne pripadaju.

Ovako analizirane strukture različitih dokumenata omogućuju lako indeksiranje i pretraživanje sadržaja odgovarajućih elemenata tj. strukture i informacije dobijene na ovaj način tek mogu ići u dalje procese obrade odnosno mogu da budu ulaz u nove korake neke globalne analize.

LITERATURA

- [1] Razvoj softvera za full text baze SNTIS, Univerzitet u Nišu JUNIS, 1995 (radni tekst projekta).
- [2] Information Processing - Text and office systems - Standard Generalized Markup Language (SGML), First edition 1986.

- [3] Information processing - SGML support facilites - Techniques for using SGML, First edition 1988.
- [4] Information processing - Text and office systems - Office Document Architecture (ODA) and interchange format - Part 1: Intraduction and general principles, First edition 1989.
- [5] Christophides V., Abiteboul S., Cluet S. and Scholl M., From Structured Documents to Novel Query Facilites.
- [6] Mariano P. and Tova Milo, Optimizing Queries on Files.
- [7] Macleod I.A., A Query Languange for Retrieving Information from Hierarchic Text Strucures.
- [8] Kim W., Introduction to Object-Oriented Databases, The MIT Press Cambridge, 1990.
- [9] Korunović D., Cvetanović S. i Jovev Lj., *Servisi za obradu i pretraživanje dokumenata sa punim sadržajem*, YUINFO, Brezovica 2-5 april 1995.
- [10] Cvetanović S., i Maksimović M., *Primena standarda u obradi dokumenata sa punim sadržajem*, YUINFO, Brezovica 2-5 april 1995.

Poglavlje 8.

BIBLIOTEČKI INFORMACIONI SISTEM

U projektu [1] predloženo je da se razvije više logičkih mreža za potrebe Sistema Naučnih i Tehnoloških Informacija Srbije (SNTIS). Jedna od tih mreža je i Bibliotečka mreža (BM). Koncepcija BM SNTIS data je u projektu [2].

Bibliotečka mreža, kao infrastrukturni deo SNTIS-a, jeste sistem međusobno povezanih bibliotečko informacionih i dokumentacionih institucija sa značajnim fondovima naučne i stručne literature koje, procesom uzajamnog rada, a uz podršku informacionog servisa i izgrađene komunikacione mreže, obezbeđuje korisnicima, na neposredan, brz i lak način bibliotečke informacije i dokumente [2].

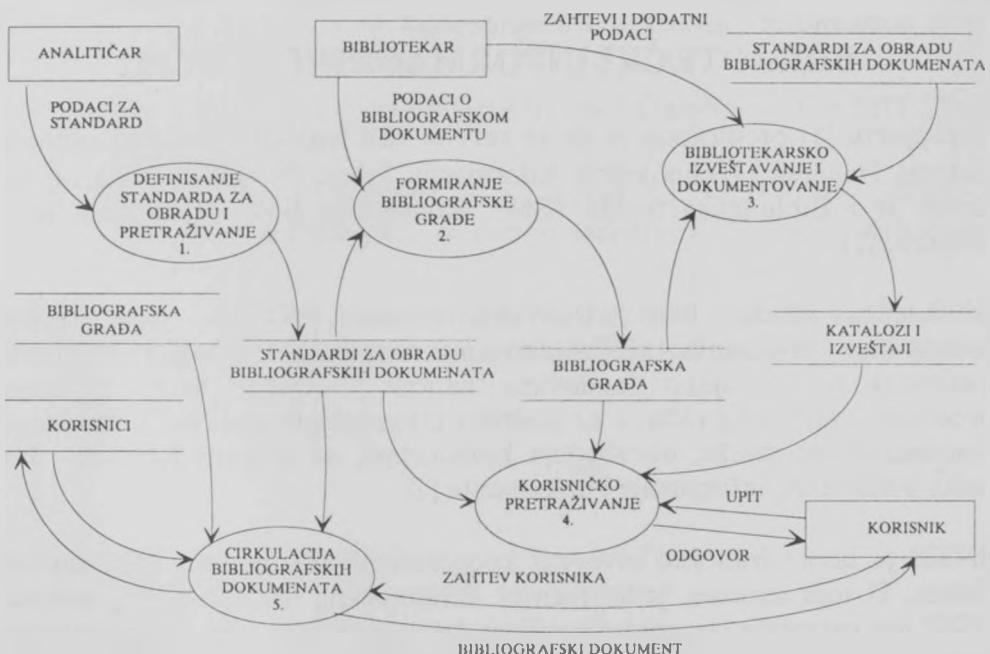
SNTIS je koncipiran kao otvoreni kooperativni distribuirani informacioni sistem. U tom sistemu, jedinstvenost Bibliotečkog informacionog sistema (BIS) ne ostvaruje se centralizacijom baza podataka već kooperativnim radom različitih autonomnih jedinica na osnovu definisanih standarda i jedinstvenog sistema za pretraživanje SNTIS-a zasnovanog na opštem modelu baze podataka. Rad BIS-a u lokalnu baziran je na standardima koji se definišu u rečniku lokalne baze podataka.

Lokalni segment BIS-a je razvijen na Institutu za matematiku, Prirodno-matematičkog fakulteta u Novom Sadu. Za indeksiranje i pretraživanje bibliografske građe, korišćen je jedinstveni sistem za indeksiranje i pretraživanje dokumenata, opisan u Poglavlju 6.

U cilju lakšeg praćenja teksta u ovom poglavlju definišu se pojmovi: *dokument, zapis i bibliografska građa*. Pod dokumentom se podrazumeva izvorna bibliografska jedinica (knjiga, časopis, itd.). Zapis je skup podataka o dokumentu. Bibliografska građa je skup zapisa.

8.1. FUNKCIJE BIBLIOTEČKOG INFORMACIONOG SISTEMA

Na Slici 8.1 prikazan je prvi nivo DTP BIS-a. Pored osnovnih funkcija (FORMIRANJE BIBLIOGRAFSKE GRAĐE, BIBLIOTEKARSKO IZVEŠTAVANJE I DOKUMENTOVANJE, KORISNIČKO PRETRAŽIVANJE i CIRKULACIJA BIBLIOGRAFSKIH DOKUMENATA), BIS obuhvata i funkciju DEFINISANJE STANDARDA ZA OBRADU I PRETRAŽIVANJE.



Slika 8.1. Funkcije Bibliotečkog informacionog sistema

Osnovu za strukturiranje podataka o bibliografskoj građi, obradu bibliografskih dokumenata i pretraživanje zapisa, čine standardi koje definiše funkcija DEFINISANJE STANDARDA ZA OBRADU I PRETRAŽIVANJE. Ova funkcija treba da omogući opis strukture pojedinih standarda i njihovo predstavljanje u bazi podataka preko kojih će se obavljati osnovne funkcije BIS-a. Na Slici 8.2 prikazan je DTP za funkciju DEFINISANJE STANDARDA ZA OBRADU I PRETRAŽIVANJE. Posebno se definisu standardi za održavanje bibliografske grde i strukturiranje zapisa, radno okruženje bibliotekara i formati kataloga i izveštaja. Kao osnova za razmenu podataka i strukturiranje baze

bibliografske građe u okviru BIS-a usvojen je UNIMARC format. Proces DEFINISANJE STANDARDA ZA ODRŽAVANJE (UNIMARC) bibliografske građe treba da omogući opis strukture UNIMARC formata i njegovo predstavljanje u bazi podataka kako bi se obezbedio mehanizam za obavljanje ostalih funkcija sistema u skladu sa usvojenim standardom. Proces DEFINISANJE RADNOG OKRUŽENJA BIBLIOTEKARA treba da obezbedi evidentiranje bibliotekara i da omogući opis različitih modela obrade dokumenata i njihovo predstavljanje u bazi podataka. Na osnovu toga bi se selektovali modeli obrade, koji su usvojeni u bibliografskim institucijama. Proces DEFINISANJE FORMATA KATALOGA I IZVEŠTAJA treba da omogući apstraktni opis strukture proizvoljnog izveštaja i kataloga o dokumentima koji se formiraju iz baze podataka BIS-a, i predstavljanje te strukture u bazi podataka. Na osnovu toga bi se vršilo struktuiranje rezultata pretraživanja, bibliotekarskih izveštaja i kataloga dokumenata kao i odgovora na korisničke upite, koji su rezultat pretraživanja baze podataka BIS-a na osnovu bibliotekarskih zahteva i upita korisnika.



Slika 8.2. Definisanje standarda za obradu i pretraživanje

Funkcija FORMIRANJE BIBLIOGRAFSKE GRAĐE treba da omogući obradu dokumenata i struktuiranje zapisa prema usvojenom standardu, preuzimajući parametre konkretnе obrade na osnovu unapred definisanih modela obrade dokumenata.

BIBLIOTEKARSKO IZVEŠTAVANJE I DOKUMENTOVANJE je funkcija koja treba da omogući formiranje specifičnih bibliotekarskih izveštaja i kataloga o dokumentima. Njihovo formiranje se vrši na osnovu sadržaja skladišta BIBLIOGRAFSKA GRAĐA i apstraktno definisane strukture i formata izveštaja u okviru skladišta STANDARDI ZA OBRADU BIBLIOGRAFSKIH DOKUMENATA.

Funkcija KORISNIČKO PRETRAŽIVANJE treba da obezbedi obradu upita korisnika i struktuiranje odgovora odnosno rezultata pretraživanja, na osnovu sadržaja baze podataka bibliografske građe. Struktuiranje rezultata pretraživanja vrši se na osnovu unapred definisanih struktura, sadržaja i formata prikaza koji su definisani u okviru skladišta STANDARDI ZA OBRADU BIBLIOGRAFSKIH DOKUMENATA.

Funkcija CIRKULACIJA BIBLIOGRAFSKIH DOKUMENATA treba da omogući evidentiranje korisnika biblioteke, kao i praćenje korišćenja fonda biblioteke, rezervisanje i zaduživanje dokumenata. Zbog toga je, pored baze podataka o bibliografskoj građi, potrebno obezbediti i bazu podataka o korisnicima biblioteke.

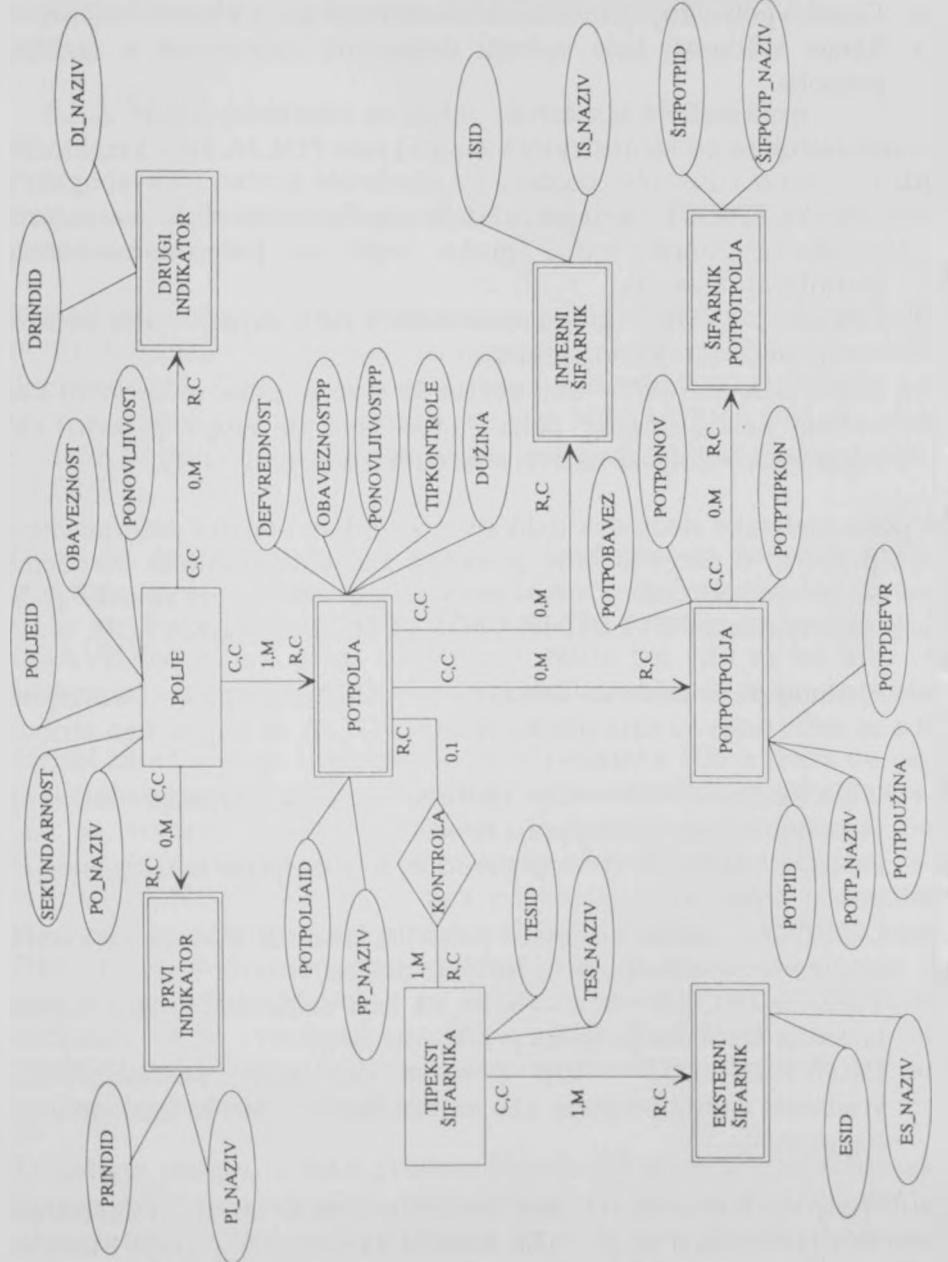
Održavanje baze podataka standarda za obradu dokumenata vrši se centralizovano, a obavlja ga lice sa posebnim ovlašćenjima, što je na dijagramu predstavljeno interfejsom ANALITIČAR.

8.2. SPECIFIKACIJA BAZNOG OBJEKTA BIBLIOTEČKOG INFORMACIONOG SISTEMA

Na osnovu standarda razvoja otvorenih sistema (Poglavlje 4) specifikacija baznog objekta BIS-a izvršena je preko podmodela PMOV, čijom integracijom se dobija jedinstveni bazni objekat [3].

8.2.1. Model podataka standarda za održavanje (UNIMARC)

Model podataka standarda za održavanje baze podataka BIS-a prikazan je na Slici 8.3. Obzirom na činjenicu da veliki broj ustanova koristi UNIMARC [4] kao interni format čuvanja podataka o bibliografskim jedinicama, ili ga upotrebljava kao format za razmenu, on je prihvaćen kao standard za bibliografske podatke i podatke o stanju fonda u okviru BIS-a.



Slika 8.3. Model podataka za održavanje (UNIMARC)

Prema UNIMARC formatu, da bi se zapis mogao koristiti za standardizovanu razmenu mora se sastojati od :

- Oznake koja se odnosi na strukturu zapisa,
- Oznaka polja, koje jedinstveno identikuju polje u okviru zapisa,
- Skupa vrednosti koje opisuju dokument, razvrstanih u različita potpolja.

Svaki zapis sastoji se od identifikatora zapisa i niza POLJA koja karakterišu atributi:

- OBAVEZNOST - koji govori da li se polje mora obavezno nalaziti u okviru zapisa (polja preko kojih se jednoznačno može identifikovati zapis),
- PONOVLJIVOST - kojim se naznačava da li se polje može pojaviti više puta unutar jednog zapisa,
- SEKUNDARNOST - koji govori da li se to polje može javiti kao sadržaj nekog drugog polja (polja koja se mogu pojaviti kao vrednosti potpolja se nazivaju sekundarna polja).

Svako polje podataka sude dva indikatora koji daju dodatne informacije o sadržaju polja, ili o akciji koju je potrebno primeniti prilikom određenih procesa rukovanja podacima. Dozvoljene vrednosti indikatora za neko polje daju se preko objekata PRVI INDIKATOR i DRUGI INDIKATOR.

U okviru jednog polja može se nalaziti više različitih potpolja. Dozvoljena potpolja za neko polje su data objektom POTPOLJA uz koji se kao atributi javljaju:

- OBAVEZNOSTPP - koji definiše da li se potpolje obavezno pojavljuje unutar nekog polja ili ne,
- PONOVLJIVOSTPP - koji govori da li se potpolje može ponavljati unutar polja,
- DUŽINA - dužina vrednosti potpolja, izražena u broju karaktera, ako je vrednost potpolja ograničene dužine,
- TIPKONTROLE - koji definiše na koji način će se vršiti logička kontrola vrednosti potpolja prilikom unosa,
- DEFVREDNOST - koji služi za definisanje podrazumevane vrednosti nekog potpolja ako se ona često ponavlja kao vrednost tog potpolja.

Vrednost potpolja kojom se vrši opis dokumenta može biti tekst ili šifrirani podatak. Šifrirani podaci mogu važiti samo u okviru jednog polja i tada su dati objektom INTERNI ŠIFARNIK. Ako se mogu koristiti u više od jednog potpolja, tada pripadaju objektu EKSTERNI ŠIFARNIK. Eksterni šifarnici razvrstani su po tipovima (šifarnik zemalja, kodova autorstva, jezika

itd.). Vrednosti potpolja u okviru polja od nacionalnog značaja (blok 9 UNIMARC-a) struktuirane su u još jedan nivo podstrukture (objekat POTPOTPOLJA). Ako se kao sadržaj potpotpolja unose šifrirani podaci značenje šifara dato je objektom ŠIFARNIK POTPOTPOLJA.

8.2.2. Model podataka za radno okruženje bibliotekara

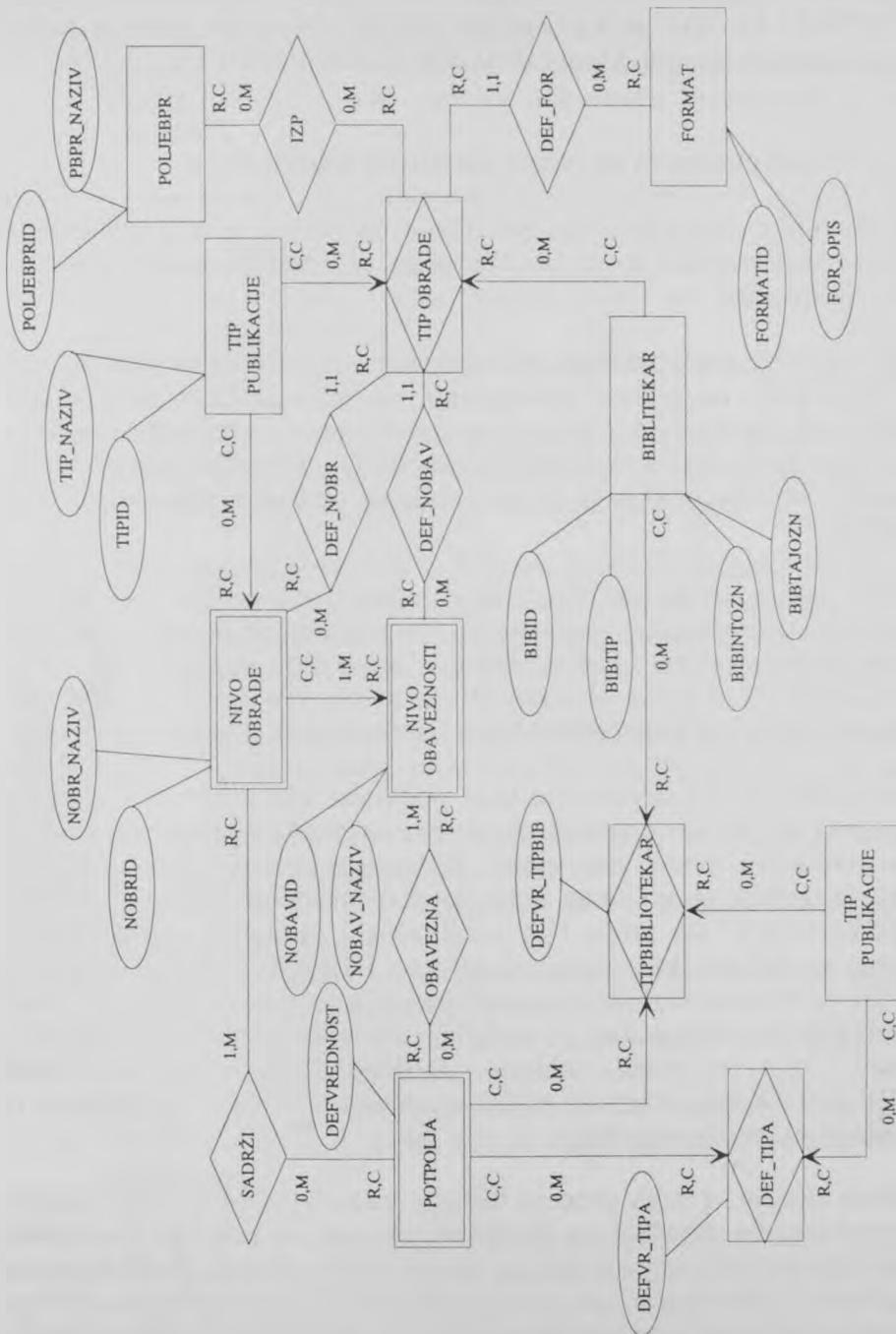
Prilagođavanje radnog okruženja specifičnim zahtevima svakog bibliotekara i tipovima dokumenata koje on obrađuje, obezbeđuje model podataka prikazan na Slici 8.4.

U bibliotekarskoj praksi dokumenti su podeljeni po tipovima (objekat TIP PUBLIKACIJE), na primer, monografije, serijske publikacije, doktorske disertacije itd. Zavisno od vrste i obima podataka koji se definišu prilikom obrade, postoji više nivoa obrade (objekat NIVO OBRADE). Za svaki nivo se definišu potpolja za koja se unose vrednosti na tom nivou obrade (veza SADRŽI).

Da bi se dokument mogao smatrati obrađenim na zadatom nivou, nije potrebno uneti (definisati) vrednosti za sva polja/potpolja tog nivoa, nego samo za obavezna (veza OBAVEZNA). Ta potpolja su vezana za NIVO OBAVEZNOSTI, a nivoa obaveznosti može biti više za isti nivo obrade, zavisno od vrste posla koju bibliotekar prilikom obrade dokumenta obavlja.

Modul za ažuriranje i održavanje baze podataka BIS-a treba da obezbedi bibliotekaru mogućnost preuzimanja podrazumevanih vrednosti za potpolja čije se vrednosti često ponavljaju. Podrazumevane vrednosti potpolja moguće je definisati na nivou potpolja, kao vrednost njegovog atributa DEFVREDNOST. Na nivou tipa publikacije i potpolja, podrazumevane vrednosti se definišu kao vrednost atributa DEFVR_TIPA uz agregaciju DEF_TIPA. Podrazumevanu vrednost potpolja koja važi samo za obrade koje vrši pojedini bibliotekar, po različitim tipovima publikacija, moguće je definisati kao vrednost atributa DEFVR_TIPBIB uz agregaciju TIPBIBLIOTEKAR. Važnost podrazumevanih vrednosti je obrnuta od prethodnog redosleda navođenja.

U opštem slučaju, u toku procesa obrade dokumenta bibliotekar različite tipove publikacija obrađuje na različitim nivoima, sa posebno definisanim nivoom obaveznosti unosa podataka, za svaki nivo obrade. Podrazumevani nivo obrade i obaveznosti, za jednog bibliotekara, po različitim tipovima publikacija dat je preko veza aggregacije TIP OBRADE sa odgovarajućim objektima (veze DEF_NOBR i DEF_NOBAV).



Slika 8.4. Model podataka za radno okruženje bibliotekara

Svakoj obradi dokumenta prethodi upit u bazu podataka da bi se proverilo da li je on već obrađen. Pri tome je vezom IZP definisan skup prefiksa za pretraživanje (objekat POLJEBPR) po kojima će bibliotekar vršiti upit, posebno za svaki tip publikacije. Odabrani skup prefiksa se pojavljuje kao inicijalni prilikom startovanja modula za pretraživanje baze bibliografske građe.

