Дистрибуирани библиотечки информациони систем БИСИС

Distributed Library Information System BISIS

Едиција
Рачунарске науке

Уредници едиције
проф. др Душан Сурла
проф. др Зора Коњовић

Издавач
Група за информационе технологије
Нови Сад, Бул. Краља Петра I 13
gint@eunet.yu

Издавач задржава сва права. Без писмене сагласности издавача није дозвољено прештампавање нити преузимање односно репродуковање књиге нити њених појединих делова.

CIP – Каталогизација у публикацији
Библиотека Матице српске, Нови Сад

021.63:004.4

ДИСТРИБУИРАНИ библиотечки информациони систем БИСИС / уредници Душан Сурла, Зора Коњовић ; [аутори Душан Сурла... и др.]. – Нови Сад : Група за информационе технологије, 2004 ([Нови Сад] : АБМ Економик). – 262 стр. : илустр. ; 25 cm. – (Едиција Рачунарске науке)

Део текста на енгл. језику. – Тираж 500. – Библиографија. – Abstracts.

ISBN 86-7444-006-1

1. Сурла, Душан
а) Библиотечко информациони систем БИСИС

Дистрибуирани библиотечки информациони систем БИСИС

Уредници

Душан Сурла
Зора Коњовић

Нови Сад 2004.

Едиција
Рачунарске науке

Уредници
др Душан Сурла
др Зора Коњовић

Аутори
др Душан Сурла
др Зора Коњовић
мр Биљана Пуповац
др Милан Видаковић
мр Татјана Зубић
др Милош Рацковић
др Бранко Милосављевић
мр Татјана Тошић
Милан Керац, дипл. инж.
Иван Нејгебауер, дипл. инж.

Рецензенти
проф. др Иван Седер, Fachhochschule Merseburg,
University of Applied Sciences, Germany
проф. др Маргита Кон-Поповска, Faculty of Natural Sciences and Mathematics,
University "Ss. Cyril and Methodius", Skopje, FYR Macedonia

Корице
Мирјана Исаков

Штампа
АБМ Економик

Тираж
500

Трошкове штампања ове монографије обезбедила је Европска Комисија преко TEMPUS пројекта JEP 16114.

Printing costs for this monograph were covered by the European Commission through the TEMPUS project JEP 16114.

Садржај

Foreword 9
Предговор 11
Author Information 13
Подаци о ауторима 15
Abstracts 17
Сажеци 23

1. Преглед развоја Библиотечког информационог система БИСИС 29
1.1. Захтеви библиотечке мреже 30
1.2. Верзије софтверског система БИСИС 31
1.2.1. БИСИС верзија 1 33
1.2.2. БИСИС верзија 2 и верзија 3 33
1.3. Дигитална библиотека Универзитета у Новом Саду 34
1.3.1. Успостављање Сервиса 35
1.3.2. Преузимање записа 35
1.3.3. Виртуелни централни каталог библиографских записа 37
1.4. XML технологије и библиотечки системи 38
1.5. Закључак 40

2. YUMARC формат 43
2.1. Дијаграм класа YUMARC формата 46
2.2. Апликација за ажурирање YUMARC формата 49
2.2.1. Прозор за ажурирање поља 49
2.3. Закључак 52

3. Моделирање и имплементација радног окружења библиотекара.....55
3.1. Дијаграм класа радног окружења библиотекара 56
3.2. Апликација за ажурирање радног окружења библиотекара 58
3.2.1. Прозор за ажурирање типа публикације 59
3.2.2. Прозор за ажурирање потпоља типа публикације 60
3.3. Закључак 62

4. Обрада библиографске грађе 65
4.1. Обрада записа библиографске грађе 66
4.1.1. Претрага библиографске грађе 66
4.1.2. Креирање новог записа 66
4.1.3. Креирање новог записа на основу постојећег 67
4.1.4 Модификација постојећег записа 68
4.1.5. Обрада меморијске структуре која репрезентује запис 69

4.1.6. Снимање записа 71
4.2. Архитектура апликације 72
4.2.1. Пакети клијент апликације 72
4.2.2. Класе за обраду меморијске структуре 73
4.3. Имплементација 77
4.3.1 Обрада записа у директоријуму 79
4.4. Закључак 80

5. Команде библиотечког информационог система 83
5.1. Дијаграм случајева коришћења команди 86
5.2. Дијаграм класа команди 88
5.2.1. Дијаграм класа команде delete 90
5.3. Пример коришћења команди 91
5.4. Закључак 93

6. Сервер за узајамну каталогизацију 95
6.1. Сервер за узајамну каталогизацију 96
6.1.1. Рад са удаљеном базом библиографских записа 96
6.1.2. Архитектура серверске апликације 96
6.2. Имплементација 98
6.2.1. Имплементација RMI технологијом 98
6.2.2. Имплементација у EJB технологији 99
6.3. Узајамна каталогизација употребом агентске технологије 102
6.4. Закључак 103

7. Извештавање о библиографској грађи.....105
7.1. Библиотечки извештаји 105
7.1.1. Извештаји о библиографској грађи 106
7.1.2. ISBD стандард 107
7.1.3. Каталошки листићи и други извештаји 108
7.2. Стуктура извештаја о библиографској грађи 110
7.3. Модел извештаја о библиографској грађи 112
7.4. Имплементација 114
7.5. Закључак 118

8. Текст сервер UNIMARC записа 121
8.1. Функције текст сервера 125
8.2. Модел података текст сервера 127
8.3. Подсистем за руковање записима 129
8.4. Подсистем за претраживање записа 131
8.4.1. SELECT упити 131
8.4.2. Коректност SELECT упита 134
8.4.3. EXPAND упити 136
8.4.4. Класе подсистема за претраживање записа 137
8.5. Закључак 141

9. Корисничко претраживање 143
9.1. Елементи софтверске архитектуре 145
9.2. Корисничке сесије 146
9.3. Структура веб сајта 148
9.4. Интерпретација упита 150
9.5. Имплементација 152
9.6. Закључак 154

10. Коришћење библиотечке грађе.....157
10.1. Дијаграм класа коришћења библиотечке грађе 158
10.2. Апликација Циркулација 163
10.2.1. Учлањење 164
10.2.2. Задужење и раздужење 165
10.2.3. Извештаји 166
10.2.4. Систем 168
10.3. Извештавање у циркулацији 168
10.3.1. Статистички извештаји о корисницима библиотеке 169
10.3.2. Извештаји о читаности библиотечке грађе 171
10.4. Закључак 175

11. Интернационализација и локализација система БИСИС.....177
11.1. Руковање вишејезичним текстом 178
11.1.1. Кодни распореди 178
11.1.2. Складиштење докумената 180
11.1.3. Приказ и унос текста 182
11.1.4. Претраживање 183
11.2. Локализација корисничког интерфејса 185
11.2.1. Подршка Unicode стандарду у Swing и JSP апликацијама 185
11.2.2. Класе за локализацију у програмском језику Java 186
11.2.3. Поступак интернационализације и локализације Java програма 189
11.3. Закључак 190

12. Мрежна инфраструктура система БИСИС 193
12.1. Локална рачунарска мрежа библиотеке 193
12.1.1. Дефиниција и функције 193
12.1.2. Физичка архитектура локалне рачунарске мреже библиотеке 193
12.1.3. Полазне поставке 194
12.1.4. Комуникациона чворишта локалне рачунарске мреже библиотеке 196
12.1.5. Комуникациони путеви 202
12.1.6. Корисничка прикључна места 204
12.1.7. Остала опрема 204
12.2. Веза ка другим мрежама и Интернету 205
12.2.1. Повезивање локалне мреже библиотеке на Интернет 205
12.2.2. Међусобно повезивање локалних рачунарских мрежа библиотека 205
12.3. Минимални захтеви логичке архитектуре мреже библиотеке 205

12.3.1. План употребе јавног IP адресног простора 206
12.3.2. Хијерархијско приватно IP адресирање 206
12.3.3. Хијерахијска организација имена 207
12.3.4. Виртуелне локалне рачунарске мреже 207
12.4. Организација права приступа појединим модулима система 207
12.5. Пример 208
12.5.1. Физичка архитектура 209
12.5.2. Логичка архитектура 211
12.6. Закључак 216

Прилог 1: Приказ магистарских теза и докторских дисертација 219

Прилог 2: Библиографија 251

Foreword

The development of the library information system BISIS started in 1992 with the project *Plan of the Development of the Library Network of Serbia*. The Ministry of Science and Development of the Republic of Serbia provided financing for the project. The same Ministry financed development of the System of the Scientific and Technological Information of Serbia during the period 1993 - 1996. One of the results emerging from this development is the library information system entitled BISIS ver. 1. The development of the BISIS after 1996 was resumed at the University of Novi Sad, resulting in the BISIS ver. 2 and BISIS ver. 3. The BISIS ver. 3 supports the following functions: *Defining the Standards for Bibliographical Material Processing and Records Structuring*, *Defining the Librarian Working Environment*, *Creating Bibliographical Material, Libarary Reporting and Documenting*, *User Search* and *Usage of the Bibliographical Material*.

For modelling of the system the Unified Modelling Language (UML) is used. The system BISIS ver. 3 is implemented as an Internet application in *Java* environment. The principal characteristics of this application are: specialised text server for indexing and retrieving UNIMARC bibliographic records; multi-tiered system architecture; deployment of the Java platform and independence of relational database management system; Unicode standard consequently implemented throughout the complete application.

The aim of publishing this monograph is to collect and present in a single printed document modelling and implementation of all functions of the system BISIS ver.3. The monograph contains the following chapters:

1. An Overview of the Development of the Library Information System BISIS
2. UNIMARC Format
3. Modelling and Implementation of the Librarian Working Environment
4. Bibliographical Material Processing
5. Commands of the Library Information System
6. Shared Cataloguing Server
7. Reporting on Bibliographical Material
8. Text Server of UNIMARC Records
9. User Search
10. Usage of the Bibliographical Material

11. Internationalisation and Localisation of the BISIS System
12. BISIS Network Infrastructure

In the first chapter the editors presented an overview of the development of the system so far, as well as the directions of its future development. The other chapters are devoted to the presentation of modelling and implementation of the system BISIS ver.3. The functionality of the system is described by Use case diagrams. The static model of the system is described by Class diagrams. The usage of the application is also illustrated.

Two appendices go along with the aforementioned chapters. Appendix 1 presents an overview of the 12 M.Sc. and 2 Ph.D. theses with subject matter related to the development of the Library Information System BISIS. The bibliography containg published results is presented in Appendix 2. As a result of the development of the BISIS system, 148 scientific papers have been published in international and national journals and conference proceedings. In addition to it, two books have been published and approx. 15 graduation theses dealing with modelling and implementation of library information system have been done.

We appreciate very much the work and valuable suggestions of the reviewers Prof. Dr Ivan Seder and Prof. Dr Margita Kon-Popovska who have contibuted significantly to the clarity of the final version of the text. The reviewers have had an opportunity to get to know the system BISIS in detail through joint work on the TEMPUS project *Web Based Interuniversity Library Network* (TEMPUS JEP 16114). Prof. Dr Ivan Seder represents Fachhochschule Merseburg from Merseburg, Germany which is the project contractor. Prof. Dr Margita Kon-Popovska represents University Ss. Cyril and Methodius from Skopje, FYR Macedonia which is one of the partners in the consortium. Within this project the localisation of the system BISIS is done to German, Macedonian, Greek and English languages. The printing and publishing costs of this publication are fully covered from the budget of the above mentioned TEMPUS project.

Novi Sad, 2004

Editors

Предговор

Развој библиотечког информационог система БИСИС започет је 1992. године израдом пројекта *План развоја библиотечке мреже Србије*. Пројекат је финансирало Министарство за науку и технологију Републике Србије. Исто министарство је, у периоду од 1993. до 1996. године, финансирало развој Система Научних и Технолошких Информација Србије. Део резултата тог развоја је библиотечки информациони систем, назван БИСИС вер. 1. Развој система БИСИС настављен је и после 1996. године на Универзитету у Новом Саду. Резултат тог развоја су БИСИС вер. 2 и БИСИС вер. 3. Верзија 3 подржава следеће функције: Дефинисање стандарда за одржавање библиографске грађе и структурирање записа, Дефинисање радног окружења библиотекара, Формирање библиографске грађе, Библиотекарско извештавање и документовање, Корисничко претраживање и Коришћење библиографске грађе.

Моделирање система извршено је коришћењем обједињеног језика моделирања (*Unified Modeling Language* – UML). Имплементација система БИСИС вер. 3 реализована је као Интернет апликација у Java окружењу. Главне карактеристике ове апликације су: специјализовани текст сервер за индексирање и претраживање библиографских записа у UNIMARC формату; вишеслојна архитектура система; коришћење Java платформе и независност од коришћеног релационог система за управљање базом података; доследно спроведена подршка за *Unicode* стандард у целокупној апликацији.

Циљ публиковања ове монографије је да се на једном месту прикаже моделирање и имплементација свих функција система БИСИС вер. 3. Монографија садржи следећа поглавља:

1. Преглед развоја библиотечког информационог система БИСИС
2. UNIMARC формат
3. Радно окружење библиотекара
4. Обрада библиографске грађе
5. Команде библиотечког информационог система
6. Сервер за узајамну каталогизацију
7. Извештавање о библиографској грађи
8. Текст сервер UNIMARC записа
9. Корисничко претраживање
10. Коришћење библиотечке грађе
11. Интернационализација и локализација система БИСИС
12. Мрежна инфраструктура система БИСИС

Редактори су у првом поглављу дали преглед досадашњег развоја система БИСИС, као и даље правце његовог развоја. У осталим поглављима дат је приказ моделирања и имплементације система БИСИС вер. 3. Помоћу случајева коришћења описана је функционалност система. Статички део система дат је помоћу дијаграма класа. Илустровано је и коришћење апликације.

Поред наведених поглавља постоје и два прилога. У Прилогу 1 дат је приказ 12 магистарских теза и 2 докторске дисертације у којима су се теме истраживања односиле на развој система БИСИС. Библиографија са објављеним резултатима истраживања на развоју система БИСИС дата је у Прилогу 2. Објављена су 148 научних радова, који су публиковани у међународним и домаћим часописима и зборницима са конференција. Поред тога, објављене су две књиге и урађено је и око 15 дипломских радова са темом из области моделирања и имплементације библиотечког информационог система.

Захваљујемо рецензентима на веома корисним примедбама и сугестијама које су допринеле да коначна верзија текста буде јаснија. Рецензенти су се детаљно упознали са системом БИСИС вер. 3 у оквиру ТЕМПУС пројекта под називом *Web Based Interuniversity Library Network* (TEMPUS JEP 16114). Проф. др Иван Седер је представник Fachhochschule Merseburg, University of Applied Sciences, Немачка који је контрактор овог пројекта. Проф. др Маргита Кон-Поповска је представник Faculty of Natural Sciences and Mathematics, University “Ss. Cyril and Methodius”, Скопје, БЈР Македонија који је један од партнера у конзорцијуму. У оквиру овог пројекта извршена је локализација система БИСИС вер. 3 за немачки, грчки, македонски и енглески језик. Трошкове штампања ове монографије покрива наведени ТЕМПУС пројекат.

Нови Сад, 2004.

Уредници

Author Information

Dušan Surla, PhD, Full Professor
Faculty of Science
Department of Mathematics and Informatics
21000 Novi Sad
Trg Dositeja Obradovića 4
Serbia & Montenegro
Tel: +381 21 458 743
Fax: +381 21 350 458
email: surla@uns.ns.ac.yu

Zora Konjović, PhD, Full Professor
Faculty of Engineering
Computing and Control Department
21000 Novi Sad
Fruškogorska 11
Serbia & Montenegro
Tel: +381 21 450 716
Fax: +381 21 350 727
email: ftn_zora@uns.ns.ac.yu

Biljana Pupovac, MSc
Faculty of Science
21000 Novi Sad
Trg Dositeja Obradovića 4
Serbia & Montenegro
Tel: +381 21 458 743
Fax: +381 21 350 458
email: biljanap@uns.ns.ac.yu

Milan Vidaković, PhD, Assistant Professor
Faculty of Engineering
Computing and Control Department
21000 Novi Sad
Fruškogorska 11
Serbia & Montenegro
Tel: +381 21 450 716
Fax: +381 21 350 727
email: minja@uns.ns.ac.yu

Tatjana Zubić, MSc
University of Novi Sad, ARMUNS
21000 Novi Sad
Trg Dositeja Obradovića 6
Serbia & Montenegro
Tel: +381 21 350 122 ext 101
Fax: +381 21 450 418
email: tanja@uns.ns.ac.yu

Miloš Racković, PhD, Associate Professor
Faculty of Science
Department of Mathematics and Informatics
21000 Novi Sad
Trg Dositeja Obradovića 4
Serbia & Montenegro
Tel: +381 21 458 743
Fax: +381 21 350 458
email: rackovic@im.ns.ac.yu

Branko Milosavljević, PhD, Assistant Professor
Faculty of Engineering
Computing and Control Department
21000 Novi Sad
Fruškogorska 11
Serbia & Montenegro
Tel: +381 21 450 716
Fax: +381 21 350 727
email: mbranko@uns.ns.ac.yu

Tatjana Tošić, MSc
University of Novi Sad, ARMUNS
21000 Novi Sad
Trg Dositeja Obradovića 6
Serbia & Montenegro
Tel: +381 21 350 122 ext 101
Fax: +381 21 450 418
email: ttosic@uns.ns.ac.yu

Milan Kerac, BSc
University of Novi Sad, ARMUNS
21000 Novi Sad
Trg Dositeja Obradovića 6
Serbia & Montenegro
Tel: +381 21 350 122 ext 808
Fax: +381 21 450 418
email: kerac@uns.ns.ac.yu

Ivan Nejgebauer, BSc
University of Novi Sad, ARMUNS
21000 Novi Sad
Trg Dositeja Obradovića 6
Serbia & Montenegro
Tel: +381 21 350 122 ext 808
Fax: +381 21 450 418
email: ian@uns.ns.ac.yu

Подаци о ауторима

др Душан Сурла, редовни професор
Природно-математички факултет
Департман за математику и информатику
21000 Нови Сад
Трг Доситеја Обрадовића 4
Србија и Црна Гора
Тел.: 021 458 743
Факс: 021 350 458
email: surla@uns.ns.ac.yu

др Зора Коњовић, редовни професор
Факултет техничких наука
Катедра за рачунарске науке и информатику
21000 Нови Сад
Трг Доситеја Обрадовића 6
Србија и Црна Гора
Тел.: 021 450 716
Факс: 021 350 727
email: ftn_zora@uns.ns.ac.yu

мр Биљана Пуповац
Природно-математички факултет
21000 Нови Сад
Трг Доситеја Обрадовића 4
Србија и Црна Гора
Тел.: 021 458 743
Факс: 021 350 458
email: biljanap@uns.ns.ac.yu

др Милан Видаковић, доцент
Факултет техничких наука
Катедра за рачунарске науке и информатику
21000 Нови Сад
Трг Доситеја Обрадовића 6
Србија и Црна Гора
Тел.: 021 450 716
Факс: 021 350 727
email: minja@uns.ns.ac.yu

мр Татјана Зубић
Универзитет у Новом Саду, АРМУНС
Трг Доситеја Обрадовића 6
21000 Нови Сад
Србија и Црна Гора
Тел.: 021 350 122 лок. 101
Факс: 021 450 418
email: tanja@uns.ns.ac.yu

др Милош Рацковић, ванредни професор
Природно-математички факултет
Департман за математику и информатику
21000 Нови Сад
Трг Доситеја Обрадовића 4
Србија и Црна Гора
Тел.: 021 458 743
Факс: 021 350 458
email: rackovic@im.ns.ac.yu

др Бранко Милосављевић, доцент
Факултет техничких наука
Катедра за рачунарске науке и информатику
21000 Нови Сад
Трг Доситеја Обрадовића 6
Србија и Црна Гора
Тел.: 021 450 716
Факс: 021 350 727
email: mbranko@uns.ns.ac.yu

мр Татјана Тошић
Универзитет у Новом Саду, АРМУНС
Трг Доситеја Обрадовића 6
21000 Нови Сад
Србија и Црна Гора
Тел.: 021 350 122 лок. 101
Факс: 021 450 418
email: ttosic@uns.ns.ac.yu

Милан Керац, дипл. инж.
Универзитет у Новом Саду, АРМУНС
Трг Доситеја Обрадовића 6
21000 Нови Сад
Србија и Црна Гора
Тел.: 021 350 122 лок. 808
Факс: 021 450 418
email: kerac@uns.ns.ac.yu

Иван Нејгебауер, дипл. инж.
Универзитет у Новом Саду, АРМУНС
Трг Доситеја Обрадовића 6
21000 Нови Сад
Србија и Црна Гора
Тел.: 021 350 122 лок. 808
Факс: 021 450 418
email: ian@uns.ns.ac.yu

Abstracts

Distributed Library Information System BISIS

D. Surla and Z. Konjović, editors

Chapter 1
An Overview of the Development of the Library Information System BISIS

D. Surla, Z. Konjović

The development of the library software system BISIS started in 1992 and several versions have been developed so far. The first version of the system was deployed in faculty libraries in 1996, while the third version is being exploited at present. During its development the system BISIS evolved from the application developed for a centralised architecture to the Internet oriented application. Other important characteristics of the BISIS are specialised text server for UNIMARC records, consequent implementation of the Unicode standard and software globalisation and localisation principles. During the year 2003 at the University of Novi Sad the software system *Networked Digital Library of Graduation, Master and Doctoral Theses* was developed and connected to the world's networked digital library (NDLTD). The research activities related to the development of the new versions of the system BISIS are still in progress at the University of Novi Sad.

Chapter 2
YUMARC Format

B. Pupovac, T. Zubić

The Library Information System BISIS ver. 3 is based on the YUMARC format. In this Chapter the UML class diagaram of the YUMARC format is presented. The database scheme of the YUMARC format was created using this diagram and the case tool *Rational Rose*. The application for updating YUMARC format provides for updating the database of the YUMARC format. This Chapter also presents the description of the window for updating YUMARC format fields as an illustrative example and the description of the relations between graphical components of this window and the columns of the corresponding relational model table.

Chapter 3
Modelling and Implementation of the Librarian Working Environment

B. Pupovac

The Librarian Working Environment defines the data necessary for adapting the application for bibliographical material processing to specific demands of a single librarian and the types of documents processed by this particular librarian. This Chapter presents UML class diagaram of the Librarian Working Environment. The application for updating database of Librarian Working Environment, which is implemented as a module of the software system BISIS, provides this updating. The Chapter features descriptions of the windows for updating concepts of publication type and subfileds of publication type, which build an initial segment defining the structure supporting hierarchical publication processing. The relation between graphical components of these windows and the columns of the corresponding table of the Librarian Working Environment database is also described.

Chapter 4
Bibliographical Material Processing

M. Vidaković

Bibliographical material processing is the central part of the library information system BISIS. It is provided by the client application of the library information system. The client application supports the following operations on bibliographical records: bibliographical records search, creating new records, creating new records based on existing ones, modification and deleting existing records. Record editing is based on the UNIMARC format of the record. Therefore, the following operations on the record are available: adding, modification and deleting the field, subfield and subsubfield, as well as modification of indicators. Operations are not performed directly on the record. Instead, they are performed on the memory structure representing a bibliographical record. All the operations are consistent with the UNIMARC standard and with the librarian environment settings.

Chapter 5
Commands of the Library Information System

B. Pupovac

This Chapter presents the list of the commands used in the application for bibliographical data processing and the description of the functionality of each command. It also presents the use case diagram and the brief description of each

use case. The class digram of the command *delete* and the brief description of this diagram are also shown. The final part of this Chapter shows an example illustrating usage of the *delete* command. The example describes the way of issuing the command, functionality of the command and the appearance of the application window while the command is being executed.

Chapter 6
Shared Cataloguing Server

M. Vidaković

Shared cataloguing represents the possibility of bibliographical material interchange between libraries. Bibliographical records editing and creation is performed by librarians and is based on the UNIMARC international standard. The main goal of shared cataloguing is to enable librarians to download records from other libraries in order to complete the local record database. This means that it is necessary to enable librarians to search, select, download and store locally records which exist in a remote record database. Records downloaded from the remote database are saved locally in compliance with the local database environment settings.

Chapter 7
Reporting on Bibliographical Material

T. Zubić, M. Racković

Object-oriented specification and implementation of the BISIS system reporting process are illustrated by the class diagram representing bibliographical data reports hierarchical structure and the system of rules applied to that structure. Bibliographical data reports are formed by applying defined rules and report structure to records from bibliographical databases. The structure of bibliographical data reports is hierarchical. To each element of hierarchy some rules can be applied. Rules, according to the ISBD, define an element data source, punctuation, and final layout. The rules actually form content of the structure elements, change their properties and define visibility of the element content in the report. Since the BISIS supports the UNIMARC format, generation of bibliographical data reports requires conversion from the UNIMARC records to the bibliographical description defined by the ISBD.

Chapter 8
Text Server for UNIMARC Records

B. Milosavljević

A text server is a software system for retrieval of textual documents. Library information systems use the UNIMARC standard as a document format. The presented text server utilises a variant of the Dialog query language. This chapter describes the text server software architecture, its data model, as well as algorithms for query interpretation. The text server uses a relational database for data persistence, and is independent from a particular relational database server. It is designed as a middle-tier component in a three-tier system.

Chapter 9
User Search

B. Milosavljević

User search is a web-based interface for searching bibliographical databases designed for end-users of a library. The implementation relies on the text server for UNIMARC records. The web site is designed to support users with different skill levels in data retrieval. This chapter describes the software architecture, user session management, the structure of the web site and the process of query interpretation.

Chapter 10
Usage of the Bibliographical Material

T. Tošić, D. Surla

Usage of the bibliographical material is the important work segement of each library. This Chapter presents detailed description of the bibliographical material circulation class diagram. It also presents a brief description of the application *Circulation* which is implemented within the software system BISIS ver. 3. The basis for the implementation is the analysis of the usage of the bibliographical material in public city libraries and faculty libraries.

Chapter 11
Internationalisation and Localisation of the BISIS System

M. Vidaković, B. Milosavljević

Internationalisation and localisation means that programmes should be made so as to provide for regional and language independence. This means that existing programmes should be modified and new programmes should be made facilitating adaptation to a particular region and language without intervention on the source code. The library information system BISIS enables equivalent usage of all available languages supported by the operating system (both fonts and keyboard layouts). This is due to the Unicode support present in the Java programming language. On the other hand, for the internal text representation the UNIMARC standard is used. This standard enables an efficient conversion of different letters and a joint Cyrillic and Latin search. As a final result, the BISIS system enables the analogous use of both Cyrillic and Latin for input and display, and for defining query text, where both Cyrillic and Latin queries produce the same result.

Chapter 12
BISIS Network Infrastructure

M. Kerac, I. Nejgebauer

This chapter describes the physical and logical architecture of the computer network which supports the library information system BISIS. The chapter suggests the means of connecting the local library computer network to the Internet and describes the organisation of access rights to the individual modules of the library information system BISIS.

Сажеци

Дистрибуирани библиотечки информациони систем БИСИС

Д. Сурла, З. Коњовић, уредници

Поглавље 1
Преглед развоја библиотечког информационог система БИСИС

Д. Сурла, З. Коњовић

Библиотечки софтверски систем БИСИС развија се од 1992. године и до данас је израђено више верзија. Прва верзија система почела је да се користи у факултетским бибиотекама 1996. године, док је сада у експлоатацији трећа верзија система. У току свог развоја БИСИС је еволуирао од апликације развијене за централизовану архитектуру до Интернет оријентисане апликације. Остале значајне карактеристике система БИСИС су специјализовани текст сервер за UNIMARC записе, доследна примена *Unicode* стандарда и принципа глобализације и локализације софтвера. У току 2003. године на Универзитету у Новом Саду развијен је софтверски систем *Мрежна дигитална библиотека дипломских радова, магистарских теза и докторских дисертација*, која је званично прикључена светској мрежи дигиталних библиотека (NDLTD). На Универзитету у Новом Саду су и даље у току истраживања на развоју нових верзија система БИСИС.

Поглавље 2
YUMARC формат

Б. Пуповац, Т. Зубић

Библиотечки софтверски систем БИСИС вер. 3 базиран је на YUMARC формату. У овом поглављу приказан је дијаграм класа YUMARC формата моделиран коришћењем UML нотације. Помоћу тог дијаграма формирана је шема базе података YUMARC формата коришћењем алата *Rational Rose*. Ажурирања података базе података YUMARC формата омогућава апликација за ажурирање YUMARC формата. У овом поглављу дат је и опис прозора за ажурирање поља YUMARC формата као илустративни пример и описана је веза између графичких компоненти овог прозора и колона одговарајуће табеле релационог модела.

Поглавље 3
Моделирање и имплементација радног окружења библиотекара
Б. Пуповац

Радно окружење библиотекара дефинише податке који се користе у циљу прилагођавања апликације за обраду библиографске грађе специфичним захтевима сваког библиотекара, као и типовима докумената које он обрађује. У овом поглављу приказан је дијаграм класа радног окружења библиотекара, моделиран коришћењем UML нотације. Ажурирања базе података радног окружења библиотекара омогућава апликација за ажурирање радног окружења библиотекара имплементирана у оквиру софтверског система БИСИС. У овом поглављу дат је и опис прозора за ажурирање концепата типа публикације и потпоља типа публикације као почетног сегмента у дефинисању структуре за хијерархијску обраду публикација. Описана је и веза између графичких компоненти ових прозора и колона одговарајуће табеле базе података радног окружења библиотекара.

Поглавље 4
Обрада библиографске грађе
М. Видаковић

Обрада библиографске грађе представља централни део библиотечког софтверског система БИСИС. Обрада библиографске грађе обухвата креирање нових записа, као и измену и брисање постојећих. Приликом креирања нових записа, као и приликом обраде постојећих, обезбеђена је могућност додавања, измене и брисања UNIMARC поља, потпоља и потпотпоља, као и могућност обраде индикатора. Наведене операције се не врше директно над записима, већ на меморијској структури која их репрезентује. Приликом снимања, врши се конверзија из интерне меморијске структуре у UNIMARC записе. Све операције над записима су у складу са UNIMARC стандардом, као и са окружењем библиотекара.

Поглавље 5
Команде библиотечког информационог система
Б. Пуповац

У овом поглављу дат је списак команди које се користе у апликацији за обраду библиографске грађе, као и опис њихове функционалности. Приказан

је дијаграм случајева коришћења команди и дат је кратак опис за сваки случај коришћења. Такође, приказан је дијаграм класа команде *delete* и дат кратак опис овог дијаграма. На крају поглавља, приказан је илустративни пример коришћења команде *delete* у апликацији за обраду библиографске грађе који описује начин задавања команде, функционалност команде и изглед прозора апликације током извршавања ове команде.

Поглавље 6
Сервер за узајамну каталогизацију
М. Видаковић

Узајамна каталогизација представља могућност размене података између две библиотеке на нивоу записа. Посао обраде и уноса библиографске грађе обављају библиотекари на основу међународног стандарда UNIMARC. Циљ узајамне каталогизације је да библиотекари могу да преузимају записе из других библиотека у циљу комплетирања сопствене базе записа у контролисаним условима. То значи да треба обезбедити библиотекару могућност да претражује друге базе записа, одабере записе и меморише их у сопственој локалној бази, модификоване у складу са специфичностима библиотеке којој локална база припада.

Поглавље 7
Извештавање о библиографској грађи
Т. Зубић, М. Рацковић

Дефинисан је и имплементиран објектно-оријентисани модел извештаја о библиографској грађи који моделира њихову хијерархијску структуру као и систем правила који се над том структуром примењују. Применом правила и структуре извештаја на конкретне податке из записа базе библиографске грађе, формира се извештај о библиографској грађи и презентује се кориснику. Структура извештаја о библиографској грађи је хијерархијска, при чему је за сваки елеменат хијерархије дефинисан извор података, интерпункција и форма приказа. Над елементима хијерархије – концептима, су дефинисана правила која формирају садржај елемента структуре, мењају неке његове особине или одређују да ли ће се елемент појавити на извештају или не. Систем БИСИС подржава UNIMARC формат. Приликом извештавања врши се конверзија UNIMARC записа у формат библиографског описа дефинисаног ISBD-ом.

Поглавље 8
Текст сервер UNIMARC записа
Б. Милосављевић

Текст сервер је софтверски систем намењен претраживању текстуалних докумената. Формат текстуалних докумената у библиотечким системима дефинисан је UNIMARC стандардом. Приказани специјализовани текст сервер за UNIMARC записе користи варијанту Dialog упитног језика. Описана је софтверска архитектура текст сервера, модел података, као и алгоритми интерпретације упита. Текст сервер користи релациони систем за управљање базама података, али је независан од конкретног система. Предвиђен је за рад у средњем слоју трослојне архитектуре.

Поглавље 9
Корисничко претраживање
Б. Милосављевић

Корисничко претраживање представља веб базирани интерфејс за претраживање фонда библиографске грађе намењен за крајње кориснике библиотеке. Ово претраживање ослања се на текст сервер UNIMARC записа. Концепција веб сајта подржава рад са корисницима различитих нивоа вештине. У овом поглављу су описани софтверска архитектура овог подсистема, механизам управљања корисничким сесијама, структура самог веб сајта и начин интерпретирања упита.

Поглавље 10
Коришћење библиотечке грађе
Т. Тошић, Д. Сурла

Коришћење библиотечке грађе представља значајан сегмент рада сваке библиотеке. У овом поглављу је детаљно описан дијаграм класа коришћења библиотечке. Укратко је описана апликација *Циркулација*, имплементирана у оквиру софтверског система БИСИС верзија 3.01. Имплементација је реализована на основу анализе процеса коришћења библиотечке грађе у факултетским и градским библиотекама.

Поглавље 11
Интернационализација и локализација система БИСИС

М. Видаковић, Б. Милосављевић

Интернационализација и локализација програма представља потребу да се програми пишу тако да буду независни од конкретног региона и језика, и да се лако прилагоде њима. Ова потреба изискује прилагођавање постојећих апликација и писање нових на такав начин да се прилагођавање програма изводи без интервенције на коду. Библиотечки информациони систем БИСИС, у оквиру својих функција, омогућава равноправно коришћење свих писама за потребе писања текста за које постоји одговарајућа системска подршка (распореди тастатура и фонтови). Као основа имплементације која подржава ову особину користи се стандардни *Unicode* кодни распоред. Овај кодни распоред је подржан у програмском језику Java, који је употребљен за имплементацију програмског система БИСИС. Са друге стране, за потребе интерне репрезентације и складиштења текста систем користи кодни распоред дефинисан UNIMARC стандардом. Употреба овог стандарда омогућава ефикасну конверзију различитих писама и обједињено претраживање ћирилице и латинице.

Поглавље 12
Мрежна инфраструктура система БИСИС

М. Керац, И. Нејгебауер

У овом поглављу су описани физичка и логичка архитектура рачунарске мреже за подршку библиотечког информационог система БИСИС. Предложен је начин повезивања локалне рачунарске мреже библиотеке са Интернетом и описана је организација права приступа појединим модулима библиотечког информационог система БИСИС.

Поглавље 1

Преглед развоја Библиотечког информационог система БИСИС

Д. Сурла, З. Коњовић

Министарство за науку и технологију Републике Србије је у току 1991. године започело активности на развоју Система Научних и Технолошких Информација Србије (СНТИС). Један од сегмената СНТИС-а је и Бибиотечка Мрежа (БМ) Србије. Концепција БМ СНТИС дата је у посебном Пројекту [1]. СНТИС је конципиран као отворени кооперативни дистрибуирани информациони систем. У том систему, јединственост библиотечког информационог система не остварује се централизацијом база података већ кооперативним радом различитих аутономних јединица на основу дефинисаних стандарда.

Министарство за науку и технологију Републике Србије у оквиру СНТИС-а финансирало је, у периоду од 1993. до 1996. године, пројекат *Систем за формирање и претраживање информација*. Руководилац овог пројекта био је проф. др Бранислав Лазаревић. Пројекат је реализован у заједничком раду неколико научно-истраживачких институција (Факултет организационих наука, Београд; Институт "Михајло Пупин", Београд; Природно-математички факултет, Нови Сад; Универзитет у Нишу, ЈУНИС; Рачунарски центар Универзитета у Београду).

Резултати истраживања на овом пројекту приказани су у монографији [2]. Резултати потпројекта *Пројектовање и реализација Јединственог система за индексирање и претраживање докумената* приказани су у 6. поглављу (реализатор Институт "Михајло Пупин", Београд), а потпројекта *Пројектовање и изградња Библиотечког подсистема СНТИС* у 8. поглављу (реализатор Природно-математички факултет, Нови Сад). Као резултат истраживања на ова два потпројекта добијен је софтверски систем за библиотечко пословање, назван БИСИС вер. 1.

Развој система БИСИС настављен је и после 1996. године на Универзитету у Новом Саду. Резултат тог развоја су БИСИС вер. 2.0 и БИСИС вер.

3.0. Универзитет у Новом Саду је са системом БИСИС вер. 3.0 у оквиру ТЕМПУС пројеката добио 2002. године двогодишњи пројекат под називом *Web Based Interuniversity Library Network* (JEP 16114).

У току 2003. на Универзитету у Новом Саду развијен је и пуштен у рад софтверски систем *Дигитална библиотека дипломских радова, магистарских теза и докторских дисертација*. Пројекат је финансирао Покрајински секретаријат за науку и технолошки развој АП Војводине. Поред тога, у току су интензивна истраживања у области примене XML технологија у пројектовању библиотечких информационих система.

У Прилогу 1 дат је приказ 12 магистарских теза и 2 докторске дисертација у којима су се теме истраживања односиле на развој система БИСИС. Библиографија са објављеним резултатима истраживања на развоју система БИСИС дата је у Прилог 2. Објављено је 165 научних радова а у Табели 1.1 приказан је број тих радова по категоријама. Поред тога, објављене су две књиге и урађено је и око 15 дипломских радова у оквиру развоја система БИСИС.

Категорија	**Број**
Монографија националног значаја (R13)	3
Поглавље у монографији (R23)	13
Признати програмски систем на међународном нивоу (R31)	1
Рад у водећем часопису међународног значаја (R51)	1
Рад у часопису међународног значаја (R52)	6
Пленарно предавање по позиву на скупу међународног значаја штампано у целини (R53)	2
Рад саопштен на скупу међународног значаја штампан у целини (R54)	26
Рад у часопису националног значаја (R62)	14
Рад саопштен на скупу националног значаја штампан у целини (R65)	72
Рад саопштен на скупу националног значаја штампан у изводу (R73)	13
Одбрана докторске дисертације (R81)	2
Одбрана магистарске тезе (R82)	12

Табела 1.1. Преглед броја објављених радова по категоријама

1.1. Захтеви библиотечке мреже

У пројекту [1] успостављени су основни елементи развоја компјутеризоване библиотечке мреже у погледу опреме и програмске подршке као и стандарда који се морају примењивати приликом пројектовања и набавке опреме, израде библиографских информација, односно база података библио-

графских записа. Наведени су задаци БМ СНТИС на локалном нивоу као и на нивоу информационог сервиса.

Задаци БМ СНТИС-а који се остварују на локалном нивоу су:

– локална подршка узајамне каталогизације,
– *on-line* претраживање локалних база,
– OPAC,
– набавка,
– циркулација,
– библиотечка статистика,
– контрола комплетности серијских публикација,
– локална подршка међубиблиотечке позајмице,
– локално издавање секундарних публикација и
– локални сегмент координиране набавке.

Задаци БМ СНТИС-а који се остварују на нивоу информационог сервиса су:

– узајамна каталогизација примарних докумената,
– одржавање централних каталога,
– *on-line* претраживање,
– међубиблиотечка позајмица,
– координација набавке примарних докумената,
– издавање секундарних публикација,
– дисеминација информација,
– инфо-сервис и
– приступ иностраним информационим изворима.

Прва верзија система БИСИС развијана је са циљем да се реализују наведени задаци БМ СНТИС.

1.2. Верзије софтверског система БИСИС

У поглављу 8 монографије [2] дефинисани су следећи појмови: *документ*, *запис* и *библиографска грађа*. Под документом се подразумева изворна библиографска јединица (књига, часопис, итд.). Запис представља скуп података о документу, а скуп записа представља библиографску грађу. У истој монографији дефинисане су следеће функције Библиотечког информационог система:

– дефинисање стандарда за обраду и претраживање,

– формирање библиографске грађе,
– библиотечко извештавање и документовање,
– корисничко претраживање и
– коришћење библиотечке грађе.

Дефинисање стандарда за обраду и претраживање омогућава опис структуре појединих стандарда преко којих ће се обављати основне функције БИСИС-а и њихово представљање у бази података. Посебно се дефинишу стандарди за одржавање библиографске грађе и структурирање записа, радно окружење библиотекара и формати каталога и извештаја. Као основа за размену података и структурирање Базе библиографске грађе у систему БИСИС, усвојен је UNIMARC формат [3]. Дефинисање стандарда за одржавање библиографске грађе треба да омогући опис структуре UNIMARC формата и његово представљање у бази података, како би се обезбедио механизам за обављање осталих функција система у складу са усвојеним стандардом. Дефинисање радног окружења библиотекара треба да обезбеди евидентирање библиотекара и да омогући опис различитих типова докумената и њихово представљање у бази података. Дефинисање формата каталога и извештаја треба да омогући апстрактни опис структуре произвољног извештаја и каталога о документима који се формирају из базе података БИСИС-а, и представљање те структуре у бази података. На основу тога би се вршило структурирање резултата претраживања, библиотекарских извештаја и каталога докумената, као и одговора на корисничке упите, који су резултат претраживања базе података на основу библиотекарских захтева и упита корисника.

Формирање библиографске грађе подразумева обраду докумената и структурирање записа према усвојеном стандарду, преузимајући параметре конкретне обраде на основу унапред дефинисаних модела обраде докумената.

Библиотечко извештавање и документовање омогућава формирање специфичних библиотекарских извештаја и каталога о документима. Формирање извештаја и каталога се врши на основу Базе библиографске грађе и Базе стандарда за обраду библиографских докумената.

Корисничко претраживање обезбеђује обраду упита корисника и структурирање одговора, односно резултата претраживања, на основу садржаја Базе библиографске грађе. Структурирање резултата претраживања врши се на основу унапред дефинисаних структура, садржаја и формата приказа који су дефинисани у Бази стандарда за обраду библиографских докумената.

Коришћење библиотечке грађе омогућава евидентирање корисника библиотеке, као и праћење коришћења фонда библиотеке, резервисање и заду-

живање докумената. Због тога је, поред базе података о библиографској грађи, потребно обезбедити и базу података о корисницима библиотеке.

1.2.1. БИСИС верзија 1

Софтверски систем БИСИС вер. 1 реализован је у програмском језику C, а као систем за управљање базом података коришћена је db_VISTA. За обраду библиографске грађе усвојен је UNIMARC формат. У овој верзији дефинисане су следеће базе података: База UNIMARC формата, База радног окружења библиотекара, База библиографске грађе и База коришћења библиотечке грађе. Овако формиране базе података су се уз потребне измене користиле и у наредним верзијама система.

У овој верзији система подржане су следеће функције: Дефинисање стандарда за одржавање библиографске грађе и структурирање записа, Дефинисање радног окружења библиотекара, Формирање библиографске грађе, Библиотекарско извештавање и документовање и Корисничко претраживање.

Спецификација информационих захтева ове верзије урађена је по методологији описаној у 4. поглављу монографије [2], која се базира на Структурној систем анализи и Проширеном моделу објекти-везе. Та методологија примењивана је уз коришћење *CASE* алата *ARTIST* који је развијен на Факултету организационих наука у Београду. У спецификацији информационих захтева учествовали су представници: Библиотеке Матице српске, Нови Сад; Универзитетске библиотеке “Светозар Марковић”, Београд; и Народне библиотеке Србије, Београд.

1.2.2. БИСИС верзија 2 и верзија 3

Софтверски систем БИСИС вер. 2 реализован је у *Oracle* окружењу и програмском језику *Java*. Од *Oracle*-ових производа коришћен је *Oracle Server* верзија 8.0.3 или новија и текст сервер *ConText Cartridge* верзија 2.3.6 или новија, која подржава рад са текстом у *Unicode*-у. Систем је предвиђен да ради у Интернет, односно Интранет окружењу. Кориснички интерфејс је имплементиран као *Java* аплет. Овом верзијом су подржане све функције из претходне верзије система.

Основна разлика између верзија 2 и 3 је текст сервер за индексирање и претраживање библиографских записа. У верзији 3 развијен је сопствени текст сервер за индексирање и претраживање библиографских записа у UNIMARC формату. Главне карактеристике овог сервера су: специјализованост која резултује бољим перформансама, трослојна архитектура, коришће-

ње *Java* платформе и независност од коришћеног релационог система за управљање базом података. Подршка за *Unicode* стандард доследно је спроведена у целокупном систему БИСИС верзија 3. Ова верзија, поред функција које подржава прва верзија, подржава и функцију Коришћење библиотечке грађе.

Моделирање система извршено је коришћењем обједињеног језика моделирања (*Unified Modeling Language* – UML). Имплементација система реализована је као Интернет апликација у *Java* окружењу. Детаљно упутство за коришћење система БИСИС вер. 3 дато је у књизи [4].

1.3. Дигитална библиотека Универзитета у Новом Саду

У раду [5] приказана је реализација софтверског система *Дигитална библиотека дипломских радова, магистарских теза и докторских дисертација***.** Систем је развијан са циљем да буде отворена архива која ће олакшати приступ резултатима истраживања, у виду електронских издања, студентима, научним радницима и свим заинтересованим. При томе су прихваћене смернице које је одредила иницијатива мреже дигиталних библиотека теза и дисертација (*Networked Digital Library of Theses and Dissertations*, NDLTD) [6].

Пројекат NDLTD започет је на универзитету Virginia Tech, САД. Главни циљ NDLTD пројекта је једноставна и ефикасна централизована каталогизација докторских и магистарских теза одбрањених у оквиру институција чланова. Дигитална библиотека подразумева руковање дигиталним и/или дигитализованим документима. Информациони систем дигиталне библиотеке рукује како описима докумената, тако и самим документима у дигиталном облику.

NDLTD систем се састоји од чворова у мрежи који представљају аутономна складишта докумената. Оригинални, комплетни документи се чувају на одговарајућем чвору у неком од погодних формата дигиталних докумената (најчешће PDF). Чворови мреже омогућавају проналажење докумената које сами складиште. Процес проналажења докумената користи метаподатке о документима који чине саставни део описа докумената. У формирању метаподатака за документ учествују аутор документа и надлежни библиотекари. Приликом прегледа резултата проналажења, могуће је преузети и оригинални облик траженог документа. NDLTD мрежа не намеће посебне функционалне захтеве у проналажењу докумената имплементираном у оквиру појединих чворова.

Централни систем за проналажење докумената [7] постоји као посебан чвор мреже који омогућава проналажење докумената складиштених на свим осталим чворовима. Проналажење користи метаподатке формиране на оригиналним чворовима. Централни систем не складишти оригиналне документе, већ само URL адресе до оригиналног чвора на коме су складиштени.

БИСИС је, са својим текст сервером, искоришћен као језгро система. Део за претраживање овог система поседује све карактеристике претраге које су својствене БИСИС-у. Равноправни третман ћириличног и латиничног текста омогућава истовремено проналажење ћириличних и латиничних докумената. Вишејезичност, као једна од основних карактеристика система, потпуно је подржана у текст серверу тако да је омогућено истовремено проналажење докумената писаних на различитим језицима. Детаљи руковања вишејезичним текстом у текст серверу БИСИС-а приказани су у [8].

1.3.1. Успостављање Сервиса

Софтверски систем Мрежне дигиталне библиотеке Универзитета у Новом Саду пуштен је у рад 15. маја 2003. године. За комуникациону инфраструктуру искоришћена је Академска рачунарска мрежа. На овој мрежи инсталирана је рачунарско-комуникациона и софтверска подршка система Мрежне дигиталне библиотеке. Пошто је Академска рачунарска мрежа укључена на Интернет, овај систем је доступан путем Интернета. Све функције система су веб оријентисане, тако да је унос података и претраживање информација доступно са сваког клијента који је на адекватан начин укључен на Академску рачунарску мрежу, односно на Интернет. Серверу система додељена је Интернет адреса http://diglib.ns.ac.yu.

Оперативни систем сервера је *Linux* а систем за управљање релационим базама података је SAP DB. Наведени системски софтвер је јавно доступан. Клијенти система могу да користе било који оперативни систем који омогућава приступ Интернету односно вебу.

Имплементација система реализована је у програмском језику Java и J2EE технологији. За J2EE апликативни сервер изабран је јавно доступни систем JBoss. За моделирање и пројектовање апликације коришћени су доступни CASE алати.

1.3.2. Преузимање записа

NDLTD мрежа базирана је на иницијативи отворених архива (*Open Archive Initiative – OAI, ОА иницијатива*) [9]. Ова иницијатива осмишљена је

са циљем развоја и промовисања стандарда за који би требали да омогуће ефикасну размену садржаја различитих архива. За размену тих садржаја дефинисан је Протокол за прикупљање метаподатака (*Protocol for Metadata Harvesting* – PMH) [10].

У раду [11] описана је имплементација PMH протокола, у оном његовом делу који дефинише публиковање података у *Dublin Core* формату [12], који је дефинисан под окриљем OA иницијативе. Дат је и кратак опис смерница за имплементацију неколико других формата метаподатка (*MARC, MARC21).* Обрађена је проблематика минималних техничких захтева које мора имплементирати сваки систем који претендује да се мрежи OAI институција прикључи у својству система који податке из своје архиве публикује за остале чланове OA иницијативе (тј. да се појављује у улози *data provider*-а), као и смерницама за имплементацију протокола у системима који у мрежи OAI институција имају улогу клијентских система.

У протоколу је извршена потпуна спецификација формата XML докумената који се размењују, као и дефиниција свих термина везаних уз сам протокол и његову имплементацу у оквиру OA иницијативе. OAI-PMH протокол је базиранан на HTTP протоколу, тј. функционише по принципу захтев – одговор. Захтев шаље клијент који жели прикупити податке, док одговор шаље серверска апликација на страни архиве која се јавља у својству понуђача података. Захтеви се шаљу у виду стандардних параметара HTTP захтева, док је одговор у форми XML документа.

OAI-PMH подржава размену података у различитим форматима. Захтева се да сваки архив који учествује у OA иницијативи подржи минимално *Dublin Core* формат. У оквиру протокола се не праве претпоставке о томе како неки формат метаподатака треба да изгледа, већ се само дефинише елемент XML документа у оквиру кога се метаподаци уписују у складу са траженим (XML) форматом метаподатака.

Дигитална библиотека Универзитета у Новом Саду развијана је као интегрални део библиотечког софтверског система BISIS. На тај начин омогућено је преузимање метаподатака у локалне базе записа. У оквиру истог протокола могуће је моделирати и друге формате (нпр. UNIMARC, MARC, MARC21). У том случају могуће је реализовати и преузимање комплетне библиографске записе.

1.3.3. Виртуелни централни каталог библиографских записа

Узајамна каталогизација представља могућност размене података између две библиотеке на нивоу библиографских записа. Циљ узајамне каталогизације је да библиотекари могу да преузимају записе из других библиотека, тј. централног каталога библиографских записа. Централни каталог се може имплементирати на два начина: обједињеним сервером записа и виртуелним централним каталогом.

Алтернативни приступ у дизајну централног каталога представља концепт виртуелног централног каталога. То значи да централни каталог не поседује обједињену базу библиографских записа која представља скуп свих локалних база записа, већ је у стању да све упите делегира свим локалним библиотечким системима, да сакупи резултате њиховог рада и да их прикаже као јединствен резултат рада. Овај приступ не захтева постојање обједињене базе записа, већ се заснива на кооперативном раду локалних библиотечких система. Концепт виртуелног централног каталога се може реализовати у софтверској агентској технологији.

У раду [13] предложена је имплементација централног каталога записа, која не захтева обједињену базу библиографске грађе, већ се претраживање централног каталога своди на скупљање резултата претрага локалних база. Имплементација агентском технологијом решава проблем дистрибуираног извршења претраге и осталих библиотечких задатака на ефикасан начин зато што постојећи концепти у агентској технологији пружају дистрибуирано извршење, миграцију кода, размену порука и проналажење и ангажовање агената. Имплементација је реализована помоћу *Проширивог агентског окружења базираног на Java технологији* (*Extensible Java-based Agent Framework* – XJAF), које је детаљно описано у дисертацији [14]. Ово агентско окружење је базирано на технологији дистрибуираних компоненти и подржава FIPA (*Foundation for Intelligent Physical Agents*) спецификацију. Имплементација овог окружења је базирана на J2EE (*Java 2 platform – Enterprise Edition*) технологији.

Предложено решење омогућује формирање виртуелне мреже библиотека, где би мрежа агентских окружења повезаних са библиотечким серверима дефинисала топологију виртуелне библиотечке мреже. Такође, овај систем није везан искључиво за један тип библиотечког сервера, већ омогућује формирање мреже хетерогених библиотечких сервера на описан начин.

1.4. XML технологије и библиотечки системи

Примена XML технологија у развоју библиотечког софтверског система БИСИС била је предмет истраживања четири магистарске тезе и једне докторске дисертације. Циљ истраживања у тези [15] био је да се изврши моделирање UNIMARC формата у XML технологији, и то како самог формата, тако и библиографских записа по UNIMARC формату, затим да се имплементира конверзија библиографских записа по UNIMARC формату у XML библиографске записе, као и валидација добијених XML записа. Доказано је да се помоћу XML Schema језика могу моделирати сви концепти UNIMARC формата. Такође је доказано и да се сви концепти библиографског записа по UNIMARC формату могу моделирати помоћу XML Schema језика. Добијени резултати у овој тези послужили су за дефинисање нових тема за истраживање.

Циљ истраживања у магистарској тези [16] био је да се моделира и имплементира XML едитор за UNIMARC формат. За остварење овог циља коришћена је XML шема UNIMARC формата предложена у [15]. Поред тога, детаљно су проучене могућности тренутно доступног софтверског пакета Multi Schema Validator за репрезентацију XML Schema шеме објектним моделом у програмском језику Java. Постојећа верзија овог софтверског пакета нуди објектни модел шеме који није прилагођен за интерпретирање из апликације. Зато је у овој тези описано моделирање и имплементација специјализованог објектног модела шеме као и библиотека класа помоћу којих се тај модел интерпретира и користи. На основу тога имплементиран је XML едитор за опис UNIMARC формата у Java окружењу. Резултат употребе (едитовање, валидација и меморисање) овог едитора је XML документ који садржи све потребне описе структурних концепата стандарда. У имплементацију едитора је укључен посебно реализован метод интернационализације заснован такође на XML ресурсним датотекама.

У магистарској тези [17] извршено је моделирање библиотечких каталошких листића у облику XML докумената на основу XML библиографских записа. Концепти разних типова каталошких листића описани су XML шема језиком, као и садржај конкретних каталошких листића. Резултат издвајања садржаја каталошког листића из записа је такође дат у облику XML документа. Визуализација овог документа је извршена у програмском језику Java, а резултат визуализације је HTML документ. Добијени резултати су основа за укључивање описаног система за генерисање библиотечких каталошких листића у оквиру библиотечких софтверских система.

Главни научни резултати добијени у магистарској тези [18] су:

- дефинисање и систематизација контрола за утврђивање квалитета библиографских записа,
- моделирање концепата контрола записа помоћу XML шема језика,
- XSLT спецификација за опис контрола које се не могу описати XML шема језиком и
- имплементација система за контролу квалитета XML библиографских записа у Java окружењу.

Добијени резултати у оквиру ове магистарске тезе омогућују даљи развој библиотечких софтверских система у следећим сегментима:

- имплементација XML едитора за библиографске записе који не би захтевао познавање XML-а од стране корисника,
- развој веб сервиса за контролу квалитета XML библиографских записа,
- дефиниција и имплементација интерпретера за формални опис контрола квалитета библиографских записа или
- развој експертног система за моделирање система за контролу квалитета XML библиографских записа на основу XML шеме одговарајућег библиографског формата, уз употребу интерпретера за формални опис контрола.

Циљ истраживања у докторској дисертацији [19] био је развој модела проширивог система за проналажење структурираних мултимедијалних докумената (*extensible multimedia information retrieval system*, XMIRS) који омогућава интеграцију различитих постојећих решења за проналажење мултимедијалних објеката и употребу различитих модела проналажења докумената. На овај начин, систем за проналажење докумената не намеће свој модел проналажења нити свој модел репрезентације садржаја њихових мултимедијалних елемената. Верификација предложеног модела извршена је на систему Мрежне дигиталне библиотеке докторских и магистарских радова (NDLTD). Верификација је спроводена помоћу развијеног XMIRS прототипа конфигурисаног за конкретну примену, што обухвата следеће активности:

- моделирање докумената дигиталне библиотеке,
- дефинисање корисничких потреба везаних за могућности процеса проналажења докумената,
- приказ употребљених постојећих софтверских модула који обезбеђују поједине елементе функционалности претраживања,
- приказ конфигурације система која повезује језгро система и употребљене модуле у функционалну целину и
- илустрацију примера коришћења овако формираног система.

Приказана структура докумената дигиталне библиотеке обухвата четири типа медија (структурирани текст, неструктурирани текст, битмапиране слике и видео записе).

1.5. Закључак

Библиотечки софтверски систем БИСИС развија се од 1992. године и до данас је израђено више верзија. Прва верзија система почела је да се користи у пет факултетских библиотека 1996. године, док је сада у експлоатацији трећа верзија система. Део развоја система БИСИС реализује се и у оквиру ТЕМПУС пројекта под називом *Web Based Interuniversity Library Network* (JEP 16114).

У току 2003. године на Универзитету у Новом Саду развијен је софтверски систем *Дигитална библиотека дипломских радова, магистарских теза и докторских дисертација*, која је званично прикључена светској мрежи дигиталних библиотека (NDLTD). На Универзитету у Новом Саду су и даље у току истраживања на развоју овог система. Актуелне теме су: развој система за дигиталне библиотеке; развој библиотечког система базираног на XML технологији; претраживање мултимедијалних библиографских докумената; примена агентске технологије у претраживању библиографских записа.

Референце

[1] Пројекат *План развоја библиотечке мреже Србије*. Руководилац Д. Сурла, Министарство за науку и технолошки развој Србије, Београд, 1992.

[2] Монографија *Формирање и претраживање база података у систему научних и технолошких информација Србије*. Редактор Б. Лазаревић, Министарство за науку и технолошки развој Србије, Београд, 1996.

[3] *UNIMARC Manual: Bibliographic Format/ International Federation of Library Associations and Institutions*. IFLA Universal Bibliographic Control and International MARC Programme, New Providence, London 1994.

[4] Д. Сурла, З. Коњовић и др. *Упутство за коришћење библиотечког софтверског система БИСИС вер. 3*. Група за информационе технологије, Нови Сад, 2003.

[5] Д. Сурла, З. Коњовић и др. *Приказ реализације мрежне дигиталне библиотеке докторских, магистарских и дипломских радова***.** ИНФОТЕКА, Београд, бр. 1-2/2004. стр. 75-86.

[6] *Networked Digital Library of Theses and Dissertations*. Viginia Tech, USA. http://www.ndltd.org

[7] *NDLTD Union Catalog*. VTLS Inc., USA. http://www.vtls.com/ndltd

[8] Б. Милосављевић. *Текст сервер UNIMARC записа*. Магистарска теза, Факултет техничких наука, Нови Сад 1999. http://diglib.ns.ac.yu/ndltd/docs/set1/ndltd6/textsrv.pdf.

[9] *The Open Archives Initiative*. http://www.openarchives.org/

[10] *The Open Archives Initiative Protocol for Metadata Harvesting.* http://www.openarchives.org/OAI/openarchivesprotocol.html

[11] М. Зарић, Д. Сурла, *Имплементација протокола за прикупљање метаподатака у мрежи дигиталних библиотека теза и дисертација*. ИНФОТЕКА, Београд, бр. 1-2/2004. стр. 99-112.

[12] *Dublin Core format specification.* http://www.openarchives.org/OAI/2.0/oai_dc

[13] М. Видаковић, З. Коњовић. *Предлог решење централног каталога за локалне базе библиотечких записа*. ИНФОТЕКА, Београд, бр. 1-2/2004. стр. 113-119.

[14] M. Vidaković. *Proširivo agentsko okruženje bazirano na Java tehnologiji*. Doktorska disertacija, Fakultet tehničkih nauka, Novi Sad, 2003. http://diglib.ns.ac.yu/ndltd/docs/set2/ndltd344/MinjaDoktorat.pdf

[15] М. Зеремски**.** *Моделирање UNIMARC формата у XML технологији.* Магистарска теза, Природно-математички факултет, Департман за математику и информатику, Нови Сад, 2002. http://diglib.ns.ac.yu/ndltd/docs/set1/ndltd64/ZeremskiMMagistarskaTeza.pdf

[16] В. Мијић. *XML едитор за опис UNIMARC формата.* Магистарска теза, Природно-математички факултет, Департман за математику и информатику, Нови Сад, 2003. http://diglib.ns.ac.yu/ndltd/docs/set1/ndltd133/MijicVMagistarskaTeza.pdf

[17] Ј. Видаковић. *Моделирање и имплементација библиотечких каталошких листића у XML технологији.* Магистарска теза, Природно-математички факултет, Департман за математику и информатику, Нови Сад, 2004. http://diglib.ns.ac.yu/ndltd/docs/set2/ndltd335/JovanaMagistarski.pdf

[18] Г. Будимир. *Контрола квалитета библиографских записа*. Магистарска теза, Природно-математички факултет, Департман за математику и информатику, Нови Сад, 2004.

[19] B. Milosavljević. *Proširivi sistem za pronalaženje multimedijalnih dokumenata*. Doktorska disertacija, Fakultet tehničkih nauka, 2003. http://diglib.ns.ac.yu/ndltd/docs/set1/ndltd143/xmirs.pdf

Поглавље 2

YUMARC формат

Б. Пуповац, Т. Зубић

Библиографски формат MARC (*Machine Readable Cataloguing*) [1], као формат за бележење и размену библиографских и њима сродних података у машински читљивом облику, представљао је основу за дефинисање других библиографских MARC формата намењених за потребе различитих држава и библиотечких система. Тако је за потребе европских држава развијен UNIMARC (*Universal Machine Readable Cataloguing*) формат [2]. UNIMARC формат представља универзални формат каталогизације библиографске грађе за бележење библиографских података у машински читљивом облику, и омогућава размену података између националних и међународних библиографских установа.

На основу UNIMARC формата развијени су и неки национални библиографски формати. Формат YUMARC у потпуности одражава основну структуру и обим података UNIMARC формата, а допуњен је за националну употребу и прилагођен софтверском систему БИСИС. У раду [3] дат је преглед развоја библиографских формата.

Формат YUMARC дефинише структуру за библиографски опис библиотечких јединица. Библиографски опис једне библиотечке јединице формиран по YUMARC формату називамо *YUMARC запис* или *библиографски запис*. Скуп библиографских записа чини *библиографску грађу*.

Према YUMARC формату, запис се састоји од *идентификатора записа* (ознаке слога) и низа *поља*. Поље се састоји од низа *потпоља*, а придружена су му и два *индикатора*. Садржаји потпоља представљају податке о публикацији која се каталогизира. Значење података јединствено је идентификовано пољем, придруженим индикаторима и потпољем чији је садржај.

Поља су дефинисана ознаком која се састоји од три цифре и груписана су у десет блокова (прва цифра ознаке поља означава припадност блоку, на пример, поље 101 припада блоку 1):

0 - Блок за идентификацију,
1 - Блок кодираних информација,
2 - Блок главног описа,
3 - Блок напомена,
4 - Блок за повезивање библиографских записа,
5 - Блок сродних наслова,
6 - Блок садржајне анализе,
7 - Блок података о одговорности,
8 - Блок за међународну употребу,
9 - Блок за националну употребу.

Свим пољима придружена су два индикатора који ближе одређују значење података у пољу, однос између одређеног поља и осталих поља у запису или појаву података из поља у извештајима. Индикатори не морају имати дефинисану вредност за свако поље.

Подаци у пољима подељени су у више потпоља. Потпоља су, у оквиру поља, дефинисана ознаком која је један алфанумерички знак.

Податак у потпољу може бити текст или шифра из неког од шифарника YUMARC формата, што је одређено *типом контроле* потпоља. Шифрирани подаци могу важити у оквиру једног или више потпоља. Они који важе само у оквиру једног потпоља називају се *интерне шифре*. Ако се могу користити у више потпоља, тада представљају *екстерне шифре*. Екстерне шифре резврстане су по типовима, као на пример, шифарник земаља, језика, кодова ауторства, итд. Поред тога, у оквиру блока 4 и блока 9 јављају се потпоља која имају специфичну структуру.

Блок 4 је блок за повезивање библиографских записа (нпр., користи се при повезивању публикација са њиховим прилозима или прикљученим радовима који се обрађују као посебне библиографске јединице). Веза између записа остварује се уносом текстуалних или нумеричких података у одговарајућа поља, односно потпоља, блока 4. Ти подаци могу бити идентификациони број записа са којим се врши повезивање или нова поља којима се описује документ са којим вршимо повезивање. Поља која се могу појавити као садржај потпоља поља из блока 4 називају се *секундарна поља*.

