

SISTEM ZA PRETRAŽIVANJE RESURSA BIBLIOTEKE MAŠINSKOG FAKULTETA U NIŠU

SYSTEM FOR BROWSING RESURSES OF A LIBRARY OF FACULTY OF MECHANICAL ENGINEERING AT NIŠ

Milan Zdravković, Miroslav Trajanović
Mašinski fakultet Univerziteta u Nišu

Sadržaj - Nakon 6 meseci eksperimentalnog rada, grupa saradnika Mašinskog fakulteta u Nišu je projektovala novu verziju sistema za pretraživanje resursa fakultetske biblioteke, i to na osnovu uočenih poteškoća u radu i brojnih sugestija administratora i korisnika. Glavne karakteristike ove verzije su novi korisnički interfejs koji omogućava korisniku da brzo i jednostavno dođe do potrebne informacije, i brža komunikacija sa bazom podataka, korišćenjem parametrizovanih upita.

Abstract - After 6 months of experimental work, a group of associates at Faculty of Mechanical Engineering, designed a new version of system for browsing resurses of a faculty library, based on encountered difficulties and many suggestions from administrator and users. Main features of this version are new user interface which enable a user to find a information fast and easily, and faster communication with back-end database, by using parametrized queries.

1. UVOD

Informacioni sistem biblioteke Mašinskog fakulteta u Nišu je nastao na bazi potrebe interaktivnog pristupa resursima biblioteke od strane korisnika, u cilju bržeg nalaženja potrebne informacije. Razvoj se odvijao u 3 faze: projektovanje administratorske aplikacije, uz pomoć Microsoft Access alata; unos podataka o resursima biblioteke sa papirnate dokumentacije; projektovanje mrežne aplikacije i njen probni rad.

Nakon 6 meseci probnog rada, a na osnovu zapažanja administratora i korisnika mrežne aplikacije, projektovana je druga verzija. Osnovne karakteristike nove verzije su novi korisnički interfejs, koji omogućava formiranje složenijih SQL upita, čiji je cilj brže dobijanje potrebne informacije sa servera baze podataka bibliografskih jedinica, i korišćenje parametrizovanih upita koje smanjuje mrežni saobraćaj i ubrzava izvršenje upita na serveru.

Izabrana troslojna (three-tier) arhitektura, odnosno, njena realizacija korišćenjem programskog jezika JAVA, se u probnom radu pokazala kao brzo i sigurno rešenje, primenjiva u heterogenom TCP-IP okruženju uz distribuciju klijent aplikacije putem Interneta i online vezu sa bazom podataka.

2. ARHITEKTURA INFORMACIONOG SISTEMA BIBLIOTEKE MAŠINSKOG FAKULTETA U NIŠU

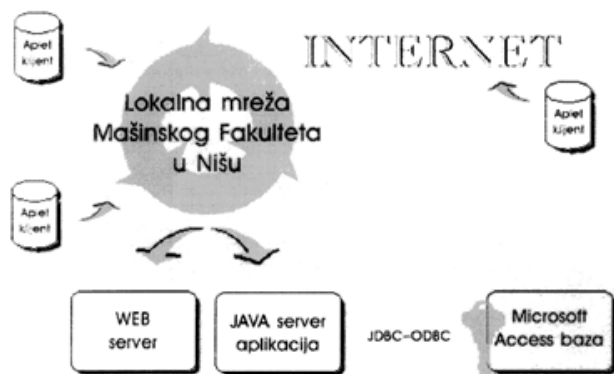
Sistem za pretraživanje resursa biblioteke Mašinskog fakulteta je nastao na temelju troslojne klijent-server arhitekture, čija je koncepcija i realizacija u programskom jeziku JAVA, detaljno prikazana u [1].

U daljem tekstu su navedene glavne karakteristike ove realizacije.

Osnovu informacionog sistema biblioteke predstavlja Microsoft Access baza podataka, koja se koristi pri unosu, pregledu i izdavanju inventarskih jedinica. Ova baza se nalazi na PC računaru u prostorijama biblioteke i sastoji se od administratorske aplikacije i samih podataka.

Zbog karakteristika brzine pristupa, jednostavnog administriranja i distribuiranja klijenata, za programsku realizaciju klijent - server arhitekture, izabran je troslojni (three-tier) pristup, implementiran uz pomoć JAVA razvojnog alata.

Na slici 2.1. prikazana je uprošćena šema programske realizacije informacionog sistema biblioteke.



2.1. Programska realizacija informacionog sistema biblioteke Mašinskog Fakulteta u Nišu

Klijent je non-trusted aplet, koji se download-uje sa WEB servera Mašinskog fakulteta. U sebi sadrži korisnički interfejs, klase za komunikaciju sa server aplikacijom putem otvorenog socket-a i klase za obradu i prikazivanje dobijenog ResultSet-a.