U okviru sistema neophodno je obezbediti različite formate izveštaja i prikaza rezultata pretraživanja. Na osnovu unapred definisane strukture izveštaja, za svakog bibliotekara na nivou tipa publikacije, definiše se podrazumevani format (objekat FORMAT) prikazivanja rezultata upita (vezom DEF_FOR). U toku rada bibliotekar može izvršiti izbor nekog drugog postojećeg formata ili definisati novi format prikaza. Struktura formata se definiše vrednošću atributa FOR_OPIS.

Uz objekat BIBLIOTEKAR dat je i skup atributa koji ga karakterišu:

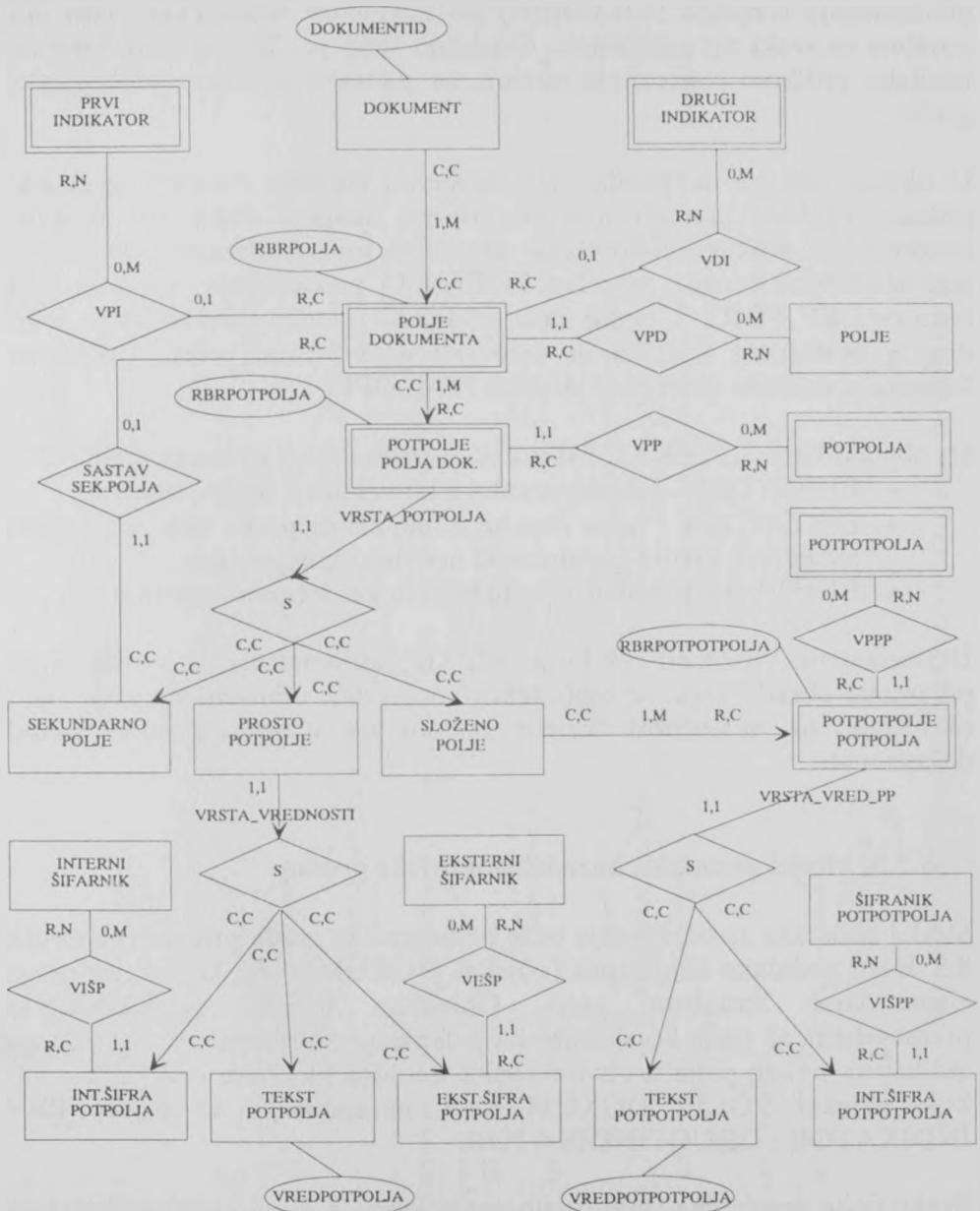
- BIBINTOZN - interna oznaka bibliotekara u okviru sistema,
- BIBTAJOZN - tajna oznaka bibliotekara preko koje se realizuje jedan nivo zaštite podataka od neovlašćenog pristupa,
- BIBTIP - tip publikacije koju bibliotekar trenutno obraduje.

Definisanjem vrednosti atributa BIBTIP automatski su određeni svi parametri obrade koju će bibliotekar sprovoditi, odnosno njegovo radno okruženje, uz mogućnost izmene parametara u toku procesa obrade dokumenata.

8.2.3. Model podataka baze bibliografske građe

Model podataka za održavanje baze bibliografske građe prikazan je na Slici 8.5. Bazu podataka čine zapisi (objekat DOKUMENT), koji se sastoje od elementarnih struktura, polja. Objektom POLJE DOKUMENTA predstavljena su polja koja obuhvata jedan zapis. Vrednosti prvog i drugog indikatora nekog polja u okviru zapisa definišu se preko veza VPI i VDI koje objekat POLJE DOKUMENTA povezuju sa objektima PRVI INDIKATOR i DRUGI INDIKATOR.

Svako polje zapisa ima skup svojih potpolja što je na modelu predstavljeno objektom POTPOLJE POLJA DOK. Uopšteno možemo reći da potpolja u okviru polja zapisa sadrže vrednosti koje opisuju neki dokument. Vrednost potpolja može biti prosta (objekat PROSTO POTPOLJE), strukturirana u još jedan nivo podstrukture (objekat SLOŽENO POTPOLJE) ili



Slika 8.5. Model podataka baze bibliografske grade

predstavljena struktrom tipa polje potpolja, što je u modelu prikazano objektom SEKUNDARNO POLJE. Polja koja se mogu pojaviti kao vrednost potpolja nazivaju se sekundarna, a pomoću njih se vrši povezivanje dokumenata prema UNIMARC formatu. Njihova struktura je ista kao i struktura ostalih polja UNIMARC-a, osim što se u okviru njih ne pojavljuju druga sekundarna polja. Veza SASTAV SEK. POLJA definiše strukturu sekundarnog polja.

Vrednost prostog potpolja može biti tekst, interna šifra ili eksterna šifra (specijalizacija VRSTA_VREDNOSTI). Veza VIŠP daje vrednost šifriranog podatka koji važi samo u okviru tog potpolja, a veza VEŠP daje kodiranu vrednost potpolja ako se ona koristi u više različitih potpolja.

Složeno potpolje sadrži skup potpotpolja (objekat POTPOTPOLJE POTPOLJA) u okviru kojih se definišu vrednosti. Vrednost potpotpolja može biti tekst ili šifrirani podatak iz internog šifarnika potpotpolja (specijalizacija VRSTA_VRED_PP)

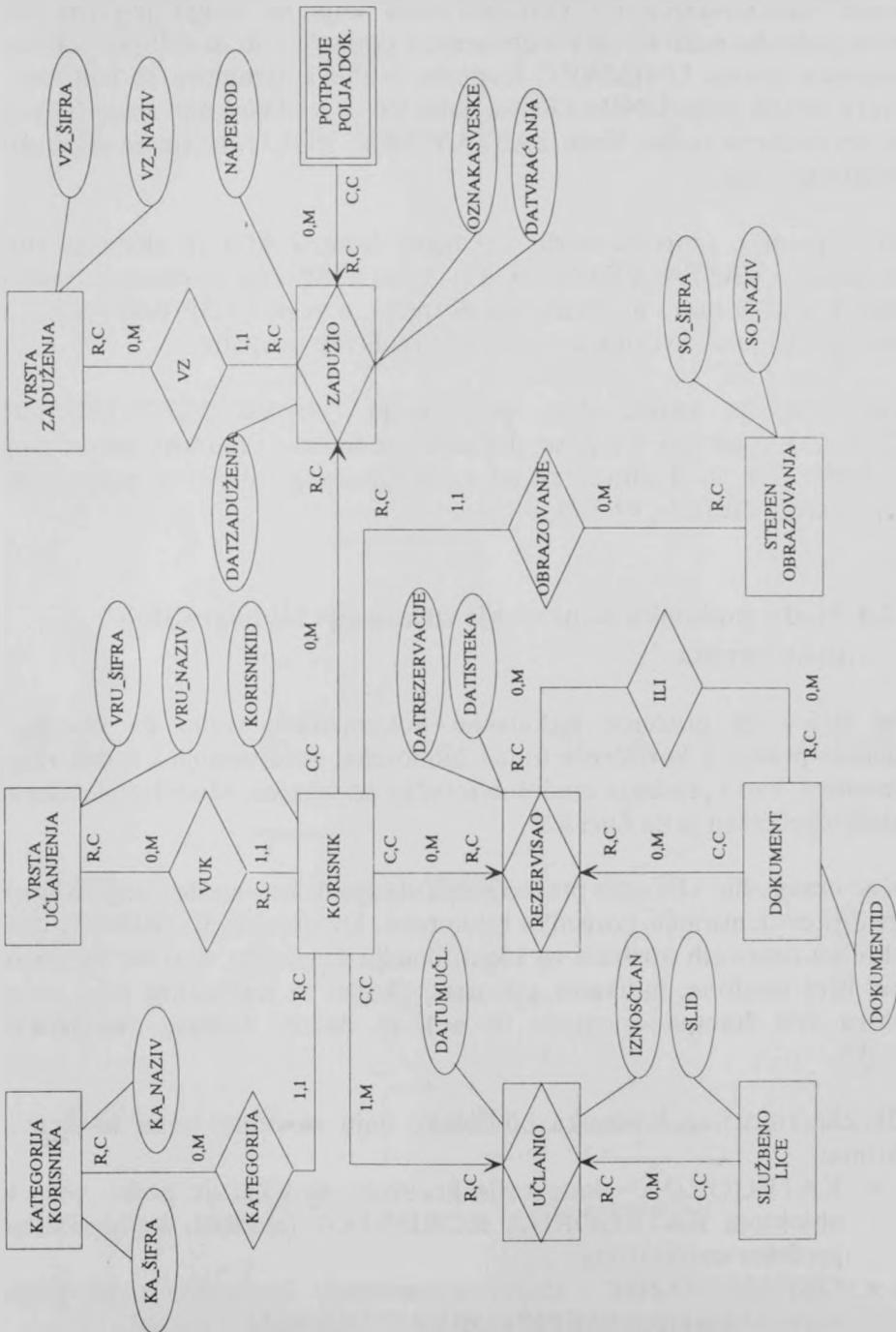
8.2.4. Model podataka za praćenje cirkulacije bibliografskih dokumenata

Modul BIS-a za praćenje cirkulacije dokumenata, treba da obezbedi mogućnost praćenja korišćenja fonda biblioteke, rezervisanja i zaduživanja dokumenata, kao i praćenja međubibliotečke pozajmice. Model podataka za cirkulaciju prikazan je na Slici 8.6.

Da bi se omogućilo efikasno praćenje cirkulacije dokumenata neophodno je obezbediti evidentiranje korisnika biblioteke. Uz objekat KORISNIK dato je nekoliko osnovnih atributa za identifikaciju korisnika, kao što su naziv, adresa, broj telefona, indikator opomene (kojim se naznačava da li se za korisnika vrši stampa opomena ili ne), ili datum zadnjeg zaduživanja korisnika.

Ostale karakteristike korisnika biblioteke daju se preko veza sa drugim objektima:

- KATEGORIJA - kategorija korisnika se iskazuje preko veze sa objektom KATEGORIJA KORISNIKA (student, srednjoškolac, profesor univerziteta ...),
- OBRAZOVANJE - stepen obrazovanja korisnika je dat preko veze sa objektom STEPEN OBRAZOVANJA,



Slika 8.6. Model podataka za cirkulaciju bibliografskih dokumenata

- VUK - vrsta učlanjenja korisnika je data preko veze sa objektom VRSTA UČLANJENJA (godišnji član, počasni, sa posebnim statusom ...).

Praćenje učlanjivanja korisnika potrebno je obezbediti tako da se čuvaju podaci za svako novo učlanjenje korisnika sa mogućnošću formiranja stanja primljene članarine (dnevno, mesečno,...) po svakom bibliotekaru koji učlanjuje korisnike (objekat SLUŽBENO LICE). Zbog toga se učlanjivanje prati agregacijom UČLANIO uz koju se kao vrednost atributa DATUMUČL čuva datum učlanjenja korisnika i visina članarine kao vrednost atributa IZNOSČLAN. Rezervacija dokumenata za korisnika prati se preko agregacije REZERVISAO uz koju su kao atributi dati:

- DATREZERVACIJE - datum kada je korisnik rezervisao neki dokument,
- DATISTEKA - datum kada ističe rezervacija dokumenta.

Korisnik može izvršiti i uslovnu rezervaciju dokumenata (veza ILI) kojom se definiše zamena za prvo rezervisani naslov ili izdanje dokumenta.

Podaci o stanju fonda prate se ponavljanjem vrednosti polja 996, za monografije, odnosno, 997 za serijske publikacije (svakom primerku dokumenta odgovara jedno polje 996 ili 997). Praćenje zaduživanja dokumenata vrši se preko agregacije ZADUŽIO, koja povezuje, sa jedne strane korisnika, a sa druge POTPOLJE POLJA DOK. (oznaku inventara ili signature) polja 996 ili 997 primeraka dokumenta koji je zadužio neki korisnik. Uz aggregaciju su dati atributi:

- DATZADUŽENJA - datum kada je korisnik zadužio dokument,
- NAPERIOD - vremenski period na koji je dokument zadužen,
- DATVRAĆANJA - datum kada je korisnik dokument razdužio,
- OZNAKASVESKE - oznaka sveske koja je zadužena, odnosi se na zaduživanje serijskih publikacija koje nisu ukoričene ili su delimično ukoričene u jedan fizički povez (jednom polju 997 odgovara više primeraka).

VRSTA ZADUŽENJA dokumenta (u čitaonici, van biblioteke, interno zaduženje,...) prati se preko veze VZ.

8.3. SPECIFIKACIJA APLIKACIONIH OBJEKATA

Na dijagramima toka podataka prikazanim na Slici 8.1 i Slici 8.2, BIS je predstavljen kao skup nezavisnih aplikacionih objekata koji komuniciraju sa baznim objektom. Dalji razvoj sistema realizuje se kroz projektovanje i implementaciju ovih objekata. Ovde je data specifikacija samo funkcije Formiranje bibliografske građe koja predstavlja jezgro autonomnog BIS-a.

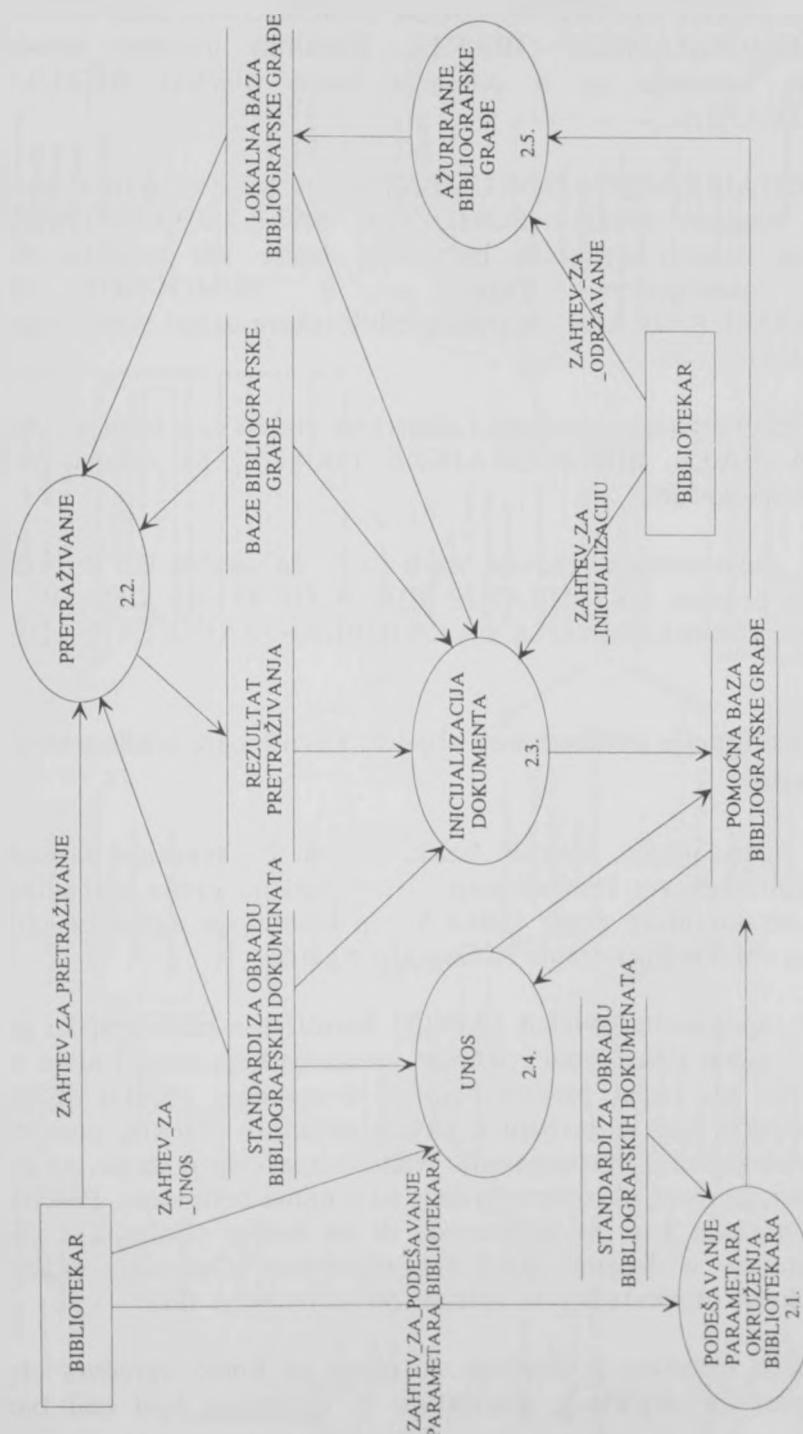
8.3.1. Funkcija Formiranje bibliografske građe

Funkcija Formiranje bibliografske građe treba da omogući bibliotekaru da na osnovu dokumenta formira zapis u bazi bibliografske građe. To podrazumeva proveru postojanja zapisa o dokumentu, izbor načina obrade dokumenta i formiranje samog zapisa. Pri obradi dokumenta, bibliotekar treba da ima direktni pristup strukturi zapisa i podacima u njemu, kako bi se obezbedio brz pristup u toku unosa, a redosled zadavanja podataka prepustio bibliotekaru. Posto se u bazi bibliografske građe čuvaju zapisi koji obavezno sadrže bar minimalan skup podataka o dokumentu, potrebno je obezbediti čuvanje međurezultata ažuriranja, tako da vreme trajanja formiranja zapisa ne bude ograničeno. Iz svega navedenog i na osnovu metodologije, (Poglavlje 4) u okviru funkcije Formiranje bibliografske građe uočena je dugačka logička atomska transakcija pa je potrebna dekompozicija funkcije na fizičke atomske procese, kao i uvođenje pomoćnog baznog objekta.

Na Slici 8.7 prikazana je dekompozicija funkcije Formiranje bibliografske građe. Izdvojeni su sledeći procesi: PODEŠAVANJE PARAMETARA OKRUŽENJA BIBLIOTEKARA, PRETRAŽIVANJE, INICIJALIZACIJA DOKUMENTA i UNOS I AŽURIRANJE BIBLIOGRAFSKE GRAĐE.

Proces PODEŠAVANJE PARAMETARA OKRUŽENJA BIBLIOTEKARA omogućava da se na zahtev bibliotekara preuzmu parametri konkretnе obrade iz baznog objekta STANDARDI ZA OBRADU BIBLIOGRAFSKIH DOKUMENATA, koji definiše modele obrade. Preuzeti parametri inicijalizuju strukturu zapisa u pomoćnom baznom objektu POMOĆNA BAZA BIBLIOGRAFSKE GRAĐE.

Proces PRETRAŽIVANJE, nakon zahteva bibliotekara, a na osnovu definisanog formata, pretražuje dostupne baze bibliografske građe. Pretraživanje se vrši u lokalnoj bazi i/ili u udaljenim bazama bibliografske



Slika 87. Dekompozicija funkcije Formiranje bibliografske grada

građe (bazni objekti LOKALNA BAZA BIBLIOGRAFSKE GRAĐE i BAZE BIBLIOGRAFSKE GRAĐE). Rezultati uspešno izvedenog pretraživanja smeštaju se u pomoćni bazni objekat REZULTAT PRETRAŽIVANJA.

Proces INICIJALIZACIJA DOKUMENTA podešava strukturu zapisa u pomoćnom baznom objektu POMOĆNA BAZA BIBLIOGRAFSKE GRAĐE na osnovu strukture izabranog zapisa i/ili modela obrade definisanog standardom. Tako se u POMOĆNOJ BAZI BIBLIOGRAFSKE GRAĐE za jednog bibliotekara nalazi samo zapis čija je obrada u toku.

Proces UNOS omogućava unošenje i ažuriranje podataka u baznom objektu POMOĆNA BAZA BIBLIOGRAFSKE GRAĐE, na osnovu pravila definisanih standardom.

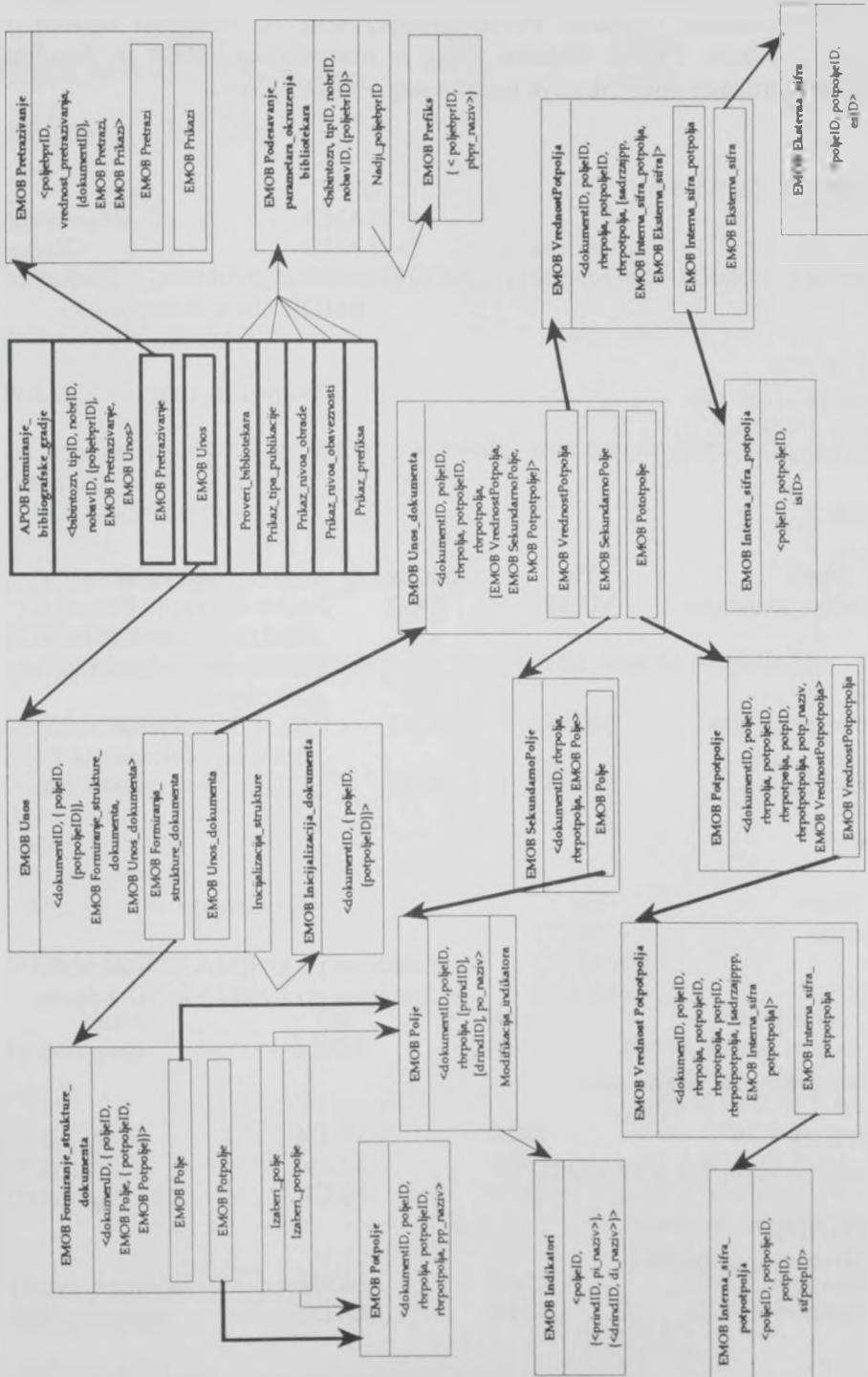
Bibliotekar, posle završene obrade, zapis može da sačuva, što se realizuje izvršavanjem procesa AŽURIRANJE BIBLIOGRAFSKE GRAĐE, koja ažurira bazni objekat LOKALNA BAZA BIBLIOGRAFSKE GRAĐE.