Блок 9 дефинише се у националном оквиру. Намењен је за унос података који су од локалног значаја за библиотечку мрежу или појединy библиотеку. У овом блоку постоје поља чија потпоља имају податке структуриране у још један ниво структуре, а који називамо *потпотпоља*. Ако се као вредност

потпотпоља уносе шифрирани подаци, значење шифара дато је *шифарником потпотпоља*.

Следећи пример илуструје обраду једне монографске публикације обрађене у складу са YUMARC форматом. Нека је дата публикација под називом “Структуре података и организација датотека” чији је аутор Павле Могин. Књигу је издала издавачка кућа “Наука” из Београда 1991. године у 500 примерака, а штампала штампарија Бакар из Бора. Књига има 497 страна у осам поглавља, садржи графичке приказе, а има и библиографију и регистар. ISBN број ове публикације је 86-7621-055-1. У инвентарној књизи налази се под бројем 19067, а њену локацију одређује сигнатура M-19067. Аутор записа у бази је Гога, а запис је креиран у фебруару 1997. године. На слици 2.1 приказан је YUMARC запис за наведену публикацију.

001	[a]p[b]a[c]m[d]0[7]ba
010	[a]86-7621-055-1
100	[b]d[c]1991[h]scr
1010	[a]scr
102	[a]yug[b]sr
105	[a]d[b]j
2000	[a]Strukture podataka i organizacija datoteka[f]Pavle Mogin
210	[a]Beograd[c]Nauka[d]1991[e]Bor[g]Bakar
215	[a]VIII, 497 str.[c]graf. prikazi[d]24 cm
3001	[a]Tiraž 500
320	[a]Bibliografija: str. 288-290
320	[a]Registar.
606	[a]Nauka o računarima[x]Organizacija podataka[x]Struktura podataka
675	[a]519.72
700 1	[a]Mogin[b]Pavle
992	[b]crgoga9702
99602	[d]<l>M<n>19067[f]000019067[v]a

Слика 2.1. YUMARC запис

Запис је приказан у више редова, по један ред за свако поље записа. Прва три знака у реду представљају идентификатор поља, следећа два представљају вредности првог и другог индикатора (ако им вредност није дефинисана, остављено је празно место). Ознаке потпоља уписане су између знакова [и], а иза њих наводи се садржај потпоља. Ознаке потпотпоља уписане су између знакова < и >.

У наредном тексту дат је опис структуре записа приказаног на слици 2.1. Блок 0 је блок за идентификацију. У поље *001* уписују се неопходни

кодирани подаци који описују публикацију. Јединствени ISBN број књиге уноси се у поље *010*, потпоље *a*. Поља из блока 1 предвиђена су за кодиране информације о типу публикације, језику каталогизације, језику публикације и слично. Главни стварни наслов књиге смештен је у поље *200*, потпоље *a*, а аутор у потпоље *f*. Подаци о аутору уписују се у поље *700*, при чему се другим индикатором одређује да ли је у потпољу *a* име или презиме аутора. У потпоља поља *210* уписују се подаци о издавању, док се материјални опис публикације уноси у потпоља поља *215*. Напомене, односно подаци о тиражу, регистрима, библиографијама, уносе се у потпоља поља из блока 3. Блок 6 резервисан је за унос података садржајне анализе публикације. У поље *992* потпоље *b* уносе се подаци о креатору и датум креирања записа. У потпоља поља *996* уносе се подаци о сигнатури и инвентарском броју публикације. Потпоље *d* садржи потпотпоља за структурирање података из сигнатуре. Тако се у потпотпоље *l* уноси податак о локацији на којој је смештена публикација, док се у потпотпоље *n* уноси број који одређује њено место на наведеној локацији.

Усвајањем YUMARC формата ствара се могућност успостављања узајамне каталогизације у оквиру конкретне библиотечке мреже. Једном обрађену публикацију у електронском каталогу могу преузети сви учесници библиотечке мреже, што омогућава рационализацију пословања у библиотечко-информационој делатности.

2.1. Дијаграм класа YUMARC формата

Дијаграм класа YUMARC формата, моделиран коришћењем UML (*Unified Modeling Language*) нотације [4], приказан на слици 2.2, садржи следеће класе: *Polje*, *Potpolja*, *Potpotpolja*, *Prvi_indikator*, *Drugi_indikator*, *Tipekst_sifarnika*, *Eksterni_sifarnik*, *Interni_sifarnik*, *Sifarnik_potpotpolja* и *Tip_kontrole*.

Класа *Polje* садржи атрибут *poljeid* који дефинише јединствену вредност поља. Такође, садржи и атрибуте:

- *po_naziv* – дефинише назив поља,
- *obaveznost* – означава да ли се поље мора обавезно налазити у оквиру записа,
- *ponovljivost* – означава да ли се поље може појавити више пута унутар једног записа, и
- *sekundarnost* – означава да ли се то поље може јавити као садржај неког другог поља.

Осим тога, атрибути *opis_pi*, *opis_di*, *def_pi*, *def_di* дефинишу постојање првог, односно другог индикатора за дато поље, и њихову подразумевану вредност (ако постоје).

Класа *Potpolja* садржи атрибут *potpoljaid* који дефинише јединствену вредност потпоља у оквиру поља. Такође, садржи и атрибуте:

- *pp_naziv* – дефинише назив потпоља,
- *duzina* – дефинише дужину вредности потпоља, изражену у броју знакова, ако је вредност потпоља ограничене дужине,
- *obaveznostpp* – означава да ли се потпоље мора обавезно појавити у оквиру поља,
- *ponovljivostpp* – означава да ли се потпоље може појавити више пута у оквиру једног поља,
- *status* – дефинише да ли се потпоље активно користи у обради записа, и
- *defvrednost* – дефинише подразумевану вредност потпоља.

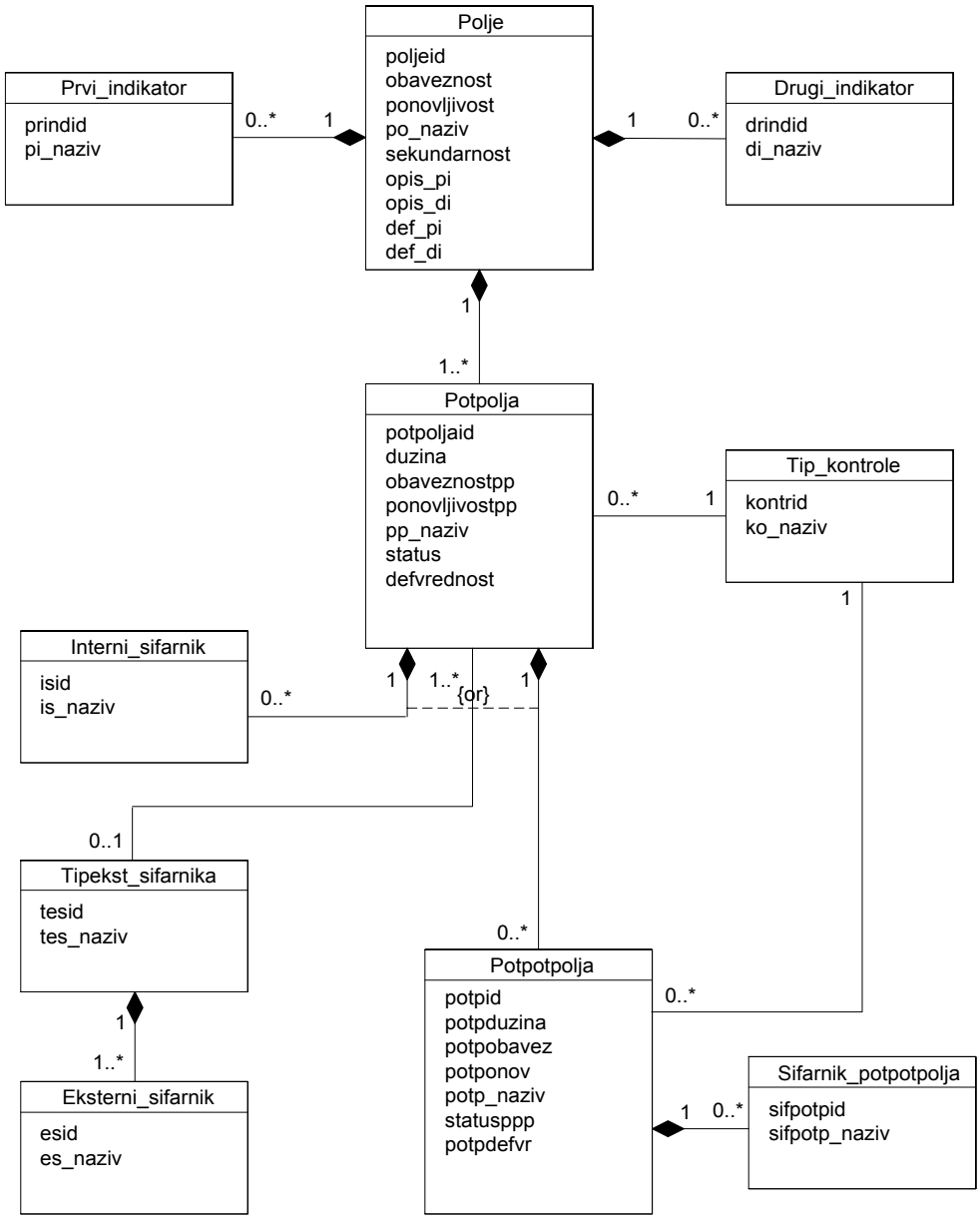
Класа *Potpotpolja* садржи атрибут *potpid* који дефинише јединствену вредност потпотпоља у оквиру потпоља, као и атрибуте *potp_naziv*, *potpduzina*, *potpobavez*, *potponov* и *statusppp* који дефинишу назив и особине потпотпоља. Значења ових атрибута иста су као код класе *Potpolja*, али се односе на потпотпоља. Атрибут *potpdefvr* дефинише подразумевану вредност потпотпоља.

Класа *Prvi_indikator* садржи атрибут *prindid* који дефинише јединствену вредност првог индикатора у оквиру поља и атрибут *pi_naziv* који дефинише значење одређеног индикатора.

Класа *Drugi_indikator* садржи атрибут *drindid* који дефинише јединствену вредност другог индикатора у оквиру поља и атрибут *di_naziv* који дефинише значење одређеног индикатора.

Класа *Tipekst_sifarnika* дефинише типове екстерних шифарника (нпр. шифарник језика, земаља) који могу бити везани за једно или више потпоља. Садржи атрибут *tesid* који дефинише јединствену вредност типа екстерног шифарника и атрибут *tes_naziv* који описује значење типа екстерног шифарника.

Класа *Eksterni_sifarnik* садржи атрибут *esid* који дефинише јединствену вредност екстерне шифре у оквиру типа екстерног шифарника и атрибут *es_naziv* који дефинише значење одређене екстерне шифре.



Слика 2.2. Дијаграм класа YUMARC формата

Класа *Interni_sifarnik* садржи атрибут *isid* који дефинише јединствену вредност интерне шифре у оквиру датог потпоља и атрибут *is_naziv* који дефинише значење интерне шифре.

Класа *Sifarnik_potpotpolja* садржи атрибут *sifpotpid* који дефинише јединствену вредност шифре у оквиру потпотпоља и атрибут *sifpotp_naziv* који дефинише значење шифре.

Класа *Tip_kontrole* представља различите типове контроле вредности потпоља односно потпотпоља (нпр. контрола формата датума). Садржи атрибуте *kontrid* који дефинише јединствену вредност типа контроле и *ko_naziv* који описује значење одређене контроле.

Помоћу CASE алата *Rational Rose* на основу дијаграма класа (слика 2.2) генерисана је шема базе података YUMARC формата. Свакој класи са дијаграма класа одговара тачно једна табела (са истим именом) чији називи колона одговарају називима атрибута класе.

2.2. Апликација за ажурирање YUMARC формата

У оквиру софтверског система БИСИС верзија 3 имплементирана је апликација за ажурирање YUMARC формата [5] у Java окружењу [6, 7].

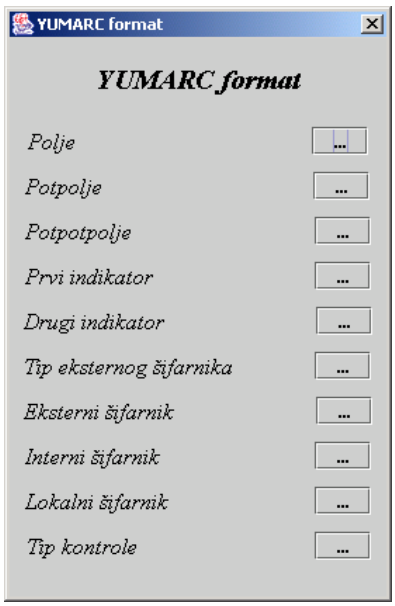
Изглед почетног прозора за ажурирање YUMARC формата приказан је на слици 2.3. Овај прозор омогућава прелаз на конкретан прозор за ажурирање одабраног концепта YUMARC формата (поље, потпоље, потпотпоље...). Сваки од прозора омогућава ажурирање слогова одговарајуће табеле базе података YUMARC формата, као и навигацију кроз постојеће слогове.

2.2.1. Прозор за ажурирање поља

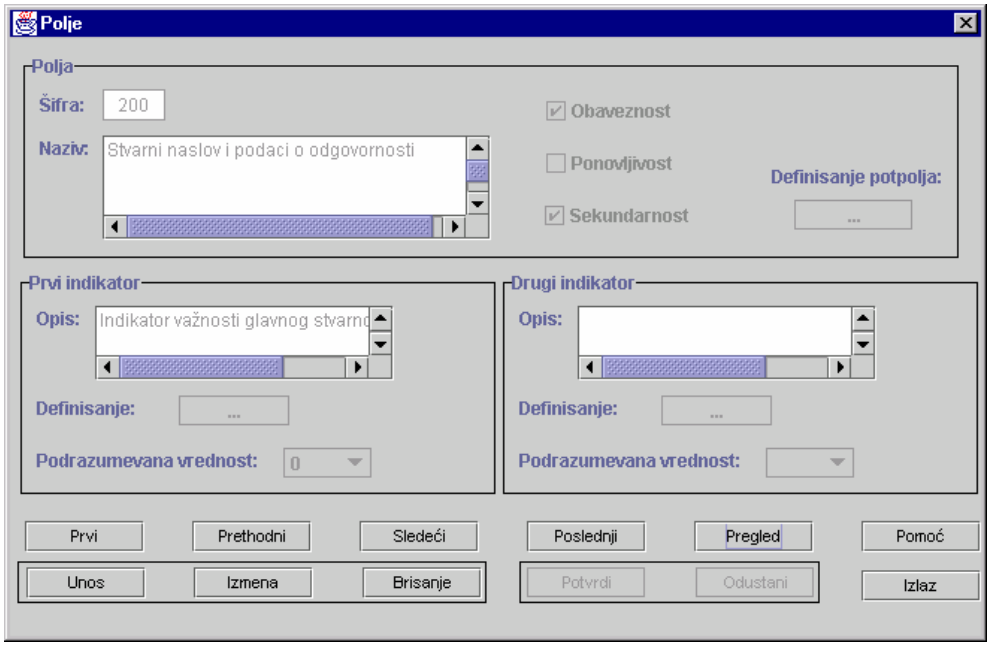
Прозор **Polje**, приказан на слици 2.4, служи за унос, модификацију и брисање слогова из табеле *Polje*, као и за навигацију кроз постојеће слогове те табеле.

У једнолинијски едитор **Šifra** уноси/приказује се шифра поља - 3 цифре. Овај едитор омогућава да се унесе/прикаже садржај колоне *poljeid*.

У вишелинијски едитор **Naziv** уноси/приказује се назив поља максималне дужине 160 знакова. Овај едитор омогућава да се унесе/прикаже садржај колоне *po_naziv*.



Слика 2.3. Почетни прозор



Слика 2.4. Прозор **Polje**

Простори за потврду **Obaveznost**, **Ponovljivost** и **Sekundarnost** иницијално нису означени (односно, поље није обавезно, поновљиво и секундарно). Означавањем ових простора за потврду постиже се да вредност у бази података буде постављена на 1 (односно, поље је обавезно, поновљиво и секундарно). Ови простори за потврду одговарају садржајима колона *obaveznost, ponovljivost*, *sekundarnost* респективно.

Definisanje potpolja је дугме које је активно приликом уноса новог поља или приликом измене постојећег поља. Притиском на ово дугме отвара се прозор за унос потпоља одговарајућег поља. Овај прозор омогућава ажурирање слогова табеле *Potpolja* придружене текућем пољу, као и навигацију кроз постојећа потпоља текућег поља (уколико постоје). Корисник обавезно мора да унесе бар једно потпоље за свако ново поље.

Вишелинијски едитор **Opis** (првог индикатора) омогућава унос описа за први индикатор максималне дужине 240 знакова. Овај едитор омогућава да се унесе/прикаже садржај колоне *opis_pi*. Уколико се опис унесе, захтева се унос вредности првог индикатора.

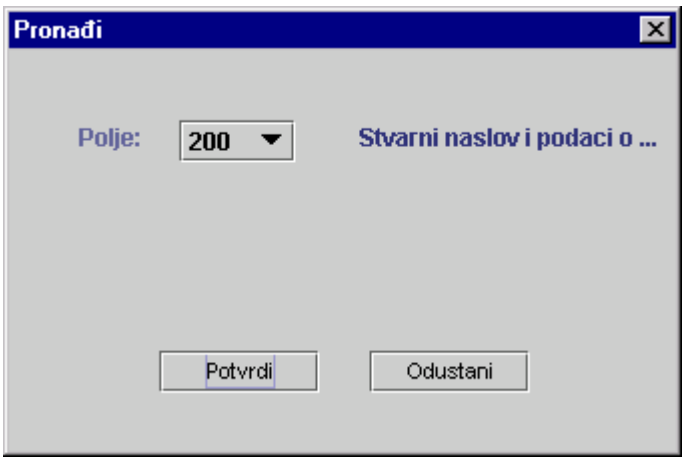
Дефинисање вредности првог индикатора врши се преко новог прозора, који се позива притиском на дугме **Definisanje** у делу који се односи на први индикатор. Овај прозор омогућава ажурирање слогова табеле *Prvi_indikator* придружене текућем пољу, као и навигацију кроз вредности првог индикатора текућег поља (ако постоје).

Из падајуће листе **Podrazumevana vrednost** (првог индикатора) бира се вредност првог индикатора из већ дефинисаног скупа вредности првог индикатора. Одабрана вредност из ове графичке компоненте одговара вредности колоне *def_pi*.

Функционалност дела прозора **Polje** која се односи на други индикатор је иста као што је описано за први индикатор (само се подаци односе на други индикатор).

На слици 2.4 приказано је поље *200*, чији назив је *Stvarni naslov i podaci o odgovornosti*. Поље је обавезно и секундарно. За ово поље опис првог индикатора је *Indikator važnosti glavnog stvarnog naslova*, а подразумевана вредност му је *0*.

Прозор за проналажење жељеног слога концепта поље. Овај прозор, приказан на слици 2.5, позива се притиском на дугме **Pregled**. Корисник бира шифру поља. Притиском на дугме **Potvrdi** проналази се жељени слог (на слици 2.5 поље *200*). Притиском на дугме **Odustani** одустаје се од акције.



Слика 2.5. Прозор за проналажење слога концепта поље

На сличан начин имплементирани су прозори за ажурирање осталих концепата YUMARC формата. Детаљно упутство за коришћење апликације за ажурирање YUMARC формата дато је у књизи [8].

2.3. Закључак

Велики напредак у библиотечком пословању представља дефинисање формата за бележење библиографских података у машински читљивом облику. На тај начин омогућена је размена података између националних и међународних библиографских установа. Библиотечки софтверски систем БИСИС вер. 3, имплементиран у Java окружењу, је систем који подржава библиотечко пословање. Систем је базиран на YUMARC формату, а као саставни део система имплемантирана је апликација која омогућава ажурирање свих концепата YUMARC формата. На тај начин обезбеђена је конфигурабилност софтверског система БИСИС у смислу дефинисања и одржавања YUMARC формата према потребама конкретне библиотеке.

Референце

[1] *MARC Format Overview*. Network Development and MARC Standards Office, Library of Congress, 08/13/2003., http:// www.loc.gov/marc/status.html

[2] *UNIMARC Manual: Bibliographic Format/International Federation of Library Associations and Institutions*. IFLA Universal Bibliographic Control and International MARC Programme, New Providence, London 1994.

[3] Г. Будимир. *MARC записи и XML*. ИНФОТЕКА, фебруар 2004, година. 4, бр. 1, Београд.

[4] G. Booch, J. Rumbaugh, I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley Longman, Inc., 1999.

[5] I. Holo, B. Pupovac, T. Tošić. *The user interface for data maintenance of the UNIMARC format*. Proceedings of the XIII Conference on Applied Mathematics PRIM'98, Igalo, 1998., pp. 43-50

[6] B. Eckel. *Thinking in Java*, 2nd edition. 2000., http://www.EckelObjects.com

[7] *JavaTM 2 Platform, Standard Edition, v 1.3.1 API Specification*. http://java.sun.com/j2se/1.3/docs/api/

[8] Д. Сурла, З. Коњовић, Б. Пуповац, Б. Милосављевић, М. Видаковић, Т. Тошић, Т. Зубић. *Упутство за коришћење бибилотечког софтверског система БИСИС вер. 3*. Група за информационе технологије, Нови Сад, 2003.

Поглавље 3

Моделирање и имплементација радног окружења библиотекара

Б. Пуповац

Да би се обезбедила комплетна функционалност софтверског система БИСИС, подаци о радном окружењу библиотекара налазе се у бази података радног окружења библиотекара [1, 2], која представља саставни део система. Ови подаци користе се у циљу прилагођавања апликације за обраду библиографске грађе специфичним захтевима сваког библиотекара, као и типовима докумената које он обрађује.

Да би библиотекар могао да ради у апликацији за обраду библиографске грађе (описана у наредном поглављу) подаци о њему морају се налазити у бази података радног окружења библиотекара. Приликом стартовања апликације, преузимају се подаци из те базе и подешава се радно окружење за пријављеног библиотекара. Подешавање радног окружења подразумева постављање типа публикације, нивоа обраде, нивоа обавезности, постављање пет префикса претраживања и формата исписа за сваког пријављеног библиотекара понаособ.

База података радног окружења библиотекара, такође, омогућава рад библиотекара са различитим типовима докумената. Монографије, серијске публикације, докторске дисертације, представљају неке од типова докумената. За сваки тип документа могу се дефинисати поља/потпоља која омогућавају да се обраде подаци о таквом документу. У зависности од врсте и обима података који се дефинишу приликом обраде документа, постоји више нивоа обраде за сваки тип публикације. За сваки ниво обраде дефинишу се поља/потпоља за која се уносе вредности на том нивоу. Сваком нивоу обраде дефинисаном за одређени тип публикације, може се придружити више нивоа обавезности. Документ се сматра обрађеним уколико се унесу вредности за поља/потпоља дефинисана на датом нивоу обавезности.

База података радног окружења библиотекара садржи и различите формате исписа, који омогућавају различите приказе извештаја и резултата претраживања. Поред тога, ова база садржи префиксе претраживања помоћу којих се врши формирање упита над базом података библиографске грађе.

3.1. Дијаграм класа радног окружења библиотекара

Дијаграм класа радног окружења библиотекара, моделиран коришћењем UML (*Unified Modeling Language*) нотације [3], приказан на слици 3.1, садржи следеће класе: *Tip_publikacije*, *Def_tipa*, *Nivo_obrade*, *Nivo_obrade_potpolja*, *Nivo_obaveznosti*, *Nivo_obaveznosti_potpolja*, *Bibliotekar*, *Tipbibliotekara*, *Poljebpr*, *Poljebpr_potpolja*, *Formatf*, *Formatf_poljebpr*, *Tip_obrade* и *Poljebpr_tip_obrade*. Класа *Potpolja* је класа дијаграма класа YUMARC формата (описаног у Поглављу 2) и представља везу између ова два дијаграма.

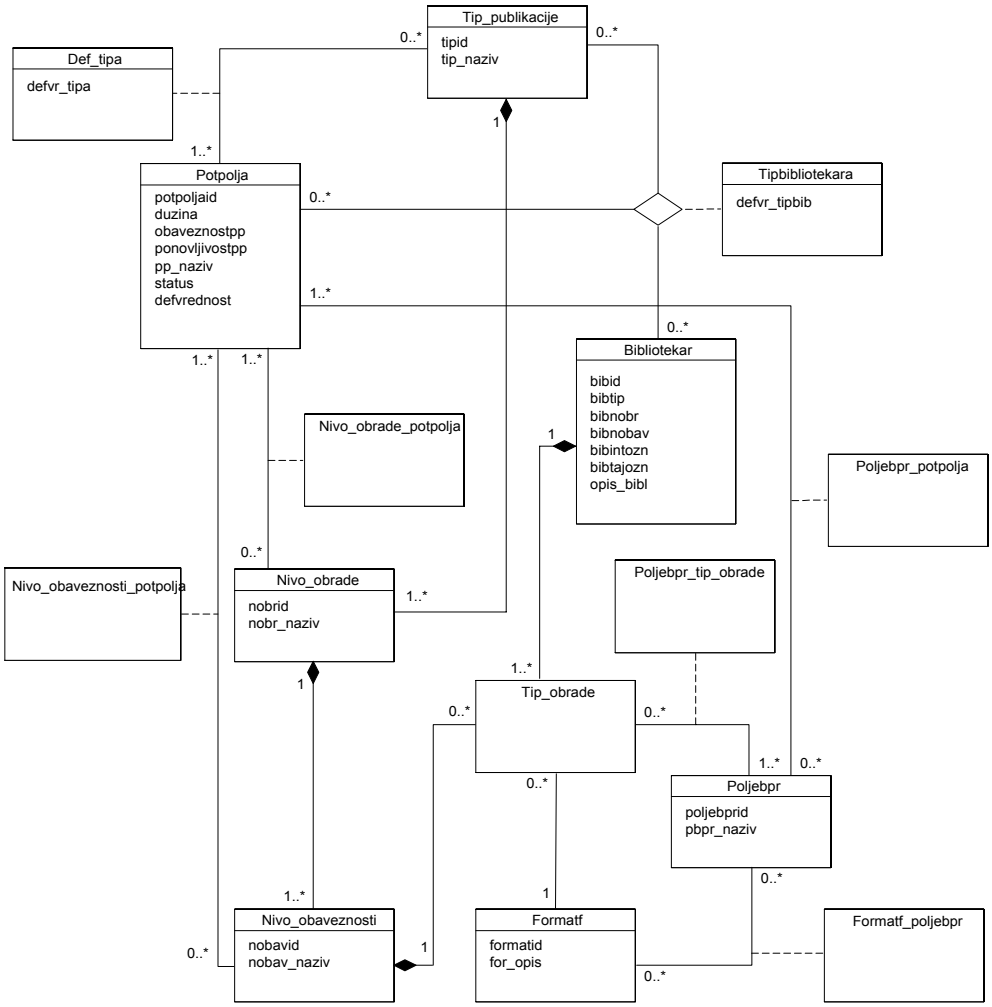
Класа *Tip_publikacije* садржи атрибут *tipid* који дефинише јединствену вредност типа публикације и атрибут *tip_naziv* који дефинише назив одређеног типа публикације.

Класа асоцијације *Def_tipa* дефинише поља/потпоља која се могу обрађивати за одређени тип публикације и опционо, подразумевану вредност таквих потпоља (атрибут *defvr_tipa*).

Класа *Nivo_obrade* садржи атрибут *nobrid* који дефинише јединствену вредност нивоа обраде у оквиру одређеног типа публикације. Такође, ова класа садржи и атрибут *nobr_naziv* којим је дефинисано значење одређеног нивоа обраде. Класа асоцијације *Nivo_obrade_potpolja* повезана са везом између класа *Nivo_obrade* и *Potpolja* дефинише поља/потпоља која се могу обрађивати на одређеном нивоу обраде и која представљају подскуп поља/потпоља дефинисана за надређени тип публикације.

Класа асоцијације *Def_tipa* дефинише поља/потпоља која се могу обрађивати за одређени тип публикације и опционо, подразумевану вредност таквих потпоља (атрибут *defvr_tipa*).

Класа *Nivo_obrade* садржи атрибут *nobrid* који дефинише јединствену вредност нивоа обраде у оквиру одређеног типа публикације. Такође, ова класа садржи и атрибут *nobr_naziv* којим је дефинисано значење одређеног нивоа обраде. Класа асоцијације *Nivo_obrade_potpolja* повезана са везом између класа *Nivo_obrade* и *Potpolja* дефинише поља/потпоља која се могу обрађивати на одређеном нивоу обраде и која представљају подскуп поља/потпоља дефинисаних за одређени тип публикације.



Слика 3.1. Дијаграм класа радног окружења библиотекара

Класа *Bibliotekar* дефинише библиотекаре који раде у систему. Атрибут *bibid* дефинише јединствену шифру библиотекара, док атрибути *bibintozn* и *bibtajozn* дефинишу његову интерну, односно тајну ознаку у систему. Атрибути *bibtip, bibnobr* и *bibnobav* дефинишу подразумевану вредност типа публикације, нивоа обраде и нивоа обавезности који се постављају за библиотекара након пријављивања на систем. Атрибут *opis_bibl* садржи додатне податке о библиотекару.

Класа *Tipbibliotekara* дефинише подразумевану вредност поља/потпоља (атрибут *defvr_tipbib*) за одређеног библиотекара и за одређени тип публикације.

Класа *Poljebpr* дефинише префиксе претраживања базе података библиографске грађе. Атрибут *poljebprid* представља јединствену вредност префикса, а атрибут *pbpr_naziv* значење одређеног префикса претраживања. Класа асоцијације *Poljebpr_potpolja* повезана са везом између класа *Poljebpr* и *Potpolja* дефинише поља/потпоља која се претражују у оквиру одређеног префикса претраживања.

Класа *Formatf* дефинише формате исписа предвиђене у систему. Атрибут *formatid* представља јединствени назив формата, а атрибут *for_opis* опис датог формата исписа. Класа асоцијације *Formatf_poljebpr* повезана са везом између класа *Formatf* и *Poljebpr* дефинише формате исписа који приказују садржаје одређених префикса претраживања након претраживања базе података библиографске грађе.

Класа *Tip_obrade* дефинише радно окружење библиотекара преко веза са класама *Bibliotekar*, *Nivo_obaveznosti*, *Formatf* и *Poljebpr*. Класа асоцијације *Poljebpr_tip_obrade* повезана са везом између класа *Tip_obrade* и *Poljebpr* дефинише пет префикса за одређени тип обраде библиотекара.

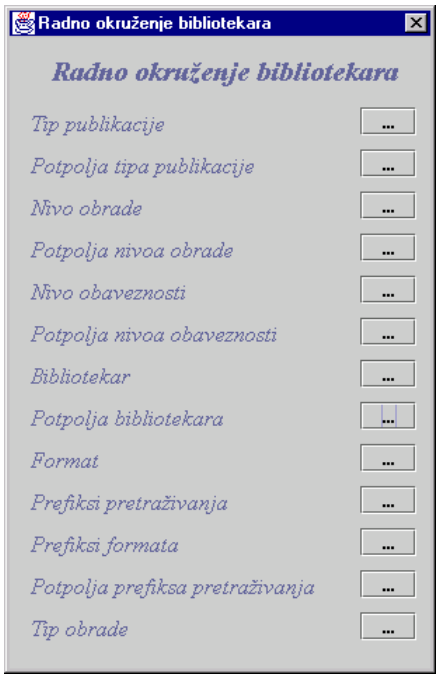
Помоћу CASE алата *Rational Rose* на основу дијаграма класа (слика 3.1) генерисана је шема базе података радног окружења библиотекара. Свакој класи са дијаграма класа одговара тачно једна табела (са истим именом) чији називи колона одговарају називима атрибута класе.

3.2. Апликација за ажурирање радног окружења библиотекара

Саставни део софтверског система БИСИС верзија 3 је и апликација за ажурирање радног окружења библиотекара [4] имплементирана у Java окружењу [5, 6].

Изглед почетног прозора за ажурирање радног окружења библиотекара приказан је на слици 3.2. Овај прозор омогућава прелаз на конкретан прозор за ажурирање сваког од концепата радног окружења библиотекара (тип публикације, потпоља типа публикације, ниво обраде...). Сваки од прозора омогућава ажурирање слогова одговарајуће табеле базе података радног окружења библиотекара, као и навигацију кроз постојеће слогове. У наредном делу текста приказани су описи прозора за ажурирање типа публикације

и прозора за ажурирање потпоља типа публикације као почетног сегмента у дефинисању структуре за хијерархијску обраду публикација.



Слика 3.2. Почетни прозор

3.2.1. Прозор за ажурирање типа публикације

Прозор **Tip publikacije**, приказан на слици 3.3, служи за унос, модификацију и брисање слогова из табеле *Tip_publikacije*, као и за навигацију кроз постојеће слогове те табеле.

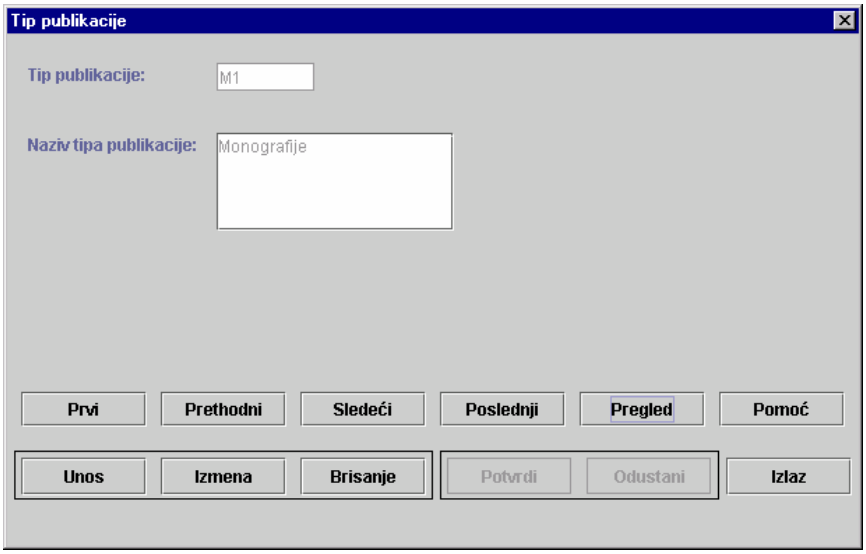
У једнолинијски едитор **Tip publikacije** уноси/приказује се вредност типа публикације – 2 знака. Овај едитор омогућава да се унесе/прикаже садржај колоне *tipid*.

У вишелинијски едитор **Naziv tipa publikacije** уноси/приказује се назив типа публикације максималне дужине 160 знакова. Овај едитор омогућава да се унесе/прикаже садржај колоне *tip_naziv*.

На слици 3.3 приказан је тип публикације *M1*, чији назив је *Monografije*.

Прозор за проналажење жељеног слога концепта тип публикације. Овај прозор, приказан на слици 3.4, позива се притиском на дугме **Pregled**.

Корисник бира вредност типа публикације. Притиском на дугме **Potvrdi** проналази се жељена појава (на слици 3.4 тип публикације *M1*). Притиском на дугме **Odustani** одустаје се од акције.



Слика 3.3. Прозор **Tip publikacije**

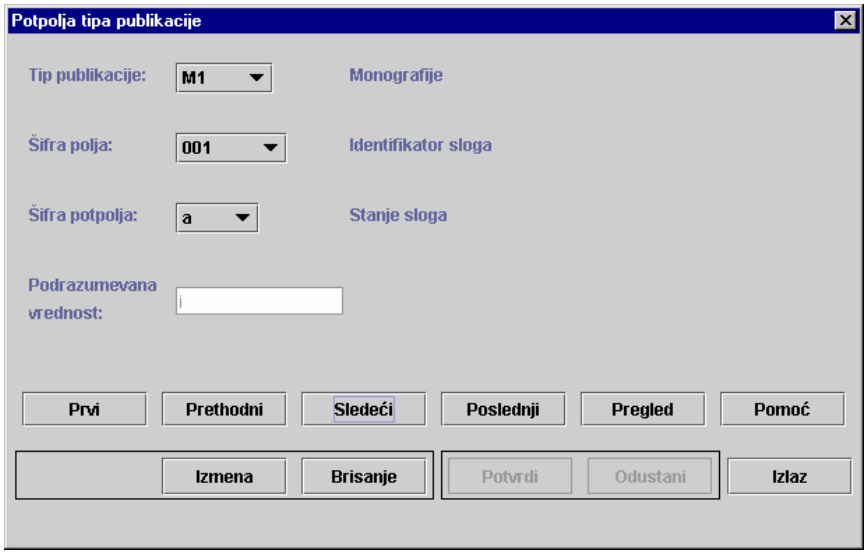
Слика 3.4. Прозор за проналажење слога концепта тип публикације

3.2.2. Прозор за ажурирање потпоља типа публикације

Прозор **Potpolja tipa publikacije**, приказан на слици 3.5, служи за унос, модификацију и брисање слогова из табеле *Def_tipa*, као и за навигацију кроз постојеће слогове те табеле.

Падајућа листа **Tip publikacije** садржи све унете вредности типа публикације. Садржај ове графичке компоненте су све вредности колоне *tipid* табеле *Tip_publikacije*. Одабрана вредност (на слици 3.5 вредност *M1*) ове компоненте одговара садржају колоне табеле *Def_tipa* (у којој се чувају вредности типа публикације), која настаје на основу везе те табеле са везом између табела *Tip_publikacije* и *Potpolja*.

Падајућа листа **Šifra polja** садржи све унете шифре поља. Садржај ове графичке компоненте су све вредности колоне табеле *Potpolja* (у којој се чувају шифре поља) настале на основу везе ове табеле и табеле *Polje* (описана у Поглављу 2). Одабрана вредност (на слици 3.5 вредност *001*) ове компоненте одговара садржају колоне табеле *Def_tipa* (у којој се чувају шифре поља), која настаје на основу везе те табеле са везом између табела *Tip_publikacije* и *Potpolja*.



Слика 3.5. Прозор **Potpolja tipa publikacije**

Падајућа листа **Šifra potpolja** садржи све шифре потпоља које се односе на одабрану шифру поља. Садржај ове графичке компоненте су све вредности колоне *potpoljaid* табеле *Potpolja* које се односе на одабрано поље. Одабрана вредност (на слици 3.5 вредност *a*) ове компоненте одговара садржају колоне табеле *Def_tipa* (у којој се чувају шифре потпоља), која настаје на основу везе те табеле са везом између табела *Tip_publikacije* и *Potpolja*.

У једнолинијски едитор **Podrazumevana vrednost** уноси/приказује се подразумевана вредност потпоља. Овај едитор омогућава да се унесе/прикаже садржај колоне *defvr_tipa*.

На слици 3.5 приказана је подразумевана вредност *i* потпоља *a* поља *001*, који су придружени типу публикације *M1*.

Прозор за проналажење жељеног слога концепта потпоља типа публикације. Овај прозор, приказан на слици 3.6, позива се притиском на дугме **Pregled**. Корисник бира жељену вредност типа публикације. Након избора вредности типа публикације, корисник бира шифру поља и потпоља. Притиском на дугме **Potvrdi** проналази се жељена појава (на слици 3.6 шифра потпоља *a* поља *001* придружени су типу публикације *M1*). Притиском на дугме **Odustani** одустаје се од акције.

Слика 3.6. Прозор за проналажење слога концепта потпоља типа публикације

На сличан начин имплементирани су прозори за ажурирање осталих концепата радног окружења библиотекара. Детаљно упутство за коришћење апликације за ажурирање радног окружења библиотекара дато је у књизи [7].

3.3. Закључак

Радно окружење библиотекара омогућава дефинисање различитих нивоа и начина обраде публикација. Такође, дефинишу се вредности које омогућавају да се библиотекар пријави на систем. Поред тога, за сваког библиотекара, који ради у систему, дефинишу се обавезе и одговорности приликом обраде публикација. Апликација за ажурирање радног окружења библиотекара представља саставни део софтверског система БИСИС вер. 3. Она омогућава ажурирање података који се користе у циљу прилагођавања апликације за обраду библиографске грађе специфичним захтевима сваког

библиотекара, као и типовима докумената које он обрађује. Сем тога, подржани су и специфични захтеви обраде публикација, као и хијерархијска обрада публикација, а све према потребама конкретне библиотеке.

Референце

[1] *Formiranje i pretraživanje baza podataka u sistemu naučnih i tehnoloških informacija Srbije*. Redaktor Branislav Lazarević, Ministarstvo za nauku i tehnologiju Republike Srbije, Beograd 1996.

[2] Б. Пуповац. *Моделирање и имплементација команди библиотечког информационог система*. Магистарска теза, Природно-математички факултет, Институт за математику, Нови Сад, 2000. http://diglib.ns.ac.yu/ndltd/docs/set1/ndltd19/magist.pdf

[3] G. Booch. J. Rumbaugh, I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley Longman, Inc., 1999.

[4] Б. Пуповац. *Радно окружење библиотекара*. ИНФОТЕКА, 2 (2001) 1-2, Београд, стр. 105-109.

[5] B. Eckel. *Thinking in Java*, 2nd edition. 2000. http://www.EckelObjects.com

[6] *JavaTM 2 Platform, Standard Edition, v 1.3.1 API Specification*. http://java.sun.com/j2se/1.3/docs/api/

[7] Д. Сурла, З. Коњовић, Б. Пуповац, Б. Милосављевић, М. Видаковић, Т. Тошић, Т. Зубић. *Упутство за коришћење бибилотечког софтверског система БИСИС вер. 3*. Група за информационе технологије, Нови Сад, 2003.

Поглавље 4

Обрада библиографске грађе

М. Видаковић

Обрада библиографске грађе [1, 2] представља централни део софтверског система БИСИС [3]. Обраду библиографске грађе обезбеђује клијент апликација библиотечког информационог система. Основни процеси ове апликације су:

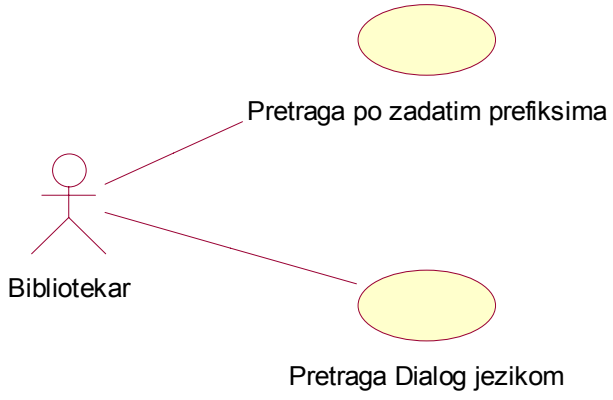
- претрага библиографске грађе,
- креирање нових записа,
- креирање нових записа на основу постојећих,
- кодификација постојећег записа,
- обрада меморијске структуре која репрезентује запис,
- снимање записа и
- брисање постојећих записа.

Процес *Претрага библиографске грађе* врши претраживање базе података библиографских записа. Процеси *Креирање нових записа*, *Креирање нових записа на основу постојећег* и *Модификација постојећих записа* се ослањају на рад са меморијском структуром која репрезентује запис по UNIMARC формату [4]. Процес креирања новог записа подразумева да се прво креира одговарајућа меморијска структура, која се затим обрађује. Процес *Модификација постојећег записа* подразумева креирање одговарајуће меморијске структуре, која се затим попуњава конкретним подацима из записа. Процес *Обрада меморијске структуре која репрезентује запис* врши обраду меморијске структуре додавањем, изменом и брисањем поља, секундарних поља, потпоља и потпотпоља. Снимање записа (процес *Снимање записа*), без обзира да ли је управо креиран или се обрађује постојећи запис, је у ствари превођење меморијске структуре у UNIMARC запис и његово снимање у базу података. Брисање записа (процес *Брисање постојећих записа*) се своди на брисање одговарајућих слогова из базе података. За то је задужен текст сервер библиотечког информационог система [5], преко команде *delete*.

4.1. Обрада записа библиографске грађе

4.1.1. Претрага библиографске грађе

Претрага библиографске грађе подразумева унос параметара за претрагу, иницирање претраге и приказ резултата. Унос параметара и приказ резултата претраге обавља клијентска апликација, а претрагу обавља текст сервер [5], који је описан и у поглављу 8.



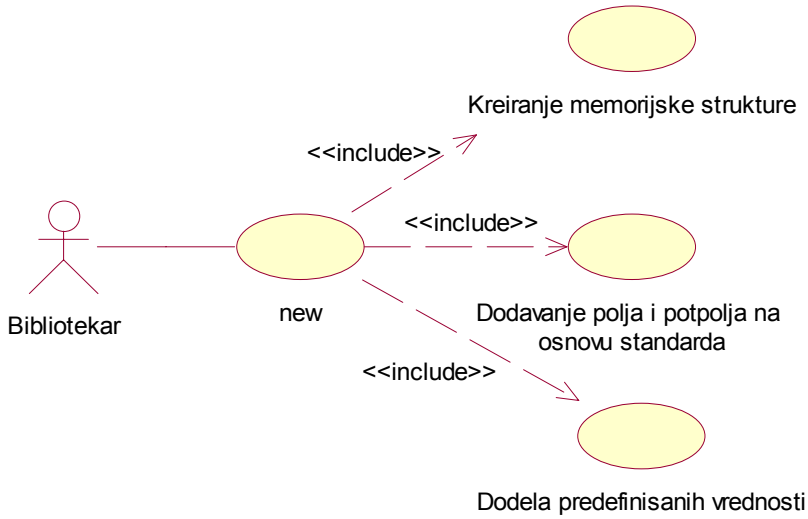
Слика 4.1. Дијаграм случајева коришћења процеса претраге

На слици 4.1. приказани су случајеви коришћења за процес претраге библиографске грађе. Постоје две врсте претраге у клијентској апликацији: по задатим префиксима и употребом *Dialog* језика [6, 7]. Случај коришћења *Pretraga po zadatim prefiksima* дефинише претрагу која се своди на попуњавање једног или више поља (максимално пет), одабир врсте префикса по којем ће се записи претраживати и, ако се претражује по више од једног префикса, одабир логичке везе између префикса (логичко И, ИЛИ и НЕ).

Случај коришћења *Pretraga Dialog jezikom* дефинише претрагу која се иницира укуцавањем *Dialog* упита у одговарајуће поље. Резултат претраге *Dialog* језиком је исти као код претраге по задатим префиксима, те је даљи рад на записима исти без обзира који тип претраге је извршен.

4.1.2. Креирање новог записа

Креирање новог записа почиње од креирања меморијске структуре која га представља, као што је дато дијаграмом случајева коришћења на слици 4.2.



Слика 4.2. Дијаграм случајева коришћења за креирање новог записа

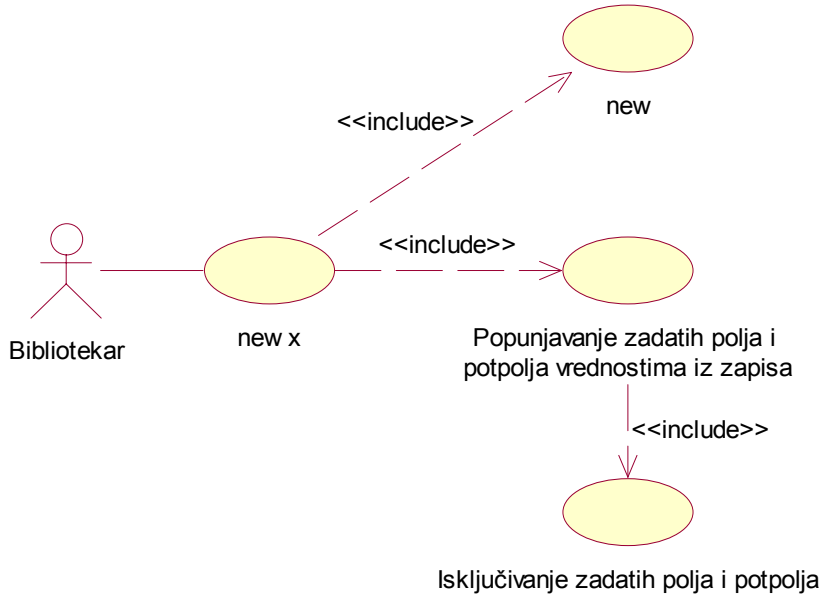
По иницирању команде *new* из командне линије (случај коришћења *new*), систем креира меморијску структуру која ће репрезентовати запис (случај коришћења *Kreiranje memorijske strukture*). Садржај меморијске структуре је дефинисан на основу података из окружења библиотекара. Окружење одређује списак поља и потпоља која ће по UNIMARC стандарду постојати у новом запису (случај коришћења *Dodavanje polja i potpolja na osnovu standarda*). Списак поља са потпољима зависи од типа публикације, нивоа обраде и нивоа обавезности, који су везани за библиотекара који је иницирао креирање записа (детаљно описано у поглављу 3).

На основу UNIMARC формата, одређена потпоља имају предефинисане (*default*) вредности, па се те вредности смештају у задата потпоља (случај коришћења *Dodela predefinisanih vrednosti*). Након попуњавања предефинисаним вредностима, запис је спреман за обраду која се своди на обраду меморијске структуре.

4.1.3. Креирање новог записа на основу постојећег

Креирање записа на основу постојећег записа представља специјалан случај креирања новог записа. Команда *new x*, где је *x* број неког постојећег записа, креира нов запис, попуњен подацима из постојећег. Да нови и постојећи запис не би били идентични, одређена поља се не преузимају из посто-

јећег записа. Дијаграм случајева коришћења на слици 4.3 илуструје креирање оваквих записа.

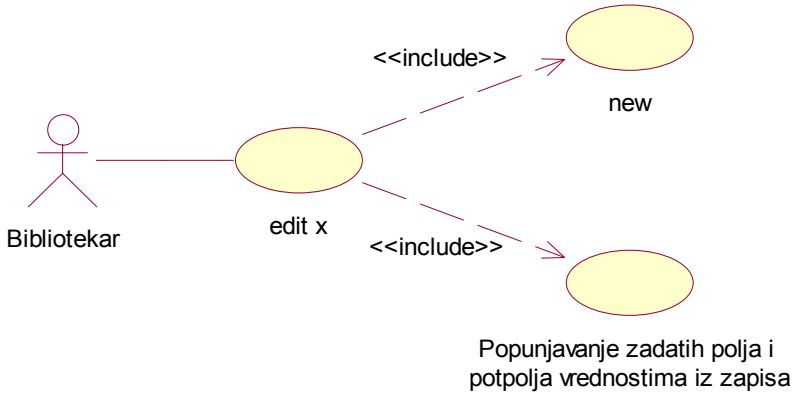


Слика 4.3. Дијаграм случајева коришћења креирања новог записа на основу постојећег

По иницирању команде *new x* (случај коришћења *new x*), креира се нов запис (случај коришћења *new*). Меморијска структура новог записа се попуњава вредностима из постојећег записа (случај коришћења *Popunjavanje zadatih polja i potpolja vrednostima iz zapisa*). Приликом попуњавања вредностима из постојећег записа, вредности одређених поља и потпоља се не преносе да би се избегло креирање идентичног записа (случај коришћења *Isključivanje zadatih polja i potpolja*). Корисник ова поља и потпоља мора да попуни конкретним вредностима за задату библиографску јединицу.

4.1.4. Модификација постојећег записа

Модификација постојећег записа се своди на његово претварање у меморијску структуру и обраду те структуре. Дијаграм случајева коришћења на слици 4.4 илуструје процес модификације записа.



Слика 4.4. Дијаграм случајева коришћења модификације постојећег записа

По иницирању команде *edit x* из командне линије (случај коришћења *edit x*), креира се и иницијализује меморијска структура (случај коришћења *new*). Меморијска структура се затим попуњава вредностима поља и потпоља из записа (случај коришћења *Popunjavanje zadatih polja i potpolja vrednostima iz zapisa*).

4.1.5. Обрада меморијске структуре која репрезентује запис

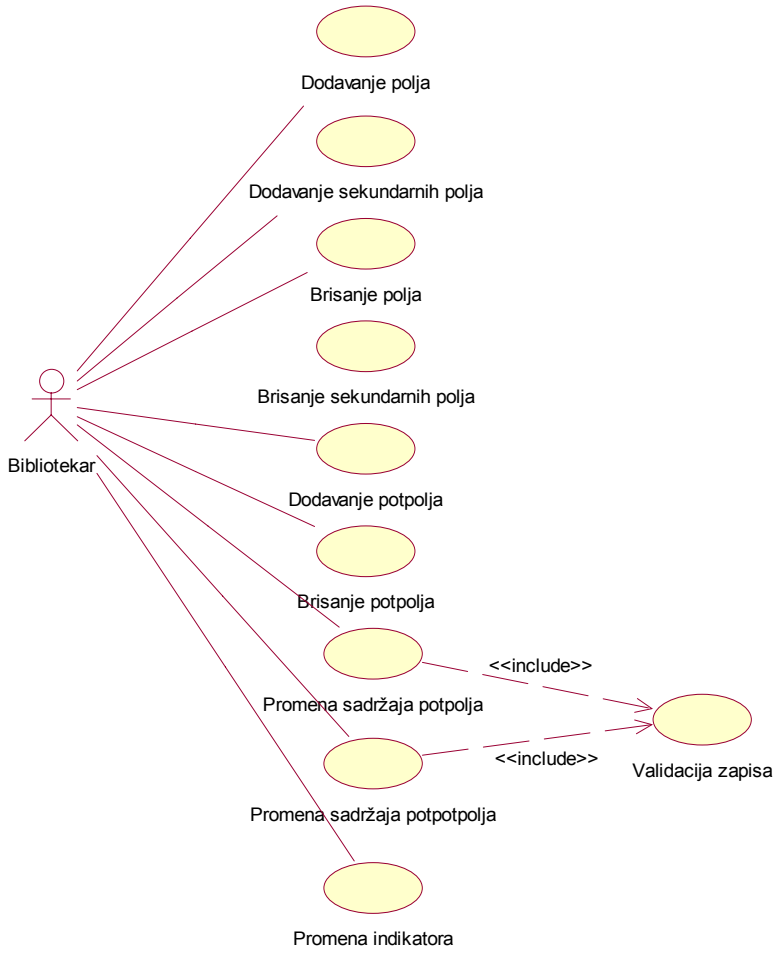
Након што је креирана и иницирана меморијска структура која репрезентује запис (командом *new*, *new x* или *edit x*), даља обрада записа се своди на обраду меморијске структуре.

Обрада меморијске структуре се своди на креирање, модификацију и брисање поља, потпоља и потпотпоља у меморијској структури, у складу са UNIMARC стандардом. Дијаграм случајева коришћења са слике 4.5 приказује процес обраде меморијске структуре.

Додавање нових поља (обичних или секундарних) и потпоља се своди на додавање одговарајућих објеката у меморијску структуру и извршава се у складу са UNIMARC форматом и окружењем библиотекара (случајеви коришћења *Dodavanje polja*, *Dodavanje sekundarnih polja* и *Dodavanje potpolja*). Брисање поља (обичних или секундарних) и потпоља заправо брише одговарајуће објекте из меморијске структуре (случајеви коришћења *Brisanje polja*, *Brisanje sekundarnih polja* и *Brisanje potpolja*).

Промена садржаја потпоља и потпотпоља (случајеви коришћења *Promena sadržaja potpolja* и *Promena sadržaja potpotpolja*) пролази кроз процес валидације (случај коришћења *Validacija zapisa*) којим се проверава исправ-

ност унетих података. Промена индикатора поља (случај коришћења *Promena Indikatora*) се изводи у складу са UNIMARC форматом (вредност индикатора није произвољна, већ се бира из списка понуђених вредности, које су у складу са UNIMARC форматом).

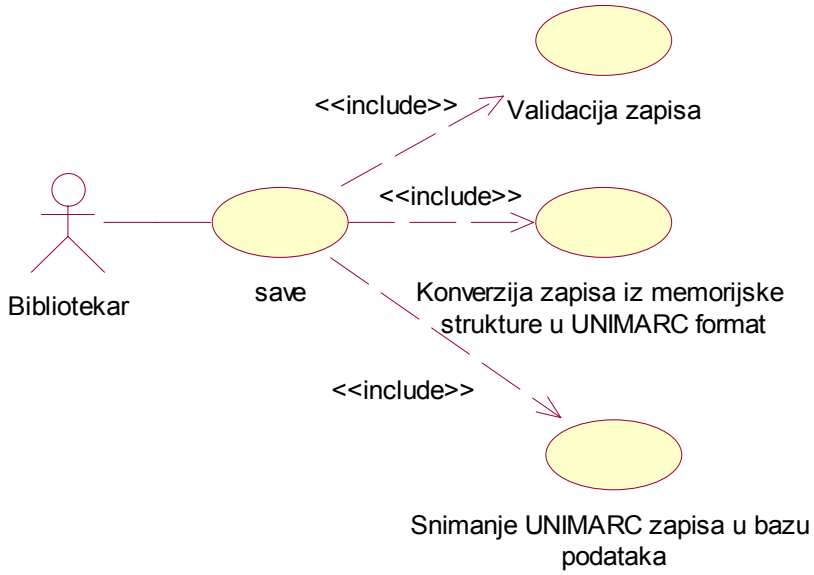


Слика 4.5. Дијаграм случајева коришћења обраде записа

Промена записа се не рефлектује моментално у бази података. Запис се све време обрађује само у меморији рачунара. Да би се промене учиниле трајним, потребно је снимити запис.

4.1.6. Снимање записа

Снимање записа у базу се своди на превођење записа из меморијске структуре у UNIMARC формат и снимање у базу података.



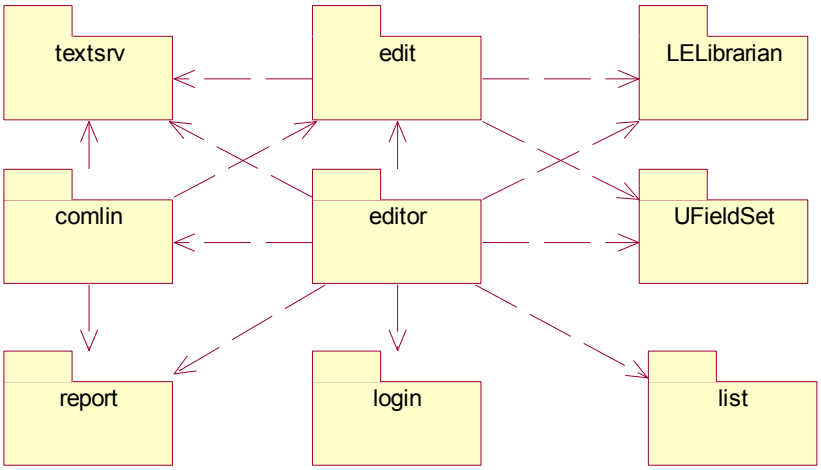
Слика 4.6. Дијаграм случајева коришћења за снимање записа

По иницирању команде *save* (случај коришћења *save*), врши се процес валидације (случај коришћења *Validacija zapisa*), којим се проверава постојање свих потребних поља и потпоља у запису. Окружење библиотекара и UNIMARC формат дефинишу обавезна поља и потпоља у запису. Након провере, запис се преводи у UNIMARC формат (случај коришћења *Konverzija memorijske strukture u UNIMARC format*) и предаје текст серверу. Без обзира да ли је запис нови или постојећи, текст сервер ће га снимити у базу података и ажурирати све потребне структуре (случај коришћења *Snimanje UNIMARC zapisa u bazu podataka*).

4.2. Архитектура апликације

4.2.1. Пакети клијент апликације

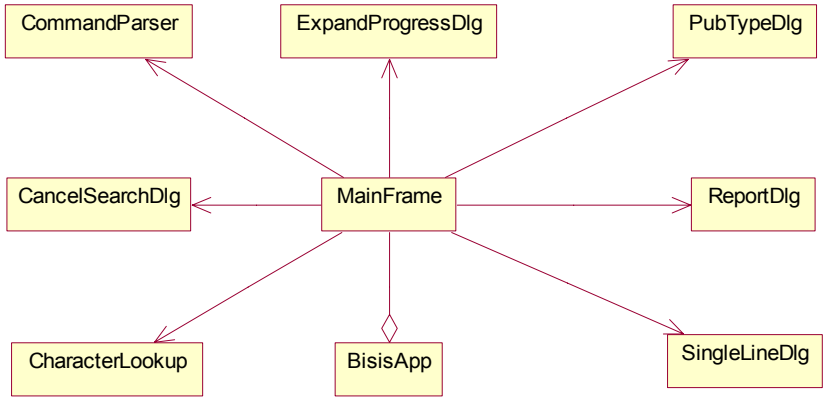
Објектно оријентисана спецификација клијент апликације, као и елемената који се баве обрадом библиографске грађе је дата у овом поглављу. На слици 4.7 је дат дијаграм компоненти клијентске апликације.



Слика 4.7. Архитектура клијент апликације

Основни пакет клијентске апликације је *editor* који садржи кориснички интерфејс и имплементира функције уноса и претраживања библиографске грађе. Пакет *edit* садржи класе које обрађују запис. Пакет *comlin* садржи класе које реализују интерпретер командне линије, пакет *login* садржи класе којима се корисник пријављује на систем. Пакет *textsrv* садржи класе за рад са текст сервером библиотечког информационог система, пакет *UFieldSet* репрезентује структуру података која описује UNIMARC формат, а пакет *LELibrarian* структуру података која описује окружење библиотекара. Пакет *list* садржи класе које визуализују податке који се користе у току обраде, а пакет *report* класе којима се генеришу извештаји.

На слици 4.8 дат је дијаграм класа пакета едитор. Основна класа овог пакета је класа *BisisApp*. Она управља корисничким интерфејсом, обезбеђује унос упита, приказ резултата претраге, као и измену структуре библиографске грађе. Она агрегира класу *MainFrame*, која је основна класа корисничког интерфејса и она контролише рад свих визуелних компоненти клијентске апликације.



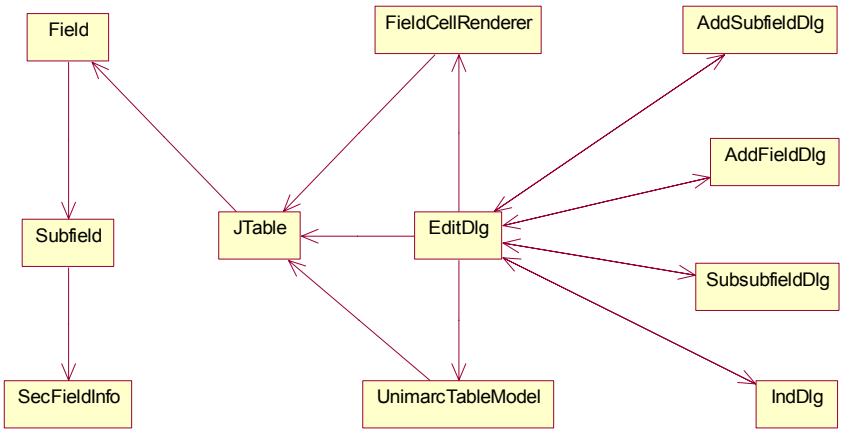
Слика 4.8. Дијаграм класа пакета *editor*

Класа *SingleLineDlg* репрезентује дијалог за унос библиотечких команди у *Dialog* језику. Класа `ReportDlg` обезбеђује рад са извештајима. Класа *PubTypeDlg* служи за дефинисање параметара окружења библиотекара (тип публикације, ниво обраде и ниво обавезности). Класа *ExpandProgressDlg* обезбеђује визуализацију резултата рада *expand* команде. Класа *CommandParser* реализује интерпретер командне линије. Класа *CancelSearchDialog* обезбеђује могућност прекида претраге. Класа *CharacterLookup* реализује одабир специјалних карактера који нису доступни са тастатуре.

4.2.2. Класе за обраду меморијске структуре

Обрада библиографске грађе подразумева креирање нових, као и модификацију и брисање постојећих записа. Записи су репрезентовани на два начина: у облику UNIMARC записа и у облику интерне меморијске структуре. UNIMARC запис се користи приликом снимања записа у базу података.

Интерна меморијска структура се користи приликом обраде записа и може се описати дијаграмом класа приказаним на слици 4.9.