JAVA server aplikacija predstavlja rudimentarni database middleware sloj, koji ima sledeće funkcije:

- ✓ Ostvarivanje mrežne konekcije sa klijentima
Server aplikacija se izvršava na računaru na kome se nalaze HTTP server i RDBMS. Preko otvorenog socket-a na portu 9000, ona ostvaruje vezu sa apletima na klijent strani arhitekture.
- ✓ Dodeljivanje resursa klijent-server transakcijama korišćenjem JAVA multithreading klasa
Svskom pozivu sa klijent strane se dodeljuje po jedna nit procesa, pri čemu JAVA virtuelna mašina vrši automatsku raspodelu memorijskih resursa i CPU vremena.
- ✓ Provera validnosti SQL rečenice
Provera validnosti SQL rečenice predstavlja sigurnosnu meru koja obezbeđuje zaštitu strukture baze i integriteta podataka u njoj. Ova mera je primenjena u obliku obične sintaksne provere, koju server aplikacija vrši nad SQL rečenicom, pre njenog izvršavanja. Zabranjeno je izvršavanje svih iskaza koji ne počinju naredbom SELECT.

- ✓ Komunikacija sa bazom posredstvom JDBC:ODBC interfejsa

Kao server baze podataka, koristi se Microsoft Access RDBMS koja se nalazi na računaru u prostorijama biblioteke, gde se vrši i njeno administriranje.

3. KORISNIČKI INTERFEJS

Posebna pažnja u izradi nove verzije je posvećena projektovanju novog korisničkog interfejsa i to na osnovu iskustava u radu sa prethodnom verzijom, odnosno, zapažanja i primedbi korisnika. Na osnovu prikupljenih podataka, urađena je analiza čiji je rezultat bio niz zahteva koje novi interfejs treba da ispuni:

- ✓ Dinamičko formiranje SQL upita, koja korisniku treba da omogući komfornije definisanje uslova pretraživanja, kao i optimalan prenos podataka kroz mrežu.
- ✓ Dobijanje potpunijih informacija o traženom delu, koje ga potpuno definišu kao bibliografsku jedinicu, sa stanovišta organizacije biblioteke fakulteta.
- ✓ Pretraživanje baze na osnovu elemenata polja koja su dobijena kao rezultat pretraživanja, pri čemu ovi elementi asociraju korisnika na sledeći korak pretraživanja.

Elementi novog korisničkog interfejsa omogućavaju formiranje složenih SQL rečenica, tako da je korisniku omogućeno da sa što manje koraka dodje do traženog polja u bazi. Pri formiranju upita, od korisnika se traži da definiše uslove i izabere parametre pretraživanja.

Uslovi pretraživanja predstavljaju parametre SQL upita.

Izborom parametara pretraživanja, korisnik definiše domen dejstva formiranog SQL upita. Pritom se u domen pretraživanja mogu uključiti podaci iz tabela: Knjige, Diplomski radovi, Magistarski radovi, Doktorske disertacije i Časopisi.

USLOVI PRETRAŽIVANJA

SELECT *

FROM 'Knjige', 'Magistarski radovi'

<input checked="" type="checkbox"/>	Transfer svih informacija o delu
<input checked="" type="checkbox"/>	Knjige
<input type="checkbox"/>	Diplomski radovi
<input checked="" type="checkbox"/>	Magistarski radovi
<input type="checkbox"/>	Doktorske disertacije
<input type="checkbox"/>	Časopisi

PARAMETRI PRETRAŽIVANJA

<input checked="" type="checkbox"/>	Naslov dela:	<input type="text"/>
<input checked="" type="checkbox"/>	Autor:	<input type="text"/>
<input type="checkbox"/>	Ključne reči:	<input type="text"/>