8.3.2. Specifikacija aplikacionog objekta Formiranje bibliografske građe

Na osnovu metodologije, atomski fizički procesi Podešavanje parametara okruženja bibliotekara, Pretraživanje, Unos, Inicijalizacija dokumenta i Ažuriranje bibliografske građe (Slika 8.7) predstavljaju ugrađene objekte aplikacionog objekta Formiranje bibliografske građe.

Na Slici 8.8, aplikacioni objekat (APOB) Formiranje bibliografske građe, prikazan je svojom strukturom i skupom operacija koje se nad njom mogu izvršavati. Na isti način predstavljeni su i ugrađeni objekti (EMOB). Ugrađeni objekti, koji se javljaju u okviru strukture objekta, posebno su istaknuti upisivanjem u pravougaonik. Njihovo značenje je da se, na zahtev ili automatski, izvršava niz operacija koje su u njima definisane. Posebno su navedene operacije koje se automatski ili na zahtev pozivaju, a čije se definicije nalaze u drugim ugrađenim objektima. Operacije definisane lokalno, u okviru objekata koji se opisuje, ne nalaze se na slici.

Dekompozicija objekata je izvršena do nivoa na kome ugrađeni objekti pozivaju operacije objektnog analizatora ili izvršavaju upit nad baznim objektima.



S 88 Struktura aplikacionog objekta Formiranje bibliografske grade

U strukturi aplikacionog objekta Formiranje bibliografske građe mogu se uočiti dva ugrađena objekta: Pretraživanje, koji je detaljno opisan u Poglavlju 6 i Unos. Prema sintaksi dатој u metodologiji data je detaljna objektno orijentisana specifikacija ugrađenog objekta Unos.

EMOB Unos

STRUCTURE

<documentID, {poljeID, {potpoljeID}}, EMOB Formiranje_strukture_dokumenta,
EMOB Unos_dokumenta>

OPERATIONS

Inicijalizacija_strukture
Formiranje_strukture
Unos_sadržaja

DYNAMICS

Naziv	Tip	Uslov	Napomena
Inicijalizacija_strukture	MESS	REQ	Poziva operaciju Prikaži _strukturu_dokumenta iz EMOB Inicijalizacija _dokumenata
Formiranje_strukture	MESS	REQ	EMOB Formiranje _strukture_dokumenta
Unos_sadržaja	MESS	REQ	EMOB Unos_dokumenta

FIELDS

Naziv	Tip	Ograničenje
dokumentID	CHAR(7)	ograničenje po UNIMARC standardu
poljeID	CHAR(4)	ograničenje po UNIMARC standardu
potpoljeID	CHAR(1)	ograničenje po UNIMARC standardu

EMOB Inicijalizacija_dokumenta

STRUCTURE

<documentID, {poljeID, {potpoljeID}}>

OPERATIONS

Prikaži_strukturu_dokumenta

DYNAMICS

Naziv	Tip	Uslov	Napomena
Prikaži_strukturu_dokumenta	FUN	REQ	-

FIELDS

Naziv	Tip	Ograničenje
dokumentID	CHAR(7)	-
poljeID	CHAR(4)	ograničenje po UNIMARC standardu
potpoljeID	CHAR(1)	ograničenje po UNIMARC standardu

FUNCTION LIBRARY

Prikaži_strukturu_dokumenta

Preuzmi skup polja i potpolja iz dokumenta, a na osnovu Standarda za formiranje baze podataka;
Prikaži datu strukturu.

EMOB Formiranje_strukture_dokumenta

STRUCTURE

<dokumentID, {poljeID, EMOB Polje, {potpoljeID, EMOB Potpolje }}>

OPERATIONS

Izaberi_polje
Dodaj_polje
Briši_polje
Izaberi_potpolje
Dodaj_potpolje
Briši_potpolje

DYNAMICS

Naziv	Tip	Uslov	Napomena
Izaberi_polje	MESS	REQ	Poziva operaciju Prikaži_indikatore i Prikaži_naziv_polja iz EMOB Polje
Dodaj_polje	MESS	REQ	EMOB Polje
Briši_polje	MESS	REQ	EMOB Polje
Izaberi_potpolje	MESS	REQ	Poziva operaciju Prikaži_naziv_potpolja iz EMOB Potpolje
Dodaj_potpolje	MESS	REQ	EMOB Potpolje
Briši_potpolje	MESS	REQ	EMOB Potpolje

FIELDS

Naziv	Tip	Ogranicenje
dokumentID	CHAR(7)	
poljeID	CHAR(4)	ograničenje po UNIMARC standardu
potpoljeID	CHAR(1)	ograničenje po UNIMARC standardu

EMOB Unos_dokumenta

STRUCTURE

<dokumentID, poljeID, rbrpolja, potpoljeID, rbrpotpolja,
[EMOB VrednostPotpolja, EMOB SekundarnoPolje, EMOB Potpotpolje] >

OPERATIONS

Prikaži_vrednost_potpolja
Prikaži_sekundarno_polje
Prikaži_potpotpolje

DYNAMICS

Naziv	Tip	Uslov	Napomena
Prikaži_vrednost_potpolja	MESS	REQ	EMOB VrednostPotpolja
Prikaži_sekundarno_polje	MESS	REQ	EMOB SekundarnoPolje
Prikaži_potpotpolje	MESS	REQ	EMOB Potpotpolje

FIELDS

Naziv	Tip	Ogranicenje
dokumentID	CHAR(7)	
poljeID	CHAR(4)	ograničenje po UNIMARC standardu
rbrpolja	INTEGER(2)	
potpoljeID	CHAR(1)	ograničenje po UNIMARC standardu
rbrpotpolja	INTEGER(2)	

EMOB Polje

STRUCTURE

<dokumentID, poljeID, rbrpolja, [prindID], [drindID], po_naziv>

OPERATIONS

Prikaži_indikatore
Prikaži_naziv_polja
Modifikacija_indikatora
UBACI
IZBACI

DYNAMICS

Naziv	Tip	Uslov	Napomena
Prikaži_indikatore	FUN	AU - po izboru polja	-
Prikaži_naziv_polja	FUN	AU - po izboru polja	-
Modifikacija_indikatora	MESS	REQ	Poziva operaciju Izaberi _indikator EMOB Indikatori

FIELDS

Naziv	Tip	Ograničenje
dokumentID	CHAR(7)	
poljeID	CHAR(4)	ograničenje po UNIMARC standardu
prindID	CHAR(1)	ograničenje po UNIMARC standardu
drindID	CHAR(1)	ograničenje po UNIMARC standardu
po_naziv	CHAR(51)	-
rbrpolja	INTEGER(2)	-

FUNCTION LIBRARY

Prikaži_indikatore

```
SELECT PRINDID, DRINDID INTO <prindID, drindID>
FROM PRVI_INDIKATOR, DRUGI_INDIKATOR
WHERE PRVI_INDIKATOR.DOKUMENTID = dokumentID AND
      PRVI_INDIKATOR.POLJEID = poljeID AND
      DRUGI_INDIKATOR.DOKUMENTID = dokumentID AND
      DRUGI_INDIKATOR.POLJEID = poljeID
```

Prikaži_naziv_polja

```
SELECT PO_NAZIV INTO <po_naziv>
FROM POLJE
WHERE POLJE.POLJEID = poljeID
```

EMOB Indikatori

STRUCTURE
<poljeID, {<prindID, pi_naziv>}, {<drindID, di_naziv>} >

OPERATIONS

Prikaži_prvi_indikator

Prikaži_drugi_indikator

Izaberi_indikator

DYNAMICS

Naziv	Tip	Uslov	Napomena
Prikaži_prvi_indikator	FUN	AU - po izboru polja	-
Prikaži_drugi_indikator	FUN	AU - po izboru polja	-
Izaberu_indikator	FUN	REQ	-

FIELDS

Naziv	Tip	Ograničenje
poljeID	CHAR(4)	ograničenje po UNIMARC standardu
prindID	CHAR(1)	ograničenje po UNIMARC standardu
drindID	CHAR(1)	ograničenje po UNIMARC standardu
pi_naziv	CHAR(51)	
di_naziv	CHAR(51)	

FUNCTION LIBRARY

Prikaži_prvi_indikator

```
SELECT PRINDID, PI_NAZIV INTO < prindID, pi_naziv >
FROM PRVI_INDIKATOR
WHERE PRVI_INDIKATOR.POLJEID = poljeID
```

Prikaži_drugi_indikator

```
SELECT DRINDID, DI_NAZIV INTO < drindID, di_naziv >
FROM DRUGI_INDIKATOR
WHERE DRUGI_INDIKATOR.POLJEID = poljeID
```

Izaberu_indikator

Izaberu jedan od prikazanih indikatora;
Prenesi PRINDID u prindID ili DRINDID u drindID;

EMOB Potpolje

STRUCTURE

< dokumentID, poljeID, rbrpolja, potpoljeID, rbrpotpolja, pp_naziv >

OPERATIONS

Prikaži_naziv_potpolja

UBACI

IZBACI

DYNAMICS

Naziv	Tip	Uslov	Napomena
Prikaži_naziv_potpolja	FUN	AU - po izboru polja	-

FIELDS

Naziv	Tip	Ograničenje
dokumentID	CHAR(7)	
poljeID	CHAR(4)	ograničenje po UNIMARC standardu
rbrpolja	INTEGER(2)	
potpoljeID	CHAR(1)	ograničenje po UNIMARC standardu
rbrpotpolja	INTEGER(2)	
pp_naziv	CHAR(51)	

FUNCTION LIBRARY

Prikaži_naziv_potpolja

```
SELECT PP_NAZIV INTO <pp_naziv>
FROM POTPOLJA
WHERE POTPOLJA.POTPOLJAID = potpoljaID
```

EMOB Potpotpolje

STRUCTURE

```
< dokumentID, poljeID, rbrpolja, potpoljeID, rbrpotpolja, potpID,
    rbrpotpotpolja, potp_naziv, EMOB VrednostPotpotpolja >
```

OPERATIONS

Prikaži_potpotpolja
Izaberi_potpotpolje
Prikaži_vrednost_potpotpolja

DYNAMICS

Naziv	Tip	Uslov	Napomena
Prikaži_potpotpolja	FUN	AU - po izboru potpolja	-
Izaberi_potpotpolje	FUN	REQ	-
Prikaži_vrednost _potpotpolja	MESS	REQ	EMOB VrednostPotpotpolja

FIELDS

Naziv	Tip	Ograničenje
dokumentID	CHAR(7)	
poljeID	CHAR(4)	ograničenje po UNIMARC standardu
rbrpolja	INTEGER(2)	
potpoljeID	CHAR(1)	ograničenje po UNIMARC standardu
rbrpotpolja	INTEGER(2)	
potpID	CHAR(1)	ograničenje po UNIMARC standardu
rbrpotpotpolja	INTEGER(2)	
potp_naziv	CHAR(51)	

FUNCTION LIBRARY

Prikaži_potpotpolja

```
SELECT POTPID, POTP_NAZIV INTO < potpID, potp_naziv >
FROM POTPOTPOLJE
WHERE POTPOTPOLJE.DOKUMENTID = dokumentID AND
      POTPOTPOLJE.POLJEID = poljeID AND
      POTPOTPOLJE.POTPOTPOLJEID = potpoljeID
```

Izberi_potpotpolje

Izberi jedno od prikazanih potpotpolja;

EMOB SekundarnoPolje

STRUCTURE
<dokumentID, rbrpolja, rbrpotpolja, EMOB Polje>

OPERATIONS

UBACI

Ubaci_sekundarno_polje

IZBACI

Izbaci_sekundarno_polje

DYNAMICS

Naziv	Tip	Uслов	Napomena
Ubaci_sekundarno_polje	MESS	REQ	EMOB Polje
Izbaci_sekundarno_polje	MESS	REQ	EMOB Polje

FIELDS

Naziv	Tip	Ograničenje
dokumentID	CHAR(7)	-
rbrpolja	INTEGER(2)	-
rbrpotpolja	INTEGER(2)	-

EMOB VrednostPotpolja

STRUCTURE

<dokumentID, poljeID, rbrpolja, potpoljeID, rbrpotpolja,
[sadržajpp, EMOB Interna_šifra_potpolja, EMOB Eksterna_šifra]>

OPERATIONS

UBACI
IZBACI
PROMENI
Izaberi_internu_šifru_potpolja
Izaberi_eksternu_šifru_potpolja

DYNAMICS

Naziv	Tip	Uslov	Napomena
Izaberi_internu_šifru_potpolja	MESS	REQ	EMOB Interna_šifra_potpolja
Izaberi_eksternu_šifru_potpolja	MESS	REQ	EMOB Eksterna_šifra

FIELDS

Naziv	Tip	Ograničenje
dokumentID	CHAR(7)	-
poljeID	CHAR(4)	ograničenje po UNIMARC standardu
rbrpolja	INTEGER(2)	-
potpoljeID	CHAR(1)	ograničenje po UNIMARC standardu
rbrpotpolja	INTEGER(2)	-
sadržajpp	CHAR(2048)	-

EMOB VrednostPotpotpolja

STRUCTURE

<dokumentID, poljeID, rbrpolja, potpoljeID, rbrpotpolja, potpID,
rbrpotpotpolja, [sadržajpp, EMOB Interna_šifra_potpotpolja]>

OPERATIONS

UBACI

IZBACI

PROMENI

Izaberi_internu_šifru_potpotpolja

DYNAMICS

Naziv	Tip	Uslov	Napomena
Izaberi_internu_šifru_potpotpolja	MESS	REQ	EMOB Interna_sifra_potpotpolja

FIELDS

Naziv	Tip	Ograničenje
dokumentID	CHAR(7)	ograničenje po UNIMARC standardu
poljeID	CHAR(4)	ograničenje po UNIMARC standardu
rbrpolja	INTEGER(2)	ograničenje po UNIMARC standardu
potpoljeID	CHAR(1)	ograničenje po UNIMARC standardu
rbrpotpolja	INTEGER(2)	ograničenje po UNIMARC standardu
potpID	CHAR(1)	ograničenje po UNIMARC standardu
rbrpotpotpolja	INTEGER(2)	ograničenje po UNIMARC standardu
sadržajppp	CHAR(2048)	

EMOB Interna_sifra_potpolja

STRUCTURE

< poljeID, potpoljeID, isID >

OPERATIONS

Prikaži_interne_šifre_pp

Izaberi_internu_šifru_pp

DYNAMICS

Naziv	Tip	Uslov	Napomena
Prikaži_interne_šifre_pp	FUN	REQ	
Izaberi_internu_šifru_pp	FUN	REQ	

FIELDS

Naziv	Tip	Ogranicenje
poljeID	CHAR(4)	ograničenje po UNIMARC standardu
potpoljeID	CHAR(1)	ograničenje po UNIMARC standardu
isID	CHAR(4)	ograničenje po UNIMARC standardu

FUNCTION LIBRARY

Prikaži_interne_šifre_pp

```
SELECT ISID INTO < isID >
FROM INTERNI_ŠIFARNIK
WHERE INTERNI_ŠIFARNIK.POLJEID = poljeID AND
      INTERNI_ŠIFARNIK.POTPOTPOLJEID = potpoljeID
```

Izaberi_internu_šifru_pp

Izaberi jednu od prikazanih internih šifri potpolja,
Prenesi ISID u isID;

EMOB Interna_šifra_potpotpolja

STRUCTURE

```
< poljeID, potpoljeID, potpID, šifpotpID >
```

OPERATIONS

Prikaži_interne_šifre_ppp

Izaberi_internu_šifru_ppp

DYNAMICS

Naziv	Tip	Uslov	Napomena
Prikaži_interne_šifre_ppp	FUN	REQ	-
Izaberi_internu_šifru_ppp	FUN	REQ	-

FIELDS

Naziv	Tip	Ograničenje
poljeID	CHAR(4)	ograničenje po UNIMARC standardu
potpoljeID	CHAR(1)	ograničenje po UNIMARC standardu
potpID	CHAR(1)	ograničenje po UNIMARC standardu
isID	CHAR(4)	ograničenje po UNIMARC standardu

FUNCTION LIBRARY

Prikaži_interne_šifre_ppp

```
SELECT ŠIFPOTPID INTO < šifpotpID >
FROM ŠIFARNIK_POTPOTPOLJA
WHERE ŠIFARNIK_POTPOTPOLJA.POLJEID = poljeID AND
      ŠIFARNIK_POTPOTPOLJA.POTPOTPOLJEID = potpoljeID
      ŠIFARNIK_POTPOTPOLJA.POTPID = potpID
```

Izaberi_internu_šifru_potpotpolja

Izaberi jednu od prikazanih internih šifri potpotpolja,
Prenesi ŠIFPOTPID u šifpotID;

EMOB Eksterna_šifra

STRUCTURE

< poljeID, potpoljeID, esID >

OPERATIONS

Prikaži_eksterne_šifre

Izaberi_eksternu_šifru

DYNAMICS

Naziv	Tip	Uslov	Napomena
Prikaži_eksterne_šifre	FUN	REQ	
Izaberi_eksternu_šifru	FUN	REQ	

FIELDS

Naziv	Tip	Ogranicenje
poljeID	CHAR(4)	ograničenje po UNIMARC standardu
potpoljeID	CHAR(1)	ograničenje po UNIMARC standardu
esID	CHAR(6)	ograničenje po UNIMARC standardu

FUNCTION LIBRARY

Prikaži_eksterne_šifre

```
SELECT ESID INTO < esID >
FROM EKSTERNI_ŠIFARNIK
WHERE KONTROLA.POLJEID = poljeID AND
      KONTROLA.POTPOTPOLJEID = potpoljeID AND
      EKSTERNI_ŠIFARNIK.TESID=KONTROLA.TESID
```

Izaberi_eksternu_šifru

Izaberi jednu od prikazanih eksternih sifri;
Prenesi ESID u esID;

Na Slici 8.9 prikazana je distribucija procesa funkcije Formiranje bibliografske građe po slojevima i jedna moguća distribucija po različitim

čvorovima mreže (odnosno "client-server" arhitektura). Prepostavlja se da se pomoćna baza bibliografske građe, lokalna baza bibliografske građe, standard za obradu bibliografskih dokumenata, kao i druge baze bibliografske građe nalaze na različitim sistemima za upravljanje bazom podataka.

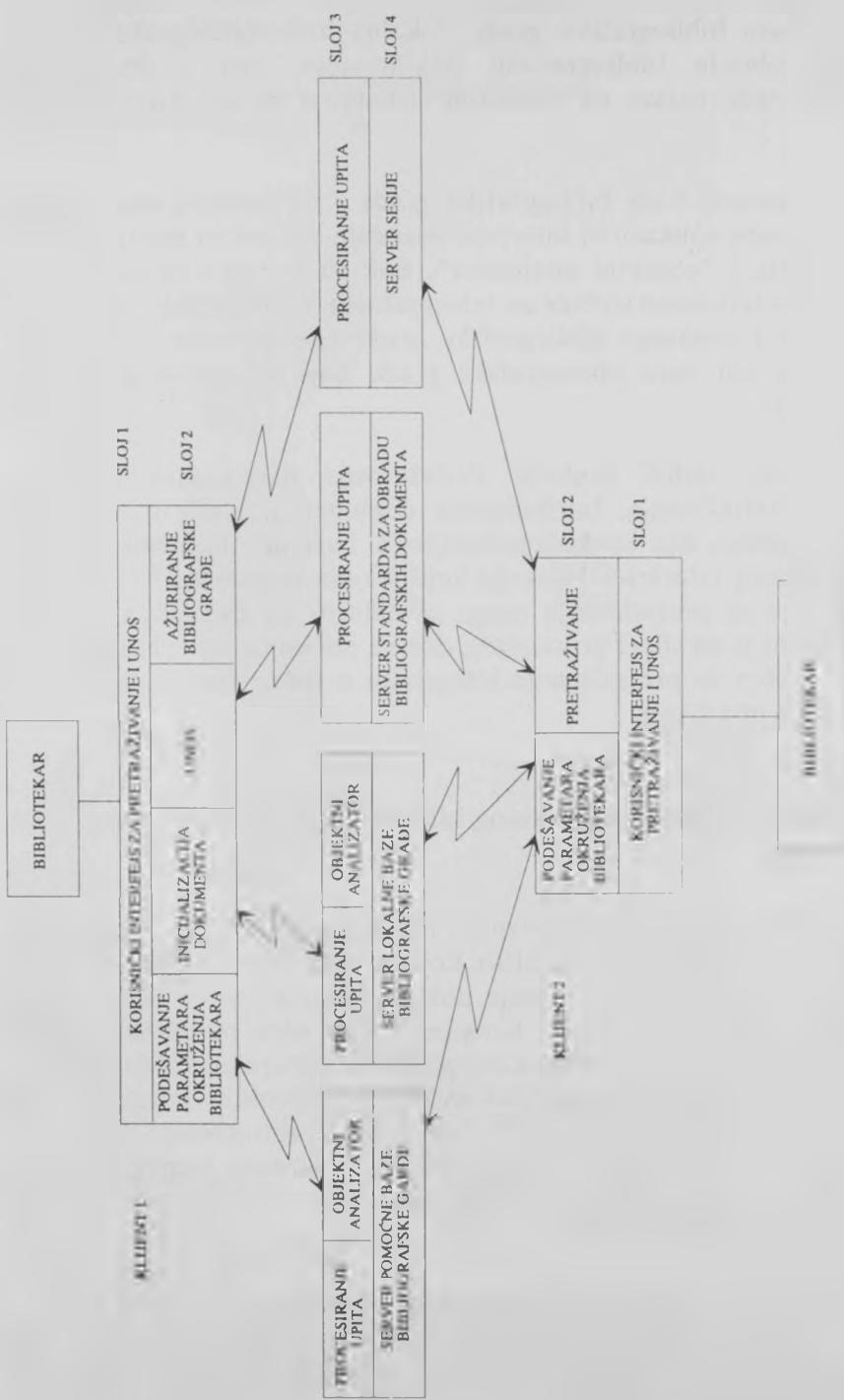
Na serveru pomoćne baze bibliografske građe i na serveru lokalne baze bibliografske građe aplikacioni interfejsi poseduju softver za interpretaciju i izvršavanje upita i "objektni analizator", dok se na serveru standarda i serveru sesije nalazi samo softver za interpretaciju i izvršavanje upita, jer u okviru funkcije Formiranje bibliografske građe nije dozvoljeno ažuriranje baze standarda niti baza bibliografske građe koje se nalaze na drugim čvorovima mreže.

Jezgro aplikacije sadrži funkcije Podešavanje parametara okruženja bibliotekara, Pretraživanje, Inicijalizacija dokumenta, Unos i Ažuriranje bibliografske građe, čija se komunikacija sa baznim objektom realizuje preko aplikacionog interfejsa. Funkcije koje se odnose na unos bibliografske građe i funkcije za pretraživanje mogu se nalaziti na dva fizički različita čvora mreže (što je na slici i prikazano), dok su korisnički interfejs za unos i korisnički interfejs za pretraživanje integrисани u jedan korisnički interfejs koji se nalazi na oba čvora.