Слика 4.9. Дијаграм класа за обраду меморијске структуре

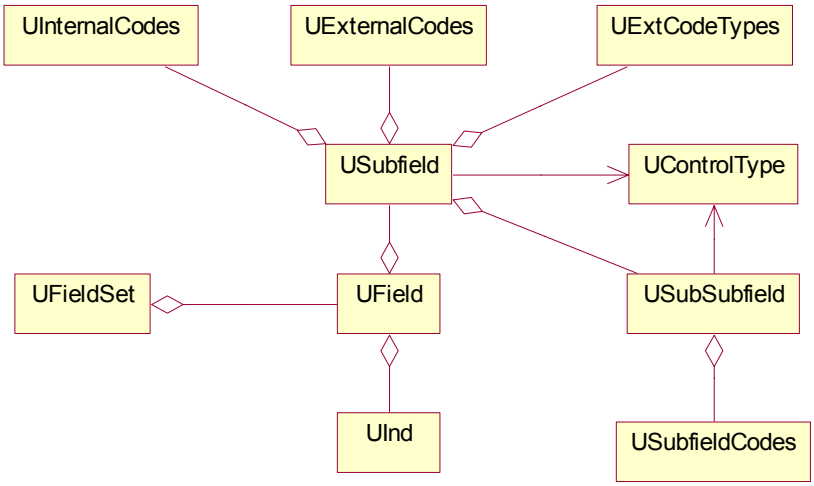
Класа *Field* представља једно поље UNIMARC записа. Класа *Subfield* представља потпоље UNIMARC записа. Класа *SecFieldInfo* описује секундарно поље по UNIMARC стандарду. Класа *JTable* садржи вектор објеката класе *Field* и та колекција објеката репрезентује запис. Класе *UnimarcTableModel* и *FieldCellRenderer* служе за визуелну контролу класе *JTable*, која приказује запис у табеларном облику.

Класа *EditDlg* садржи све визуелне компоненте које служе за обраду записа. Она садржи следеће класе: *IndDlg*, *AddFieldDlg*, *AddSubfieldDlg*, и *SubsubfieldDlg*. Ове класе омогућују манипулацију пољима, потпољима и потпотпољима (њихово додавање и модификацију).

Приликом обраде, запис се преводи из UNIMARC формата у меморијску структуру. Ако се креира нови запис, креира се и нова меморијска структура. У оба случаја, користе се пакети *UFieldSet* и *LELibrarian*, који садрже класе за дефиницију UNIMARC формата и окружења библиотекара.

4.2.2.1. UFieldSet пакет

UFieldSet пакет дефинише UNIMARC формат записа у меморији рачунара. UNIMARC формат записа је трајно дефинисан у бази података, а објекти класа из овог пакета га за потребе обраде записа читају из базе података и попуњавају своје атрибуте. Дијаграм класа који описује *UFieldSet* пакет је дат на слици 4.10.



Слика 4.10. Дијаграм класа *UFieldSet* пакета

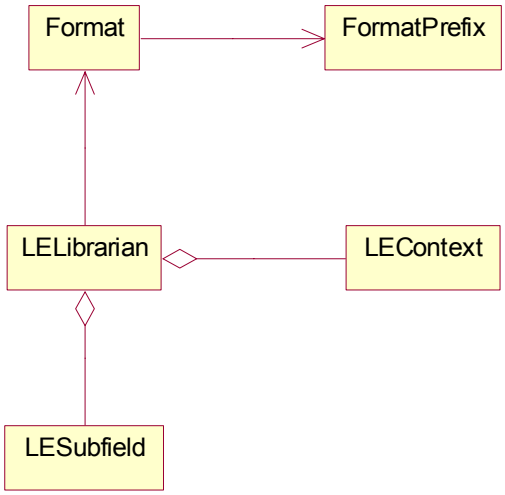
Класа *UFieldSet* обезбеђује приступ дефиницији UNIMARC стандарда. Класа *UField* представља опис поља по UNIMARC стандарду. Класе *USubfield*, *USubSubfield* и *UInd* представљају редом потпоља, потпотпоља и индикаторе. Класе *UInternalCodes*, *UExternalCodes* и *UExtCodeTypes* представљају интерне и екстерне шифарнике поља, као и типове шифарника, респективно. Класе *USubfieldCodes* и *UControlType* представљају шифарнике и типове шифарника за потпоља.

4.2.2.2. LELibrarian пакет

Овај пакет подржава окружење библиотекара. Подаци о окружењу се налазе у бази података, а објекти класа из овог пакета учитавају те податке и њима попуњавају своје атрибуте. Подаци које репрезентују ове класе су:

- подаци о библиотекару (корисничко име, шифра, стварно име и презиме, итд.),
- типови публикације које библиотекар може да обрађује,
- ниво обраде који има библиотекар,
- ниво обавезности који је додељен библиотекару,
- подразумеване вредности потпоља за одговарајуће записе и
- формате листања погодака.

Дијаграм класа је дат на слици 4.11.



Слика 4.11. Дијаграм класа *LELibrarian* пакета

Класа *LELibrarian* омогућује приступ окружењу библиотекара. Сваки библиотекар има дефинисан један или више контекста (класа *LEContext*) у оквиру којих може да обрађује библиографске записе. Контекст библиотекара садржи:

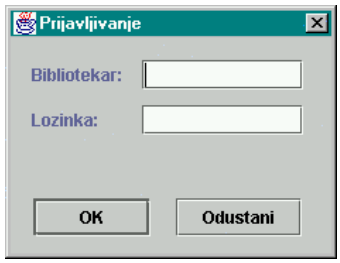
- тип публикације,
- ниво обраде,
- ниво обавезности,
- листу поља са потпољима за задати тип публикације (класа *LESubfield*),
- листу поља са потпољима за задати тип публикације и ниво обраде (класа *LESubfield*),
- листу поља са потпољима за задати тип публикације, ниво обраде и ниво обавезности (класа *LESubfield*).

Класе *Format* и *FormatPrefixes* дефинишу формате листања погодака приликом претраге библиографске грађе. Ове класе подржавају основне (предефинисане) формате и кориснички дефинисане формате. Основни формати се читају из базе података окружења библиотекара, а кориснички дефинисане формате дефинише библиотекар из клијентске апликације (детаљније описано у следећем поглављу).

4.3. Имплементација

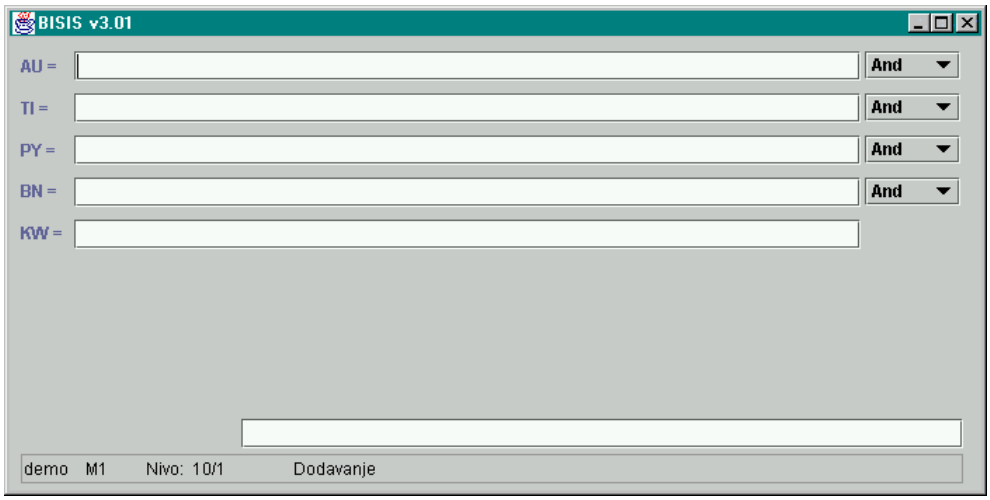
Имплементација апликације за обраду библиотечких записа је изведена у програмском језику Java, уз ослонац на изабрани релациони систем за управљање базом података. Тестирани су следећи системи за управљање базама података: *Oracle*, *SAP DB*, *Informix* и *MS SQL Server*.

По стартовању апликације добија се прозор за пријављивање библиотекара (слика 4.12).



Слика 4.12. Прозор за пријављивање на систем

По успешном пријављивању на систем, добија се основни прозор за претраживање (слика 4.13).



Слика 4.13. Основни прозор за претраживање

Прозор за претраживање се састоји из следећих елемената:

– пет линија за претраживање;
– командне линије;
– статусне линије.

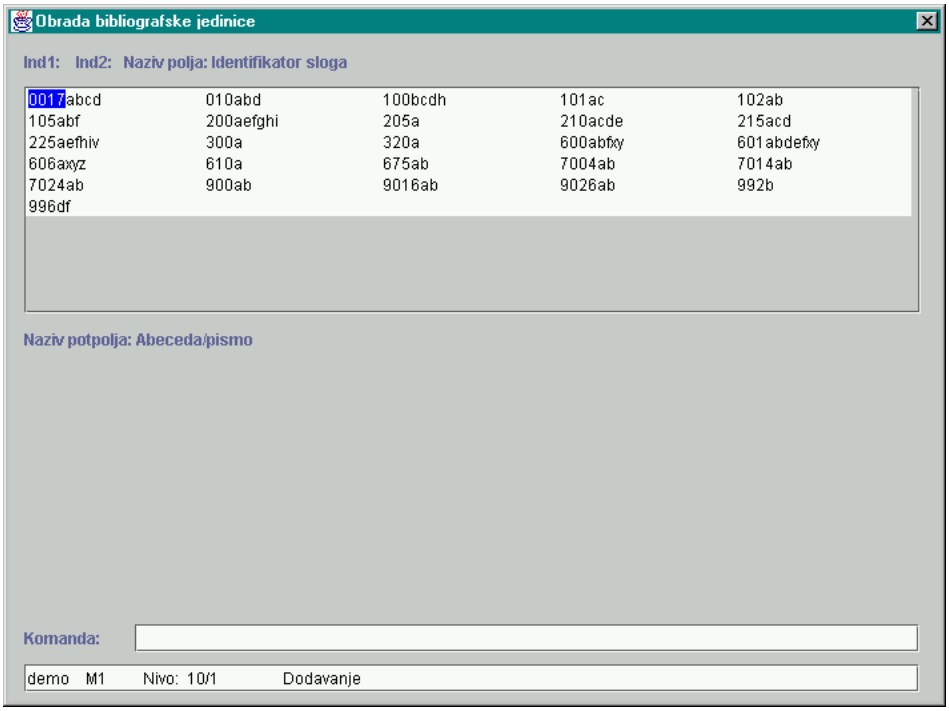
Линија за претраживање састоји се од префикса за претраживање који се може изабрати из шифарника префикса, једнолинијског едитора за унос појмова за претраживање и падајуће листе за избор оператора.

Командна линија приказује број погодака, односно записа пронађених у бази на основу појмова за претраживање и омогућава задавање команди система.

Статусна линија приказује податке о текућем режиму обраде (пријављени библиотекар, тип публикације, итд.).

Библиографски записи претражују се уносом параметара претраге у једној или више линија за претраживање или уносом *Dialog* упита у одговарајућем прозору. Број погодака који задовољавају параметре претраге је приказан у заглављу командне линије. Пронађени записи се могу обрадити или обрисати. Осим наведених операција могуће је и креирати нов запис, као потпуно нов запис или на основу пронађеног записа.

Без обзира да ли се обрађује пронађени запис или креира нов, активира се прозор за обраду записа (слика 4.14).



Слика 4.14. Прозор за обраду записа

Прозор за обраду записа састоји се од следећих делова:

- прва линија за поруке;
- директоријум;
- друга линија за поруке;
- едитор за унос података;
- командна линија;
- статусна линија.

Директоријум је део прозора за обраду записа који служи за приказ структуре записа тј. његових поља и потпоља.

У првој линији за поруке исписују се вредности индикатора и назив тренутно селектованог (текућег) поља у директоријуму.

У другој линији за поруке исписује се назив текућег потпоља у директоријуму.

Едитор за унос података отвара се након позиционирања на одговарајуће поље и потпоље у директоријуму и притиска на тастер **<Enter>**. Едитор може бити вишелинијски или једнолинијски.

Командна линија служи за унос команди.

У статусној линији исписују се исти подаци као и у статусној линији у прозору за претраживање.

4.3.1. Обрада записа у директоријуму

Директоријум је део прозора за обраду записа који служи за приказ структуре записа, односно његових поља и потпоља. Кретање по пољима врши се притиском на тастере **<↑>**, **<↓>**, **<→>** и **<←>**. Текуће поље, односно потпоље, су јаче осветљени у односу на остала поља и потпоља. У директоријуму су могуће следеће акције:

- унос садржаја потпоља
- додељивање вредности индикатора;
- додавање поља;
- брисање поља;
- враћање избрисаног поља;
- додавање потпоља;
- брисање потпоља;
- враћање избрисаног потпоља;
- снимање записа;

- излаз из записа без снимања измена;
- промена нивоа обраде, нивоа обавезности и типа публикације;
- унос команде.

Детаљан опис обраде библиографске грађе и упутство за коришћење клијентске апликације дато је у [8].

4.4. Закључак

У овом поглављу описана је софтверска архитектура и имплементација клијентске апликације која омогућава претрагу, измену, брисање и креирање нових библиографских записа. Реализована је меморијска структура која репрезентује библиографски запис по UNIMARC формату, чиме је модификација записа сведена на рад са меморијском структуром. Осим основних операција над записом (креирање нових, измена и брисање постојећих), обезбеђено је да су све операције у сагласности са UNIMARC стандардом, као и са окружењем библиотекара. Такође, рад на елементима појединачног записа (поља, потпоља, потпотпоља, секундарна поља и индикатори) је у сагласности са UNIMARC стандардом.

Референце

[1] M. Vidaković. *Implementacija korisničkog interfejsa na bibliotečkom informacionom sistemu u Java okruženju*. Zbornik apstrakata SIFON'97, Zlatibor 1997., str. 10.

[2] М. Видаковић. *Објектно орјентисана спецификација и имплементација корисничког интерфејса за библиотечки систем*. Магистарска теза, Факултет техничких наука, Нови Сад, 1998. http://diglib.ns.ac.yu/ndltd/docs/set1/ndltd7/MinjaMagistarski.PDF

[3] D. Surla, Z. Konjović, B. Milosavljević, M. Vidaković. *Bibliotečki informacioni sistem BISIS ver. 3.01*. Deveta međunarodna konferencija "Informatika u obrazovanju, kvalitet i nove informacione tehnologije", Zrenjanin 2000., str 494-504.

[4] D. Todorić, T. Vulić. *Data structure of UNIMARC standard*. Proceedings of the X Conference on Applied Mathematics PRIM'95, Budva 1995, pp. 309-314.

[5] Б. Милосављевић. *Текст сервер UNIMARC записа*. Магистарска теза, Факултет техничких наука, Нови Сад, 1999., http://diglib.ns.ac.yu/ndltd/docs/set1/ndltd6/textsrv.pdf

[6] I. Holo. *Command language functional description of library information system*. Proceedings of the XI Conference on Applied Mathematics PRIM'96, Budva 1996, pp. 117-126.

[7] I. Fišl, Z. Konjović, D. Surla. *Design and Implementation of the Query Language Interpreter for Bibliographic Data Retrieval*. Novi Sad Journal of Mathematics, vol. 28. num. 3, 1998., pp. 11-20.

[8] Д. Сурла, З. Коњовић, Б. Пуповац, Б. Милосављевић, М. Видаковић, Т. Тошић, Т. Зубић. *Упутство за коришћење библиотечког софтверског система БИСИС вер. 3*. Група за Информационе технологије, Нови Сад, 2003.

Поглавље 5

Команде библиотечког информационог система

Б. Пуповац

Команде у библиотечком информационом систему уведене су да би се библиотекарима обезбедио комфорнији начин рада при обради библиографске грађе. У раду [1] приказан је модел команди на основу којег је дат предлог за имплементацију команди у софтверском систему БИСИС вер. 1. У раду [2] дата је објектно-оријентисана спецификација команди која се односи на команде софтверског система БИСИС вер. 3. За приказ функционалности дела система који се односи на команде са тачке гледишта корисника система коришћен је дијаграм случајева коришћења. Статички аспекти приказани су дијаграмима класа, док су за приказ динамичких аспеката коришћени дијаграми секвенци, сарадње, стања и активности. Моделирање команди извршено је коришћењем UML (*Unified Modeling Language*) нотације [3, 4].

Апликација за обраду библиографске грађе омогућава библиотекару претраживање базе података библиографске грађе и обраду библиотечких јединица. Када библиотекар врши претраживање базе података библиографске грађе, апликација се налази у моду претраживања. Када библиотекар ради са записом, било да едитује структуру записа или креира нови запис, апликација се налази у моду директоријума, а када библотекар уноси садржај потпоља записа апликација се налази у моду уноса. У зависности од мода, библиотекар може да користи различит скуп команди, при чему у моду уноса није дозвољен рад са командама.

У оквиру софтверског система БИСИС вер. 3 реализоване су следеће команде:

- *connect*,
- *delete*,
- *display*,
- *edit*,

- *exit*,
- *format*,
- *list*,
- *new*,
- *print*,
- *quit*,
- *save*.

Без обзира на мод, команде се уносе из командне линије. Команде *display*, *format*, *list* и *quit* могуће је задати и у једном и у другом моду, али у сваком од њих, команда има или различиту функционалност, или се разликује по томе да ли се задаје са или без параметара. У наредном тексту дат је кратак опис наведених команди.

Команда *connect* омогућава кориснику комуникацију са сервером за узајамну каталогизацију. Ова команда се може применити искључиво у моду претраживања. Детаљнији опис функционалности ове команде описан је у Поглављу 6 ове монографије.

Команда *delete* служи за брисање записа из базе података библиографске грађе. Команда се примењује искључиво у моду директоријума. Да би се ова команда успешно извршила, библиотекар мора претходно да едитује запис из базе података библиографске грађе кога жели да обрише. Када библиотекар зада ову команду, приказује се порука која омогућава избор између две опције: опција којом библиотекар потврђује брисање записа и опција којом се одустаје од брисања. При задавању ове команде не задају се параметри.

Команда *display* омогућава исписивање записа на екрану у облику који зависи од изабраног формата исписа. Ова команда се може применити у моду претраживања и у моду директоријума. Када се ова команда задаје у моду претраживања, потребно је навести параметре који представљају редне бројеве погодака записа за које се жели формирање исписа. Сходно томе, примена команде у овом моду захтева да се претходно изврши претраживање базе података библиографске грађе и да је то претраживање успешно. У моду директоријума ова команда омогућава формирање исписа у изабраном формату записа који се обрађује. Када се примењује у овом моду команда *display* нема параметре.

Команда *edit* омогућава приказ структуре и ажурирање изабраног записа. Примена ове команде захтева да се претходно изврши претраживање базе података библиографске грађе и да је то претраживање успешно. Као параме-

тар ове команде наводи се редни број поготка, који припада скупу резултата претходног претраживања. Команда *edit* се примењује искључиво у моду претраживања.

Команда *exit* се може применити у моду директоријума. Када библиотекар зада ову команду, приказује се порука која омогућава избор између две опције. Након избора опције библиотекар прелази из мода директоријума у мод претраживања, с тим да се, уколико је библиотекар изабрао опцију снимања записа, запис снима у базу података библиографске грађе. У супротном, запис неће бити снимљен. При задавању ове команде не задају се параметри.

Команда *format* омогућава постављање текућег формата исписа, дефинисање нових формата или преглед постојећих формата у зависности од синтаксе којом се ова команда задаје. Ако се уз команду *format* зада назив већ постојећег формата, текући формат исписа поставља се на формат чији је назив задат. Дефинисање новог формата врши се тако што се уз команду наведе име новог формата и скуп префикса, међусобно одвојених зарезима. Овако дефинисан формат омогућава приказ записа по префиксима, који су наведени у скупу. Преглед постојећих формата врши се тако што се као параметар наведе знак ?. Тада се добија списак свих дефинисаних формата, које библотекар може да изабере за испис записа. Ову команду могуће је применити у моду претраживања и у моду директоријума.

Команда *list* омогућава испис извештаја (каталошких листића) на екрану. Ова команда се може применити у моду претраживања и у моду директоријума. Када се ова команда задаје у моду претраживања, потребно је навести параметре који представљају редне бројеве погодака записа за које се жели формирање извештаја. Сходно томе, примена команде у овом моду захтева да се претходно изврши претраживање базе података библиографске грађе и да је то претраживање успешно. У моду директоријума ова команда омогућава формирање извештаја текућег записа чија структура се едитује. Када се примењује у овом моду команда *list* нема параметре. Без обзира у ком моду се задаје, за ову команду се може навести тип извештаја. Уколико се тип извештаја не наведе, структура извештаја се формира на основу текуће вредности типа извештаја библиотекара, који је преузет приликом иницијализације параметара апликације, или је постављен у току извршавања апликације.

Команда *new* омогућава креирање новог записа. Ова команда се може применити искључиво у моду претраживања са или без параметара. Команда се задаје без параметара уколико се јединица коју је потребно обрадити не налази у бази података библиографске грађе. Сходно томе, примени команде

new без параметара претходи неуспешно претраживање ове базе. Када се ова команда примењује са параметром, параметар представља редни број поготка који ће представљати узорак. Узорак је запис библиографске грађе који садржи велики број истих података као и документ који се обрађује. Сходно томе, примена команде *new* са параметром захтева да се претходно изврши претраживање базе података библиографске грађе и да је то претраживање успешно.

Команда *print* омогућава да се библиотекарима прикажу сложени извештаји, као што су инвентарска књига, библиографија, билтен приновљених књига. Ова команда се може применити искључиво у моду претраживања, без параметара. Детаљнији опис функционалности ове команде описан је у Поглављу 7 ове монографије.

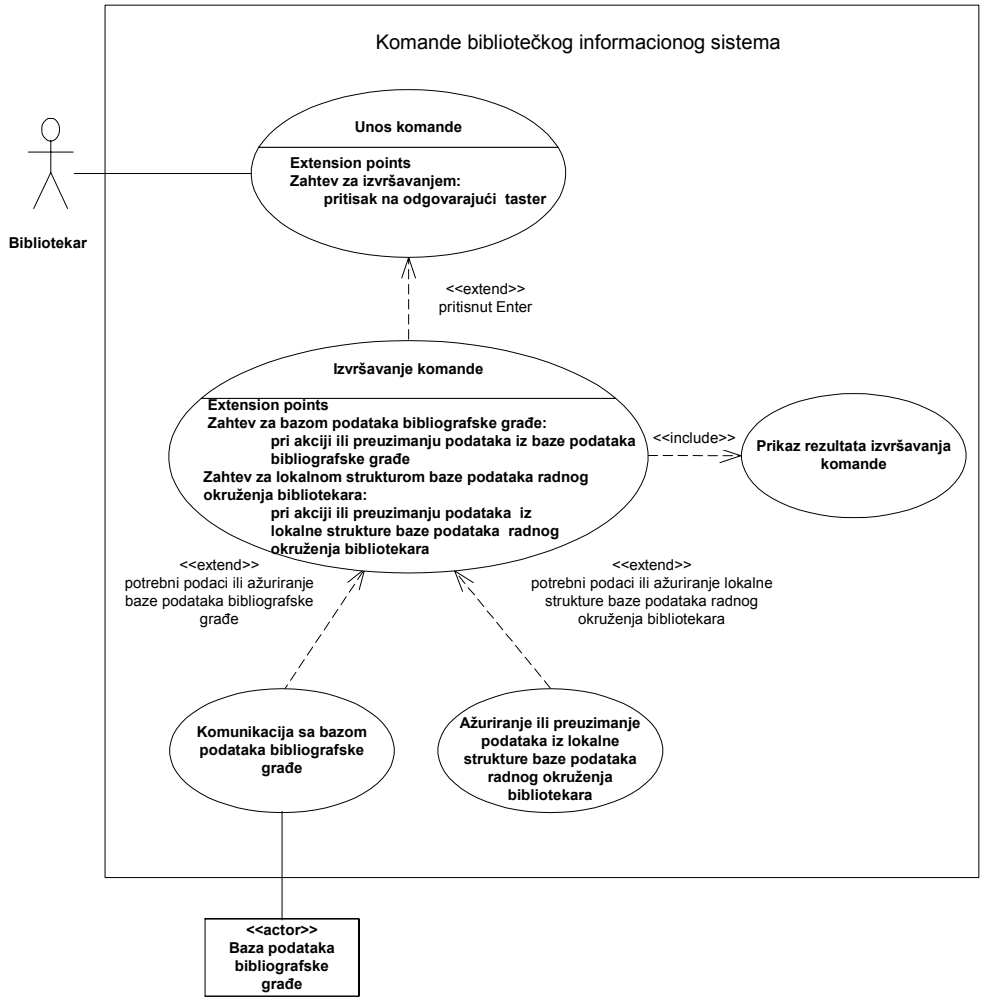
Команда *quit* се може применити у моду претраживања и у моду директоријума. Функционалност команде зависи од мода извршавања. Ако се команда зада у моду претраживања, извршавањем ове команде излази се из апликације за обраду библиографске грађе. Ако се команда зада у моду директоријума, библиотекар прелази из мода директоријума у мод претраживања, а запис чија структура је едитована неће бити снимљен у базу података библиографске грађе.

Команда *save* врши снимање записа у базу података библиографске грађе. Команда се може применити искључиво у моду директоријума. Да би запис био успешно снимљен потребно је да библиотекар унесе садржај свих потпоља дефинисаних за ниво обавезности који је постављен за радно окружење библиотекара који обрађује запис. Приликом задавања ове команде, пре снимања записа у базу библиографске грађе, билиотекару се приказују основни подаци записа који се обрађује, као што су идентификациони број записа, аутор, наслов, сигнатура. Након тога, библиотекар може да бира између две опције: опција којом потврђује снимање записа у базу података библиографске грађе и опција којом запис неће бити снимљен (а апликација ће остати у моду директоријума). При задавању ове команде не задају се параметри.

5.1. Дијаграм случајева коришћења команди

Дијаграм случајева коришћења команди приказан је на слици 5.1. Приликом моделирања уочени су следећи случајеви коришћења: *Unos komande*, *Izvršavanje komande*, *Prikaz rezultata izvršavanja komande*, *Komunikacija sa bazom podataka bibliografske građe* и *Ažuriranje ili preuzimanje podataka iz*

lokalne strukture baze podataka radnog okruženja bibliotekara, као и учесници *Bibliotekar* и *Baza podataka bibliografske građe*.



Слика 5.1. Дијаграм случајева коришћења команди

Случај коришћења *Unos komande* описује функцију уноса команде у командну линију. Том приликом, библиотекар уноси команду и по потреби параметре команде. Уколико је задовољен специфициран услов *pritisnut Enter* овај случај коришћења проширује се случајем коришћења *Izvršavanje komande*.

Случај коришћења *Izvršavanje komande* описује функцију извршавања команде у систему. Том приликом, врши се парсирање садржаја командне линије и извршава се конкретна команда. Извршавање овог случаја коришћења укључује и извршавање случаја коришћења *Prikaz rezultata izvršavanja komande*. Уколико је задовољен услов *potrebni podaci ili ažuriranje baze podataka bibliografske građe* извршава се случај коришћења *Komunikacija sa bazom podataka bibliografske građe*, а уколико је задовољен услов *potrebni podaci ili ažuriranje lokalne strukture baze podataka radnog okruženja bibliotekara* извршава се случај коришћења *Ažuriranje ili preuzimanje podataka iz lokalne strukture baze podataka radnog okruženja bibliotekara*.

Случај коришћења *Prikaz rezultata izvršavanja komande* описује функцију која реализује приказ резултата извршавања команде, било оно успешно или не. Уколико је команда успешно извршена изглед резултата извршавања зависи од врсте команде. Уколико је команда неуспешно извршена библиотекару се приказује порука о грешци. Уколико је команда неисправно задата приказује се порука о погрешно унетом називу команде или погрешној синтакси којом је библиотекар задао команду.

Случај коришћења *Ažuriranje ili preuzimanje podataka iz lokalne strukture baze podataka radnog okruženja bibliotekara* описује функцију која омогућава комуникација са локалном структуром, у којој се налазе подаци из базе података радног окружења библиотекара, ради преузимања или ажурирања података из ове структуре. У случају извршавања неких од команди врши се преузимање података из те локалне структуре ради добијања додатних података у циљу извршавања тих команди. Неке од команди врше ажурирање података те структуре у циљу даљег рада библотекара у апликацији.

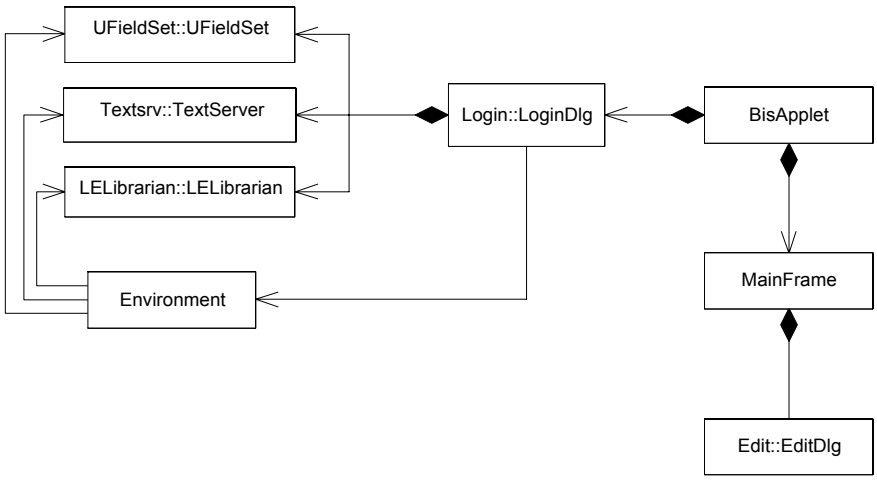
Случај коришћења *Komunikacija sa bazom podataka bibliografske građe* описује функцију која омогућава комуникацију са базом података библиографске грађе ради преузимања или ажурирања података из ове базе. У случају неких команди врши се преузимање записа из базе података библиографске грађе ради даљег рада са тим записима. Неке од команди врше ажурирање записа базе података библиографске грађе.

5.2. Дијаграм класа команди

Дијаграм класа команди библиотечког информационог система моделиран је коришћењем UML-а. Основна идеја приликом креирања дијаграма класа команди је да се искористи принцип полиморфизма, који подржава програмски језик Java [5, 6]. Овај принцип искоришћен је тако што је креи-

рана апстрактна класа са апстрактном операцијом *execute*. Све класе које имплементирају конкретне команде наслеђују ову апстрактну класу и реализују операцију *execute*.

Приликом формирања дијаграма класа за сваку команду може се издвојити дијаграм класа који је заједнички за све команде, а који се може повезати са дијаграмом класа сваке конкретне команде. Овај дијаграм приказан је на слици 5.2.



Слика 5.2. Дијаграм класа заједнички за све команде

Стартна класа софтверског система БИСИС вер. 3 је класа *BisApplet*. Из ове класе приступа се објекту класе *LoginDlg* која припада пакету *Login* и која омогућава пријављивање библиотекара на систем. Приликом пријављивања конкретног библиотекара, креирају се објекти класа *UFieldSet* и *LELibrarian* који омогућавају да се подаци из базе података YUMARC формата и базе података радног окружења библиотекара чувају у локалној структури. Такође, формира се објекат класе *TextServer* преко кога се остварује веза са базом података библиографске грађе. Класа *Environment* чува све битне податке који су потребни приликом извршавања апликације и рада библиотекара као што су, на пример, референце на објекте који омогућавају рад са локалним структурама, текст сервером и други.

Мод претраживања и мод директоријума, осим што утичу на врсту команди које се могу извршити, одређују и изглед прозора апликације. Класа која имплементира мод претраживања је класа *MainFrame*, а класа која имплементира мод директоријума је класа *EditDlg*. Свака од ових класа, као

графичку компоненту садржи командну линију, која омогућава библиотекару унос одговарајућих команди.

У наредном делу текста приказан је дијаграм класа команде *delete*, који представља пример дијаграма класа конкретне команде библиотечког информационог система. Дијаграма класа команди *display* и *new*, као и све врсте динамичких дијаграма команди *delete*, *display* и *new* приказане су у раду [2].

5.2.1. Дијаграм класа команде delete

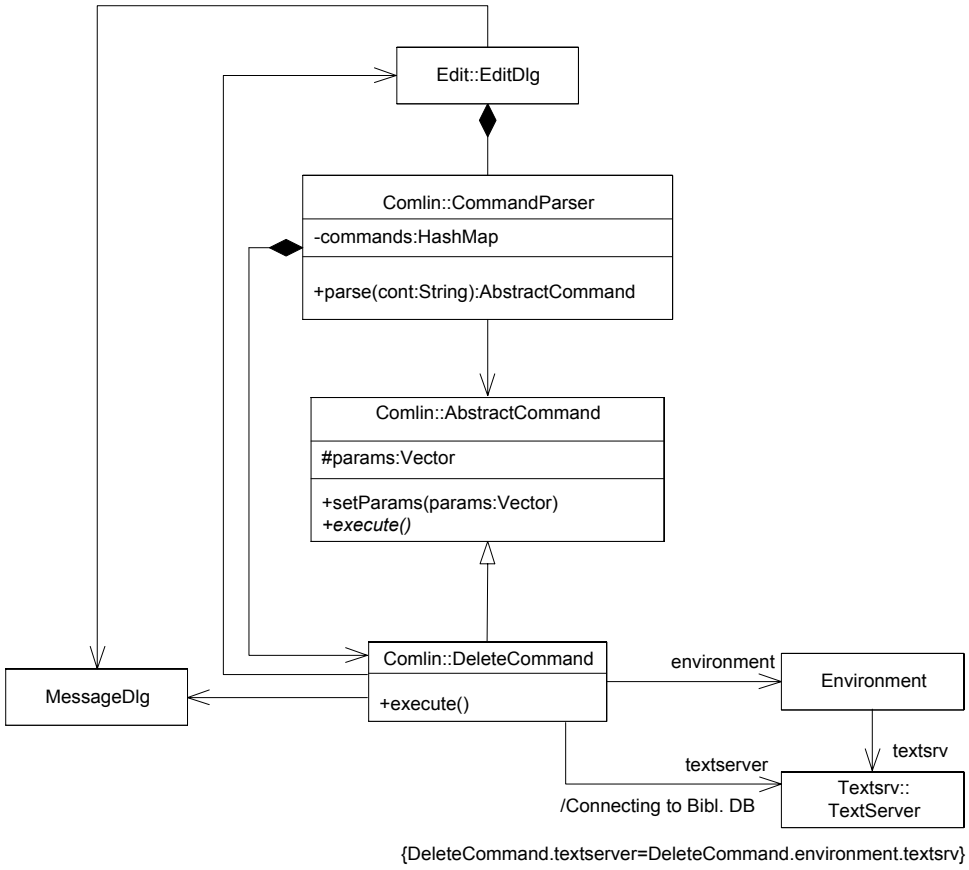
Команда *delete*, као што је већ речено, може да се извршава у моду директоријума. Дијаграм класа команде *delete* дат је на слици 5.3. Објекат типа *CommandParser* креира се у оквиру класе *EditDlg*. Приликом креирања тог објекта креирају се објекти класа које имплементирају конкретне команде које се могу извршити у овом моду. Референце на овако креиране објекте чувају се у структури *HashMap*, где се налазе парови (назив_команде, референца_на_објекат_команде).

Парсирање садржаја командне линије извршава се у оквиру операције *parse*. Приликом парсирања, одваја се назив команде од параметара. Параметри се операцијом *setParams* класе *AbstractCommand* постављају у променљиву *params* тако да буду доступни свим класама које наслеђују ову класу. Уколико је конкретна команда пронађена у структури *HashMap* на основу свог назива, операција *parse* враћа објекат типа *AbstractCommand* и на основу враћеног објекта позива се операција *execute* конкретне команде, у овом случају команде *delete* коју реализује класа *DeleteCommand*.

Како је приликом извршавања команде *delete* потребно имати везу са базом података библиографске грађе, класа *DeleteCommand* је повезана са класом *Environment* која чува референцу на објекат класе *TextServer* који омогућава акције над том базом.

Веза са класом *EditDlg* остварена је због постављања вредности одговарајућих параметара, који зависе од извршавања ове команде.

Приликом извршавања команде *delete* поруке које се приказују библиотекару реализују се помоћу операција класе *MessageDlg*.

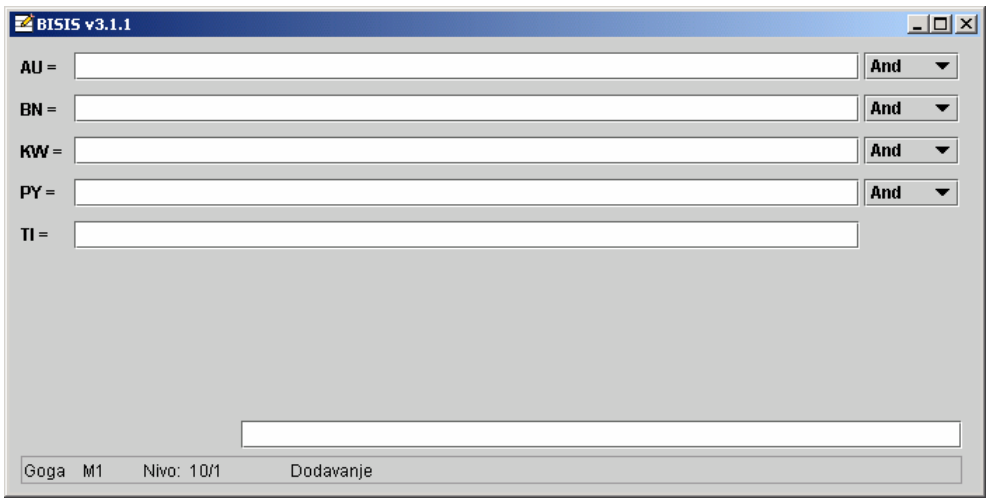


Слика 5.3. Дијаграм класа команде *delete*

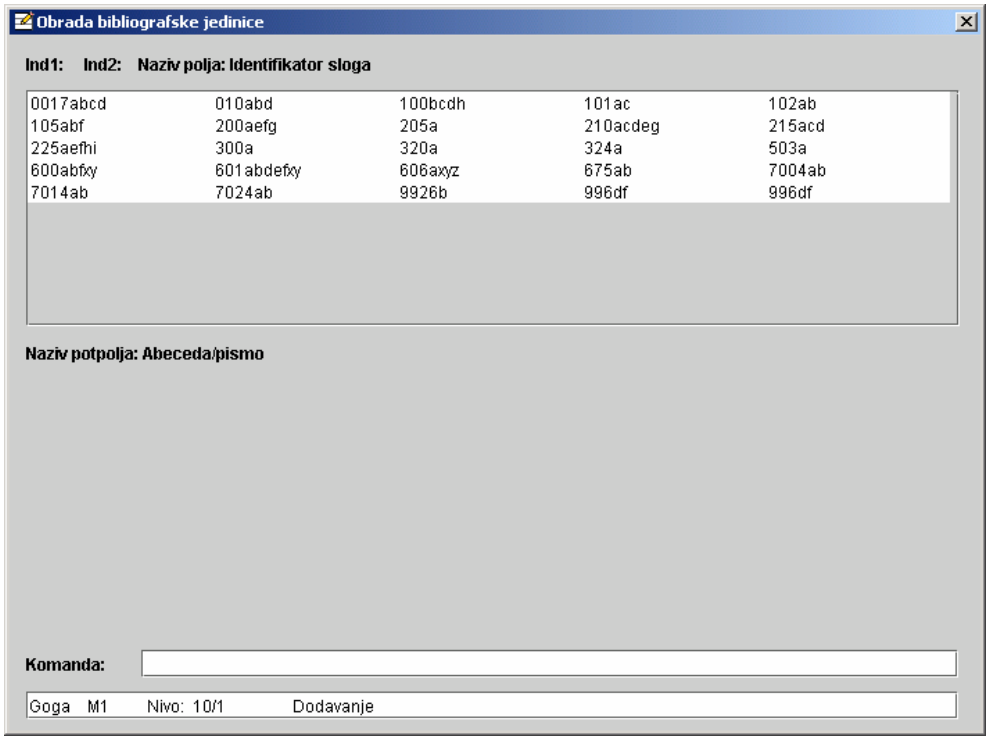
5.3. Пример коришћења команди

У овом делу приказан је пример, који илуструје начин задавања, изглед прозора апликације и акције које се извршавају задавањем команде *delete* у апликацији за обраду библиографске грађе.

Као што је речено у уводу, библиотекар може да зада команде у моду претраживања и моду директоријума. Сваки мод представљен је посебним прозором. Изглед прозора који одговара моду претраживања дат је на слици 5.4, док је изглед прозора која одговара моду директоријума дат на слици 5.5.

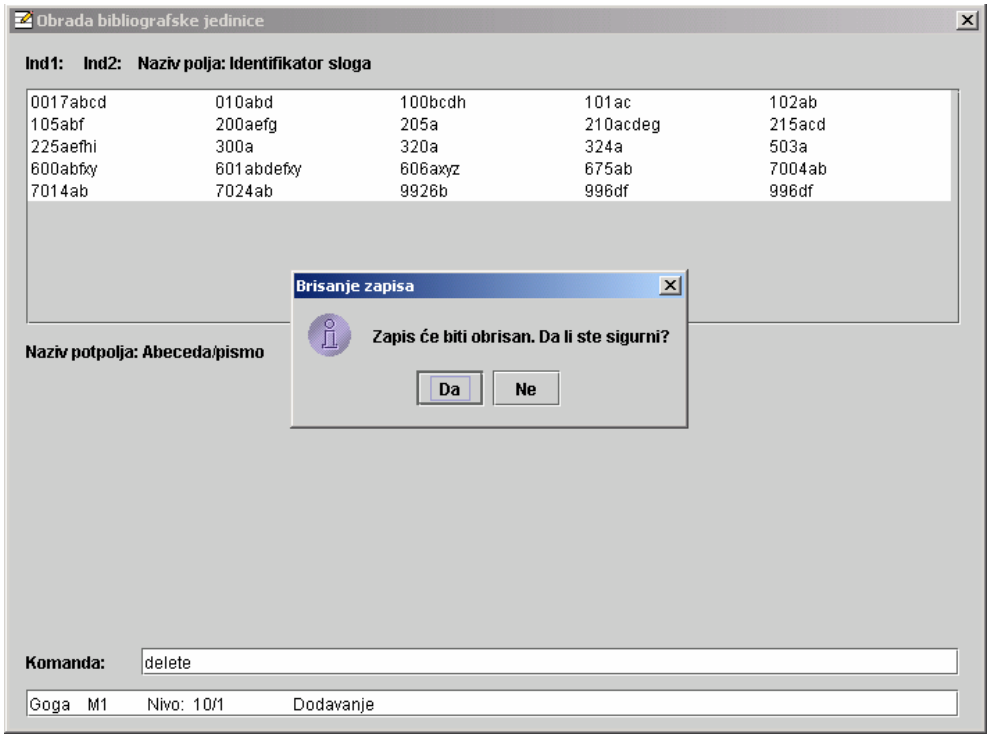


Слика 5.4. Изглед прозора у моду претраживања



Слика 5.5. Изглед прозора у моду директоријума

Библиотекар може да брише запис у моду директоријума задавањем команде *delete* из командне линије. Приликом брисања записа, библиотекару се приказује упозорење да ће запис бити обрисан. Библиотекар може да потврди акцију брисања, или да одустане од ње, као што се види на слици 5.6. Уколико је библиотекар потврдио акцију брисања, запис се брише из базе података библиографске грађе при чему се приказује порука да ли је запис успешно обрисан. Опис коришћења свих команди детаљно је описано у [7].



Слика 5.6. Изглед прозора приликом брисања записа

5.4. Закључак

Команде у библиотечком информационом систему уведене су да би се библиотекарима обезбедио комфорнији начин рада при обради библиографске грађе. Команде се користе у апликацији за обраду библиографске грађе, при чему се у зависности од мода апликације (мод претраживања или мод директоријума) разликују скупови команди које се могу примењивати у одређеном моду. Објектно-оријентисана спецификација команди извршена је коришћењем UML нотације. На основу коришћења принципа полиморфизма

омогућено је једноставно додавање нових команди. Нова команда би се имплементирала тако што би наследила апстрактну класу *AbstractCommand* и имплементирала методу *execute*. На тај начин не би се мењала имплементација постојећих команди приликом додавања нове команде у систем.

Референце

[1] И. Холо. *Командни језик библиотечког информационог система*. Магистарска теза, Природно-математички факултет, Институт за математику, Нови Сад, 1997.

[2] Б. Пуповац. *Моделирање и импелемнтација команди библиотечког информационог система*. Магистарска теза, Природно-математички факултет, Институт за математику, Нови Сад, 2000.

[3] G. Booch, J. Rumbaugh, I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley Longman, Inc., 1999.

[4] J. Rumbaugh, I. Jacobson, G. Booch. *The Unified Modeling Language Reference Manual*. Addison-Wesley Longman, Inc., 1999.

[5] B. Eckel. *Thinking in Java*, 2nd edition. 2000. http://www.EckelObjects.com

[6] *JavaTM 2 Platform, Standard Edition, v 1.3.1 API Specification*. http://java.sun.com/j2se/1.3/docs/api/

[7] Д. Сурла, З. Коњовић, Б. Пуповац, Б. Милосављевић, М. Видаковић, Т. Тошић, Т. Зубић. *Упутство за коришћење бибилотечког софтверског система БИСИС вер. 3*. Група за информационе технологије, Нови Сад, 2003.

Поглавље 6

Сервер за узајамну каталогизацију

М. Видаковић

Узајамна каталогизација [1, 2] је процес преузимања записа о обрађеним публикацијама са неког од доступних сервера библиографских записа. Овим процесом добија се квалитетнија и бржа обрада библиографске грађе у локалном сегменту библиотечке мреже. Процес узајамне каталогизације базиран је на порукама типа захтев и одговор. Фазе овог процеса су:

- библиотекар шаље захтев за претраживање доступним серверима библиографских записа,
- сервер шаље библиотекару одговор о резултатима претраживања,
- на основу добијеног одговора библиотекар шаље захтев за преузимање записа,
- сервер шаље библиотекару одговор у коме се налазе тражени записи у одговарајућем формату и
- библиотекар у локалу допуњава обраду преузетих записа.

Систем подржава преузимање записа у облику:

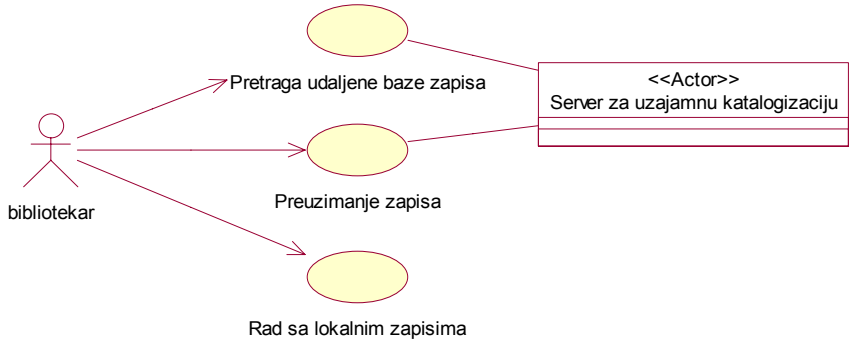
- стандардних библиотечких извештаја претраживања,
- UNIMARC формата [3] и
- XML докумената [4].

Систем БИСИС [5] обухвата два механизма за извођење узајамне каталогизације. Једна могућност је путем веб интерфејса за корисничко претраживање [6], а друга је употребом сервера за узајамну каталогизацију.

6.1. Сервер за узајамну каталогизацију

6.1.1. Рад са удаљеном базом библиографских записа

Сервер за узајамну каталогизацију је апликација која се инсталира у оквиру БИСИС система и која омогућује свим пријављеним клијентским апликацијама приступ локалној бази података. Слика 6.1 приказује дијаграм случаја коришћења сервера за узајамну каталогизацију. Клијентска апликација користи сервер за узајамну каталогизацију да би вршила претрагу удаљене базе (случај коришћења *Pretraga udaljene baze zapisa*) и да би преузимала записе са удаљене базе (случај коришћења *Preuzimanje zapisa*). Остале акције (креирање, брисање и измена локалних записа) се извршавају над локалним записима клијентске апликације (случај коришћења *Rad sa lokalnim zapisima*).

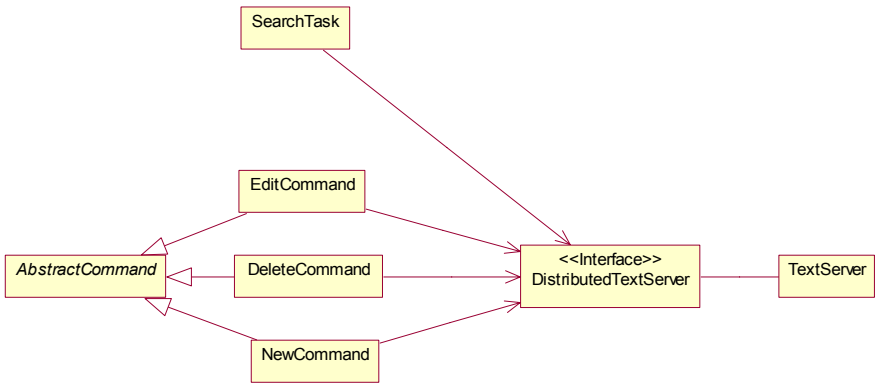


Слика 6.1. Дијаграм случајева коришћења сервера за узајамну каталогизацију

6.1.2. Архитектура серверске апликације

Сервер за узајамну каталогизацију заправо представља колекцију *TextServer* објеката, који представљају инстанце текст сервера за претрагу и индексирање библиографских записа [7], а који се уступају клијентским системима на употребу. Из безбедносних разлога, *TextServer* објекти омогућују само претрагу и преузимање записа. Измена и брисање нису дозвољени. Дијаграм класа ове архитектуре дат је на слици 6.2.

Клијентске апликације приликом пријављивања на сервер за узајамну каталогизацију добијају на коришћење један објекат класе *TextServer* и по завршетку рада га враћају у складиште (техника *Resource Pooling*). То значи да локалној бази приступају локални корисници (библиотекари) са једне стране, и удаљени корисници, преко објеката класе *TextServer*.

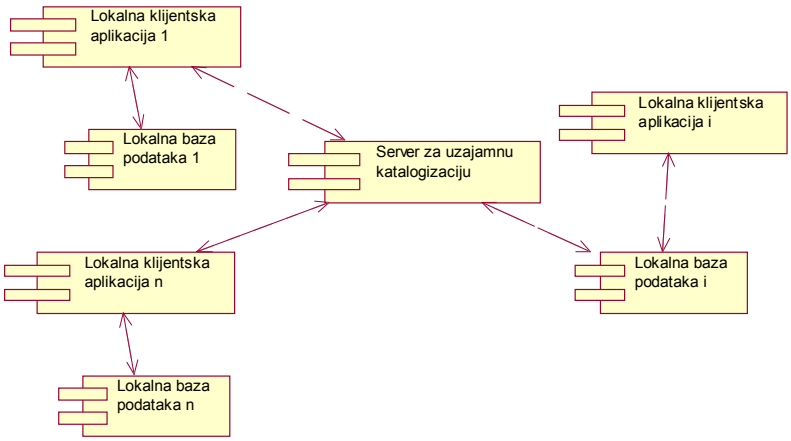


Слика 6.2. Дијаграм класа серверске апликације

Слика 6.2 приказује дијаграм класа који илуструје употребу удаљеног објекта класе *TextServer*. Класа *SearchTask*, као и све класе које наслеђују класу *AbstractCommand* и раде са записима (класе *EditCommand*, *DeleteCommand* и *NewCommand*), не виде класу *TextServer* већ само интерфејс *DistributedTextServer*. Објекти класе *TextServer* се заправо налазе на серверу за узајамну каталогизацију и на њему се извршавају. Параметри метода се шаљу са клијентске стране, а резултати рада се прослеђују клијентској страни. Параметри и резултати рада се прослеђују преко рачунарске мреже, односно Интернета. Потребно је само одабрати систем за удаљено извршавање метода. Развијене су две верзије сервера за узајамну каталогизацију: RMI (*Remote Method Invocation*) технологијом [8] и EJB (*Enterprise Java Beans*) технологијом [9].

Слика 6.3 приказује дијаграм развоја БИСИС система са сервером за узајамну каталогизацију. На њој је приказано *n* сервера и клијената од којих је на *i*-том постављен сервер за узајамну каталогизацију. Све клијентске апликације могу да приступе бази *i*-тог сервера преко своје инстанце објекта класе *TextServer* .

Ако је потребно да неки од система уступи другим системима своје записе, довољно је да се на њему инсталира сервер за узајамну каталогизацију.



Слика 6.3. Архитектура БИСИС система са сервером за узајамну каталогизацију

6.2. Имплементација

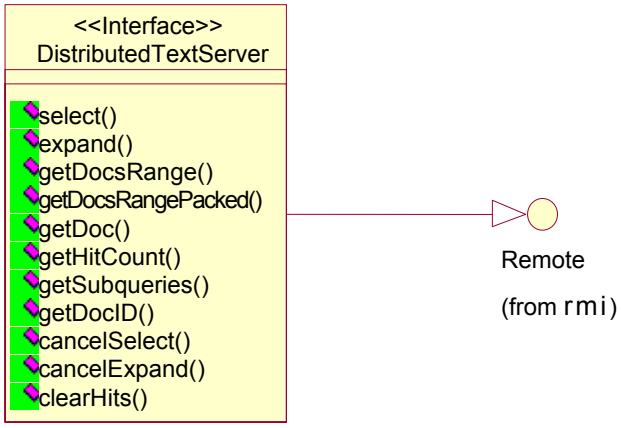
6.2.1. Имплементација RMI технологијом

Имплементација сервера за узајамну каталогизацију је изведена у програмском језику Java. Осим имплементације серверске апликације, проширен је и скуп библиотечких команди клијентске апликације.

За комуникацију између клијента и сервера коришћена је техника *Remote Method Invocation* (RMI), која омогућује да се на серверској страни извршава програмски кôд а да се на клијентској страни само прихватају резултати рада. Сервер је имплементиран као RMI сервер, који прослеђује објекте класе *TextServer* клијентима. Да би се обезбедило да више корисника могу симултано да затраже и добију објекте класе *TextServer*, имплементирана је техника *Resource Pooling*, која омогућује да се из складишта објеката корисницима додељују појединачни објекти, које ови по завршетку рада враћају у складиште. Заједнички интерфејс и за клијентску и за серверску страну је интерфејс *DistributedTextServer*, где се наводе све методе класе *TextServer* које ће бити видљиве на клијентској страни, као што је приказано на слици 6.4.

Да би клијент приступио серверу за узајамну каталогизацију, потребно је да се унесе следећа команда:

```
connect adresa_servera_za_uzajamnu_katalogizaciju
```



Слика 6.4. Интерфејс *DistributedTextServer*

Овом наредбом, клијент се спаја на сервер за узајамну каталогизацију и сво претраживање се преусмерава на њега. По успешном спајању на сервер, ускраћује се право извршења наредби `edit` и `delete`, јер модификовање и брисање записа није дозвољено на удаљеном систему. Записи се могу преузимати наредбом:

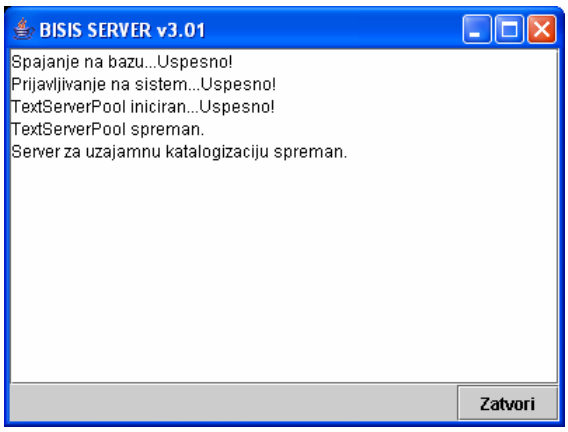
```
new x
```

где је *x* број поготка који је пронађен на удаљеном систему.

Сервер узајамне каталогизације омогућује библиотекарима преузимање записа из других база, потребне модификације и меморисање модификованих записа у локалну базу. Сервер је имплементиран коришћењем RMI и *Resource Pooling* технологија. Пошто се сви приступи удаљеном систему извршавају на серверској страни, саобраћај између клијента и сервера је сведен на минимум. Прозор RMI сервера за узајамну каталогизацију је приказан на слици 6.5.

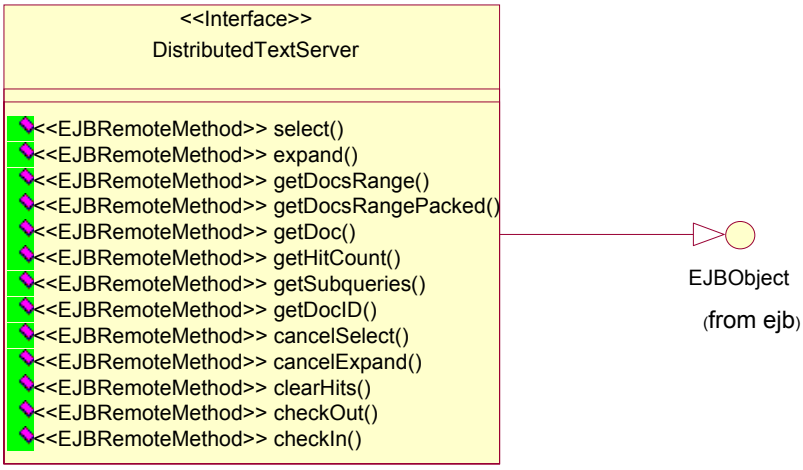
6.2.2. Имплементација у EJB технологији

Имплементација сервера за узајамну каталогизацију у EJB технологији се ослања на EJB апликационе сервере. За потребе узајамне каталогизације, имплементирана је EJB компонента за рад са записима (класа *EJBTextServer*), а постојећа клијентска апликација је прилагођена за рад са EJB оријентисаним сервером за узајамну каталогизацију.



Слика 6.5. Прозор RMI сервера за узајамну каталогизацију

Апликациони сервер омогућује клијентским апликацијама да користе EJB компоненту, чији се код извршава на серверској страни, а на клијентској страни се само прихватају резултати рада. Заједнички интерфејс и за клијентску и за серверску страну је интерфејс *DistributedTextServer*, где се наводе све методе класе *EJBTextServer* које ће бити видљиве на клијентској страни, као што је приказано на слици 6.6. Овај интерфејс представља *Remote* интерфејс у EJB терминологији.



Слика 6.6. Интерфејс `DistributedTextServer`

Да би се одабрао одговарајући тип имплементације (RMI или EJB), користи се команда *server*, којом се бира тип сервера за узајамну каталогиза-

цију. Након одабира типа сервера за узајамну каталогизацију, клијентска апликација се спаја на сервер командом:

```
connect adresa_servera_za_uzajamnu_katalogizaciju
```

Овом наредбом, клијент се спаја на сервер за узајамну каталогизацију и претраживање се преусмерава на њега. По успешном спајању на сервер, ускраћује се право извршења наредби *edit* и *delete*, јер модификовање и брисање записа није дозвољено на удаљеном систему. Записи се могу преузимати наредбом:

```
new x
```

где је *x* број поготка који је пронађен на удаљеном систему.

EJB сервер за узајамну каталогизацију представља унапређење у смислу побољшања перформанси и безбедности имплементацијом у EJB технологији. Ова технологија омогућава једноставну имплементацију ограничења приступа и провере корисника, као и напредније технике рада са корисничким апликацијама, као што су *Load Balancing* и *Session Failover* који су неопходни за системе са великим бројем корисника. Овај систем не зависи од конкретног апликационог сервера, већ се може извршавати на свим апликационим серверима који подржавају EJB спецификацију. На слици 6.7 је приказан прозор апликационог сервера на којем је подигнут EJB сервер за узајамну каталогизацију.

Слика 6.7. Прозор EJB сервера за узајамну каталогизацију

6.3. Узајамна каталогизација употребом агентске технологије

Клијентске апликације су у стању да претражују и преузимају записе само из оних удаљених база на којима је инсталиран сервер за узајамну каталогизацију. Међутим, да би пронашли неки запис унутар библиотечке мреже, библиотекари пре свега морају да знају на којим је све серверима библиотечке мреже подигнут сервер за узајамну каталогизацију. Затим, библиотекари морају да се пријаве на сваки од познатих сервера и да тамо задају исти упит.

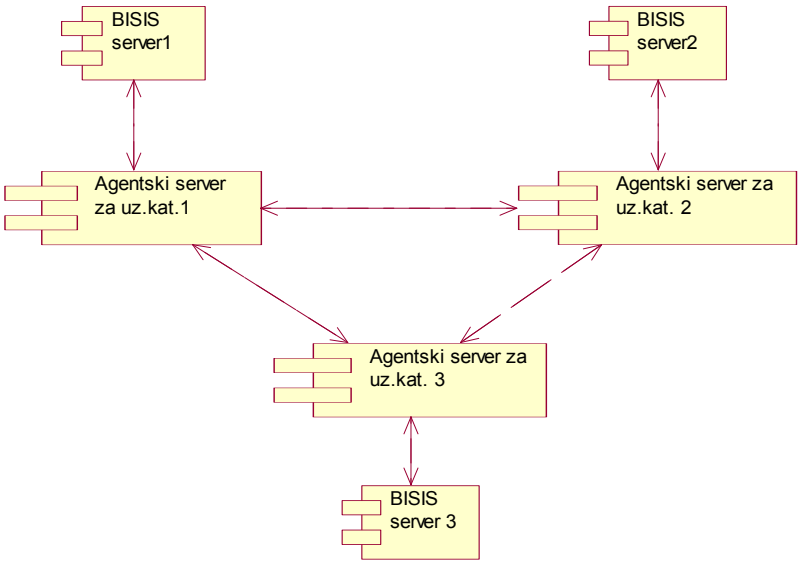
Ако неки запис постоји на више сервера, потребно је пронаћи најквалитетнији и њега преузети. Одређивање квалитета се своди на појединачно преузимање записа, његов преглед и оцењивање од стане библиотекара.

Из наведеног се види да сервер за узајамну каталогизацију пружа огромне могућности за библиотекаре, али и да је потребно систем унапредити на такав начин да се процес узајамне каталогизације поједностави и аутоматизује.

За овакве потребе, као предлог решења узајамне каталогизације, развијен је *агентски сервер за узајамну каталогизацију* [10]. На тај начин, илустровано је претраживање библиографских записа у библиотечкој мрежи и оцена квалитета записа. Овај агентски сервер за узајамну каталогизацију је имплементиран у оквиру Проширивог агентског окружења базираног на Java технологији [11]. Мрежа библиотечких сервера са агентским серверима за узајамну каталогизацију представља окружење у којем се врши узајамна каталогизација употребом агената. Пример архитектуре мреже од три агентска сервера за узајамну каталогизацију је приказан на слици 6.8.

На слици 6.8 је приказан пример мреже од три агентска сервера. Додавање новог сервера у мрежу је аутоматизовано поступком регистрације агентског центра у коме је сервер имплементиран. Овим се нови агентски сервер прикључује у мрежу без додатних процедура.

Сваки агентски сервер је репрезентован скупом библиотечких агената којима се делегирају библиотечки послови. Ови агенти су у стању да приступе библиотечким записима преко библиотечког сервиса за приступ библиотечким серверима. На овај начин, агентски сервер је директно повезан са библиотечким сервером. Агентска окружења унутар којих се извршавају библиотечки агенти формирају мрежу која омогућује овим агентима да прелазе из једног агентског сервера у други, да проналазе одговарајуће агенте на другим серверима и да међусобно комуницирају.



Слика 6.8. Дијаграм компоненти мреже агентских сервера
за узајамну каталогизацију

Библиотечки агенти су у стању да пруже следеће услуге: претраживање записа по библиотечкој мрежи и оцену квалитета записа.

Претраживање записа по библиотечкој мрежи се ослања на мобилност агената. Агент који се ангажује да пронађе запис обилази библиотечку мрежу и на сваком чвору мреже претражује базу записа употребом сервиса за рад са БИСИС системом. Овај сервис омогућује агентима претрагу записа, њихово преузимање и оцењивање.

Оцена квалитета записа се такође заснива на агентима. На сваком серверу за узајамну каталогизацију постоје специјализовани агенти за оцену квалитета који ће бити активирани KQML (*Knowledge Query and Manipulation Language*) поруком која садржи идентификатор записа чији се квалитет оцењује. Резултат рада ових агената се путем KQML поруке прослеђује клијентима који су их ангажовали.

6.4. Закључак

Узајамна каталогизација омогућује претраживање и преузимање записа из удаљених база библиографске грађе. Реализација узајамне каталогизације

подразумева употребу сервера за узајамну каталогизацију који су у могућности да контролисано опслужују већи број клијената. Један предлог даљег побољшања концепта узајамне каталогизације подразумева агентску технологију. Предлог агентског сервера за узајамну каталогизацију обухвата агентско претраживање библиотечке мреже и агентско оцењивање квалитета записа. Предложено решење за узајамну каталогизацију (преузимање библиографских записа) може се користити и у имплементацији стандарда за преузимање библиографских записа као што су Z39.50 и OAI-PMH (*Open Archive Initiative Protocol for Metadata Harvesting*).

Референце

[1] D. Todorić, D. Surla. *Shared Cataloguing Server's functions*. Zbornik radova YUINFO'98, 1998., str. 262-272.

[2] D. Todorić. *Uzajamna katalogizacija i korisničko pretraživanje bibliografskih podataka*. Magistarska teza, Prirodno-matematički fakultet, Novi Sad, 1998.