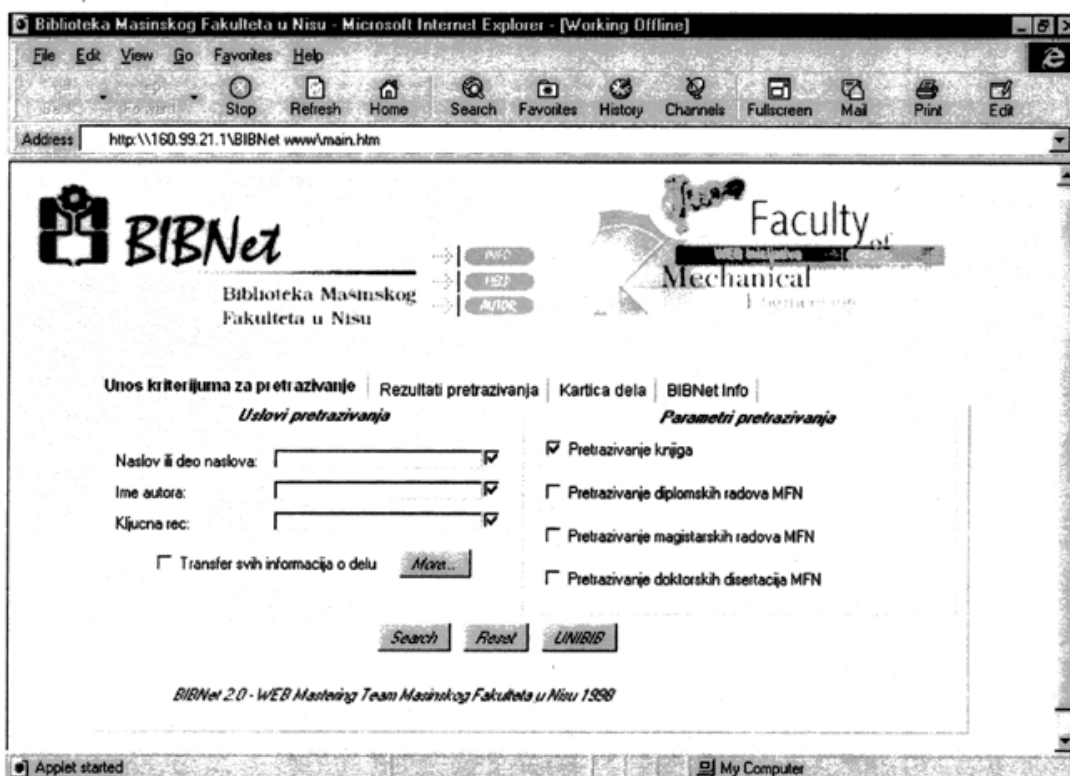
WHERE

```
Knjige.Naslov
LIKE '%<TextField1.getText()%'
AND
'Magistarski radovi'.Naslov
LIKE '%<TextField1.getText()%'
AND
Knjige.Autor
LIKE '%<TextField2.getText()%'
AND
'Magistarski radovi'.Autor
LIKE '%<TextField2.getText()%'
```

sl.3.1 Parametri i uslovi pretraživanja

Korisniku je ostavljeno da izabere prikaz minimalnog ili proširenog seta informacija o delu. Prošireni set informacija potpuno definiše delo ili rad kao bibliografsku jedinicu, a od ovakvog seta informacija se generiše kartica dela. Takođe, elementi traženog polja se mogu koristiti za formiranje narednog upita, tako da korisnik može, na osnovu jednog polja u bazi, sa informacijama od interesa, da dobije i pregled njemu srodnih, prema izabranom kriterijumu. Da bi ovo bilo moguće, mora da postoji kartica dela, odnosno da bude uključen prenos proširenog seta informacija.

Na slici 3.2, prikazana je WEB stranica biblioteke Mašinskog fakulteta u Nišu, sa učitanim apletom, na kojoj se vide elementi korisničkog interfejsa za definisanje uslova i parametara pretraživanja.



Slika 3.2 WEB stranica Biblioteke Mašinskog Fakulteta u Nišu

4. IZVRŠENJE SQL UPITA POSREDTVOM JDBC-ODBC VEZE (KORIŠĆENJE PARAMETRIZOVANIH UPITA)

JDBC podržava tri tipa iskaza:

- Statement
- PreparedStatement
- CallableStatement

JAVA program najpre smešta SQL iskaz u string bafer, a nakon toga sadržaj bafera prosledjuje ka DBMS, i to posredstvom nekog API poziva. Pre samog izvršenja, ovaj iskaz mora proći kroz tri koraka: sintaksnu proveru, optimizaciju i konverziju u izvršnu formu.

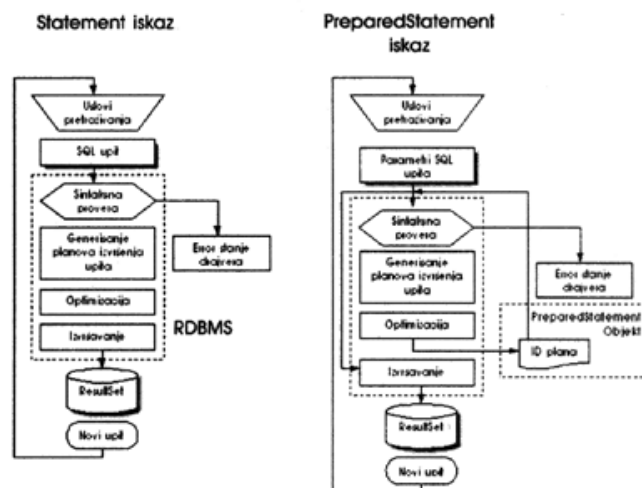
Pošto DBMS primi SQL iskaz, u obliku string promenljive, vrši se raščlanjivanje iskaza i provera ispravnosti sintakse. Ako iskaz ne prodje ovu proveru, DBMS prijavljuje error stanje drajveru koji služi za vezu sa JDBC interfejsom, i on generiše SQLException izuzetak.