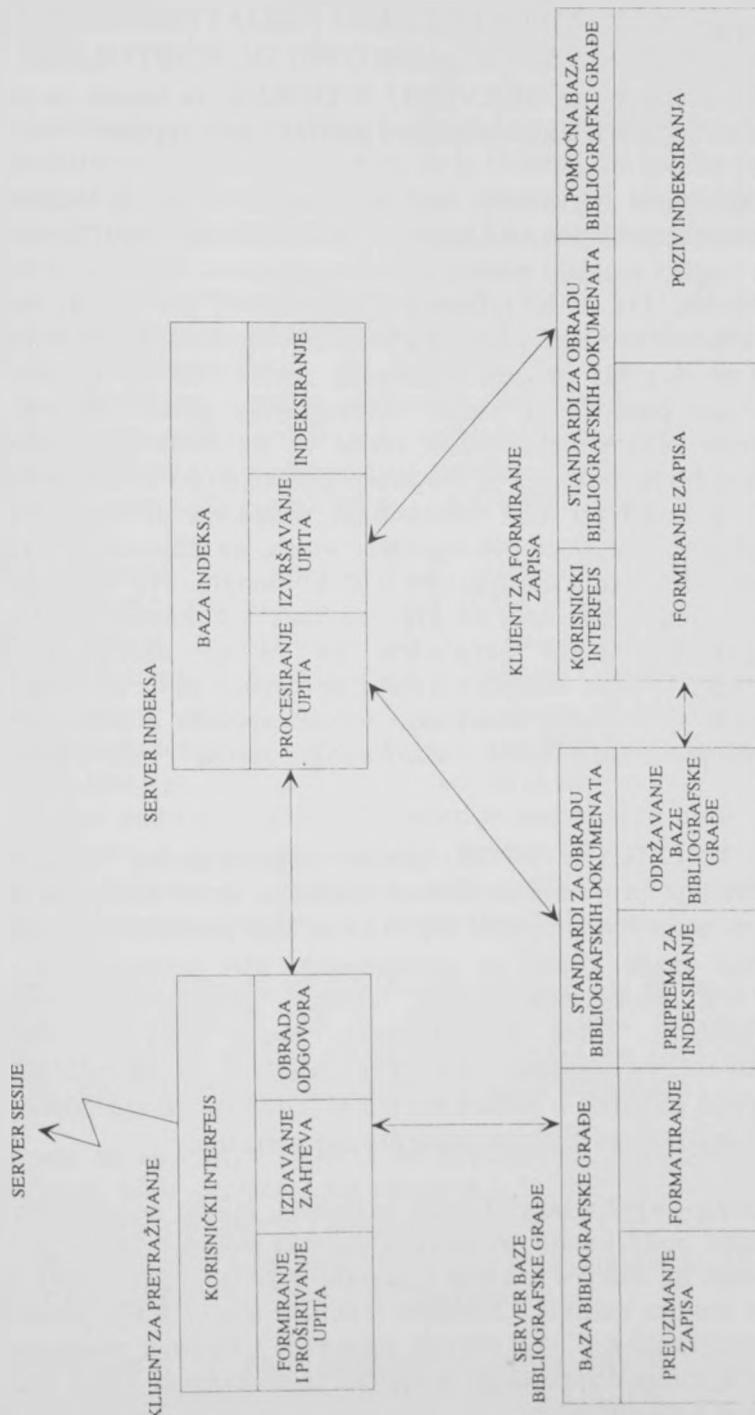
8.3.3. Fizička arhitektura lokalnog bibliotečkog informacionog sistema

Prihvaćena metodologija projektovanja i realizacije kao i opšta arhitektura SNTIS, obezbedili su otvorenost BIS-a koji se, oslanjajući se na jedinstveni sistem za indeksiranje i pretraživanje dokumenata, u osnovi implementira kao autonoman (lokalni) sistem. Kooperativnost rada pojedinih lokalnih bibliotečkih informacionih sistema za podršku uzajamne katalogizacije i pretraživanja više baza podataka na osnovu jednog upita, ostvaruje se na osnovu globalnog rečnika i jedinstvenog sistema za pretraživanje. Moguće je ostvariti fizičku distribuciju i kooperativnost softvera samog lokalnog bibliotečkog informacionog sistema kako na jednom računaru, tako i na više fizičkih servera sa klijentima [5].

Fizička arhitektura lokalnog BIS-a šematski je prikazana na Slici 8.10. Na SERVERU BAZE BIBLIOGRAFSKE GRAĐE pored baze bibliografske građe, nalazi se i baza standarda za obradu bibliografskih dokumenata. Na tom serveru izvršava se deo aplikacije za preuzimanje podataka o obrađenim



Slika 8.9. Distribucija funkcije Formiranje bibliografske građe



Slika 8.10. Fizička arhitektura lokalnog bibliotečkog informacionog sistema

dokumentima, održavanje baze bibliografske građe, kao i priprema za indeksiranje zapisa.

Indeksiranje zapisa vrši se na SERVERU INDEKSA na kojem se pored baze indeksa nalazi i sistem za indeksiranje i pretraživanje zapisa.

Softver za formiranje i održavanje zapisa izvršava se na pojedinačnim klijentima za formiranje zapisa. KLIJENT ZA FORMIRANJE ZAPISA, obavlja dugački logički atomski proces formiranja zapisa, odnosno složenog aplikacionog objekta. Da bi se sačuvali međurezultati pri obradi jednog dokumenta, na klijentu se nalazi pomoćna baza bibliografske građe kojom je obezbeđeno da se dugački proces formiranja zapisa razbijе na atomske fizičke procese nad pomoćnom bazom bibliografske građe. Procesi nad pomoćnom bazom bibliografske građe svode se na dodavanje odnosno brisanje pojedinog polja, potpolja ili vrednosti potpolja (po UNIMARC-u), preko korisničkog interfejsa koji obezbeđuje sintaksnu proveru ulaznih podataka. Pored pomoćne baze bibliografske građe na klijentu se nalazi i kopija standarda za obradu bibliografskih dokumenata. Na taj način se obezbeđuje da obrada dokumenta na klijentu bude autonoman proces sve do trenutka dok bibliotekar eksplicitno ne zatraži ažuriranje baze bibliografske građe. U tom trenutku klijent se obraća prvo serveru baze bibliografske građe radi njenog ažuriranja novim zapisom, a zatim serveru indeksa prosleđuje identifikator ažuriranog zapisa radi njegovog indeksiranja.

KLIJENT ZA PRETRAŽIVANJE, preko odgovarajućeg korisničkog interfejsa, obezbeđuje prihvat i sintaksnu validaciju upita korisnika kao i obradu odgovora na postavljeni upit [6]. Ako se vrši pretraživanje lokalne baze bibliografske građe klijent za pretraživanje upit prosleđuje serveru indeksa koji vrši procesiranje upita. Server indeksa kao rezultat pretraživanja klijentu vraća identifikatore zapisa koji su rezultat pretraživanja. Na osnovu dobijenih identifikatora klijent za pretraživanje od servera baze dobija podatke o dokumentima koji zadovoljavaju postavljeni upit, vrši obradu odgovora i njegovu prezentaciju korisniku.

Ako se vrši pretraživanje i udaljenih baza podataka, klijent za pretraživanje osim prosleđivanja upita lokalnom serveru indeksa izdaje i zahtev serveru sesije. Server sesije na osnovu zahteva i globalnog rečnika prosleđuje upite za pretraživanje drugim bazama. Dobijene rezultate server sesije prosleđuje klijentu za pretraživanje koji vrši obradu odgovora i njihovo prezentiranje korisniku. Na ovaj način obezbeđuje se logička jedinstvenost BIS-a i rešava problem uzajamne katalogizacije.

8.4. IMPLEMENTACIJA LOKALNOG SEGMENTA BIBLIOTEČKOG INFORMACIONOG SISTEMA

Implementacija lokalnog BIS-a izvršena je na osnovu specifikacije opisane u prethodnim poglavlјima. Obzirom da je Bibliotečka mreža projektovana kao heterogeni distribuirani informacioni sistem, pri implementaciji se vodilo računa o mogućnosti prenošenja softvera na različite računarske platforme.

Softver je realizovan na programskom jeziku C, a da bi se obezbedila njegova nezavisnost od računarske platforme korišćen je transformacioni pristup u implementaciji, pri čemu se viši nivo softvera preko nižeg (transformacionog) nivoa adaptira na konkretno implementaciono okruženje. Transformacioni nivo softvera se oslanja na alate i biblioteke funkcija konkretnog računarskog sistema i posebno se razvija za svaku platformu, a do sada je razvijen za VAX/VMS i SCO UNIX okruženje.

Pored potrebe za prenosivošću softvera i specifični zahtevi za funkcionalnošću sistema bili su razlozi što je za BIS u potpunosti razvijen poseban sistem za korisnički interfejs. Sistem za generisanje korisničkog interfejsa baziran je na događajima, porukama i objektima [7-13]. Objekti su prozori koji pripadaju nekoj klasi, pri čemu se klasom prozora definiše njegov izgled na ekranu i akcije koje se nad njim mogu izvršiti. Definisano je dvadeset klasa prozora u hijerarhijskoj strukturi uz mogućnost nasleđivanja osobina od nadređene klase. Korišćen je mehanizam upravljanja objektima preko poruka koje se generišu događajima. Sistem obezbeđuje mogućnost izgradnje kontekstno osetljivog help sistema, što olakšava korišćenje aplikacija i omogućuje jednostavnu obuku korisnika.

Transformacija sistema za generisanje korisničkog interfejsa na konkretnoj računarskoj platformi vrši se preko biblioteke funkcija za rukovanje uređajima, koja se oslanja na sistemske biblioteke konkretnog operativnog sistema (npr. na VAX-u Screen Management biblioteka).

Obzirom na specifične zahteve bibliografskih i dokumentacionih ustanova pri obradi bibliografskih dokumenata i heterogenost računarske opreme koje poseduju, razvijen je i dizajner fontova [14-17]. Time je omogućen unos bibliografske građe i rariteta od nacionalnog značaja korišćenjem različitih pisama na različitim računarskim platformama.

Implementirana je baza podataka u mrežnom modelu na sistemu za upravljanje bazom podataka (SUBP) db_VISTA [18]. Baza obuhvata model podataka standarda za održavanje (UNIMARC) (Slika 8.3), model podataka

za radno okruženje bibliotekara (Slika 8.4) i model podataka baze bibliografske građe (Slika 8.5). Takođe su implementirane i funkcije za održavanje ove baze [19-26].

U ovom trenutku jedino ograničenje BIS-a u pogledu njegove prenosivosti na druge računarske sisteme je odabrani SUBP db_VISTA, ali je on inače raspoloživ na više operativnih sistema što umanjuje značaj ove činjenice. Namena je da se i na operacije za rukovanje bazom podataka dosledno primeni transformacioni pristup.

8.5. ZAKLJUČAK

Specifikacija informacionih zahteva za Bibliotečki informacioni sistem (BIS) izvršena je primenom usvojenih standarda za razvoj otvorenih sistema. Funkcije ovog sistema su Definisanje standarda za obradu i pretraživanje, Formiranje bibliografske građe, Bibliotekarsko izvestavanje i dokumentovanje, Korisničko pretraživanje i Cirkulacija bibliotečkih dokumenata.

Implementirane su prve dve funkcije BIS-a. U okviru prve funkcije usvojen je UNIMARC format za razmenu podataka i struktuiranje baze bibliografske građe. Funkcija Formiranje bibliografske građe omogućava obradu dokumenata, struktuiranje zapisa po usvojenom standardu i formiranje baze bibliografske građe.

Integracijom funkcija Definisanje standarda za obradu i pretraživanje i Formiranje bibliografske građe sa softverom za indeksiranje i pretraživanje dokumenata, opisanim u Poglavlju 6, formirana je prva verzija programa za BIS. Ona omogućava formiranje baze bibliografske građe, indeksiranje i pretraživanje.

U toku je implementacija funkcija Bibliotekarsko izvestavanje i dokumentovanje, Korisničko pretraživanje i Cirkulacija bibliotečkih dokumenata.

LITERATURA

- [1] Koncepcija razvoja sistema naučnih i tehnoloških informacija u Republici Srbiji, Ministarstvo za nauku i tehnoloski razvoj Republike Srbije, Beograd 1991. (Rukovodilac projekta B. Lazarević).

- [2] Plan razvoja bibliotečke mreže Srbije, Ministarstvo za nauku i tehnološki razvoj Republike Srbije, Beograd 1992. (Rukovodilac projekta D. Surla).
- [3] Razvoj softvera za lokalni segment bibliotečke mreže SNTIS za VMS i UNIX okruženje, Ministarstvo za nauku i tehnološki razvoj Republike Srbije, Beograd 1994. (Rukovodilac projekta D. Surla).
- [4] UNIMARC manual: bibliographic format/ International Federation of Library Associations and Institutions, IFLA Universal Bibliographic Control and International MARC Programme, New Providence, London 1994.
- [5] Surla, D., Konjovic, Z., *Distribuirani unos podataka u bibliotečkom informacionom sistemu*, Zbornik apstrakata, SinfoN'95, Zlatibor, 1995, str. 2.
- [6] Todorić, D., *Specifikacija zahteva korisničkog interfejsa za pretraživanje bibliografske građe*, Zbornik apstrakata, SinfoN'95, Zlatibor, 1995, str. 3.
- [7] VMS RTL Screen Management (SMG\$) Manual, Digital Equipment Corporation, Maynard, Massachusetts, 1988.
- [8] Introduction to VMS System Routines, Digital Equipment Corporation, Maynard, Massachusetts, 1990.
- [9] The C++ Programming Language, Bjarne Stroustrup, New Jersey, 1994.
- [10] Tričković, I., *Generisanje ekranskih formi na VAX/VMS operativnom sistemu*, Zbornik radova, SinfoN'94, Zlatibor, 1994, str. 11-15.
- [11] Tričković, I., Lomić, M., *Implementacija različitih fontova u korisničkom interfejsu na računaru VAX/VMS*, Zbornik radova, YU INFO, Brezovica, 1995, (u štampi).
- [12] Tričković, I., *Implementacija korisničkog interfejsa na operativnom sistemu SCO UNIX*, Zbornik sažetaka, PriM'95, Budva, 1995, str. 100.
- [13] Tričković, I., *Implementacija sistema za generisanje ekranskih formi na različitim računarskim platformama*, Zbornik proširenih sažetaka, ITP'95, Novi Sad, 1995, str. 94-95.
- [14] VT 420 Programmer, Digital Equipment Corporation, Maynard, Massachusetts, 1992.
- [15] Installing and Using The VT 420 Video Terminal, Digital Equipment Corporation, Maynard, Massachusetts, 1989.

- [16] Lomić, M., *Kreiranje karaktera za terminal VT 420 na VMS operativnom sistemu*, Zbornik sažetaka, PriM'95, Budva, 1995, str. 78.
- [17] Lomić, M., *Softverski paket za generisanje različitih fontova na računarskoj platformi VAX/VMS*, Zbornik proširenih sažetaka, ITP'95, Novi Sad, 1995, str. 96-97.
- [18] db_Vista III, Database management system for “C” language application software development, Raima Corporation, USA 1992.
- [19] Surla, D., Bender, M., *Baze podataka sekundarnih dokumenata*, JISA info, 6/1994, str. 44-46.
- [20] Holo, I., *Implementacija funkcija za unos podataka Bibliotečkog informacionog sistema*, Zbornik radova, SinfoN'94, Zlatibor, 1994, str. 37-42.
- [21] Holo, I., Bender, M., *Implementacija bibliotečkog okruženja za unos bibliografske građe*, Zbornik sažetaka, PriM'95, Budva, 1995, str. 71.
- [22] Holo, I., Todorović, D., *Komandni jezik bibliotečkog informacionog sistema*, Zbornik proširenih sažetaka, ITP'95, Novi Sad, 1995, str. 54-55.
- [23] Holo, I., *Sintaksa komandnog jezika bibliotečkog informacionog sistema*, Zbornik apstrakata, SinfoN'95, Zlatibor, 1995, str. 3.
- [24] Savić, N., Vujić, T., *Struktura kataloških listića u bibliotečkom informacionom sistemu*, Zbornik proširenih sažetaka, ITP'95, Novi Sad, 1995, str. 62-63.
- [25] Vujić, T., Todorović, D., *Struktura podataka UNIMARC standarda*, Zbornik sažetaka, PriM'95, Budva, 1995, str. 101-102.
- [26] Vujić, T., Savić, N., *Algoritam za generisanje kataloških listića bibliotečkog informacionog sistema*, Zbornik sažetaka, SinfoN'95, Zlatibor, 1995, str. 4.

Poglavlje 9.

REČNIK PODATAKA SNTIS

Rečnik podataka (koji se često naziva katalog, direktorijum ili meta-baza podataka) se u opštem slučaju definiše kao baza podataka koja opisuje ostale objekte u informacionom sistemu. U SNTIS, Rečnik podataka opisuje logičku strukturu sistema - uključene institucije, njihove uloge i međusobne odnose, dostupne baze podataka i informacione servise, kadrove, korisnike, kao i njegovu fizičku strukturu - fizičke mreže, mrežne čvorove, fizičke karakteristike njihovog povezivanja i funkcije koje imaju u mreži, računarsku i komunikacionu opremu, odnos prema institucijama, itd.

Definicija Rečnika podataka kao "informacionog sistema o informacionom sistemu" već sama po sebi govori da je njegova osnovna namena da bude vodič kroz neki informacioni sistem. Rečnik podataka SNTIS treba da odgovori na širok skup mogućih korisničkih i projektantskih zahteva, bilo direktno, pretraživanjem informacija sadržanih u bazi Rečnika, ili indirektno, pružajući interpretativnu podršku izvršavanju aplikacija koje mogu da se izvršavaju samo ako su "vođene" Rečnikom. Time se opisuje dvostruka uloga Rečnika podataka SNTIS: "pasivna" u kojoj je Rečnik podataka korisnicima i projektantima na raspolaganju kao specifična baza podataka koja opisuje ostale objekte u sistemu SNTIS i "aktivna" koja omogućava izvršavanje rečnički vođenih aplikacija za pristup heterogenim bazama podataka, pružajući im informacije o tome na koji je način apstraktna specifikacija baza podataka i programa implementirana u konkretnim okruženjima. Pasivna uloga Rečnika podataka je ostvarenje funkcije vodiča ili informatora o sistemu, a aktivna je osnova za ostvarenje željene transparentnosti sistema u odnosu na distribuciju i različite načine implementacije.

U prvom delu ovog poglavlja opisuje se arhitektura Rečnika u skladu sa ANSI i ISO standardima, dok je u drugom delu u osnovnim crtama

prikazano kako se vrši formiranje i pretraživanje Rečnika podataka. Treće poglavlje opisuje sadržaj Rečnika, gde je najpre prikazana struktura Rečničke baze podataka preko pojedinih podmodela objekti-veze, a zatim i izgled baze sekundarnih dokumenata Rečnika gde je dat format nekoliko tipova dokumenata koji se iz Rečnika dobijaju SPI pretraživanjem. Četvrti deo poglavlja diskutuje aktivnu ulogu Rečnika, posebno u transparentnom pretraživanju distribuiranog SPI. Poslednji, peti deo, opisuje implementaciju Rečnika na centralnom i lokalnom nivou i način njegove distribucije korišćenjem pojedinih Internet servisa.

9.1. ARHITEKTURA REČNIKA PODATAKA U SKLADU SA STANDARDIMA

Arhitektura Rečnika podataka SNTIS je definisana u skladu sa struktrom koju predlažu ANSI [1] i ISO [2] standardi, ne predstavljajući, pri tom, strogu implementaciju ni jednog od njih.

Način na koji je u Rečniku SNTIS ostvarena četvoronivoska struktura IRDS standarda prikazan je na Slici 9.1 - levi deo slike. U desnom delu je prikazan CASE alat ARTIST, kao jedna od mogućnosti formiranja i održavanja Rečnika.

Najviši hijerarhijski nivo u standardnoj četvoronivoskoj strukturi Rečnika podataka je Definicija Šeme Rečnika Informacionih Resursa, gde se usvajaju i definisu koncepti preko kojih će se opisati neki rečnik podataka. Za prikazivanje koncepata i dozvoljenih veza između njih, ANSI i ISO standardi preporučuju specifične verzije modela objekti-veze, a u Rečniku podataka SNTIS se koristi semantički najbogatiji Prošireni model objekti-veze (PMOV). On je primenjen i kao deo standardne metodologije projektovanja informacionih sistema i baza podataka u SNTIS, čime se postiže da se specifikacija (logička struktura) svih baza podataka u SNTIS iskazuje na isti način. Detaljniji prikaz PMOV se može naći u radu pod referencom [3].

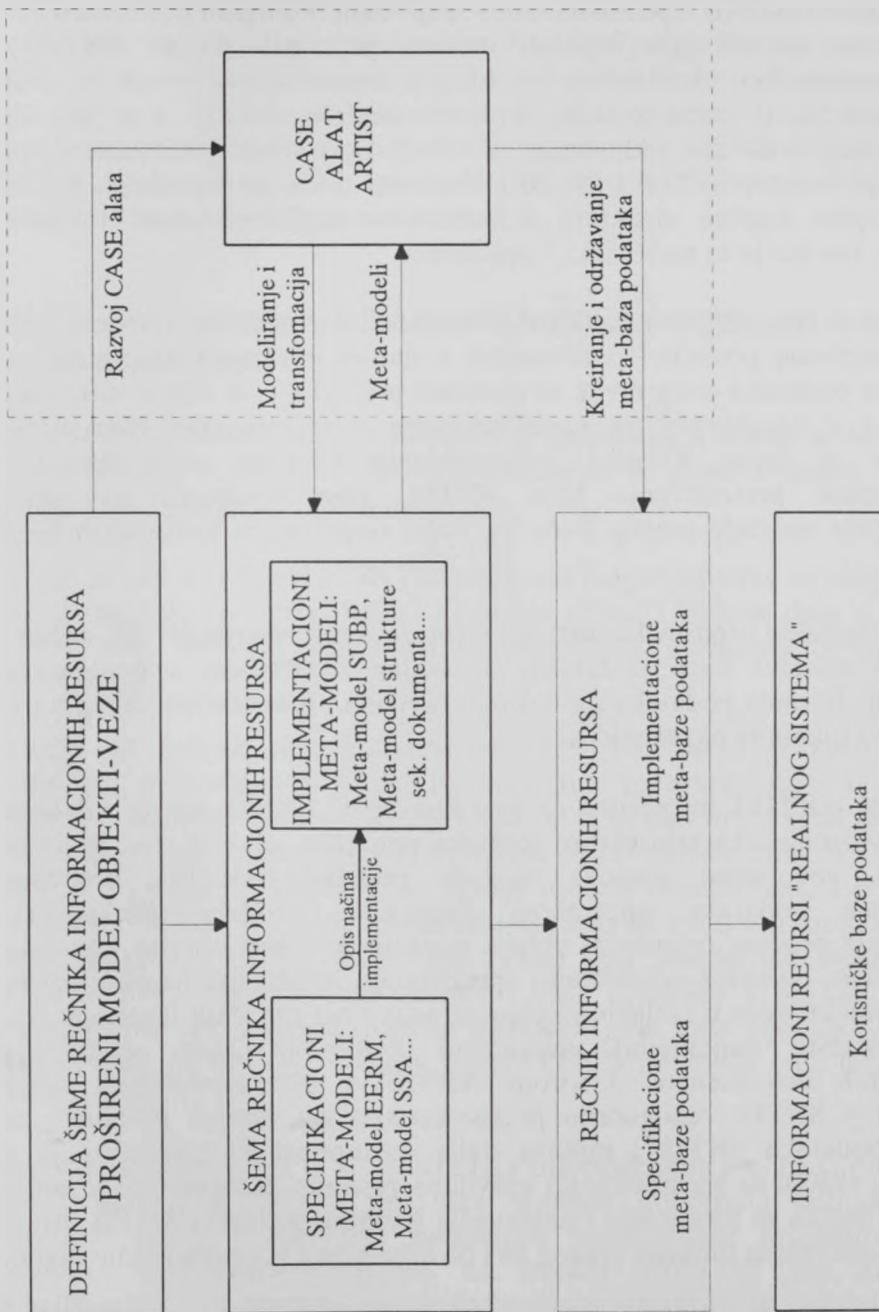
Treći hijerarhijski nivo je Šema Rečnika Informacionih Resursa i predstavlja opis konkretnog rečnika podataka pomoću koncepata definisanih na četvrtom nivou. Usvajanjem standardne metodologije projektovanja na četvrtom hijerarhijskom nivou, na trećem nivou se kao zajednički specifikacioni deo Rečnika nalazi meta-model objekti-veze (model objekti-veze modela objekti-veze). Tu su i različiti meta-modeli za opis logičkih struktura baza podataka u sistemu, modeli implementacije baza podataka,

njihove distribucije po čvorovima u računarskoj mreži SNTIS, strukture sekundarnih dokumenata itd. Treba uočiti da se na ovom nivou nalaze dva tipa meta-modela: (i) specifikacioni - koji daju kompletну, formalnu i konzistentnu specifikaciju logičkih modela pojedinih delova SNTIS i (ii) implementacioni - koji sadrže lokalne, implementacione zavisne modele delova sistema, tj. meta-modele implementacionih okolina u kojima se razvijaju baze podataka, uključujući pre svega opis primenjenih sistema za upravljanje bazama podataka (SUBP). Na ovom nivou se uspostavlja i veza između opšte logičke strukture i konkretno implementiranih lokalnih struktura, kao što je to na Slici 9.1 i prikazano.