[3] D. Todorić, T. Vulić. *Data structure of UNIMARC standard*. Proceedings of the X Conference on Applied Mathematics PRIM'95, Budva 1995, pp. 309-314.

[4] M. Zeremski, D. Surla. *Modeliranje bibliografskih zapisa pomoću XML Schema jezika*. Zbornik radova sa XXVIII Jugoslovenskog simpozijuma o operacionim istraživanjima SYM-OP-IS 2001, Beograd 2001. str. 265-268.

[5] D. Surla, Z. Konjović, B. Milosavljević, M. Vidaković. *Bibliotečki informacioni sistem BISIS ver. 3.01*. Deveta međunarodna konferencija "Informatika u obrazovanju, kvalitet i nove informacione tehnologije", Zrenjanin 2000., str 494-504.

[6] B. Milosavljević. *Uzajamna katalogizacija: preuzimanje bibliografskih zapisa*. Infoteka, 1 (2000) 1, str. 19-23.

[7] Б. Милосављевић. *Текст сервер UNIMARC записа*. Магистарска теза, Факултет техничких наука, Нови Сад, 1999. http://diglib.ns.ac.yu/ndltd/docs/set1/ndltd6/textsrv.pdf

[8] M. Vidaković, Z. Konjović. *Server za uzajamnu katalogizaciju u bibliotečkom informacionom sistemu BISIS*. YUInfo 2001, Kopaonik 2001.

[9] M. Vidaković, B. Milosavljević. *Implementacija servera za uzajamnu katalogizaciju u EJB tehnologiji*. YUInfo 2002, Kopaonik 2002.

[10] M. Vidaković, Z. Konjović. *One Implementation of Virtual Library Based on Agent Technology*. Proceedings of the 4th Conference on Informatics and Information Technology, Bitola, 11-14 of December, 2003, (in print).

[11] M. Vidaković. *Proširivo agentsko okruženje bazirano na Java tehnologiji*. Doktorska distertacija, Fakultet tehničkih nauka, 2003. http://diglib.ns.ac.yu/ndltd/docs/set2/ndltd344/MinjaDoktorat.pdf

Поглавље 7

Извештавање о библиографској грађи

Т. Зубић, М. Рацковић

Податке о библиографској грађи, ускладиштене у базама библиографске грађе, потребно је на адекватан начин презентовати крајњим корисницима и библиотекарима. Сегмент библиотечког система који се бави тим аспектом библиотечког пословања назива се библиотечко извештавање.

При формирању библиотечких извештаја треба узети у обзир с једне стране, све новине које доносе савремене информационе технологије, као што су нови начини чувања података о библиотечкој грађи, као и нови начини приступа и презентовања тих података, глобално повезивање путем комуникационих мрежа и Интернета, а с друге, библиотечку праксу проистеклу из вишевековног искуства у каталогизацији и класификацији библиотечке грађе.

Систем БИСИС вер. 3 као Интернет/Интранет апликација, обезбеђује библиотекарима у оквиру Интранета извештаје које захтева пракса (извештаји о стању библиографског записа, каталошки листићи, спискови књига, инвентарна књига), а корисницима преко Интернета претраживање и добијање информација – извештаја о фонду библиотеке. Детаљно упутство за коришћење система БИСИС вер. 3 дато је у [1].

7.1. Библиотечки извештаји

Библиотечки извештаји могу се оквирно груписати на извештаје намењене корисницима библиотеке и извештаје намењене библиотекарима [2].

Корисници библиотеке извештаје о фонду библиотеке у систему БИСИС добијају преко Интернета и њима су најинтересантнији типови извештаја који на јасан начин описују публикацију и идентификују је као физичку јединицу у библиотечком фонду. У том смислу, за кориснике су обезбеђени кратки и проширени исписи резултата претраживања, а отворена

је могућност додавања и других типова извештаја. Кориснички извештаји детаљније су описани у поглављу 9.

Друга група извештаја подржава библиотечко пословање у појединим фазама и може се груписати на [3]:

- извештаје о стандардима,
- извештаје о библиографској грађи,
- извештаје за потребе служби за рад са корисницима и
- извештаје о раду библиотеке.

Извештаји о стандардима подразумевају извештаје о елементима структуре библиотечких стандарда (UNIMARC формата, радног окружења библиотекара). Они обухватају извештаје о, на пример, пољима и потпољима UNIMARC формата, екстерним и интерним шифарницима, типовима публикација, пољима и потпољима нивоа обраде и нивоа обавезности, и слично. Они су детаљније описани у поглављима 2 и 3.

Извештаји о библиографској грађи подразумевају извештаје о обрађеним публикацијама у базама библиографске грађе који су потребни библиотекарима у њиховом пословању и биће детаљније описани у даљем тексту.

Извештаји за потребе служби за рад са корисницима обухватају извештаје који су потребни приликом издавања библиотечке грађе, односно, извештаје о коришћењу фонда библиотеке по различитим критеријумима. Детаљније су описани у поглављу 10.

Извештаји о раду библиотеке подразумевају различите статистичке прегледе о раду библиотекара или саме библиотеке.

7.1.1. Извештаји о библиографској грађи

Извештаји о библиографској грађи обухватају велику групу извештаја:

- каталошки и CIP листићи,
- библиографије, билтени, каталози и остале секундарне публикације,
- инвентарне књиге,
- исписи за потребе набавке, размене и слања обавезног примерка,
- исписи за потребе координације набавке,
- исписи за потребе међубиблиотечке позајмице,
- остали исписи (спискови предметних одредница, и сл.).

Већина ових извештаја садржи у себи одређени опис публикација у библиотеци (односно, базама библиотечке грађе). Сваки опис публикације мора да јединствено идентификује публикацију у библиотечком фонду. У зависности од врсте извештаја, идентификација публикације врши се на основу одреднице, каталошког описа, сигнатуре и инвентарског броја. *Одредница* представља скуп речи које у зависности од врсте извештаја одређују место публикације у библиотечким каталозима. *Каталошки опис* садржи опис публикације који је довољан да се публикација идентификује. *Сигнатура* одређује топографски смештај публикације у библиотечком фонду, а *инвентарски број* је број који је јединствено додељен публикацији у одређеној библиотеци. Сви ови елементи описа публикације израђују се на основу правилника и приручника за израду библиотечких каталога [4, 5] који се заснивају на ISBD (*International Standard Bibliographic Description*) стандарду [6].

С обзиром да је UNIMARC формат [7] прихваћен као стандард за опис публикација у базама библиографске грађе система БИСИС, приликом извештавања врши се конверзија UNIMARC записа у формат каталошког описа описаног ISBD стандардом. Пошто је UNIMARC формат компатибилан са ISBD стандардима пресликавање је углавном једнозначно.

И поред стандардизације, приликом израде библиотечких извештаја потребно је обратити пажњу на различите облике истих врста извештаја у зависности од величине, намене и врсте библиотеке која их користи.

7.1.2. ISBD стандард

ISBD (*International Standard Bibliographic Description*) представља групу међународних стандарда за библиографски (каталошки) опис различитих типова публикација. Основни стандард је ISBD(G) – *General International Standard Bibliographic Description* – општи међународни стандардни библиографски опис који је основа свих специјализованих стандарда. Специјализовани стандарди су:

- ISBD(M) *International Standard Bibliographic Description for Monographic Publications* – међународни стандардни библиографски опис монографских публикација који одређује елементе за опис и идентификацију монографских публикација;
- ISBD(S) – међународни стандардни библиографски опис серијских публикација;
- ISBD(A) – међународни стандардни библиографски опис раритета;
- ISBD(NBN) – међународни стандардни библиографски опис некњижног библиотечког материјала;

- ISBD(CM) - међународни стандардни библиографски опис картографског материјала (мапа, атласа);
- ISBD(CF) - међународни стандардни библиографски опис рачунарских датотека;
- ISBD(PM) - међународни стандардни библиографски опис музикалија;
- ISBD(ER) - међународни стандардни библиографски опис електронских докумената.

Основне карактеристике ISBD стандарда су [4]:

- прецизно утврђен редослед библиографских података,
- примена специфичног интерпункцијског система и
- преузимање података из утврђених извора.

Елементи библиографског описа по ISBD стандарду груписани су у подручја чији су редослед, садржај, језик и писмо прецизно одређени.

Редослед библиографских података дефинише ***структуру*** библиографског описа.

Под термином *интерпункција* подразумева се унапред дефинисан специјални знак или група знакова који у испису претходе или прате елементе описа публикације. Пошто је UNIMARC формат креиран на основу ISBD стандарда, постоји пресликавање између елемената описа публикације дефинисаних ISBD-ом и поља, односно потпоља UNIMARC-а. То значи да се приликом извештавања из база библиографске грађе, заснованих на UNIMARC формату врши пресликавање садржаја поља и потпоља UNIMARC записа на елементе библиографског описа по ISBD-у. Пошто је за сваки елемент описа по ISBD-у везана одређена интерпункција, и садржају поља и потпоља који се јављају у испису претходи дефинисана интерпункција.

Поред структуре и интерпункције, за потпуну дефиницију библиографског описа потребно је утврдити и његову ***форму***. Под формом елемента библиографског описа подразумевамо величину, маргине, проред, начин растављања речи, начин прелома, поравнање текста, писмо, фонт, итд.

7.1.3. Каталошки листићи и други извештаји

Каталошки листићи (каталошке јединице) представљају основну врсту извештаја о библиографској грађи. Садрже све елементе библиографског описа: сигнатуру, једну или више одредница, каталошки опис и инвентарски број. У зависности од публикације коју описују могу бити каталошки листићи

за монографије, серијске публикације, некњижну грађу итд. На основу својих задатака и обима података које садрже, каталошке јединице могу бити: *главне, јединствене и помоћне.*

У наставку је дат пример пресликавања UNIMARC записа на једну врсту каталошких листића – споредни предметни каталошки листић за монографије.

UNIMARC формат дефинише структуру за библиографски опис библиотечке јединице. Библиографски опис једне библиотечке јединице формиран по UNIMARC формату називамо *библиографски запис*. Скуп библиографских записа чини *библиографску грађу,* односно, базу података библиотечке грађе.

```
$000  #a00101#b1996101719980518#cBIS.IM::GORDANA
        #dBIBFF.FF::GORDANA#e0#f0
$001  #7ba#ac#ba#cm#d0
$010  #a86-387-0372-4#d70
$100  #bd#c1993#hscr
$1010 #ascr
$102  #ayug#bsr
$105  #aa#bf
$200  #aLeksikon hispanoameričke književnosti#fLjiljana Pavlović-
        Samurović#g[indekse priredila Vesna Dickov]
$210  #aBeograd#cSavremena administracija#d1993
$215  #a871 str. #cilustr. #d25 cm.
$320  #aBibliografija : str. 791-795
$320  #aIndeks termina#aIndeks imena.
$6061 #aLatinsko-američka književnost#wLeksikoni
$675  #a860(7/8)(031) #b860
$700 1#aPavlović-Samurović#bLjiljana
$70201#4906#aDickov#bVesna
$900 6#aSamurović#bLjiljana Pavlović-#zPavlović-Samurović, Ljiljana
$992  #bvig9610
$9960 #dlC\n18583#f010018583
```

Слика 7.1. Низ поља једног библиографског записа

Сваки запис састоји се од *идентификатора записа* и низа *поља.* На слици 7.1 приказан је пример низа поља једног библиографског записа. Поље се састоји од *ознаке поља* (на слици $), *назива поља* и низа *потпоља*, а придружена су му и два *индикатора*. Потпоље се састоји од *ознаке потпоља* (на слици #), *назива потпоља* и *садржаја потпоља*. Садржаји потпоља пред-

стављају податке о публикацији која се обрађује. Значење податка јединствено је идентификовано пољем, индикаторима и потпољем чији је садржај.

На слици 7.2 приказана су подручја библиографског описа за споредни предметни каталошки листић за монографије по ISBD-у.

СИГНАТУРА

ПРЕДМЕТНА ОДРЕДНИЦА
ОДРЕДНИЦА
ГЛАВНИ ОПИС

НАПОМЕНЕ
ИСБН БРОЈ
- - ПРИЛОЗИ

АРАПСКИ ТРЕЈСИНГ
РИМСКИ ТРЕЈСИНГ
ПРЕДМЕТНИ ТРЕЈСИНГ
UDC БРОЈ

ИД БРОЈ

ИНВЕНТАРСКИ БРОЈ

Слика 7.2. Подручја библиографског описа на споредној, предметној каталошкој јединици

Каталошки листић формиран на основу наведеног UNIMARC записа приказан је на слици 7.3. Сваком подручју описа придружени су одговарајући подаци из UNIMARC записа са слике 7.1.

Сложени типови извештаја о библиографској грађи као што су библиографије или инвентарске књиге састоје се од појединих елемената каталошког описа или самих каталошких јединица сортираних по одређеном критеријуму. На слици 7.4 приказана је инвентарска књига која се састоји од каталошких описа публикација сортираних по инвентарском броју.

7.2. Стуктура извештаја о библиографској грађи

Да би се дефинисао модел извештаја о библиографској грађи било је потребно извршити њихову анализу, као и анализу структуре UNIMARC записа на основу кога се извештаји формирају [3], како би се установило одговарајуће пресликавање записа на подручја библиографског (каталошког) описа и остале елементе извештаја.

C - 18583

LATINSKO-američka književnost – Leksikoni

SAMUROVIĆ, Ljiljana Pavlović-

Leksikon hispanoameričke književnost / Ljiljana Pavlović-Samurović ; [indekse priredila Vesna Dickov]. – Beograd : Savremena administracija, 1993. - 871 str. : ilustr.; 25 cm.

Bibliografija : str. 791-795. - Indeks termina - Indeks imena.
ISBN 86-387-0372-4

I Samurović, Ljiljana Pavlović- V. Pavlović-Samurović, Ljiljana
a) LATINSKO-američka književnost – Leksikoni

Слика 7.3. Пример споредног предметног каталошког листића за монографије

INVENTARNA KNJIGA za godinu 1995 Str. 5

Inv.broj	Datum	Signatura	Opis jedinice	Dobavljač	Cena/del.br.	ID/Napomene
000774369	09.08.	II 423540/1	PRIRUČNIK za polaganje stručnog ispita za turističkog vodiča. - Beograd : Viša turistička škola, 1995 Ćir.	Obavezni primerak NBS V-1995		93301255
000774370	10.08	II 423488	PAŠIĆ, Milutin: Autori i dela. - Požega : Svitak, 1995 Ćir.	Obavezni primerak NBS V-1995		93338375
000774371	10.08.	II 423482	PETROVIĆ, Nada: Dodiri. - Kragijevac : Zmaj, 1995 Ćir.	Obavezni primerak NBS V-1995		93338887
000774372	10.08.	II 423483	KONTARIS, Antoni M.: Besede na pravoslavnim opelima. - Kragujevac : Kalenić, 1995 Ćir.	Obavezni primerak NBS V-1995		93339143
000774373	10.08.	II 423481	EIĆ, Ljubica M.: San i posle sna. - Beograd : Zajednica književnih klubova Srbije, 1995	Obavezni primerak NBS V-1995		93339911
000774374	10.08.	II423476	SLIJEPČEVIĆ, Ignjatije: Obala na mesečini. - Kruševac : Bagdala, 1995	Obavezni primerak NBS V-1995		93340423

Слика 7.4. Пример инвентарске књиге

Анализом извештаја о библиографској грађи дошло се до закључка да се извештаји састоје од подручја библиографског описа. За потребе моделирања ова подручја и његове елементе називамо заједничким именом *концепт.* Концепт чине други концепти или поља, односно, потпоља UNIMARC формата, а сама поља се састоје од потпоља, која опет могу садржавати потпотпоља, или њихов садржај замењује унапред дефинисан текст.

Дакле, структура извештаја је хијерархијска при чему најнижи ниво у хијерархији представљају подаци из потпоља, потпотпоља и предефинисан текст, док извештаји, каталошки листићи, концепти и поља представљају сложене елементе који се састоје од других елемената.

Приликом обраде сваког елемента структуре примењују се одређена *правила,* односно, морају да буду задовољени одређени услови. Под *правилом* подразумевамо низ акција које формирају садржај елемента, мењају неке особине елемента или одређују да ли ће се елемент појавити на извештају или не. У *особине елемента* убрајамо структуру за формирање његовог садржаја, интерпункцију, атрибуте исписа и писмо. У том смислу можемо извршити и груписање правила на:

- правила структуре,
- правила писма,
- правила за форматирање,
- правила интерпункције,
- правила за мењање садржаја,
- правила појаве у испису.

Над сваким елементом структуре извештаја може се применити више правила.

7.3. Модел извештаја о библиографској грађи

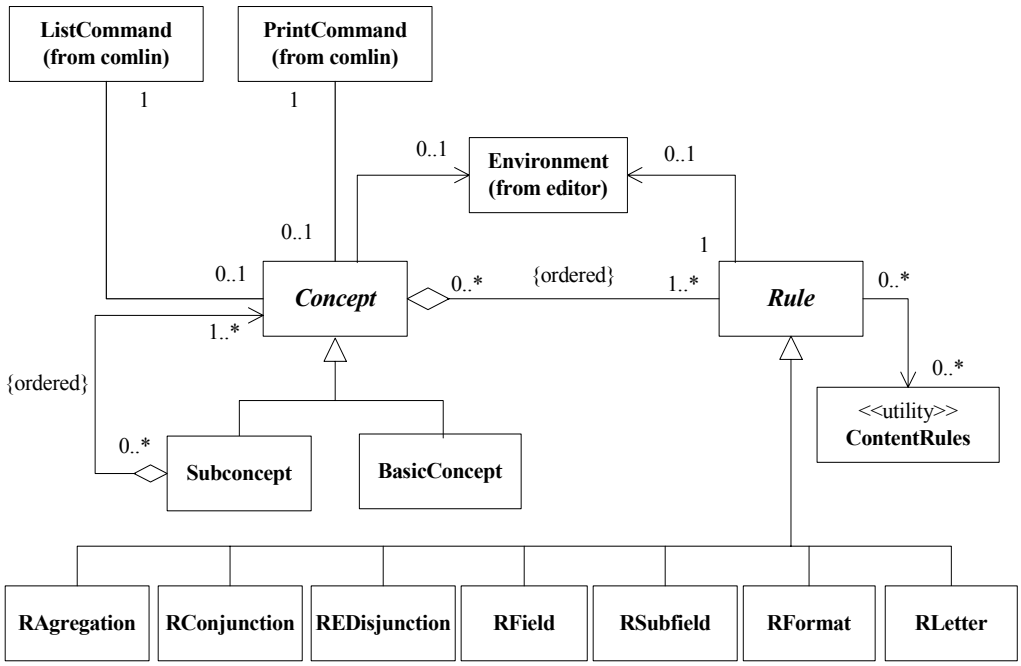
У складу са извршеном анализом извештаја о библиографској грађи, формиран је објектно – оријентисани модел коришћењем Обједињеног језика моделирања [3, 8, 9, 10]. На слици 7.5 приказан је дијаграм класа за модел структуре извештаја.

Хијерархијска и рекурзивна структура извештаја представљена је структуром класа [3]: *Concept*, *Subconcept* и *BasicConcept*. Основна идеја моделирања је увођење апстрактне класе *Concept*, која декларише интерфејс конкретних класа *Subconcept* и *BasicConcept*. Класа *Subconcept* представља сложене елементе структуре извештаја који се састоје од других сложених

или простих елемената. Ова веза је моделирана преко агрегације класе *Subconcept* са класом *Concept* и наслеђивања. Класа *BasicConcept* представља „лист“ у хијерархијској структури извештаја. Увођењем описане структуре класа постигнуто је следеће:

- уведена је хијерархија која се састоји од простих и сложених објеката, чиме је омогућено да прости елементи буду укомпоновани у сложене, који опет могу бити део других сложених елемената, итд.
- клијент структуре се упрошћава, пошто, због заједничког интерфејса, једнако третира и просте и сложене елементе структуре.

У анализи је истакнуто да је сваки елемент структуре повезан са одговарајућим правилима за његово формирање. Групе правила моделиране су на два начина у дијаграму класа структуре извештаја.



Слика 7.5. Дијаграм класа за модел структуре извештаја

Правила за измену садржаја могу се користити у оквиру других правила, па су представљена класом *ContentRules*, која је везом асоцијације повезана са апстрактном класом *Rule*.

Већина правила (правила структуре, правила везана за интерпункцију, правила појаве поља у испису и правила писма) могу се применити на различите елементе структуре и то у различитим међусобним комбинацијама. Због тога су она моделирана као класе учаурених алгоритама са заједничким интерфејсом, представљеним апстрактном класом *Rule*. Дакле, наведена правила су конкретне подкласе апстрактне класе *Rule*. Веза класа *Subconcept* и *BasicConcept* са класом *Rule* остварује се преко везе агрегације између класа *Concept* и *Rule*. Приликом иницијализације објеката класа *Subconcept* и *BasicConcept* врши се повезивање са одговарајућим подкласама класе *Rule*.

Правила нису специфицирана као методе класе *Concept*, односно класа *Subconcept* и *BasicConcept,* из неколико разлога:

- тиме би наведене класе постале много комплексније и теже за одржавање, пошто би подржавале више сличних метода које се углавном не користе истовремено,
- било би много теже додавати нове или мењати постојеће методе, поготово ако се има у виду хијерархијска структура наведених класа.

Дакле, одвајањем правила од структуре извештаја постиже се лакше одржавање како скупа правила, тако и структуре извештаја.

Класе *ListCommand* и *PrintCommand* из пакета *comlin* представљају команде *list* и *print* којима се покреће формирање извештаја о библиографској грађи. Ове класе на основу задатих параметара команди и резултата претраживања преузимају записе из базе библиографске грађе из којих се преузимају подаци за извештаје.

Класа *Environment* садржи инстанце UNIMARC формата, окружења библиотекара и друге, тако да омогућава преузимање података потребних при формирању извештаја.

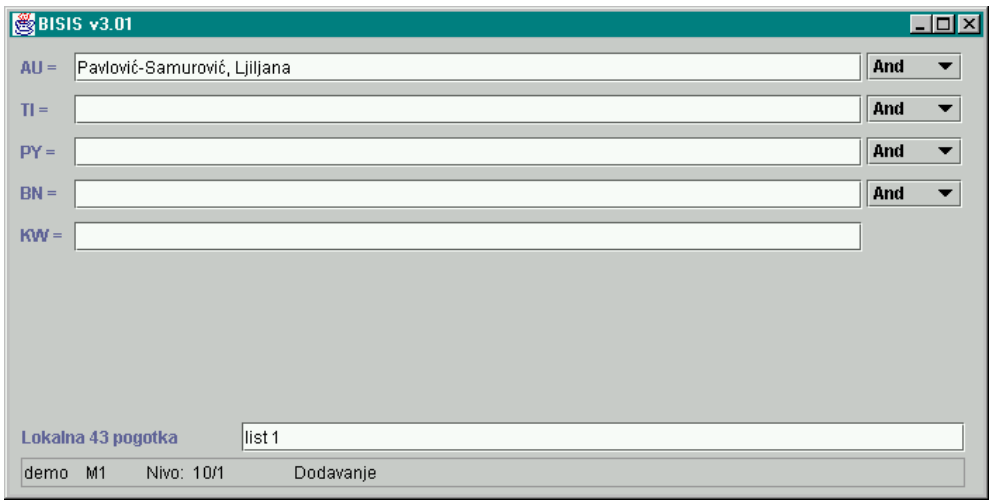
На основу описаног дијаграма генерисан је програмски код у програмском језику Java који је представља окосницу за имплементацију подсистема за извештавање БИСИС-а.

7.4. Имплементација

Систем БИСИС верзија 3 имплементиран је у Java окружењу. Основне акције при раду у БИСИС-у остварују се преко система команди. Команде везане за извештавање су *list*, *list?* и *print*.

Команда *list?* даје приказ свих типова извештаја о библиографској грађи који су дефинисани у систему. Команда *list?* нема параметре и може се позивати и приликом претраживања и приликом обраде записа.

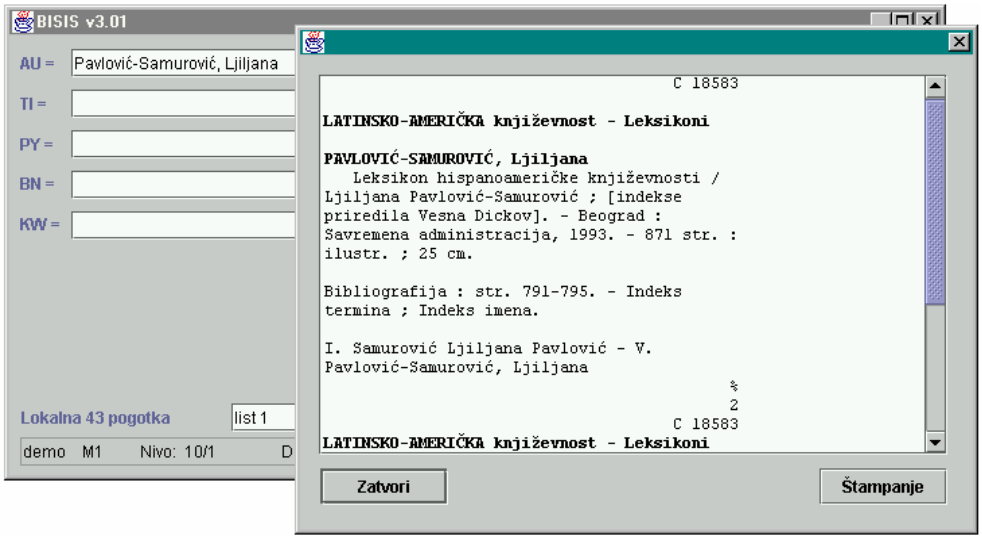
Процес извештавања покреће се позивом команде *list* из командне линије у оквиру прозора за претраживање или прозора за обраду записа.



Слика 7.6. Позив команде *list*

Извршавањем команде *list* отвара се прозор у коме се исписује извештај. Ова команда може се активирати након извршеног претраживања у бази података, или приликом обраде записа. Зависно од места позива, команда има различите параметре. Ако се позива након извршеног претраживања неопходно је навести редне бројеве погодака из скупа погодака који су резултат успешно извршеног претраживања. Ако се, пак, позива приликом обраде записа могуће је добити извештај (најчешће, каталошки листић) само за запис који се тренутно обрађује, тако да није потребно наводити бројеве погодака. Други параметар, тип извештаја, је опциони. Уколико се не наведе, извештај се формира на основу текуће вредности типа извештаја која је преузета из радног окружења библиотекара приликом иницијализације апликације, или је постављена од стране библиотекара у току рада.

На слици 7.6 приказан је позив команде *list* из прозора за претраживање за први погодак при претраживању и текућу вредност типа извештаја (у овом случају, предметног листића за монографије). На слици 7.7 приказан је резултат извршавања команде *list*: предметни листић за изабрану монографску публикацију.



Слика 7.7. Резултат извршавања команде *list*

Штампање извештаја реализује се притиском на дугме **Štampanje**. При том се отвара прозор за избор штампача и подешавање параметара штампања.

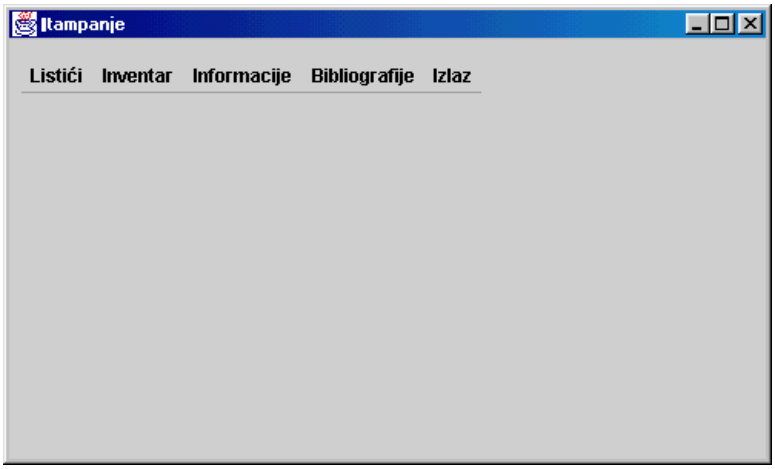
У систему БИСИС имплементирани су следећи типови каталошких листића: абецедни, топографски, упутни, као и споредни (предметни, ауторски, UDC) листићи за монографске публикације, серијске публикације, докторске дисертације, географске карте и некњижну грађу.

Такође, имплементирани су и следећи сложени извештаји: инвентарска књига, библиографија, билтен приновљених књига. Креирање ових извештаја врши се позивом команде *print* из командне линије у оквиру прозора за претраживање. Када се зада ова команда отвара се прозор приказан на слици 7.8 са менијем за избор жељеног извештаја.

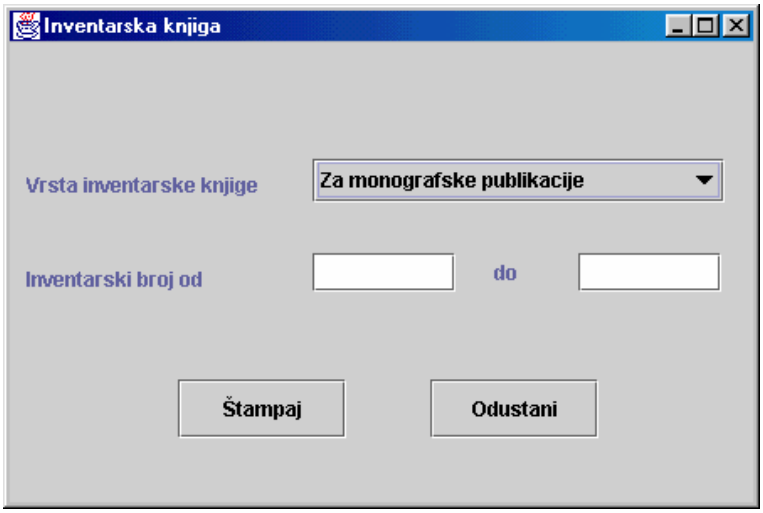
На пример, *Инвентарска књига* добија се тако што се у оквиру ставке менија **Inventar** бира ставка **Inventarska knjiga**. Након избора отвара се прозор у коме се врши избор врсте инвентарске књиге (нпр. за монографске публикације, за серијске публикације итд.) и избор интервала инвентарских бројева за инвентарску књигу (слика 7.9). Притиском на дугме **Štampaj** отвара се прозор са генерисаном одабраном инвентарском књигом. Штампање инвентарске књиге врши се притиском на дугме **Štampaj** у истом прозору.

На сличан начин имплементирани су и други сложени извештаји у БИСИС-у.

Коришћењем *Unicode* стандарда омогућено је приказивање различитих писама исписа (зависно од писма каталогизације и публикације), тако да су подржани и ћирилични и латинични каталози.



Слика 7.8. Мени за избор извештаја за штампање



Слика 7.9. Прозор за избор врсте инвентарске књиге и интервала инвентарских бројева

7.5. Закључак

Систем БИСИС вер. 3 као Интернет/Интранет апликација, обезбеђује библиотекарима у оквиру Интранета извештаје које захтева пракса (извештаји о стању библиографског записа, каталошки листићи, спискови књига, инвентарна књига), а корисницима преко Интернета претраживање и добијање информација – извештаја о фонду библиотеке.

Извештаји о библиографској грађи подразумевају извештаје о обрађеним публикацијама у базама библиографске грађе који су потребни библиотекарима у њиховом пословању.

Извештаји о библиографској грађи најчешће су описани ISBD стандардом. С обзиром да базе библиографске грађе у БИСИС-у подржавају UNIMARC формат, приликом извештавања врши се конверзија UNIMARC записа у формат библиографског описа дефинисаног ISBD-ом.

Структура извештаја о библиографској грађи је хијерархијска, при чему је за сваки елеменат хијерархије дефинисан извор података, интерпункција и форма приказа. Над елементима хијерархије – концептима, су дефинисана правила која формирају садржај елемента структуре, мењају неке његове особине или одређују да ли ће се елемент појавити на извештају или не.

Дефинисан је објектно оријентисани модел извештаја о библиографској грађи који моделира њихову хијерархијску структуру као и систем правила који се над том структуром примењују. Применом правила и структуре извештаја на конкретне податке из записа базе библиографске грађе, формира се извештај о библиографској грађи и презентује се кориснику.

Референце

[1] Д. Сурла, З. Коњовић, Б. Пуповац, Б. Милосављевић, М. Видаковић, Т. Тошић, Т. Зубић. *Упутство за коришћење библиотечког софтверског система БИСИС вер. 3.* Група за информационе технологије, Нови Сад, 2003.

[2] Т. Зубић, Г. Поповић-Бошковић. *Моделирање и имплементација извештаја о библиографској грађи*. ИНФОТЕКА: Часопис за информатику и библиотекарство, децембар 2000, год. 1, бр.2, Београд, 2000., стр. 59-66.

[3] Т. Вулић. *Извештавање и документовање у библиотечком информационом систему*. Магистарска теза, Природно-математички факултет, Нови Сад, 1997.

[4] С. Бабовић, Б. Јанчић, М. Булатовић. *Алфабетски каталог*. Народна библиотека Србије, Београд, 1991.

[5] Т. Вулић, Г. Билић. Р*етроспективна конверзија каталошких листића у формат YUMARC*. Институт за математику у Новом Саду, Едиција ”Рачунари у универзитетској пракси” бр. 5, Нови Сад, 1997.

[6] The International Federation of Library Associations and Institutions. http://www.ifla.org

[7] *UNIMARC Manual: bibliographic format / International Federation of Library Association and Institutions*. IFLA Universal Bibliographic Control and International MARC Programme, New Providence, London, 1994.

[8] G. Booch, J. Rumbaugh, I. Jacobson. *Unified Modeling Language, Version 1.3*. Rational Software Corporation, September 1999, http://www.rational.com

[9] Т. Вулић. *Спецификација захтева подсистема за извештавање библиотечког информационог система*. Зборник радова YU INFO’98, Копаоник, 1998., стр. 222-226.

[10] Т. Зубић, Д. Сурла. *Објектно оријентисано моделирање извештавања у библиотечком информационом систему*. CD ROM Зборник радова YU INFO’2000, Копаоник, 2000.

Поглавље 8

Текст сервер UNIMARC записа

Б. Милосављевић

Проблеми претраживања колекција текстуалних докумената разматрају се у оквиру области које се баве базама података (*databases*) и проналажењем информација (*information retrieval*). Текстуални документи се, у различитим системима за претраживање (текст серверима), третирају на различит начин. Један вид класификације текст сервера узима у обзир доступност оригиналног текста у електронском облику. Класификацијом према доступности оригиналног текста системи за претраживање су подељени у две групе:

- сервере који рукују оригиналним документима у електронском облику, процесирајући целокупан њихов садржај (*full text*) и
- сервере који рукују метаподацима, тј. подацима који на одговарајући начин репрезентују карактеристике оригиналног документа.

Мотивација за руковање метаподацима уместо целокупним садржајем докумената лежи у знатно мањим меморијским захтевима за складиштење метаподатака и захтевима за њихово процесирање. Са друге стране, како метаподаци представљају само посредничку репрезентацију садржаја датог оригиналним документом, *full text* системи су у могућности да избегну непрецизност или непотпуност описа докумената помоћу метаподатака рукујући комплетним садржајем оригиналних докумената.

Други вид класификације текст сервера узима у обзир начин формирања *индекса*, података који се директно користе приликом процеса претраге. У односу на начин формирања индекса системи се деле на:

- системе који користе ручно формиране индексе, тј. индексе чији садржај формирају квалификована лица и
- системе који користе аутоматски формиране индексе.

Текст сервери који користе ручно формиране индексе омогућавају уграђивање експертског знања у садржај индекса. Са друге стране, формира-

ње оваквих индекса подразумева учешће човека и није увек изводљиво или оправдано. Садржај аутоматски формираних индекса најчешће нема квалитет као код претходне групе јер је аутоматска екстракција семантике из текстуалних докумената изузетно сложен проблем.

Трећи начин класификације текст сервера односи се на модел докумената. На овај начин системи се деле на:

- системе који рукују неструктурираним документима и
- системе који рукују структуририраним документима.

Неструктурирани документи су представљени самим својим текстом сачињеним од низа слова. Структурирани документи су ентитети који имају унутрашњу структуру, код које елементи структуре садрже фрагменте текста. Модел структуре докумената може бити различит код оваквих система.

Библиотечки информациони системи за подршку пословања класичних библиотека рукују само репрезентацијама оригиналних докумената које чува библиотека. Садржај тих репрезентација формирају библиотекари. Репрезентације самих докумената су сложени ентитети са унутрашњом структуром. Са становишта претходно изнетих начина класификације система за претраживање текста, библиотечки информациони системи рукују метаподацима (уместо садржајем оригиналних докумената) који представљају ручно формирани индекс (садржај метаподатака, тј. описа оригиналних докумената креирају библиотекари) а сами метаподаци су структурирани текстуални документи.

UNIMARC (*Universal Machine Readable Cataloguing Format*) [1] је стандард који је првенствено намењен за међународну размену библиографских података у машински читљивом облику између националних библиографских институција. Он дефинише ознаке садржаја (поља, индикаторе и потпоља) које се користе за формирање библиографских записа и њихов логички и физички формат. Обухвата монографске, серијске и картографске публикације, музику, звучне записе, слике, видео материјале а оставља место и за додавање рачунарских датотека.

UNIMARC је замишљен као стандардни формат за размену библиографске грађе, и не прописује начин обраде и чувања грађе унутар појединих библиографских институција. Међутим, практично све институције које су прихватиле UNIMARC као стандард за размену користе UNIMARC или неку његову једноставну модификацију у оквиру својих система како би поједноставиле конверзију из UNIMARC-а у интерни формат и обрнуто. У Србији и Црној Гори се као стандард користи формат настао на основу формата

UNIMARC, формата COMARC [2] и захтева који су проистекли из пројекта БМ СНТИС [3]. У даљем тексту ће бити приказане основне карактеристике UNIMARC-а које су релевантне за потребе спецификације текст сервера за UNIMARC записе, а све што ће бити изнето односи се у потпуности и на YUMARC. Разлика између UNIMARC-а и YUMARC-а постоји само у формату физичког записа слога на рачунару.

У даљем тексту ће се под именом *UNIMARC формат* подразумевати формат којег прописује стандард, а под именом *UNIMARC запис* – једна библиографска јединица дата у UNIMARC формату, односно један слог.

Сваки UNIMARC запис се састоји од коначног скупа *поља*, при чему не морају сва поља бити дефинисана. Уколико садржај неког поља није дефинисан, оно се не наводи у запису. Поља се једнозначно одређују троцифреним бројем. Садржај поља чине два *индикатора* и коначан скуп *потпоља*. Потпоља су једнозначно одређена једним алфанумеричким знаком. Аналогно пољима, ако садржај неког потпоља није дефинисан, оно се не наводи у скупу потпоља одговарајућег поља. Садржај потпоља је текст. Садржај једног потпоља је основна јединица информације (*data element*) коју поседује UNIMARC запис. Индикатори су намењени да ближе опишу садржаје дате у потпољима. Разликују се по положају у запису (*први* и *други*) и сваки је представљен једним знаком.

Нека поља су *понављива*, тј. могу се наћи више пута у једном запису. Исто важи и за одређена потпоља у оквиру поља. Такође постоје поља која су *обавезна*, односно она која се морају наћи у запису. Слично томе постоје и обавезна потпоља у оквиру појединих поља. Детаљнији опис UNIMARC формата дат је и у Поглављу 2.

Претраживање библиографске грађе, тј. UNIMARC записа, не врши се коришћењем претходно описаних концепата поља и потпоља, већ префикса. Посматрано са овог аспекта, запис се састоји из скупа префикса. Сваки префикс је једнозначно одређен називом од два знака. Садржај префикса је текст. Један префикс може имати више садржаја у запису. Префикси који нису дефинисани се не наводе. Основна јединица информације у оваквом погледу на запис је један садржај једног префикса. Пресликавање концепата UNIMARC-а (поља и потпоља) на префиксе је прецизно дефинисано. Како ово мапирање није бијективно, инверзно мапирање није могуће. У књизи [4] је наведен садржај пресликавања поља и потпоља на префиксе.

Претраживање базе записа представља проналажење оних записа који задовољавају дати упит, тј. критеријум претраживања. Упит је *квалификован* уколико се наведе у ком префиксу се тражи дати израз. У другом случају

каже се да је упит *неквалификован*. Префикси су подељени у два дисјунктна скупа: *базни* и *додатни*. Базни префикси су они префикси који се претражују приликом обраде неквалификованих упита.

Појам *речи* унутар садржаја префикса се дефинише као низ знакова који је са обе стране ограничен *делимитерима речи*. Делимитери речи су знаци који нису саставни део речи, а налазе се непосредно поред дате речи у низу знакова који чине садржај префикса. Аналогно се дефинише и појам *реченице*: она представља низ речи који је ограничен *делимитерима реченице*. *Фраза* представља низ суседних речи.

Претраживање записа помоћу концепта префикса треба да има следеће могућности:

- претраживање се врши по речима или фразама, уз коришћење џокер-знакова;
- претраживање може бити квалификовано, тј. може се навести у ком префиксу се тражи дати садржај; уколико се квалификација не наведе, претраживање се врши по свим *базним* префиксима;
- од елементарних упита, који се састоје од једне тражене речи (терма), могуће је саставити сложене коришћењем логичких и близинских оператора;
- за сложене упите може се добити информација о резултатима претраге за подупите;
- могуће је добити листу свих префикса који почињу датим изразом.

За формирање упита користи се синтакса упитног језика *Dialog* [5]. Команде *Dialog* језика омогућавају претраживање, листање резултата, додатно руковање резултатима, подешавање корисничког окружења итд. За текст сервер UNIMARC записа усвојене су само команде SELECT и EXPAND које омогућавају унос упита за претраживање. Комплетан приказ *Dialog* корисничког окружења и команди интерпретера је дат у [5].

Спецификација команде SELECT дефинише и синтаксу упита. Упити се састоје од термова повезаних операторима. Елементарни упит се састоји од само једног терма. Терм представља реч која се тражи у записима. Речи се могу навести са џокер-знацима ‘?’ (замењује било који знак) и ‘*’ (замењује било који низ знакова). Следи пример једног елементарног квалификованог упита:

```
select AU=Petrović
```

Наредба EXPAND је намењена за листање садржаја текст индекса. Она приказује списак термова који се налазе у текст индексу. Наредба EXPAND

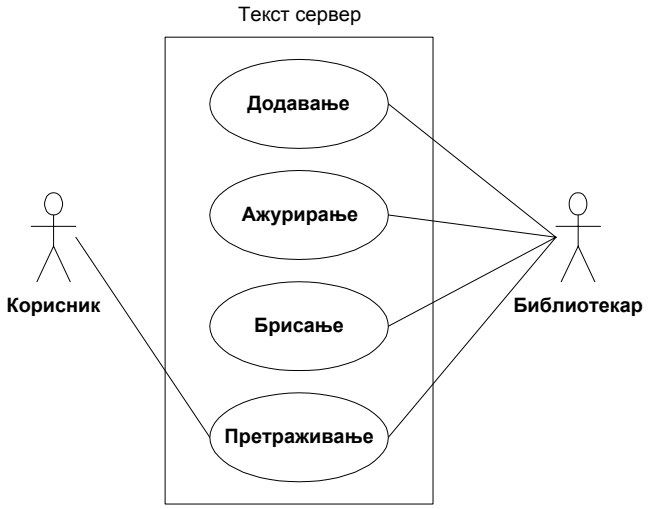
прима само један параметар који представља терм. Претрага се врши по свим префиксима, без обзира да ли су базни или додатни. Резултат извршавања наредбе EXPAND је списак са 12 ставки, где је трећа ставка дати терм (претходне две ставке се налазе "испред" у индексу; индекс је сортиран по абецеди). Следи пример једне EXPAND наредбе:

```
expand AU=Einstein
```

Развој текст сервера за UNIMARC записе текао је у више фаза. У раду [6] приказан је текст сервер намењен претраживању докумената датих префиксима имплементиран у *Oracle* окружењу и базиран на ускладиштеним процедурама *Oracle* система. Радови [7, 8] приказују текст сервер UNIMARC записа са подршком за претраживање помоћу префикса имплементиран помоћу *Oracle ConText* текст сервера. Садашња верзија текст сервера БИСИС-а детаљно је представљена у [9]. Одређени детаљи спецификације и имплементације овог система приказани су у [10, 11].

8.1. Функције текст сервера

Основна намена текст сервера је да омогући складиштење и претраживање библиотечке грађе која се обрађује и чува у датом библиотечком систему. Овакав текст сервер користе две класе корисника: библиотекари и (спољни) корисници. Библиотекари имају приступ свим функцијама текст сервера које укључују додавање нових докумената у индекс, ажурирање или уклањање постојећих докумената и претраживање. Библиотекари текст серверу приступају са клијената који се налазе у локалној рачунарској мрежи. Корисници имају право само да претражују индекс текст сервера, при чему им је омогућен приступ било из локалне рачунарске мреже (интранета), било са Интернета. Ови захтеви у погледу коришћења текст сервера представљени су дијаграмом случајева коришћења са слике 8.1.



Слика 8.1. Дијаграм случајева коришћења текст сервера

Документи којима рукује текст сервер су UNIMARC записи. Претраживање базе записа се врши коришћењем модификованог *Dialog* упитног језика приказаног у [3]. Основне карактеристике претраживања су следеће.

Расположиве су команде SELECT и EXPAND дефинисане у *Dialog*-у. Команда SELECT као параметар има текст који представља упит. Упит може да садржи следеће операторе:

- AND – израчунава пресек скупова погодака свог левог и десног операнда;
- OR – израчунава унију скупова погодака свог левог и десног операнда;
- NOT – израчунава скуповну разлику скупова погодака свог левог и десног операнда;
- [F] – захтева да се изрази дати левим и десним операндом налазе у истом префиксу;
- [S] – захтева да се изрази дати левим и десним операндом налазе у истој реченици;
- [Wn] – захтева да изрази дати левим и десним операндом буду удаљени највише *n* речи један од другог; ако се *n* не наведе, подразумева се 1 (суседне речи).

Приоритет оператора је следећи (оператори су наведени у опадајућем редоследу својих приоритета):

[Wn]

[S]

[F]

AND, OR, NOT

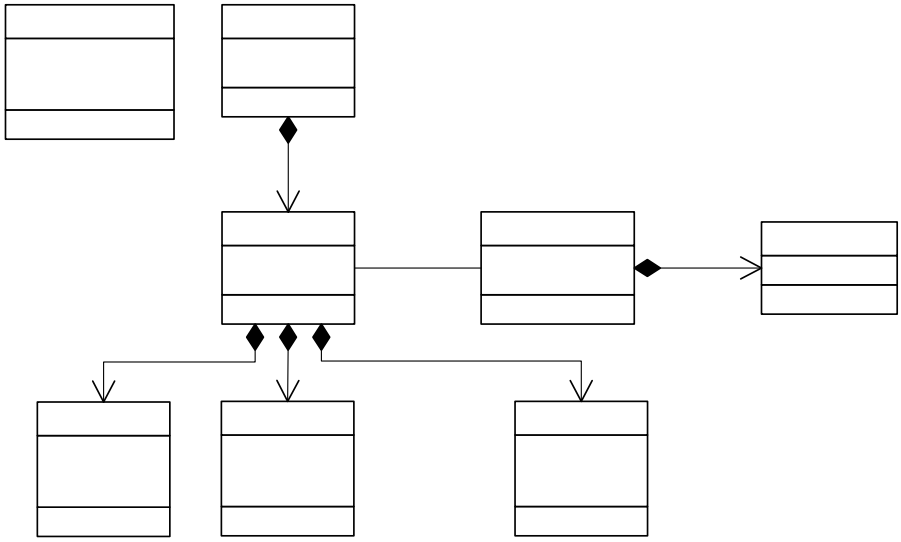
На редослед извршавања операција може се утицати коришћењем заграда на уобичајен начин.

За потребе физичког смештања података, текст сервер се ослања на неки од постојећих система за управљање базама података (СУБП). Подразумева се да је СУБП заснован на релационом моделу података који је данас најчешћи у употреби. Текст сервер је пројектован тако да у што већој мери буде независан од конкретног СУБП. Како сви комерцијално расположиви СУБП имају одређене специфичности које се огледају у начину њиховог коришћења, текст сервер има могућност да се те специфичности искористе са циљем постизања квалитетних перформанси.

8.2. Модел података текст сервера

Модел података текст сервера је предвиђен за имплементацију на неком од комерцијално доступних СУБП заснованих на релационом моделу података. Спецификација која ће овде бити изложена не зависи од конкретног СУБП. На слици 8.2 приказан је дијаграм класа који представља модел података текст сервера.

Класа *Document* представља документ којим рукује текст сервер, тј. UNIMARC запис. Њени атрибути су *docID* (интерни идентификатор документа) и *content* (садржај документа, тј. сам запис).



Слика 8.2. Модел података текст сервера

Класа *Prefix* представља префикс који се користи приликом претраживања. Њени атрибути су *prefName* (назив префикса, идентификатор) и *prefType* (тип префикса – базни или додатни, онако како су дефинисани у уводном одељку).

Класа *PrefixMap* је намењена за смештање пресликавања префикса на потпоља UNIMARC формата. Једини атрибут је *subfield* (потпоље које се пресликава на дати префикс).

Класа *PrefixContent* представља један садржај неког од префикса који припадају датом документу. Њени атрибути су *prefID* (идентификатор), и *content* (садржај префикса).

Класе *Pref_AB*, *Pref_AU* итд. до *Pref_Y2* представљају речи које припадају одговарајућем префиксу (AB, AU, итд. до Y2). Свака од њих има исте атрибуте: *wordPos* (редни број речи у префиксу), *sentPos* (редни број реченице у којој се налази дата реч) и *content* (садржај, тј. сама реч).

Класа *Delimiters* представља конфигурацију парсера текста. Њени атрибути су *configID* (идентификатор), *wordDelims* (карактери који представљају делимитере речи) и *sentDelims* (карактери који представљају делимитере реченице). Подразумева се да су делимитери реченице уједно и делимитери речи.

Како су данас најчешће у употреби СУБП засновани на релационом моделу података, претходно описани дијаграм класа је искоришћен за генерисање шеме релационе базе података. Шема је генерисана у оквиру одговарајућег CASE алата. Шеме релација добијене на овај начин приказане су на листингу 8.1. Курзивом су дати називи атрибута који улазе у састав примарног кључа.

```
DELIMITERS(config_id, word_delims, sent_delims)
PREFIXES(pref_name, pref_type)
PREFIX_MAP(pref_name, subfield)
DOCUMENTS(doc_id, content)
PREFIX_CONTENTS(doc_id, pref_id, pref_name, content)
PREF_AB(doc_id, pref_id, word_pos, sent_pos, content)
PREF_AU(doc_id, pref_id, word_pos, sent_pos, content)
…
PREF_Y2(doc_id, pref_id, word_pos, sent_pos, content)
```

Листинг 8.1. Шеме релација текст сервера

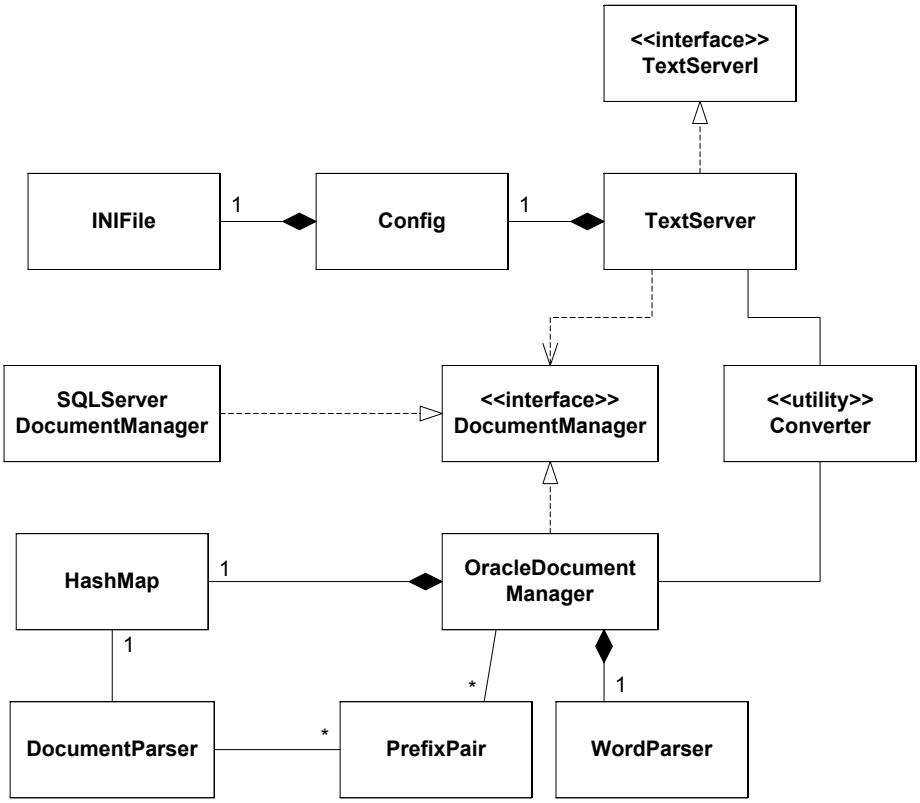
8.3. Подсистем за руковање записима

Руковање UNIMARC записима обухвата следеће операције: додавање, уклањање и ажурирање записа. На слици 8.3 је приказан дијаграм класа овог подсистема. У раду [10] приказани су још неки детаљи овог подсистема.

Интерфејс *TextServerI* представља везу овог система са корисником. Он дефинише методе које су кориснику (тј. клијент-апликацији) доступне. Класа *TextServer* имплементира овај интерфејс. Више детаља о разлозима зашто се комуникација са корисником одвија преко интерфејса (а не директно са главном класом) дато је у поглављу о имплементацији.

Класа *Config* је намењена за чување података о конфигурацији текст сервера. Конфигурацију чине параметри текст сервера који зависе од конкретне инсталације програма (адреса сервера са базом података, тип сервера, кориснички налог на серверу итд.).

Подаци о конфигурацији програма се чувају у датотеци која има формат INI датотека познатих из фамилије *Windows* оперативних система. INI датотеке су текстуалне датотеке подељене на више секција, где су у свакој секцији дефинисани одговарајући параметри. Пример једне INI датотеке дат је на листингу 8.2.



Слика 8.3. Дијаграм класа подсистема за руковање записима

```
[database]
dbtype = oracle  ; tip baze: {oracle,sqlserver,informix}
jdbc = oracle.jdbc.driver.OracleDriver
connection = jdbc:oracle:thin:@sunce.tmd.ns.ac.yu:1526:BIS8
username = mbranko
password = D23F44G

[config]
parserID = 1
hexconv  = 0
```

Листинг 8.2. Пример INI датотеке

Класа *Converter* садржи методе за међусобну конверзију различитих кодних распореда. Интерфејс *DocumentManager* дефинише све операције које остатак система упућује према серверу базе података. Његова намена је да одвоји онај део система који зависи од конкретног употребљеног СУБП од

остатка система. Овај интерфејс имплементирају различите класе намењене раду са СУБП различитих произвођача. На дијаграму је приказана класа *OracleDocumentManager* која имплементира овај интерфејс у смислу дефинисања операција које се врше над базом података којом управља СУБП компаније *Oracle*. Систем може да поседује више оваквих класа. Избор конкретне класе која имплементира дати интерфејс (а биће коришћена у одређеној инстанци програма) врши се приликом стартовања програма. Тада се анализира INI датотека и на основу типа СУБП који је у њој наведен бира се одговарајућа класа. На дијаграму је, као пример још једне класе која имплементира интерфејс *DocumentManager* приказана класа *SQLServerDocumentManager*.

Класа *DocumentParser* је намењена за конверзију UNIMARC записа у префиксе. Класа *PrefixPair* представља уређени пар (префикс, садржај). Низ објеката ове класе представља парсиран UNIMARC запис. Класа *WordParser* служи за парсирање текста на речи и реченице у складу са дефинисаним делимитерима. Користи се за парсирање префикса у процесу индексирања. Класа *HashMap* садржи мапирање потпоља на префиксе.

Процес додавања записа у базу се састоји од конверзије UNIMARC записа у префиксе, парсирања добијених префикса на речи и њиховог смештања у базу. Процес се иницира позивом методе *insertDocument* интерфејса *DocumentManager*, односно неке од класа које имплементирају овај интерфејс. Процес конверзије записа у префиксе се иницира позивом методе *parse* класе *DocumentParser*. Уклањање записа из базе се своди на уклањање одговарајућих *n*-торки у релацијама базе података. Ажурирање записа је заправо секвенца коју чине његово брисање и поновно додавање (са новим садржајем).

8.4. Подсистем за претраживање записа

8.4.1. SELECT упити

Процес претраживања базе записа у највећој мери одређују два захтева: рад са упитима формираним по модификованој *Dialog* синтакси и потреба за познавањем резултата претраге по свим елементарним упитима који чине дати упит. У монографији [3] је приказан начин обраде упита који је и овде усвојен уз одређене модификације.

Процес обраде SELECT упита има следеће фазе:

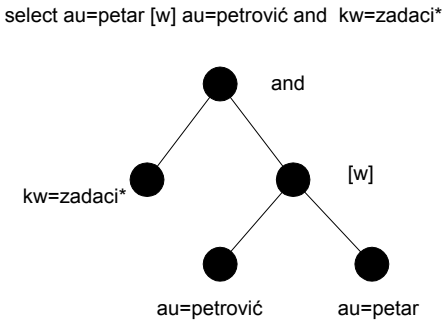
1. *Лексичка анализа*. У овој фази се добијени упит парсира и формира се листа чији су елементи термови претраге (квалификовани или

неквалификовани), оператори и евентуалне заграде. Такође се открива и евентуална употреба непостојећих префикса.

2. *Синтаксна и семантичка анализа*. На основу претходно генерисане листе, формира се синтаксно стабло упита које представља и план извршавања упита. Поред тога, врши се и контрола исправности употребе појединих оператора.

3. *Интерпретација плана извршавања упита*. На основу креираног синтаксног стабла врше се одговарајуће претраге у бази података и израчунава се коначан резултат претраживања.

Пример једног SELECT израза и одговарајућег синтаксног стабла дат је на слици 8.4. У примеру се траже записи чијем се префиксу AU налази реч *petar* одмах поред речи *petrović*, а истовремено се у префиксу KW налази реч која почиње са *zadaci*. На основу датог упита формирано је стабло које представља и план извршавања упита: стабло се рекурзивно обилази и за сваки чвор се израчунава скуп погодака. Скуп погодака корена представља резултат извршавања упита. Скупови погодака у листовима стабла представљају резултате извршавања елементарних подупита.



Слика 8.4. Пример синтаксног стабла упита

За потребе овог текст сервера под поготком се подразумева следећа уређена четворка: (*doc_id*, *pref_id*, *word_pos*, *sent_pos*).

Елементи уређене четворке су дефинисани на следећи начин:

- *doc_id*: идентификатор документа,
- *pref_id*: идентификатор префикса,
- *word_pos*: редни број речи у префиксу и
- *sent_pos*: редни број реченице у којој се реч налази.

Скуп погодака у листовима стабла се добија директно из базе података, постављањем одговарајућег SQL упита. Један од SQL упита који би био постављен у горњем примеру је:

```
SELECT doc_id, pref_id, word_pos, sent_pos
  FROM pref_AU
  WHERE content = 'PETROVIĆ';
```

Из примера се види да је одговарајући SQL упит могуће једноставно генерисати – једини променљиви елементи у изразу су назив табеле у којој се врши претрага и стринг који се тражи.

У чворовима стабла који нису листови налазе се оператори. За ове чворове се одговарајући скуп погодака израчунава на основу скупова погодака два подређена чвора. Начин израчунавања резултујућег скупа погодака зависи од врсте оператора.

За оператор AND резултујући скуп се рачуна као пресек скупова погодака подређених чворова, при чему су два поготка једнака ако им је једнак *doc_id*.

За оператор OR резултујући скуп се рачуна као унија скупова погодака подређених чворова, при чему су два поготка једнака ако им је једнак *doc_id*.

За оператор NOT резултујући скуп се рачуна као разлика скупова погодака чвора који одговара левом операнду и чвора који одговара десном операнду, при чему су два поготка једнака ако им је једнак *doc_id*.

За оператор [F] резултујући скуп се рачуна као пресек скупова погодака подређених чворова, при чему су два поготка једнака ако су им једнаки одговарајући *doc_id* и *pref_id*.

За оператор [S] резултујући скуп се рачуна као пресек скупова погодака подређених чворова, при чему су два поготка једнака ако су им једнаки одговарајући *doc_id*, *pref_id* и *sent_pos*.

За оператор [Wn] резултујући скуп се рачуна као пресек скупова погодака подређених чворова, при чему су два поготка једнака ако су им једнаки одговарајући *doc_id* и *pref_id*, а апсолутна вредост разлике *word_pos* је мања или једнака *n*.

Из описаног начина израчунавања резултата се види да се израчунавање скупа погодака "не-лист" чворова стабла своди на скуповне операције, при чему критеријум једнакости два елемента скупа зависи од коришћеног оператора. Табела 8.1 сумира претходно изложене операторе и одговарајуће критеријуме једнакости погодака.

Оператор	Критеријум једнакости	Скуповна операција
AND	doc_id1 = doc_id2	пресек
OR	doc_id1 = doc_id2	унија
NOT	doc_id1 = doc_id2	разлика
[F]	doc_id1 = doc_id2 pref_id1 = pref_id2	пресек
[S]	doc_id1 = doc_id2 pref_id1 = pref_id2 sent_pos1 = sent_pos2	пресек
[Wn]	doc_id1 = doc_id2 pref_id1 = pref_id2 |word_pos1 – word_pos2| <= n	пресек

Табела 8.1. Преглед коришћених операција приликом израчунавања скупова погодака

8.4.2. Коректност SELECT упита

На основу дефиниције приказаних оператора види се да није могуће комбиновати операторе у оквиру упита на произвољан начин зато што се приликом израчунавања неког међурезултата одређени подаци одбацују. Следи пример једног коректног упита:

```
select au=petar [w] au=petrović and kw=zadaci          (*)
```

Приликом обраде претходног упита прво ће се обрађивати термови са елементарним упитима, затим ће се рачунати скуп погодака за оператор [W] и, потом, скуп погодака за оператор AND (стабло овог упита је дато на слици 8.4). Овде је важно приметити да је оператор [W] употребљен *пре* оператора AND. Сада следи пример једног некоректног упита:

```
select au=petar [w] (au=petrović and kw=zadaci)        (**)
```

Уколико би овакав упит био допустив, тада би се прво израчунавао скуп погодака за оператор AND па затим за оператор [W]. Приликом рачунања скупа погодака за оператор AND погоци са једнаким елементом *doc_id* би се сматрали једнаким, и само један од њих би ушао у резултујући скуп (дупликати у скупу нису дозвољени). Тај елемент, који је у резултујући скуп ушао као представник погодака са једнаким *doc_id,* не мора касније (приликом рачунања скупа погодака за [W]) бити једнак неком елементу из другог скупа погодака, што не гарантује да неки други елемент који је претходно елиминисан не би био једнак. Овај проблем ће сада бити илустрован на примеру.

Нека су скупови погодака за елементарне упите следећи:

терм	погоци
au=petar	(1, 1, 2, 1)
au=petrović	(1, 1, 1, 1)
kw=zadaci	(1, 2, 1, 1)

Како се, за упит (**), прво израчунава скуп погодака за оператор AND, (дакле *au=petrović and kw=zadaci*) рецимо да је у пресечни скуп ушао погодак (1, 2, 1, 1). Тада ће скуп погодака за оператор [W] бити празан скуп, јер (1, 2, 1, 1) није једнак поготку (1, 1, 2, 1) добијеном из терма *au=petar*. У случају да је у први пресечни скуп ушао погодак (1, 1, 1, 1), тада укупан резултујући скуп не би био празан.

Због претходно описаног проблема уводи се следеће ограничење: *није дозвољено да било који потомак чвора у стаблу има категорију већу од категорије претка*. Категорија се дефинише као додатни атрибут чвора и представљен је целим бројем. Вредност категорије зависи од типа чвора. Табела 8.2 наводи врсте чворова и њихове категорије.

тип чвора	категорија
and, or, not	1
[F]	2
[S]	3
[Wn]	4
list	5

Табела 8.2. Категорије чворова синтаксног стабла

Посебан проблем остаје случај узастопног коришћења истог оператора. Упит као што је:

```
select A and B and C
```

је дозвољен. Резултат упита је пресек скупова погодака, где је у оба случаја за једнакост елемената узиман критеријум за AND оператор. Међутим, формално је дозвољен и следећи упит:

```
select A [w] B [w] C                                    (***)
```

Резултат извршавања овог упита не мора бити познат, бар за претраживање онакво какво је до сада дефинисано. Проблем представља избор одговарајућег елемента који ће ући у пресечни скуп. Нека су скупови погодака за елементарне подупите претходног упита следећи:

терм	погоци
A	(1, 1, 1, 1)
B	(1, 1, 2, 1)
C	(1, 1, 3, 1)

Како се прво рачуна скуп погодака за израз **A [w] B**, у резултујући скуп може ући или погодак (1,1,1,1) или погодак (1,1,2,1). Ако се изабере први погодак, укупан резултат упита ће бити празан скуп (јер погодак (1,1,1,1) неће проћи поређење са поготком (1,1,3,1)), а ако се изабере други погодак, укупан резултат упита неће бити празан скуп.