U sledećem koraku, zavisno od DBMS, generiše se više takozvanih planova izvršenja upita (Query plan), od kojih se nakon optimizacije bira jedan, koji se, uslovno rečeno, kompajlira i, konačno izvršava.

Tri tipa iskaza, koje podržava JDBC se razlikuju u pripremi SQL iskaza i u njegovom izvršenju. U slučaju iskaza tipa PreparedStatement, aplikacija priprema SQL iskaz, korišćenjem određenog metoda koji prima SQL string i prosledjuje ga ka DBMS. DBMS prolazi kroz faze sintaksne provere, optimizacije planova izvršenja upita, generisanja izvršnog plana, ali ne izvršava SQL iskaz. DBMS vraća ID optimizovanog i kompajliranog plana izvršenja upita, koji je JDBC drajver, sačuvao u PreparedStatement objektu.

Pri izvršenju upita, drajver, zapravo DBMS-u šalje samo ID plana izvršenja upita i parametre. Ovo rezultira u manjem saobraćaju kroz mrežu.

Korišćenjem običnog Statement iskaza, svaki put kada drajver pošalje SQL upit ka DBMS, on mora da prodje kroz sintaksnu proveru, optimizaciju plana izvršenja upita i njegovo kompajliranje. U slučaju PreparedStatement iskaza, ovo nije neophodno, pa korisnik za kraće vreme dobija rezultate svog upita.



4.1 Algoritmi izvršenja klijent - server transakcije korišćenjem Statement i PreparedStatement iskaza

U prethodnoj verziji sistema za pretraživanje resursa biblioteke, izvršavan je tzv. generički upit, koji je bio iste strukture, nezavisno od uslova pretraživanja, definisanih od strane korisnika. U situaciji kada korisnik zada manji broj uslova od ukupnog broja ponuđenih, izvršavanje generičkog upita nije optimalno, jer algoritam za pretraživanje baze skroluje i po kolonama za koje nisu definisani uslovi.

Iako se pri definisanju uslova i parametara pretraživanja ne dobijaju generički upiti, već klijent aplet može da generiše upite raznovrsne strukture, navedene osobine PreparedStatement iskaza mogu u određenim situacijama da značajno doprinesu brzini rada.

5. ZAKLJUČAK

Pouzdanost troslojne klijent-server arhitekture, kao i portabilnost, robustnost i ugradjeni sigurnosni sistemi JAVA programske platforme garantuju pouzdane performanse informacionog sistema, projektovanog na ovim osnovama.

Pojava JAVA programskog jezika je prouzrokovala trend ubrzanog razvoja tehnologija za projektovanje distribuiranih baza podataka. Na osnovu obavljenog istraživanja [2], snimanja potreba i analize raspoloživih resursa, za realizaciju sistema za pretraživanje resursa biblioteke je izabrano prikazano rešenje.

Novi korisnički interfejs je uslovio povećanje performansi sistema za pretraživanje, tako što je

omogućio optimalno "ponašanje" krajnjeg korisnika. Pod ovim se podrazumeva kreiranje složenih, dinamičkih upita, koji treba da omoguće korisniku da u što manje koraka dodje do tražene informacije.

Sa druge strane, upotreba parametrizovanih upita, uz štednju mrežnih resursa i brži rad algoritama za pretraživanje na samom serveru baza podataka, u određenim situacijama može ubrzati rad sistema i do 50%.

6. LITERATURA

1. Milan Zdravković, Miroslav Trajanović, "Realizacija klijent-server arhitekture u programskom jeziku JAVA, na primeru informacionog sistema biblioteke Mašinskog Fakulteta u Nišu", YU INFO, 1998
2. Milan Zdravković, Miroslav Trajanović, "Specifičnosti troslojne klijent-server arhitekture, implementirane u JAVA okruženju", rad u pripremi
3. Joseph Weber, "Special Edition Using JAVA, 2nd edition", QUE, Indianapolis, 1996
4. Laura Lemay, Charles L. Perkins, Michael Morrison, "Teach Yourself JAVA in 21 Days, Professional Reference Edition", Sams.net Publishing, Indianapolis, 1996