Drugi nivo u hijerarhijskoj strukturi je Rečnik Informacionih Resursa, koji sadrži konkretne podatke o resursima u datom informacionom sistemu. Meta-baze podataka ovog nivoa se direktno pretražuju od strane korisnika sistema, a u kombinaciji sa specifikacionim i implementacionim meta-modelima iz Šeme Rečnika Informacionih Resursa omogućavaju i transparentno pretraživanje kroz SNTIS, koje obuhvata ne samo pretraživanje sadržaja samog Rečnika, već i raspoloživih korisničkih baza podataka.

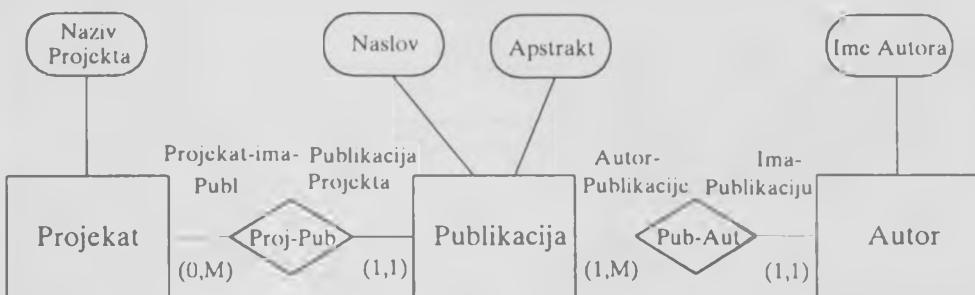
Prvi nivo Rečnika predstavlja instance informacionih resursa SNTIS, dakle - stvarne korisničke baze podataka, sa svojim sopstvenim sistemima za upravljanje bazama podataka i internim pravilima i mogućnostima pristupa, u skladu sa lokalnim okruženjem.

CASE alat ARTIST je razvijen na bazi koncepata definisanih na najvišem nivou četvoronivoske arhitekture Rečnika podataka, ali nije deo Rečnika podataka, već samo njegova moguća podrška. ARTIST podržava modeliranje podataka upotrebom Proširenog modela objekti-veze, modeliranje procesa primenom verzije strukturne sistem analize koja kao rezultat daje objektno-orientisanu specifikaciju aplikacija, transformaciju modela objekti-veze u različite implementacije i niz različitih izveštaja, gde treba posebno napomenuti mogućnost automatizovanog generisanja sekundarnih dokumenata. Izborom ARTIST-a za projektovanje baza podataka u SNTIS, omogućeno je automatsko generisanje podataka za Rečnik podataka SNTIS i njihova dalja dostupnost za pretraživanje u sistemu u skladu sa specificiranim pravilima pristupa. Naravno, to je samo jedan od načina za formiranje i održavanje Rečnika podataka SNTIS, što je jedna od obaveznih funkcija svakog SPI na lokalnom i kooperativnom nivou.



Slika 9.1. Standardna arhitektura Rečnika podataka

Ilustracija sadržaja Rečnika podataka ovakve arhitekture je data na Slici 9.2. Reč je o delu baze podataka o projektima Ministarstva za nauku i tehnologiju Republike Srbije, koji je prikazan modelom objekti-veze na Slici 9.2.a. ARTIST je upotrebljen za kreiranje ovog modela i za njegovu transformaciju u dva različita SUBP - relacioni i mrežni. Struktura sekundarnih dokumenata je takođe automatski kreirana upotrebom ovog CASE alata. Meta-modeli i meta-baze podataka (Slika 9.2.b-e) su predstavljeni u obliku relacija, jednostavnosti radi, čime se uopšte ne insistira na upotrebi relacionog modela u konkretnoj implementaciji [4,5].



9.2a. Primer modela objekti-veze

ENTITETI		VEZE		ATRIBUTI	
Ent ID	Naziv Entiteta	Veza ID	Naziv Veze	Ent 1	Ent 2
1	Entitet	1	ImaAtribut	1	2
2	Atribut	2	DomenPreslikavanja	1	4
3	Veza	3	RangPreslikavanja	1	4
4	Preslikavanje				

9.2b. Specifikacioni meta-model - Meta-model Proširenog modela objekti-veze

ENTITETI		VEZE		ATRIBUTI	
Ent ID	Naziv Entiteta	Veza ID	Naziv Veze	Ent 1	Ent 2
1	BazaPodataka	1	Baza-Cvor	1	2
2	Cvor	2	Baza-Tabela	1	3
3	Tabela	3	Baza-Rekord	1	5
4	Kolona	4	Tabela-Kolona	3	4
5	Rekord	5	Vlasnik Seta	6	5
6	Set	6	Clan Seta	6	5
7	Dokument	7	Baza-Dokument	1	7
8	Polje	8	Dokument-Polje	7	8

9.2c. Implementacioni meta-model - Opis implementacionog okruženja baze podataka

ENTITETI

<u>Ent</u>	<u>Naziv</u>
<u>ID</u>	<u>Entiteta</u>
e1	Projekat
e2	Publikacija
e3	Autor

ATRIBUTI

<u>Atr</u>	<u>Naziv</u>	<u>Ent</u>
<u>ID</u>	<u>Atributa</u>	<u>ID</u>
a1	NazivProjekta	1
a2	Naslov	1
a3	Apstrakt	1
a4	ImcAutora	3

PRESLIKAVANJA

<u>Presl</u>	<u>Naziv</u>	<u>Inv.</u>	<u>Kard.</u>	<u>Ent</u>
<u>ID</u>	<u>Preslikavanja</u>			<u>ID</u>
p1	Projekat-ima-Publ	m2	0,M	e1
p2	PublikacijaProjekta	m1	1,1	e2
p3	Autor-Publikacije	m4	1,M	e2
p4	Ima-Publikaciju	m3	0,M	e3

9.2d. Specifikaciona meta-baza podataka
ČVOR

<u>Čvor</u>	<u>Naziv</u>	<u>Adresa</u>
<u>ID</u>	<u>Čvora</u>	<u>Čvora</u>
n1	UBBG	147.91.8.6
n2	FON	147.91.2.1

BAZA PODATAKA

<u>BP</u>	<u>Naziv BP</u>	<u>SUBP</u>	<u>Čvor</u>
<u>ID</u>			
b1	Projekti	Relational	n1
b2	InfoTech	CODASYL	n2

TABELA

<u>Tab</u>	<u>Naziv Tabele</u>	<u>BP</u>
<u>ID</u>		
t1	Projekat	b1
t2	Publikacija	b1
t3	Autor	b1
t4	AutorPublikacije	b1

KOLONA

<u>Kol</u>	<u>Naziv Kolone</u>	<u>Tab</u>
<u>ID</u>		<u>ID</u>
c1	Plme	t1
c2	Naslov	t2
c3	Apstrakt	t2
c4	Almc	t3

DOKUMENT

<u>Dok</u>	<u>Naziv</u>	<u>DB</u>
<u>ID</u>	<u>Dokumenta</u>	
d1	ProjekatDok	b1
d2	InfoTechDok	b2

POLJE

<u>Polje</u>	<u>Naziv Polja</u>	<u>Dok</u>
<u>ID</u>		<u>ID</u>
f1	NazivProjekta	d1
f2	Naslov	d1
f3	Apstrakt	d1
f4	AutorIme	d1

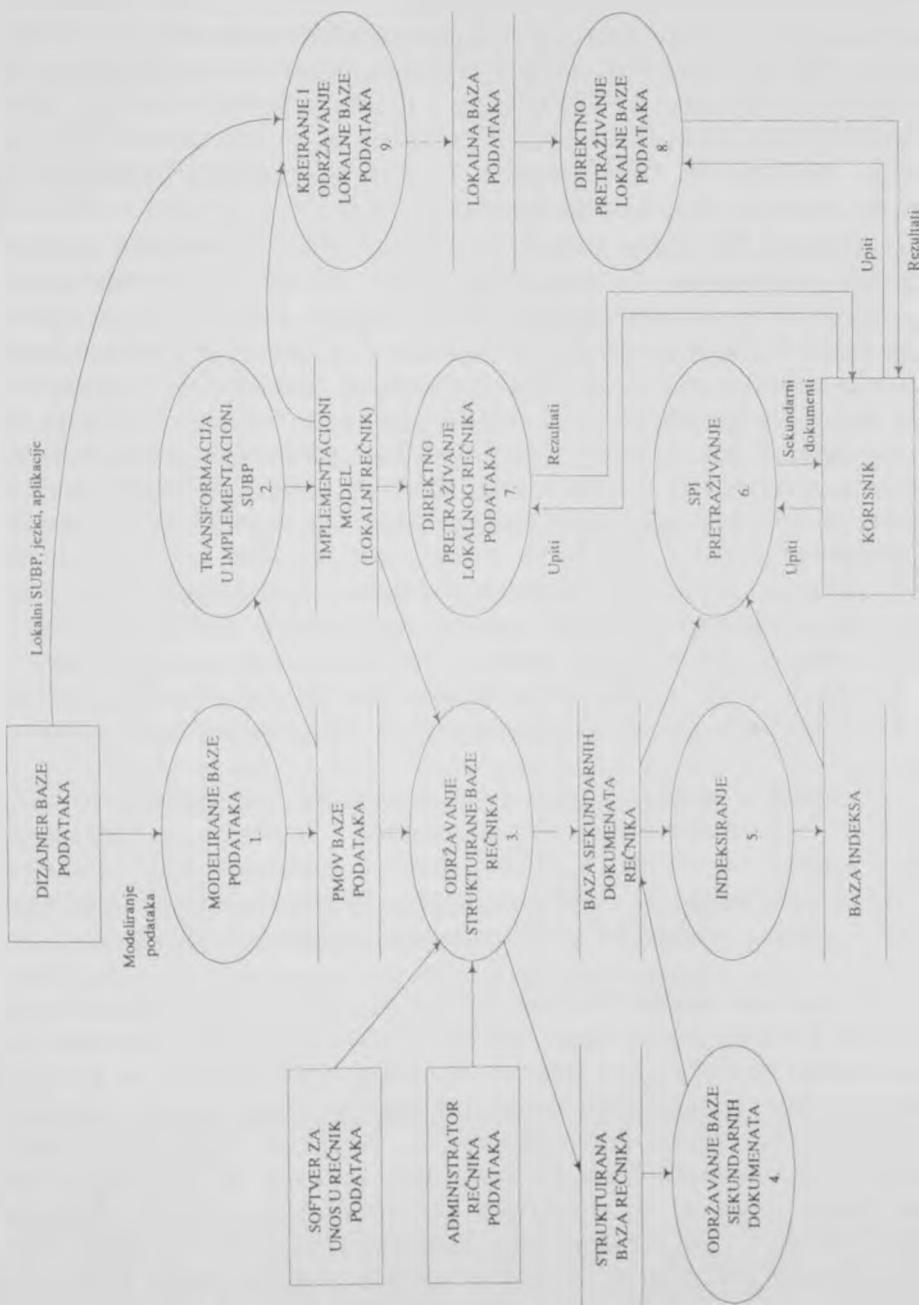
Ostale tabele za specifične SUBP

9.2e. Implementaciona meta-baza podataka

Slika 9.2. Primer sadržaja Rečnika podataka

9.2. FORMIRANJE I PRETRAŽIVANJE REČNIKA PODATAKA

Na Slici 9.3 dat je dijagram toka podataka koji logički prikazuje popunjavanje, indeksiranje i pretraživanje dela Rečnika podataka za opis baza podataka.



Slika 93 Formiranje i preraživanje Rečnika podataka

Dizajner kreira model podataka (PMOV) za svoju bazu (Proces 1.) i prevodi ga u određeno implementaciono okruženje i SUBP (Proces 2.). Na osnovu modela baze podataka i sadržaja lokalnog rečnika podataka formira se sekundarni dokument koji sadrži opšte karakteristike lokalne baze podataka i strukturu "Ravne baze dokumenata". Takav dokument se preslikava u struktuiranu bazu Rečnika SNTIS (Proces 3.) i odatle posle uvrštava u Bazu sekundarnih dokumenata (Proces 4.). Isprekidanom linijom između procesa održavanja struktuirane baze Rečnika i baze sekundarnih dokumenata Rečnika se potencira činjenica da na ulazu u ovaj proces podaci za Rečnik dolaze u traženoj SPI formi i da bi se mogli i direktno uključiti u bazu sekundarnih dokumenata. Razlozi zbog kojih se primljeni dokumenti ipak najpre preslikavaju u struktuiranu bazu Rečnika bice izneti u opisu implementacije Recnika podataka, u poglavljiju 9.5. Proces održavanja baze sekundarnih dokumenata je, ustvari, priprema za indeksiranje i razvija se posebno za svaku specifičnu bazu podatka (pa i za Rečnik), u skladu sa strukturom baze. Njime se složena struktura baze podataka - primarne baze dokumenata, "izravnava" u jednostavnu strukturu koja je sačinjena iz liste polja. Detaljan prikaz strukture Rečnika podataka za opis strukture "ravnih baza dokumenata" dat je modelom objekti-veze na Slici 5.4, u poglavljiju 5.1.1. "Formiranje i održavanje Rečnika podataka". Treba naglasiti da se za deo Rečnika podataka koji se bavi opisom komponenti SNTIS ravnjanje u bazu sekundarnih dokumenata izvršava preslikavanjem u redundantnu, stalnu kolekciju sekundarnih dokumenata, dok se za preostali sadržaj Rečnika preslikavanje u ravnu strukturu obavlja tek na zahtev i privremeno.

Ako se za proces dizajniranja baze podataka koristi CASE alat ARTIST, obezbeđena je automatska transformacija modela u izabранo implementaciono okruženje i SUBP, kao i kreiranje odgovarajućeg sekundarnog dokumenta za Rečnik podataka. SPI dokument za opis baze podataka se kreira i upotrebom softvera za unos u Rečnik podataka.

Proces 5. je indeksni modul SPI, koji na osnovu sadržaja baze sekundarnih dokumenata Rečnika kreira bazu indeksa. Baza indeksa se pretražuje na svaki korisnički zahtev, a na osnovu rezultata pretraživanja se iz baze sekundarnih dokumenata preuzimaju i korisniku "isporučuju" oni koji zadovoljavaju postavljeni upit (Proces 6). Koristeći podatke iz baze sekundarnih dokumenata Rečnika podataka, korisnik sa odgovarajućim pravima može nastaviti sa direktnim pretraživanjem lokalnih Rečnika podataka (Proces 7.) i/ili lokalnih baza podataka (Proces 8). To može biti i SPI pretraživanje, pri čemu se pretražuje lokalno formirana baza indeksa, a dokumenta se preuzimaju iz lokalne baze, pri čemu se njihovo formiranje obavlja na osnovu pravila preslikavanja iz lokalne strukture u strukturu

dokumenta i definisani format dokumenta koji se pamti u lokalnom rečniku podataka. Naravno, pretraživanje u lokalnu se može nastaviti i upotrebom internog upitnog jezika korisničke baze podataka i prilagođavanjem pravilima lokalnog SUBP-a. Kreiranje i održavanje korisničkih baza podataka (Proces 9.) je potpuno pod kontrolom lokalnog okruženja.

9.3. SADRŽAJ REČNIKA

Rečnik podataka SNTIS, ostvarujući pasivnu ulogu informatora o sistemu, obuhvata četiri osnovna skupa podataka: o institucijama, o mrežnim čvorovima, o opštim karakteristikama baza podataka, o strukturi baza podataka za pretraživanje. Svaki od ovih delova Rečnika je opisan modelom objekti-veze i može se implementirati na različite načine na lokalnom, kooperativnom i centralnom nivou. U centralnom Rečniku za implementaciju je iskorišćena db_VISTA mrežna baza podataka, pri čemu je, za potrebe pretraživanja, svaki od navedenih "pasivnih" delova Rečnika raspoloživ u obliku jednog tipa sekundarnog dokumenta.

Deo baze Rečnika podataka je posvećen opisu različitih standarda koji omogućavaju rad u heterogenom SPI. Time su obuhvaceni: standardi za razmenu podataka, standardi za opis strukture baze za formiranje i održavanje dokumenata, standardi za strukturiranje baze podataka za pretraživanje i standardi za izveštavanje i publikovanje. Tu su i opisi pojedinačnih komponenti sistema preko tih standarda. Opisi pojedinih standarda i primeri njihove primene su detaljno dati u poglavljima koja se na njima baziraju.

U ovom poglavlju će biti prikazani modeli objekti-veze za opis mrežnih čvorova i opis opštih karakteristika baza podataka, a zatim i struktura sekundarnih dokumenata koji odgovaraju ovim kolekcijama unutar Rečnika [5].

9.3.1. Rečnik podataka za opis opštih karakteristika baza podataka u SNTIS

S obzirom na činjenicu da se ceo sistem SNTIS prevashodno razvija sa ciljem da omogući razmenu naučnih i tehnoloških informacija, verovatno je najvažniji deo Rečnika podataka onaj koji opisuje sadržaj i strukturu baza podataka koje su prisutne i raspoložive u mreži SNTIS.

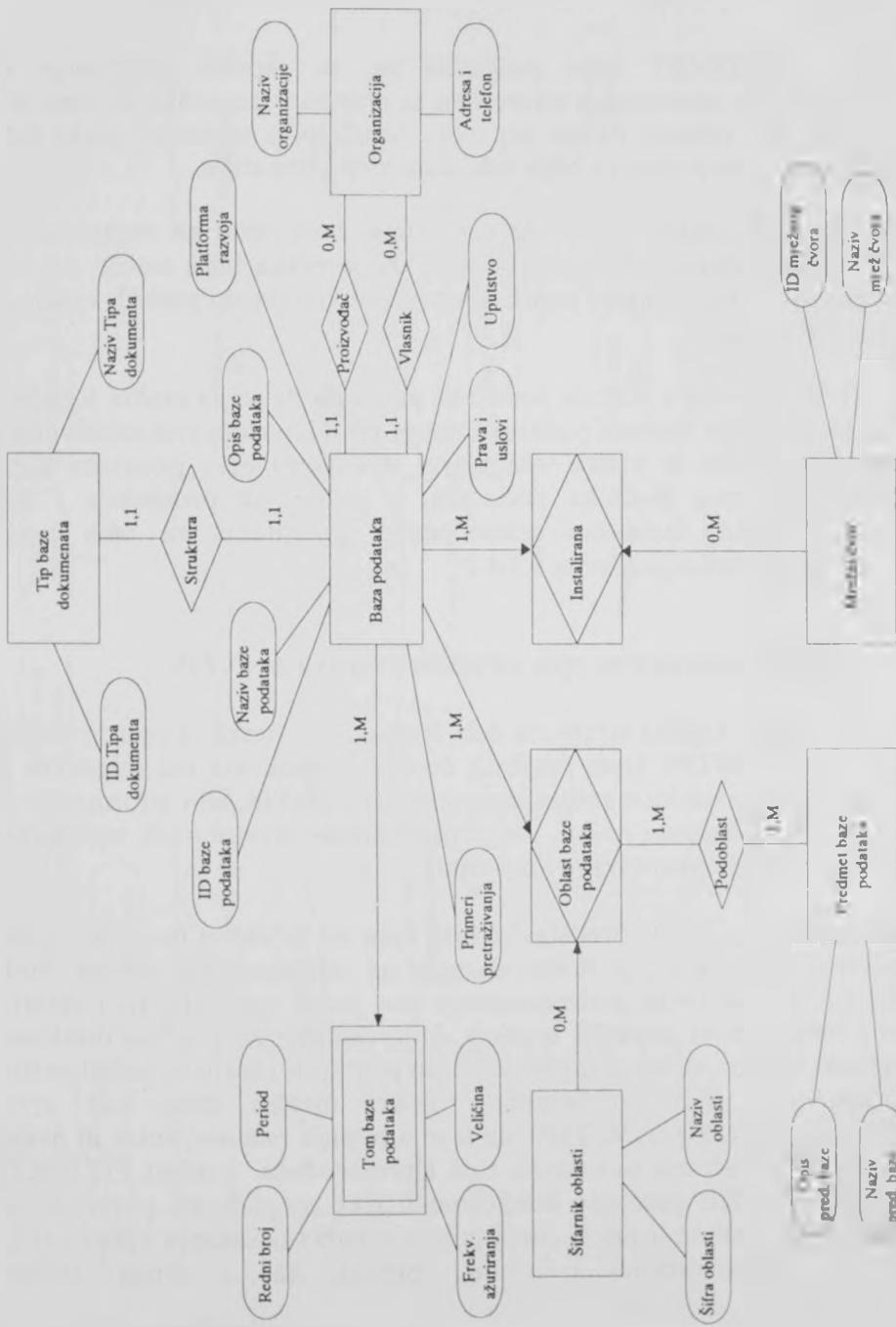
Na Slici 9.4 prikazan je PMOV koji predstavlja logičku strukturu dela Rečnika podataka za opis opštih karakteristika baza podataka SNTIS.

Detaljan opis baze podataka obuhvata niz tehničkih, opisnih i administrativnih detalja. PLATFORMA RAZVOJA je grupni atribut koji navodi: primjenjeni SUBP, jezik za pretraživanje baze podataka, izvorni jezik u kom su pisane aplikacije za pristup bazi, način na koji je konstruisan korisnički interfejs, operativni sistem ili više njih na kojima data baza podataka funkcioniše. Softver za lokalni unos u Rečnik podataka nudi spisak najfrekventnijih SUBP i upitnih jezika.

Mogućnost pristupa bazi podataka se detaljno opisuje navođenjem PRAVA I USLOVA KORIŠĆENJA. Prava se odnose na prioritete koji se daju korisnicima ili institucijama, a uslovi obuhvataju ne samo detalje oko toga na koji način se stiče pravo pristupa, već i informacije o dostupnosti servisa baze podataka određenim klasama korisnika, na koji način i pod kojim tehničkim pretpostavkama. UPUTSTVO ZA KORIŠĆENJE obuhvata kratki tekst i referencu na lokaciju u SNTIS na kojoj se može potražiti detaljno uputstvo za korisnički rad. Uputstvo za korišćenje je dopunjeno i atributom PRIMERI PRETRAŽIVANJA, gde se navode reprezentativne mogućnosti pretraživanja u skladu sa lokalnom implementacijom.

Direktno pretraživanje baze podataka proizvoljnog tipa zahteva poznavanje njene strukture. Vezom ka objektu TIP BAZE DOKUMENTA dobijaju se informacije o strukturi same baze podataka (baze dokumenata) i o strukturi baze podataka za pretraživanje. Naime, svaka baza podataka u SNTIS se posmatra kao baza dokumenata jednog tipa, s tim što se opis tih dokumenata daje primenom standarda za opis dokumenata koji je u skladu sa načinom implementacije baze. Svaki tip dokumenta ima svoj jedinstveni naziv i identifikator i oni predstavljaju deo opsteg opisa baze, omogućavajući referencu na opis strukture baze dat na neki od standardnih načina. Objekat TIP BAZE DOKUMENATA je deo Rečnika podataka za opis baze podataka za pretraživanje, kao što je to prikazano na modelu objekti-veze na Slici 5.4 u poglavlju 5.1.1. Kompletna struktura baze podataka za pretraživanje (desni deo slike 5.4) ulazi u sastav sekundarnog dokumenta Rečnika za opis baza podataka.

Rečnik obuhvata i popis svih organizacija koje su uključene u SNTIS, a u vezi sa bazama podataka od interesa su informacije o proizvođaču i vlasniku. Baza podataka o organizacijama u SNTIS je jedna od baza podataka koje se razvijaju u okviru Upravljačke mreže SNTIS.



Slika 9.4. Model objekti-veze za opšte opštih karakteristika baza podataka

Baza se instalira na jednom ili više mrežnih čvorova i na svakom od njih se navodi skup korisniku dostupnih servisa za rad sa bazom, što je detaljno prikazano na PMOV koji prikazuje deo Rečnika za opis mrežnih čvorova u SNTIS.