Због овог проблема се, за израчунавање резултујућег скупа погодака за оператор [Wn], уводи додатно правило: *у резултујући скуп погодака улази онај погодак (од два једнака) који припада скупу погодака десног операнда*.

На овај начин је решен и проблем постављања упита типа (***), што је изузетно важно, јер упити овог типа представљају, заправо, претраживање по фразама. Узастопно коришћење свих других оператора осим [W] не ствара овај проблем, па се ово додатно правило на њих не мора односити.

8.4.3. EXPAND упити

Друга врста упита који се могу поставити текст серверу су EXPAND упити. Овакви упити, као што је раније речено, имају једноставну синтаксу: упит се састоји од само једног терма (квалификованог или неквалификованог). Резултат извршавања је скуп парова (број, префикс) где је *префикс* сваки префикс у бази који почиње датим термом, а *број* представља број појављивања тог префикса у бази података. Пример једног EXPAND упита би био:

```
expand au=petr
```

Резултат овог упита би били они садржаји префикса AU који почињу термом *petr* (на пример, *Петровић*, *Петрановић*, итд.). Овакав упит се, на основу шема релација приказаних на листингу 8.1 може конвертовати у следећи SQL исказ:

```
SELECT COUNT(*), a.content
  FROM Prefix_contents a, pref_AU b
  WHERE b.content LIKE 'petr%'
    AND b.word_pos = 1
    AND b.pref_id = a.pref_id
  ORDER BY a.content;
```

Из примера се види да је једини променљиви део овог SQL исказа стринг који се тражи (*petr%*) и назив табеле са речима (*pref_AU*). Генерисање одговарајућег SQL упита на основу датог EXPAND упита је релативно једноставно.

8.4.4. Класе подсистема за претраживање записа

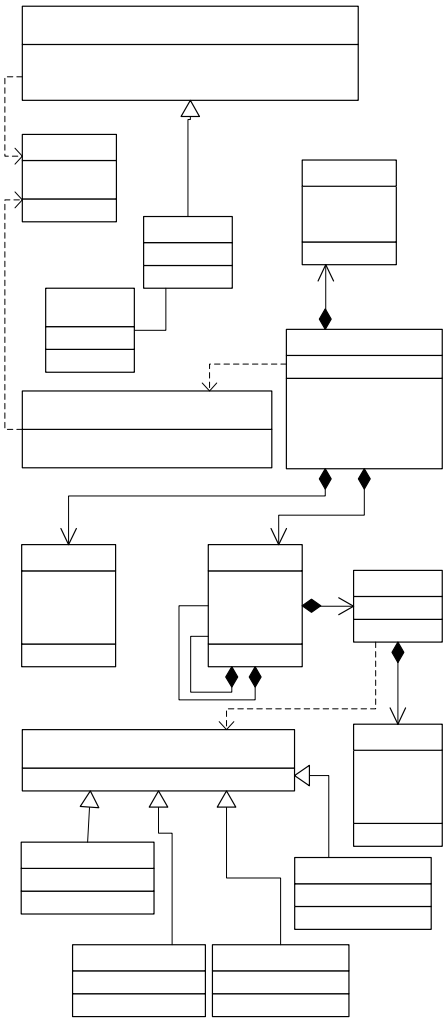
На слици 8.5 представљен је дијаграм класа подсистема за претраживање записа. Интерфејс *TextServerI* и класа *TextServer* представљају везу са корисником и већ су приказане на дијаграму класа подсистема за руковање записима (слика 8.3). Осим метода намењених руковању записима, овај интерфејс и класа дефинишу методе које се користе током претраживања. Интерфејс *DocumentManager* се такође појављује и у овом дијаграму. Он омогућава приступ конкретном коришћеном СУБП. Има дефинисане методе које се користе приликом претраживања базе података.

Један од главних атрибута класе *TextServer*, свакако најважнији за претраживање је атрибут класе *ExpressionTree*. Ова класа представља стабло претраге и дефинише методе за постављање упита и преузимање резултата. Класа *ListElem* се користи у процесу обраде упита за креирање листе термова, оператора и заграда који чине упит.

Класа *TreeNode* представља чвор стабла претраге. Сваки објекат класе *TreeNode* је повезан са још три чвора – непосредним родитељем и непосредним потомцима. Корен стабла је онај чвор чији родитељ није дефинисан. Један од атрибута класе *TreeNode* је и атрибут класе *HashSet* која је намењена за чување скупова погодака и имплементира потребне скуповне операције над њима. Класа *SearchHit* представља један погодак у процесу претраживања.

Интерфејс *BinaryPredicate* дефинише само једну методу. Она се користи за поређење елемената скупа унутар класе *HashSet*, приликом израчунавања скуповних операција. Овај интерфејс имплементира више класа. Класа *BooleanComparator* је намењена за поређење елемената приликом обраде логичких оператора (*and*, *or* и *not*). Класа *DistanceComparator* се користи приликом обраде оператора [Wn]. Класа *SentenceComparator* се користи приликом обраде оператора [S]. Класа *FieldComparator* се користи приликом обраде оператора [F].

Класа *SubqueryHit* представља један извештај о резултату претраге за елементарни подупит. Класа *ExpandPair* представља једну ставку извештаја који представља резултат упита типа EXPAND. Класа *Converter* дефинише методе за међусобне конверзије различитих кодних распореда.



Слика 8.5. Дијаграм класа подсистема за претраживање записа

Процес обраде SELECT упита се иницира (са стране корисника) позивом одговарајуће методе интерфејса *TextServerI*, односно класе *TextServer*. Овај упит се одмах прослеђује класи *ExpressionTree* позивом одговарајуће

методе. Класа *ExpressionTree* обраду упита спроводи у четири фазе (метода *execute*):

1. креирање инфиксне листе (метода *createInfixList*),
2. креирање постфиксне листе (метода *createPostfixList*),
3. креирање стабла претраге (метода *createTree*),
4. обилазак стабла и рачунање резултата (метода *traverse*).

Прва фаза дефинисана је алгоритмом који је овде дат својим вербалним описом.

Текст упита се подели на токене око делимитера (,), = и <бланко>. За сваки токен се ради следеће:

1. Ако је токен <бланко>, игнорише се.
2. Ако је токен отворена заграда, додаје се у инфиксну листу.
3. Ако је токен затворена заграда, додаје се у инфиксну листу.
4. Ако је токен знак једнакости, проверава се да ли је претходни токен био назив префикса. Ако није, генерише се извештај о грешци. Ако јесте, токен се игнорише.
5. Ако је токен назив префикса, памти се посебно.
6. Ако је токен оператор, додаје се у инфиксну листу.
7. Једина преостала могућност је да је токен терм за претрагу. Ако претходно додати елемент у инфиксној листи постоји и није оператор или отворена заграда, генерише се извештај о грешци. Иначе се терм смешта у инфиксну листу, уз евентуалну квалификацију префиксом ако је он претходно дефинисан.

Друга фаза представља алгоритам за конверзију израза из инфиксне у постфиксну нотацију. Овај алгоритам дат је својим следећим вербалним описом:

1. Узима се елемент из инфиксне листе, са почетка.
2. Ако је текући елемент терм, додаје се на крај постфиксне листе.
3. Ако је текући елемент отворена заграда, смешта се на стек.
4. Ако је текући елемент затворена заграда, са стека се скидају сви елементи до прве отворене заграде и додају на крај постфиксне листе у редоследу у ком су скидани са стека. Отворена заграда се скида са стека, али се не додаје у постфиксну листу.
5. Ако је текући елемент оператор, са стека се скидају елементи који имају већи приоритет од текућег оператора. Скидање се врши до првог елемента који има мањи приоритет или до прве отворене

заграде. Елементи се додају на крај постфиксне листе по редоследу скидања са стека. Текући елемент се, затим, ставља на стек.

6. Кораци 1-5 се понављају док постоје елементи у инфиксној листи.
7. Преостали елементи са стека се додају на крај инфиксне листе у редоследу скидања.

Трећа фаза је алгоритам за креирање синтаксног стабла на основу постфиксне листе. Овај алгоритам приказан је својим вербалним описом.

1. Из постфиксне листе се узима елемент са краја.
2. Ако је стабло празно, тај елемент постаје корен стабла, и текући чвор стабла постаје корен. Прелази се на корак 5.
3. Ако је текући чвор стабла оператор, провери се да ли постоји десни подређени чвор текућег чвора. Ако не постоји, елемент из листе постаје десни подређени чвор, а затим постаје и текући чвор стабла. Ако десни подређени чвор постоји, због природе алгоритма леви подређени сигурно не постоји и елемент из листе се смешта у стабло на место левог подређеног чвора, а затим постаје и текући чвор стабла. У овом кораку се врши и провера категорија родитељског и новог чвора. Прелази се на корак 5.
4. Ако је текући чвор стабла терм, тражи се први његов предак који нема левог подређеног а представља оператор. Када се такав елемент нађе, елемент из листе постаје његов леви подређени чвор, а затим постаје и текући чвор стабла.
5. Уколико постфиксна листа није празна, прелази се на корак 1. У другом случају стабло је формирано.

Четврта фаза (обилазак стабла претраге) представља варијанту рекурзивног алгоритма за обилазак бинарног стабла и одређивања скупа погодака за сваки чвор стабла. Одређивање скупа погодака зависи од врсте чвора. Ако је чвор уједно и лист стабла (тј. представља елементарни упит), помоћу одговарајуће методе интерфејса *DocumentManager* се добија одговарајући скуп погодака директно из базе података. Ако чвор представља оператор, скуп погодака се рачуна на начин описан у одељцима 4.2 и 4.3. У ту сврху се користи класа *HashSet* и одговарајућа класа из групе класа које имплементирају интерфејс *BinaryPredicate*.

Имплементација обраде упита типа EXPAND је далеко једноставнија од обраде SELECT упита. Обрада EXPAND упита се ослања на одговарајућу методу интерфејса *DocumentManager* (и класе која га имплементира) која преузима упит и, на основу генерисаног SQL упита, враћа резултате. Генерисање оваквих SQL упита је приказано у одељку 8.4.3.

У следећем поглављу описано је корисничко претраживање UNIMARC записа путем Интернета које је базирано на овде приказаном текст серверу.

8.5. Закључак

Најважније карактеристике приказаног текст сервера су следеће: специјализованост, коришћење Java платформе, трослојна архитектура и независност од коришћеног СУБП.

Специјализованост. Основна одредница у спецификацији сервера је ослонац на два стандарда који се јављају у библиотекарству – формат UNIMARC, који дефинише рачунарску репрезентацију библиографске грађе, и упитни језик *Dialog*. Добијени резултат представља текст сервер који је специјализован за примену у библиотечком информационом систему.

Коришћење Java платформе. Коришћењем програмског језика Java у имплементацији постигнута је независност од коришћене рачунарске платформе. На тај начин се овај сервер може користити на различитим серверским машинама, у складу са потребама и могућностима конкретног информационог система.

Трослојна архитектура. Текст сервер је коципиран да функционише као средњи слој у оквиру трослојне архитектуре. Његов дизајн омогућава имплементацију у облику *Enterprise JavaBeans* компоненти. Оваква организација сервера омогућава његово коришћење у различитим, хетерогеним информационим системима. Поред тога, оваква архитектура обезбеђује задовољавајућу скалабилност и поузданост рада у системима са великим бројем клијената или обимом података.

Независност од коришћеног СУБП. Текст сервер је изграђен тако да буде у великој мери независан од коришћеног система за управљање базама података. Додавање подршке за нови СУБП се своди на писање одговарајућег, прецизно дефинисаног модула који се једноставно интегрише у постојећи систем.

Данас постоји већи број комерцијално расположивих текст сервера. У питању су, најчешће, текст сервери опште намене, оријентисани ка *full-text* претраживању докумената у различитим форматима (ASCII, HTML, *MS Word* итд.). Њихова област примене је далеко шира од области примене текст сервера UNIMARC записа. Са друге стране, недостатак директне подршке за UNIMARC формат чини их непогодним за употребу у библиотечком информационом систему.

Референце

[1] *UNIMARC Manual: Bibliographic format*, *International Federation of Library Association and Institutions*. IFLA Universal Bibliografic Control and International MARC Programme, New Providence, London 1994.

[2] *COMARC/R Format*. COBISS, Univerzitetni institut informacijskih znanosti, IZUM, Maribor 1992.

[3] B. Lazarević, redaktor. *Formiranje i pretraživanje baza podataka u Sistemu naučnih i tehnoloških informacija Srbije*. Ministarstvo za nauku i tehnologiju Republike Srbije, Beograd, 1996.

[4] Д. Сурла, З. Коњовић, Б. Пуповац, Б. Милосављевић, М. Видаковић, Т. Зубић, Т. Тошић. *Упутство за коришћење Библиотечког софтверског система БИСИС вер. 3.* Група за информационе технологије, Нови Сад, 2003.

[5] *Successful Searching on Dialog*. Dialog Corporation, 1998. http://support.dialog.com/searchaids/success/

[6] B. Milosavljević, D. Surla. *Implementacija tekst servera za indeksiranje i pretraživanje bibliotečke građe u Oracle okruženju*. Zbornik radova Info-Teh'97, Vrnjačka Banja, 1997. str. 452-456.

[7] B. Milosavljević, Z. Konjović. *Tekst server bibliotečkog informacionog sistema zasnovan na Oracle Context Option-u*. Zbornik apstrakata Sinfon'97, Zlatibor, 1997.

[8] B. Milosavljević. *Tekst server bibliotečkog informacionog sistema zasnovan na Oracle Context Option-u*. Zbornik radova YU Info 98, Kopaonik, 1998.

[9] Б. Милосављевић. *Текст сервер UNIMARC записа*. Магистарска теза, Факултет техничких наука, Нови Сад, 1999.

[10] B. Milosavljević, Z. Konjović. *Specifikacija sistema za indeksiranje UNIMARC zapisa*. Zbornik radova YU Info 99, Kopaonik, 1999.

[11] B. Milosavljević, Z. Konjović. *Specifikacija pretraživanja u tekst serveru za UNIMARC zapise*. Zbornik radova Sym-op-is 99, str. 145-148, Beograd, 1999.

Поглавље 9

Корисничко претраживање

Б. Милосављевић

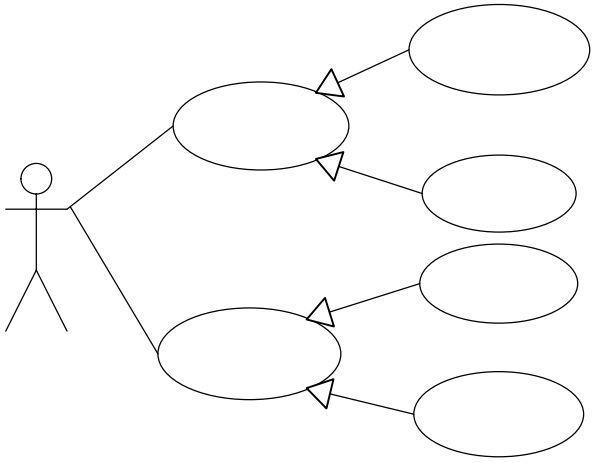
Веб базирани кориснички интерфејс данас представља стандардни интерфејс за приступ и претраживање јавних база података. Најраширеније и најчешће употребљаване базе података су претраживачи јавног Веба као што је Google [1]. Овакви сервиси потпуно аутоматизовано анализирају садржај докумената које уносе у свој индекс. Квалитет каталогизованих података и резултати претраживања директно зависе од садржаја докумената и метаподатака које је аутор документа самостално формирао. Структура метаподатака за опис садржаја докумената на стандардном јавном вебу [2] омогућавају репрезентацију само основних информација о документу, као што су аутор и кључне речи. Употреба функција претраживања докумената у оваквим системима је изузетно једноставна и најчешће се своди на унос текстуалног критеријума претраге који садржи тражене речи. Функција метаподатака у овом случају је да обезбеди боље рангирање пронађених докумената.

Са друге стране, системи каталогизације докумената који у својој основи користе богатије структуре за репрезентацију метаподатака имају потенцијал за изградњу корисничког интерфејса који омогућава прецизније претраживање. Библиотечки информациони системи засновани на UNIMARC формату записа [3] омогућавају претраживање метаподатака о оригиналним документима (фонду библиотеке) при чему формирање UNIMARC записа, односно метаподатака, обављају високо квалификована лица. Квалитет метаподатака формираних уз учешће човека (уместо аутоматски) који сачињавају индекс далеко је већи него у претходном случају.

Јавно доступна функција претраживања каталога библиотеке би требало да задовољи два, у извесној мери опречна, захтева. Са једне стране, употреба функција претраживања мора бити што једноставнија, у идеалном случају као код веб претраживача. Са друге стране, кориснику је потребно обезбедити механизме за формирање прецизних упита по жељеним елемен-

тима индекса што за собом повлачи и дизајн компликованијег корисничког интерфејса.

Приступ функцијама претраживања за стручно особље библиотеке одвија се кроз кориснички интерфејс посебне наменске апликације [13] приказане у четвртом поглављу. Наменске апликације приступају бази записа путем текст сервера [12], посебног модула намењеног индексирању и претраживању UNIMARC записа који је описан у седмом поглављу. Под корисничким претраживањем [11] се, у контексту библиотечких информационих система, подразумева претраживање фонда од стране њених крајњих корисника (функција која се често назива OPAC – *online public access catalogue*). Функционалност корисничког претраживања приказана је дијаграмом случајева коришћења на слици 9.1.



Слика 9.1. Случајеви коришћења за корисничко претраживање

Случај коришћења *Формирање упита* представља унос жељеног критеријума претраге од стране корисника у оквиру веб апликације. Ради постизања што веће једноставности коришћења система, корисницима је омогућено постављање *Једноставних упита*, који се дефинишу критеријумом претраге по највише два елемента метаподатака (нпр. аутор, наслов или кључне речи). *Сложени упити* омогућавају кориснику формирање критеријума претраге по више елемената метаподатака истовремено, при чему се поједини делови критеријума претраге повезују логичким операторима. Како је овај најопштији вид формирања критеријума претраге нешто сложенији за употребу, он је јасно издвојен у односу на једноставне и најчешће постављане упите.

Случај коришћења *Преглед резултата* представља ситуацију када је корисник поставио одговарајући упит и почео са прегледом његових резул-

тата. Резултати постављеног упита могу се прегледати у два различита режима – *Сажетом прегледу* и *Детаљном прегледу*. Ова два типа приказивања резултата разликују се по количини приказаних података о сваком пронађеном документу, као и у начину њиховог визуелног форматирања.

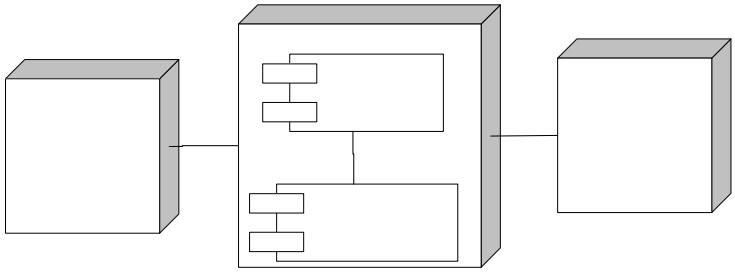
У овом поглављу представљена је софтверска архитектура и дизајн веб апликације за корисничко претраживање у систему БИСИС. У наредном одељку приказани су одређени елементи софтверске архитектуре карактеристични за ову веб апликацију. Други одељак садржи приказ структура података потребних за вођење различитих истовремених корисничких сесија. У трећем одељку описана је структура веб сајта и функционалност појединих страница. Четврти одељак приказује начин интерпретације корисничких упита са ослонцем на релациону базу података. Пети одељак описује имплементацију система и садржи кратак опис коришћења овог подсистема. У шестом одељку дата су закључна разматрања и анализа могућности за даља унапређења подсистема за корисничко претраживање.

9.1. Елементи софтверске архитектуре

Типичне веб апликације омогућавају ажурирање података у релационој бази коју деле са другим модулима система. За потребе комуницирања са системом за управљање базом података (СУБП) оне морају да користе механизме контроле истовременог приступа подацима. Ови механизми контроле обухватају трансакциони режим рада СУБП, одржавање копије најчешће приступаним подацима у сопственој меморији (*cache*) и руковање конекцијама са базом података по принципу коришћења дељених ресурса (*connection pooling*) [4].

Веб апликација за корисничко претраживање користи се искључиво за приступ и читање података из локалне базе БИСИС-а. Операције ажурирања података у бази нису нити доступне нити имплементиране. Стога нема потребе да се за приступ бази података користи више од једне конекције, јер ће све SQL команде упућене СУБП бити SELECT команде, које не врше измене података нити њихово закључавање (спречавање приступа подацима од стране других корисника, тј. кроз друге конекције). Одржавање меморијских *cache* структура за једну коришћену конекцију одвија се у оквиру СУБП и сервер веб апликације нема потребе за одржавањем сопствене *cache* структуре. Са друге стране, информације о различитим сесијама корисника који истовремено користе веб апликацију морају се чувати у меморији веб апликације. Ове информације користе се приликом генерисања HTML стра-

ница које се шаљу појединим клијентима. Слика 9.2 приказује компоненте веб апликације БИСИС-а.



Слика 9.2. Компоненте веб апликације за корисничко претраживање

За имплементацију корисничког претраживања БИСИС-а користи се програмски језик Java и скуп сродних технологија за писање веб апликација. У питању је сервлет технологија [5], односно JSP технологија за генерисање динамичких HTML страница [6]. Приказано решење веб апликације није у великој мери зависно од употребљене имплементационе технологије, тако да се може користити и у архитектурама које нису засноване на програмском језику Java.

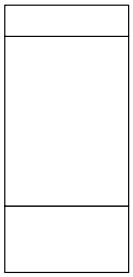
9.2. Корисничке сесије

Комуникација између веб клијената и веб сервера одвија се путем HTTP протокола [7] у паровима порука типа захтев/одговор. Једна од важних карактеристика овог протокола је да он не обезбеђује чување стања у комуникацији између више узастопних парова порука. На тај начин сваки захтев упућен веб серверу третира се на исти начин, без обзира који клијент га је упутио. Насупрот томе савремене веб апликације подразумевају сложенији кориснички интерфејс који подразумева да се интеракција између клијента и сервера одвија кроз више захтева како би се извршила одређена функција апликације. Како HTTP протокол не обезбеђује међусобно разликовање различитих стварних клијената на серверу, развијени су механизми који превазилазе овај недостатак.

Механизам „колачића“ (*cookies*) [8] представља једно решење наведеног проблема. *Cookie* представља једнозначни идентификатор кога сервер додељује сваком клијенту приликом слања одговора на први захтев клијента. У наредним паровима порука захтев/одговор овај идентификатор се смешта у поруку тако да се сваки клијент може тачно одредити. Други механизам за идентификацију клијената се користи у случајевима када веб клијент не подр-

жава рад са *cookie* идентификаторима и своди се на уграђивање сличних идентификатора у облику параметара у URL адресама које се користе у комуникацији (*URL rewriting*) [9].

Веб сервери који подржавају сервлет односно JSP технологију имају уграђене механизме за рад са *cookie* идентификаторима или модификацију URL адреса. Сервис једнозначне идентификације клијената представља стандардни сервис оваквих веб сервера коме се приступа путем датог програмског интерфејса [5, 6]. Из тог разлога се приликом моделирања података о корисничким сесијама не мора посебно обрађивати механизам идентификације клијената. Класа *UserInfo* приказана на дијаграму са слике 9.3 намењена је за евиденцију података о корисничкој сесији. Подаци из корисничких сесија не складиште се у бази података, већ постоје искључиво у меморији веб сервера. Уклањање ових података из меморије извршава се након истека одређеног временског интервала у коме је дати клијент био неактиван, односно није упућивао захтеве серверу.

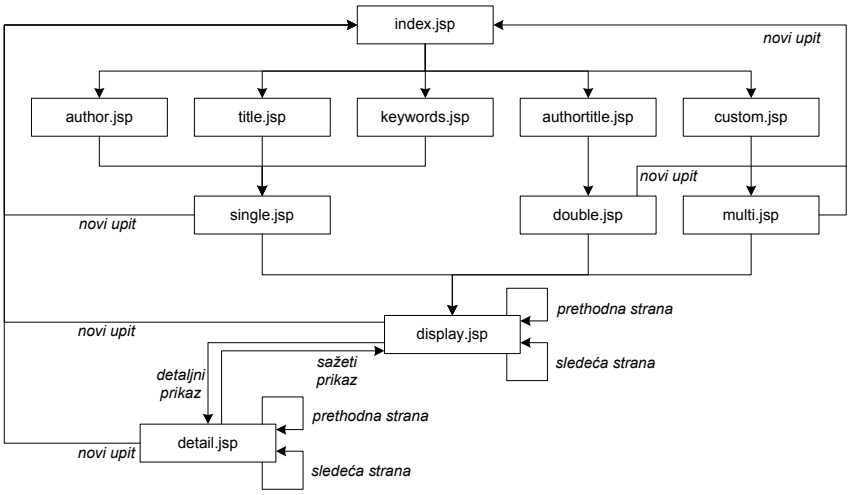


Слика 9.3. Класа *UserInfo* за репрезентацију корисничке сесије

Низ *briefs* класе *UserInfo* намењен је за чување идентификатора свих пронађених UNIMARC записа. Сви пронађени записи се приказују у сажетом прегледу. У оквиру сажетог прегледа неки од приказаних записа могу бити означени за приказ у детаљном прегледу. Идентификатори ових записа чувају се у низу *details*. Генерисање садржаја низа *details* обавља се у методи *createDetails* на основу садржаја низа *checked* који чува податке о записима који су означени за детаљан преглед. Приликом листања резултата кроз више страница низ *checked* се ажурира помоћу методе *updateChecked*. Листање резултата кроз више страница врши се због прегледности и брзине преноса појединих страница до клијента. Текући приказани сегмент укупног списка погодака одређен је индексима низа идентификатора *startIndexBrief* и *endIndexBrief* за сажети преглед, односно *startIndexDetail* и *endIndexDetail* за детаљни преглед.

9.3. Структура веб сајта

Веб сајт за корисничко претраживање састоји се из одређеног броја динамичких HTML страница. На слици 9.4 дат је дијаграм који описује навигацију корисника кроз странице сајта.



Слика 9.4. Навигација кроз странице сајта

Страница *index.jsp* је почетна страница сајта. На њој је понуђен избор од неколико начина претраживања: по аутору (страница *author.jsp*), по наслову (*title.jsp*), по кључним речима (*keywords.jsp*), по аутору и наслову (*authortitle.jsp*) и по сложеном критеријуму (*custom.jsp*). Претрага по аутору, наслову и кључним речима представља претрагу по само једном критеријуму – префиксу. Претрага по аутору и наслову је претрага по два префикса, при чему је дати садржај префикса за аутора и наслов повезан логичком конјункцијом, тј. захтева се постојање оба тражена садржаја. Претрага по сложеном критеријуму састоји се од највише пет компоненти, где свака компонента представља претрагу по префиксу изабраном из укупног скупа префикса, а компоненте су међусобно повезане логичким операторима AND, OR и NOT.

Странице *author.jsp*, *title.jsp*, *keywords.jsp*, *authortitle.jsp* и *custom.jsp* садрже само HTML форме за унос критеријума претраге. Након попуњавања форми и притиска на одговарајуће дугме, веб читач се упућује на наредну страницу током чије интерпретације ће се извршити претраживање базе података и приказати извештај о броју пронађених погодака. Овом приликом се попуњава одговарајућа инстанца класе *UserInfo* подацима из резултата

претраге. Обрада упита који садрже само један префикс одвија се у оквиру странице *single.jsp*. Упити који садрже два префикса повезана конјункцијом обрађују се у оквиру странице *double.jsp*. Сложени упити састављени из више компоненти повезаних различитим логичким операторима израчунавају се у оквиру странице *multi.jsp*. Све три наведене странице приказују резултат претраге на исти начин: приказује се број пронађених погодака и кориснику нуди могућност да прегледа резултате претраге или да постави нови упит. Постављање новог упита полази од почетне странице сајта *index.jsp*.

Избором опције за преглед пронађених погодака корисник се иницијално налази у режиму сажетог прегледа кога имплементира страница *display.jsp*. У овом режиму приказују се сви пронађени записи. Приказ ових записа подељен је у више сегмената (логичких страница) ради прегледности. Кретање кроз поједине сегменте резултата омогућено је посебним командама за прелазак на претходни односно следећи сегмент. Страница сажетог прегледа садржи и просторе за потврду додељене сваком приказаном запису. Овај простор за потврду користи се да се дати запис означи за детаљни преглед. Активирањем опције за постављање новог упита прелази се на почетну страницу сајта.

Активирањем опције за прелазак у режим детаљног прегледа прелази се на страницу *detail.jsp*. Идентификатори докумената изабраних за детаљни преглед налазе се у оквиру одговарајуће инстанце класе *UserInfo* која је везана за сесију текућег корисника. Приказ погодака у режиму детаљног прегледа такође се одвија кроз сегменте, и то на исти начин као и код сажетог прегледа. Активирањем опције за постављање новог упита прелази се на почетну страницу сајта. Активирањем опције за прелазак у режим сажетог прегледа корисник се враћа на страницу *display.jsp* која приказује онај сегмент из кога је корисник и прешао у детаљни преглед.

Током навигације кроз веб сајт за корисничко претраживање интеракција са базом података одвија се на два начина. Прво, приликом постављања упита и његовог интерпретирања врши се претраживање базе података за траженим записима. Том приликом се у корисничку сесију уписују идентификатори свих пронађених записа. Ова операција у случају обимних база података или великог броја пронађених записа може бити дуготрајна. Друго, приликом приказа пронађених записа (било у сажетом или детаљном прегледу) из базе се преузимају записи на основу њиховог идентификатора који је већ раније преузет. Тиме је приступ појединачним записима учињен ефикасним и ова операција није временски критична у поређењу са претходном. Овом приликом одвија се и форматирање преузетих записа из базе ради приказа у жељеном режиму. Како корисник највећи део времена у интеракцији са системом

проводи у прегледању резултата претраге, оваквим приступом је избегнуто велико оптерећење сервера у амбијенту са великим бројем истовремених корисника.

9.4. Интерпретација упита

За потребе физичког руковања подацима у оквиру система БИСИС може да се користи СУБП заснован на релационом моделу за кога постоји одговарајући JDBC [10] интерфејс. Део шеме релационе базе података која се користи у подсистему за корисничко претраживање представљен је на листингу 9.1. Курзивом су дати називи атрибута који улазе у састав примарног кључа шема релација.

```
DOCUMENTS(doc_id, document)
PREFIX_CONTENTS(doc_id, pref_id, pref_name, content)
PREF_AB(doc_id, pref_id, word_pos, sent_pos, content)
PREF_AU(doc_id, pref_id, word_pos, sent_pos, content)
...
PREF_Y2(doc_id, pref_id, word_pos, sent_pos, content)
```

Листинг 9.1. Шема базе података за корисничко претраживање

Шема релације *Documents* је намењена за чување самих UNIMARC записа у оригиналном облику (атрибут *document*). Шема релације *Prefix_Contents* представља UNIMARC записе конвертоване у скуп префикса. Атрибут *content* представља садржај префикса, а атрибут *pref_name* његов назив. Шеме релација од *Pref_AB* до *Pref_Y2* су намењене за смештање парсираних токена (атрибут *content*) који се налазе у одговарајућим префиксима од AB до Y2. Токени су овде дефинисани као непразни подстрингови у оквиру садржаја једног префикса који су са своје леве и десне стране ограничени специјалним карактерима названим делимитери. Атрибути *word_pos* и *sent_pos* представљају редни број речи у оквиру реченице и редни број реченице у оквиру садржаја префикса. Речи и реченице су дефинисане одговарајућим делимитерима. Над оваквим релацијама су изграђени одговарајући индекси како би се убрзао поступак претраживања. Све *Pref_XX* релације имају дефинисан индекс над атрибутом content. Релација *Prefix_Contents* има дефинисан индекс над паром атрибута (*doc_id*, *pref_name*).

Елементарни упити у оквиру HTML форми су упити који представљају претрагу једне речи (токена) по једном префиксу. За сваки такав елементарни упит може се генерисати одговарајући SQL упит који ће вратити идентификаторе оних записа који задовољавају критеријум претраге. На пример, за упит *AU=Petrović*, одговарајући SQL упит би гласио:

```
SELECT DISTINCT doc_id
  FROM pref_au
  WHERE content = 'Petrović';
```

Из претходног примера се види да су једина два променљива елемента у оваквим упитима назив релације у којој се врши претрага (који директно зависи од назива коришћеног префикса) и садржај у колони *content* који се тражи. Уколико је корисник употребио џокер-знаке, уместо оператора '=' може се користити SQL оператор LIKE. Овакав упит ће за своје извршавање увек користити индекс генерисан над атрибутом *content*.

Уколико је корисник задао упит који се састоји од више речи (токена) у оквиру једног префикса, одговарајући SQL упит је нешто сложенији. На пример, ако корисников упит гласи *AU=Selimović Meša*, одговарајући SQL упит би гласио:

```
SELECT DISTINCT doc_id
  FROM pref_au
  WHERE content = 'Selimović' AND
    pref_id IN
     (SELECT pref_id FROM pref_au
        WHERE content = 'Meša');
```

Из овог примера се види и начин генерисања SQL упита за корисничке упите који у једном префиксу претраге имају више речи: за сваки елементарни упит генерише се SELECT упит по колони *content*, а они се затим угњеждавају помоћу AND *pref_id* IN (...) конструкције у SQL-у.

Уколико је корисник поставио упит са претрагом по више различитих префикса (у оквиру којих је задао једну или више речи), за сваки подупит по једном префиксу се генерише SQL подупит на претходно описани начин, а потом се ти упити повезују SQL операторима INTERSECT, UNION и MINUS, у складу са коришћеним логичким операторима AND, OR и NOT.

Када корисник постави упит, на основу њега се генерише одговарајући SQL упит и његови резултати (идентификатори записа који задовољавају критеријум претраге) се смештају у оквиру података везаних за корисничку сесију. При преласку корисника на странице за преглед погодака ови идентификатори се користе за брзо проналажење одговарајућих записа. Приликом приказивања погодака у сажетом и детаљном режиму користе се подаци из релације *Prefix_Contents*. У оквиру режима за сажети приказ погодака, за проналажење одговарајућих префикса користи се SQL упит следећег облика:

```
SELECT prefid, pref_name, content, doc_id
  FROM prefix_contents
  WHERE doc_id IN
```

```
(id1, id2, ..., idn) AND
pref_name IN ('AU', 'TI', 'PP',
 'PY', 'PU', 'SP', 'DC', 'TS',
 'LA', 'BN');
```

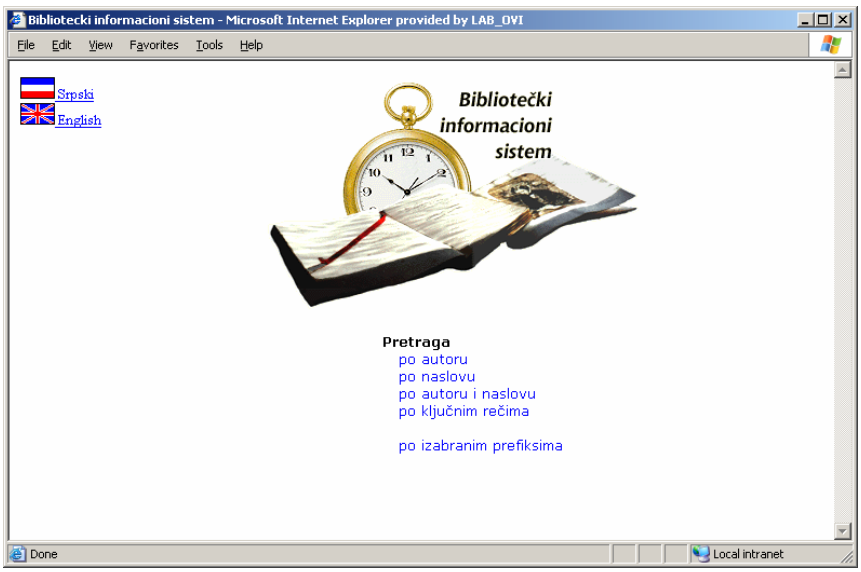
где су *id1, id2, ... idn* идентификатори оних записа које је потребно приказати на текућој страници. 'AU', 'TI', до 'BN' су називи оних префикса које је потребно приказати. Приликом овог претраживања увек се користи индекс генерисан над паром атрибута (*doc_id*, *pref_name*).

9.5. Имплементација

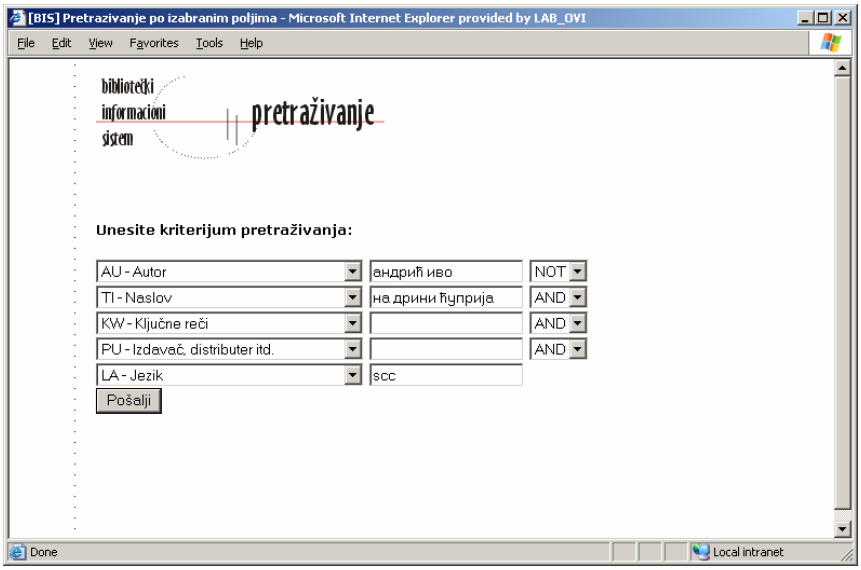
Имплементација корисничког претраживања користи програмски језик Java, и технологије сервлета [5] и JSP страница [6]. Као веб сервер за овај подсистем може се користити било који сервер који подржава наведене технологије. За развој система коришћен је веб сервер који представља референтну имплементацију сервлет и JSP технологија, *Jakarta Tomcat* [14], који је у јавном власништву.

Имплементирана веб апликација се може инсталирати и у оквиру апликативних сервера који задовољавају потпуну J2EE спецификацију. Рад апликације у кластеру који се састоји од више веб сервера је могућ уколико дати сервер обезбеђује одговарајуће механизме за контролисани приступ дељеним објектима који у овом случају представљају корисничке сесије, тј. инстанце класе *UserInfo*.

Основна страница веб сајта за корисничко претраживање приказана је на слици 9.5. Основна страница омогућава избор језика корисничког интерфејса за текућег корисника и садржи линкове ка страницама за формирање упита претраге. Страница за формирање сложеног критеријума претраге приказана је на слици 9.6. Функционалност страница за формирање упита описана је у одељку 9.3. Детаљно упутство за коришћење подсистема за корисничко претраживање дато је у књизи [15].



Слика 9.5. Основна страница сајта за корисничко претраживање



Слика 9.6. Страница за формирање упита по сложеном критеријуму

9.6. Закључак

Веб базирани кориснички интерфејс карактеришу једноставност употребе и широка доступност. Због оваквих својих особина он представља и природно решење за имплементацију система за корисничко претраживање јавних библиотечких каталога. Специфичност система за корисничко претраживање у односу на друге веб апликације је недостатак функција које ажурирају податке у систему. Из тог разлога могуће је користити једноставније софтверске архитектуре у имплементацији оваквих система, а истовремено очувати прихватљиве перформансе система у окружењима са великим бројем истовремених корисника.

Приступ подсистему за корисничко претраживање одвија се путем стандардног HTTP протокола који не обезбеђује праћење стања у комуникацији између клијента и сервера која се састоји из више парова захтев/одговор. Употреба стандардних механизама за превазилажење овог проблема (пре свега *cookie* механизма) је омогућена у оквиру веб сервера са подршком за програмски језик Java и сродне технологије, у првом реду сервлете и JSP странице. Имплементација веб апликација у оваквом окружењу је значајно поједностављена, али је уједно и приказано решење на једноставан начин могуће имплементирати и у другим окружењима за развој веб апликација.

Структура веб сајта за корисничко претраживање подржава рад са корисницима различитих нивоа вештине. Постојање различитих начина за постављање упита, различитих по својој сложености, као и постојање више режима прегледа пронађених докумената омогућава корисницима да начин интеракције са системом прилагоде својим потребама. Поред тога, структура веб сајта и начин руковања подацима су прилагођени потребама ефикасног опслуживања великог броја корисника. Временски најзахтевнија операција, интерпретирање корисничког упита и израчунавање резултата, обавља се само једном за дати упит. Релациона база података која се користи у овом случају има пре свега функције физичког руковања подацима и приступа подацима преко индекса имплементираног у оквиру самог СУБП.

Даљи развој подсистема за корисничко претраживање лежи пре свега у додавању приступа другим функцијама система БИСИС, пре свега функцијама подсистема за циркулацију докумената. На тај начин може се имплементирати резервација библиотечке грађе за регистроване кориснике библиотеке.

Референце

[1] *Google*. http://www.google.com

[2] D. Raggett, A. Le Hors, I. Jacobs. *HTML 4.01 Specification*. W3C Consortium, 1999. http://www.w3.org/TR/html4/

[3] *UNIMARC Manual: Bibliographic format*, *International Federation of Library Association and Institutions*. IFLA Universal Bibliografic Control and International MARC Programme, New Providence, London 1994.

[4] B. Milosavljević, M. Vidaković. *Java i Internet programiranje*. Grupa za informacione tehnologije, Novi Sad, 2002.

[5] D. Coward. *Java Servlet Specification Version 2.3*, Sun Microsystems, 2001. http://java.sun.com/j2ee

[6] M. Roth, E. Pelegrí-Llopart. *JavaServer Pages Specification Version 2.0*. Sun Microsystems, 2003. http://java.sun.com/j2ee

[7] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. *Hypertext Transfer Protocol – HTTP/1.1*. IETF RFC 2616. http://www.ietf.org/rfc/rfc2616.txt

[8] *Persistent Client State: HTTP Cookies*. Netscape Communications. http://wp.netscape.com/newsref/std/cookie_spec.html

[9] R.S. Engelschall. *A Users Guide to URL Rewriting with the Apache Webserver*. Apache Software Foundation. http://www.engelschall.com/pw/apache/rewriteguide/

[10] J. Ellis, L. Ho, M. Fisher. *JDBC 3.0 Specification*. Sun Microsystems, 2001. http://java.sun.com/j2se/

[11] B. Milosavljević, Z. Konjović. *Arhitektura WWW prezentacije za pretraživanje u bibliotečkom informacionom sistemu*. InfoScience, 8(4):18-23, 2000.

[12] Б. Милосављевић. *Текст сервер UNIMARC записа*. Магистарска теза, Факултет техничких наука, Нови Сад, 1999.

[13] М. Видаковић. *Објектно-оријентисана спецификација и имплементација корисничког интерфејса за библиотечки информациони систем*. Магистарска теза, Факултет техничких наука, Нови Сад, 1998.

[14] *Jakarta Tomcat*. http://jakarta.apache.org/tomcat

[15] Д. Сурла, З. Коњовић, Б. Пуповац, Б. Милосављевић, М. Видаковић, Т. Зубић, Т. Тошић. *Упутство за коришћење Библиотечког софтверског система БИСИС вер. 3*. Група за информационе технологије, Нови Сад, 2003.

Поглавље 10

Коришћење библиотечке грађе

Т. Тошић, Д. Сурла

Посебан део библиотечког пословања представља сегмент који се односи на коришћење библиотечке грађе, односно праћење коришћења фонда библиотеке, резервисања и задуживања публикација, као и праћење међубиблиотечке позајмице. За разлику од осталих сегмената система, сегмент коришћења библиотечке грађе је специфичан за различите врсте библиотека. Спецификација информационих захтева овог сегмента урађена је у сарадњи са факултетским и градским библиотекама. Та сарадња била је посебно изражена са библиотеком Департмана за математику и информатику Природно-математичког факултета у Новом Саду и Библиотеком града Београда [1, 2].

У зависности од врсте услуга коју очекују од библиотеке, разликујемо три типа корисника: индивидуалне кориснике, колективне кориснике и институције позајмице. Индивидуални корисници представљају физичка лица која се учлањују у библиотеку ради позајмљивања књига из њеног фонда и коришћења осталих услуга библиотеке. Колективни корисници су организације које врше колективно учлањење својих чланова у библиотеку. Сваки од чланова те организације се затим посебно учлањује у библиотеку као индивидуални члан. Институције позајмице представљају организације (библиотеке) које позајмљују библиографску грађу односно дају библиографску грађу на позајмицу.

Да би се омогућило ефикасно праћење библиотечких докумената, неопходно је обезбедити евидентирање корисника библиотеке, као и евидентирање података о резервисаним и задуженим публикацијама. У наредном одељку описан је дијаграм класа коришћења библиотечке грађе.

10.1. Дијаграм класа коришћења библиотечке грађе

Моделирање шеме базе података коришћења библиотечке грађе извршено је коришћењем UML нотације [3], а реализовано у CASE алату *Rational Rose* [4]. Логички модел базе података дат је на дијаграму класа приказаном на слици 10.1. На основу дијаграма класа, коришћењем одговарајућих алата, генерисан је физички модел података. У наредном тексту детаљно је објашњен дијаграм класа коришћења библиотечке грађе.

Класа *CIRC_DOCS* представља све публикације у фонду библиотеке. Атрибут *ctlg_no* (инвентарни број) јединствено идентификује публикацију. Класа садржи још и атрибуте: *sig* (сигнатура), *docid* (идентификатор документа), *status* (статус публикације – слободна или заузета), *type* (тип публикације – монографија, серијска публикација ...), *state* (стање публикације) и *rcv_req_id* (идентификатор примљеног захтева за резервацију).

Класом *OFFICIALS* представљено је службено лице (библиотекар) које врши ажурирање базе циркулације библиотечке грађе. Атрибут *id* јединствено идентификује библиотекара. Класа садржи још и атрибуте *last_name* (презиме библиотекара), *first_name* (име библиотекара) и *email* (e-mail адреса библиотекара).

Класа *USER_TYPES* представља шифарник типова корисника библиотеке. Атрибут *id* јединствено идентификује тип корисника. Атрибут *name* означава име типа корисника и може имати следеће вредности: индивидуални корисник, колективни корисник и институција размене. Атрибут *titles_no* означава број публикација које корисник поједине врсте може да узме из библиотеке, а *period* максималан период на који се може позајмити публикација.

Класа *USERS* садржи основне податке за сваки тип корисника. Атрибут **id** представља јединствени идентификатор корисника, док су остали подаци о кориснику представљени атрибутима: *address* (адреса корисника), *sign_date* (датум учлањења корисника), *warning* (индикатор слања опомена кориснику), *city* (место корисника), *zipcode* (поштански број), *phone* (број телефона) и *email* (e-mail адреса корисника). На основу везе са класом *USER_TYPES* идентификује се тип корисника. Класа *USERS* специјализује се на класе: *SINGLE* – за индивидуалне кориснике, *GROUPS* – за колективне кориснике и *INSTITUTIONS* – за институције размене.

Класа *SINGLE* представља индивидуалне кориснике библиотеке. Садржи атрибуте: *first_name* (име), *last_name* (презиме), *parent_name* (име родитеља), *gender* (пол), *occupation* (занимање), *title* (звање), *permission* (дозволе

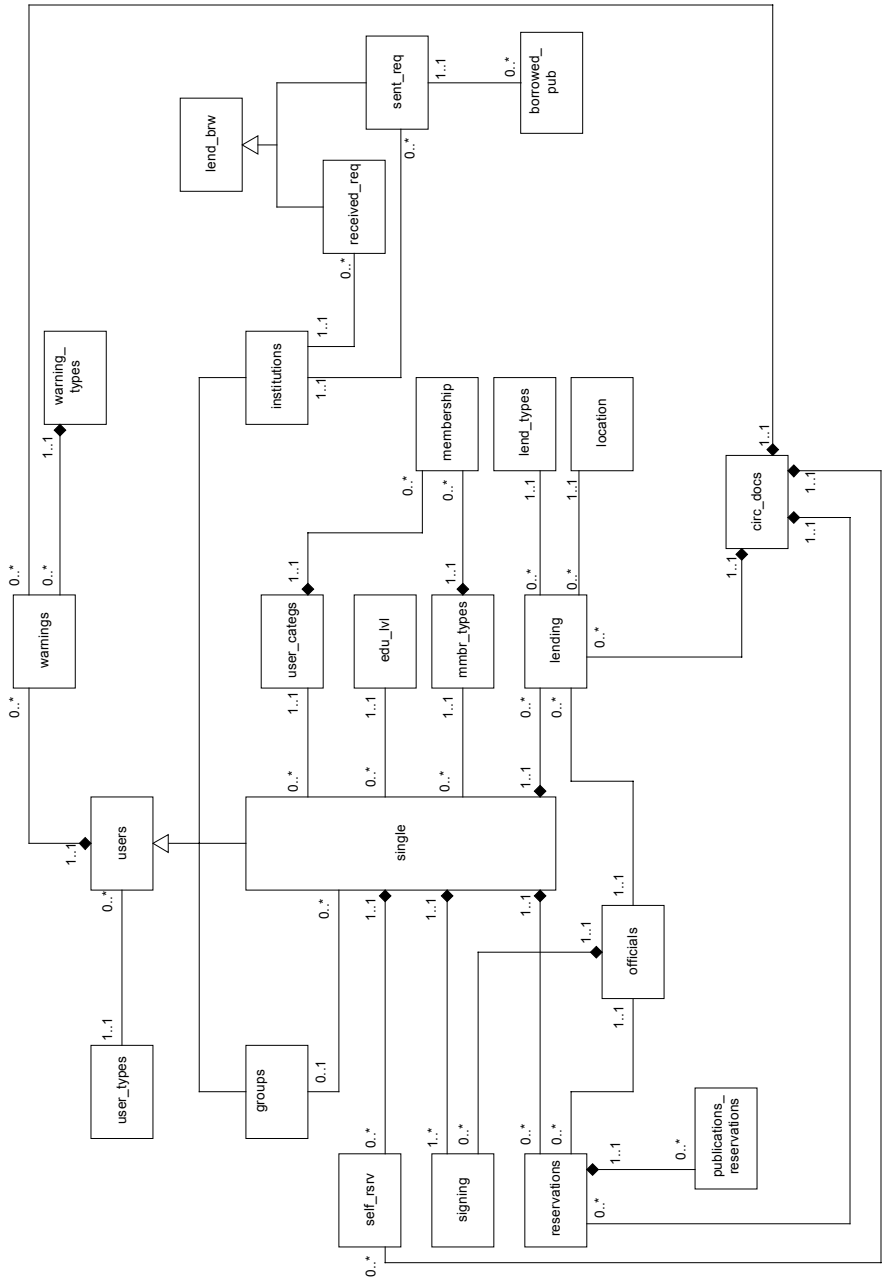
корисника за коришћење фонда библиотеке), *jmbg* (јединствени матични број грађана), *interests* (интересовања корисника за одређене врсте публикација), *note* (напомена), *index_no* (број индекса, за студенте), *doc_id* (ознака врсте документа за идентификацију – лична карта, пасош или избегличка легитимација), *doc_no* (број документа за идентификацију), *doc_city* (место издавања документа за идентификацију), *temp_addr* (пребивалиште – адреса), *temp_city* (пребивалиште – место), *temp_zip* (пребивалиште – поштански број), *temp_phone* (пребивалиште – број телефона), *organization* (установа у којој је корисник запослен), *org_address* (адреса установе у којој је корисник запослен), *org_city* (место установе у којој је корисник запослен), *org_zip* (поштански број установе у којој је корисник запослен) и *org_phone* (телефонски број установе у којој је корисник запослен). На основу везе са класом *EDU_LVL* идентификује се степен образовања корисника, на основу везе са класом *USER_CATEGS* идентификује се категорија корисника а на основу везе са класом *MMBR_TYPES* идентификује се врста учлањења корисника.

Класа *GROUPS* представља колективне кориснике библиотеке. Садржи атрибуте: *inst_name* (назив организације), *sec_address* (додатна адреса организације – улица и број), *sec_city* (додатна адреса – град), *sec_zip* (додатна адреса – поштански број), *sec_phone* (додатна адреса – број телефона), *fax* (број факса), *cont_email* (e-mail адреса за контакт), *cont_fname* (име особе за контакт), *cont_lname* (презиме особе за контакт) и *telex* (број телефакса).

Класа *INSTITUTIONS* представља институције за међубиблиотечку позајмицу, односно институције са којима библиотека врши размену библиотечке грађе. Садржи атрибуте: *inst_name* (назив институције), *cont_fname* (име особе за контакт) и *cont_lname* (презиме особе за контакт), *cont_phone* (број телефона за контакт), *cont_email* (e-mail адреса за контакт), *sec_phone* (додатни број телефона), *fax* (број факса) и *telex* (број телефакса).

Класа *EDU_LVL* представља шифарник степена образовања. Атрибут *id* јединствено идентификује степен образовања, док атрибут *name* представља опис степена образовања.

Класа *USER_CATEGS* представља шифарник категорија корисника библиотеке. Атрибут *id* јединствено идентификује категорију корисника, док атрибут *name* представља опис категорије корисника. Атрибут *titles_no* означава број публикација које корисник поједине врсте може да узме из библиотеке, а *period* максималан период на који се може позајмити публикација.



Слика 10.1. Модел података за коришћење библиотечке грађе

Класа *MMBR_TYPES* представља шифарник врста учлањења корисника библиотеке. Атрибут *id* јединствено идентификује врсту учлањења, док атрибут *name* представља опис врсте учлањења. Атрибут *titles_no* означава број публикација које корисник поједине врсте може да узме из библиотеке, а *period* максималан период на који се може позајмити публикација.

Класа *MEMBERSHIP* представља висну чланарине за кориснике библиотеке. На основу везе са класом *MMBR_TYPES* идентификује се врста учлањења, док се на основу везе са класом *USER_CATEGS* идентификује категорија корисника. Атрибут *mdate* (датум важења) јединствено одређује чланарину за одговарајући тип корисника и врсту учлањења. Класа садржи још и атрибуте *cost* (динарски износ чланарине) и *hard_curr* (износ чланарине у страној валути).

Продужење чланарине индивидуалних корисника библиотеке представљено је класом *SIGNING*. Она садржи атрибуте *cost* (динарски износ чланарине) и *hard_curr* (износ чланарине у страној валути). На основу везе са класом *SINGLE* идентификује се индивидуални корисник библиотеке, док се на основу везе са класом *OFFICIALS* идентификује библиотекар који је извршио продужење чланарине. Атрибут *sdate* представља датум продужења чланарине и јединствено идентификује продужење чланарине за одговарајућег корисника.

Класа *LOCATION* представља просторе у којима се врши задуживање корисника публикацијама. Атрибут *id* јединствено идентификује простор, док атрибут *name* представља назив простора.

Класом *LEND_TYPES* представљене су врсте задужења корисника публикацијама. Атрибут *id* јединствено идентификује врсту задужења, док атрибут *name* представља назив врсте задужења. Атрибут *titles_no* означава број публикација које корисник поједине врсте може да узме из библиотеке, а *period* максималан период на који се може позајмити публикација.

Класа *LENDING* представља задужење корисника библиотеке публикацијом. На основу везе са класом *SINGLE* идентификује се корисник библиотеке док се на основу везе са класом *CIRC_DOCS* идентификује публикација која се задужује. Атрибут *lend_date* (датум задужења) јединствено идентификује задужење одговарајућег корисника одабраном публикацијом. Атрибути *return_date* и *resume_date* представљају датум враћања односно датум продужења задужења. Класа *LENDING* има везу са класом *OFFICIALS* којом се идентификује службено лице које је извршило задужење и везу са класама *LEND_TYPES* и *LOCATION* којима се идентификује врста задужења и простор у коме је извршено задуживање корисника публикацијом.

Резервисање публикација за неког корисника врши се путем класе *RESERVATIONS*. Класа садржи атрибуте *rsrv_date* (датум резервисања), *notification*, *resume_date* (датум продужетка резервације). На основу везе са класом *SINGLE* се идентификује корисник који резервише публикацију, док се на основу везе са класом *CIRC_DOCS* идентификује публикација која се резервише. Атрибут *rsrv_date* јединствено идентификује резервацију одабране публикације за одговарајућег корисника. Класа такође има везу са класом *OFFICIALS* којом се идентификује библиотекар који је извршио резервацију.

Приликом резервисања, корисник поставља један или више услова које публикација треба да задовољи. Обично те услове задовољава више од једне публикације. Класа *PUBLICATIONS_RESERVATIONS* омогућава библиотекару да за корисника библиотеке резервише више публикација које задовољавају одређени критеријум. На основу везе са класом *RESERVATIONS* и вредности атрибута *ctlg_no* јединствено се идентификује конкретан примерак књиге.

Корисник може и сам извршити резервисање публикације, на пример путем Интернета. Та резервација је означена класом *SELF_RSRV*. Класа садржи атрибуте *rsrv_date* (датум резервисања), *notification* и *resume_date* (датум продужетка резервације). На основу везе са класом *SINGLE* се идентификује корисник који резервише публикацију, док се на основу везе са класом *CIRC_DOCS* идентификује публикација која се резервише. Атрибут *rsrv_date* јединствено идентификује резервацију одабране публикације за одговарајућег корисника.

Међубиблиотечка позајмица врши се путем класе *LEND_BRW*. Класа садржи атрибуте: *author* (аутор), *title* (наслов), *payment* (начин плаћања), *cost* (цена), *type* (тип), *pieces_no* (број публикација), *comments* (напомене) и *hard_curr* (износ цене у страној валути). Атрибут *id* јединствено идентификује позајмицу. Класа *LEND_BRW* специјализује се на класе *RECEIVED_REQ* – за примљене захтеве и *SENT_REQ* – за издате захтеве.

Класа *RECEIVED_REQ* представља примљене захтеве за међубиблиотечку позајмицу. Садржи атрибуте: *accept_date* (датум примања захтева), *institution* (опис институције), *lend_date* (датум издавања публикације на позајмицу) и *return_date* (датум враћања). На основу везе са класом *INSTITUTIONS* идентификује се институција са којом се врши позајмица.

Класа *SENT_REQ* представља издате захтеве за међубиблиотечку позајмицу. Садржи атрибуте: *accept_date* (датум узимања публикације на позајмицу), *institution* (опис институције), *send_date* (датум слања захтева) и *return_date* (датум враћања). На основу везе са класом *INSTITUTIONS* идентификује се институција од које се врши позајмљивање публикације.

Класа *BORROWED_PUB* представља позајмљену публикацију од неке институције за међубиблиотечку позајмицу. Атрибут *id* јединствено идентификује позајмљену публикацију. Класа садржи још и атрибуте: *catalogue_no* (инвентарни број), *author* (аутор публикације) и *title* (наслов публикације). На основу везе са класом *SENT_REQ* идентификује се послати захтев за позајмицу.

Класом *WARNINGS* представљене су опомене послате кориснику. Садржи атрибуте *wdate* (датум издавања опомене) и *returned* (ознака да ли је публикација враћена). Опомена се јединствено идентификује на основу типа опомене (веза са класом *WARNING_TYPES*), корисника коме је опомена послата (веза са класом *USERS*) и публикације коју је корисник задужио (веза са класом *CIRC_DOCS*).

Класом *WARNING_TYPES* представљене су врсте опомена које се могу слати корисницима. Атрибут *id* јединствено идентификује врсту опомене, док атрибут *wtext* садржи текст опомене.

10.2. Апликација Cirkulacija

Имплементација процеса коришћења библиотечке грађе извршена је у апликацији *Cirkulacija*, као део софтверског система БИСИС верзија 3. За имплементацију је коришћено Java окружење [5, 6] па је стога апликација независна од рачунарске платформе и система за управљање базама података који се користи. У овом поглављу је укратко описана функционалност имплементираних екранских форми апликације *Cirkulacija*, док је детаљно упутство за коришћење ове апликације дато у [7].

Главни мени апликације (слика 10.2) садржи следеће подменије:

- *Učlanjenje,*
- *Zaduženje i razduženje,*
- *Rezervacija,*
- *Pozajmica,*
- *Izveštaji* и
- *Sistem*.

У верзији 3.01 апликације *Cirkulacija* имплементирано је *Učlanjenje*, *Zaduženje i razduženje*, *Izveštaji* и *Sistem*.

Слика 10.2. Основни прозор апликације *Cirkulacija*

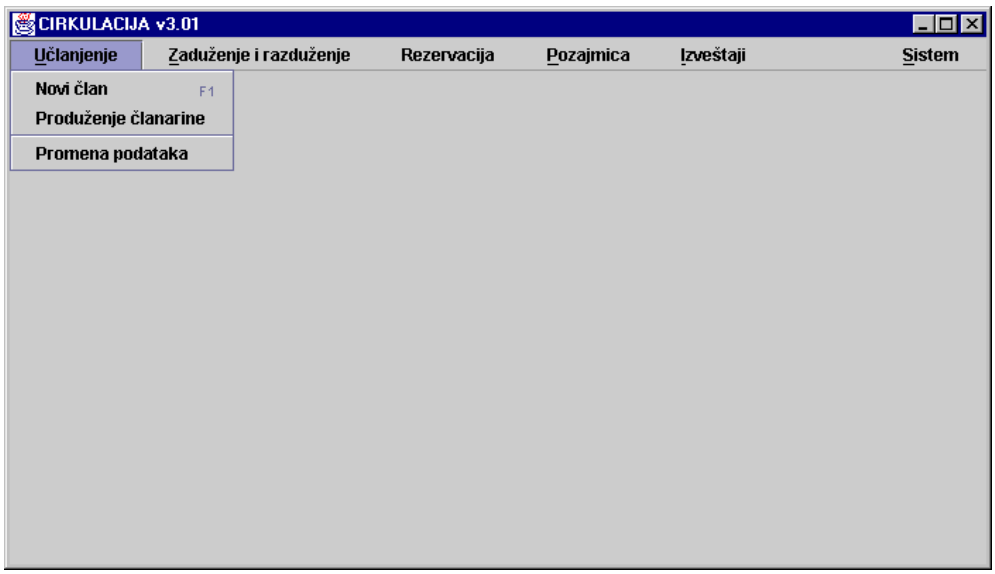
10.2.1. Учлањење

У оквиру менија *Učlanjenje* (слика 10.3) могуће је извршити три акције: учлањење новог члана библиотеке, продужење чланарине за индивидуалне чланове библиотеке и промена података чланова библиотеке.

Приликом учлањавања новог члана библиотеке потребно је најпре изабрати тип корисника који се учлањује у библиотеку: индивидуални члан, колективни члан или институција позајмице. Одабирањем типа корисника отвара се одговарајућа екранска форма за унос података.

Приликом промене података о члану библиотеке, најпре се проналази члан на основу броја чланске карте или неког од понуђених параметара за претраживање (име, презиме, ЈМБГ, институција ...), а затим се отвара одговарајућа екранска форма за ажурирање података.

Приликом учлањења новог члана односно промене података о члану библиотеке, врши се упис односно ажурирање података у одговарајућим табелама базе података, у зависности од типа корисника који се учлањује/ажурира. За индивидуалне кориснике, подаци се уносе/ажурирају у табелама класе *users* и *single*. За колективне чланове, подаци се уносе/ажурирају у табелама класе *users* и *groups*. За институције размене, подаци се уносе/ажурирају у табелама класе *users* и *institutions*.



Слика 10.3. Акције у оквиру менија *Učlanjenje*

Продужење чланарине може се вршити само за индивидуалне чланове библиотеке. Приликом продужења чланарине, додаје се нова врста у табелу *signing*.