OBLAST i PREDMET baze podataka su, sa aspekta grupisanja i pretraživanja, među značajnijim atributima za opis baze podataka. Oblasti se grupišu u šifarnik oblasti, fiksan po svom sadržaju, a u okviru svake od oblasti predmet baze podataka bliže određuje njenu tematiku.

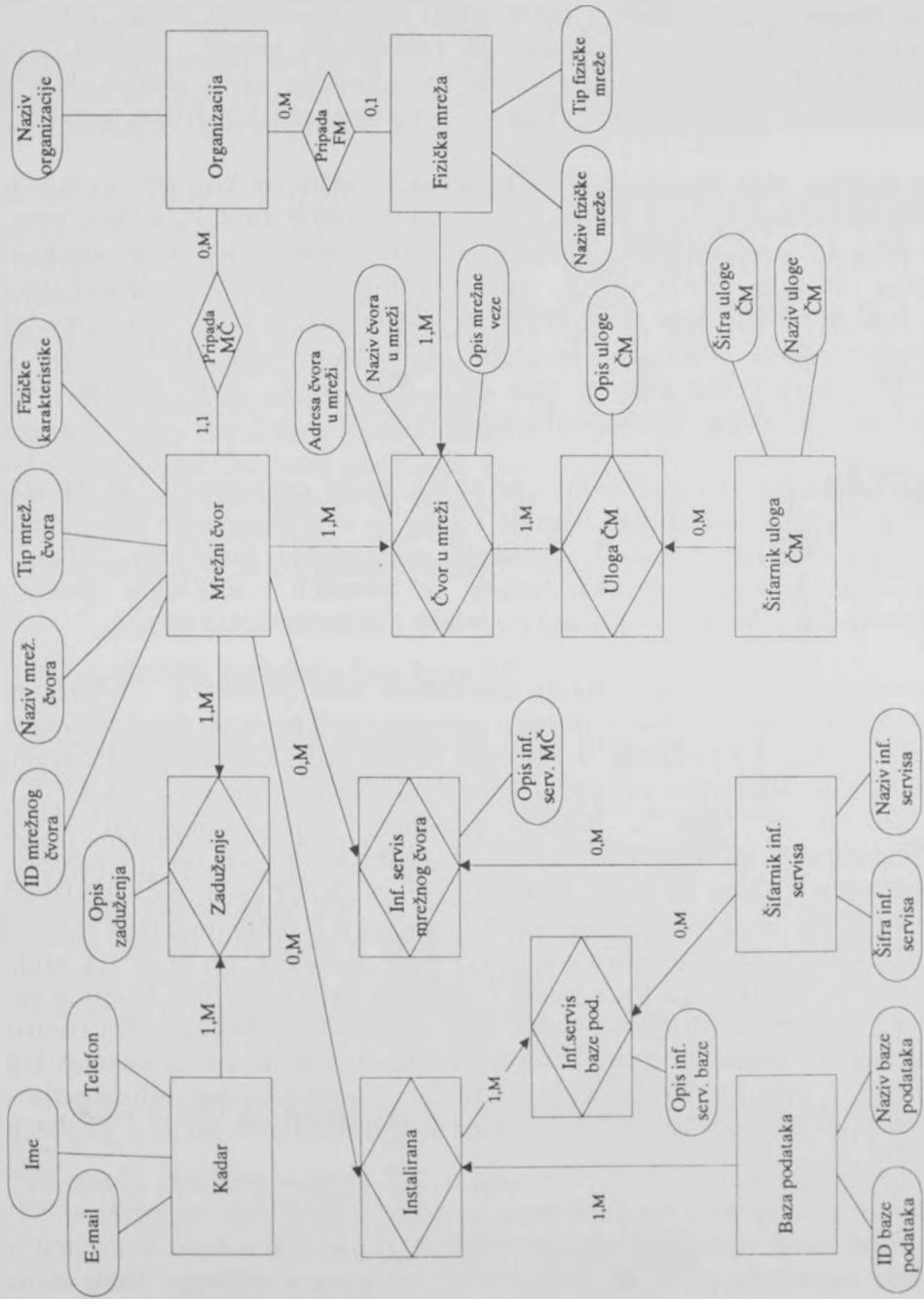
Statistika baza podataka se vodi po tomovima, po ugledu na organizaciju DIALOG baza podataka. Podrazumeva se da se svaka baza sastoji bar iz jednog toma, a za svaki tom se pamti obuhvaćeni vremenski period, veličina i frekvencija ažuriranja.

Softver za lokalni unos u Rečnik podataka predviđa da se na svakoj lokaciji podaci o raspoloživim bazama podataka unose prvi. Unosom svih zahtevanih podataka, automatski se kreira sekundarni dokument baze podataka koji postaje deo lokalnog Rečnika podataka, a potom se prosleđuje i do centralnog Rečnika. Struktura sekundarnog dokumenta za opis baza podataka je prikazana u poglavljju 9.3.4.1.

9.3.2. Recnik podataka za opis mrežnih čvorova u SNTIS

Na Slici 9.5. je data logička struktura dela Rečnika podataka za opis mrežnih čvorova u mreži SNTIS. Opis mrežnih čvorova obuhvata niz tehničkih i administrativnih podataka o računarima u sistemu SNTIS, ali i informacije u kojim se mrežama računar nalazi, koje informacione servise nudi, koje baze podataka su na njemu instalirane i dostupne.

Objekat MREŽNI ČVOR predstavlja bilo koju od tačaka u mreži kojoj se može pristupiti i na kojoj je dostupan neki od informacionih servisa. Pod mrežnim čvorom se ovde podrazumevaju pre svega računari, ali i ruteri, bridževi i ostali tipovi mrežnih uređaja, koji, doduše, ne pružaju direktnе informacione usluge, ali im se može direktno pristupiti i često su jedini način povezivanja dva ili više računara. Svaki mrežni čvor ima svoj IDENTIFIKATOR, NAZIV, TIP - da li je u pitanju računar, ruter ili neka druga oprema sa tačnom naznakom tipa i proizvođača. Atribut FIZIČKE KARAKTERISTIKE precizira konkretnе fizičke mogućnosti povezivanja datog čvora sa ostalima u mreži (broj portova, način obraćanja njima i sl.), raspoloživu komunikacionu i ostalu opremu, kao i druge fizičke



Slika 9.5. Model objekti-veze za opis mrežnih čvorova

karakteristike koje su od značaja za funkcionisanje datog uređaja i/ili mreže u celini.

Mrežni čvor može biti uključen u više fizičkih mreža, a u svakoj od njih ima svoju adresu, lokalni naziv i može imati više specifičnih uloga. Sifarnik navodi moguće uloge - primarni ili sekundarni server, "lokalni NIC (Network Information Center)", host za određenu vrstu servisa, klijent odnosno server za određenu vrstu usluga, gateway ka nekoj drugoj mreži itd.

Opis mrežne veze pri agregaciji ČM navodi način na koji se uspostavlja fizičko uključenje čvora u mrežu. Tip veze je određen time koja vrsta veze i koji protokol su zastupljeni. Vrste veze su, npr. komutirane ili iznajmljene serijske linije za WAN (Wide Area Network), i FDDI, Token Ring i Ethernet za LAN (Local Area Network). Za određene vrste veza mogu biti primenjeni samo neki od postojećih protokola: X.25, HDLC, UUCP, TCP/IP, SLIP, token passing ring, IPX, DECNET... Opis mrežne veze uključuje i navođenje obezbeđenih mrežnih servisa.

Svaka fizička mreža i mrežni čvor pripadaju nekoj organizaciji. Od interesa za bazu Rečnika podataka je otuda i koja je to organizacija, kao i njena struktura. Objekat KADAR određuje pojedinačna zaduženja (VRSTA ZADATKA), koja mogu biti vezana za tehničku i sistemsku podršku, komunikacije, administraciju baza podataka i informacionih usluga.

Na svakom čvoru se identifikuju instalirane baze podataka. Za bazu na čvoru čuvaju se podaci o dozvoljenim servisima nad njom, sa specifičnostima lokalne primene sadržanim u atributu OPIS VSBP. Servisi nad bazama podataka su objedinjeni sa opštim informacionim servisima u ŠIFARNIKU IS. Softver za unos u Rečnik podataka pruža mogućnost izbora najfrekventnijih informacionih servisa, s tim što se za svaki od njih navode i specifičnosti lokalne podrške. Kada se radi o identifikaciji baza podataka koje se nalaze na nekom čvoru, softver za unos u Rečnik podataka insistira da se najpre unesu informacije o samoj bazi podataka, da bi se tek onda, izborom iz liste lokalno instaliranih i opisanih baza podataka, izabrala ona koja se pridružuje mrežnom čvoru koji se trenutno obrađuje. Na osnovu podataka o jednom mrežnom čvoru formira se sekundarni dokument koji uključuje i opis strukture baze podataka za pretraživanje informacija o mrežnim čvorovima. Izgled tog sekundarnog dokumenta dat je u poglavljju 9.3.4.2.

9.3.3. Rečnik podataka za opis strukture baze podataka za pretraživanje

Standardni jezik za pretraživanje dokumenata, primjenjen u SPI, pretpostavlja "ravnu" strukturu baze dokumenata, pristupajući joj kao unapred definisanoj listi polja. Za potrebe pretraživanja se zato svaka baza podataka složene strukture preslikava u zahtevani oblik baze podataka za pretraživanje - kolekciju dokumenata koji se sastoje iz polja.

Model objekti-veze koji opisuje strukturu baze podataka za pretraživanje i njenu vezu sa bazama dokumenata prikazan je u poglavlju 5.1.1 "Formiranje i održavanje rečnika podataka" na Slici 5.4. Baza podataka se posmatra kao primarna baza dokumenata jednog tipa, kome se dodeljuje jedinstveni identifikator i naziv. U sadržaj dokumenta ulazi niz polja baze podataka, po određenom redosledu. Ta polja se preslikavaju u tipove polja baze podataka za pretraživanje, u jedno ili više njih, što se definiše lokalnim pravilima transformacije. Struktura baze podataka za pretraživanje je izvedena po ugledu na DIALOG baze podataka, kao i izgled sekundarnih dokumenata koji se dobijaju kao odgovor na korisničke upite. Postoji i mogućnost izbora formata za prikazivanje, čime se određuje podskup polja baze za pretraživanje koja se korisniku pružaju na uvid.

9.3.4. Rečnik podataka kao baza SPI

Rečnik podataka se može pretraživati i kao baza podataka SPI. Na osnovu sadržaja db_VISTA baze Rečnika formira se baza sekundarnih dokumenata koji po izgledu odgovaraju sadržaju tzv. Bluesheet dokumenata za DIALOG baze podataka. U ovom poglavlju je data forma Blusheet-a za opis opštih karakteristika baza podataka u SNTIS i za opis mrežnih čvorova, u skladu sa modelima objekti-veze koji su prikazani u prethodnim poglavljkima. Uz osnovni sadržaj ovih delova Rečnika, SPI dokumenti sadrže takođe i opis strukture baze podataka za pretraživanje, u vidu liste baznih i dodatnih indeksa koji se koriste za pristup odabranoj kolekciji podataka.

9.3.4.1. SPI dokument za opis opštih karakteristika baza podataka

Informacije sadržane u ovom SPI dokumentu (tip dokumenta 0002, sadržan u identifikatoru dokumenata iz ove kolekcije) su podeljene u nekoliko grupa. Prva obuhvata opšte karakteristike, "ličnu kartu" baze podataka, zatim slede tehničke karakteristike koje se odnose na platformu razvoja,

uputstvo za korišćenje i lokaciju baze podataka. Potom slede statisticke informacije, da bi se na kraju prikazala struktura baze podataka za pretraživanje, definisana preslikavanjem iz originalne baze dokumenata. Baza podataka za pretraživanje (BPR) se opisuje navođenjem liste definisanih polja i načina na koji se vrši pretraživanje po njima. Jedan od dokumenata u ovoj kolekciji je i onaj koji daje opis opštih karakteristika baze podataka Rečnika za opis opštih karakteristika baza podataka. Tamo će se ovde priloženi spisak polja pojaviti kao originalna BPR. Taj dokument je deo uputstva za korišćenje Rečnika podataka i javno je raspoloživ na Internet servisima u SNTIS.

REČNIK PODATAKA - OPIS OPŠTIH KARAKTERISTIKA BAZA PODATAKA

Identifikator dokumenta (AN):

Opšte karakteristike

Naziv baze podataka (NA):

Opis baze podataka (DE):

Proizvođač baze podataka (OR):

Adresa (AD):

Telefon (PH):

Vlasnik baze podataka (OW):

Adresa (AV):

Telefon (PV):

Oblast baze podataka (AR):

Predmet baze podataka (SC):

Tehničke karakteristike

Platforma razvoja: Sistem za upravljanje bazama podataka (DB):

Upitni jezik (QL):

Izvorni programski jezik (SL):

Korisnički interfejs (UI):

Operativni sistem (OS):

Korišćenje baze podataka

Prava i uslovi korišćenja (PU):

Uputstvo za korišćenje (UG):

Lokacija detaljnog uputstva: Mrežni čvor (HO): Putanja (PA):

Primeri pretraživanja (EX):

Instalirana na računaru: Naziv (NC):

Adresa (AC):

Statistika

Redni broj toma(TO):

Obuhvaćeni period (DA):

Veličina (FS):

Frekvencija ažuriranja (UF):

Opcije za pretraživanje*Bazni indeksi:*

ID Polja	Naziv polja	Sufiks	Redni broj	Način indeksiranja	Vrsta operatora	Sortiranje	Primer pretraživanja

Dodatni indeksi:

ID Polja	Naziv polja	Prefixi	Način indeksiranja	Vrsta operatora	Sortiranje	Primer pretraživanja

Predefinisani formati

Redni broj	Sadržaj

Opcije za pretraživanje dela Rečnika podataka za opis opštih karakteristika baza podataka*Bazni indeksi:*

ID Polja	Naziv polja	Sufiks	Redni broj	Način indeksiranja	Vrsta operatora	Sortiranje	Primer pretraživanja
NB	Naziv baze podataka	/NA /NB	1	reč, fraza	(w)		s BIMP/NA select Projekti*
DE	Opis baze podataka	/DE	2	reč	(f), (w)		select SNTIS s Rečnik (w) SNTIS
AR	Oblast baze podataka	/AR	3	fraza	(w)		select Obrazovanje s Racunari/AR
SC	Predmet baze podataka	/SC	4	reč, fraza			s Kursevi /SC select Performanse*

Dodatni indeksi:

ID Polja	Naziv polja	Prefixi	Način indeksiranja	Vrsta operatora	Sortiranje	Primer pretraživanja
AN	Identifikator dokumenta	AN=	reč			s AN=00010002000001 select AN=0001*
NB	Naziv baze podataka	NA= NB=	reč, fraza			s NB=BIMP
DE	Opis baze podataka	DE=	reč			s DE= "Računarske telekomunikacije"
OR	Proizvođač baze podataka	OR=	fraza			s OR=RUB s OR= Institut*
OW	Vlasnik baze podataka	OW=	fraza			s OW= Institut "Mihailo Pupin"
AR	Oblast baze podataka	AR=	fraza			s AR= Društvene nauke
SC	Predmet baze podataka	SC=	fraza			s SC= Ustav SRJ
DB	SUBP	DB=	fraza			s DB=ORACLE*
QL	Upitni jezik	QL=	reč			select QL=SQL

OS	Operativni sistem	OS=	reč, fraza			select OS= VM*, select OS=UNIX
TO	Tom baze podataka	TO=	reč			select (NA=Prijemni*) AND TO=1
DA	Obuhvaćeni period	DA=	reč, fraza			s DA=1993. -*
NC	Naziv mrežnog čvora	NC= AC=	reč, fraza			select NC=UBBG select AC=147.91.2.32

Predefinisani formati

Redni broj	Sadržaj
Format 1	Identifikator dokumenta i opšte karakteristike
Format 2	Format 1 + Tehničke karakteristike
Format 3	Format 1 + Korišćenje baze podataka
Format 4	Format 2 + Statistika
Format 5	Puni sadržaj, bez opcija za pretraživanje
Format 6	Opcije za pretraživanje
Format 7	Puni sadržaj sa opcijama za pretraživanje

9.3.4.2. SPI dokument za opis mrežnih čvorova

SPI dokumenti za opis mrežnih čvorova pripadaju tipu dokumenta 0001, što se vidi iz Identifikatora dokumenta (AN), gde je tip dokumenta predstavljen sa druge četiri cifre. Osnovne informacije o mrežnom čvoru obuhvataju njegov naziv, organizaciju kojoj pripada i fizičke karakteristike. Potom sledi niz informacija o fizičkim mrežama u koje se čvor uključuje - opis same mreže, identifikacija čvora u njoj i specifičnosti fizickog ostvarenja mrežne veze. Cela grupa ovih podataka se ponavlja za svaku mrežu u koju je čvor uključen. Sledеća grupa podataka se odnosi na baze podataka i informacione servise dostupne na čvoru. Na kraju, sledi lista kontakt-informacija za pojedince sa specifičnim zaduženjima. Pored toga, dat je i opis strukture baze podataka za pretraživanje za opis mrežnih čvorova.

REČNIK PODATAKA - OPIS MREŽNIH ČVOROVA

Identifikator dokumenta (AN):

Opšte karakteristike

Naziv mrežnog čvora (NC):

Vlasnik mrežnog čvora (OR):

Adresa (AD):

Tip mrežnog čvora (TC):

Telefon (PH):

Fizičke karakteristike (FK):

Ukličenost u mreže**Naziv mreže (FM):****Tip mreže (TM):****Vlasnik mreže (OW):****Opis mreže (DE):****Adresa čvora u mreži (AC):****Naziv čvora u mreži (CM):****Uloga čvora u mreži (UC):**Informacioni servisi i baze podataka**Šifra informacionog servisa (IS):** **Naziv informacionog servisa (NI):****Opis IS na čvoru (OI):****Naziv baze podataka na čvoru (NB):**Zaduženja**Ime i prezime (KD):****Telefon (PK):****E-mail adresa (EM):****Zaduženje (ZA):**Opcije za pretraživanje**Bazni indeksi:**

ID Polja	Naziv polja	Sufiks	Redni broj	Način indeksiranja	Vrsta operatora	Sortiranje	Primer pretraživanja
NC	Naziv mrežnog čvora	/NA /NC	1	reč, fraza			s UBBG/NC s ETF
TC	Tip mrežnog čvora	/TC	2	reč, fraza			s IBM/TC
FK	Fizičke karakteristike	/FK	3	reč	(w), (f)		s printer (f) laser*
FM	Fizička mreža	/FM	4	reč, fraza			s LOSTUD
DE	Opis mrežne veze	/DE	5	reč	(w), (f)		s Ethernet (f) TCP

Dodatni indeksi:

ID Polja	Naziv polja	Prefiksi	Način indeksiranja	Vrsta operatora	Sortiranje	Primer pretraživanja
AN	Identifikator dokumenta	AN=	reč			s AN=00020001000045 s AN=0003*
NC	Naziv mrežnog čvora	NA= NC=	reč, fraza			select NA=UBBG* select NC=ETF
OR	Vlasnik mrežnog čvora	OR=	reč, fraza			select OR=Ministarstvo*
TC	Tip mrežnog čvora	TC=	reč, fraza			s TC=Silicon Grafic*
FM	Fizičke mreža	FM=	reč, fraza			s FM=LOTEH
TM	Tip mreže	TM=	reč			select TM=BITNET
DE	Opis mrežne veze	DE=	reč	(w), (f)		select DE=FDDI select DE=optika
OW	Vlasnik mreže	OW=	fraza			s OW=Univerzitet*

AC	Adresa čvora u mreži	AC=	reč, fraza		s AC=147.91.1.1 select AC=160.99*
CM	Naziv čvora u mreži	CM=	reč, fraza		s CM=rcub.rcub.bg.ac.yu select CM=junis
UC	Uloga čvora u mreži	UC=	reč, fraza		s UC= Name server
NI	Naziv informacionog servisa	NI=	reč, fraza		s Obrazovanje korisnika
OI	Opis informacionog servisa	OI=	reč		select Kurs (f) ORACLE*
NB	Naziv baze podataka	NB=	reč, fraza		s NB = Prijemni ispit*
KD	Kadar	KD=	reč, fraza		s KD=Marko Markovic
ZA	Zaduženje	ZA=	reč, fraza	(w), (f)	s ZA= Administrator mreže

Predefinisani formati

Redni broj	Sadržaj
Format 1	Identifikator dokumenta i opšte karakteristike
Format 2	Format 1 + Uključenost u mreže
Format 3	Format 1 + Informacioni servisi
Format 4	Puni sadržaj, bez opcija za pretraživanje
Format 5	Opcije za pretraživanje
Format 6	Puni sadržaj sa opcijama za pretraživanje

9.4. REČNIKA PODATAK KAO AKTIVNI DEO SPI

Kao što je u uvodnom poglavlju rečeno, Rečnik podataka na svim nivoima distribucije, centralnom, kooperativnom i lokalnom, ima i aktivnu ulogu koja omogućava izvršenje Rečnikom vođenih aplikacija. Osnovna namena aktivnog dela Rečnika podataka je da obezbedi jedinstvenost pretraživanja na nivou celokupnog NTIS, primenom opšteg modela baze podataka i obezbeđenjem mehanizama preslikavanja između ovog modela, modela baza podataka pojedinih kooperativnih podistema i modela baza podataka lokalnih SPI.

Na lokalnom nivou, Rečnici podataka čuvaju opis lokalne baze i pravila preslikavanja iz lokalne u opštu strukturu SPI. Preslikavanje u opštu strukturu centralnog ili kooperativnog SPI podrazumeva da se u lokalnom Rečniku podataka definiše format dokumenta za preuzimanje podataka iz baze, uz detaljnu informaciju o tome koji se deo lokalne baze i na koji način preslikava u tu SPI formu.

Na kooperativnom i centralnom nivou Rečnika podataka čuva se opis svih lokalnih baza podataka u podsistemu, odnosno sistemu, uz definiciju opštih standarda koji se koriste u SPI. Aktivna uloga Rečnika podataka na ovim nivoima se odnosi na procesiranje upita čime se vrši izbor baza ka kojima će se upit rutirati. Time se ostvaruje funkcija transparentnog pretraživanja u SNTIS, dok se samo izvršavanje upita odvija lokalno i pod autonomnom kontrolom lokalnih Servera. Ovo poglavlje detaljno obrađuje samo ovaj aspekt aktivne uloge Rečnika, dok implementacija aktivnosti na lokalnom nivou predstavlja odgovornost administratora lokalnih baza podataka.

9.4.1. Izbor baze podataka za pretraživanje i rutiranje upita

Rečnik podataka, osim opisa opštih karakteristika baza podataka u sistemu, poseduje i informacije o strukturi njihovih baza podataka za pretraživanje, kao što je opisano kroz sadržaj Rečnika. Centralni Rečnik podataka poseduje globalnu bazu za pretraživanje celog sistema. Kao što je to već naglašeno u par navrata, istim tipom polja BPR se može opisati više različitih baza dokumenata, pri čemu u svakoj od tih baza polje BPR dobija svoje specifičnosti po pitanju pretraživosti i načina indeksiranja. Otuda se veliki deo korisničkih upita može izvršiti nad skupom postojećih baza podataka. Aktivna uloga centralnog i kooperativnih Rečnika podataka se sastoji u procesu analiziranja korisničkog upita i izbora baza podataka nad kojima se upit može izvršiti. Server rečnika SPI obavlja procesiranje upita i njegovo izvršenje na rečniku, a rezultat se vraća serveru sesije koji vrši rutiranje ka odabranim bazama podataka.

Pre nego što se pristupi opisu samog postupka analize korisničkih upita, treba izdvojiti nekoliko osnovnih postavki:

- Svaki korisnički upit se sastoji iz jednog ili više termova koji su povezani logičkim (*and*, *or*, *not*) ili blizinskim (w, f) operatorima.
- Termovi mogu biti kvalifikovani, što znači da se uz traženi sadržaj navodi i naziv polja po kome se zahteva pretraživanje, u sufiksnoj ili prefiksnoj notaciji, u zavisnosti od toga da li polje ulazi u bazni ili dodatni indeks.
- Nekvalifikovani termovi navode samo traženi sadržaj i tada se pretraživanje vrši nad skupom definisanih baznih indeksa, tako što se u procesu izvršavanja upita vrši transformacija takvog terma u niz termova oblika "Sadržaj/Bazni indeks", povezanih operatorom *or*. Provera ispunjenja nekvalifikovanog terma se zato nad svakom lokalnom bazom vrši na drugačiji način, jer se konsultuju lokalni bazni indeksi.

- Analiza korisničkog upita obuhvata samo kvalifikovane termove i logičke operatore. Blizinski operatori ne spadaju u domen strukture baze podataka za pretraživanje, već se bave samim sadržajem polja u dokumentu, zbog čega se analiza njihovog ispunjenja izvršava tek na nivou lokalnog pretraživanja. Isti slučaj je i sa pretraživanjem po nekvalifikovanim termovima, koji se takođe bave samo sadržajem, i transformacija u bazno-kvalifikovane termove se vrši lokalno.
- Specifičan tretman se primenjuje i na termove kojima prethodi *not*, koji se u SPI koristi kao binarni operator i za rezultat ima skupovnu razliku svojih operanada. Primena ovog operatora zahteva da njegov desni operand (term) ne bude tačan, što je ispunjeno u dva slučaja: ili polja koja se pojavljuju u termu ne postoje u bazi nad kojom se vrši pretraživanje, ili polja datog naziva postoje, ali nije ispunjen uslov zadatog sadržaja. Sa gledišta analize strukture korisničkog upita, *not(term)* je ispunjen bez obzira na to da li ciljna baza sadrži polja navedena u termu ili ne, i u tom kontekstu se tretira kao *and all*.

Postupak procesiranja upita se odvija u više koraka, čiji opis sledi. Radi pojašnjenja rezultata primene svakog od koraka, kao primer je uzet jedan apstraktни korisnički upit sledeće formulacije:

```
select ( ( T1=a and T2=b or c/T3 ) and ( d and T4=e(w)f ) ) not ( g/T4 )
       and ( h/T5 or i )
```

- Korisnički upit se analizira po termovima, ne vodeći računa o redosledu izvršavanja koji nameće primjenjeni logički operatori; izdvajaju se samo kvalifikovani termovi, odnosno nazivi polja koji se u njima pojavljuju. Za dati primer, dobija se lista tipova polja T1-T5.
- Za svako polje čiji se naziv pominje u nekom od kvalifikovanih termova pretražuje se Rečnik podataka i pronalaze one baze dokumenata čija je struktura opisana preko polja BPR datog imena. Za dati primer pretpostavićemo da se polja T1-T5 javljaju samo u bazama B1-B5, i tada kao rezultat pretraživanja po Rečniku za opis strukture "ravnih" baza dokumenata, dobijamo:

Polje BPR	Skup baza podataka
T1	B1, B4
T2	B1
T3	B2, B3
T4	B1, B2, B3, B5
T5	B1, B2, B4

Primljeni korisnički upit se zatim deli na podizraze, po prioritetima izvršavanja. Ovaj postupak je analogan definisanju stabla izvršavanja upita, koje se sprovodi kod sintaksne analize upita SPI. Za dati primer, izdvajaju se sledeći nivoi:

Prioritet	Izraz
1	I1 $T1=a \text{ and } T2=b$
1	I2 $d \text{ and } T4=e(w)f$
1	I3 $h/T5 \text{ or } i$
2	I4 $I1 \text{ or } c/T3$
3	I5 $I4 \text{ and } I2$
4	I6 $I5 \text{ not } (g/T4)$
5	I7 $I6 \text{ and } I3$

Analiza kreće od prioriteta 1 za svaki od izraza zasebno. Preuzeti izraz se razbija na termove i izdvajaju se kvalifikovani i nekvalifikovani među njima. Iz svakog kvalifikovanog terma izdvaja se naziv polja i pridružuje mu se utvrđeni skup baza podataka koji ga poseduje u svojoj strukturi. Svaki nekvalifikovani term se zamjenjuje sa "all", označavajući da se term može primeniti na svim bazama podataka u sistemu. Zatim se nad skupovima baza pridruženim termovima primenjuju logički operatori. Primena logičkog operatora *and* sa "all" kao rezultat daje skup baza drugog operanda. Sto se tiče primene logičkog operatora *or* na "all" postoje dve varijante primene. Po jednoj, koja odgovara matematičkoj definiciji unije, rezultat je "all", dakle skup svih postojećih baza podataka; to je u suštini i tačno jer se pretraživanje po zadatom sadržaju nekvalifikovanog terma zaista može vršiti po svim bazama podataka u sistemu, po njihovim baznim indeksima. Druga varijanta je da se primeni isti tretman kao i u slučaju logičkog operatora *and*, tj. da se tražena logička operacija u ovoj analizi uopšte ne primeni. To je strožija varijanta koja se ipak primenjuje kao default opcija u procesiranju upita. S obzirom da se primenom ovih dveju varijanti u opštem slučaju dobijaju različiti skupovi ciljnih baza podataka, moguće je izvrstiti oba načina analize, i korisniku prikazati oba rezultata, sa izborom kome će se prikloniti. Za dati primer, za izraze I1, I2 i I3 dobijamo sledeći izlaz:

Izraz	Baze podataka
I1: $T1=a \text{ and } T2=b$	$(B1, B4) \text{ and } (B1) = B1$
I2: $d \text{ and } T4=e(w)f$	$\text{all and } (B1, B2, B3, B5) = (B1, B2, B3, B5)$
I3: $h/T5 \text{ or } i$	$(B1, B2, B4) \text{ or all} = (B1, B2, B4) \quad \text{I}$ $= \text{all} \quad \text{II}$

- Sledi procesiranje po višim nivoima prioriteta. Treba primetiti da se primena operatora *not*, u skladu sa načinom na koji on funkcioniše u SPI, ignoriše tj. prevodi u izraz oblika *term1 and all*. Osim toga, vidi se da primena prve ili druge varijante tretmana operatora *or* nad all, daje i razlike rezultate. U konkretnom primeru, primena blaže varijante ima smisla, jer se kao rezultat dobija konačan skup ciljnih baza podataka. Međutim, kada se kao rezultat ove varijante dobije rezultat "all" (sto bi bio slučaj da je I7 = I6 or I3), onda će ceo proces analize upita čini besmislenim, pa se primena *or* nad "all" ne izvrsava.

Prioritet 2:

Izraz	Baze podataka
I4: I1 or c/T3	B1 or (B2, B3) = (B1, B2, B3)

Prioritet 3:

Izraz	Baze podataka
I5: I4 and I2	(B1, B2, B3) and (B1, B2, B3, B5) = (B1, B2, B3)

Prioritet 4:

Izraz	Baze podataka
I6: I5 not (g/T4)	(B1, B2, B3) and all = (B1, B2, B3)

Prioritet 5:

Izraz	Baze podataka	
I7: I6 and I3	(B1, B2, B3) and (B1, B2, B4) = (B1, B2)	I
	(B1, B2, B3) and all = (B1, B2, B3)	II

- Kada se ustanovi konačni skup baza podataka koje po strukturi mogu odgovoriti na postavljeni upit, rezultat procesiranja upita se ispostavlja Serveru sesije, gde postoji nekoliko mogućnosti dalje komunikacije sa korisnikom. Najpre, može mu se dobiti skup baza podataka ponuditi na izbor. Kao dodatna pomoć u odluci nad kojom bazom će se nastaviti dalje pretraživanje, korisniku se uz svaku od baza može prikazati i njen Bluesheet dokument, a poslednja opcija je automatski nastavak pretraživanja, rutiranjem upita ka celom skupu ciljnih baza podataka.
- Rutirani upit se, na svakoj od baza kojima je upućen, pretražuje korišćenjem lokalnog SPI, koji obavlja pretraživanje nad bazom indeksa i sav ostali lokalni posao preslikavanja iz baze dokumenata i formatiranja izlaznog SPI dokumenta.
- Korisniku se kao rezultat vraća objedinjena lista dokumenata koji zadovoljavaju postavljeni upit i njihov sadržaj mu se odmah stavlja na raspolaganje. Uz svaki pristigli dokument sledi i informacija iz koje baze je potekao, tako da korisnik, po želji, dalje pretraživanje može da nastavi i lokalno.

Nadgradnja primjenjenog algoritma za procesiranje upita i dalje sužavanje skupa ciljnih baza dokumenata se može postići realizacijom centralne baze indeksa pri Rečniku. Preuzimanjem ili kopiranjem lokalnih baza indeksa u Rečnik podataka, omogućava se i dodatna provera uspešnosti realizacije upita i po sadržaju termova (bili oni kvalifikovani ili ne), a ne samo po njihovoј kvalifikaciji. Time se podržava i provera blizinskih faktora, kao i operatora *not*. Na ovaj način se dobija skup baza podataka koje nisu samo potencijalni, već i potpuno provereni izvori informacija po traženom upitu. Međutim, preuzimanje dokumenata se mora obaviti lokalno, pa se ciljnim bazama podataka upućuju samo zahtevi za preuzimanje navedene liste dokumenata. Preostaje, ipak, pitanje svrshishodnosti ovakvog postupka i detaljna analiza koja bi razmotrla prednosti i nedostatke, pre svega po pitanju predviđenog ukupnog vremena odziva.

9.5. IMPLEMENTACIJA REČNIKA PODATAKA

Rečnik podataka, kao osnova za ostvarivanje integrisanosti i kooperativnosti u otvorenom i distribuiranom sistemu SNTIS, se, prirodno, i sam razvija kao distribuirani informacioni sistem, sa tri nivoa distribucije: Centralnim Rečnikom, Kooperativnim Rečnicima pojedinih podistema i lokalnim Rečnicima podataka.

Centralni Rečnik podataka sadrži informacije o svim informacionim resursima u SNTIS, kao i definiciju usvojenih standarda koji obezbeđuju usaglašen rad heterogenih komponenti sistema. Zadužen je za ostvarenje transparentnog globalnog pretraživanja u sistemu, analizom upita i njihovim rutiranjem ka lokalnim bazama podataka, kao što je to detaljno opisano u prethodnom poglavlju.

Kooperativni Rečnici podataka preslikavaju strukturu sadržaja Centralnog Rečnika, ali nad suženim skupom podataka, vezanim za određeni fizički ili logički deo mreže SNTIS.

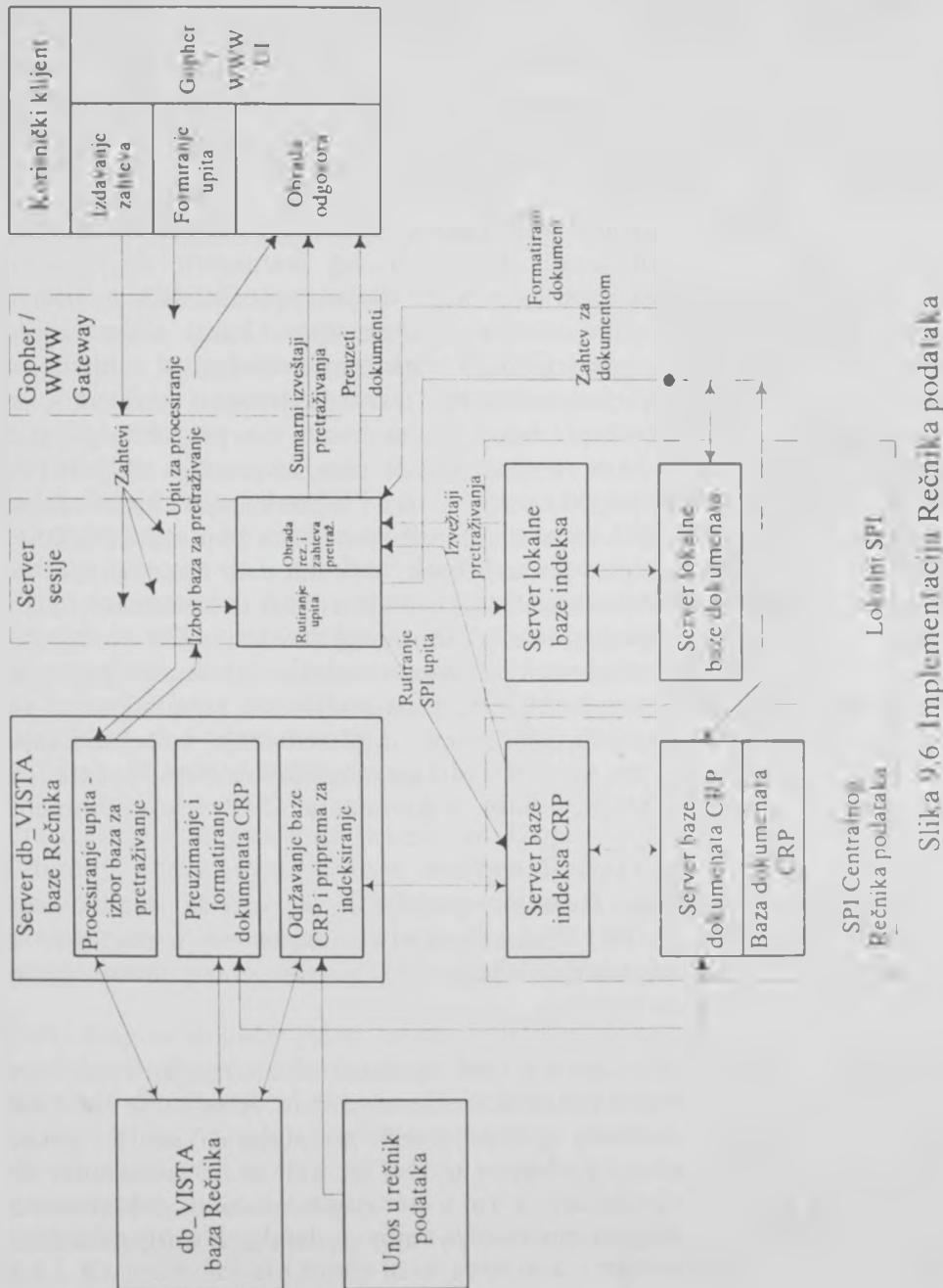
Lokalni rečnici podataka su interni rečnici konkretnih korisničkih baza podataka u sistemu, uz dodatne informacije o načinu preslikavanja iz forme SPI dokumenata koji su opisani u Rečniku u konkretnu implementaciju baze podataka, čime se obezbeđuje izvršavanje Rečnički vođenih aplikacija i pretraživanje heterogenih baza podataka.

Implementacija lokalnih i kooperativnih Rečnika podataka je potpuno nezavisna od ostatka sistema, prema kome se jedino u interfejsu moraju ispoštovati utvrđeni standardi. Način aplikativnog ostvarenja tog interfejsa, izmene u lokalnoj strukturi ili nacinu pristupa su sasvim autonomni i neprepoznatljivi su od ostatka sistema. Otuda se u ovom poglavlju detaljno opisuje samo implementacija Centralnog Rečnika podataka (CRP) i način ostvarivanja distribuiranog pretraživanja korišćenjem Internet servisa i usluga lokalnih SPI. Usluge Kooperativnih Rečnika se ne koriste kao međukorak u ovom pretraživanju, već se ono obavlja direktno iz Centralnog Rečnika podataka. Kooperativni Rečnici svoju ulogu ostvaruju tek lokalno, preuzimajući unutar konkretnе fizičke ili logičke mreže funkcije Centralnog Rečnika podataka, prateći isti koncept implementacije, prikazan na Slici 9.6.

Centralni Rečnik je strukturana baza, realizovana primenom sistema za upravljanje bazama podataka db_VISTA, koji podržava mrežni model podataka. db_VISTA baza Centralnog Rečnika sadrži dve osnovne celine. Prva objedinjuje podatke o informacionim resursima iz cele mreže SNTIS čime se ostvaruje informativni karakter Rečnika. Tu su podaci o institucijama, računarskim mrežama i mrežnim čvorovima, bazama podataka i različitim informacionim servisima i uslugama. Drugi deo db_VISTA baze Rečnika je posvećen ostvarivanju transparentnog pretraživanja; tu se nalaze definicije različitih standarda i opisi komponenti sistema (baza podataka tipova dokumenata) primenom tih standarda. Server strukturane baze Rečnika podataka je jedini način pristupa njenom sadržaju i obuhvata sve aktivnosti od formiranja i održavanja baze Rečnika, preko prevođenja sadržaja Rečnika u ravnu formu SPI dokumenata, do izvršavanja upita nad db_VISTA bazom i procesiranja korisničkih upita i izbora baza podataka ka kojima će se upit rutirati.

Uz db_VISTA bazu CRP nalazi se i baza sekundarnih SPI dokumenata koji preslikavaju deo Rečnika sa opisom informacionih resursa u SNTIS. U poglavlju 5, koje opisuje arhitekturu SPI, opisan je postupak kojim se baza podataka bilo koje strukture prevodi u "ravnu" strukturu, koja se potom indeksira i nad kojom je moguće SPI pretraživanje. Sam proces indeksiranja se, u skladu sa definisanim pravilima preslikavanja iz strukturane baze u bazu podataka za pretraživanje, može obaviti na dva načina:

- Direktnom međuprocesnom komunikacijom, kojom se isčitava polje po polje strukturane baze i po definisanim pravilima preslikava u polje baze podataka za pretraživanje i uključuje u bazu indeksa.
- Formiranjem privremene ili stalne "ravne" baze dokumenata, nad kojom se tek u narednom koraku izvršava indeksiranje.



Slika 9.6 Implementacija Rečnika podataka

SPI Centralno
Rečnika podataka

db_VISTA baza CRP se, kao jedan tip struktuirane baze podataka, indeksira kombinovanjem oba ova načina. Samo indeksiranje započinje direktno iz db_VISTA baze, preko serverskog modula "Priprema za indeksiranje". Kao što se unutar svakog Lokalnog Rečnika podataka čuvaju pravila preslikavanja iz struktuirane baze u formatirani SPI dokument, tako se i u db_VISTA bazi CRP za svaki rekord i polje u njemu zna u koja polja "ravne strukture" će biti uključen, u koji konkretni dokument, kojim redosledom i na koji način. Priprema za indeksiranje, na osnovu ovih pravila u bazi, formira polje po polje budućeg SPI dokumenta i šalje ga na indeksiranje koje se obavlja unutar Servera baze indeksa.

Sadržaju baze CRP odgovara više tipova SPI dokumenata. Međutim, posebno značajni su dokumenti informativnog karaktera sa opisom pojedinih komponenti sistema, jer im se najčešće pristupa. Otuda se za njih u procesu indeksiranja formira stalna, redundantna baza sekundarnih dokumenata - Baza dokumenata CRP. S obzirom na učestanost zahteva za ovim dokumentima i vrlo retke izmene, redundantnost u podacima je opravdana, jer se preuzimanje dokumenata svodi na prezentaciju već gotovih izvestaja, umesto da se za svaki zahtev nanovo pristupa db_VISTA bazi, konsultuju pravila preslikavanja i vrši formatiranje. Sama Baza dokumenata CRP je organizovana u vidu direktorijuma koji sadrže fajlove sa dokumentima istog tipa. Server baze dokumenata na zahtev za preuzimanjem nekog od dokumenata sačuvanih u Bazi dokumenata CRP, reaguje prostim isporučivanjem već formiranog izvestaja. Za slučaj da Serveru baze dokumenata stigne zahtev za preuzimanje dokumenta koji nije ponovljen u Bazi dokumenata CRP, uspostavlja se komunikacija sa Serverom struktuirane baze Rečnika, vrši se preuzimanje i formatiranje traženog SPI dokumenta na osnovu sadržaja db_VISTA baze Rečnika, i takav izveštaj se preko Servera baze dokumenata CRP vraća krajnjem korisniku.

Baza indeksa Centralnog Rečnika podataka je takođe realizovana upotrebom db_VISTA SUBP. Njena struktura je prikazana u poglavljju 5 koje opisuje arhitekturu i organizaciju SPI i ista je za svaku bazu dokumenata nad kojom se formira.

Osnovna ideja distribucije rečnika počiva na distribuiranom, lokalizovanom unosu podataka o sopstvenim informacionim resursima. Ažuriranje sadržaja Rečnika se najčešće svodi na prijem novih podataka o institucijama, mrežnim čvorovima, bazama podataka u SNTIS, koji stizu iz lokalnih ili kooperativnih Rečnika podataka, i to u vidu sekundarnih dokumenata definisanog sadržaja i formata. Softver za unos podataka o informacionim

resursima u SNTIS kao izlaz formira takvu strukturu dokumenata, posćbno za svaku Rečničku kolekciju podataka.

Pristigle dokumente obrađuje modul "Održavanje baze CRP i priprema za indeksiranje", koji pristupa pravilima preslikavanja definisanim u db_VISTA bazi Rečnika i njihovom primenom svako polje iz dokumenta na odgovarajući način preslikava u mrežnu strukturu baze Rečnika. U isto vreme se vrši provera postojećih i formiranje novih veza koje se u db_VISTA bazi uspostavljaju između različitih kolekcija i dokumenata (između baze i čvora, srodnih dokumenata, polja itd.). Nakon što se novi dokument preslika u bazu Rečnika, vrši se njegovo indeksiranje i formatirani izveštaj se smešta u Bazu dokumenata CRP.

Proces održavanja baze CRP, koji struktuiru ravan dokument, i proces pripreme za indeksiranje, koji je zadužen za "ravnjanje" strukture, su očigledno komplementarni, ali funkcionišu primenom istih pravila preslikavanja i aplikativnih rešenja. Oba procesa funkcionišu u stand-by režimu i pokreću se u određenim vremenskim intervalima, gde se najpre za sve pristigle dokumente vrši preslikavanje u bazu, a tek potom priprema za indeksiranje i indeksiranje. Unutar servera db_VISTA baze Rečnika, modul "Preuzimanje i formatiranje dokumenata iz CRP" za traženi dokument iščitava odgovarajući deo baze Rečnika i formira izveštaj koji se smešta u Bazu dokumenata. Ovaj izveštaj je obogaćen dodatnim informacijama u odnosu na originalni sadržaj. Za dokument koji opisuje neku bazu podataka se, na primer, dodaju i reference na dokumente koji opisuju mrežne čvorove na kojima je baza instalirana, baze podataka koje se odnose na istu oblast ili podoblast, deo tezaurusa koji opisuje veze polja za pretraživanje ove baze podataka sa nekim drugim bazama itd.

Pretraživanje se realizuje primenom Gopher i WorldWideWeb (WWW) Internet servisa. Pretraživanje sadržaja Rečnika podataka je, upotrebom ovih servisa, ostvareno u više nivoa, od elementarne opcije statičkog pretraživanja kretanjem kroz Gopher ili WWW hijerarhiju do transparentnog pretraživanja lokalnih baza podataka [4,6].

Prvi i najprostiji način pretraživanja koristi samo mogućnosti koje ovi servisi pružaju sami po sebi - pristup Bazi sekundarnih dokumenata CRP se ugrađuje direktno kroz Gopher ili WWW hijerarhiju, u vidu predefinisanih upita. Uspostavljena je statična hijerarhija podataka, gde je grupisanje po direktorijumima izvršeno najpre po pripadnosti nekom od Rečničkih tipova dokumenata, a zatim i po specifičnim sadržinskim kriterijuma (pregled baza podataka po oblastima, mrežnih čvorova po logičkim i fizičkim mrežama itd.). Korisnik se sam kreće kroz strukturu i na najnižem nivou hijerarhije

izborom neke od stavki u meniju pokreće izvršenje predefinisanog upita, koji kao rezultat vraća identifikaciju jednog ili više dokumenata iz Baze dokumenata CRP. Izborom nekog od njih, sadržaj se iz baze dokumenata CRP preuzima u zatečenom formatu. Statičnost pretraživanja ovde ne označava nepromenljivost podataka, jer se aplikacije direktno obraćaju bazi Rečnika i vraćaju trenutno ažuran sadržaj, već samo nemogućnost izbora parametara po kojima se pretraživanje vrši. Osim toga, organizacija "putovanja" kroz Internet svet korisniku omogućava da se u bilo kom momentu može preseliti na neki drugi Gopher ili WWW server u sistemu i tamo nastaviti dalji rad, nad lokalnim Rečnicima i bazama podataka.