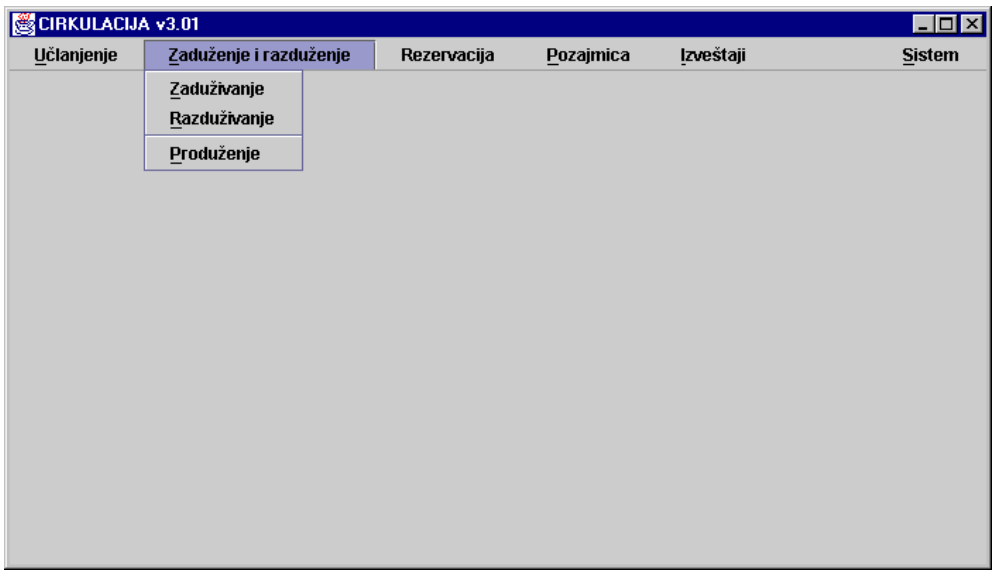
10.2.2. Задужење и раздужење

У оквиру менија *Zaduženje i razduženje* (слика 10.4) могу се извршити акције задуживања корисника публикацијама, раздуживање корисника и продуживање задужења корисника. Ове акције односе се само на индивидуалне чланове библиотеке.

У оквиру сваке од ових акција најпре се врши проналажење одговарајућег члана библиотеке – на основу задатог услова, проналази се корисник постављањем упита у табелу *single*.

Информације о задужењу, раздужењу односно продужењу задужења, уписују се у табелу *lending*. У тој табели се чува и историја задуживања, тако да се приликом раздужења корисника не брише одговарајућа врста те табеле, већ се у колону *return_date* уписује датум раздужења.

У оквиру акције задуживање, могуће је одштампати реверс кориснику библиотеке за издату публикацију. Такође, у оквиру акција раздуживање и продужење, могуће је и одштампати опомену кориснику, у случају да је прошао рок за враћање неке од публикација којима је задужен.



Слика 10.4. Акције у оквиру менија *Zaduženje i razduženje*

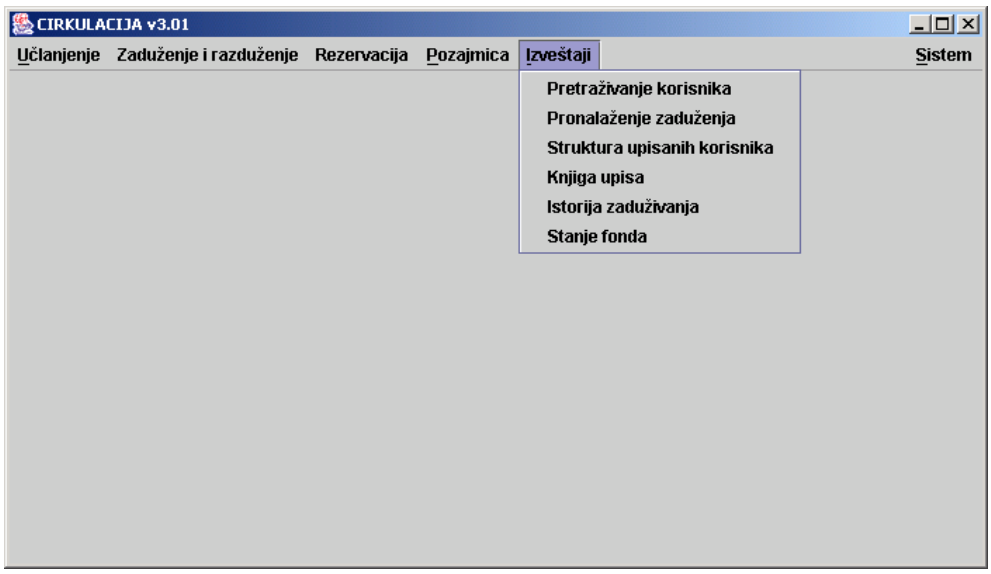
10.2.3. Извештаји

У оквиру менија *Izveštaji* (слика 10.5) налазе се следеће акције:

– *pretraživanje korisnika*,
– *pronalaženje zaduženja*,
– *struktura upisanih korisnika*,
– *knjiga upisa*,
– *istorija zaduživanja* и
– *stanje fonda*.

Pretraživanje korisnika се користи за проналажење информација о кориснику (у првом реду броја чланске карте) на основу података који су познати. Претраживање индивидуалних корисника може се вршити по следећим критеријумима:

– Презиме, Име,
– Презиме, Име родитеља, Име,
– Презиме,
– Име и
– ЈМБГ.



Слика 10.5. Акције у оквиру менија *Izveštaji*

Колективни корисници и институције позајмице могу се претраживати само на основу назива организације односно институције позајмице.

Проналажење корисника врши се претраживањем табела *users*, *single*, *groups* и *institutions*.

Pronalaženje zaduženja је извештај који за публикацију која је заузета проналази корисника библиотеке код кога се она налази. То се постиже претраживањем табеле *lending* на основу инвентарног броја публикације, а затим проналажењем података о кориснику библиотеке у табелама *users* и *single* на основу пронађеног броја чланске карте. У оквиру ове акције могуће је и одштампати опомену кориснику библиотеке код кога се налази тражена публикација.

Struktura upisanih korisnika је извештај који за одређени временски период приказује податке о броју корисника учлањених у библиотеку, по категоријама корисника и врсти учлањења.

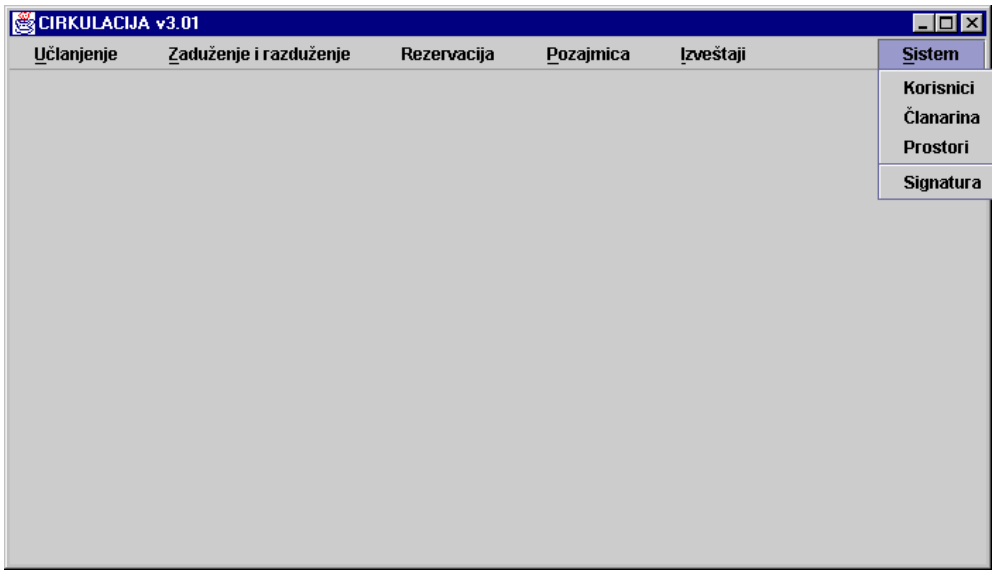
Knjiga upisa је дневни извештај који приказује податке о уписаним члановима библиотеке у току једног дана.

Istorija zaduživanja је извештај којим се за једну публикацију проналазе сви корисници код којих је она била на задужењу. Параметри за овај тип извештаја су: инвентарни број и период за који се проналазе задужења. Резултати се добијају претраживањем табеле *lending*.

Stanje fonda је извештај који приказује укупан број публикација у библиотеци, сортиран по типу публикације. Резултати се добијају претраживањем табеле *circ_docs*.

10.2.4. Систем

У оквиру менија *Sistem* налазе се акције за попуњавање шифарника потребних за рад у апликацији *Cikulacija* (слика 10.6). У оквиру групе *Korisnici* налазе се следећи шифарници: типови корисника (подаци из табеле *user_types*), категорије корисника (подаци из табеле *user_categs*), степени образовања (подаци из табеле *edu_lvl*) и врсте учлањења (подаци из табеле *mmbr_types*). У оквиру групе *Članarina* налази се шифарник чланарина (подаци из табеле *membership*). У оквиру групе *Prostori* налазе се шифарници: простори задуживања (подаци из табеле *location*) и врсте задужења (подаци из табеле *lend_types*).



Слика 10.6. Акције у оквиру менија *Sistem*

10.3. Извештавање у циркулацији

На основу анализе процеса коришћења библиотечке грађе у Библиотеци града Београда [1], уочене су две врсте извештаја који се очекују у систему:

- дневни и периодични статистички извештаји везани за кориснике библиотеке о њиховом броју и структури као и финансијким показатељима и
- извештаји који су везани за читаност библиотечке грађе по основним УДК групама и категоријама корисника.

У последњој верзији апликације *Cirkulacija* имплементирани су само статистички извештаји везани за кориснике библиотеке (прва врста извештаја).

10.3.1. Статистички извештаји о корисницима библиотеке

Прва врста извештаја користи податке из базе циркулације библиотечких докумената. У ове извештаје спадају:

- Периодични извештај о броју и структури уписаних корисника са финансијским показатељима,
- Књига уписа – дневни извештај уписаних чланова са билансом,
- Опомена,
- Опомена пред утужење и
- Структура уписаних корисника.

Периодични извештај о броју и структури уписаних корисника са финансијским показатељима је извештај који се даје за одређени временски период и садржи податке о броју корисника учлањених у библиотеку, по категоријама корисника и врсти учлањења корисника. Садржи и податке о финансијским показатељима, односно цену чланарине и укупан износ наплаћене чланарине.

Периодични извештај о броју и структури уписаних корисника
са финансијским показатељима

од дана: дд.мм.гггг до дана дд.мм.гггг

Категорија	Број чланова	Цена	Укупно динара
Индивидуални члан			
врста учлањења 1			
врста учлањења 2			
…			
Колективни члан			
УКУПНО			

Структура уписаних корисника је извештај који се даје за одређени временски период. Садржи број уписаних корисника библиотеке у наведеном периоду, по категоријама читалаца.

Структура уписаних корисника
период: од дд.мм.гггг. до дд.мм.гггг.

Категорија читалаца	Број
Деца до 14 година	
Ученици средњих школа	

Књига уписа је дневни извештај који садржи податке о уписаним члановима библиотеке у току једног дана, као и финансијски биланс наплаћене чланарине остварен у току дана.

КЊИГА УПИСА

Дневни извештај уписаних чланова са билансом
Датум: дд.мм.гггг **Лист бр. 1**

Ред. бр.	Бр.чл. карте	Име и презиме	Адреса	Пошта и место	Документ	Матични број	Врста учлањења

Категорија	Број чланова	Цена	Укупно динара
Индивидуални члан			
врста учлањења 1			
врста учлањења 2			
…			
Колективни члан			
УКУПНО			

Опомена и опомена пред утужење имају сличну форму. У заглављу се наводи назив и адреса библиотеке која издаје опомену, а затим подаци о књигама које нису враћене: наслов, аутор, инвентарни број, сигнатура и број

дана кашњења. Наводи се и износ казне који је потребно платити, за сваку књигу и укупно.

Опомена:

Назив библиотеке

Адреса библиотеке – место датум враћања дд.мм.гггг.

Увидом у нашу евиденцију установили смо да нисте у предвиђеном року вратили позамљене књигу-е:

Наслов	**Аутор**	**Инвентарни број**	**Сигнатура**	**Број дана кашњења**

Молимо вас да своју обавезу испуните у року од 3 (три) дана како не би дошло до примене других предвиђених мера, као и да приликом враћања књига извршите обавезу накокнаде трошкова у износу:

надокнада за прекорачење рока
дневно по свакој књизи: хх.хх дин.
надокнада за ПТТ трошкове: хх.хх дин.

до дд.мм.гггг. укупно динара

Број чланске карте
Име Презиме
Адреса
ПТТ Место

Напомена: подаци који следе исписују се само на опомени пред утужење
Име родитеља
ЈМБГ
Број личне карте, место издавања

10.3.2. Извештаји о читаности библиотечке грађе

Друга врста извештаја користи податке из две базе података: базе циркулације библиотечких докумената, и из базе библиографске грађе. У ове извештаје спадају:

- Картица књиге,
- Најчитаније књиге,
- Извештај о позајмљеним књигама по категоријама корисника,
- Најчитаније књиге југословенске књижевности,
- Најчитаније књиге из УДК,
- Статистика издатих и враћених књига по УДК,

- Статистика најчитанијих издавача,
- Неактуелне библиографске јединице,
- Читаност стране литературе за УДК и
- Статистика читалачких интересовања.

Извештај о позајмљеним књигама по категоријама корисника даје се за одређени временски период и за сваку категорију корисника наводи се број позајмљених наслова и број позајмљених примерака књиге, разврстан по основним УДК бројевима.

Извештај о позајмљеним књигама
по категоријама корисника
период: од дд.мм.гггг. до дд.мм.гггг.

	УДК						
	0		**1**		**...**	**9**	
Категорија корисника	**Број наслова**	***Број примерака***	**Број наслова**	***Број примерака***		**Број наслова**	***Број примерака***
Деца до 14 година							
Ученици средњих школа							

Најчитаније књиге југословенске књижевности је периодичан извештај који даје *n* (најчешће двадесет) најчитанијих наслова југословенске књижевности у наведеном периоду.

Најчитаније књиге југословенске књижевности
период: од дд.мм.гггг. до дд.мм.гггг.

Ред.бр.	**Аутор**	**Наслов**	**Година издања**
1			
...			
n			

Најчитаније књиге из УДК је периодичан извештај за један основни УДК који даје *n* (најчешће двадесет) најчитанијих наслова који припрадају том УДК, у наведеном периоду.

Најчитанијe књиге
из УДК (опис УДК)
период: од дд.мм.гггг. до дд.мм.гггг.

Ред.бр.	Наслов	Аутор	Број примерака
1			
...			
n			

Статистика издатих и враћених књига по УДК је периодичан извештај који за сваки основни УДК даје број издатих и број враћених књига (број наслова и број примерака).

Статистика издатих и враћених књига по УДК
период: од дд.мм.гггг. до дд.мм.гггг.

УДК	Издато књига		Враћено књига	
	Број наслова	Број примерака	Број наслова	Број примерака
0				
1				
2				
3				
5				
6				
7				
8				
9				

Картица књиге је периодичан извештај који за једну књигу даје податке о њеном издавању корисницима библиотеке и враћању, као и бројеве чланских карти чланова библиотеке који су књигу узимали на позајмицу.

Картица књиге
период: од дд.мм.гггг. до дд.мм.гггг.

Аутор: **Инвентарни број:**
Наслов: **Сигнатура:**
Издавач, место и година издања:

Ред.бр.	Датум издавања	Број чланске карте	Датум враћања

Најчитаније књиге је периодичан извештај који даје *n* (најчешће двадесет) најчитанијих наслова у наведеном периоду.

Најчитаније књиге
период: од дд.мм.гггг. до дд.мм.гггг.

Ред.бр.	Аутор	Наслов
1		
...		
n		

Статистика најчитанијих издавача је периодичан извештај у коме се наводи број публикација издатих на позајмицу и број публикација у фонду библиотеке, по издавачима.

Статистика најчитанијих издавача
период: од дд.мм.гггг. до дд.мм.гггг.

Ред.бр.	Издавач	Место издања	Обрађених публикација	Читаност

Неактуелне библиографске јединице је периодичан извештај који даје податке о публикацијама које нису узимане на позајмицу у наведеном периоду.

Неактуелне библиографске јединице
период: од дд.мм.гггг. до дд.мм.гггг.

Ред. бр.	Сигнатура	Инвентарни број	Аутор	Наслов	Издавач,место и година издања	УДК

Читаност стране литературе за УДК је периодичан извештај који даје *n* (најчешће двадесет) најчитанијих наслова стране књижевности који припадају наведеној основној УДК групи, у наведеном периоду.

Читаност стране литературе
за УДК (назив УДК)
период: од дд.мм.гггг. до дд.мм.гггг.

Ред.бр	Наслов	Број примерака
1		
...		
n		

Статистика читалачких интересовања је периодичан извештај којим се евидентирају сва потраживања књига која нису могла бити испуњена. Евидентира се разлог због ког корисник библиотеке није могао да позајми књигу – таква књига не постоји у фонду, или књига постоји у фонду али нема слободних примерака.

Статистика читалачких интересовања
период: од дд.мм.гггг. до дд.мм.гггг.

Ред. бр.	Број чланске карте	Аутор	Наслов	Број потраживања	Нема у фонду	Недовољан број примерка

10.4. Закључак

Коришћење библиотечке грађе представља један од сегмената рада сваке библиотеке. Статистички извештаји које нуди овај систем омогућавају брз увид у стање фонда библиотеке и лако праћење финансијских показатеља. Такође, извештаји о читаности публикација и читалачким интересовањима омогућавају праћење коришћења библиотечке грађе као и планирање набавке оних публикација које су заиста тражене.

У описаној апликацији *Циркулација* имплементирани су само неки делови овог сложеног процеса. Имплементирано је евидентирање корисника библиотеке, задужење, раздужење и продужење задужења, статистички извештаји о корисницима библиотеке као и екранске форме за ажурирање шифарника неопходних за рад апликације. Потребно је још имплементирати процес резервације публикација као и међубиблиотечку позајмицу. Такође, потребно је направити и извештаје о читаности библиотечке грађе, како би систем могао потпуније да одговори на све захтеве библиотекара.

Референце

[1] Т. Тошић, Ј. Срејић, Т. Анисијевић. *Коришћење библиографске грађе*. ИНФОТЕКА: Часопис за информатику и библиотекарство, Заједница библиотека универзитета у Србији, Београд, Септембар, 2000, стр. 31-38.

[2] T. Tošić. *Modeling of the Library Collection Units Reservation Process*. Proceedings of Abstracts of the XV Conference on Applied Mathematics PriM'02, Zlatibor, 2002, pp 19-20.

[3] G. Booch, J. Rumbaugh, I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley Longman, Inc., 1999.

[4] *Rational Rose*, http://www.rational.com

[5] B. Eckel. *Thinking in Java*, 2nd edition. 2000., http://www.EckelObjects.com

[6] *JavaTM 2 Platform, Standard Edition, v 1.3.1 API Specification.* http://java.sun.com/j2se/1.3/docs/api/

[7] Д. Сурла, З. Коњовић, Б. Пуповац, Б. Милосављевић, М. Видаковић, Т. Зубић, Т. Тошић. *Упутство за коришћење библиотечког софтверског система БИСИС вер. 3.* Група за информационе технологије, Нови Сад, 2003.

Поглавље 11

Интернационализација и локализација система БИСИС

М. Видаковић, Б. Милосављевић

Писање софтверских апликација чије је функционисање независно од земље и језика корисника и које је једноставно прилагодити конкретној држави или писму разматра се у оквиру проблема који се називају интернационализација и локализација [1]. Интернационализација апликације подразумева да се апликација напише тако да се једноставно може прилагодити употреби у различитим географским регионима и различитим језицима без интервенције на програмском коду апликације. Локализација представља процес прилагођавања програма за употребу у конкретном подручју (локалу). Локал је географски или политички специфично подручје унутар кога се користи исти језик.

У процесу моделирања и имплементације система БИСИС интернационализација је спроведена доследним коришћењем *Unicode* распореда [2] за унос и приказ вишејезичког текста, уз ослонац на стандардну подршку Java виртуелне машине за ове потребе.

Локализација апликација система БИСИС обухвата три сегмента:

- локализацију садржаја базе података,
- локализацију библиотекарских апликација и
- локализацију веб сајта за корисничко претраживање.

Предуслов да се локализација садржаја базе података може обавити јесте да коришћени систем за управљање базом података (СУБП) има могућност складиштења текста кодираног *Unicode* распоредом [2] као и одговарајућу подршку за *Unicode* у свом JDBC [3] интерфејсу. Сама локализација садржаја се потом може извести на два начина:

- превођењем потребног садржаја у базу података на дати језик, при чему се у бази података чува само једна верзија садржаја – оног на изабраном језику, и
- модификацијом шеме базе података тако да се у њој истовремено чувају исто подаци дати на више језика истовремено.

Друга варијанта локализације додатно усложњава развој шеме базе података, као и саме апликације која је користи. Прва варијанта локализације не захтева измене у постојећој шеми базе података, као и програмском коду апликација, али захтева чување посебне верзије садржаја базе података за сваки коришћени језик. Поред тога, са првом варијантом није могуће истовремено користити податке из базе на више језика у оквиру једне корисничке сесије. Систем БИСИС користи прву варијанту локализације базе података.

Локализација библиотекарских апликација, које имају форму класичних самосталних апликација које користе графички кориснички интерфејс, и локализација веб сајта за корисничко претраживање користе исте механизме за локализацију базиране на подршци за локализацију која је уграђена у стандардну библиотеку програмског језика Java. Ови механизми биће детаљно описани у другом одељку.

Поред решења за интернационализацију и локализацију апликација која се могу применити на највећи број светских језика, у развоју система БИСИС је посебна пажња посвећена специфичностима српског језика која потичу из равноправне употребе два писма у српском језику – латинице и ћирилице. Како се, према *Unicode* стандарду, текст писан ћирилицом и латиницом репрезентује различитим нумеричким вредностима, то ће претрага ћириличним упитом над текстуалним подацима записаних латиницом (и обрнуто) увек имати као резултат празан скуп погодака. Једна од особина система БИСИС је и могућност равноправне употребе ћириличног и латиничног писма у току претраживања текста на српском језику. Руковање текстом у оквиру система БИСИС које подржава ову функционалност описано је у наредном одељку.

11.1. Руковање вишејезичним текстом

11.1.1. Кодни распореди

За потребе рачунарске репрезентације текста у употреби су бројни стандарди који дефинишу одговарајуће кодне распореде знакова. Стандарде прописују тела која се баве стандардизацијом као што је ISO (*International Organization for Standardization*) или компаније као што је *Microsoft*. ISO

стандарди који дефинишу кодне распореде за репрезентацију текста припадају фамилији ISO 8859 [4, 5] и налазе се у употреби на Интернету. У питању су 8-битни кодни распореди фиксне ширине (*single byte fixed width*) од којих су у нашем окружењу најинтересантнији ISO 8859-2 [4] (источноевропска латиница) и ISO 8859-5 [5] (ћирилица). Компанија *Microsoft* у оквиру својих оперативних система није, нажалост, усвојила ове стандарде већ је дефинисала сопствене који се незнатно разликују од постојећих ISO стандарда. Међу њима су и кодне стране 1250 (источноевропска латиница) и 1251 (ћирилица) [6].

У размени вишејезичних текстова кодираних наведеним стандардима постоји више потешкоћа. Пре свега, ту је недостатак подршке за *све* стандарде на *свим* рачунарским платформама: на *Microsoft*-овим оперативним системима користе се кодни распореди које је дефинисао *Microsoft*, док се на *Unix*/*Linux* оперативним системима обично користе ISO кодни распореди. Под подршком за неки стандард подразумева се механизам који омогућава унос и приказ текста датог одговарајућим кодним распоредом у оквиру апликација и промену текућег коришћеног кодног распореда. Поред тога, постоји и недостатак одговарајућих фонтова који ће у оквиру графичких корисничких интерфејса обезбедити приказ текста датог у одговарајућим кодним распоредима.

Unicode стандард, кога је дефинисао *Unicode* конзорцијум а прихваћен је и као ISO стандард ISO 10646, полази од нешто другачијих принципа: тежи да обезбеди јединствени код за сваки знак, без обзира на коришћену рачунарску платформу, апликацију или језик. Основна идеја у његовој реализацији је да се за кодирање сваког знака користи 16 бита, а не 8 како је то обично случај. Употреба 16 бита омогућава истовремено представљање 2^{16} различитих знакова што је, претпоставља се, довољно за све језике који су данас у употреби. *Unicode*-ом кодиран текст може имати више репрезентација, од којих су нам од значаја UTF-8 и UTF-16. Сваки UTF (*Unicode Transformation Format*) представља алгоритам мапирања сваке *Unicode* скаларне вредности на низ бајтова. Инверзно мапирање је такође дефинисано.

Основна карактеристика UTF-8 формата је да је то једнобајтни распоред променљиве дужине (*single byte variable width*); различити знакови у UTF-8 распореду заузимају један, два или више бајтова. Конципиран је тако да знакови из ASCII скупа имају исте кодове и у UTF-8, чиме се поједностављује руковање *Unicode* текстом у старијим системима који нису предвиђени за тако нешто. Због тога се налази у најширој употреби управо на Интернету као један од могућих распореда у којима су дати текстуални садржаји, пре свега HTML [7] и XML [8] документи.

UTF-16 формат је оно што се често подразумева под именом *Unicode* – двобајтни распоред фиксне дужине (*double byte fixed width*), где је сваки знак представљен помоћу тачно два бајта. На овај начин могуће је приказати око 63000 знакова, док се помоћу механизма тзв. парова сурогата може представити додатних око милион знакова. Механизам сурогата није од интереса за ову тему. UTF-16 формат је погодан за интерну репрезентацију текста у оквиру рачунарских програма, па се тако и користи у оквиру *Java* и *.NET* окружења.

За потребе формирања описа и складиштења библиографске грађе библиотечки информациони систем БИСИС вер. 3 користи стандард UNIMARC [9], који представља међународни библиотекарски стандард из ове области, односно његову домаћу верзију YUMARC [10, 11, 12]. Овај стандард, између осталог, дефинише и сопствени кодни распоред који се користи за репрезентацију вишејезичног текста. У питању је комбинација неколико ISO стандарда. Основни концепт који је карактеристика овог кодног распореда је концепт *текућег писма*. Наиме, један исти код може представљати знакове различитих писама. Правилна интерпретација знака зависи од тога које писмо је тренутно постављено као текуће писмо. Текуће писмо се поставља у оквиру кодираног текста посебним једнобајтним ознакама. Уколико на самом почетку текста није дефинисано текуће писмо, подразумева се латиница. Текуће писмо важи као текуће све до наиласка на нову ознаку текућег писма или до краја текста. Кодирање знакова у оквиру UNIMARC-а је такво да он представља још један једнобајтни распоред променљиве дужине. YUMARC стандард обухвата ћирилично, латинично, грчко и старословенско писмо. Мапирање UNIMARC распореда на *Unicode* и обрнуто је дефинисано за све знакове који се налазе у оквиру оба стандарда, а имплементирано је у систему БИСИС верзија 3.

11.1.2. Складиштење докумената

Главна јединица обраде у систему БИСИС је UNIMARC запис, који представља машински читљив облик библиографске јединице. UNIMARC стандард дефинише формат намењен за складиштење и размену оваквих записа. Формат је секвенцијалне природе и прилагођен је складиштењу на различитим уређајима, укључујући данас доминантне хард дискове али и магнетне траке. За потребе физичког руковања подацима систем БИСИС користи релациону базу података, али и чува UNIMARC записе у њиховом оригиналном облику како би се поједноставила размена записа између различитих библиотечких институција. Као основне јединице информације

UNIMARC запис чува стрингове текста. Стрингови су кодирани сопственим UNIMARC кодним распоредом.

Следи пример једног стринга који представља текст *Petar Petrović Njegoš* (латиницом). Слика 11.1 представља дати стринг и његову кодирану репрезентацију у UNIMARC запису, где су вредности бајтова представљене хексадецимално. На самом почетку стринга налази се ознака која поставља латинично писмо као текуће писмо. Сви латинични знакови у овом примеру кодирани су помоћу једног бајта. Слика 11.2 представља еквивалентан текст *Петар Петровић Његош* дат ћирилицом. Сви знакови осим слова *Њ* кодирани су помоћу једног бајта, док је велико слово *Њ* кодирано помоћу два бајта. Код за ћирилично слово *Њ* је идентичан низу кодова за латинична слова *n* и *j*. Аналогно томе, кодови за ћирилична слова *Љ* и *Џ* су идентични низовима кодова за латинична слова *lj* и *dž*. Захваљујући тој особини UNIMARC кодног распореда текстови писани ћирилицом и латиницом разликују само у ознаци текућег писма. У примеру са слика 11.1 и 11.2 то су ознаке за латиницу (кодирано 1C) и ћирилицу (кодирано 1D).

1C	50	65	74	61	72	20	50	65	74	72	6F	76	69	C9	20	4E	6A	65	67	6F	E9
[lat]	P	e	t	a	r		P	e	t	r	o	v	i	ć		N	j	e	g	o	š

Слика 11.1. Пример кодирања латиничног текста у UNIMARC запису

1D	50	65	74	61	72	20	50	65	74	72	6F	76	69	C9	20	4E	6A	65	67	6F	E9
[cir]	П	е	т	а	р		П	е	т	р	о	в	и	ћ		Њ		е	г	о	ш

Слика 11.2. Пример кодирања ћириличног текста у UNIMARC запису

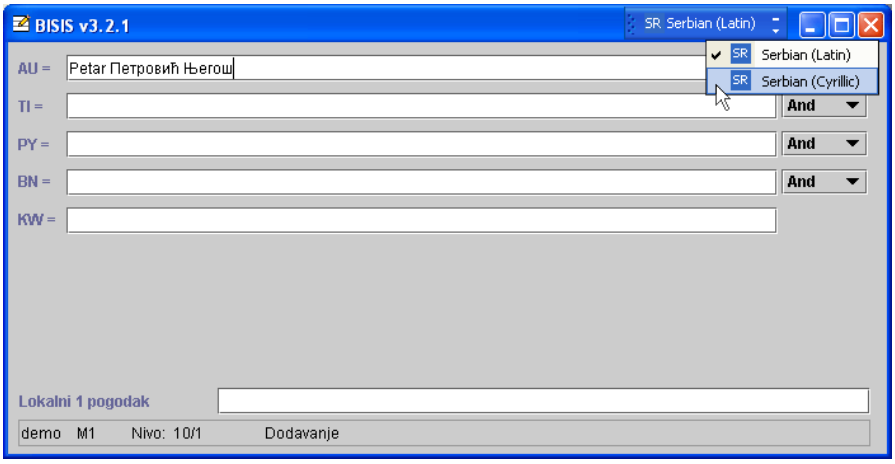
Уколико се ознаке текућег писма уклоне из кодираног текста добиће се идентични стрингови. Такви стрингови се могу ефикасно програмски поредити при чему се обезбеђује да се као једнаки стрингови појављују и парови стрингова датих у различитим писмима.

Приликом складиштења UNIMARC записа у базу података, осим складиштења самог записа који се чува у оригиналном облику, формирају се и индекси – структуре података чија намена је да обезбеде брзо претраживање текста. У оквиру индекса текст се складишти управо тако да се ознаке текућег писма уклоне из кодираног текста. Поред тога, текст се смешта у још једном облику: из њега се акцентована слова српске латинице (*ččšžđ*) заме-

њују својим неакцентованим верзијама (*cszdj*) како би се корисницима који немају могућност уноса ових слова омогућила претрага. Индексирање и претраживање текста у оквиру текст сервера БИСИС-а детаљно је приказано у [13].

11.1.3. Приказ и унос текста

Библиотечки информациони систем БИСИС је Интернет/интранет апликација. У оквиру интранета њени корисници су библиотекари који се превасходно баве обрадом библиографске грађе односно формирањем UNIMARC записа. За ту намену развијена је посебна клијент-апликација која омогућава претраживање и обраду библиографске грађе која је детаљно описана у Поглављу 4. Клијент апликација представља класичну апликацију која користи графички кориснички интерфејс. Развијена је у програмском језику Java који, између осталог, карактере чува у *Unicode* распореду односно UTF-16 формату. Програмско окружење језика Java користи стандардне механизме оперативног система за избор кодног распореда и распореда тастатуре приликом уноса текста, чиме је омогућено да се у оквиру Java-апликације може унети произвољан знак обухваћен *Unicode* стандардом. Слика 11.3 приказује форму за претраживање базе библиографске грађе у оквиру апликације. Слика илуструје могућност уноса *Unicode* знакова различитих писама помоћу стандардног механизма за избор тастатуре.



Слика 11.3. Унос текста са више писама у оквиру Java-апликације

Java апликације за потребе приказа текста користе стандардне *TrueType* фонтове (*Arial* на *Windows* платформи, *Lucida Sans* на *Linux*-у, итд.) који у себи садрже описе словних ликова мапиране по *Unicode* стандарду. Уместо

ових фонтова могуће је користити било који други *Unicode* фонт у *TrueType* или *OpenType* формату.

Јавни приступ библиотечком фонду у оквиру система БИСИС је могућ путем модула за корисничко претраживање приказаном у Поглављу 9. Овај модул представља веб презентацију библиотеке која омогућава претраживање библиографске грађе и њено преузимање. HTML странице које чине ову презентацију кодиране су по UTF-8 распореду *Unicode* стандарда. Овај распоред подржавају сви савремени веб читачи. За коректан приказ странице потребно је да на клијент-рачунару постоје одговарајући *Unicode* фонтови. Веб странице за корисничко претраживање користе такве фонтове који су јавно доступни (*Verdana, Arial*, односно *Helvetica*). Слика 11.4 представља пример странице која представља листу пронађених записа који садрже текст дат у више писама.

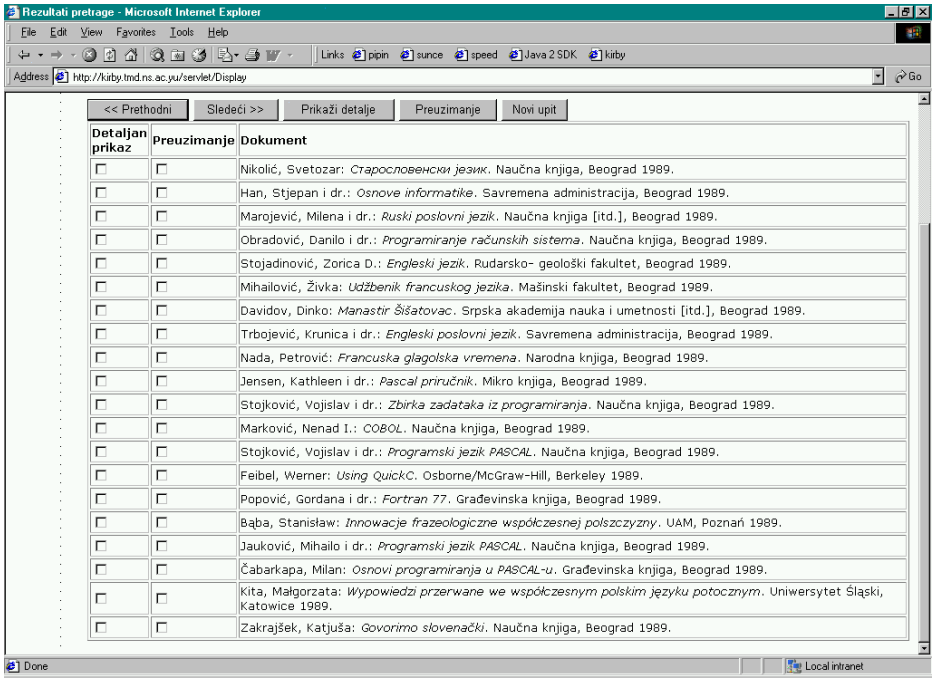
Унос текста у оквиру модула за корисничко претраживање се своди на попуњавање поља форме за постављање упита претраге. На жалост, различити веб читачи на различит начин кодирају знакове које уноси корисник у поља HTML форми. *Netscape* читачи на *Windows* платформи, на пример, сав текст кодирају према текућој системској кодној страни. Са друге стране, *Microsoft Internet Explorer* користи *Unicode* распоред за све знаке који не улазе у састав текуће кодне стране. Због тога је потребно извршити анализу унетог текста и података о карактеристикама веб читача на страни веб сервера да би се добијени подаци правилно интерпретирали. Подаци који су потребни за правилну интерпретацију се налазе у оквиру заглавља сваког захтева који веб читач упућује веб серверу по HTTP протоколу [14].

11.1.4. Претраживање

На основу карактеристика складиштења текста, његовог приказа на екрану и метода уноса описаних у претходним одељцима можемо анализирати процес претраживања у оквиру система БИСИС. На страни корисника, у оквиру интранет клијент-апликација, сав текст се приказује у *Unicode* (UTF-16) распореду. Такође се сав текст и уноси по UTF-16 распореду. Приликом постављања упита за претрагу клијент-апликација формира одговарајући текстуални упит на основу попуњених поља форме. Тако формирани упит шаље се текст серверу, посебном програмском модулу чија намена је да омогући складиштење, индексирање и претраживање UNIMARC записа описаном у Поглављу 8. По пријему упита текст сервер извршава следеће операције:

– конвертује добијени *Unicode* текст у UNIMARC распоред,

– из њега избацује све ознаке текућег писма (тиме се цео текст конвертује у латиницу) и
– такав текст користи приликом претраживања индекса.



Слика 11.4. Пронађени записи који користе више писама

Овакво претраживање ће као поготке пронаћи све записе у којима се тражени текст појављује, без обзира на писмо којим су записи формирани. Када клијент затражи одређени запис који је пронађен током претраге, сервер ће му послати оригинални UNIMARC запис у коме су очуване све информације о коришћеном писму. Клијент ће потом извршити конверзију из UNIMARC распореда у *Unicode* за потребе локалне репрезентације записа и његовог приказа на екрану.

Са стране корисничког претраживања упити се текст серверу шаљу од стране веб сервера. Задатак веб сервера је да формира одговарајући упит на основу попуњених поља у HTML форми веб странице. Формирање овог упита обухвата конверзију текста кога је послао веб читач у *Unicode* распоред. Такав упит се упућује текст серверу који се понаша као у претходно описаном случају. Веб сервер, по пријему резултата претраживања, преузима

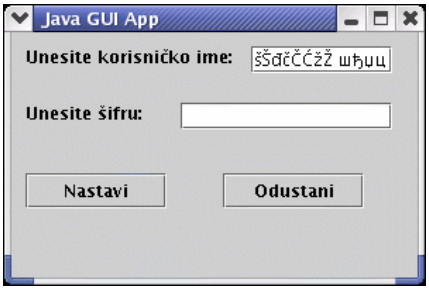
пронађене UNIMARC записе у њиховом оригиналном облику, текстуалне информације у њима конвертује из UNIMARC кодног распореда у UTF-16 распоред, и потом те информације уграђује у веб страницу конверзијом из UTF-16 у UTF-8 распоред. Коначан резултат кога види веб читач је страница са *Unicode* текстом где се као формат користи UTF-8, који је и стандардан формат за веб окружење.

11.2. Локализација корисничког интерфејса

11.2.1. Подршка Unicode стандарду у Swing и JSP апликацијама

Графичке Java апликације написане употребом Swing библиотеке могу се ефикасно интернационализовати и локализовати зато што све Swing контроле подржавају *Unicode*, а од верзије 1.4 програмског језика Java, омогућен је унос *Unicode* карактера са тастатуре на свим платформама. Унос је имплементиран тако да зависи од системског мапера тастатуре, и подржава оне језике који су инсталирани у систему. Са друге стране, приказ *Unicode* знакова директно зависи од имплементације *Unicode* фонтова и подршке *Unicode* кодне стране у GUI подсистему. Слика 11.5 илуструје унос и приказ *Unicode* карактера у Java GUI апликацији.

Интернационализација JSP апликација подразумева такође подршку за исправан унос и приказ *Unicode* карактера. Чињеница да су JSP апликације заправо веб апликације и да корисник приступа њима употребом веб читача значи да постоји могућност да се *Unicode* карактери не прикажу коректно, као и могућност да се унети карактери не преузму од читача на коректан начин. Ако читач подржава *Unicode* стандард, тада је потребно придржавати се следећих правила.



Слика 11.5. Унос и приказ *Unicode* карактера

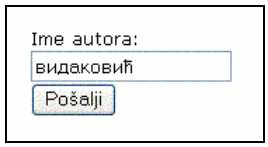
У JSP страници је потребно унети следећи скриптлет:

```
<% response.setContentType("text/html; charset=utf-8"); %>
```

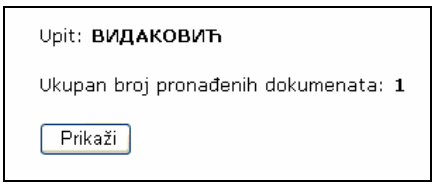
У заглављу генерисане HTML странице потребно је да постоји следећи таг:

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset="utf-8">
```

Сваки испис *Unicode* стрингова се аутоматски преводи у UTF-8 стрингове који се затим приказују на екрану читача. Са друге стране, сваки унос података у корисничким формама се преноси у JSP странице као UTF-16 стринг.



Слика 11.6. Унос *Unicode* текста у HTML форми



Слика 11.7. Приказ *Unicode* текста у HTML страници

Слике 11.6 и 11.7 илуструју унос и приказ *Unicode* текста у JSP апликацији. Сав текст који се уноси и обрађује у JSP странама третира се као UTF-16 стринг, а у веб читачу као UTF-8 стринг. То значи да се стрингови у JSP страници не морају ручно конвертовати у UTF-8 стрингове због исписа, нити да се унети подаци са странице морају конвертовати из UTF-8 стрингова у UTF-16. Конверзију аутоматски обавља апликациони сервер. Овде је битно напоменути да ће коректан приказ и унос у читачу функционисати само ако читач правилно рукује UTF-8 стринговима. Тренутно то подржавају *Internet Explorer*, *Netscape Navigator*, *Mozilla*, *Opera* и *Konqueror* читачи.

11.2.2. Класе за интернационализацију и локализацију у програмском језику Java

Интернационализација програма подразумева да се подаци могу приказати независно од језика и подручја у којем се користи. За идентификацију подручја и језика користи се *Locale* класа. Објекти *Locale* класе се конструишу једним од три конструктора:

```
Locale(String language);
Locale(String language, String country);
Locale(String language, String country, String variant);
```

Први конструктор идентификује језик, други идентификује земљу и језик, а трећи језик, земљу и варијанту (која је обично зависна од платформе или веб читача). Прва два параметра морају бити валидни ISO кодови језика и земље [15, 16].

Java платформа не захтева употребу истог локала кроз цео програм. Исто тако, спецификација локала није обавезна и програми који нису интернационализовани користе подразумевани локал кога поставља JVM (Java виртуелна машина) и који представља локал платформе на којој се извршава Java програм.

Осим *Locale* класе, за смештање локално зависних података користи се *ResourceBundle* механизам. Овај механизам се ослања на *ResourceBundle* апстрактну класу и њене класе наследнице: *PropertyResourceBundle* и *ListResourceBundle*. Ове две класе имплементирају смештај локално зависних података у виду асоцијативне листе парова (кључ, вредност), где се на основу кључа идентификује локално зависан податак.

PropertyResourceBundle класа је задужена за рад са **.properties* датотекама које садрже локално зависне стрингове. *ListResourceBundle* је такође апстрактна класа, а њене класе наследнице садрже листу са паровима (кључ, вредност) локално зависних података који не морају бити само стрингови (као што је био случај са *PropertyResourceBundle* класом).

Добијање локално зависних података из *ResourceBundle* система се заснива на *ResourceBundle fallback detection* механизму, који подразумева навођење локала којим се идентификује одговарајућа инстанца складишта (**.properties* датотека или **.class* датотека за наследнице *ListResourceBundle* класе). Овај механизам ће пробати да пронађе датотеку са задатим називом који је проширен кодовима језика, земље и варијанте, а ако је не нађе, потражиће мање специјализовану датотеку (језик и земља). Ако ни њу не нађе, потражиће још мање специјализовану датотеку (означену само кодом језика), а ако ни њу не нађе, потражиће датотеку у чијем називу се налази подразумевани локал. Ако ни њу не нађе, потражиће датотеку основног имена, без локалних карактеристика. Ако ни њу не нађе, изазваће изузетак *MissingResourceException*. Редослед претраживања датотека је приказан следећим примером:

```
ButtonLabel_sh_YU_WINDOWS
ButtonLabel_sh_YU
ButtonLabel_sh
ButtonLabel_en_US
ButtonLabel_en
ButtonLabel
```

Осим стрингова, који представљају најочигледније ресурсе зависне од локала, зависни су и бројеви, новчане вредности, датум и време. Локализација стрингова се најједноставније обавља уз помоћ *PropertyResourceBundle* класе, као на листингу 11.1.

```
Locale currentLocale = new Locale("sh", "YU", "WINDOWS");
ResourceBundle labels =
 ProperyResourceBundle.getBundle(
 "ButtonLabel", currentLocale);
String computer = labels.getString("s1");
```

Листинг 11.1. Пример локализације стрингова

Класа *PropertyResourceBundle* класа захтева постојање одговарајуће **.properties* датотеке као на листингу 11.2.

```
# Ovo je
# ButtonLabel_sh_YU_WINDOWS.properties
# datoteka
s1 = kompjuter
s2 = memorija
s3 = ku\u0107i\u0161te
```

Листинг 11.2. Пример .properties датотеке

Локализација стрингова, иако изгледа најједноставније, заправо представља најкомплекснији облик јер директно зависи од граматике циљног језика. У том случају се може користити *MessageFormat* класа која омогућује параметризовано форматирање података, као у примеру са листинга 11.3.

```
MessageFormat formatter = new MessageFormat("");
formatter.setLocale(currentLocale);
formatter.applyPattern(template));
String output = formatter.format(messageArguments);
```

Листинг 11.3. Параметризовано форматирање података

Ова класа омогућује употребу шаблона, где се унутар реченице идентификују променљиви делови, а затим се читава реченица генерише заменом променљивих делова стварним параметрима, форматираним на задати начин. Осим овога, програмски језик Java садржи и класу *ChoiceFormat*, која омогућава додељивање формата опсегу бројева, чиме се може обезбедити граматичка исправност реченица при различитим опсезима вредности променљивих делова. За форматирање бројева и новчаних вредности користи се *NumberFormat* класа. За додатно форматирање децималних бројева користи се *DecimalFormat* класа. За форматирање датума и времена користи се *DateFormat* класа. За специфично форматирање користи се *SimpleDateFormat* класа.

11.2.3. Поступак интернационализације и локализације Java програма

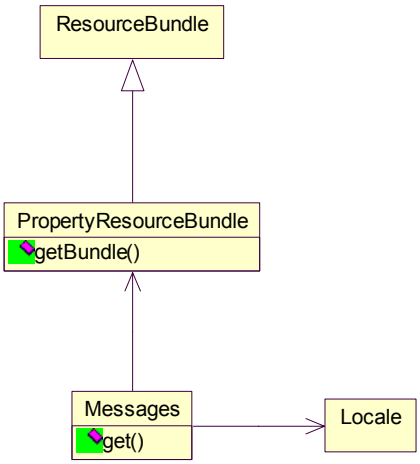
Интернационализација и локализација Java програма подразумева модификацију кода тако да се прилагођавање локалном језику и писму обавља без накнадне интервенције програмера. Прилагођавање кода се своди на издвајање језички зависног садржаја у репозиторијуме изван кода и имплементацију механизма који ће из репозиторијума узимати садржај који одговара локалном језику. Да би се то постигло, потребно је урадити следеће кораке:

- идентификовати језички зависне податке,
- изоловати текст за превођење у **.properties* датотеке,
- имплементирати одговарајући механизам који обавља аутоматизовано превођење и
- заменити сваку појаву стринга позивом одговарајућег механизма који ће вратити локално зависан стринг.

Један предлог механизма за аутоматско превођење стрингова је дат на слици 11.8. која приказује дијаграм класа за реализацију механизма аутоматског превођења стрингова. Механизам се заснива на статичкој методи *get* класе *Messages* која враћа преведени стринг на основу кључа и задатог језика. Произвољна класа која жели да користи овај механизам треба да позове *get* методу и као резултат добиће преведени стринг. Класа *Messages* дата је на листингу 11.4.

```
public class Messages {
  public static String get(String msgName) {
    return rb.getString(msgName);
  }
  private static ResourceBundle rb;
  static {
    Messages dummy = new Messages();
    INIFile iniFile = new INIFile(dummy.getClass().
      getResource(/com/gint/app/bisis/editor/editor.ini"));
    String locale = iniFile.getString("general", "locale");
    rb = PropertyResourceBundle.getBundle(
      "com.gint.app.bisis.editor.Messages",
      new Locale(locale, locale));
  }
}
```

Листинг 11.4. Класа *Messages*



Слика 11.8. Дијаграм класа *Messages* механизма

Класа *Messages* има статички део кода, који се извршава приликом првог коришћења ове класе. Овај код учитава текући локал из конфигурационе датотеке, и на основу њега креира објекат класе *Locale*. Овим је дефинисан локал који ће се користити у апликацији. На основу овог локала, учитава се одговарајућа **.properties* датотека у којој се налазе преведени стрингови. Статичка метода *get* враћа преведени стринг на основу кључа задатог параметром *msgName*.

Дефиниција кључа подлеже правилима којима се омогућује једнозначно и једноставно препознавање текста у листингу, на основу текста кључа. Текст кључа почиње називом програма, након чега следи назив класе. После назива класе следи тип текста и његов скраћени назив. Ако је тип текста садржај графичке контроле, њен назив се наводи. Ако је у питању текст који се налази унутар произвољне методе, ставља се назив методе, а ако је атрибут, ставља се назив атрибута. Следе примери неколико кључева:

```
BISISAPP_LOGINDLG_LABELPASS=Password:
BISISAPP_LOGINDLG_BUTTONCANCEL=Cancel
BISISAPP_MAINFRAME_STATUSADD=Add
```

11.3. Закључак

Библиотечки информациони систем БИСИС, у оквиру својих функција, омогућава равноправно коришћење свих писама за потребе писања текста за које постоји одговарајућа системска подршка (распореди тастатура и фонто-

ви). Као основа имплементације која подржава ову особину користи се стандардни *Unicode* кодни распоред. Сва интеракција корисника са системом се одвија путем овог кодног распореда, било из интранет апликација или путем јавно доступних веб презентација. Са друге стране, за потребе интерне репрезентације и складиштења текста систем користи кодни распоред дефинисан UNIMARC стандардом. Употреба овог стандарда омогућава ефикасну конверзију различитих писама и обједињено претраживање ћирилице и латинице. Као коначан резултат систем БИСИС обезбеђује равноправну употребу ћирилице и латинице за унос и приказ текста, као и у процесу претраживања где ће постављање упита коришћењем ћирилице и латинице дати исте резултате, који обухватају документе писане помоћу оба писма.

Предложени механизам за локализацију Java апликација који је примењен на систему БИСИС није специфичан за овај систем већ је опште применљив за локализацију Java апликација које имају графички кориснички интерфејс или веб сајт.

У оквиру пројекта TEMPUS JEP 16114 извршена је локализација система БИСИС за енглески, немачки, грчки, македонски и српски (ћирилица и латиница) језик.

Референце

[1] *Java ™ 2 SDK Internationalization Overview*. Sun Microsystems. http://java.sun.com/j2se/1.4.2/docs/guide/intl/

[2] *The Unicode Standard, Version 3.1*. Unicode Consortium, 2001. http://www.unicode.org

[3] J. Ellis, L. Ho, M. Fisher. *JDBC 3.0 Specification*. Sun Microsystems, 2001. http://java.sun.com/j2se/

[4] ISO/IEC 8859-2:1987, *Information Processing – 8-bit Single Byte Coded Character Sets – Part 2: Latin Alphabet No. 2*. International Organization for Standards, http://www.iso.ch

[5] ISO/IEC 8859-5:1988, *Information Processing – 8-bit Single Byte Coded Character Sets – Part 5: Latin/Cyrillic Alphabet*. International Organization for Standards, http://www.iso.ch

[6] Microsoft Corporation, http://www.microsoft.com

[7] D. Raggett, A. Le Hors, I. Jacobs. *HTML 4.01 Specification*. W3C Consortium, 1999. http://www.w3.org/TR/html4/

[8] T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maler. *Extensible Markup Language (XML) 1.0 (Second Edition)*. W3C Recommendation, 2000. http://www.w3.org/TR/REC-xml

[9] *UNIMARC Manual: Bibliographic format*, *International Federation of Library Association and Institutions*. IFLA Universal Bibliografic Control and International MARC Programme, New Providence, London 1994.

[10] *COMARC/R Format*. COBISS, Univerzitetni institut informacijskih znanosti, IZUM, Maribor 1992.

[11] B. Lazarević, redaktor. *Formiranje i pretraživanje baza podataka u Sistemu naučnih i tehnoloških informacija Srbije*. Ministarstvo za nauku i tehnologiju Republike Srbije, Beograd, 1996.

[12] Т. Вулић, Г. Билић. *Ретроспективна конверзија каталошких листића у формат YUMARC*. Институт за математику у Новом Саду, Едициија „Рачунари у универзитетској пракси“ бр. 5, Нови Сад, 1997.

[13] Б. Милосављевић. *Текст сервер UNIMARC записа*. Магистарска теза, Факултет техничких наука, Нови Сад, 1999.

[14] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. *Hypertext Transfer Protocol – HTTP/1.1*. IETF RFC 2616. http://www.ietf.org/rfc/rfc2616.txt

[15] ISO 639 – *Language Codes*. http://www.ics.uci.edu/pub/ietf/http/related/iso639.txt

[16] ISO 3166 – *Country Codes*. http://www.din.de/gremien/nas/nabd/iso3166ma/codlstp1/en_listp1.html

Поглавље 12

Мрежна инфраструктура система БИСИС

М. Керац, И. Нејгебауер

Функционисање дистрибуираних информационих система заснованих на клијент-сервер моделу зависи од мрежне инфраструктуре која даје подршку том информационом систему. TCP/IP фамилија протокола доминантно се користи приликом реализације дистрибуираних информационих система. Систем БИСИС, као дистрибуирани информациони систем који се заснива на клијент-сервер моделу, користи TCP/IP фамилију протокола. Мрежна инфраструктура обухвата физички ниво, ниво преноса података, мрежни ниво и транспортни ниво TCP/IP фамилије протокола. У поглављу су описани физичка и логичка архитектура рачунарске мреже за подршку библиотечког информационог система БИСИС. Предложен је начин повезивања локалне рачунарске мреже библиотеке са Интернетом и описана је организација права приступа појединим модулима библиотечког информационог система БИСИС.

12.1. Локална рачунарска мрежа библиотеке

12.1.1. Дефиниција и функције

Локална рачунарска мреже библиотеке је мрежа за пренос података намењена повезивању приступних прикључних места у библиотеци намењених пружању мрежних сервиса особљу библиотеке.

12.1.2. Физичка архитектура локалне рачунарске мреже библиотеке

Физичка архитектура локалне рачунарске мреже библиотеке обухвата:

– комуникациона чворишта:

  - локацију;
  - комуникациону опрему;
- комуникационе путеве за које је дефинисана пропусна моћ, преносни медијум и трасе полагања;
- корисничка прикључна места;
- осталу опрему.

12.1.3. Полазне поставке

Физичка архитектура локалне рачунарске мреже библиотеке треба да обезбеди:

- неометан проток података у оквиру библиотеке,
- отпорност на отказе критичних сегмената мреже,
- функционалну аутономност логичких целина мреже у условима отказа,
- могућност једноставне промене физичке конфигурације мреже и
- неометано логичко (ре)конфигурисање мреже.

12.1.3.1. Неометан проток података

Неометан проток података остварује се на бази коректне процене саобраћаја на мрежи. Процена саобраћаја подразумева:

- уочавање сегмената мреже који трпе највећи саобраћај,
- уочавање сегмената мреже који преносе временски критичне податке и
- процену очекиваног просечног саобраћаја на сегментима мреже.

На основу процењеног саобраћаја димензионишу се пропусна моћ комуникационих канала и бира се комуникациона опрема.

Са становишта интензитета саобраћаја издвајају се два нивоа комуникационих путева:

- основна магистрала (магистрала првог нивоа) која трпи највећи саобраћај и повезује локације у библиотеци које генеришу највећи саобраћај (чворишта);
- комуникациони путеви за повезивање крајњих прикључних тачака мреже библиотеке.

12.1.3.2. Отпорност на отказе критичних сегмената мреже

Да би се остварила отпорност на отказе критичних сегмената мреже потребно је:

- уочити критичне сегменте;
- одабрати одговарајући начин повезивања увођењем алтернативних комуникационих путева.

Магистрални комуникациони путеви представљају критичне сегменте.

На робусност мреже библиотеке утичу следећи фактори:

- тип преносника;
- топологија мреже;
- исправно функционисање капиталних комуникационих уређаја.

Могући типови преносника за магистралне комуникационе путеве су различити типови оптичких влакана (мултимодна или мономодна) или бакарни парични каблови (категорије 5, 5e)[2]. Сви предвиђени типови преносника се могу користити за пренос података на брзинама до 1Gb/s. При употреби бакарних паричних каблова мора се водити рачуна о ограничењима која се односе на максималну удаљеност активних компоненти (до 100m), максималну брзину протока (до сада 1Gb/s) и осетљивост на утицаје спољашњег електромагнетног зрачења. Преносни путеви изведени оптичким кабловима остварују максималне удаљености до 3 km, брзине преноса могу бити знатно веће од 1Gb/s и много мање су осетљиви на утицаје спољашње средине (потпуно су отпорони на спољашња електромагнетна зрачења).

С друге стране, топологија мреже утиче на робусност мреже, јер добро одабрана топологија сама по себи повећава степен робусности мреже. Узимајући у обзир важеће стандарде који се односе на комуникационе системе и рачунарске мреже топологија звезде намеће се као природан начин организовања мреже. Оваква топологија, уз правилан избор броја бакарних каблова и/или влакана у оптичком преносном путу, омогућује, поред повећања пропусне моћи, и повећање робусности.

На робусност мреже пресудно може да утиче отказ капиталних комуникационих уређаја, те је неопходно предвидети разуман ниво редунданције и квалитета комуникационих уређаја, који обезбеђује прихватљив ниво функционалности мреже.

12.1.3.3. Функционална аутономност логичких целина мреже у условима отказа

Функционална аутономност у условима отказа подразумева несметано комуницирање учесника унутар једне од логичких целина у условима отказа, независно од стања мреже ван те логичке целине. Функционална аутономност постиже се доследном применом топологије звезде на свим хијерархијским нивоима.

12.1.3.4. Могућност промене физичке конфигурације мреже

Ова могућност подразумева лако физичко проширивање мреже новим целинама као и једноставно одвајање делова мреже који својом нефункционалношћу ометају саобраћај на мрежи. Постиже се топологијом звезде и примењеним комуникационим уређајима.

12.1.3.5. Неометано логичко (ре)конфигурисање мреже

Неометано логичко (ре)конфигурисање мреже подразумева могућност динамичког успостављања логичких целина према потреби. Постиже се механизмом адресирања елемента мреже и конфигурисањем DNS (*Domain Name Service*) [1] сервера у складу са жељеном логичком архитектуром, као и коришћењем комуникационе опреме која подржава употребу виртуелних LAN-ова (*Local Area Network*).

12.1.4. Комуникациона чворишта локалне рачунарске мреже библиотеке

Библиотека може да буде самостална институција или део веће институције. У првом случају, следећа класификација односи се на библиотеку као целину, а у другом на институцију у оквиру које се библиотека налази.

Комуникациона чворишта локалне рачунарске мреже хијерахијски су организована и деле се у три групе:

- Комуникационо чвориште за целу локалну мрежу – КЧМ;
- Комуникациона чворишта за локалне мреже објеката институције – КЧО;
- Комуникациона чворишта за крајње приступне тачке локалне рачунарске мреже – КЧП.

Основне особине група чворишта су:

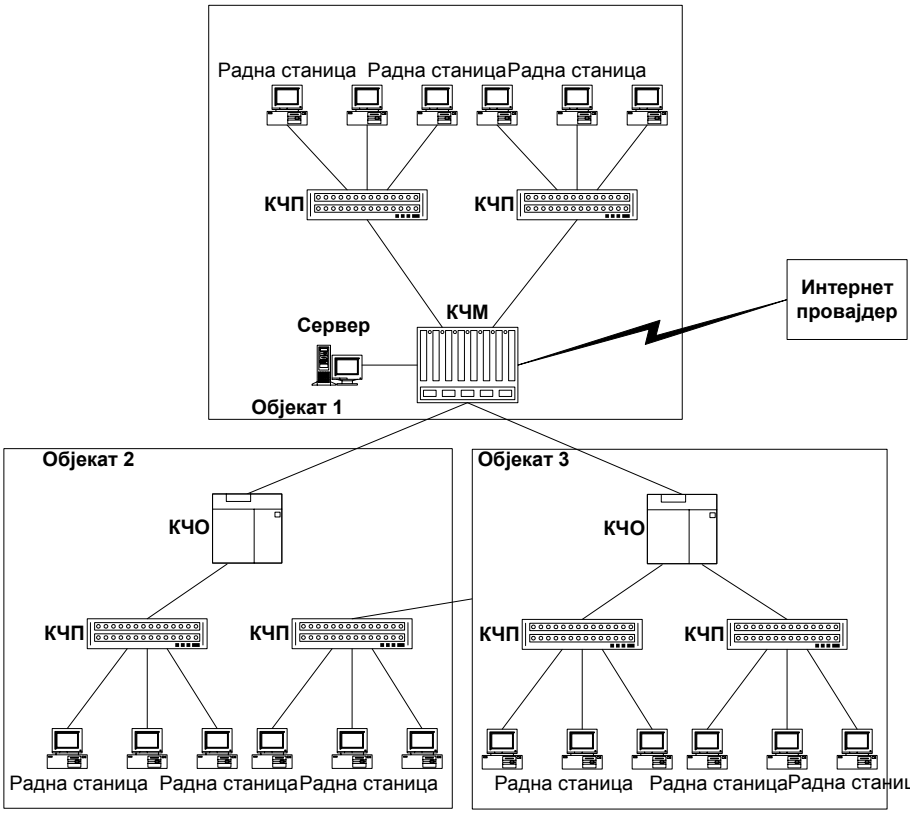
- Комуникационо чвориште за целу локалну мрежу – КЧМ. КЧМ су чворишта која повезују више КЧО, у случају да је институција смештена у више објеката или више КЧП, у случају да је институција смештена у само једном објекту. У КЧМ се налази активна опрема која повезује активну опрему смештену у КЧО/КЧП и сервере задужене за пружање интерних сервиса институције. Поред наведене активне комуникационе опреме, у КЧМ се налази активна опрема намењена за повезивање са Интернет провајдером. У КЧМ се смешта и пасивна опрема од које се пружају комуникациони путеви до КЧО/КЧП и до Интернет провајдера.
- Комуникациона чворишта за локалне мреже објеката институције – КЧО. КЧО су чворишта која повезују више КЧП и која се повезују са КЧМ. У КЧО се налази активна опрема која повезује активну опрему смештену у КЧП и која се повезује са активном опремом у КЧМ и пасивна опрема од које се пружају комуникациони путеви до КЧП и на којој се завршава комуникациони пут од КЧМ до КЧО. КЧО постоје само у случају да је институција смештена у више објеката. У случају да је институција смештена само у једном објекту, функцију КЧО преузима КЧМ.
- Комуникациона чворишта за крајње приступне тачке локалне рачунарске мреже – КЧП. КЧП су чворишта која повезују крајња корисничка прикључна места и која су повезана са КЧО, у случају да је институција смештена у више објеката, или са КЧМ у случају да је институција смештена у једном објекту. У КЧП се налази активна опрема која повезује корисничке радне станице и која је повезана са активном опремом КЧО/КЧМ. Од пасивне комуникационе опреме, смештене у КЧП, пружају се комуникациони путеви ка крајњим приступним тачкама и завршавају се комуникациони путеви од КЧО/КЧМ до КЧП.

12.1.4.1. Локација комуникационог чворишта

Приликом избора локација за комуникациона чворишта потребно је поштовати следеће принципе:

- локација за комуникационо чвориште треба да обезбеди могућност постављања кабловског развода који испуњава све захтеве прописане стандардима који се односе на структурирано каблирање [2],

– локација за комуникационо чвориште мора да буде просторија са обезбеђеним контролисаним приступом ради физичког обезбеђења активне и пасивне опреме,
– избор локације за комуникационо чвориште не сме да доведе до значајног повећања трошкова везаних за постављање кабловског развода.



Слика 12.1. Физичка архитектура локалне мреже

12.1.4.2. Функционална спецификација активне комуникационе опреме

Како се локације за комуникациона чворишта могу налазити у канцеларијским просторима и просторијама са јавним приступом (читаоница), сва активна комуникациона опрема која се налази у комуникационим чвориштима мора имати следеће особине:

– опрема посебно намењена за употребу у канцеларијским просторима,

- задовољава све међународне стандарде који се односе на употребу комуникационе опреме у оваквим условима (*FCC Class A digital device (part 15 of the FCC rules), CE Class A digital device (European Low Voltage Directive 73/23/EEC and EMC Directive 89/336/EEC as amended by European Directive 93/68/EEC)*).

Функционална спецификација активне комуникационе опреме у КЧМ. Основне карактеристике КЧМ су:

- КЧМ трпи највећи саобраћај локалне рачунарске мреже.
- Отказ у КЧМ је критичан по питању основне функционалности локалне рачунарске мреже.
- У КЧМ се дефинише логичка архитектура локалне рачунарске мреже.
- У КЧМ се налази опрема намењена за повезивање са Интернет провајдером.

Анализом карактеристика КЧМ дефинише се функционална спецификација активне комуникационе опреме у КЧМ.

Функционална спецификација активне комуникационе опреме намењене за повезивање са КЧО/КЧП – комуникациони уређај са функцијом свичовања (свич) који има:

- Минимално 12 портова који подржавају брзине рада од 100Mb/s или 1000 Mb/s,
- Портове који су:
  - стандардни RJ45, намењени за повезивање са бакарним паричним кабловима. Преко RJ45 портова повезивали би се сервери, комуникациона опрема намењена за повезивање са Интернет провајдером и активна комуникациона опрема из КЧО/КЧП у случају да се КЧО/КЧП налазе на удаљености мањој од 90m кабловске трасе од КЧМ.
  - стандардни SC, намењени за повезивање оптичких влакана. Преко SC портова повезивала би се активна комуникациона опрема из КЧО/КЧП.
- Стандардни серијски интерфејс намењен за конфигурисање уређаја,
- Могућност конфигурисања преко стандардних мрежних сервиса (*Telnet*, *Web*),
- Подршку за SNMP (*Simple Network Management Protocol*),
- Подршку за VLAN (*Virtual LAN*),

- L3 подршку, која, у случају могућности касније имплементације на уређају, није обавезна у првој фази реализације локалне рачунарске мреже и
- Могућност проширења броја портова.

Функционална спецификација активне комуникационе опреме намењене за повезивање са Интернет провајдером – комуникациони уређај са функцијом рутирања који има:

- Минимално један *Ethernet* RJ45 порт који подржава брзине рада од 10 Mb/s или 100 Mb/s,
- Минимално један ISDN BRI порт или минимално један серијски порт који подржава брзине рада до 2Mb/s,
- Стандардни серијски интерфејс намењен за конфигурисање уређаја,
- Могућност конфигурисања преко стандардних мрежних сервиса (*Telnet*, *Web*),
- Подршку за SNMP и
- Подршку за OSPF (*Open Shortest Path First*) протокол за динамичко рутирање.

Функционална спецификација активне комуникационе опреме у КЧО. Основне карактеристике КЧО су следеће:

- КЧО трпи највећи саобраћај локалне рачунарске мреже једног објекта.
- Отказ у КЧО је критичан по питању основне функционалности локалне рачунарске мреже једног објекта.
- У КЧО се налази опрема намењена за повезивање са КЧМ. Веза са КЧМ се реализује у зависности од географске позиције КЧО у односу на КЧМ.
- Ако се објекти налазе на удаљености мањој од 2km, КЧО се са КЧМ може повезати оптичким каблом.
- Ако се објекти налазе на удаљености већој од 2km, КЧО се са КЧМ може повезати употрбом xDSL уређајем са *Bridge* модулом. Овакав начин повезивања захтева изнајмљену паричну линију између објеката.

Анализом карактеристика КЧО дефинише се функционална спецификација активне комуникационе опреме у КЧО.

Функционална спецификација активне комуникационе опреме намењене за повезивање са КЧП – комуникациони уређај са функцијом свичовања (свич) који има:

- Минимално 12 портова који подржавају брзине рада од 100 Mb/s или 1000 Mb/s,
- Портове који су:
  - Стандардни RJ45, намењени за повезивање са бакарним паричним кабловима. Преко RJ45 портова повезивала би се активна комуникациона опрема из КЧП у случају да се КЧП налазе на удаљености мањој од 90m кабловске трасе од КЧО и комуникациона опрема намењена за повезивање са КЧМ ако услови диктирају употребу xDSL уређаја.
  - стандардни SC, намењени за повезивање оптичких влакана. Преко SC портова повезивала би се активна комуникациона опрема из КЧП и активна комуникациона опрема из КЧМ у случају да постоје услови.
- Стандардни серијски интерфејс намењен за конфигурисање уређаја,
- Могућност конфигурисања преко стандардних мрежних сервиса (*Telnet*, *Web*),
- Подршку за SNMP,
- Подршку за VLAN и
- Могућност проширења броја портова.

Функционална спецификација активне комуникационе опреме намењене за повезивање са КЧМ у случају да се објекти налазе на већој удаљености од 2km – фамилија комуникационих уређаја xDSL који има:

- Минимално 1 порт који подржавају брзине рада од 10 Mb/s или 1000 Mb/s,
- Минимално 1 порт који подржава xDSL линију на брзинама већим од 1Mb/s,
- Подршку за рад у *Bridging* режиму,
- Могућност конфигурисања преко стандардних мрежних сервиса (*Telnet*, *Web*),
- Подршку за SNMP и
- Стандардни серијски интерфејс намењен за конфигурисање уређаја.

Функционална спецификација активне комуникационе опреме у КЧП. Основне карактеристике КЧП су:

- Мрежни саобраћај у КЧП није критичан и
- Отказ у КЧП је критичан по питању основне функционалности само једног дела локалне рачунарске мреже.

Анализом карактеристика КЧП дефинише се функционална спецификација активне комуникационе опреме у КЧП.

Функционална спецификација активне комуникационе опреме у КЧП – комуникациони уређај са функцијом свичовања (свич) који има:

- Минимално 12 портова који подржавају брзине рада од 10 Mb/s или 100 Mb/s и минимално један порт који подржава брзине рада од 100 Mb/s или 1000 Mb/s,
- Портове који су:
  - стандардни RJ45, намењени за повезивање са бакарним паричним кабловима. Преко RJ45 портова повезивале би се корисничке радне станице и активна комуникациона опрема из КЧО у случају да се КЧП и КЧО налазе на удаљености мањој од 90m кабловске трасе од КЧО корисничке радне станице.
  - стандардни SC, намењени за повезивање оптичких влакана. Преко SC портова повезивала би се активна комуникациона опрема из КЧО.
- Подршку за SNMP и
- Могућност проширења броја портова.

12.1.4.3. Функционална спецификација пасивне комуникационе опреме

Сва пасивна комуникациона опрема која се налази у комуникационом чвориштима мора да задовољава међународне стандарде који се односе на постављање кабловске инфраструктуре у пословним објектима [2].

Функционална спецификација пасивне комуникационе опреме у КЧМ/КЧО/КЧП:

- *Patch* панели са одговарајућим бројем RJ45 портова категорије 5 или 5e и одговарајућим бројем SC портова;
- *Patch* каблови са RJ45 конекторима категорије 5 или 5e;
- Оптички *Patch* каблови са SC конекторима.

12.1.5. Комуникациони путеви

Комуникациони путеви у локалној рачунарској мрежи могу се поделити у три групе:

- Магистрални комуникациони путеви, који спајају КЧМ са КЧО/КЧП и КЧО са КЧП. Магистрални комуникациони путеви могу се реализовати са бакарним паричним кабловима ако је дужина кабловске

трасе магистралног комуникационог пута мања од 90m или оптичким кабловима.

- Резервни комуникациони путеви, који утичу на повећану редудантност локалне мреже. Резервни комуникациони путеви се постављају између КЧП и КЧП, КЧО и КЧО, КЧП и КЧО, КЧП и КЧМ, КЧО и КЧМ. Резервни комуникациони путеви могу се реализовати са бакарним паричним кабловима ако је дужина кабловске трасе магистралног комуникационог пута мања од 90m или оптичким кабловима.
- Приступни комуникациони путеви, који повезују крајње корисничке прикључне тачке са КЧП. Приступни комуникациони путеви се изводе бакарним паричним кабловима.

12.1.5.1. Функционална спецификација магистралних комуникационих путева

- Минимално два бакарна парична UTP (*Unshielded Twisted Pair*) кабла категорије 5 или 5e, у случају да се користе бакарни каблови.
- Минимално један оптички кабел са 6 мултимодних оптичких влакана, оптимизованих за рад у прозору од 1300 nm, у случају да се користе оптички каблови.
- Сви каблови се постављају у ПВЦ каналнице или спуштене плафоне или подигнуте патосе или у наменске просторе за полагање каблова у зидовима.

12.1.5.2. Функционална спецификација резервних комуникационих путева

- Минимално два бакарна парична UTP кабла категорије 5 или 5e, у случају да се користе бакарни каблови.
- Минимално један оптички кабел са 6 мултимодних оптичких влакана, оптимизованих за рад у прозору од 1300 nm, у случају да се користе оптички каблови.
- Сви каблови се постављају у ПВЦ каналнице или спуштене плафоне или подигнуте патосе или у наменске просторе за полагање каблова у зидовима.

12.1.5.3. Функционална спецификација приступних комуникационих путева

- Бакарни парични UTP кабел категорије 5 или 5е до сваког прикључног места
- Сви каблови се постављају у ПВЦ каналнице или спуштене плафоне или подигнуте патосе или у наменске просторе за полагање каблова у зидовима.

12.1.6. Корисничка прикључна места

Приступни комуникациони путеви завршавају се на корисничким прикључним местима. Корисничка прикључна места су прикључне кутије опремљене утичницама за спајање корисничких радних станица.

12.1.6.1. Функционална спецификација корисничких прикључних места

- Назидна или узидна прикључна кутија са местом за две утичнице
- RJ45 утичнице категорије 5 или 5е.

Крајња корисничка прикључна места могу бити:

- пасивна прикључна места, у потпуности покривена пасивном инфраструктуром, али нису прикључена на портове активне опреме у одговарајућем КЧП и
- активна прикључна места, у потпуности покривена пасивном инфраструктуром и прикључена на портове активне опреме у одговарајућем КЧП.

12.1.7. Остала опрема

Сва комуникациона чворишта морају бити смештена у посебне ормане намењене за смештај комуникационе опреме. У орманима је потребно обезбедити изворе напајања у складу са националним стандардима који се односе на постављање система за напајање електричном енергијом.

12.1.7.1. Функционална спецификација ормана за смештај комуникационе опреме

- Ормани намењени за смештај комуникационе опреме ширине 19”
- Могућност уградње вентилационог система у ормане
- Ормани димензија:
  - 42 U, за КЧМ регионалног центра,

- 18 U, за КЧМ и КЧО,
- 12 U, за КЧП.

12.2. Веза ка другим мрежама и Интернету

12.2.1. Повезивање локалне мреже библиотеке на Интернет

Ако је библиотека самостална институција, њена веза ка Интернету ће такође бити самостална. Оваква веза је могућа и у случају да библиотека припада другој институцији, мада је очекивана ситуација да се веза остварује преко матичне институције. У оба случаја веза ка Интернету мора минимално да задовољи следеће услове.

Библиотека се на Интернет повезује преко најповољнијег локалног Интернет провајдера који је у стању да пружи услугу повезивања која одговара техничким условима потребним за коректно функционисање Библиотечког информационог система. Веза са Интернет провајдером остварује се преко јавних телекомуникационих сервиса. Минималан капацитет везе са Интернет провајдером је 64 kb/s. Прегледом стања јавне комуникационе мреже утврђено је да су на располагању следећи телекомуникациони сервиси:

- семиперманентни ISDN базни приступ,
- *Frame Relay* са минималним CIR од 128 kb/s и
- синхроне попречне серијске везе.

За потребе библиотеке, без обзира на начин повезивања, мора се обезбедити бар једна стална јавна IP адреса.

12.2.2. Међусобно повезивање локалних рачунарских мрежа библиотека

Прављење приватне мреже за међусобно повезивање библиотека је технички непотребно и финансијски неисплативо. Везе између локалних рачунарских мрежа библиотека остварују се преко глобалног Интернета.

12.3. Минимални захтеви логичке архитектуре мреже библиотеке

Имплементација логичке архитектуре треба да се изврши применом:

- плана употребе јавног IP адресног простора који је на располагању библиотеци,
- хијерархијског приватног IP адресирања у оквиру целокупне мреже,
- хијерахијске организације имена у DNS конфигурацији мреже и
- технологије виртуелних локалних рачунарских мрежа (*ВЛРМ*) у оквиру локалне мреже институције.

Описана логичка архитектура се заснива на примени IP фамилије протокола [3].

12.3.1. План употребе јавног IP адресног простора

Приликом израде плана употребе јавног IP адресног простора потребно је за Библиотечки информациони систем обезбедити:

- бар једну јавну IP адресу, која би служила као приступна тачка Web интерфејсу за корисничко претраживање библиотечких података;
- бар једну јавну IP адресу као полазну тачку за повезивање са удаљеним библиотечким системима;
- бар једну јавну IP адресу за административни приступ библиотечком систему у случају да се одржавање система не обавља локално.

Све наведене функције могу се обављати преко исте јавне IP адресе. Ако услови налажу технички је изводљиво и конфигурисање библиотечког система тако да користи само адресе из приватног IP адресног простора, али то решење захтева сложенију конфигурацију других мрежних уређаја.

12.3.2. Хијерархијско приватно IP адресирање

Ако институција у оквиру које функционише библиотека има свој адресни план, библиотека се мора прилагодити том плану. Пожељно је да се за потребе библиотеке обезбеди посебан блок IP адреса који би се налазио у сопственом *broadcast* домену, чиме се смањује могућност да нестабилност или аномалија на другим сегментима локалне мреже утичу на рад библиотечког система.

Ако је библиотека самостална институција, организационим јединицама у оквиру библиотеке потребно је доделити одвојене блокове приватног адресног простора [4] у засебним логичким сегментима, чиме се смањује могућност да нестабилност или аномалија на другим сегментима локалне мреже утичу на рад библиотечког система.

12.3.3. Хијерахијска организација имена

Све активне компоненте библиотечког система треба да имају препознатљива имена која указују на њихову припадност библиотечком систему. Пожељно је да организација имена компоненти библиотечког система прати модел организације имена у DNS систему [1].

12.3.4. Виртуелне локалне рачунарске мреже

Виртуелне локалне рачунарске мреже могу се користити као средство за одвајање мрежног саобраћаја у локалној рачунарској мрежи библиотеке. Примера ради за библиотеку која је самостална институција могла би се направити следећа подела у логичке сегменте:

- *ВЛРМ библиотечког особља*. У овој ВЛРМ треба да се налазе радне станице библиотекара.
- *ВЛРМ читаонице*. У овој ВЛРМ треба да се налазе радне станице у јавно доступним читаоницама библиотеке.
- *ВЛРМ административног особља*. У овој ВЛРМ треба да се налазе радне станице запослених у административним службама.

12.4. Организација права приступа појединим модулима система

БИСИС је модуларан систем чији су модули споља доступни корисницима преко TCP сервиса:

- Систем за управљање релационом базом података (РСУБП). Овај систем ће бити доступан преко TCP портова који зависе од изабраног програмског пакета за управљање релационом базом података.
- Web интерфејс за корисничко претраживање који је доступан преко стандардног TCP порта 80 (Web).
- Модул за узајамну каталогизацију који је доступан преко регистрованог Java RMI TCP порта 1099 (RMI).
- Систем за безбедни удаљени терминалски приступ, преко стандардног SSH TCP порта 22 (SSH).

Корисници БИСИС система могу се поделити у неколико категорија:

- Библиотечко особље матичне библиотеке (Локални).
- Библиотечко особље удаљених библиотека (Удаљени).
- Удаљено административно особље (Администратори).

– Некласификовани корисници Интернета (Остали).

Права приступа сервисима треба да буду усклађена са категоријом корисника тако да свака категорија може користити сервисе који су јој намењени у складу са табелом 12.1.

		Категорија корисника			
		Локални	Удаљени	Администратори	Остали
Сервиси	РСУБП	✓		✓	
	Web	✓	✓	✓	✓
	RMI	✓	✓	✓	
	SSH			✓	

Табела 12.1. Матрица права приступа сервисима БИСИС-а

Права приступа дефинишу се применом пакетског филтрирања на мрежном и транспортном нивоу и аутентификацијом за оне сервисе који подржавају те механизме. Пакетски филтери могу се дефинисати на мрежним уређајима и на серверском систему БИСИС-а. Аутентификација се остварује у појединим апликативним модулима БИСИС-а.

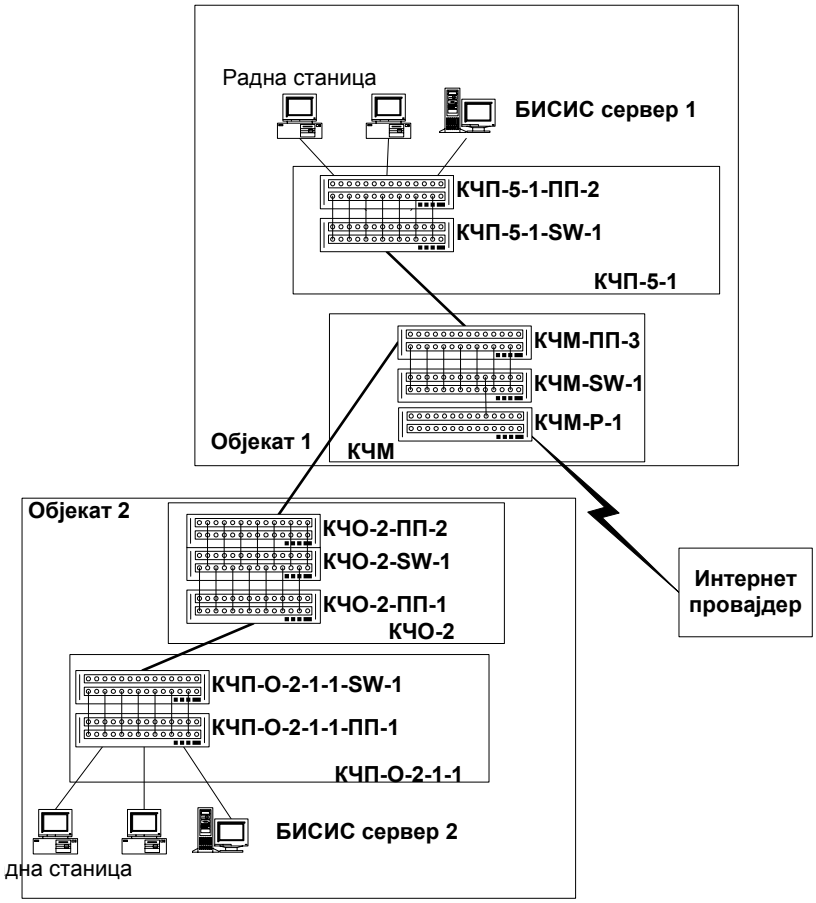
12.5. Пример

У датом примеру описана је мрежна инфраструктура две библиотеке у оквиру једне институције. Полазне претпоставке:

- Оба сервера БИСИС-а су доступна за претраживање од стране корисника Интернета.
- Оба сервера БИСИС-а допуштају међусобну узајамну катологизацију.
- Оба сервера БИСИС-а имају могућност преузимања записа један од другог.
- БИСИС сервер Библиотеке 1 допушта могућност преузимања записа класификованим библиотекама са Интернета.
- БИСИС сервер Библиотеке 2 не допушта могућност преузимања записа преко Интернета.
- Библиотечко особље библиотеке 1 може да приступа систему за РСУБП само на серверу библиотеке 1.
- Библиотечко особље библиотеке 2 може да приступа систему за РСУБП само на серверу библиотеке 2.

12.5.1. Физичка архитектура

Физичка архитектура система из примера дата је на слици 12.2.



Слика 12.2. Физичка архитектура локалне мреже

12.5.1.1. Библиотека 1

Библиотека 1 налази се у склопу централног објекта институције. Двоструке корисничке прикључне кутије намењене за повезивање радних станица библиотечког особља библиотеке 1 и сервера БИСИС-а библиотеке 1 налазе се у просторијама библиотеке 1. Све корисничке прикључне кутије опремљене су прикључницама које задовољавају категорију 5е. На корисничким прикључним кутијама завршавају се кориснички комуникациони путеви који полазе од КЧП-5-1. Ови комуникациони путеви изведени су бакарним парич-

ним кабловима категорије 5е који су положени у назидне пластичне каналнице. Почетак корисничких комуникационих путева је на *Patch* панелу у КЧП 5-1. Ознака *Patch* панела је КЧП-5-1-ПП-2. *Patch* панел има 20 прикључница које задовољавају категорију 5Е и две SC мултимодне оптичке прикључнице. Све корисничке прикључне кутије су, преко одговарајућих прикључница на *Patch* панелу и преспојних каблова, повезане на портове свича који се налази у КЧП. Ознака свича је КЧП-5-1-SW-1. Свич има 24 RJ45 порта који подржавају брзине рада од 10Mb/s или 100 Mb/s и један SC оптички мултимодни порт који подржава брзину од 1000 Mb/s.

Чвориште КЧП-5-1 је даље повезано са чвориштем КЧМ. Ова веза остварује се са SC оптичког порта свича КЧП-5-1-SW-1 преко оптичког мултимодног преспојног кабла до SC порта *Patch* панела КЧП-5-1-ПП-2 и даље секундарног комуникационог пута. Секундарни комуникациони пут завршава се на *Patch* панелу КЧП-5-1-ПП-2, а почиње на *Patch* панелу КЧМ-ПП-3 (12 SC оптичких мултимодних прикључница). Овај комуникациони пут изведен је оптичким каблом са четири мултимодна влакна 50/125μm и положен је у пластичне каналнице. Са прикључнице КЧМ-ПП-3 преко преспојног кабла веза се спаја на централни свич КЧМ-SW-1. Свич КЧМ-SW-1 има 12 SC оптичких мултимодних портова који подржавају рад на брзини од 1000 Mb/s и два RJ45 порта који подржавају рад на брзини од 1000 Mb/s.

12.5.1.2. Библиотека 2

Библиотека 2 налази се у удаљеном објекту институције. Двоструке корисничке прикључне кутије намењене за повезивање радних станица библиотечког особља библиотеке 2 и сервера БИСИС-а библиотеке 2 налазе се у просторијама библиотеке 2. Све корисничке прикључне кутије опремљене су прикључницама које задовољавају категорију 5Е. На корисничким прикључним кутијама завршавају се кориснички комуникациони путеви који полазе од КЧП-О-2-1-1. Ови комуникациони путеви изведени су бакарним паричним кабловима категорије 5Е који су положени у назидне пластичне каналнице. Почетак корисничких комуникационих путева је на *Patch* панелу у КЧП-О-2-1-1. Ознака *Patch* панела је КЧП-О-2-1-1-ПП-1. *Patch* панел има 24 прикључнице које задовољавају категорију 5Е. Све корисничке прикључне кутије су, преко одговарајућих прикључница на *Patch* панелу и преспојних каблова, повезане на портове свича који се налази у КЧП. Ознака свича је КЧП-О-2-1-1-SW-1. Свич има 24 RJ45 порта који подржавају брзине рада од 10 Mb/s или 100 Mb/s и један RJ45 порт који подржава брзину од 1000 Mb/s.

Чвориште КЧП-О-2-1-1 је даље повезано са чвориштем КЧО-2. Ова веза остварује се са 1000 Mb/s RJ45 порта свича КЧП-О-2-1-1-SW-1 преко

преспојног кабла до порта *Patch* панела КЧП-О-2-1-1-ПП-1 и даље преко секундарног комуникационог пута. Секундарни комуникациони пут завршава се на *Patch* панелу КЧП-О-2-1-1-ПП-1, а почиње на *Patch* панелу КЧО-2-ПП-1 (24 прикључнице категорије 5Е). Овај комуникациони пут изведен је бакарним паричним каблом категорије 5Е и положен је у пластичне каналнице. Са прикључнице КЧО-2-ПП-1 преко преспојног кабла веза се спаја на свич КЧО-2-SW-1. Свич КЧО-2-SW-1 има 12 RJ45 портова који подржавају рад на брзини од 1000 Mb/s и два SC оптичка мултимодна порта који подржавају рад на брзини од 1000 Mb/s.

Чвориште КЧО-2 је даље повезано са чвориштем КЧМ. Ова веза остварује се са SC оптичког порта свича КЧО-2-SW-1 преко оптичког мултимодног преспојног кабла до SC порта *Patch* панела КЧО-2-ПП-2 (20 прикључница категорије 5Е и две SC мултимодне оптичке прикључнице) и даље преко примарног комуникационог пута. Примарни комуникациони пут завршава се на *Patch* панелу КЧО-2-ПП-2, а почиње на *Patch* панелу КЧМ-ПП-3. Овај комуникациони пут изведен је оптичким каблом са четири мултимодна влакна 50/125μm и положен је у подземну кабловску канализацију између објеката и пластичне каналнице унутар објекта. Са прикључнице КЧМ-ПП-3 преко преспојног кабла веза се спаја на централни свич КЧМ-SW-1. Свич КЧМ-SW-1 има 12 SC оптичких мултимодних портова који подржавају рад на брзини од 1000 Mb/s и два RJ45 порта који подржавају рад на брзини од 1000 Mb/s.

12.5.1.3. Централно чвориште

У централном чворишту се поред уређаја за подршку локалној мрежи налази и уређај са функцијом рутирања КЧМ-Р-1, преко кога се остварује веза са Интернетом.

12.5.2. Логичка архитектура

12.5.2.1. Јавни IP адресни простор

Библиотекама 1 и 2 на располагању је део јавног адресног простора институције у чијем саставу функционишу библиотеке. Библиотекама је на располагању опсег јавних IP адреса са следећим особинама:

- Идентификација мреже је 147.91.179.32;
- Мрежна маска је 255.255.255.248, односно дужина маске је 29;
- *Broadcast* адреса је 147.91.179.39;

Из горе наведеног извршена је расподела јавних IP адреса на следећи начин:

- Јавна IP адреса намењена серверу БИСИС-а библиотеке 1 је 147.91.179.33/29;
- Јавна IP адреса намењена серверу БИСИС-а библиотеке 2 је 147.91.179.34/29;
- Остале јавне IP адресе из овог опсега додељују се комуникационим уређајима задуженим за функционисање дела мреже који користи овај адресни опсег.

12.5.2.2. Приватни IP адресни простор

Библиотекама 1 и 2 на располагању је део приватног IP адресног простора који користи институција у оквиру које функционишу библиотеке. За библиотеке 1 и 2 предвиђена је следећа расподела приватног IP адресног простора:

- За библиотеку 1 предвиђен је адресни простор 192.168.171.0/24;
- За библиотеку 2 предвиђен је адресни простор 192.168.172.0/24;
- За јавно доступне читаонице предвиђен је адресни простор 192.168.111.0/24.
- За административно особље предвиђен је адресни простор 192.168.254.0/24.

12.5.2.3. Организација имена

Сва имена активних компоненти које се користе у библиотекама 1 и 2 су из следећих подскупова имена скупа *institucija.co.yu*:

- Сва имена активних компоненти библиотеке 1 припадају подскупу имена biblioteka1.institucija.co.yu (www.biblioteka1.institucija.co.yu, katalog.biblioteka1.institucija.co.yu).
- Сва имена активних компоненти библиотеке 2 припадају подскупу имена biblioteka2.institucija.co.yu (www.biblioteka2.institucija.co.yu, katalog.biblioteka2.institucija.co.yu).

12.5.2.4. Виртуелне локалне мреже

Да би се могла применити изнета логичка архитектура на локалној мрежи дефинишу се следеће виртуелне локалне мреже (ВЛРМ):

- *ВЛРМ библиотечког особља библиотеке 1*. У овој ВЛРМ треба да се налазе радне станице библиотекара библиотеке 1 и интерфејс БИСИС сервера библиотеке 1 коме приступају библиотекари библиотеке 1. Идентификација овог ВЛРМ је Б1ВЛРМ и он се везује са IP адресним простором 192.168.171.0/24.
- *ВЛРМ библиотечког особља библиотеке 2*. У овој ВЛРМ треба да се налазе радне станице библиотекара библиотеке 2 и интерфејс БИСИС сервера библиотеке 2 коме приступају библиотекари библиотеке 2. Идентификација овог ВЛРМ је Б2ВЛРМ и он се везује са адресним простором 192.168.172.0/24.
- *ВЛРМ читаонице*. У овој ВЛРМ треба да се налазе радне станице у јавно доступним читаоницама библиотеке. Идентификација овог ВЛРМ је ЦИТВЛРМ и он се везује са адресним простором 192.168.111.0/24.
- *ВЛРМ јавног приступа*. У овој ВЛРМ треба да се налазе интерфејси БИСИС сервера библиотеке 1 и 2 намењени за приступ корисницима Интернета. Идентификација овог ВЛРМ је ЈАВВЛРМ и он се везује са адресним простором 147.91.179.32/29.
- *ВЛРМ административног особља*. У овој ВЛРМ треба да се налазе радне станице запослених у административним службама. Идентификација овог ВЛРМ је АДМВЛРМ и он се везује са адресним простором 192.168.254.0/24.

12.5.2.5. Управљање током саобраћаја

На свичевима у мрежи дефинишу се следећи ВЛРМ:

- На свичу КЧП-5-1-SW-1: Б1ВЛРМ, ЈАВВЛРМ и АДМВЛРМ.
- На свичу КЧП-О-2-1-1-SW-1: Б2ВЛРМ, ЈАВВЛРМ и АДМВЛРМ.
- На свичу КЧО-2-SW-1: Б2ВЛРМ, ЈАВВЛРМ и АДМВЛРМ.
- На свичу КЧМ-SW-1: Б1ВЛРМ, Б2ВЛРМ, ЈАВВЛРМ, ЦИТВЛРМ и АДМВЛРМ.

На свичевима у мрежи дефинисана је припадност појединих портова одређеним ВЛРМ. Изузетак су портови који спајају свичеве и портови на којима су повезани БИСИС сервери библиотека. Ови портови конфигурисани су тако да преко протокола IEEE 802.1q преносе информације о ВЛРМ. Протокол IEEE 802.1q подржан је на свим свичевима у мрежи као и серверима БИСИС-а.

На свичу КЧМ-SW-1 дефинишу се правила усмеравања саобраћаја између појединих приватних IP мрежа тако да се са сваке приватне IP мреже

може доспети до било које IP мреже од значаја за библиотеке 1 и 2. На рутеру КЧМ-Р-1 и свичу КЧМ-SW-1 дефинишу се правила усмеравања IP саобраћаја тако да се са Интернета може остварити двосмерна комуникација са интерфејсима БИСИС сервера који се налазе у јавно доступном IP адресном простору. Било какво директно усмеравање саобраћаја са приватних IP мрежа на јавну IP мрежу и обратно је правилима усмеравања саобраћаја забрањено.

12.5.2.6. Конфигурација мрежних сервиса сервера БИСИС

Сервери БИСИС библиотека 1 и 2 имају подршку за IEEE 802.1q. Подигнути мрежни интерфејси и њихове карактеристике дате су у табели 12.2. Мрежни сервиси подигнути на серверима БИСИС библиотека 1 и 2 дати су у табели 12.3.

Сервер БИСИС-а	Ознака мрежног интерфејса	ВЛРМ коме припада IP адреса мрежног интерфејса
библиотека 1	eth0.1	ЈАВВЛРМ
	eth0.2	Б1ВЛРМ
библиотека 2	eth0.1	ЈАВВЛРМ
	eth0.2	Б2ВЛРМ

Табела 12.2. Опис мрежних интерфејса сервера система БИСИС

Сервер БИСИС-а	**Опис сервиса**	**Протокол**	**Порт**	**Ознака ВЛРМ којој припада IP адреса за приступ сервису**
библиотека 1	Web интерфејс	TCP	80	ЈАВВЛРМ
	Web интерфејс	TCP	80	Б1ВЛРМ
	РСУБП	TCP	7200	Б1ВЛРМ
	RMI	TCP	1099	ЈАВВЛРМ
	RMI	TCP	1099	Б1ВЛРМ
	SSH	TCP	22	Б1ВЛРМ
библиотека 2	Web интерфејс	TCP	80	ЈАВВЛРМ
	Web интерфејс	TCP	80	Б2ВЛРМ
	РСУБП	TCP	7200	Б2ВЛРМ
	RMI	TCP	1099	Б2ВЛРМ
	SSH	TCP	22	Б2ВЛРМ

Табела 12.3. Опис мрежних сервиса сервера система БИСИС

12.5.2.7. Контрола приступа

Контрола приступа мрежним сервисима и појединим модулима система БИСИС врши се на више нивоа:

- Поделом на ВЛРМ извршена је подела *Broadcast* домена и на тај начин онемогућена комуникација између логичких IP целина ван дефинисаних правила усмеравања саобраћаја.
- Применом пакетских филтера на уређајима КЧМ-Р-1, КЧМ-SW-1 и на серверима БИСИС-а.
  - КЧМ-Р-1 физички интерфејс са Интернета у смеру ка мрежи институције: Према јавним адресама сервера БИСИС-а библиотека 1 и 2 допушта се саобраћај само на TCP порт 80 са свих јавних IP адреса (сем са адреса резервисаних за библиотеке 1 и 2) и са свих TCP портова. Према јавној адреси сервера БИСИС-а библиотеке 1 допушта се саобраћај само на TCP порт 1099 са јавних IP адреса које припадају класификованим библиотекама и са свих TCP портова. Сав остали саобраћај ка јавним IP адресама сервера БИСИС-а библиотека 1 и 2 се забрањује.
  - КЧМ-SW-1 виртуелни интерфејс за ЈАВВЛРМ у смеру ка мрежи: Према јавним адресама сервера БИСИС-а библиотека 1 и 2 допушта се саобраћај само на TCP порт 80 са свих јавних IP адреса (сем са адреса резервисаних за библиотеке 1 и 2) и са свих TCP портова. Према јавној адреси сервера БИСИС-а библиотеке 1 допушта се саобраћај само на TCP порт 1099 са јавних IP адреса које припадају класификованим библиотекама и са свих TCP портова. Поред ових правила могу се поставити и додатна правила у складу са политиком администрације целокупне мреже институције а која неће кршити раније наведена правила. Сав остали саобраћај ка јавним IP адресама сервера БИСИС-а библиотека 1 и 2 се забрањује.
  - КЧМ-SW-1 виртуелни интерфејс за Б1ВЛРМ у смеру ка мрежи: Према приватној адреси сервера БИСИС библиотеке 1 допушта се саобраћај на TCP портове 22, 1099, 7200 и 80 са приватних IP адреса које су везане за АДМВРЛМ. Према приватној адреси сервера БИСИС библиотеке 1 допушта се саобраћај на TCP портове 80 и 1099 са приватних IP адреса које су везане за Б2ВРЛМ. Према приватној адреси сервера БИСИС библиотеке 1 допушта се саобраћај на TCP порт 80 са приватних IP адреса које су везане за ЦИТВРЛМ и осталих приватних IP адреса институције са којих се по правилима усмеравања саобраћаја може доспети до сервера БИСИС-а библиотеке 1. Поред ових правила могу се поставити и додатна правила у складу са политиком администрације целокупне мреже институције а која неће кршити раније наведена прави-

ла. Сав остали саобраћај ка приватниј IP адреси сервера БИСИС библиотеке 1 се забрањује.

- КЧМ-SW-1 виртуелни интерфејс за Б2ВЛРМ у смеру ка мрежи: Према приватној адреси сервера БИСИС библиотеке 2 допушта се саобраћај на TCP портове 22, 1099, 7200 и 80 са приватних IP адреса које су везане за АДМВРЛМ. Према приватној адреси сервера БИСИС библиотеке 2 допушта се саобраћај на TCP портове 80 и 1099 са приватних IP адреса које су везане за Б1ВРЛМ. Према приватној адреси сервера БИСИС библиотеке 2 допушта се саобраћај на TCP порт 80 са приватних IP адреса које су везане за ЦИТВРЛМ и осталих приватних IP адреса институције са којих се по правилима усмеравања саобраћаја може доспети до сервера БИСИС-а библиотеке 2. Поред ових правила могу се поставити и додатна правила у складу са политиком администрације целокупне мреже институције а која неће кршити раније наведена правила. Сав остали саобраћај ка приватниј IP адреси сервера БИСИС библиотеке 2 се забрањује.
- БИСИС сервер библиотеке 1 виртуелни интерфејс са Б1ВЛРМ: Допушта се саобраћај ка портовима 80 и 7200 са класификованих радних станица које су у оквиру Б1ВЛРМ-а и ка порту 80 са осталих радних станица које су у Б1ВЛРМ. Сав остали саобраћај са радних станица које су у оквиру Б1ВЛРМ се забрањује.
- БИСИС сервер библиотеке 2 виртуелни интерфејс са Б2ВЛРМ: Допушта се саобраћај ка портовима 80 и 7200 са класификованих радних станица које су у оквиру Б2ВЛРМ-а и ка порту 80 са осталих радних станица које су у Б2ВЛРМ. Сав остали саобраћај са радних станица које су у оквиру Б2ВЛРМ се забрањује.

- Применом аутентификације корисника на свим сервисима БИСИС сервера библиотеке 1 и 2 сем на сервису за корисничко претраживање који је доступан преко TCP порта 80.

12.6. Закључак

Описана физичка и логичка архитектура мреже пружа подршку имплементацији библиотечког информационог система. Изнети принципи омогућавају висок ниво доступности и перформанси система, док се организацијом права приступа појединим модулима систем обезбеђује од неауторизованог коришћења чиме се додатно обезбеђује стабилност у раду библиотечког информационог система. Увођење нових сервиса и категорија корисника не

захтева измене основних поставки већ само допуну постојећих правила, чиме се омогућава несметан даљи развој целокупног система.

Референце

[1] P.V. Mockapetris,*Domain names - concepts and facilities.* RFC 1034, November 1987.

[2] ANSI/TIA/EIA-568-A, *Commercial Building Telecommunications Cabling Standard.*

[3] R. Braden (Ed.). *Requirements for Internet Hosts – Communication Layers*. RFC 1122, October 1989.

[4] Y. Rekhter et al. *Address Allocation for Private Internets*. RFC 1918, February 1996.

Прилог 1

Приказ магистарских теза и докторских дисертација

Д. Сурла, З. Коњовић

У овом Прилогу дат је приказ магистарских теза и докторских дисертација у којима су се теме истраживања односиле на развој система БИСИС.

Гордана Будимир
Контрола квалитета XML библиографских записа

Магистарска теза припада области информационих система, и то оном делу који се односи на развој информационих система дигиталних библиотека базираних на XML технологији. Написана је на српском језику (ћирилица) а извод тезе је српски/енглески. Садржи 7 поглавља, 127 страна Б5 формата, са 99 литерарних цитата из предметне области, 9 табела и 8 слика. На почетку текста дата је насловна страна, предговор и садржај на 8 страна а на крају текста дата је биографија и кључна документацијска информација на укупно 9 страна.

Циљ тезе је да се изврши контрола квалитета библиографских записа базирана на моделирању XML шема језиком и програмирањем у XSLT језику. Као верификација предложене контроле, имплементиран је систем за контролу квалитета XML библиографских записа у Java окружењу. Реализација ових задатака описана је у следећим поглављима:

1. Увод,
2. Контрола квалитета библиографских записа,
3. Моделирање контрола библиографских записа помоћу XML Schema језика,
4. Моделирање контрола библиографских записа помоћу XSLT језика,

5. Спецификација система за контролу квалитета XML библиографских записа,
6. Имплементација система за контролу квалитета XML библиографских записа и
7. Закључак.

У *првом* поглављу описан је UNIMARC библиографски формат и структура библиографских записа по UNIMARC формату. Дат је детаљан преглед развоја формата за пренос библиографских записа и описани су пројекти који повезују MARC и UNIMARC формат са XML форматом. Објашњена је повезаност XML-а са библиографским подацима и различити начини употребе XML-а у развоју библиотечких информационих система. На крају поглавља дат је преглед система за контролу квалитета библиографских записа и његове имплементације, као једног од сегмената библиотечког информационог система.

Дефиниција квалитета библиографских записана дата је у *другом* поглављу. На основу доступне литературе дат је исцрпан преглед концепата за контролу квалитета тих записа. Ове контроле су систематизоване у две групе: као контроле структуре записа и као контроле садржаја записа. Поред тога, контроле могу бити појединачне (контрола структуре или садржаја појединачног поља, потпоља или индикатора) и унакрсне (контрола структуре или садржаја међусобно повезаних више поља, потпоља или индикатора). На примеру YUMARC библиографских записа приказана је примена наведених контрола. На сличан начин могуће је описати контроле записа и за друге MARC формате.

Централни део тезе дат је у трећем и четвртом поглављу. Помоћу XML шеме дефинисана је структура XML библиографских записа чијом валидацијом се може проверити исправност библиографских записа. У *трећем* поглављу показано је да се на овај начин могу моделирати скоро сви концепти појединачних контрола као и неки концепти унакрсних контрола. Моделирање преосталих концепата контрола XML библиографских записа, који се не могу моделирати XML шемом, описано је у *четвртом* поглављу. У овом поглављу показано је да се све преостале контроле могу описати применом XSLT и XPath језика. Предложене контроле библиографских записа описане у трећем и четвртом поглављу примењене су на YUMARC библиографске записе али се могу применити и на друге MARC формате.

У *петом* поглављу описано је објектно-оријентисано моделирање система за комплетну контролу XML библиографских записа чији концепти су моделирани XML шемом, XSLT и XPath језицима и описани у трећем и

четвртом поглављу. Резултат провере квалитета записа састоји се у валидацији моделираних концепата контрола као извештај о грешкама у XML библиографском запису. Моделирање је извршено у CASE алату *Rational Rose*.

Имплементација система за контролу квалитета XML библиографских записа у Java окружењу описана је у *шестом* поглављу. У овом поглављу дат је и кратак опис коришћених готових софтверских пакета, као што су: за валидацију XML докумената – MSV; за XSLT трансформације XML докумената – *Xalan*; за парсирање XML докумената – *Xerces*. Такође су описане и класе развијене у оквиру ове апликације. На крају овог поглавља дат је опис коришћења апликације и примери контроле XML библиографских записа. Систем је имплементиран независно од библиографског формата, што значи да би се њиме могла извршити контрола квалитета XML библиографских записа било ког библиографског формата за које су дефинисане валидацијска XML шема и XSLT спецификација за контролу тих XML библиографских записа. За изабран библиографски формат, апликација омогућава проналажење свих грешака у улазним XML библиографским записима одједном, што је значајна предност у односу на апликације, нпр. XML Spy едитор, којима се проналази само прва грешка у запису, тако да се за сваку грешку мора поновно извршити контрола библиографског записа.

У *седмом* поглављу дат је закључак о значају и доприносу тезе, као и предлог за даља истраживања.

Главни научни допринос магистарске тезе је следећи:

- дефинисање и систематизација контрола за утврђивање квалитета библиографских записа,
- моделирање концепата контрола записа помоћу XML шема језика,
- XSLT спецификација за опис контрола које се не могу описати XML шема језиком и
- имплементација система за контролу квалитета XML библиографских записа у Java окружењу.

Добијени резултати у оквиру ове магистарске тезе омогућују даљи развој библиотечких софтверских система у следећим сегментима:

- имплементација XML едитора за библиографске записе који не би захтевао познавање XML-а од стране корисника,
- развој веб сервиса за контролу квалитета XML библиографских записа,
- дефиниција и имплементација интерпретера за формални опис контрола квалитета библиографских записа или

– развој експертног система за моделирање система за контролу квалитета XML библиографских записа на основу XML шеме одговарајућег библиографског формата, уз употребу интерпретера за формални опис контрола.

Магистарска теза одбрањена је 14.05.2004. године на Природно--математичком факултету, Департман за математику и информатику, Нови Сад (ментор проф. др Душан Сурла). Теза је доступна у библиотеци Департмана за математику и информатику у Новом саду и на сајту:

http://diglib.ns.ac.yu/ndltd/docs/set3/ndltd409/kontrola_kvaliteta.pdf

Милан Видаковић
Прошириво агентско окружење базирано на Java технологији

Докторска дисертација написана је на српском језику (латиница), садржи 7 поглавља, 182 стране Б5 формата, са 118 литерарних цитата из предметне области, 56 слика, и 13 табела. Испред основног текста, у ауторском репринту дисертације, дате су насловна страна, садржај и предговор на укупно 8 посебно нумерисаних страна а на крају текста дата је биографија и на српском и енглеском језику кључна документацијска информација на 7 ненумерисаних страна.

Предложена тема докторске дисертације припада области агентских система и технологија. Истраживањима у овој области посвећен је велики број радова који се могу класификовати на радове који се баве наменским агентским системима и радове који се баве агентским системима опште намене. Имплементационе технологије такође се могу класификовати у две групе: наменски развијане агентске технологије и технологије дистрибуираних система. Имплементације у већини случајева представљају агентска окружења реализована као резултат потребе да се реши конкретан проблем – дакле, наменске моделе са наменски развијаним имплементационим технологијама. У последње време појавила су се решења која воде ка систематичном развоју области (дефинисање стандарда: FIPA specifikacija, CORBA, J2EE).

Циљ дисертације је спецификација и имплементација модела проширивог агентског окружења (*Extensible Java-based Agent Framework* – XJAF) који омогућава рад агената у дистрибуираном окружењу са FIPA спецификацијом као основом за стандардизацију и J2EE технологијом као репрезентом технологије дистрибуираних софтверских компоненти за имплементацију агената.

Коришћење FIPA спецификације обезбеђује сарадњу са другим агентским окружењима, као и проширивост путем идентификације појединачних сегмената у FIPA спецификацији и њиховим моделовањем *plug-in* компонентама. Овај квалитет огледа се у два аспеката. (1) Омогућена је имплементација различитих алгоритама и њихово уграђивање у систем простим конфигурисањем. (2) Екстерни модули који задовољавају критеријуме FIPA спецификације могу се путем *plug-in* компоненти интегрисати у систем.

Реализација постављеног циља описана је у следећом поглављима:

1. Уводна разматрања,
2. Анализа агентских окружења,
3. Могућност примене J2EE технологије за имплементацију агентског окружења,
4. Формална спецификација проширивог агентског окружења,
5. Имплементација проширивог агентског окружења у J2EE технологији,
6. Верификација предложеног решења у Библиотечком информационом систему БИСИС и
7. Закључак.

Прво поглавље садржи следећа уводна разматрања: приказ дефиниција основних појмова и проблема у области агентских система, описе репрезентативних имплементационих технологија (CORBA и J2EE) и детаљан опис и анализу FIPA спецификације.

Друго поглавље се бави детаљно агентским окружењима. Дат је приказ радова који се баве агентским окружењима а затим детаљан приказ и анализа шест одабраних репрезентативних агентских окружења. Свако окружење је анализирано са становишта најбитнијих фактора агентског окружења: размена порука, мобилност агената, сигурност, сервиси и FIPA компатибилност.

Треће поглавље разматра могућности примене J2EE технологије за имплементацију агентског окружења. У њему је дат приказ и анализа техника из J2EE технологије које се могу употребити у агентској технологији. Концепти који се желе остварити су асинхрона размена порука, мобилност агената, директоријуми, сервиси и сигурносни механизми, а технологије потребне за реализацију су JMS (*Java Messaging Service*) систем порука, серијализација, JNDI технологија и J2EE сигурносни механизми. Анализирани су одабрани J2EE апликативни сервери (*Orion* и *JBoss*).

Садржаји четвртог и петог поглавља представљају централни део дисертације.

Четврто поглавље обухвата формалну спецификацију проширивог агентског окружења, коришћењем обједињеног језика моделирања. Сви потребни концепти за реализацију агентског окружења су моделирани у складу са FIPA спецификацијом. Идентификовани су елементи FIPA спецификације који се моделују као *plug-in* компоненте. Најбитније су: директоријум агената, директоријум сервиса, подсистем за размену порука и безбедносни подсистем.

У *петом* поглављу дат је опис имплементације предложеног модела на одабраном дистрибуираном систему. Имплементација је извршена коришћењем J2EE технологије. Агенти и сервиси су моделирани као EJB компоненте, комуникација између агената се остварује употребом JMS подсистема, директоријуми су имплементирани употребом JNDI подсистема, а мобилност агената и безбедносни механизми су моделовани постојећим решењима унутар J2EE технологије (серијализација и *Java Cryptography Extension*).

Шесто поглавље је посвећено верификацији предложеног агентског окружења у библиотечком информационом систему БИСИС. На почетку је представљена функционална структура БИСИС-а и анализа функција са становишта погодности и оправданости имплементације применом агентске технологије. На основу ове анализе, одабран је и имплементиран скуп функција БИСИС-а (узајамна каталогизација, контрола квалитета библиографских записа и балансирање расподеле оптерећења у процесу претраге). Функције су имплементиране коришћењем предложеног модела агентског окружења. Имплементацијом су обухваћени сви концепти предложеног модела а сама имплементација је извршена у програмском језику Java.

Седмо поглавље садржи закључна разматрања, анализу резултата и доприноса докторске дисертације, као и анализу праваца даљих истраживања.

Главни научни допринос докторске дисертације је формална спецификација и имплементација проширивог агентског окружења коју карактеришу следеће особине:

- модел проширивог агентског окружења је базиран на технологији дистрибуираних компоненти. Овај модел подржава FIPA спецификацију и следеће концепте агентског окружења: размену порука, мобилност агената, сигурносне механизме и директоријуме агената и сервиса;
- систем подржава концепт *plug-in*-ова за све битне компоненте агентског окружења. Овај концепт омогућује прилагођавање конкретне имплементације потребама корисника са становишта алгоритама за имплементацију функционалности агентског окружења;

- систем подржава концепт мобилних задатака који омогућује писање специфичних задатака који садрже у себи и програм за решавање, чиме се пројектанти ослобађају обавезе креирања специјализованих агената, а сви агентски центри обавезе да поседују све врсте агената и
- систем подржава концепт међусобног уређења односа агентских центара који омогућује једноставну експлоатацију агената из агентских центара зато што програмер не мора да познаје топологију мреже, нити стварне локације агентских центара.

Предложени модел система верификован је на на примеру Библиотечког Информационог Система БИСИС за који је реализована прототипска имплементација. Ова имплементација обухвата основне компоненте агентског окружења, чиме је потврђена практична вредност предложеног модела.

Докторска дисертација одбрањена је 06.12.2003. године на Факултету техничких наука, Катедра за рачунарство и аутоматику, Нови Сад (ментор проф. др Зора Коњовић). Дисертација је доступна у библиотеци Факултета техничких наука у Новом Саду и на сајту:

http://diglib.ns.ac.yu/ndltd/docs/set2/ndltd344/MinjaDoktorat.pdf

Јована Видаковић
Моделирање и имплементација библиотечких каталошких листића у XML технологији

Магистарска теза припада научној области Информатика, ужа научна област Информациони системи. Написана је на српском језику (ћирилица), а извод тезе је српски/енглески. Садржи 5 поглавља, 184 стране Б5 формата, са 32 литерарна цитата из предметне области, 71 сликом и 2 прилога. На почетку текста дата је насловна страна, предговор и садржај на 6 страна а на крају текста дата је биографија и кључна документацијска информација на укупно 6 страна.

У тези је постављен циљ да се изврши моделирање библиотечких каталошких листића у облику XML докумената на основу XML библиографских записа. Концепти разних типова каталошких листића описани су XML шема језиком, као и садржај конкретних каталошких листића. Резултат издвајања садржаја каталошког листића из записа је такође дат у облику XML документа. Визуализација овог документа је извршена у програмском језику Java, а резултат визуализације је HTML документ.

Магистарска теза обухвата следећа поглавља:

1. Увод,
2. Спецификација структуре каталошког листића помоћу XML шема језика,
3. Имплементација система за генерисање и приказивање каталошких листића,
4. Примери формирања каталошких листића,
5. Закључак,
6. Прилог 1: XML шема документ *Koncepti*
7. Прилог 2: XML шема документ *Dokument*

У *првом* поглављу је дат кратак опис стандарда у библиотечким информационим системима, као и опис извештаја у библиотечком информационим систему. Дат је преглед неких од постојећих система за библиотечко извештавање. Укратко је описана XML технологија и начин њене примене у тези.

У *другом* поглављу су детаљно описани уочени концепти библиотечких каталошких листића и начин на који су они моделирани. Дат је XML шема документ који описује концепте који се јављају на различитим типовима каталошких листића. Затим су приказани XML шема документи који описују конкретне каталошке листиће.

Спецификација информационих захтева је извршена коришћењем UML нотације. За опис статичке структуре коришћени су дијаграми класа, а за опис динамичке структуре дијаграми секвенци. Ова спецификација и имплементација у програмском језику Java дати су у *трећем* поглављу.

У *четвртом* поглављу су дати примери формирања неколико типова каталошких листића. Приказани су каталошки листићи добијени применом описаног система.

Закључак о значају и доприносу тезе, као и предлог за даља истраживања дат је у *петом* поглављу.

У *Прилогу 1* дат је комплетан листинг XML шема документа за опис уочених концепата каталошких листића. У *Прилогу 2* дат је опис XML шема документа на основу ког се формира XML документ који је резултат издвајања садржаја из XML библиографског записа.

Основни научни доприноси магистарске тезе су:

- уопштење описа библиотечких каталошких листића њиховом спецификацијом у XML технологији,
- формирање XML шема документа за опис концепата каталошких листића,

- формирање XML шема документа за опис конкретних каталошких листића,
- моделирање и имплементација система за генерисање садржаја и изгледа каталошких листића у Java окружењу.

Добијени резултати су основа за укључивање описаног система за генерисање библиотечких каталошких листића у оквиру библиотечких софтверских система.

Магистарска теза одбрањена је 20.10.2003. године на Природно-математичком факултету, Департман за математику и информатику, Нови Сад (ментор проф. др Милош Рацковић). Теза је доступна у библиотеци Департмана за математику и информатику у Новом саду и на сајту:

http://diglib.ns.ac.yu/ndltd/docs/set2/ndltd335/JovanaMagistarski.pdf

Бранко Милосављевић
Прошириви систем за проналажење мултимедијалних докумената

Докторска дисертација написана је на српском језику (латиница), садржи 4 поглавља, 189 страна Б5 формата, са 328 литерарних цитата из предметне области, 36 слика, и 12 табела. Испред основног текста, у ауторском репринту дисертације, дате су насловна страна, садржај и предговор на укупно 10 посебно нумерисаних страна а на крају текста дата је биографија и на српском и енглеском језику кључна документацијска информација на 7 ненумерисаних страна.

Предложена тема докторске дисертације припада области проналажења структурираних мултимедијалних докумената. Овом проблему посвећен је релативно мали број истраживања, имајући у виду укупан број објављених резултата који се односе на проналажење мултимедијалних објеката схваћених у ширем смислу. Постојећи резултати у овој области дефинишу моделе структурираних мултимедијалних докумената и моделе њиховог проналажења који, по правилу, намећу свој модел садржаја и/или свој модел проналажења за све типове медија.

Циљ дисертације је развој модела проширивог система за проналажење структурираних мултимедијалних докумената (*extensible multimedia information retrieval system*, XMIRS) који омогућава интеграцију различитих постојећих решења за проналажење мултимедијалних објеката и употребу различитих модела проналажења докумената. На овај начин, систем за прона-

лажење докумената не намеће свој модел проналажења нити свој модел репрезентације садржаја њихових мултимедијалних елемената.

Модуларност XMIRS система представља могућност употребе постојећих софтверских решења за проналажење мултимедијалних објеката. XMIRS систем састоји се од језгра и модула намењених проналажењу објеката. Способност XMIRS-а да омогући комбиновање функционалности различитих модула у оквиру изабраног модела проналажења докумената доноси нови квалитет процесу проналажења. Овај квалитет се огледа у два аспекта: (1) могућност формирања критеријума за проналажење докумената који се не могу изразити функцијама појединих модула посматраних независно, и (2) могућност прилагођавања система потребама везаним за конкретну примену, кроз додавање одговарајућих модула за проналажење објеката и додавање модела проналажења докумената.

Реализација постављеног циља описана је у следећом поглављима:

1. Уводна разматрања,
2. Модел проширивог система за проналажење мултимедијалних докумената,
3. Верификација прототипа система на дигиталној библиотеци и
4. Закључак.

Прво поглавље садржи уводна разматрања, дат је приказ дефиниција основних појмова и проблема у области проналажења информација. Потом је дат детаљан приказ објављених резултата у овој области. Резултати су разврстани у четири главне категорије: (1) истраживања која се баве класичним текстуалним системима за проналажење информација, (2) истраживања која се баве проналажењем слика по садржају, (3) она која се баве анализом видео записа и (4) истраживања која се баве проналажењем мултимедијалних објеката. Третман других типова медија је у литератури знатно мање заступљен, тако да се ове четири групе издвајају као основни правци истраживања. У овом поглављу дат је исцрпан преглед резултата истраживања за све четири поменуте групе.

Централни део дисертације дат у је *другом* поглављу. Ово поглавље садржи формалну спецификацију XMIRS система за проналажење структурираних мултимедијалних докумената. Формалном спецификацијом обухваћени су:

- дефиниције одговарајућих појмова везаних за XMIRS,
- модел докумената XMIRS-а,
- модели проналажења докумената који су прилагођени за XMIRS и

– софтверска архитектура XMIRS-а која омогућава проширивост система различитим модулима за проналажење мултимедијалних објеката и проширивост система различитим моделима проналажења докумената.

Предложени систем превазилази следеће проблеме постојећих модела, односно система: (1) руковање документима који имају унапред одређену, фиксну структуру, (2) рад са ограниченим бројем типова медија, и (3) наметање сопственог модела проналажења за све подржане типове медија.

XML језик је употребљен за спецификацију XMIRS-а. Прво, документи којима рукује систем репрезентовани су XML језиком што, имајући у виду данашњу технолошку подршку за XML, обезбеђује једноставност размене докумената са другим системима. Друго, упити упућени XMIRS-у и резултати проналажења који се враћају клијенту такође су у форми XML докумената. На тај начин омогућен је једноставан приступ систему у Интернет окружењу. На крају другог поглавља укратко је наведено имплементационо окружење прототипа XMIRS система.

У *трећем* поглављу дисертације представљена је верификација приказаног модела за конкретан систем – мрежу дигиталних библиотека докторских и магистарских радова (NDLTD). Верификација се спроводи помоћу развијеног прототипа XMIRS-а конфигурисаног за конкретну примену, што обухвата следеће активности:

– моделирање докумената дигиталне библиотеке,
– дефинисање корисничких потреба везаних за могућности процеса проналажења докумената,
– приказ употребљених постојећих софтверских модула који обезбеђују поједине елементе функционалности претраживања,
– приказ конфигурације система која повезује језгро система и употребљене модуле у функционалну целину и
– илустрацију примера коришћења овако формираног система.

Приказана структура докумената дигиталне библиотеке обухвата четири типа медија (структурирани текст, неструктурирани текст, битмапиране слике и видео записе). Демонстрација комбиноване употребе разнородних модула за претраживање представља верификацију постављене функционалности XMIRS-а: проналажења докумената по критеријумима који обухватају ограничења постављена над различитим типовима медија.

Четврто поглавље садржи закључна разматрања, анализу резултата и доприноса докторске дисертације, као и анализу праваца даљих истраживања.

Главни научни допринос докторске дисертације је у дефинисању формалне спецификације система као и израда прототипа за проналажење структурираних мултимедијалних докумената кога карактеришу следеће особине:

- коришћење различитих модела проналажења докумената,
- употреба различитих постојећих софтверских решења за проналажење мултимедијалних објеката,
- конфигурабилност система која омогућава прилагођавање конкретне имплементације потребама корисника са аспеката руковања документима и њиховог проналажења и
- комбиновање функционалности појединих модула у целину која доноси веће могућности у проналажењу докумената него појединачни модули употребљени засебно.

Предложени модел система верификован је на репрезентативном примеру мреже дигиталних библиотека докторских и магистарских радова за који је реализована и прототипска имплементација. Прототипска имплементација која испуњава циљеве у погледу функционалности постављене пред XMIRS систем представља потврду практичне вредности предложеног модела.

Докторска дисертација одбрањена је 22.07.2003. године на Факултету техничких наука, Катедра за рачунарство и аутоматику, Нови Сад (ментор проф. др Зора Коњовић). Дисертација је доступна у библиотеци Факултета техничких наука у Новом Саду и на сајту:

http://diglib.ns.ac.yu/ndltd/docs/set1/ndltd143/xmirs.pdf

Владимир Мијић
XML едитор за опис UNIMARC формата

Магистарска теза припада научној области Информатика, ужа научна област Информациони системи. Написана је на српском језику (ћирилица) а извод тезе је српски/енглески. Садржи 6 поглавља, 170 страна Б5 формата, са 21 литерарним цитатом из предметне области, 64 слике и 4 прилога. На почетку текста дата је насловна страна, предговор и садржај на 6 страна а на крају текста дата је биографија и кључна документацијска информација на укупно 6 страна.

Циљ тезе је да се моделира и имплементира XML едитор за UNIMARC формат. За остварење овог циља било је потребно претходно урадити моделирање UNIMARC формата у XML Schema језику, као и имплементаци-

ју објектног модела за XML Schema шему, заједно са библиотеком класа помоћу које би се вршила интерпретација овог модела. Реализација ових задатака описана је у следећим поглављима:

1. Увод,
2. XML шема за опис UNIMARC формата,
3. Интерпретација XML шеме,
4. Моделирање XML едитора за опис UNIMARC формата,
5. Имплементација XML едитора за опис UNIMARC формата и
6. Закључак.

У *првом* поглављу је дат кратак осврт на примену XML технологије и њену интеграцију у библиотечке информационе системе. Затим су наведене основне карактеристике XML едитора опште намене, приказани су и репрезентативни едитори за обраду библиографске грађе. На крају овог поглавља дата је анализа постојећих едитора и описан предлог спецификације специјализованог XML едитора за подршку библиотечким стандардима.

Моделирање UNIMARC формата помоћу XML Schema језика описано је у *другом* поглављу. Уочени су сви концепти UNIMARC формата и дат предлог за њихово моделирање. Комлетан листинг XML шеме за UNIMARC формат доступан је у електронској форми.

Затим су у *трећем* поглављу проучене могућности тренутно доступног софтверског пакета *Multi Schema Validator* за репрезентацију XML Schema шеме објектним моделом у програмском језику Java. Постојећа верзија овог софтверског пакета нуди објектни модел шеме који није прилагођен за интерпретирање из апликације. Зато је у овом у поглављу описано моделирање и имплементација специјализованог објектног модела шеме као и библиотека класа помоћу којих се тај модел интерпретира и користи.

У **четвртом** поглављу дата је спецификација и описано моделирање XML едитора за опис UNIMARC формата у UML нотацији. Основна идеја је да едитор у току рада консултује XML шему која представља имплементирани модел стандарда за опис библиографских записа UNIMARC формата.

У **петом** поглављу описана је имплементација едитора у програмском језику Java. Резултат употребе (едитовање, валидација и меморисање) едитора је XML документ који садржи све потребне описе структурних концепата стандарда. У имплементацију едитора је укључен посебно реализован метод интернационализације заснован такође на XML ресурсним датотекама.

У *шестом* поглављу је дат закључак о значају и доприносу тезе, као и предлог за даља истраживања и интеграцију резултата тезе у информациони систем дигиталне библиотеке.

Главни научни допринос магистарске тезе је следећи:

1. У складу са спецификацијом едитора развијен је објектни модел за шема документе, који је њихова меморијска репрезентација. Проучене су карактеристике и могућности постојећих сродних решења доступних преко Интернета. У оквиру тезе је имплементиран специјализован објектни модел шеме као и бибилиотека класа помоћу којих се тај модел интерпретира и користи. Добијени објектни модел за XML шему представља главни научни допринос тезе.
2. Моделирање софтверског система самог едитора реализовано је помоћу UML-а, а имплементација система урађена је у Java окружењу. Детаљано је описана графичка компонента навигационог стабла која се намеће као универзално решење за приказ хијерархијских структура података, какви су и XML документи. Описани су алгоритми интерпретације и валидације XML докумената као илустративни примери рада са предложеним објектним моделом шеме. Поред тога, дат је предлог имплементације вишејезичког корисничког интерфејса (локализације) у Java окружењу, реализован у потпуности преко XML технологије.

Едитор имплементиран у овој тези представља својеврсни образац по чијем узору може да се имплементира и едитор за записе, а објектни модел шеме и библиотека класа које га интерпретирају могу да се искористе и за друге апликације које треба да су у свом раду "свесне" објектног модела шеме.

Магистарска теза одбрањена је 08.07.2003. године на Природно--математичком факултету, Департман за математику и информатику, Нови Сад (ментор проф. др Душан Сурла). Теза је доступна у библиотеци Департмана за математику и информатику у Новом Саду и на сајту:

http://diglib.ns.ac.yu/ndltd/docs/set1/ndltd133/MijicVMagistarskaTeza.pdf

Мирјана Зеремски
Моделирање UNIMARC формата у XML технологији

Магистарска теза припада научној области Информатика, ужа научна област Информациони системи. Написана је на српском језику (ћирилица), а извод тезе је српски/енглески. Садржи 6 поглавља, 156 страна Б5 формата, са

38 литерарних цитата из предметне области, 25 слика и 2 прилога. На почетку текста дата је насловна страна, предговор и садржај на 6 страна а на крају текста дата је биографија и кључна документацијска информација на укупно 6 страна.

У тези је постављен циљ да се изврши моделирање UNIMARC формата у XML технологији, и то како самог формата, тако и библиографских записа по UNIMARC формату, затим да се имплементира конверзија библиографских записа по UNIMARC формату у XML библиографске записе, као и валидација добијених XML записа. Наведено моделирање и имплементација реализовано је у оквиру развоја библиотечког софтверског система БИСИС. Овај систем базиран је на подскупу UNIMARC формата познатом под називом YUMARC.

Магистарска теза садржи следећа поглавља:

1. Увод,
2. Моделирање UNIMARC формата помоћу XML Schema језика,
3. Моделирање библиографских записа помоћу XML Schema језика,
4. Конверзија библиографских записа по UNIMARC формату у XML документе,
5. Валидација XML библиографских записа и
6. Закључак.

У *првом* поглављу је на основу доступне литературе дат преглед развоја дигиталних библиотека. Даље је дат кратак преглед UNIMARC формата и XML технологија. Такође је дат преглед пројеката који се баве развојем библиотечких информационих система, чија је документација била доступна путем Интернета, и који укључују коришћење XML технологије, што је предмет истраживања и ове тезе. На крају овог поглавља истакнуто је место и значај истраживања у овој тези у оквиру прегледа наведених пројеката.

Доказано је да се помоћу XML Schema језика могу моделирати сви концепти UNIMARC формата (*друго* поглавље). Као појава ове шеме формиран је валидан XML YUMARC документ. У прилогу овог поглавља дат је комплетан листинг шеме YUMARC формата.

У *трећем* поглављу такође је доказано и да се сви концепти библиографског записа по UNIMARC формату могу моделирати помоћу XML Schema језика. Дат је пример XML библиографског записа који је валидан у односу на наведену шему. У прилогу овог поглавља дата ја комплетна шема XML библиографских записа по YUMARC формату.

Као припрема за пето поглавље, у *четвртом* поглављу наведена је физичка структура UNIMARC записа имплементирана у апликацији БИСИС. Затим је описана имплементација конверзије библиографских записа по UNIMARC формату у XML библиографске записе, која је реализована у Java окружењу.

У оквиру **петог** поглавља описана је имплементација валидације XML библиографских записа у Java окружењу. Наведени су илустративни примери валидације библиографских записа. Све информације о резултатима валидације налазе се у одређеном фајлу на основу којег се могу формирати разни статистички извештаји. На тај начин формиран је систем за контролу квалитета записа, која се остварује само на основу шеме (без додатног програмирања).

У **шестом** поглављу дат је закључак о значају и доприносу тезе, као и предлог за даља истраживања.

Главни научни допринос магистарске тезе је: моделирање комплетног UNIMARC формата помоћу XML шеме; моделирање библиографских записа по UNIMARC формату помоћу XML шеме; моделирање и имплементација валидације XML библиографских записа у Java окружењу.

Добијени резултати су основа за: развој специјализованог едитора UNIMARC формата; преузимање библиографских података са специјализованих сервера базираних на XML документима; контрола XML библиографских записа; стварање кооперативних библиотечких информационих система; развој сервера за узајамну каталогизацију базираног на XML технологији.

Магистарска теза одбрањена је 26.12.2002. године на Природно-математичком факултету, Департман за математику и информатику, Нови Сад (ментор проф. др Душан Сурла). Теза је доступна у библиотеци Департмана за математику и информатику у Новом саду и на сајту:

http://diglib.ns.ac.yu/ndltd/docs/set1/ndltd64/ZeremskiMMagistarskaTeza.pdf

Биљана Пуповац
Моделирање и имплементација команди библиотечког информационог система

Магистарска теза написана је на српском језику (ћирилица), а извод тезе је српски/енглески. Садржи 6 поглавља, 118 страна Б5 формата, са 32 литерарна цитата из предметне области, 59 слика и 3 прилога. На почетку текста дата је насловна страна, предговор и садржај на 6 посебно нумериса-

них страна а на крају текста дата је биографија и кључна документацијска информација на укупно 9 ненумерисаних страна.

Магистарска теза садржи следећа поглавља:

1. Библиотечки информациони систем,
2. Радно окружење библиотекара,
3. Команде библиотечког информационог система,
4. Имплементација радног окружења библиотекара,
5. Имплементација команди библиотечког информационог система и
6. Закључак.

У *првом* поглављу укратко су описани захтеви Библиотечке мреже на основу којих је реализован софтверски систем БИСИС. Дат је и кратак преглед развоја досадашње три верзије система БИСИС.

У *другом* поглављу приказан је модел података Базе радног окружења библиотекара формиран у CASE алату Rational Rose 2000. Као илустративни пример приказан је дијаграм класа једне од екранских форми које омогућавају ажурирање података Радног окружења библиотекара. Динамичким дијаграмима UML-а моделирана је функционалност те екранске форме. У прилогу овог поглавља дат је листинг DDL фајла Базе радног окружења.

У *трећем* поглављу дат је дијаграм случајева коришћења команди библиотечког информационог система. Статичка структура дела система која се односи на команде приказана је дијаграмом класа, док је динамика моделирана динамичким дијаграмима UML-а. Спецификација класа дијаграма класа команди дата је у прилогу овог поглавља.

У *четвртом* поглављу описана је имплементација према спецификацији библиотечког окружења датој у другом поглављу. Имплементација је извршена у оквиру библиотечког информационог система БИСИС вер. 3.0 који је имплементиран у Java окружењу. Приказани су и илустративни примери корисничког интерфејса подсистема Радног окружења библиотекара. У прилогу овог поглавља дат је списак атрибута и метода класе која имплементира изглед и функционалност екранске форме *Potpolja tipa publikacije*.

Имплементација команди библиотечког информационог система описана је у *петом* поглављу. Имплементација је извршена према спецификацији датој у трећем поглављу, такође у Java окружењу. Наведени су и илустративни примери коришћења команди библиотечког система.

У *Закључку* је истакнут научни допринос магистарске тезе и правци даљег истраживања.

Централно место тезе је дато у трећем поглављу, у коме је приказана комплетна објектно-оријентисана спецификација за команде delete, display и new. Спецификација захтева дата је преко дијаграма случајева коришћења. Осим тога, статичка структура дела система који се односи на команде илустрована је дијаграмима класа. Динамички аспекти дати су дијаграмима секвенци, сарадње, стања и активности на којима су приказани детаљи везани за функционисање поменутих команди. Аналогно се могу специфицирати остале команде дефинисане у систему. На основу тако формиране спецификације извршена је имплементација команди библиотечког информационог система у Java окружењу. Представљен је изглед екранских форми и описана функционалност за секвенцу позива команди приликом креирања новог записа, његовог снимања у Базу библиографске грађе, ажурирања записа и брисања записа из базе.

Главни научни допринос магистарске тезе је објектно-оријентисано моделирање команди библиотечког информационог система. Моделирање је извршено помоћу одговарајућих UML дијаграма који приказују све аспекте дела система БИСИС који се односи на команде. Основна идеја приликом креирања дијаграма класа је да све класе које имплементирају конкретне команде наслеђују апстрактну класу *AbstractCommand* и имплементирају апстрактну методу execute. На тај начин искоришћен је принцип полиморфизма, који омогућава бржи развој програма, бољу организацију кода, једноставније проширивање и лакше одржавање програма. Додавање нове команде у скуп постојећих команди остварује се без измене класа које имплементирају постојеће команде. Нова команда се додаје тако што класа која имплементира ту команду треба само да наследи апстрактну класу и да имплементира методу *execute*.

Магистарска теза одбрањена је 28.03.2001. године на Природно-математичком факултету, Институт за математику, Нови Сад (ментор проф. др Душан Сурла). Теза је доступна у библиотеци департмана за математику и информатику у Новом Саду и на сајту:

http://diglib.ns.ac.yu/ndltd/docs/set1/ndltd19/magist.pdf

Бранко Милосављевић
Текст сервер UNIMARC записа

Магистарска теза написана је на српском језику (ћирилица), а извод тезе је српски/енглески. Садржи 5 поглавља, 102 страна Б5 формата, са 33 литерарна цитата из предметне области, 13 табела и 33 слика. На почетку текста дата је насловна страна, садржај и предговор на 4 посебно нумерисане стране а на крају текста дат је списак коришћених скраћеница и биографија на 3 стране.

Магистарска теза садржи следећа поглавља:

1. Преглед архитектура текст сервера,
2. Репрезентација и претраживање библиографске грађе,
3. Спецификација текст сервера UNIMARC записа,
4. Имплементација текст сервера UNIMARC записа и
5. Закључак.

У *првом* поглављу дат је преглед архитектура савремених текст сервера. Текст сервери су класификовани према намени и начину рада. Једну класу чине текст сервери намењени претраживању текстуалних садржаја на Интернету. Као карактеристични представници две подврсте оваквих сервера приказани су сервери *AltaVista* и *Yahoo*. Другу класу текст сервера чине сервери намењени претраживању докумената садржаних у оквиру фајл-система. Карактеристичан представник оваквих сервера је *Microsoft Index Server* који је укратко и приказан. Трећа класа текст сервера обухвата сервере намењене претраживању текстуалних докумената садржаних у бази података. Типичан представник ове класе сервера, *Oracle ConText*, је и укратко приказан у овом поглављу.

Основни стандарди које је усвојио текст сервер који је тема овог рада су представљени у *другом* поглављу. Стандард који дефинише рачунарску репрезентацију библиографске грађе је UNIMARC формат. Усвојени стандард за упитни језик текст сервера је упитни језик *Dialog*. Ово поглавље доноси сажете приказе оба стандарда и примере UNIMARC записа и претраживања путем *Dialog* упитног језика.

Треће поглавље садржи објектно оријентисану спецификацију текст сервера UNIMARC записа дату помоћу Обједињеног језика моделирања (UML). Поред UML дијаграма, дат је и вербални опис појединих делова система. Спецификација је подељена на три подсистема: приказ модела података сервера, подсистем за руковање записима и подсистем за претраживање записа.

У *четвртом* поглављу приказан је метод имплементације система на основу дате UML спецификације и одговарајућег CASE алата, програма *Rational Rose*. За имплементацију система је коришћен програмски језик Java и технологија *Enterprise JavaBeans* (EJB) намењена за имплементацију вишеслојних клијент/сервер система. Поред тога, дат је упоредни приказ различитих технологија намењених за имплементацију вишеслојних клијент/сервер система које су доступне из окружења језика Java.

У *Закључку* су истакнути главни резултати добијени у тези и изнето мишљење о даљим правцима развоја овог система.

Основни допринос магистарске тезе је предлог спецификације и имплементације текст сервера намењеног руковању и претраживању UNIMARC записа. Текст сервер је независан од коришћене рачунарске платформе и система за управљање базама података.

Магистарска теза одбрањена је 23.07.1999. године на Факултету техничких наука, Катедра за рачунарство и аутоматику, Нови Сад (ментор проф. др Зора Коњовић). Теза је доступна у библиотеци Факултета техничких наука у Новом Саду и на сајту:

http://diglib.ns.ac.yu/ndltd/docs/set1/ndltd6/textsrv.pdf

Нинослава Савић
Систем за генерисање каталошких листића у библиотечком информационом систему

Магистарска теза написана је на српском језику (латиница), а извод тезе је српски/енглески. Садржи 7 поглавља, 93 стране А4 формата, са 21 литерарна цитата из предметне области и 15 слика. На почетку текста дата је насловна страна и кључна документацијска информација на 4 ненумерисане стране затим садржај и предговор на 5 посебно нумерисаних страна.

Магистарска теза садржи следећа поглавља:

1. Увод,
2. Библиотечки информациони систем,
3. Архитектура система за генерисање каталошких листића,
4. Проширени модел објекти-везе речника каталошких листића,
5. Опис генератора каталошких листића,
6. Имплементација и
7. Закључак.

Прво поглавље лоцира тему у оквиру библиотечког информационог система, описује предмет, циљ, хипотезе и метод истраживања и даје кратак садржај рада.

Друго поглавље садржи опис библиотечког информационог система и опис основне функције *библиотекарско извештавање и документовање*. Детаљно се обрађује основна врста извештаја – каталошки листићи и дефинише синтакса за формални опис каталошких листића. Пример формалне дефиниције каталошких листића односи се на *главни јединствени каталошки листић за монографије* који је највише коришћени и најкомплекснији тип листића.

Треће поглавље описује архитектуру система за генерисање каталошких листића.

У *четвртом* поглављу дат је проширени модел објекти – везе *речника каталочких листића* са описом његове структуре.

Пето поглавље садржи алгоритам за генерисање каталошких листића и детаљан псеудокод програма *генератора каталошких листића*.

У *шестом* поглављу издвојеном за имплементацију система за генерисање каталошких листича коришћено је развојно окружење ORACLE-а, CASE алат DESIGNER 2000 и DEVELOPER 2000. Основни модел *речника каталошких листича* описан у поглављу 4 преведен је у ORACLE-ов ER модел који је улазна величина приликом имплементације модела података. Превођењем ER модела у релациони модел генерисан је скуп шема релација на основу на основу којих су у трећој фази имплементације генерисани командни фајлови SQL-а. Њиховим активирањем креиране су табеле релационог модела *речника каталошких листича* и унети слогови дефиниције каталошких листића. У наставку је илустративним примером дат начин на који аналитичар формира *речник каталошких листића*. Опис корисничког интерфејса за рад са речником завршен је имплементацијом неколико екранских форми чиме је илустрован начин њиховог генерисања.

Седмо поглавље објашњава место и улогу предложеног система за *генерисање каталошких листича* у библиотечком информационом систему и даје могуће правце даљег развоја овог система.

Централно место тезе представља пројектовање проширеног модела објекти-везе *речника каталошких листича* као носиоца синтаксе каталошких листића. Поред спецификације извршена је и његова комплетна имплементација.

Научни допринос магистарске тезе огледа се у томе што даје заокружени приказ начина реализације једног речником вођеног система, при том се ослањајући на сложени реални информациони систем у оквиру актуелног развојног пројекта. Овај приступ изабран је насупрот класичном приступу у коме је домимантна комиуникација корисник – пројектант – програмер и у коме нова решења настају на основу корисничких захтева и активности програмера. Речником вођен систем треба да обезбеди самосталност кориснику – аналитичару и смањи време које се улаже у послове адаптивног одржавања софтвера, уколико се у пракси покаже да има задовољавујуће перформансе.

Магистарска теза одбрањена је 18.03.1999. године на Факултету организационих наука, Београд (ментор проф. др Бранислав Лазаревић). Теза је доступна у библиотеци Факултета организационих наука, Београд.

Милан Видаковић
Објектно оријентисана спецификација и имплементација корисничког интерфејса за библиотечки информациони систем

Магистарска теза написана је на српском језику (ћирилица), а извод тезе је српски/енглески. Садржи 5 поглавља, 185 страна Б5 формата, са 35 литерарна цитата из предметне области, 4 табеле, 30 слика и 1 прилог. На почетку текста дата је насловна страна, садржај и предговор на 6 посебно нумерисаних страна а на крају текста дата је биографија и кључна документацијска информација на укупно 10 ненумерисаних страна.

Магистарска теза садржи следећа поглавља:

1. Увод,
2. Објектно оријентисана спецификација корисничког интерфејса,
3. Имплементација корисничког интерфејса,
4. Имплементација система моделираног помоћу UML-а и
5. Закључак.

У *првом* поглављу је дат кратак преглед основних области са којима се треба упознати при решавању постављеног задатка. Дат је кратак преглед објектно оријентисаних метода за пројектовање информационих система, преглед програмирања у Интернет окружењу са описом најчешће коришћених софтверских алата, као и приказ основних концепата библиотечког информационог система.

У *другом* поглављу је дата објектно оријентисана спецификација корисничког интерфејса библиотечког информационог система (БИС), применом уједињеног језика моделирања (*Unified Modeling Language* – UML). Дат је предлог специфицирања изгледа екранске форме употребом алата за брз развој апликација (*Rapid Application Development* – RAD). У овом поглављу је дата спецификација изгледа екранских форми употребом RAD алата, као и предлог спецификације функционалности екранских форми употребом коначног аутомата стања.

Треће поглавље описује имплементацију објектно оријентисане спецификације корисничког интерфејса БИС-а у програмски језик Java. Дат је опис имплементације употребљених концепата UML-а кроз документацију генерисану алатом за аутоматско генерисање документације на основу изворног кода (*javadoc* програм).

Четврто поглавље садржи детаљан преглед концепата UML-а и могућности њихове имплементације у програмском језику Java.

Пето поглавље садржи закључке везане за објектно оријентисану спецификацију и имплементацију корисничког интерфејса, као и поглед на даље правце развоја у овој области.

За спецификацију информационих захтева усвојен је Обједињени језик моделирања, а за имплементацију програмски језик Java. У тези је доследно примењена објектно орјентисана спецификација и имплементација корисничког интерфејса библиотечког информационог система. Овај систем је послужио као пример на којем је верификован предложени приступ спецификације и имплементације корисничког интерфејса и за друге информационе системе. Такође је детаљно описана веза између објектно-оријентисане спецификације и објектно-оријентисане имплементације информационих система.

Главни научни допринос добијен у магистарској тези је следећи:

- Предложен је један методолошки приступ за објектно орјентисану спецификацију и имплементацију корисничког интерфејса.
- Предложени приступ је верификован на примеру сложеног корисничког интерфејса за библиотечки информациони систем.
- Извршена је детаљна анализа могућности и дат предлог имплементације система описаног обједињеним језиком моделирања у програмски језик Java.

Магистарска теза одбрањена је 04.12.1998. године на Факултету техничких наука, катедра за рачунарство и аутоматику, Нови Сад (ментор проф.

др Зора Коњовић). Теза је доступна у библиотеци Факултета техничких наука у Новом Саду и на сајту:

http://diglib.ns.ac.yu/ndltd/docs/set1/ndltd7/MinjaMagistarski.PDF

Драгана Тодорић
Узајамна каталогизација и корисничко претраживање библиографских података

Магистарска теза написана је на српском језику (ћирилица), а извод тезе је српски/енглески. Садржи 7 поглавља, 120 страна Б5 формата, са 22 литерарна цитата из предметне области и 37 слика. На почетку текста дата је насловна страна, садржај и предговор на 4 посебно нумерисане стране а на крају текста дата је биографија и кључна документацијска информација на укупно 9 ненумерисаних страна.

Магистарска теза се састоји од следећих поглавља:

1. Увод,
2. Узајамна каталогизација,
3. Корисничко претраживање,
4. Имплементација подсистема за узајамну каталогизацију,
5. Имплементација WWW сервера за корисничко претраживање,
6. Коришћење WWW сервера за корисничко претраживање и
7. Закључак.

У *уводном* поглављу описана је архитектура библиотечког информационог система, као и подсистеми за узајамну каталогизацију и корисничко претраживање. Пошто је узајамна каталогизација Интернет апликација дат је кратак осврт на рачунарске мреже. За реализацију корисничког претраживања изабран је *World Wide Web* (WWW), те су у овом поглављу представљене његове основне карактеристике.

Друго поглавље садржи спецификацију информационих захтева за формирање библиографске грађе и сервера за узајамну каталогизацију. Описан је поступак преузимања записа преко рачунарске мреже из неке од удаљених база библиографске грађе.

Детаљна спецификација сервера за корисничко претраживање дата је у *трећем* поглављу. Спецификација је дата тако да је погодна за имплементацију Интернет апликације.

Четврто и *пето* поглавље садрже опис имплементације узајамне каталогизације и корисничког претраживања. Имплементација је урађена у про-

грамском језику C на SCO UNIX платформи. Корисничко претраживање је имплементирано као WWW сервер. Представљена је структура HTML докумената који се користе у комуникацији клијента и сервера.

Шесто поглавље садржи детаљно упутство за употребу WWW сервера за корисничко претраживање. Представљен је изглед екранских форми за различите примере претраживања и извештавања који се могу јавити при његовој употреби. Типови извештаја који су на располагању корисницима су такође детаљно објашњени.

Последње, *седмо* поглавље садржи закључке о спецификацији информационих захтева сервера за узајамну каталогизацију и сервера за корисничко претраживање и могућности примене ових сервера за рад са различитим базама података и у различитим информационим системима.

За спецификацију информационих захтева коришћена је објектно-оријентисана методологија а за имплементацију коришћене су Интернет технологије за развој апликација. За формирање библиографске грађе у оквиру библиотечке мреже користи се узајамна каталогизација. У тези је дато комплетно решење сервера за узајамну каталогизацију. Поред тога, значајно место у библиотечком систему заузима корисничко претраживање. У тези је детаљно описана спецификација и имплементација WWW сервера за корисничко претраживање.

Главни научни допринос магистарске тезе приказан је у другом поглављу. Дата је детаљна објектно-оријентисана спецификација сервера за узајамну каталогизацију. На основу ове спецификације имплементиран је сервер који омогућује пренос библиографских података путем рачунарске мреже по UNIMARC формату. На исти начин, имплементиран је и WWW сервер за корисничко претраживање библиографских података, такође по UNIMARC формату. При употреби WWW сервера за рад са библиотечким системима који користе други формат библиографских података, неопходно је реализовати конверзију података према UNIMARC формату.

Идеје за реализацију сервера за узајамну каталогизацију и сервера за корисничко претраживање могу се директно или уз извесне модификације искористити и за друге Интернет апликације. Основне карактеристике ових апликација су електронска размена података (докумената) и управљање пословним системима.

Магистарска теза одбрањена је 12.01.1998. године на Природно-математичком факултету, Институт за математику, Нови Сад (ментор проф. др

Душан Сурла). Теза је доступна у библиотеци Департмана за математику и информатику у Новом Саду.

Илдико Холо
Командни језик библиотечког информационог система

Магистарска теза написана је на српском језику (ћирилица), а извод тезе је српски/енглески. Садржи 6 поглавља, 151 страна Б5 формата, са 29 литерарна цитата из предметне области, 1 табелу, 28 слика и 3 прилога. На почетку текста дата је насловна страна, садржај и предговор на 4 посебно нумерисане стране а на крају текста дата је биографија и кључна документацијска информација на укупно 9 ненумерисаних страна.

Магистарска теза садржи следећа поглавља:

1. Библиотечки систем,
2. UNIMARC формат,
3. Статички модел библиотечког система,
4. Објектно-оријентисана спецификација командног језика библиотечког система,
5. Имплементација и
6. Закључак.

У *првом* поглављу је дат приказ архитектуре и дијаграми случајева коришћења (*use cases*) библиотечког информационог система. Приказан је дијаграм случаја коришћења који описује функционалност *Формирања библиографске грађе*, као једне од основних функција у оквиру којег се позивају команде библиотечког информационог система. Описане су главне карактеристике командног језика. Поред тога, дат је скуп команди који се користе у оквиру библиотечког система.

UNIMARC формат, описан у *другом* поглављу, представља универзални формат каталогизације библиотечког система, који служи за бележење библиографских података у машински читљивом облику и омогућије размену између националних и међународних библиографских установа. Поред тога, представљен је и пратећи формат за формирање нормативних података, као и коресподенцију између поља библиографских и нормативних записа.

У *трећем* поглављу је представљен статички модел библиотечког информационог система. Као први сегмент, описани су дијаграми класа, који представљају структуру UNIMARC формата за формирање библиографске грађе, односно структуру пратећег формата за формирање норнативних пода-

така. Даље је приказан дијаграм класа Радног окружења приликом формирања записа. На основу наведених дијаграма класа, формирају се структуре записа, који су представљени дијаграмима класа структурираних записа библиографске грађе и нормативних података.

Објектно-оријентисана спецификација командног језика библиотечког система представљена је у *четвртом* поглављу. Прво је приказана општа идеја пројектовања командног језика, а потом је она примењена у библиотечком информационом систему.

У оквиру *петог* поглавља је илустрована имплементација конандног језика библиотечког информационог система. Представљен је изглед екранских форми и детаљан опис функциомалности за секвенцу позива команди приликом ажурирања записа, као обрађеног библиотечког документа.

Последње, *шесто* поглавље садржи опис главних резултата наведених по поглављима.

За спецификацију информационих захтева коришћена је објектно-оријентисана методологија (*Unified Modeling Language*), а за имплементацију Интернет технологије. У тези је дата комплетна спецификација статичког модела за: структуру UNIMARC формата, Радног окружења приликом формирања записа, Структуре записа библиотечке грађе и нормативних података, као и статичког и динамичког модела командног језика библиотечког информационог система.

Главни научни допринос магистарске тезе је детаљна објектно-оријентисана спецификација командног језика библиотечког система. Дата спецификација може се применити и на друге информационе системе. У том случају, потребно је само променити спецификацију класа за конкретно уведене команде усвојеног командног језика.

Магистарска теза одбрањена је 11.03.1998. године на Природно-математичком факултету, Институт за математику, Нови Сад (ментор проф. др Душан Сурла). Теза је доступна у библиотеци Департмана за математику и информатику у Новом Саду.

Татјана Вулић
Извештавање и документовање у библиотечком информационом систему

Магистарска теза написана је на српском језику (ћирилица), а извод тезе је српски/енглески. Садржи 5 поглавља, 122 стране Б5 формата, са 37 литерарна цитата из предметне области, 3 табеле и 27 слика. На почетку

текста дата је насловна страна, садржај и предговор на 4 посебно нумерисане стране а на крају текста дата је биографија и кључна документацијска информација на укупно 9 ненумерисаних страна.

Магистарска теза садржи следећа поглавља:

1. Увод,
2. Извештавање и документовање,
3. Објектно-оријентисана спецификација подсистема за извештавање,
4. Имплементација и
5. Закључак.

У *првом* поглављу дат је кратак преглед стандарда који се користи у библиотекарству, а значајни су за извештавање и документовање. Такође, дат је преглед објектно-оријентисаних метода за развој информационих система, са посебним освртом на Обједињени језик моделирања (*Unified Modeling Language* – UML) који је у тези усвојен за спецификацију информационих захтева. Дата је архитектура (дијаграм развоја) и функционалност (дијаграм случаја коришћења – *use case*) библиотечког информационог система.

У *другом* поглављу су детаљно описане специфичности извештавања у библиотечком информационом систему, као и дефиниције типова извештаја по усвојеном стандарду.

Централно место у тези је *треће* поглавље које садржи објектно-оријентисану спецификацију подсистема за извештавање и документовање. Приказане су фазе дефинисања захтева и анализа наведеног подсистема. Фаза дефинисања захтева приказана је преко дијаграма случајева коришћења и њихових текстуалних описа. У оквиру анализе подсистема за извештавање, приказани су дијаграми класа за структуру и правила извештавања, као и детаљна спецификација тих класа. Такође, приказан је и динамички модел подсистема за извештавање преко дијаграма секвенци, дијаграма сарадње и дијаграма активности. Ради комплетности, приказана је и структура оних подсистема библиотечког информационог система од којих зависи подсистем за извештавање.

У *четвртом* поглављу илустрована је имплементација подсистема за извештавање и документовање библиотечког информациомог система.

Пето поглавље садржи закључке везане за проблеме извештавања. Разматрана је могућност примене предложене спецификације и на друге информационе системе.

За спецификацију информационих захтева коришћена је објектно-оријентисана метода. У тези је по UML-у описана комплетна спецификација статичког и динамичког модела подсистема за извештавање у библиотечком информационом систему. Такође, приказани су и статички модели за подсистеме UNIMARC формат, Структурирани библиографски запис и Библиографска грађа.

Главни научни допринос магистарске тезе је детаљна објектно-оријентисана спецификација за дефинисање сложених извештаја у информационим системима. Предложено решење је примењено на библиотечки информациони систем, а може се применити и на друге информационе системе. У том случају, потребно је дефинисати стандарде и правила креирања извештаја посматраног система.

Магистарска теза одбрањена је 11.03.1998. године на Природно-математичком факултету, Институт за математику, Нови Сад (ментор проф. др Душан Сурла). Теза је доступна у библиотеци Департмана за математику и информатику у Новом Саду.

Ивана Тричковић
Примена Петри мрежа у спецификацији динамике информационих система

Магистарска теза написана је на српском језику (ћирилица), а извод тезе је српски/енглески. Садржи 6 поглавља, 150 страна Б5 формата, са 24 литерарна цитата из предметне области и 62 слике. На почетку текста дата је насловна страна и кључна документацијска информација на 7 ненумерисаних страна, затим садржај и предговор на 6 посебно нумерисаних страна.

Магистарска теза садржи следећа поглавља:

1. Обједињена методологија моделирања информационих система,
2. Објектно-оријентисана спецификација библиотечког информационог система,
3. Петри мреже и моделирање динамике система,
4. Објектно-оријентисана трансформациона методологија пројектовања информационих система,
5. Имплементација корисничког интерфејса библиотечког информационог система и
6. Закључак.

У *првом* поглављу описана је Обједињена методологија моделирања (*Unified Modelinng Language*). Престављени су основни концепти моделирања

којима се описује семантика проблема, као и дијаграми којима се описују статичке и динамичке карактеристике система. Ова методологија обухвата различите дијаграме: дијаграм класа, дијаграм објеката, дијаграм случаја коришћења (*use case*), дијаграм секвенци, дијаграм сарадње, дијаграм стања, дијаграм активности, дијаграм компонената и дијаграм размештаја.

Објектно-оријентисана спецификација библиотечког информационог система помоћу Обједињене методологије моделирања описана је у *другом* поглављу. Детаљно су дати статички и динамички модел сегмента тог система који се односи на претраживање библиографске грађе.

У *трећем* поглављу дати су елементи теорије Петри мрежа и предлог трансформације динамичког модела Обједињене методологије моделирања у модел Петри мреже. Динамички модел подсистема за претраживање библиотечке грађе дат у другом поглављу, у овом поглављу је представљен моделом Петри мреже.

У *четвртом* поглављу дата је спецификација библиотечког информационог система помоћу објектно-оријентисане методологије. Декомпозиција система је извршена методом структурне систем анализе. Дефинисани су апликативни, уграђени и базни објекти за подсистем претраживање библиотечке грађе. Динамика овог подсистема описана је у спецификацији наведених објеката. Дат је поступак за превођење ове динамике у модел Петри мреже.

У *петом* поглављу илустрована је имплементација објектно-оријентисане спецификације корисничког интерфејса.

Последње, *шесто* поглавље садржи закључке на основу приказаних резултата, као и даље могућности примене Петри мрежа у анализи и пројектовању информационих система.

У тези су детаљно приказане могућности примене Петри мрежа за опис динамике информационих система. Ова теорија обезбеђује математички формализам који омогућава формирање математичке презентације динамике система. Анализом Петри мреже могу се добити информације о структури и динамичком понашању моделираног система. На основу ових информација могу се проценити, односно побољшати перформансе моделираног система. Значај оваквог моделирања је у томе што иза њега стоји потпуни математички формализам. Ово омогућава израду софтверског производа за анализу и пројектовање информационих система.

Главни резултат магистарске тезе приказан је у трећем поглављу. Предложен је формални поступак превођења динамичког модела Обједињене

методологије у модел Петри мреже. Такође, у четвртом поглављу, где је динамика система описана на други начин, дат је предлог како се та динамика може трансформисати у модел Петри мреже. Овим је дат значајан научни допринос у области моделирања динамике информационих система.

Магистарска теза одбрањена је 08.07.1997. године на Природно-математичком факултету, Институт за математику, Нови Сад (ментор проф. др Душан Сурла). Теза је доступна у библиотеци Департмана за математику и информатику у Новом Саду.

Прилог 2

Библиографија

У овом Прилогу наведени су објављени научни резултати истраживања на развоју библиотечког информационог система БИСИС. Радови су разврстани по вредности индикатора научне компететности Министарства за науку и технолошки развој републике Србије.

Монографија националног значаја (R13)

1. Д. Сурла, З. Коњовић, уредници. *Дистрибуирани библиотечки информациони систем БИСИС*. Група за информационе технологије, Нови Сад, 2004.
2. Z. Konjović, D. Surla, G. Popović-Bošković, A. Marinković, D. Todorić, Lj. Milivojević, D. Prlja. *Dizajniranje i održavanje bibliotečke Web prezentacije*. CD-ROM, Zajednica biblioteka univerziteta u Srbiji, Beograd, 1999.
3. Đ. Paunović, B. Lazarević, D. Surla, M. Ivković, D. Korunović. *Formiranje i pretraživanje baza podataka u sistemu naučnih i tehnoloških informacija Srbije*. redaktor Branislav Lazarević, Ministarstvo za nauku i tehnologiju Republike Srbije, Beograd, 1996.

Поглавље у монографији националног значаја (R23)

1. Д. Сурла, З. Коњовић. *Преглед развоја библиотечког информационог система БИСИС*, Поглавље 1, монографија *Дистрибуирани библиотечки информациони систем БИСИС*, уредници Д. Сурла, З. Коњовић. Група за информационе технологије, Нови Сад, 2004.
2. Б. Пуповац. *YUMARC формат*, Поглавље 2, монографија *Дистрибуирани библиотечки информациони систем БИСИС*, уредници Д. Сурла, З. Коњовић. Група за информационе технологије, Нови Сад, 2004.
3. Б. Пуповац. *Моделирање и имплементација радног окружења библиотекара*, Поглавље 3, монографија *Дистрибуирани библиотечки информациони систем БИСИС*, уредници Д. Сурла, З. Коњовић. Група за информационе технологије, Нови Сад, 2004.
4. М. Видаковић. *Обрада библиографске грађе*, Поглавље 4, монографија *Дистрибуирани библиотечки информациони систем БИСИС*, уредници Д. Сурла, З. Коњовић. Група за информационе технологије, Нови Сад, 2004.

5. Б. Пуповац. *Команде библиотечког информационог система*, Поглавље 5, монографија *Дистрибуирани библиотечки информациони систем БИСИС*, уредници Д. Сурла, З. Коњовић. Група за информационе технологије, Нови Сад, 2004.
6. М. Видаковић. *Сервер за узајамну каталогизацију*, Поглавље 6, монографија *Дистрибуирани библиотечки информациони систем БИСИС*, уредници Д. Сурла, З. Коњовић. Група за информационе технологије, Нови Сад, 2004.
7. Т. Зубић, М. Рацковић. *Извештавање о библиографској грађи*, Поглавље 7, монографија *Дистрибуирани библиотечки информациони систем БИСИС*, уредници Д. Сурла, З. Коњовић. Група за информационе технологије, Нови Сад, 2004.
8. Б. Милосављевић. *Текст сервер UNIMARC записа*, Поглавље 8, монографија *Дистрибуирани библиотечки информациони систем БИСИС*, уредници Д. Сурла, З. Коњовић. Група за информационе технологије, Нови Сад, 2004.
9. Б. Милосављевић. *Корисничко претраживање*, Поглавље 9, монографија *Дистрибуирани библиотечки информациони систем БИСИС*, уредници Д. Сурла, З. Коњовић. Група за информационе технологије, Нови Сад, 2004.
10. Т. Тошић, Д. Сурла. *Коришћење библиотечке грађе*, Поглавље 10, монографија *Дистрибуирани библиотечки информациони систем БИСИС*, уредници Д. Сурла, З. Коњовић. Група за информационе технологије, Нови Сад, 2004.
11. М. Видаковић, Б. Милосављевић. *Интернационализација и локализација система БИСИС*, Поглавље 11, монографија *Дистрибуирани библиотечки информациони систем БИСИС*, уредници Д. Сурла, З. Коњовић. Група за информационе технологије, Нови Сад, 2004.
12. М. Керац, И. Нејгебауер. *Мрежна инфраструктура система БИСИС*, Поглавље 12, монографија *Дистрибуирани библиотечки информациони систем БИСИС*, уредници Д. Сурла, З. Коњовић. Група за информационе технологије, Нови Сад, 2004.
13. Д. Сурла, Д. Тодорић**.** *Библиотека на Интернету*, Поглавље 7, монографија *Интернет и савремено пословање*, редактори М. Ивковић и Б. Раденковић, Технички факултет ”Михајло Пупин”, Зрењанин, 1998.

Признати програмски систем на међународном нивоу (R31)

1. Библиотечки софтверски систем БИСИС верзија 3

Рад у водећем часопису међународног значаја (R51)

1. M. Jakšić**.** *Modelling of Bibliographical Standards in XML Technology*. Software Practice and Experience, 2004. (in print).

Рад у часопису међународног значаја (R52)

1. M. Zeremski, D. Surla. *Validation of XML bibliographic records in Java environment,* Novi Sad Journal of Mathematics, 2004. (in print).
2. B. Milosavljević, Z. Konjović. *Modeling and Implementation of a System for Bibliographic Records Interchange.* Novi Sad Journal of Mathematics, Vol. 31, No. 1, 2000. pp. 159-166.
3. A. Kovčić, B. Milosavljević. *Data Model for Indexing and Searching XML Documents.* Novi Sad Journal of Mathematics, Vol. 31, No. 1, 2000. pp. 151-158.
4. I. Tričković. *The Transformation of the State Diagram of the UML into the Petri Net Model*. Novi Sad Journal of Mathematics, Vol. 28, No. 3, 1998. str. 187-195.
5. D. Todorić. *The Server Specification for Information Retrieval of Bibliographic Data*, Novi Sad Journal of Mathematics, Vol. 28, No. 3, 1998. pp. 177-186.
6. I. Fišl, Z. Konjović, D. Surla. *Design and Implementation of the Query Language Interpreter for Bibliographic Data Retrieval*, Novi Sad Journal of Mathematics, Vol. 28, No. 3, 1998. pp. 11-20.

Пленарно предавање по позиву на скупу међународног значаја штампано у целини (R53)

1. D. Surla, Z. Konjović. *An Overview of the Development of the Library Information System BISIS*. Intl. Conf. on Distributed Library Information Systems, Ohrid, June 1-6, 2004. (in print).
2. G. Budimir. *Quality Control of XML Bibliographic Records*. Intl. Conf. on Distributed Library Information Systems, Ohrid, June 1-6, 2004. (in print).

Рад саопштен на скупу међународног значаја штампан у целини (R54)

1. M. Vidaković, T. Zubić, B. Milosavljević, B. Pupovac, T. Tošić. *Processing Bibliographic Documents in the Library Information System BISIS*. Intl. Conf. on Distributed Library Information Systems, Ohrid, June 1-6, 2004. (in print).
2. B. Pupovac. *Maintenance of the YUMARC Format*. Intl. Conf. on Distributed Library Information Systems, Ohrid, June 1-6, 2004. (in print).
3. B. Pupovac. *Maintenance of the Librarian Working Environment*. Intl. Conf. on Distributed Library Information Systems, Ohrid, June 1-6, 2004. (in print).
4. B. Milosavljević. *User Search in the Library Information System BISIS*. Intl. Conf. on Distributed Library Information Systems, Ohrid, June 1-6, 2004. (in print).
5. B. Milosavljević, M. Vidaković. *Administrating the Library Information System BISIS*. Intl. Conf. on Distributed Library Information Systems, Ohrid, June 1-6, 2004. (in print).

6. B. Milosavljević, M. Vidaković. *Localization of the Library Information System BISIS*. Intl. Conf. on Distributed Library Information Systems, Ohrid, June 1-6, 2004. (in print).
7. M. Kerac, I. Nejgebauer. *Network Infrastructure Design for a BISIS System in an Inter-Library Scenario*. Intl. Conf. on Distributed Library Information Systems, Ohrid, June 1-6, 2004. (in print).
8. S. Komazec, D. Okanović, Z. Protić, G. Sladić. *Software Architecture of the Digital Library of Theses and Dissertations*. Intl. Conf. on Distributed Library Information Systems, Ohrid, June 1-6, 2004. (in print).
9. B. Milosavljević, S. Komazec, Z. Protić, G. Sladić, D. Okanović. *Document Processing in the Digital Library of Theses and Dissertations*. Intl. Conf. on Distributed Library Information Systems, Ohrid, June 1-6, 2004. (in print).
10. M. Zarić, D. Okanović, Z. Protić. *Searching the Digital Library of Theses and Dissertations*. Intl. Conf. on Distributed Library Information Systems, Ohrid, June 1-6, 2004. (in print).
11. M. Zarić. *Metadata Dissemination Using OAI-PMH*. Intl. Conf. on Distributed Library Information Systems, Ohrid, June 1-6, 2004. (in print).
12. G. Sladić, M. Vidaković. *Integrated Search in the Digital Libraries System*. Intl. Conf. on Distributed Library Information Systems, Ohrid, June 1-6, 2004. (in print).
13. B. Milosavljević, S. Komazec, Z. Konjović. *Portal of the Interuniversity Library Network*. Intl. Conf. on Distributed Library Information Systems, Ohrid, June 1-6, 2004. (in print).
14. B. Milosavljević, Z. Konjović. *Multimedia Document Retrieval in the Networked Digital Library of Theses and Dissertations*. 4th Intl. Conf. on Informatics and Information Technology, Bitola, 2003. (in print)
15. M. Vidaković, Z. Konjović. *One Implementation of Virtual Library Based on Agent Technology*. 4th Intl. Conf. on Informatics and Information Technology, Bitola, 2003. (in print)
16. S. Komazec, G., Sladić. *Design Patterns in Implementation of Digital Library of Theses and Dissertations*. 4th Intl. Conf. on Informatics and Information Technology, Bitola, 2003. (in print)
17. Z. Protić, D. Okanović. *Modeling and Implementation of JSP pages in the System Digital Library of Theses and Dissertations*. 4th Intl. Conf. on Informatics and Information Technology, Bitola, 2003. (in print)
18. G. Sladić, S. Komazec. *Modeling and Implementation of Entity Beans in Software System for Networked Digital Library of Theses and Dissertations*. 4th Intl. Conf. on Informatics and Information Technology, Bitola, 2003. (in print)
19. D. Okanović, B. Milosavljević. *Digital Library of Theses and Dissertations – Data Model*. 4th Intl. Conf. on Informatics and Information Technology, Bitola, 2003. (in print)
20. M. Zarić, D. Surla. *Java Implementation of the Protocol for Metadata Harvesting*. 4th Intl. Conf. on Informatics and Information Technology, Bitola, 2003. (in print)

21. Z. Konjović, D. Surla. *An Overview of the Development of the Library Information System BISIS*. 4th Intl. Conf. on Informatics and Information Technology, Bitola, 2003. (in print)
22. B. Milosavljević, Z. Konjović. *Design of an Extensible Multimedia Information Retrieval System*. 6th Balkan Conf. on Operational Research, Thessaloniki, 2002. (in print).
23. B. Milosavljević, Z. Konjović. *Design of an XML-Based Extensible Multimedia Information Retrieval System*. IEEE Multimedia Software Engineering (MSE2002), Newport Beach, CA, 2002. pp. 114-121.
24. I. Tričković, D. Surla. *An Application of Petri Nets to the Formal Specification of the System Dynamics*. IASTED Conf. on Software Engineering, Las Vegas, 1998. ACTA Press, pp. 250-253.
25. I. Tričković. *Data model of UNIMARC format*. IASTED Conf. on Software Engineering, Las Vegas, 1998. pp. 246-249.
26. D. Surla, D. Todorić. *Presentation of Searching Report of the Documents Database in World Wide Web*. East-West International Conference Information Technology in Design, EWITD'96, Moscow, 1996. pp. 11-16.

Рад у часопису националног значаја (R62)

1. Д. Сурла, З. Коњовић, Б. Милосављевић, М. Зарић, Г. Сладић, З. Протић, С. Комазец, Д. Окановић. *Приказ реализације мрежне дигиталне библиотеке докторских, магистарских и дипломских радова.* ИНФОТЕКА, Београд, бр. 1-2/2004. стр. 75-86.
2. Б. Милосављевић, З. Коњовић. *Претраживање мултимедијалних информација у систему дигиталних библиотека магистарских и докторских радова*. ИНФОТЕКА, Београд, бр. 1-2/2004. стр. 87-98.
3. М. Зарић, Д. Сурла. *Имплементација протокола за прикупљање метаподатака у мрежи дигиталних библиотека теза и дисертација*. ИНФОТЕКА, Београд, бр. 1-2/2004. стр. 99-112.
4. М. Видаковић, З. Коњовић. *Предлог решења централног каталога за локалне базе библиотечких записа.* ИНФОТЕКА, Београд, бр. 1-2/2004. стр. 113-119.
5. Г. Будимир. *MARC записи и XML.* ИНФОТЕКА, Београд, бр. 1-2/2004. стр. 121-136.
6. М. Зеремски**.** *Преглед развоја дигиталних библиотека базираних на XML технологијама*. ИНФОТЕКА, Београд, 4(2003) 2, стр. 169-175.
7. М. Зеремски, Д. Сурла. *Моделирање библиографских записа у облику XML докумената.* ИНФОТЕКА, Београд, 2(2001) 1-2, стр. 93-97.
8. B. Pupovac. *Radno okruženje bibliotekara*. INFOTEKA, Beograd, 2001. str. 105-109.

9. J. Felbapov. *Primena XML tehnologije u bibliotečkom izveštavanju.* INFOTEKA, Beograd, 2(2001)1-2, str. 111-119.
10. T. Zubić, G. Popović-Bošković. *Modeliranje i implementacija izveštaja o bibliografskoj građi*. INFOTEKA, Beograd, Septembar, 2000. str 59-66.
11. T. Tošić, J. Srejić, T. Anisijević. *Korišćenje bibliotečke građe*. INFOTEKA, Beograd, Septembar, 2000. str 31-38.
12. B. Milosavljević. *Uzajamna katalogizacija: preuzimanje bibliografskih zapisa*. INFOTEKA, 1(2000)1, str. 19-23.
13. B. Milosavljević, Z. Konjović. *Arhitektura WWW prezentacije za pretraživanje u bibliotečkom informacionom sistemu*. Info Science, 2000, 8(4):18-23.
14. D. Surla, M. Bender. *Baze podataka sekundarnih dokumenata*, JISA Info, 6/1994, str. 44-46.

Рад саопштен на скупу националног значаја штампан у целини (R65)

1. D. Okanović, G. Sladić. *Servisi digitalne biblioteke u bibliotečkoj mreži*. Zbornik radova YU INFO ‘04, Kopaonik, 2004.
2. G. Sladić, M. Zarić. *Integrisano pretraživanje u bibliotečkoj mreži.* Zbornik radova YU INFO ‘04, Kopaonik, 2004.
3. Z. Protić, T. Šušnjević. *Servisi klasičnih biblioteka u bibliotečkoj mreži.* Zbornik radova YU INFO ‘04, Kopaonik, 2004.
4. S. Komazec, B. Milosavljević. *Arhitektura portala bibliotečke mreže.* Zbornik radova YU Info 2004, Kopaonik, 2004.
5. S. Škrbić, D. Surla. *Storing XML bibliographic records*, XVI Conference on Applied Mathematics, Budva, 2004. (in print).
6. V. Mijić, D. Surla. *Interpretation of W3C XML Schema Object Model*. Zbornik radova SYM-OP-IS 2003, Herceg Novi, 2003. pp. 199-202. (in Serbian)
7. V. Mijić, D. Surla. *Implementation of W3C XML Schema Object Model.* Zbornik zadova SYM-OP-IS 2003, Herceg Novi, 2003. pp. 203-206. (in Serbian)
8. V. Mijić, D. Surla. *Development of Internet Application Illustrated on Curriculums of University Education*. Zbornik radova YU INFO’03, Kopaonik, 2003.
9. V. Mijić, D. Surla. *Memory Representation of XML Schema with Multi Schema Validator*. Zbornik radova YU INFO’03, Kopaonik, 2003.
10. T. Zubić, D. Surla. *Modeliranje i implementacija sistema za unos podataka sa kataloških listića u XML tehnologiji*. Zbornik radova YU INFO ‘03, Kopaonik, 2003.
11. M. Vidaković, B. Milosavljević, Z. Konjović. *Internacionalizacija Java aplikacija*. Zbornik radova YU Info'03, Kopaonik, 2003.
12. M. Zeremski, D. Surla. *Validacija bibliografskih zapisa pomoću XML Schema jezika.* Zbornik radova YU INFO 2002, Kopaonik, 2002.

13. M. Zeremski, S. Škrbić. *Conversion between XML document of UNIMARC format and relational database.* XV Conference on Applied Mathematics, Zlatibor, 2002. (in print).
14. J. Vidaković, M. Racković. *Modeling The Concepts Of The Bibliographical Cataloguing Cards Using XML Schema Language.* XV Conference on Applied Mathematics – PRIM 2002, Zlatibor, 2002. (in print).
15. M. Vidaković, B. Milosavljević, Z. Konjović. *Implementacija servera za uzajamnu katalogizaciju u EJB tehnologiji*. Zbornik radova YU INFO'02, Kopaonik, 2002.
16. T. Tošić. *Modeling the library collection units reservation process*. XV Conference on Applied Mathematics, Zlatibor, 2002. (in print)
17. B. Pupovac, V**.** Mijić. *Updating the UNIMARC format defined as the XML document.* Proceedings of the XV Conference on Applied Mathematics PRIM'02, Zlatibor, 2002.
18. B. Milosavljević, M. Vidaković, D. Surla. *Rukovanje višejezičnim tekstom u Bibliotečkom informacionom sistemu BISIS ver. 3.* Zbornik radova Internet i ćirilica: srpski jezik, pismo i kultura u savremenim informacionim tehnologijama, Narodna biblioteka Srbije, Beograd, 2002. http://www.rastko.org.yu/projekti/cirilica2/net-cirilica2002/bmilosavljevic.html.
19. J. Felbapov**,** M. Racković. *Primena XML tehnologije u formiranju kataloških listića*. Zbornik radova YU INFO 2002, Kopaonik, 2002.
20. M. Zeremski, D. Surla. Modeliranje *UNIMARC formata u XML – Data jeziku.* Zbornik radova YU INFO 2001, Kopaonik, 2001.
21. M. Zeremski, D. Surla. *Modeliranje UNIMARC formata pomoću XML Schema jezika.* Zbornik radova SYM-OP-IS 2001, Beograd, 2001. str. 261-264.
22. M. Zeremski, D. Surla. *Modeliranje bibliografskih zapisa pomoću XML Schema jezika.* Zbornik radova SYM-OP-IS 2001, Beograd, 2001. str. 265-268.
23. M. Vidaković, Z. Konjović. *Server za uzajamnu katalogizaciju u bibliotečkom informacionom sistemu BISIS*. Zbornik radova YU INFO'01, Kopaonik, 2001.
24. D. Surla, J. Felbapov. *The Class Diagram for the Circulation of Bibliographical Documents*. XIV Conference on Applied Mathematics, Novi Sad, 2001. pp. 48-53.
25. B. Pupovac, D. Surla. *Radno okruženje bibliotekara*. Zbornik radova VJ INFO'01, Beograd, 2001.
26. B. Pupovac, D. Surla. *Model slučajeva korišćenja komandi bibliotečkog informacionog sistema*. Zbornik radova YU INFO'01, Kopaonik, 2001.
27. B. Pupovac. *Objektno-orijentisana specifikacija komande new u bibliotečkom informacionom sistemu*. Zbornik radova VJ INFO'01, Beograd, 2001.
28. B. Pupovac. *Objektno-orijentisana specifikacija komandi bibliotečkog informacionog sistema*. Zbornik radova YU INFO'01, Kopaonik, 2001.
29. D. Surla, Z. Konjović, B. Milosavljević, M. Vidaković. *Bibliotečki informacioni sistem BISIS ver. 3.01*. Deveta međunarodna konferencija "Informatika u obrazovanju, kvalitet i nove informacione tehnologije", Zrenjanin, 2000. str 494-504.

30. T. Vulić, D. Surla. *Objektno orijentisano modeliranje izveštavanja u bibliotečkom informacionom sistemu*. Zbornik radova YU INFO 2000, Kopaonik, 2000.
31. B. Pupovac, M. Vidaković. *Modeliranje komandi bibliotečkog informacionog sistema u UML-u*. Zbornik radova YU INFO'00, Kopaonik, 2000.
32. B. Milosavljević, M. Zeremski. *Definicija tipa dokumenta UNIMARC formata*. Zbornik radova YU INFO 2000, CD-ROM, Kopaonik, 2000.
33. B. Milosavljević, Z. Konjović. *Arhitektura WWW prezentacije bibliotečkog informacionog sistema*. Zbornik radova YU INFO 2000, CD-ROM, Kopaonik, 2000.
34. Д. Тодорић, И. Холо. *Објектно-оријентисана спецификација корисничког претраживања и извештавања о библиографској грађи*. Зборник радова YU INFO'99, Копаоник, 1999.
35. N. Savić, T. Vulić. *Generator kataloških listića u bibliotečkom informacionom sistemu*. Zbornik radova YU INFO'99, Kopaonik, 1999.
36. B. Milosavljević, D. Surla. *Arhitektura tekst servera za XML dokumente*. Zbornik radova SYM-OP-IS'99, Beograd, 1999. str. 149-152.
37. B. Milosavljević, Z. Konjović. *Specifikacija sistema za indeksiranje UNIMARC zapisa.* Zbornik radova YU INFO'99, Kopaonik, 1999.
38. B. Milosavljević, Z. Konjović. *Specifikacija pretraživanja u tekst serveru za UNIMARC zapise*. Zbornik radova SYM-OP-IS'99, Beograd, 1999. str. 145-148.
39. Д. Тодорић, Д. Сурла. *Функције сервера за узајамну каталогизацију библиографксих записа*. Зборник радова YU INFO'98, Копаоник, 1998. стр. 267-272.
40. T. Vulić. *Specifikacija zahteva podsistema za izveštavanje bibliotečkog informacionog sistema*. Zbornik radova YU INFO'98, Kopaonik, 1998. str. 222-226.
41. T. Vulić, N. Savić. *Implementation of the Cataloguing Cards Repository in the Oracle Environment*. XIII Conference on Applied Mathematics PRIM'98, Igalo, 1998. str. 187-194.
42. T. Vulić, Đ. Obradović. *A language for Formal Description of the Library Information System Reports*. XIII Conference on Applied Mathematics PRIM'98, Igalo, 1998. str. 195-202.
43. M. Vidaković, Z. Konjović. *Implementacija korisničkog interfejsa za pretraživanje bibliografskih podataka u Java okruženju*. Zbornik radova YU INFO'98, Kopaonik, 1998.
44. I. Tričković, I. Holo, B. Pupovac. *Data Model of the UNIMARC Format in Oracle Designer/2000*. XIII Conference on Applied Mathematics PRIM'98, Igalo, 1998. pp. 179-186.
45. D. Todorić, M. Vidaković. *Presentation of the Library Data in Unicode*, Proceedings of the XIII Conference on Applied Mathematics PRIM'98, Igalo, 1998. pp. 161-171.

46. D. Todorić, D. Surla. *Shared Cataloguing Server's Functions*. Proceedings of YU INFO'98, Kopaonik, 1998. pp. 267-272. (in Serbian).
47. B. Milosavljević. *Tekst server bibliotečkog informacionog sistema zasnovan na Oracle ConText Option-u*. Zbornik radova YU INFO'98, Kopaonik, 1998. str. 227-230.
48. I. Holo, T. Vulić. *Arhitektura bibliotečkog informacionog sistema.* Zbornik radova YU INFO'98, Kopaonik, 1998. str. 28-33.
49. I. Holo, B. Pupovac, T. Tošić. *The user interface for data maintenance of the UNIMARC format*. XIII Conference on Applied Mathematics PRIM'98, Igalo, 1998. pp. 43-50.
50. Д. Тодорић. *Клијент-сервер архитектура за преузимање библиографске грађе*. Зборник радова SYM-OP-IS'97, Рафаиловићи 1997. стр. 283-286.
51. T. Vulić. *SGML-based Reporting System for Bibliographic Documents*. XII Conference on Applied Mathematics PRIM'97, Palić, 1997. pp. 193-203.
52. I. Tričković. *Primena Petri mreža u specifikaciji dinamike informacionih sistema*. Zbornik radova INFOFEST'97, Budva, 1997. str. 134-135.
53. I. Tričković. *Primer objektno-orijentisane specifikacije korisničkog interfejsa.* Zbornik radova YU INFO'97, Brezovica, 1997. str. 188-193.
54. T. Tošić, D. Surla. *Indexing and Searching of Documents in Text Server*. Proceedings of SYM-OP-IS'97, Bečići, 1997. str. 291-294.
55. D. Surla, I. Tričković. *Objektno-orijentisane metode za projektovanje informacionih sistema.* Informatika i Informacione tehnologije – Teorija i praksa, Novi Sad, 1997. str. 242-251.
56. D. Surla, I. Tričković. *Objektno-orijentisane metode za projektovanje informacionih sistema.* Informatika i Informacione tehnologije – Teorija i praksa, Novi Sad, 1997. str. 242-251.
57. D. Surla, M. Lomić. *Model normativne baze podataka po UNIMARC formatu*. Zbornik radova YU INFO'97, Brezovica, 01-04. aprila 1997. str. 408-412.
58. B. Milosavljević, D. Surla. *Implementacija tekst servera za indeksiranje i pretraživanje bibliotečke građe u Oracle okruženju*. Zbornik radova Info-Teh'97, Vrnjačka Banja, 1997. str. 452-456.
59. Д. Тодорић, Т. Вулић. *Претраживање библиографске грађе употребом World Wide Web-а*. Зборник радова YU INFO'96, Брезовица, 1996.
60. Т. Вулић, Д. Тодорић. *Формирање извештаја у библиотечком информационом систему*. Зборник радова YU INFO'96, Брезовица, 1996.
61. T. Vulić. *Reporting Subsystem in the Library Information System*. XI Conference on Applied Mathematics PRIM '96, Budva, 1996. str. 451- 458.
62. I. Tričković, D. Surla. *Different Approaches to the Object-Oriented Development of Information Systems*. 11th Conference on Applied Mathematics, PRIM'96, Budva, 1996. pp. 439-450.

63. I. Tričković. *Različiti pristupi objektno-orijentisanoj specifikaciji korisničkog interfejsa*. Zbornik radova YU INFO'96, Brezovica, 1996.
64. D. Todorić. *WWW server for Information Retrieval of Bibliographic Data for SCO UNIX*, XI Conference on Applied Mathematics PRIM'96, Budva, 1996. pp. 421-430.
65. D. Surla, Z. Konjović. *Predlog rešenja bibliotečkog informacionog sistema Univerziteta u Novom Sadu*, Informatika i društvene promene, Zrenjanin, 1996. str. 46-53.
66. D. Surla. *Inverzno inženjerstvo u bibliotečkom informacionom sistemu*, Zbornik radova INFO-TEH '96, Lepenski Vir, Donji Milanovac, 1996. str. 83-87.
67. I. Holo. *Command Language Functional Description of Library Information System*. XI Conference of Applied Mathematics PRIM'96, Budva, 1996. str. 117-126.
68. D. Todorić, T. Vulić. *Data Structure of UNIMARC Standard.* X Conference on Applied Mathematics PRIM'95, Budva, 1995. pp. 309-314.
69. I. Tričković, M. Lomić. *Implementation of Different Fonts in User's Interface at Platform VAX/VMS*. Zbornik radova YU INFO'95, Brezovica, 1995. str. 273-277.
70. I. Tričković. *Implementation of User Interface on the Operating System SCO UNIX*. 10th Conference on Applied Mathematics, PRIM'95, Budva, 1995. str. 291-298.
71. I. Tričković. *Generisanje ekranskih formi na VAX/VMS operativnom sistemu*. Zbornik radova SINFON'94, Zlatibor, 1994. str. 11-15.
72. I. Holo. *Implementacija funkcija za unos podataka bibliotečkog informacionog sistema*. Zbornik radova SINFON'94, Zlatibor, 1994. str. 37-42.

Рад саопштен на скупу националног значаја штампан у изводу (R73)

1. B. Milosavljević, Z. Konjović. *Tekst server bibliotečkog informacionog sistema zasnovan na Oracle ConText Option-u*. Zbornik apstrakata SINFON'97, Zlatibor 1997. str. 7.
2. M. Vidaković. *Implementacija korisničkog interfejsa na bibliotečkom informacionom sistemu u Java okruženju*. Zbornik apstrakata SINFON'97, Zlatibor, 1997. str. 10.
3. И. Холо, Д. Тодорић. *Командни језик библиотечког информационог система*. Зборник проширених сажетака, ИТП'95, Нови Сад, 1995. стр. 54-55.
4. Д. Тодорић. *Спецификација захтева корисничког интерфејса за претраживање библиографске грађе*. Зборник сажетака СИНФОН'95, Златибор, 1995. стр. 3.
5. T. Vulić, N. Savić. *Algoritam za generisanje kataloških listića bibliotečkog informacionog sistema*. Zbornik sažetaka SINFON'95, Zlatibor, 1995. str. 4.

6. I. Tričković. *Implementacija sistema za generisanje ekranskih formi na različitim računarskim platformama*. Zbornik proširenih apstrakata ITP'95, Novi Sad, 1995. str. 94-95.
7. I. Tričković. *Implementacija funkcija za rad sa ulazno/izlaznim uređajima preko biblioteke Curses*. Zbornik apstrakata SINFON'95, Zlatibor, 1995. str. 8.
8. D. Surla, Z. Konjović. *Distribuirani unos podataka u bibliotečkom informacionom sistemu*. Zbornik apstrakata, SINFON'95, Zlatibor, 1995. str. 2.
9. N. Savić, T. Vulić. *Struktura kataloških listića u bibliotečkom informacionom sistemu*. Zbornik proširenih sažetaka ITP'95, Novi Sad, 1995. str. 62-63.
10. M. Lomić. *Softverski paket za generisanje različitih fontova na računarskoj platformi VAX/VMS*. Zbornik proširenih sažetaka ITP'95, Novi Sad, 1995. str. 96-97.
11. M. Lomić. *Kreiranje karaktera za terminal VT 420 na VMS operativnom sistemu*. Zbornik sažetaka PRIM'95, Budva, 1995. str. 78.
12. I. Holo. *Sintaksa komandnog jezika bibliotečkog informacionog sistema*. Zbornik apstrakata SINFON'95, Zlatibor, 1995. str. 23.
13. I. Holo, M. Bender. *Implementacija bibliotečkog okruženja za unos bibliografske građe*. Zbornik sažetaka PRIM'95, Budva, 1995. str. 71.

Одбрањена докторска дисертација (R81)

1. M. Vidaković. *Proširivo agentsko okruženje bazirano na Java tehnologiji*. Doktorska disertacija, Fakultet tehničkih nauka, 2003. http://diglib.ns.ac.yu/ndltd/docs/set2/ndltd344/MinjaDoktorat.pdf
2. B. Milosavljević. *Proširivi sistem za pronalaženje multimedijalnih dokumenata*. Doktorska disertacija, Fakultet tehničkih nauka, 2003. http://diglib.ns.ac.yu/ndltd/docs/set1/ndltd143/xmirs.pdf.

Одбрањена магистарска теза (R82)

1. Г. Будимир. *Контрола квалитета XML библиографских записа*. Магистарска теза, Природно-математички факултет, Департман за математику и информатику, Нови Сад, 2004. http://diglib.ns.ac.yu/ndltd/docs/set3/ndltd409/kontrola_kvaliteta.pdf
2. J. Vidaković. *Modeliranje i implementacija bibliotečkih kataloških listića u XML tehnologiji*. Magistarska teza, Prirodno-matematički fakultet, Departman za matematiku i informatiku, Novi Sad, 2003. http://diglib.ns.ac.yu/ndltd/docs/set2/ndltd335/JovanaMagistarski.pdf
3. В. Мијић. *XML едитор за опис UNIMARC формата*. Магистарска теза, Природно-математички факултет, Департман за математику и информатику,

Нови Сад, 2003.
http://diglib.ns.ac.yu/ndltd/docs/set1/ndltd133/MijicVMagistarskaTeza.pdf

4. М. Зеремски**.** *Моделирање UNIMARC формата у XML технологији.* Магистарска теза, Природно-математички факултет, Департман за математику и информатику, Нови Сад, 2002.
http://diglib.ns.ac.yu/ndltd/docs/set1/ndltd64/ZeremskiMMagistarskaTeza.pdf
5. B. Pupovac. *Modeliranje i implementacija komandi bibliotečkog informacionog sistema*. Magistarska teza, Prirodno-matematički fakultet, Institut za matematiku, Novi Sad, 2000. http://diglib.ns.ac.yu/ndltd/docs/set1/ndltd19/magist.pdf
6. N. Savić. *Sistem za generisanje kataloških listića u bibliotečkom informacionom sistemu*. Magistarska teza, Fakultet organizacionih nauka, Beograd, 1999.
7. Б. Милосављевић. *Текст сервер UNIMARC записа.* Магистарска теза, Факултет техничкх наука, 1999.
http://diglib.ns.ac.yu/ndltd/docs/set1/ndltd6/textsrv.pdf
8. T. Vulić. *Izveštavanje i dokumentovanje u bibliotečkom informacionom sistemu*. Magistarska teza, Prirodno-matematički fakultet, Institut za matematiku, Novi Sad, 1998.
9. M. Vidaković. *Objektno orjentisana specifikacija i implementacija korisničkog interfejsa za bibliotečki sistem*. Magistarska teza, Fakultet tehničkih nauka, Novi Sad, 1998.
http://diglib.ns.ac.yu/ndltd/docs/set1/ndltd7/MinjaMagistarski.PDF
10. D. Todorić. *Uzajamna katalogizacija i korisničko pretraživanje bibliografskih podataka*. Magistarska teza, Prirodno-matematički fakultet, Institut za matematiku, Novi Sad, 1998.
11. I. Holo. *Komandni jezik bibliotečkog informacionog sistema*. Magistarska teza, Prirodno-matematički fakultet, Institut za matematiku, Novi Sad, 1998.
12. I. Tričković. *Primena Petri mreža u specifikaciji dinamike informacionih sistema*. Magistarska teza, Prirodno-matematički fakultet, Institut za matematiku, Novi Sad, 1997.