Sledeći način pretraživanja nudi interaktivno zadavanje upita nad proizvoljno odabranom bazom dokumenata, pri čemu se za svaku pojedinačnu kolekciju diže po jedna instance Gopher ili WWW gateway-a za SPI. Korisnik na Gopher-u ili WWW-u upit zadaje u SPI sintaksi, upit se rutira ka lokalnoj bazi indeksa, gde se obavlja pretraživanje i isporučuje lista sekundarnih dokumenata koji zadovoljavaju upit. Izborom nekog od dokumenata iz liste korisnik izdaje zahtev za preuzimanje dokumenta, zahtev se prosleđuje lokalnom Serveru baze dokumenata gde se obavljaju sva potrebna preslikavanja iz stvarne strukture lokalne baze u SPI dokument. Format tih dokumenata je u skladu sa predefinisanim formatom za prikazivanje, pri čemu se korisniku može pružiti i mogućnost izbora jednog od više ponuđenih formata, a samo formatiranje se takođe obavlja lokalno na serveru baze dokumenata. Navođenje do baze koja će se pretraživati se opet ostvaruje kroz hijerarhijsku strukturu na Gopher-u ili WWW-u, ili prečicom - izborom baze iz uslužne forme, s tim što se na najnižem nivou ne nalaze više statične aplikacije za pretraživanje, već se otvara dijalog sa korisnikom koji, u skladu sa strukturom BPR izabrane baze, zadaje svoj upit.

Mogućnost pretraživanja nad više eksplicitno navedenih baza podataka ili potpuno transparentnog pretraživanja se ostvaruje putem WWW formi ili Gopher ASK formi, koje korisnik interaktivno popunjava, da bi se na osnovu skupa zadatih parametara izvršilo pretraživanje. Popunjavanje formi uključuje mogućnost jednoznačnog ili višestrukog izbora, unos parametara u vidu slobodnog teksta ili pritiskom na taster (WWW). Jedan od parametara je i izbor baze podataka kojoj se želi pristupiti. Pretraživanje se odvija u više koraka:

- Preuzimaju se parametri zadati u formi i na osnovu njih se formira upit u SPI sintaksi.
- Upit se prosleđuje ka Serveru struktuirane baze Rečnika, u modul "Procesiranje upita - izbor baze" gde se analizira u skladu sa

sadržajem Rečnika podataka koji čuva informacije o strukturi svih baza podataka u SNTIS i određuje se ciljni skup baza podataka, kao što je to opisano u prethodnom poglavlju. Ovaj korak se preskače u slučaju da je korisnik eksplicitno, kao jedan od parametara u formi, naveo svoj ciljni skup baza podataka.

- Spisak baza za pretraživanje se može ponuditi korisniku na uvid i na konačni izbor, inače se vrši automatsko rutiranje ka svim bazama iz utvrđenog skupa.
- Upit se rutira ka skupu baza podataka određenih za pretraživanje, gde se koriste usluge lokalnog SPI, odnosno lokalne baze indeksa. Upit može biti upućen i samom Rečniku podataka, odnosno njegovoj bazi indeksa.
- Rezultati pretraživanja nad svakom pojedinačnom lokalnom bazom indeksa se upućuju ka Serveru sesije, gde modul "Procesiranje rezultata pretraživanja" sačekuje odgovore sa svih lokalnih baza kojima je upit upućen. Dobija se finalna lista dokumenata, iz različitih kolekcija dokumenata, različite strukture, sa informacijom o poreklu i lokalnim mogućnostima pretraživanja.
- Konačni rezultat pretraživanja se prikazuje korisniku i na zahtev se vrši preuzimanje dokumenata. Po sadržaj dokumenta se "odlazi" u lokalnu bazu sekundarnih dokumenata, uspostavljanjem veze sa lokalnom interfejsnom bibliotekom, što uključuje primenu svih potrebnih pravila preslikavanja iz struktuirane baze u SPI dokument, kao i standard za izveštavanje.
- Originalni sadržaj primljenih dokumenata se, u konsultaciji sa db_VISTA bazom Centralnog Rečnika podataka, može obogatiti nizom dopunskih informacija, što je posebno značajno kod automatskog rutiranja upita. U dopunske informacije spada i identifikovanje matične baze podataka iz koje dokument dolazi, skup srodnih dokumenata, obaveštenja o lokaciji na kojoj se može dobiti detaljno korisničko uputstvo, pregled relevantnog dela tezaurusa rečnika o sinonimnim ili poljima nad istim korisničkim domenom.
- Tako dobijen tekst dokumenta je onaj koji se prezentira krajnjem korisniku, a kroz WWW je omogućeno i direktno prebacivanje na pregled srodnih sadržaja.

Gopher i naročito WWW se, osim za pretraživanje, mogu iskoristiti i za inicijalni unos u Rečnik podataka. Unos se realizuje na sličan način kao i parametrizovano pretraživanje, korišćenjem formi, i predstavlja jedan od mogućih zahteva na korisničkom klijentu. Od korisnika se zahteva unos podataka, a obezbeđena je i podrška u proveri obaveznih polja, grupnih

polja, polja koja se mogu ponavljati. Preuzimanje podataka i uvrštavanje u bazu sekundarnih dokumenata Rečnika se vrši naknadno, a korisnik koji je obavio unos se putem elektronske pošte obaveštava o obavljenom poslu, kad mu se prilaže i podaci o identifikatorima kreiranih sekundarnih dokumenata i dodatnim aspektima uvrštavanja u Centralni ili Kooperativni Rečnik podataka.

Celokupan rad korisnika sa Rečnikom podataka se beleži u serveru sesije, koji nije deo samog Rečnika, ali treba da prati njegov rad. S obzirom da se kao korisnički klijenti koriste Gopher i WWW, a da se samo pretraživanje vrši preko SPI, Gopher, odnosno WWW gateway se smešta između klijenta i ostalog dela sistema, sa zadatkom da vodi računa o zahtevanim protokolima, Gopher ili HTTP protokola sa korisničke strane i DS/RS protokola od strane sistema za pretraživanje.

Dalji razvoj Rečnika podataka je vezan pre svega za širenje dijapazona pristupa kroz raspoložive Internet informacione servise. Time se povećava i aktuelnost razmene podataka u SNTIS u ovom okruženju, pa i potreba nadgradnje postojećih i razvoja novih mogućnosti pristupa Rečniku podataka SNTIS i u njemu opisanim korisničkim bazama.

LITERATURA

- [1] American National Standard for Information Systems, Information Resource Dictionary System, ANSI X3.138 - 1988.
- [2] IRDS Service Interface, ISO/IEC JCT1/SC21/WG3 N390, N452, N600, N669; IRDS Framework, ISO DP 10027.
- [3] Lazarević, B., Mišić, V., *Extending Entity Relationship Model to Capture Dynamic Behavior*, Eur.J.Inf.System, Vol.1, No.2, pp. 95-106, iz 1991 godine.
- [4] Lazarević, B., Ivković, M., Dikanović, V., Nešković S., *Distributed Data Dictionary in the Internet Environment*, International Association for Computer Information Systems, Toronto 1995.
- [5] Lazarević, B., Ivković, M., Dikanović, V., Nešković S., *Data Dictionary of the Scientific and Technological Information System of Serbia*, Hellenic Operational Research Society, 1995.
- [6] Ivković, M., Lazarević, B., Dikanović, V., *Upravljačka mreža i Rečnik podataka SNTIS*, JISA INFO, No.2, pp. 12-16, 1995. godine.

INDEKS

—A—

agent, 121
agregacija, 166
algoritam interpretacije, 134
ANSI, 7, 46, 61, 222, 252
aplikacioni interfejs, 96, 101, 121, 213
aplikacioni objekat, 73, 74, 82, 83, 87, 88,
90, 91, 95, 96, 97, 102, 198, 200, 202, 216
aplikacioni programski interfejs (API),
124, 125, 139, 148
apstrakcija agregacije, 67
apstrakcija generalizacije, 67, 77
archie, 43
arhitektura otvorenih sistema, 2
arhitektura otvorenog sistema, 59
ARTIST, 61, 101, 102, 222, 223, 225, 228
ATLASS, 2, 7, 10
ATM, 17, 24

—B—

baza dokumenata, 4, 5, 110, 113, 116, 117,
139, 159, 160, 168, 170, 171, 174, 178,
179, 228, 230, 235, 236, 241, 242, 244,
245, 248, 250
baza indeksa, 4, 5, 114, 117, 125, 128, 129,
135, 136, 137, 139, 140, 141, 142, 143,
144, 145, 146, 150, 153, 157, 216, 228,
245, 248, 251
baze bibliografske građe, 193, 194, 213,
216, 218
bazni objekat, 7, 72, 74, 75, 84, 87, 88, 95,
97, 98, 102, 188, 198
BEONET, 18, 24, 25, 26, 27, 30, 31, 45,
bibliografska građa, 6, 7, 185, 186, 188,
193, 194, 198, 200, 202, 212, 213, 216,
217
bibliotečka mreža (BM), 46, 47, 48, 49, 58,
119, 185, 217, 219
Bibliotečki sistem, 6, 7, 47, 49
Bibliotečko izveštavanje i
dokumentovanje, 6
binarna veza, 76
blizinski operator, 128, 129, 130, 131, 132

—C—

CASE, 3, 61, 66, 74, 101, 222, 223, 225,
228
CCITT, 18
centralni host, 107
centralni rečnik podataka, 241, 245
COM, 63
CORABA, 63

—D—

db_VISTA, 5, 7, 8, 142, 217, 218, 229, 235,
246, 248, 249, 251
DECnet, 15
definicija tipa dokumenata (DTD), 161,
162, 163, 164, 166, 168, 170, 171, 174,
176, 178, 179, 181, 182
deklaracija elemenata, 162
DIALOG, 8, 110, 119, 128, 172, 232, 235
digitalna magistralna veza, 19
dijagram toka podataka (DTP), 4, 68
dijagrama objekti veze (DOV), 75, 76, 77,
166, 172, 174
dinamička pravila integriteta, 85
direktno pretraživanje, 5, 174, 176, 177,
178, 230
direktorijum, 46, 221
distribucija aplikacija, 100
distribucija rečnika, 248
Distributed Group Support System, 55.
dodatni indeks, 112, 172, 237, 239
dokument, 62, 123, 127, 140, 144, 147, 162,
163, 168, 169, 170, 171, 176, 178, 185,
190, 191, 197, 228, 232, 234, 235, 236,
238, 244, 248, 249, 250, 251
drugi indikator, 190, 193

—E—

EARN, 10, 19
EDIFACT, 62
eksterni šifarnik, 190
elektronska konferencija, 43
elektronska pošta, 43

ETHERNET, 17, 24
expand, 127, 129, 143, 153, 154, 156

—F—

FDDI, 17, 24, 25, 30, 31, 57, 234
finger, 43
fizička mreža, 234
fizičko projektovanje, 100, 101
fraza, 111, 128, 143, 144, 145, 146, 154,
156, 237, 238, 239, 240

—G—

gejtvej, 4, 120, 122, 123, 153, 155
Gopher, 4, 5, 35, 43, 117, 119, 120, 121,
122, 123, 152, 153, 155, 249, 250, 251,
252
Gopher gejtvej, 120, 153, 155
grafički tag, 169

—H—

HTML, 155, 156, 177
HTTP, 119, 120, 122, 152, 155, 157, 252

—I—

implementacioni model, 73, 95, 96, 101
indeksiranje, 4, 5, 6, 8, 107, 110, 113, 114,
115, 117, 119, 120, 121, 122, 123, 124,
125, 126, 135, 139, 140, 142, 143, 144,
148, 152, 153, 155, 156, 157, 171, 172,
173, 178, 179, 182, 183, 185, 213, 216,
218, 226, 228, 248, 249
indeksiranje dokumenta, 140, 144
informacije o adresi u elektronskoj pošti,
43
instrukcija za obradu, 162, 163
interaktivna poruka, 43
interaktivno priključenje na udaljeni
računar, 43
interfejs, 5, 61, 62, 69, 96, 100, 101, 117,
121, 124, 125, 126, 136, 139, 143, 144,
148, 149, 153, 154, 178, 179, 213, 216,
217, 219, 230, 236, 246
Internet, 4, 5
interni šifarnik, 190
interpretacija plana izvršavanja, 129, 134
irc, 43
ISO, 5, 7, 46, 161, 222

izveštavanje i publikovanje, 4

—J—

jedinstveni sistem za indeksiranje i
pretraživanje dokumenata, 6, 119, 148,
156
jezgro aplikacije, 96, 178, 213
jezika za pretraživanje, 110, 143, 171
JUNIS, 33, 34, 35
JUPAK, 17, 35, 44

—K—

katalog, 46, 48, 49, 221
klasa objekata, 75, 76, 77, 80, 83, 85
klasa veza, 76
klijent, 44, 119, 121, 122, 123, 124, 145,
146, 147, 148, 150, 151, 152, 153, 216,
234
klijentska aplikacija, 121, 122, 123, 153
komandni jezik, 127
kooperativni podsistem, 105, 107, 240
kooperativni rečnik podataka, 245
korisnička sesija, 115, 116, 117
korisnički interfejs, 96, 236
korisničko pretraživanje, 6, 186, 188, 218

—L—

leksička analiza, 129
logička mreža SNTIS, 46
logički operator, 128, 131, 132, 242, 243
logika primitivnih procesa, 71
lokalna mreža, 13, 24, 26, 27, 30, 31, 33,
35, 37, 39, 41
lokalni rečnik podataka, 245
lokalni SPI, 7, 105, 240, 244, 246, 251
LOMED, 24, 25, 30
LOSTUD, 24, 27, 45, 239
LOV, 30, 32

—M—

magistralna mreža, 25
međumesni segment, 18, 20
metodologija, 2, 3, 6, 59, 60, 61, 65, 66, 74,
95, 101, 102, 104, 157, 198, 200, 213, 222
mreža poslovno - tehnoloskih informacija
(MPTI), 53, 54, 55
mreža specijalizovanih sistema, 49, 58

mrežni čvor, 229, 232, 234, 235, 236, 238,
249
mrežni model podataka, 142

—N—

netfind, 43
news, 35, 43
nivo kooperacije, 105

—O—

objekat, 3, 4, 7, 60, 67, 68, 69, 72, 73, 74,
75, 76, 77, 78, 79, 81, 82, 83, 86, 87, 88,
89, 90, 91, 95, 97, 98, 101, 102, 110, 115,
140, 142, 153, 154, 188, 191, 193, 195,
197, 200, 217, 222, 223, 225, 228, 229,
230, 235
objektna dekompozicija, 67
objektna specifikacija softvera, 68
objektno orijentisan pristup, 3
objektno orijentisana specifikacija, 7, 72,
102
obrada dokumenata, 181, 187, 188, 218
obrada rezultata pretraživanja, 116
ODA (Office Document Architecture),
62, 160

održavanje bibliografske građe, 186
ograničenja, 66, 72, 74, 83, 84, 85, 86, 95,
110
OPAC, 48
operacija, 60, 67, 68, 72, 74, 77, 79, 81, 82,
86, 87, 88, 89, 90, 91, 93, 94, 95, 96, 97,
98, 100, 102, 128, 131, 132, 133, 139,
145, 146, 147, 150, 151, 152, 200, 218,
243
operacija održavanja, 78, 85
operacija pretraživanja, 80
optički kabal, 19
organizovana distribucija, 43
OSI, 15, 144
OSI model, 144
otvoren heterogeni sistem, 59
otvoren sistem, 2, 59, 65, 95, 188, 218

—P—

peer to peer, 122
phone, 43
polje, 3, 4, 88, 89, 90, 110, 111, 113, 114,
115, 116, 119, 122, 128, 129, 130, 132,

136, 137, 138, 140, 143, 149, 150, 151,
152, 155, 156, 171, 172, 190, 191, 193,
195, 197, 216, 226, 228, 235, 236, 237,
239, 241, 242, 243, 248, 249, 252
potpolja, 190, 191, 193, 195, 216
potpolje, 190, 191, 193, 195, 216
pozicioni tag, 169
pravila integriteta, 60, 83, 85, 86, 87, 90,
102
prenos datoteka, 43
pretraživanje, 4, 112, 115, 116, 200, 202,
213, 249, 250
pretraživanje baza podataka, 47, 54, 105,
159
pristup bazi indeksa, 126, 139, 141
Prošireni model objekti-veze (PMOV),
74, 75, 76, 77, 78, 80, 82, 83, 85, 98, 101,
139, 188, 222, 228, 230, 232
proces obrade podataka, 69
programski jezik C, 7
prvi indikator, 190, 193
PTT, 18
publikovanje, 4, 107, 109, 159, 169, 177,
229
putanja, 172, 173, 176, 182, 183

—R—

racunarska mreža, 1, 39, 41, 57, 119, 124
radno okruženje bibliotekara, 191, 218
ravna baza dokumenata, 113
RCUB, 24, 26, 27, 35, 37, 39, 45, 237
rečnik podataka, 2, 3, 7, 8, 46, 47, 61, 70,
102, 105, 107, 108, 109, 110, 111, 117,
119, 178, 221, 222, 223, 225, 226, 228,
229, 230, 232, 234, 235, 236, 237, 240,
241, 242, 245, 246, 248, 249, 251, 252
Rečnik podataka SNTIS, 7, 8, 47, 61, 221,
223, 229, 252
rečnikom vođene aplikacije, 109, 240
referenca entiteta, 162
reindeksiranje, 148
rutiranje upita, 116

—S—

SCO UNIX, 7, 217, 219
sekundarna dokumenta, 220, 222, 223,
225, 228, 229, 235, 248, 249, 250, 251,
252
sekundarno polje, 190, 195

- select, 127, 128, 129, 130, 131, 143, 153, 154, 156, 237, 238, 239, 240, 242
selectc, 127, 128, 129, 143
selects, 127, 128, 129, 143, 153, 156
semantička analiza, 129, 130
server, 3, 4, 5, 7, 31, 33, 35, 37, 39, 41, 44, 62, 96, 97, 100, 101, 102, 117, 119, 120, 121, 122, 123, 124, 125, 126, 127, 135, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 178, 181, 213, 216, 234, 240, 249, 250
server baze podataka, 96, 97, 121
server centralnog rečnika podataka, 117
server za indeksiranje i pretraživanje dokumenata, 4, 120, 121, 125
server za pristup bazi izvornih dokumenata, 4, 120, 122
servera baza dokumenata, 117
servera baze indeksa, 117, 248
SGML (Standard Generalized Markup Language), 4, 5, 62, 110, 113, 160, 161, 163, 164, 168, 169, 170, 171, 172, 173, 174, 176, 177, 180, 181, 182
sintaksna analiza, 129
Sistem naučnih i tehnoloških informacija, 1, 9, 43, 56, 57
sistem za pretraživanje informacija (SPI), 2, 3, 4, 5, 6, 7, 8, 60, 61, 62, 65, 105, 107, 108, 109, 112, 115, 117, 159, 166, 222, 228, 229, 235, 238, 240, 241, 242, 243, 244, 246, 248, 250, 251, 252
skladišta podataka, 68, 69, 70, 72
skup, 166
skupovni operator, 128, 146, 147
SNTIJ, 1, 6
SNTIS, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 31, 33, 35, 37, 39, 41, 43, 44, 46, 47, 51, 52, 55, 59, 61, 62, 105, 107, 110, 113, 119, 121, 155, 156, 157, 159, 170, 183, 185, 213, 219, 221, 222, 223, 228, 229, 230, 232, 235, 236, 237, 240, 241, 245, 246, 249, 251, 252
softverska zaostavština, 107
specifikacija aplikacionih objekata, 87, 198
specifikacija baznog objekta, 74, 188
specijalizacija, 168
SPI SNTIS, 2, 4, 5, 7, 107
standard razmene podataka, 2, 62
standardi za opis strukture baze, 109, 229
standardi za opis strukture baze za formiranje, 109, 229
standardi za razmenu podataka, 109, 229
standardi za strukturiranje baze podataka za pretraživanje, 109, 229
standardna metodologija, 2, 3, 6, 59, 60, 61, 65, 66, 74, 102, 222
standardni upitni jezik, 174
stanje fonda, 197
strukturiranje zapisa, 186, 187, 218
struktura, 3, 4, 5, 7, 8, 18, 20, 27, 47, 48, 49, 60, 67, 69, 70, 72, 74, 80, 81, 82, 87, 88, 105, 108, 110, 119, 129, 131, 134, 149, 150, 161, 168, 171, 172, 174, 176, 177, 180, 181, 182, 183, 188, 190, 193, 195, 198, 200, 202, 221, 222, 223, 228, 229, 230, 232, 234, 235, 236, 240, 242, 245, 246, 248, 249, 250
Strukturalna sistemска analiza (SSA), 68, 69, 70, 72, 74, 87
- T—
- tag, 162, 168, 169, 181, 183
talk, 43
TCP/IP, 10, 15, 33, 41, 45, 234
telnet, 35, 43
TERENA, 10
tip elementa, 166, 172
tip preslikavanja, 76
tip upita, 127
tok podataka, 69
TOKEN RING, 17, 24
transformacija plana izvršavanja, 129
transparentno pretraživanje, 222
- U—
- unija, 88, 128, 168, 172, 176
UNIMARC, 6, 7, 62, 187, 188, 190, 191, 195, 216, 218
upit, 8, 82, 110, 112, 115, 116, 117, 121, 122, 125, 127, 129, 130, 131, 137, 144, 156, 171, 172, 176, 187, 193, 216, 228, 235, 241, 242, 243, 244, 245, 246, 250, 251
upravljačka mreža SNTIS, 46
- V—
- VAX/VMS, 7, 217, 219, 220
Veronica, 43
višestruke veze, 76

vrednosna ograničenja, 74, 83, 84

—W—

Wide Area Network (WAN), 13, 33, 35,
37, 39, 41, 234

World telnet, 43

WorldWideWeb (WWW), 4, 5, 35, 43, 44,
117, 120, 121, 122, 123, 152, 155, 156,
249, 250, 251, 252

—Z—

zapis, 3, 49, 101, 143, 185, 186, 187, 190,
193, 198, 200, 216, 218

CIP - Каталогизација у публикацији
Народна библиотека Србије, Београд

007:681.32(497.11)

FORMIRANJE i pretraživanje baza podataka u
Sistemu naučnih i tehnoloških informacija
Srbije / redaktor Branislav Lazarević ;
autori Bender Miroslav, Cvetanović Svetlana,
Dikanović Violeta, Đurić Dragan, Ivković
Miodrag, Konečni Aleg, Korunović Dušan,
Lazarević Branislav, Milivojević Ljubiša,
Nešković Siniša, Paunović Đorđe, Surla Dušan.
- Beograd : Ministarstvo za nauku i

tehnologiju Republike Srbije, 1996 (Beograd :
Politop-P). - VIII, 257 str. : graf. prikazi
; 24 cm

Tiraž 600. - Bibliografija uz svako
poglavlje. - Registar.
ISBN 86-7282-001-0
1. Бендер, Мирослав 2. Лазаревић,
Бранислав
659.23:681.324 519.683.5
007:02]:681.32 681.324(497.11)
а) Информациони системи - Србија б)
Научно-техничке информације - Организација
с) Базе података - Организација д)
библиотекарство - Информациони системи е)
Рачунарске мреже - Србија

ID=46987020

Систем научних и технолошких информација (СНТИС), у свом извornом значењу, представља инфраструктурни систем за научно истраживачки рад са циљем да обезбеди потребне информације за бављењем овом делатношћу. Увиђајући технолошке промене у свету које су се догађале почетком 90-тих године у области информатке и рачунарских мрежа, као и друштвене промене у нашој земљи, Министарство за науку и технологију Републике Србије наручило је израду Концепције развоја СНТИС за период 1992-1995.

Пројектни тим под руководством професора др Бранислава Лазаревића урадио је Концепцију развоја СНТИС, а такав концепт прихватило је Министарство науке као део свог програма рада у овој области. Придржавајући се доследно ове Концепције, СНТИС се за три године развоја, кроз рад на неколико пројеката, развио у један савремени систем. Данас СНТИС представља систем који поред информација научним радницима и привреди нуди и међусобну рачунарску комуникацију и коришћење рачунарских ресурса у властитој рачунарској мрежи Интернет типа, која је уједно и прва Интернет мрежа у Југославији. Садржаји СНТИС се граде у оквиру подсистема, такозваних логичких мрежа као што су библиотечка, специјализованих система и технолошко пословних информација.

Ова књига садржи детаљне описе најважнијих резултата рада пре свега на софтверским пројектима који су реализовани у оквиру рада на СНТИС. Резултати дају додатне могућности коришћења стандардних Интернет сервиса и решавају део проблема рада библиотека у Интернет окружењу.

Књига је намењена научној јавности, заједно са серијом публикација о резултатима основних истраживања у периоду 1991 - 1995. година, са циљем упознавања и ефикаснијег коришћења овог значајног инфраструктурног система Републике Србије и СР Југославије.

Др Раšко Узуновић

ISBN 86-7282-001-0